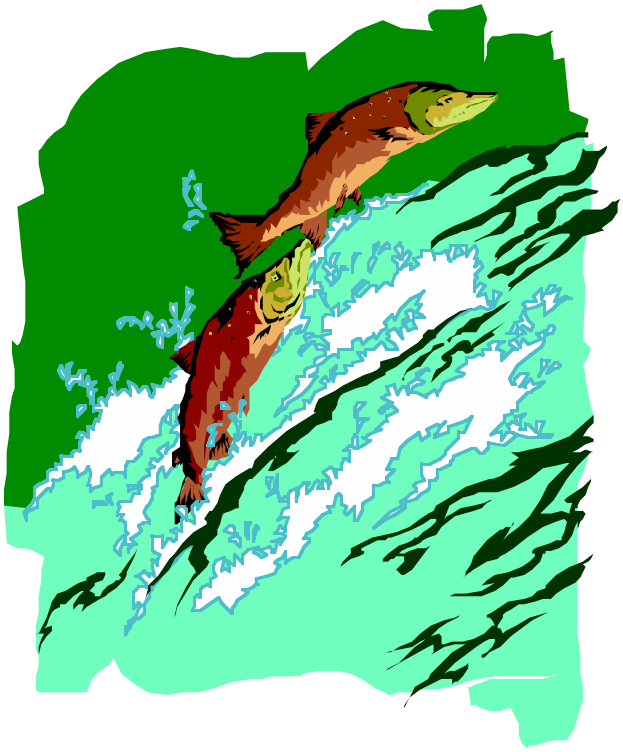


The Friedman-Rubin approach to Cluster analysis

(k-means algorithms for the Macintosh)

Agostino Tarsitano
Dipartimento di Economia e Statistica
Università degli Studi della Calabria
87030 Arcavacata di Rende (Cs)
Italy
Tel. +39-0984-492465
Fax +39-0984-492468

Internet: **agotar@unical.it**

Contents

1. Introduction

The goal of cluster analysis for a given set of data is to verify the presence (or the absence) of natural organization in a fixed number of groups. The data set D consists of n distinct entities $D=\{\boldsymbol{X}_1, \boldsymbol{X}_2,...,\boldsymbol{X}_n\}\subset R^m$ where, for each r, $\boldsymbol{X}_r$ gives the observed values of m real-valued characteristics on the entities which are assumed to be known and fixed.

Relative geometric arrangements, causing concentration and dispersion of the entities in different regions, produce clusters.

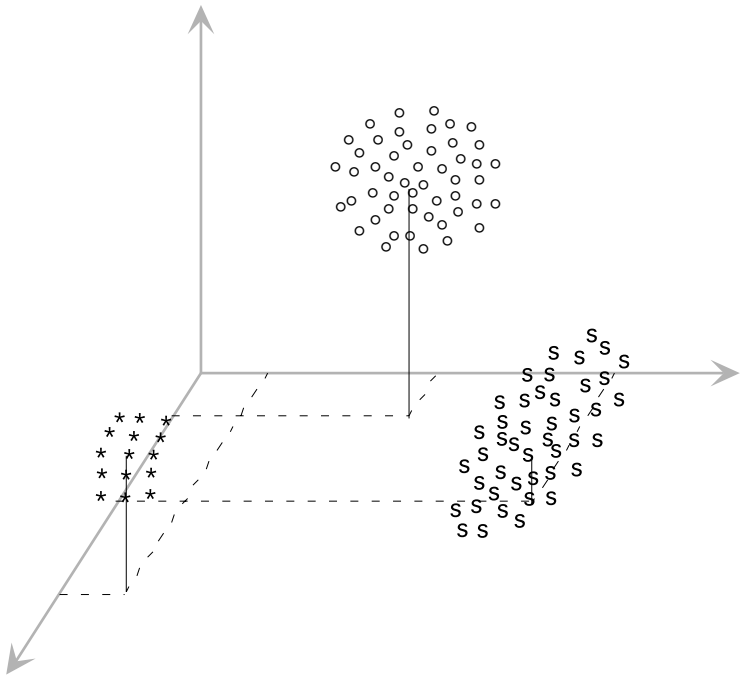


Figure 1: clusters of different shapes and sizes

The figure above depicts a strongly clustered data set consisting of clusters which are homogeneous and well separated. Homogeneity implies that entities in the same cluster are near each other. Separateness implies that entities in different clusters are farther one from the other.

The entities are unlabeled. All we have is a collection of vectors associated with a given set of variables without knowing if the entities belong to different categories, if there is more than one category and the category membership of the entities included in the data set.

1.1 Overview

To learn something from such an unpromising basis depends upon the assumptions one is willing to accept. Suppose that the entities came from a distribution for which the multivariate second moments exist. Then a compact description of the data set can be obtained by the sample mean and the sample covariance matrix.

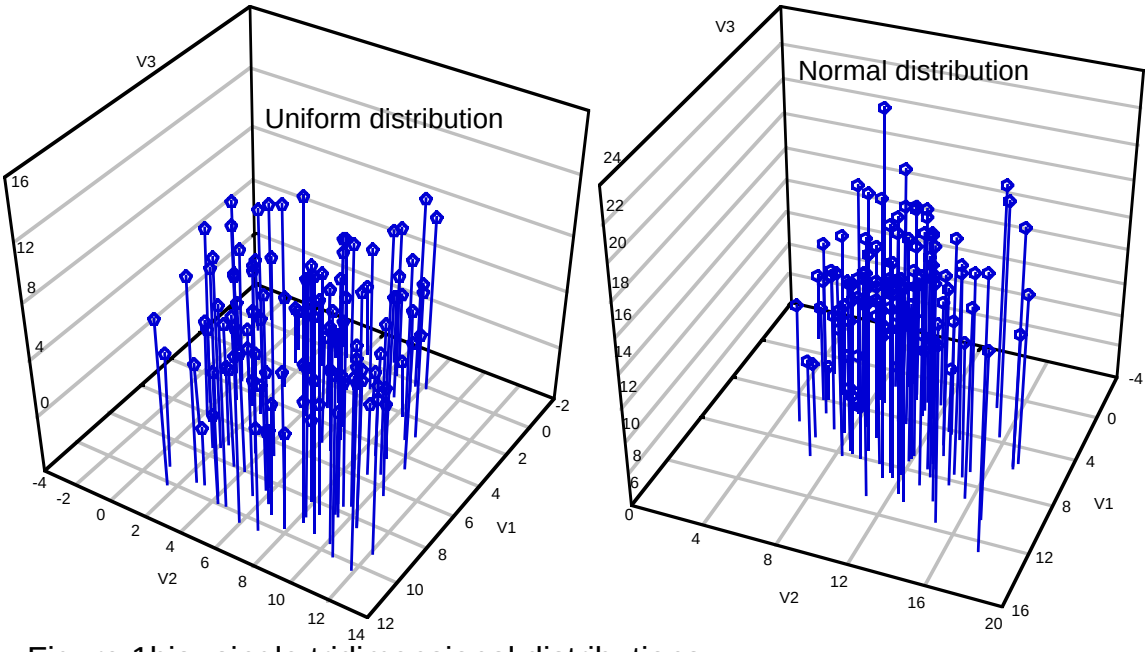


Figure 1bis: single tridimensional distributions

The two graph above represent a sample *of n=120* entities from, respectively, a uniform and a normal 3-dimensional distribution having $\mu=(5,10,15)$ *and* $\Sigma=(\sigma_{ij})$, $\sigma_{ii}=10$, $\sigma_{ij}=5$ *for* $i \neq j$. In general, second-order statistics are incapable of revealing all of the structure in a set of data since other distribution may differ for other important features then mean and covariance. If we assume that entities fall in hyperellipsoidally shaped clouds we can approximate a great variety of situations.

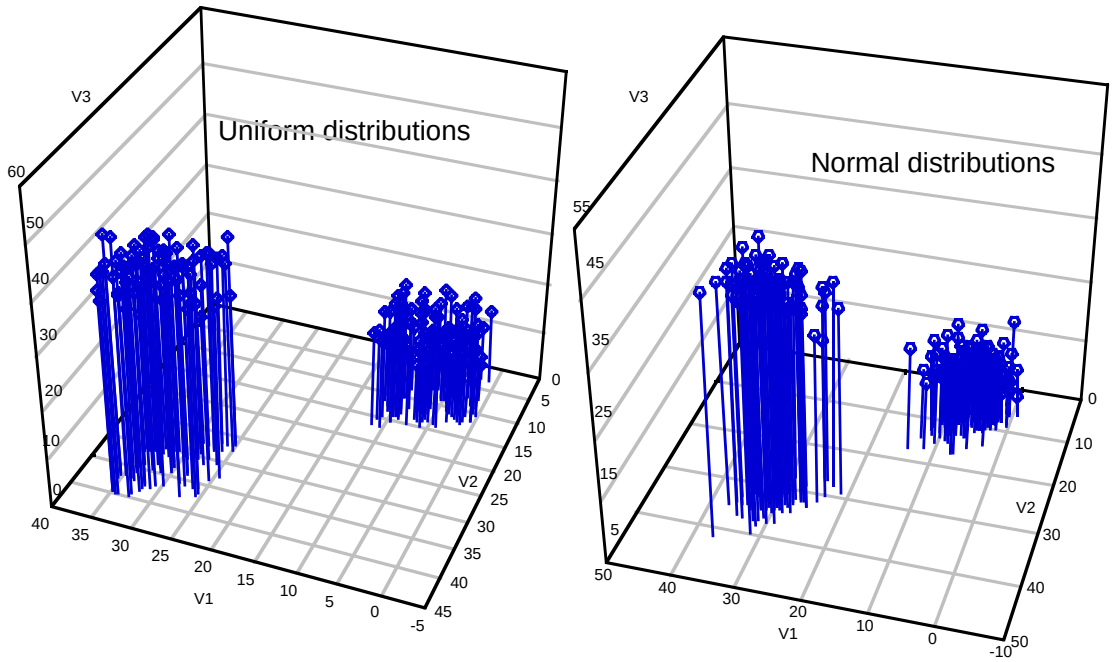


Figure 1ter : two-group tridimensional distributions

Fortunately, the type of approximation we are looking for is not hard to please. The only thing that must be learned is the values of an unknown parameter vector which maps the set of entities to the set of group labels. Figure 1ter illustrates the problem. The two clusters in both graphs have different means and different variance-covariance matrices. If the normal distribution is used to approximate the uniform distribution the results can be very misleading. But, this is not the case. The normal distribution is used for an easier task: distinguishing the entities which fall into the first cluster from the entities falling into the second cluster and this can be fully accomplished even if the approximation of the vector means and the variance-covariance matrices are poorly estimated.

The present paper assumes that the data set has clusters which tend to take the form of hyperellipsoid of various size, but with the same orientations which is the essence of the Friedman-Rubin approach to cluster analysis.

A brief outline of my method of working will help explain the contents of the article. The rest of this section reviews the problem of assessing the partitional adequacy of the subdivision into a fixed number of groups of a data set.

In section 3 a new method for relocative scheme which minimizes the determinant of the sample within-group dispersion matrix is proposed and tested by looking at various real and simulated applications. The main difference with other k-means is a transfer technique which realizes a global best step instead of a local best step.

Section 3 describes the initialization methods. Section 4 outlines the stopping rules and studies their shortcomings and merits in connection with problems arising in practical applications. The software which implements the algorithms is described in section 5.

1.2 The partition of a data set

Seber (1984, p. 379) stated that the major weakness of agglomerative hierarchical methods is the constrain that the (k+1)-partition must be included in the k-partition so that an improper fusion at an early stage cannot be corrected later. On the contrary, the essence of a k-means algorithm is the relocation of entities which gives these techniques an immense advantage over the others. This section reviews the general framework of the k-means algorithm for the subdivision of the data set into a fixed number of mutually exclusive and exhaustive clusters. A partition (or a clustering) γ of a finite set of n entities $D=(\boldsymbol{X}_1,\boldsymbol{X}_2,...,\boldsymbol{X}_n)$ is a collection of k subsets, called the clusters of γ, such that

$$C_j \neq \varnothing,\ 1 \leq k \leq n; \quad \bigcup_{j=1}^{k} C_j = D; \qquad C_i \cap C_j = \varnothing \quad (\ i \neq j\ ); \tag{1}$$

where $\varnothing$ is an empty set. The cardinalities $n_1, n_2, \ldots, n_k$ of the clusters satisfy

$$a)\ 1 \le n_i \le n-k+1,\ i=1,2,\ldots,k \qquad b)\ \sum_{i=1}^{k} n_i = n \tag{2}$$

This implies that each entity is assigned to one cluster, each cluster contains at least one entity and the partition contains all entities.

A partition can be succinctly expressed by the classification vector $\gamma=(\gamma_1, \gamma_2, \ldots, \gamma_n)$ which maps the set of entities to the set of cluster labels $\boldsymbol{X}_r : \gamma_r=j$ if $\boldsymbol{X}_r \in C_j$.
The number of clusters k is assumed to be given as input, although it is often unknown and its estimation is a topical problem in cluster analysis. The program, as such, is not able to merge small clusters that are very close and no larger clusters are broken up. The approach suggested by ***DetClus*** is to run the program for a range of values of k

$$2 \le k_1 \le k_2 \le (n\text{-}k) \tag{3}$$

and to empirically determine the best number of clusters. The upper (lower) limit for k should be at least 3 or 4 more (less) clusters than are ultimately suspected (these limits are necessary since, if k is excessively large or small, spurious or unnatural clusters tend to appear). For each k the program carries out the clustering regardless of the previous grouping and computes a series of clustering quality indices which allow the user to decide the appropriate value (or values) for the number of clusters.

1.3 The definition of a cluster

It is almost a commonplace that there exists no agreed upon idea of a cluster and that, according to the scope of the analysis, different type of clusters are allowed (a typical example of such vagueness is the distinction drawn between natural and arbitrary clusters proposed by Kruskal (1977):*" ... We call clusters natural if the membership is determined fairly well in a natural way by the data, and we call the clusters arbitrary if there is a substantial arbitrary element in the assignment process".*

The k-means approach to cluster analysis is based on a "metric" concept of a cluster. The n entities are confined to an m-dimensional parallelepiped

$$R = \left\{ X | X_i \in \left[x_i^{'}, x_i^{''} \right];\ i=1,2,\ldots,m \right\}; \qquad with \quad x_i^{'} = \underset{1 \le r \le n}{Min} \left\{ x_{ri} \right\},\ \ x_i^{''} = \underset{1 \le r \le n}{Max} \left\{ x_{ri} \right\} \tag{4}$$

Clusters are accumulations of points in distinct regions of *R* entirely surmounted by empty space (see figure 2).

Cluster analysis techniques search partitions characterized by remoteness in which, as was observed by Cormack (1971), two conflicting requirement are involved: internal compactness or cohesion (i.e. object belonging to the same cluster are in some operational sense similar to each other) and external isolation (very dissimilar entities must be placed in different clusters). The two factor are dependent: a highly dense accumulation of points (A) needs less isolation to be considered a proper cluster, and, sometimes, a very sparse group (B) is accepted as a single cluster only for the substantial gap between its entities and the others. The size of the clusters is also important: internal homogeneity tends to be greater for small clusters than for large ones; external isolation has an opposite tendency. According to Ling (1973) a cluster is judged real if it is significantly compact or isolated or both.

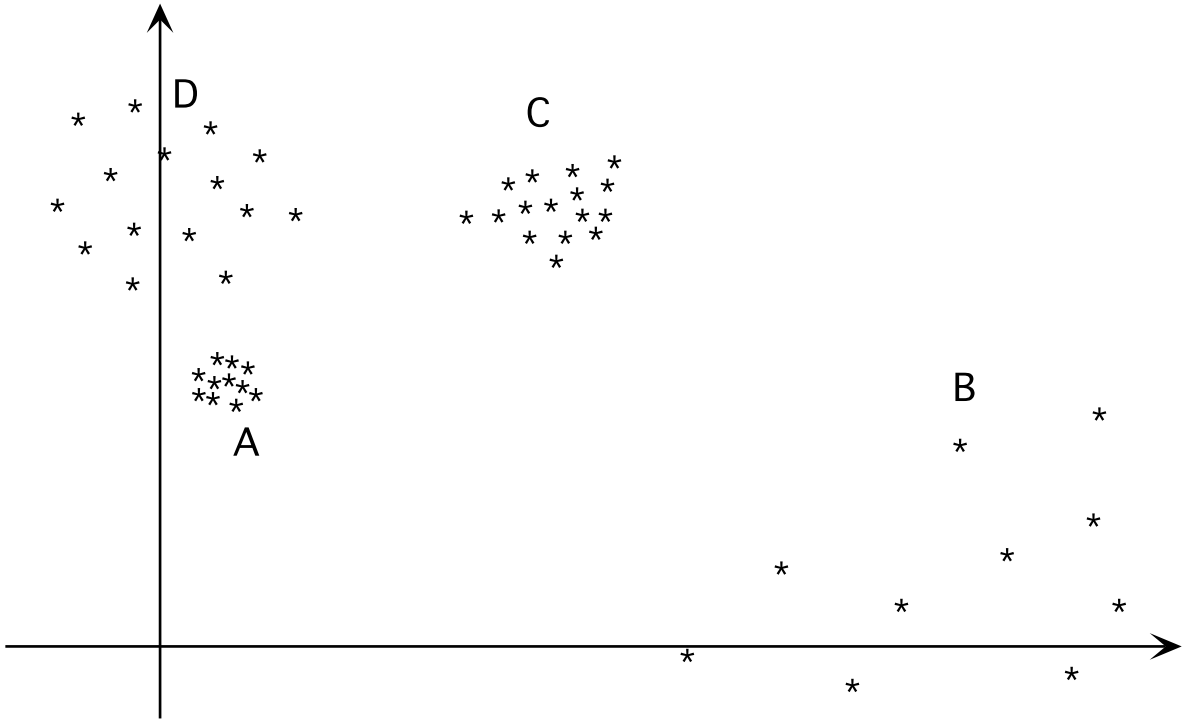


Figure 2: compactness and isolation of clusters

The concept of remoteness is vague as almost is everything referred to in cluster analysis. For instance it is not clear how to deal with the disturbing (but inescapable in real applications) phenomenon of intermediate entities (borderline or hybrid clusters) linking two cohesive and otherwise isolated clusters.

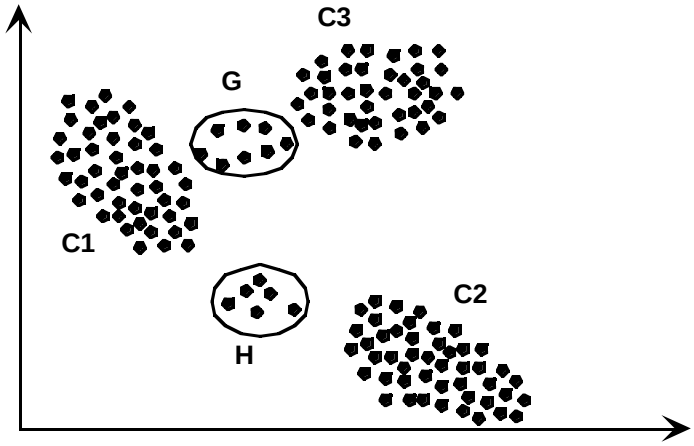


Figure 3a: hybrid clusters

The cluster G in Figure 3a is an object which depicts the transformation process followed by the entities in passing form status C1 to C3 (or *vice versa*). The cluster H is a structure formed by the entities which share the characteristics of both C1 and C2. If the clustering algorithm is such that the entities in the chain cluster G and hybrid cluster H have to be assigned to one of the two major cluster, C1 and C2, their isolation is doubtful.

Cluster Y rises another question. Is such structure shared by enough entities to be considered real and worthy of attention or is it a mere product of random turbulences in data collection? No easy answer exists.

Another somewhat undesirable phenomenon is the presence of outliers or singletons (Figure 3b) that is, clusters formed by a single entity whose distances from the other (n-1) entities are all significant. What is the correct number of clusters? Three (ignoring X), or four (considering X a genuine cluster)? If one considers X a unique case which does not reserve further treatment then the clustering algorithm can run on a reduced dataset from which X has been eliminated. This has the advantage of limiting the number of contenders to whom an entity could be assigned. If X cannot be discarded then this has dramatic effects on the general lookups of the clustering.

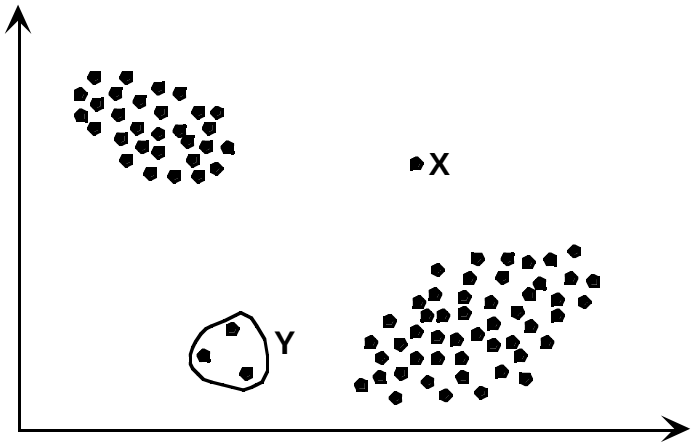


Figure 3b: a singleton and a small cluster

In this sense, the requirement of exclusive assignment 2b is particularly strong because every entity is forced to join a cluster whereas one would ordinarily be inclined to separate out outliers (entities which fits badly into existing clusters) or being just naive or intermediate entities linking two or more otherwise isolated clusters. Applications of k-means to real data should be able to handle "nuisance" entities from further clustering runs (Bayne *et al,* 1980), although such question represents an important research challenge. Since some clustering criterion is very sensitive to the presence of outliers, some attempt should be made to remove these. It is clear that choices made at this stage can have a determining influence on the output of the subsequent analysis.

1.4 Perfect and admissible clustering

Clustering methods have a common intuitive requirement: entities in the same cluster should be closer than entities in different clusters. Rubin (1967) called "well-structured" such partitions. Ideally (Rubin, 1967; Van Rijsbergern, 1970) one would ask that the maximal distance between entities in the same cluster be lower than the minimal distance between entities in different clusters.

Let

$$\Delta_j = \underset{\gamma_r=j,\gamma_s=j}{Max} \left\{ d(\boldsymbol{X}_r, \boldsymbol{X}_s);\ r,s=1,\ldots,n \right\};\ S(i,j) = \underset{\substack{\gamma_r=i,\gamma_s=j \\ i\neq j}}{Min} \left\{ d(\boldsymbol{X}_r, \boldsymbol{X}_s);\ r,s=1,\ldots,n \right\} \quad (5)$$

denote, respectively, the diameter and the moat of the cluster C_j. Two type of ideal clustering can be defined. The first is the “perfect clustering”

$$\Delta_j \ \leq\ S_j \quad j=1,2,\ldots,k \quad (6)$$

An example of perfect clustering is the disjoint partition consisting of each entity in a separate cluster. In this case $\Delta_j = 0,\ j=1,2,\ldots,k$ and $S_j > 0,\ j=1,2,\ldots,n,$ supposing that the X's are distinct. Nonetheless, perfect clustering is too restrictive (Bailey and Dubes, 1982; Tarsitano and Anania, 1995) since it eliminates many reasonable groupings.

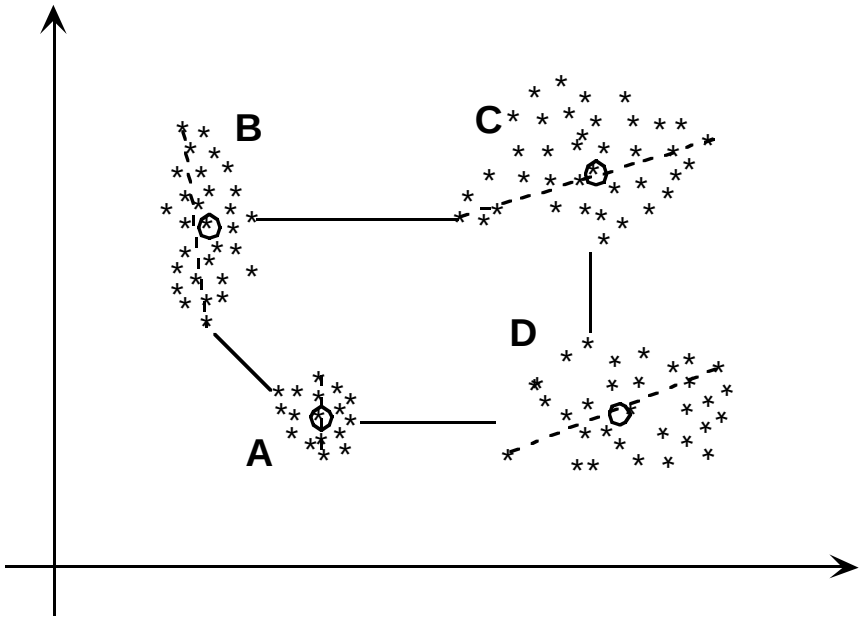


Figure 4: Admissible, but non perfect clustering

A less stringent definition, also useful from a computational point of view, is the string condition (Rao, 1971): in an admissible clustering, each group consists of at least one entity μ_j such that the distance between it and any entity that does not belong to the same cluster is not less than the distance between μ_j and any entity within the same cluster.

$$\underset{\gamma_r \neq j}{Min}\left\{d\left(X_r, \mu_j\right);\ r=1,\ldots,n\right\} \geq \underset{\gamma_r = j}{Max}\left\{d\left(X_r, \mu_j\right);\ r=1,\ldots,n\right\}; \qquad j=1,2,\ldots k \tag{7}$$

According to this definition (see Figure 4), the problem addressed by k-means algorithms is to discover, for each cluster, a representative or typical entity for whom is minimized a known function of the dissimilarities between an entity in the cluster and the centroid.

The centroid can be either a hypothetical entity which is not an entity in the cluster (*e.g.* the vector of the arithmetic mean of all entities currently in the cluster) or an existing entity (*e.g.* the most typical entity that is the entity with the smallest average or total distance between itself and the other entities in the cluster). When centroids are defined the classification vector γ is determined by assigning all entities to the most similar centroid.

The ability of a centroid to summarize the information content of the cluster depends on the actual spread of the data in the given variable space. Usually, the centroids, the cluster membership, and the variance-covariance structure are unknown and must be estimated from the data set. Since each partition provides an estimate of the parameters, some selection is necessary. A comparison can be accomplished using an objective function $L(\gamma): \gamma \in P(n,k) \rightarrow [0,\infty)$ (here γ_r denotes the cluster membership assigned to $\boldsymbol{X}_r$) such that $L(\gamma) < L(\delta)$ means that γ provides better estimates than δ (where $P(n,k)$ denotes the set of partitions of n entities into k clusters). Since the cardinality of $P(n,k)$ is finite, it exists at least one partition γ^* such that

$$L\left(\gamma^*\right) = \underset{\gamma \in P(n,k)}{Min}\left\{L(\gamma)\right\} \tag{8}$$

The most straightforward way to find γ^* is to evaluate $L(\gamma)$ for all $\gamma \in P(n,k)$. It is well known, though, that this is not a viable solution since the cardinality of $P(n,k)$ grows rapidly (it is of the order $k^n/k!$) and becomes prohibitively high even for moderate values of n and k. The k-means partitioning is a "NP-hard problem" for which no a priori guarantee can be given in terms of solution quality and running time. Although the dividing line between things which are practical to compute and things which are not is continuously pushed forward, the search for γ^* must still be conducted over a small subset of $P(n,k)$ using strategies which find solutions that are often good, but not necessarily optimal.

2. Characteristics of a k-means algorithm

There is a wide choice of clustering methods which have different adaptability to the data and different requirements of computer resources. Given an initial partition γ^q with $q=0$, k-means algorithms compute the criterion value $L(\gamma^q)$. Another partition γ^{q+1} is obtained by transferring a single entity (or block of them) from one cluster to another. The transition from $\gamma^{(q)}$ to $\gamma^{(q+1)}$ can be realized by means of up- and down-dating formulae for the exchange of entities between clusters. The new partition is accepted if $L(\gamma^{q+1})<L(\gamma^q)$ and the procedure is repeated until no further reduction of $L(\gamma^q)$ can be obtained.

The algorithm terminates after a finite (typically small) number of iterations. It is worth pointing out that the k-means partitioning is a "NP-hard problem", that is, there is no absolute guarantee in terms of solution quality and running time.

There are many variations of, and extensions to, this approach and the lack of investigations into their properties is, in large measure, due to the excess of options which form an k-means algorithm. The ***DetClus***, as with any program implementing a relocation scheme, has the following essential phases:

0) Feature selection
1) Determining the criterion (distance measure)
2) Starting the process
3) Reassigning entities
    3.1 Distribution of the entities among clusters
    3.2 Updating the centroids, cardinalities and scatter matrices
4) Overcoming local minima
5) Validation of the results
6) Interpretation of the results

The variables must be properly chosen so as to englobe as much information as possible concerning the difference between entities, but, with the minimum number of uncorrelated features. These issues are, however, outside the scope of the present section.

2.1 Determining the criterion

Iterative schemes are concerned with making membership changes which optimize a numerical criterion. The choice of the objective function $L(\gamma)$ is crucial for a K-means algorithm because it must take into account two requirement which may be difficult to reconcile. One goal (internal cluster cohesion) can conflict with another (separation between clusters).

Several criteria have been proposed and each of them is predisposed to finding certain type of clusters and has specific properties. ***DetClus*** is based on the criterion proposed by Fried-

man and Rubin (1967)

$$L(\gamma) = Min\left\{\left|\boldsymbol{W}_{\boldsymbol{q}}(\gamma)\right|\right\}$$

$$\boldsymbol{W}_q = \sum_{j=1}^{k} \boldsymbol{W}_j^q; \quad \boldsymbol{W}_j^q = \sum_{r=1}^{n} \left(\boldsymbol{X}_r - \mu_{\gamma_r}^q\right)\left(\boldsymbol{X}_r - \mu_{\gamma_r}^q\right)^t; \; \mu_j^q = \sum_{\gamma_r = j} \boldsymbol{X}_r \Big/ n_j^q \;, \; j = 1,2,\ldots,k \tag{9}$$

Where $\boldsymbol{W}_q$ is the pooled dispersion matrix across the *k* clusters (or "within-group" dispersion matrix) for the q-th classification vector. In order for (9) to be non-singular, it is required that $(n-k) \geq m$ otherwise the estimate is singular regardless the true value of **W**. Naturally, since total dispersion matrix $\boldsymbol{T}$ is fixed for every partition of the given data set, $Min\{|\boldsymbol{W}(\gamma)|\}$ is equivalent to $Max\{|\boldsymbol{T}|/|\boldsymbol{W}(\gamma)|\}$. A simple variation of (9) was proposed by Symons (1981)

$$L\left[\gamma_q\right] = nLn\left[\left|\boldsymbol{W}\left(\gamma_q\right)\right|\right] - 2\sum_{j=1}^{g} n_j^{(q)} Ln\left(n_j^{(q)}\right)$$

Some empirical results does not support such criterion since relocations based on it stop after surprisingly few iterations.

The minimization of the determinant of $\boldsymbol{W}(\gamma)|$ does not make such restrictive assumptions about the shape of the clusters as does $Min\{\mathrm{Tr}[\boldsymbol{W}(\gamma)]\}$ assuming only that the clusters has the same shape and orientation, but not that they are spherical. Although computationally more involved and expensive, the criterion $Min\{|\boldsymbol{W}(\gamma)|\}$ is invariant under the affine transformations $\boldsymbol{Y}=\boldsymbol{AX}+\boldsymbol{b}$ where **A** is non singular (this allows the question of standardization of the variables to be overcome and the results do not depend on arbitrary factors such as the units of measurement used for data acquisition). Furthermore, it reduces the repetitive effect of several highly correlated attributes by considering sums of cross products in addition to sums of squares (Arnold 1979).

The use of (9) implies that the dissimilarities between the entities are measured by the generalized (Mahalanobis) distances, each centroid coincides with the averages of all entities within the cluster and the clusters have the same variance-covariance matrix. In fact, Mahalanobis distances are equivalent to the Euclidean distances between the transformed entities:

$$\boldsymbol{Y}_i = \boldsymbol{HX}_i \,, \qquad i=1,2,\ldots,m$$

where $\boldsymbol{HH}^t$ is the Cholesky factorization of $\boldsymbol{W}$.

The generalized distance introduced by Mahalobis (1936) is a distance measure corrected in terms of the group structure of the data. Additionally, it is appropriate when the variables are

correlated because it takes into account the variability of the values in all dimensions. The point C in figure 5, which clearly lies in the domain of cluster B would be allocated in cluster A if the Euclidean metric were used. If the within-cluster covariance matrix is known, the data can be transformed $\boldsymbol{Y}_i = \boldsymbol{H}\boldsymbol{X}_i$ to make the clusters spherical so that the Euclidean distance can be used. But when we are doing a cluster analysis, we do not know what the true cluster membership is and we cannot calculate W so that an approximation should be used.

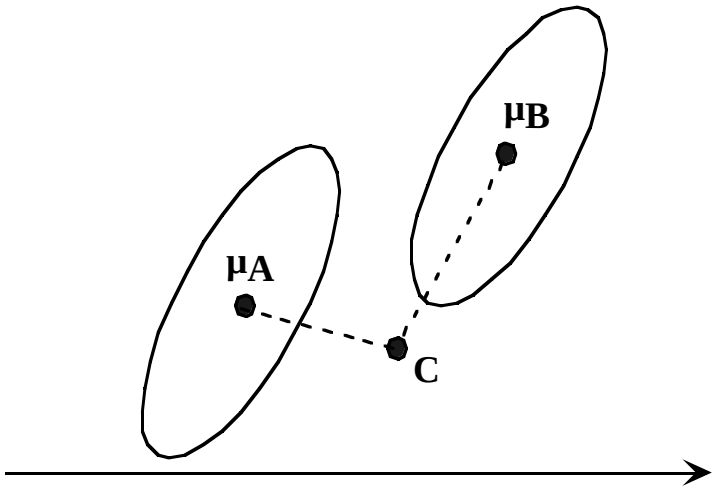


Figure 5: Euclidean vs Mahalanobis distance

Since

$$\left|\boldsymbol{W}(\gamma)\right| = \prod_{i=1}^{m} \lambda_i(\gamma)$$

where λ_i is the i-th eigenvalue of the within-cluster scatter matrix, the criterion $Min\{|\boldsymbol{W}(\gamma)|\}$ tries to minimize the volume of the hypercube defined by the variances in the direction of the m principal axes of the data set. This means that (9) is appropriate when the variables are correlated because it takes into account the variability of the values in all dimensions. However, since the within-group dispersion matrix $\boldsymbol{W}$ is an average of the variance-covariance matrices of the clusters, correlated variables in the clusters generate multicollinearity in $\boldsymbol{W}$. In other words, $|\boldsymbol{W}|$ will approach to zero as correlations grow stronger. The Mahalanobis distance between the centroids, calculated by using $\boldsymbol{W}^{-1}$, tends to infinite; as a consequence, the only clustering compatible with such conditions is the disjoint partition. However, since the within-group dispersion matrix $\boldsymbol{W}$ is an average of the variance-covariance matrices of the clusters, correlated variables in the clusters generate multicollinearity in $\boldsymbol{W}$. In other words, $|\boldsymbol{W}|$ will approach to zero as correlations grow stronger. The Mahalanobis distance between the centroids, calculated by us-

ing $\boldsymbol{W}^{-1}$, tends to infinite; as a consequence, the only clustering compatible with such conditions is the disjoint partition.

In the general case, when the centroids μ_i, $i=1,2,...,k$ and the matrix $\boldsymbol{W}$ are completely unknown, the number of unknown parameters to be estimated equals $km+m(m-1)/2$ and, therefore, reliable estimates are possible only if n is much more greater than this threshold. If $\boldsymbol{W}$ is ill-conditioned and one supposes that the k clusters lie in the same subspace, redundant features can be eliminated by representing the data in a new coordinate system in which the effective description can be given by applying techniques to reduce the dimensionality of the data. An evident technique is to apply Principal Component Analysis and to perform the cluster analysis on the factor scores of the first few leading factors instead of the complete data (the use of PCA as well as factor analysis is contraindicated if each variable is endowed of a useful and independent discriminating power). While this can be helpful for finding clusters it can make results difficult to interpret.

Dimension reduction has, however, many positive implications. Firstly, for the computational effort because reduced data require less storage space and can be manipulate more quickly than the original set of variables. Secondly, a limited set of selected features may alleviate the influence of irrelevant information (features showing little differentiation across the data set or highly correlated with other features) to whom the Mahalanobis norm give the same relative importance as the other variables thus degrading the grouping ability of the most salient features. Third, to avoid implicit weighting: if two collinear features are used, then their common dimension is effectively double weighted (Heeler and Day, 1975). In addition, eliminating redundant variables helps to interpret and compare the configurations derived by cluster analysis. Finally, some validation tests (*e.g.* the C^3 clustering criterion) designed for uncorrelated variables, becomes applicable to orthogonal principal components.

Example 1:

Economics of Cities. (http://lib.stat.cmu. edu/ DASL/).

The data represent the economic conditions in 46 cities around in world in 1991. The variables are: work (weighted average of the number of working hours in 12 occupations), price (index of the cost 112 goods and services excluding rent, Zurich =100), salary: (index of hourly earnings in 12 occupations after deductions (Zurich =100). If all the PC's are used for the calculations the Mahalanobis distances between the point of the figure 6b are equivalent to the Euclidean distance between the points of Figure 6a.

However the appearance of the data sets in the normalized PC space is different from the original space, since now, along each PC, the points have the same variance.

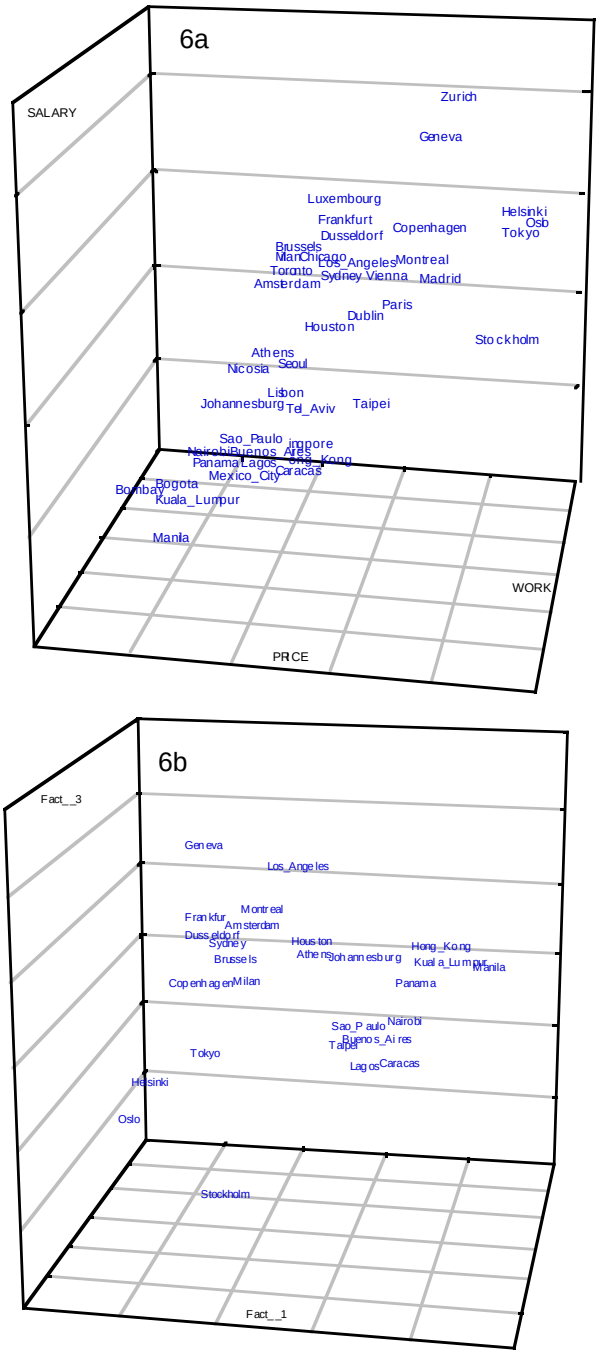


Figure 6: cities data set.
Original an normalized PC spaces

Example 2:

Sexual activity and the lifespan of male fruitflies. Source: “Sexual Activity and the Lifespan of Male Fruitflies” by Linda Partridge and Marion Farquhar. Nature, 294, 580-581, 1981. Size:125 observations, 5 variables: number of partners (0, 1 or 8), Type of companion: 0=newly pregnant female; 1= virgin female:9: not applicable (when partners=0), lifespan, in days, length of thorax, percentage of each day spent sleeping. The first two variables are used as if they were quantitative, although is questionable as to how far these variables can be c as metric.

The pooled within-cluster scatter matrix is singular for any value of k and criterion (9) cannot be applied to this data set. However, the the first four principal components of the correlation matrix (explaining the 93.2% of the total variation) indicate the presence of a group structure although the number of clusters is uncertain. ***DetClus*** run plainly on the reduced data set ensuring a perfect revovery of the five cluster present in the data.

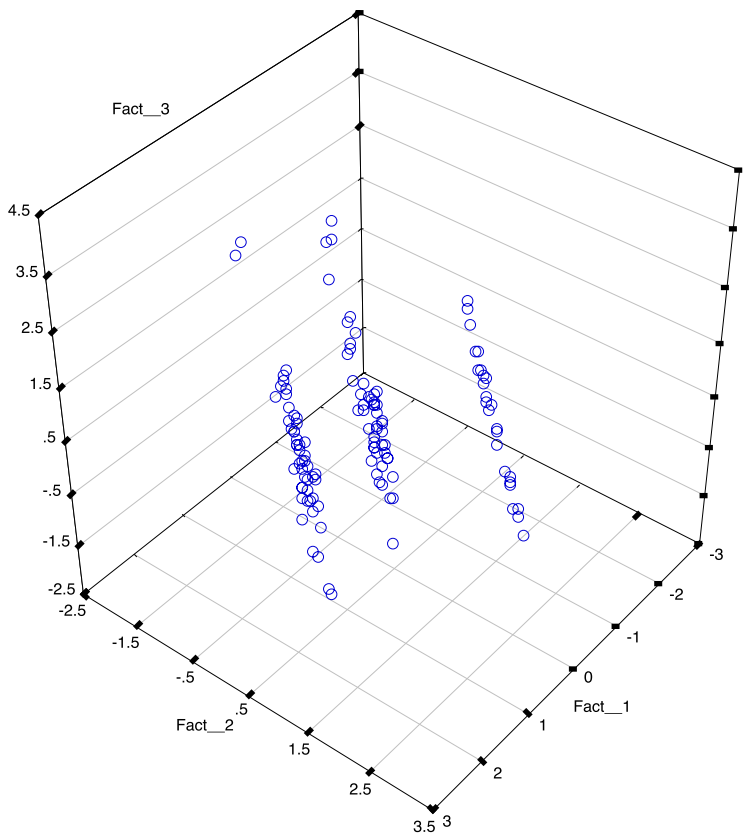


Figure 7: fruitfly reduced data set

2.2 Interpretation of the criterion

Friedman and Rubin relate *Min{|W|}* to Wilks' lambda statistic encountered in the multivariate analysis of variance. In this context, the hypothesis that the means of k normal multivariate distributions with a common dispersion matrix are equal H_0: $\mu_1=\mu_2= ...=\mu_k$ versus H_1: *at least one* $\mu_i \neq \mu_j$ *is available* is tested by considering

$$\boldsymbol{T} = \sum_{r=1}^{n} (\mathbf{X}_r - \mu)(\mathbf{X} - \mu)^t = \sum_{r=1}^{n} \left(\boldsymbol{X}_r - \mu_{\gamma_r}\right)\left(\boldsymbol{X}_r - \mu_{\gamma_r}\right)^t + \sum_{j=1}^{k} n_j \left(\mu_j - \mu\right)\left(\mu_j - \mu\right)^t = \boldsymbol{W} + \boldsymbol{B} \tag{10}$$

where μ is the total mean and $\boldsymbol{B}$ is the "between" dispersion matrix. Specifically, H_0 is rejected if $\lambda=|\boldsymbol{W}|/|\boldsymbol{W}+\boldsymbol{B}|$ is too small. Since $\boldsymbol{W}+\boldsymbol{B}$ is fixed, minimizing the determinant of the within dispersion matrix is equivalent to minimizing the p-value of the Wilk's lambda.

The minimization of $|\boldsymbol{W}|$ searches for clusters that are hyper-ellipsoidal with equal orientations. Everitt (2001), Chernoff (1970), Chen *et al.* (1974), Symons (1981) points out that the metric $\boldsymbol{W}$ could produce incorrect and misleading results when the dispersion structures of the clusters are markedly heterogeneous. In fact, the algorithm is destined to find football-shaped clusters sharing a common orientation even if there were no trace of them in the data set.

Example 3:

Diday and Govaert's data. Fifty observations from each of three bivariate normal distributions, as described in Diday and Govaert, RAIRO Informatique/Computer Sciences, 11, 329-49 (1977). The data have been studied also in Gordon (1999, 46-48). The phenomenon is clearly illustrated in Figure 8 where the minimization of the determinant has driven the algorithm along the axis of maximal dispersion.

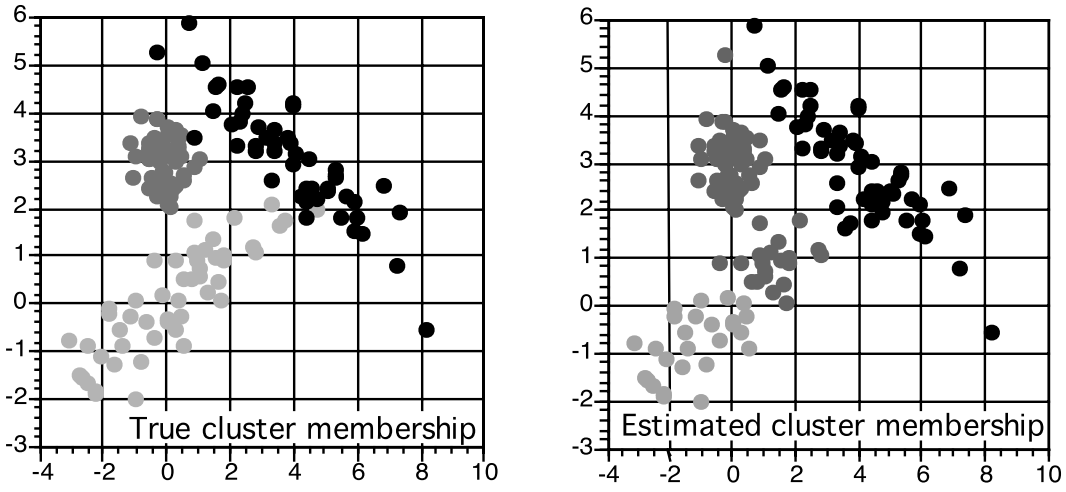


Figure 8: peculiarity of Min{|W|}

Several authors have pointed out that if the condition of homogeneity for the within-group dispersion matrices is not satisfied, the clustering based on $Min\{|\boldsymbol{W}|\}$ may impose artificial structure which precludes uncovering patterns hidden below the surface of data

Scott and Symons derived the criterion within the context of the maximum likelihood estimation of γ assuming a multivariate normal distribution for each clusters. Bayne et al. (1980) found that $|\boldsymbol{W}|$ and $Tr(\boldsymbol{W})$ do not differ significantly. Zemroch (1996) notes that the Mahalanobis distance is attractive because it emphasizes the unusualness of those entities that most defy the intrinsic relationships among the variables. Marriott argues that if one of the variables is strongly grouped, the minimum partition defined by (3) would be entirely based on that variable. Such a partition obeys the string condition, but the clusters might be unrecognizable if viewed against the entire space of the variables. Scott and Symons (1971) conjecture that the criterion encourages the formation of partitions with clusters of equal size which is not necessarily a curse, especially in many experimental designs. Exhaustive experimentation with $|\boldsymbol{W}|$ used jointly with an efficient procedures (see section 3) for determining the initial configuration, does not confirm such a tendency, at least for well structured data set.

Example 4:

Fishcatch data set (available from the data archives of the Journal of Statistical Education). A sample of 157 fishes of 7 species are caught and measured. All the fishes are caught from the same lake (Laengelmavesi) near Tampere in Finland. The best solution for k=7 and k=5 are shown in figure 9. In the first case the data are weakly clustered and the tendency to nearly equal sized clusters 8 column "E") is confirmed. For $k=5$ the group structure is more pronounced and the clustering based on $Min\{|\boldsymbol{W}(\gamma)|\}$recognizes both small and large clusters.

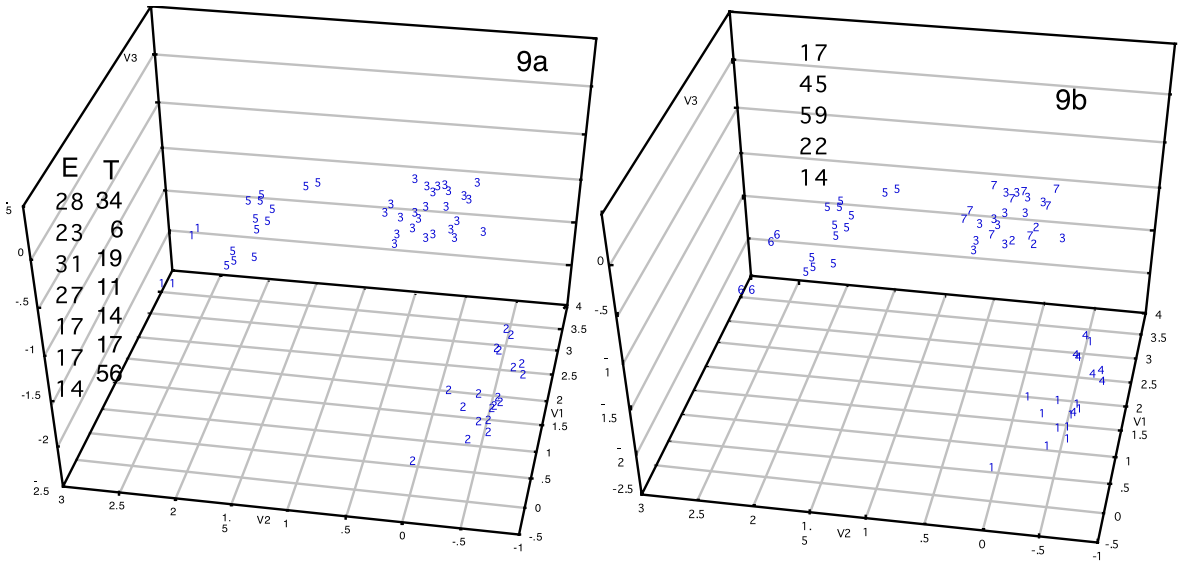
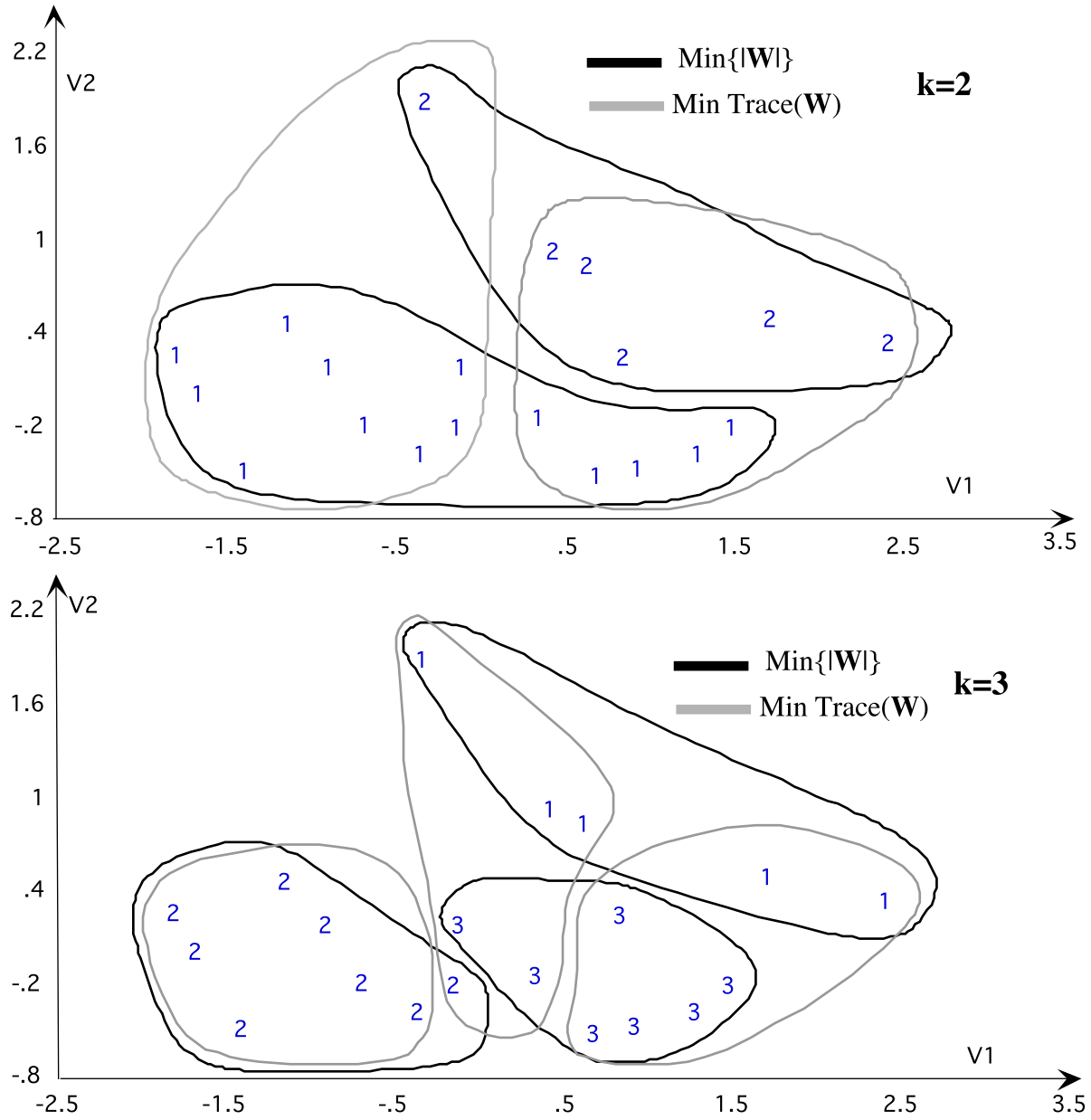


Figure 9: best solutions for k=7 and k=5

Example 5: Duda *et al.* (2001, pp. 543-548) discuss various criterion functions fro clustering by applying the criteria to a simple data set. The raw data does not exhibit any obvious clusters.



For *k=2* the clusters found by minimizing the sum os squared errors (*Tr(**W**)*) tend to favor clusters of roughly equal number of entities; in contrast, *Min*|***W***| favors one large and one fairly small cluster (Bayne *et al.* 1980 found that |***W***| and *Tr(**W**)* do not differ significantly).The clusters in the figures are stretched horizontally because the variation of the data set is greater along the V1 axis than along the V2 axis (the solution found by **Detclus** is different from that presented by Duda *et al*. 2001). For *k=3* the difference between the clusters determined by the two criteria becomes smaller (the first cluster on the left is almost the same). According to *Duda et al.* this is a general tendency.

2.3 Reassigning entities

The essence of a k-means algorithm is the reallocation phase and, in fact, the type of pass is a distinctive feature of the method. There are a number of schemes in common use to relocate entities, each reflecting a different trade-off between classification capability that can be achieved and computer time consumed. Most methods differ basically in the number of criterion evaluations required to reach a minimum and the accuracy of this minimum.

The schemes considered by ***DetClus*** are based on a combination of two distinct stages: transfers and swaps. Transfers consist of moving one entity from one cluster to another; swaps involve the exchange of two entities from different clusters.

Let $|\boldsymbol{W}_{q+1}|$ the determinant of the within-cluster dispersion matrix after that the transfer of $\boldsymbol{X}_r$ from cluster j to i has taken place (the transfer from a singleton is not considered).

$$\Delta_q(r,j,i) = \frac{\left|\mathbf{W}_{q+1}\right|}{\left|\mathbf{W}_q\right|} = \left(1+\alpha_i \mathbf{y}_i^t \mathbf{W}_q^{-1} \mathbf{y}_i\right)\left(1-\alpha_j \mathbf{y}_j^t \mathbf{W}_q^{-1} \mathbf{y}_j\right) + \alpha_i \alpha_j \left(\mathbf{y}_i^t \mathbf{W}_q^{-1} \boldsymbol{y}_j\right)^2$$
$$\alpha_i = n_i^q \Big/ \left(n_i^q + 1\right); \alpha_j = n_j^q \Big/ \left(n_j^q - 1\right); \quad \boldsymbol{y}_i = \mathbf{X}_r - \mu_i^q; \quad \boldsymbol{y}_j = \mathbf{X}_r - \mu_j^q \tag{11}$$

If $\Delta_q(r,j,i) \le \rho < 1$ then $|\boldsymbol{W}_{q+1}| < |\boldsymbol{W}_q|$. This condition ensures that the procedure does indeed produce progressively better partitions. Moreover, since $|\boldsymbol{W}_q|$ is bounded by zero, the process must converge in a finite number of steps. (Obviously it is not the convergence itself, but the rate of convergence that justifies this method in practice). A threshold lower than one (*e.g.* $\rho=0.9999$) prevents cycling divergence (that is, catastrophic recurrence of partitions which were abandoned at an earlier stage) due to numerical problems; additionally, it may help to regulate the running time of the algorithm.

The change in the scatter matrix, its inverse, centroids and cardinalities is easily computed from the following relations

$$\mathbf{W}_{q+1} = \mathbf{W}_q - \alpha_j \mathbf{y}_j \mathbf{y}_j^t + \alpha_i \mathbf{y}_i \mathbf{y}_i^t; \quad 1 - \alpha_j \mathbf{y}_j^t \mathbf{Z}_q^{-1} \mathbf{y}_j \neq 0; \; 1 + \alpha_i y_i^t \mathbf{W}_q^{-1} \mathbf{y}_i \neq 0$$
$$\mathbf{W}_{q+1}^{-1} = \mathbf{Z}_q^{-1} + \frac{\alpha_j \left(\mathbf{Z}_q^{-1} \mathbf{y}_j\right)\left(\mathbf{Z}_q^{-1} \mathbf{y}_j\right)^t}{1 - \alpha_j \mathbf{y}_j^t \mathbf{Z}_q^{-1} \mathbf{y}_j}; \quad \mathbf{Z}_q^{-1} = \mathbf{W}_q^{-1} - \frac{\alpha_i \left(\mathbf{W}_q^{-1} \mathbf{y}_i\right)\left(\mathbf{W}_q^{-1} \mathbf{y}_i\right)^t}{1 + \alpha_i y_i^t \mathbf{W}_q^{-1} \mathbf{y}_i}$$
$$\mu_i^{q+1} = \frac{n_i^q \mu_i^q + \mathbf{X}_r}{n_i^q + 1}; \quad \mu_j^q = \frac{n_j^q \mu_j^q - \mathbf{X}_r}{n_j^q - 1}; \; n_i^{q+1} = n_i^q + 1; \quad n_j^{q+1} = n_j^q - 1 \tag{12}$$

To avoid the accumulation of rounding errors, the quantities are computed directly from the data after a number v of transfers depending on the data set. ***DetClus*** uses $v=200\sqrt{n}*m$.

The sequence of the entities within the data set may exert a profound influence on k-means. An algorithm is said to be combinatorial (MacQueen, 1967) if the criterion, centroids, cardinalities and within-group scatter matrices are updated immediately after a move has been executed in order to take account of the new situation. As a result, the trajectory of the iterative process is dependent, to some extent, on the sequence in which entities are processed and different orderings may yield different clusterings. This problem can be mitigated by randomizing the choice of the entities to be reallocate or by applying data reordering techniques.

In a noncombinatorial k-means algorithm (Forgy, 1965) the moves are executed in parallel in the sense that the entities do not actually change to their new cluster membership until destinations for all entities have been determined. Hence, not only the calculations are substantially simplified, but the iterative process does not suffer from ordering effects. However, unless certain conditions are satisfied (Selim and Ismail, 1984), there is no guarantee of a net improvement in $L(\gamma)$ and no guarantee that the k-means process converges.

A relocation of the entity $\boldsymbol{X}_r$ from cluster i to cluster j causes consequential changes to the centroids μ_i and μ_j: the former is pulled toward $\boldsymbol{X}_r$ and the latter is pushed away from it. This causes the distances from the centroid of other entities in clusters i and j to decrease, such that the criterion is decreased. If $\boldsymbol{X}_r$ is shifted to its nearest cluster centroid but the centroids are not upgraded the combined effects of several moves like this may actually increase the criterion or, worse, the same reallocations are reproposed in two or more successive steps and no further improvement may be obtained by the algorithm.

Another drawback of the Forgy step is that during the relocation phase it is possible that all entities of a cluster are assigned to other clusters and at the same time no other entity is assigned to the centroid of this cluster. In this way the procedure ends up with an empty clusters and the partition is discarded.

In spite of this potential weaknesses, the Forgy approach can generate fast and reliable k-means algorithms which, nonetheless, tend to be less efficient than algorithms implementing the McQueen approach (Anderberg, 1973, p. 166). On the other hand, it has been experimentally observed that algorithms based on a combinatorial scheme are more susceptible of being trapped in local minima. At present, the effect of the choice combinatorial/non combinatorial reassignment of the entities for the criterion (9) has not yet fully established.

Option 1: first improving

The simplest reassigning pass merely consists of scanning -in a random or systematic order- the data set and computing $\Delta_q(r,j,i)$ for $i=1,2,...,k$, $i\neq j$; $r=1,2,...,n$. where the order in which the cluster are tried can also be sequential or random. If $\Delta_q(r,j,i)\leq\rho$ then $\boldsymbol{X}_r$ is immediately reclassified from its present cluster j to cluster i without checking to see if some other transfer would be better. The n entities are then checked in turn to see if another transfer decreases the criterion. For each entity, ***DetClus*** examines at most $(k-1)$ partitions (neighborhood set) derived from the current partition by moving an entity from one cluster to another. It should be noted that when the starting partition is inadequate the "quick" transfers can be slower than more complex searches executed under the other options.

The results of TFI may depend on the sequence in which the entities are processed. If the data sets are formed by compact and isolated clusters, there is a high chance that any arrangement of the data may lead to a global minimum (MacQueen, 1967). Nevertheless, more consistent and reliable comparisons can be performed if the way the entities are selected for the updating phase does not interfere with the minimization process. Peña *et al.* (1999) suggested trying many runs with different arrangements to marginalize out ordering effects, but the number of repetitions deserves further exploration. Fisher *et al*. (1992) argued that arrangements so that consecutive entities are dissimilar lead to good clustering. Further work remains to be done on connections between sorting strategies of the data and recovery rate of combinatorial k-means algorithm based on the determinant. Normally, the transfers are executed in the order in which they appear in the data set, but the flow can be altered by the user. In fact, to reduce the impact of the entity order ***DetClus***, allow randomizing both the choice of the entity to be considered for a move and the choice of the destination cluster. The current configuration of the entities is obtained by shuffling the set of entities. Let $\gamma=(\gamma_1, \gamma_2, ..., \gamma_n)$ be a vector of integers between 1 and n. By using the technique suggested by Knuth (1981, p. 139) random permutations of the γ_r's is considered. In the same way, the current sequence of the destination clusters is determined by shuffling a vector of integer between 1 and k. To alleviate the burden of computations the shuffling of the clusters is performed each five transfers and that of the entities each 20 transfers.

Option 2: local best-improving

A first-improving policy may lead to premature convergence of the k-means process. The transfer algorithm can be more effective if a local search is included between iterations. This motivated the development of several search methods to solve the problem of $Min\{|\boldsymbol{W}(\gamma)|\}$. Rubin (1967) suggested examining the potential effect of switching $\boldsymbol{X}_r$ from the cluster it occupies to each other cluster and finding the value satisfying $Min\{\Delta(r,j,i)|\Delta(r,j,i)\leq\rho,\ i=1,2,...,k;\ j\neq i\}$ Thus each entity is transferred (if transferred) to the cluster which maximizes the impact on $|\boldsymbol{W}(\gamma)|$ of

the transfer. The entities can be considered either in natural or in a random sequence. If such transfer exists then the process is moved from the current partition to the best partition among the *(k-1)* partitions belonging to the neighborhood set. When there is more than one entity whose transfer gives the same decrease of the criterion, the gaining cluster is selected by choosing the transfer with the smallest *i* among the competitors. The search is repeated -using either deterministic or stochastic sequences- for each entity of the data set. It is evident that option 2 is more computer demanding than option 1 since the latter is interrupted if also the former is interrupted, but this may continue to evaluate transfers also when the TFI does not.

Option 3: Best global improving

DetClus performs a complete scanning of the entities and produces the set of candidate transfers $E= \{\Delta(r_h,j_h,i_h)\leq\rho,\ h=1,2,\ldots,\}$. Then the elements of E are sequenced in ascending order and the corresponding transfer executed (provided that $\Delta(r,j,i)\leq\rho$ after each transfer) starting from the first, but discarding those affecting clusters already involved in a reassignment. When there is more than one entity whose transfer gives the same decrease of the criterion, the gaining cluster is selected by choosing the transfer with the smallest *i* among the competitors. The process is iterated until all entities no longer change their membership. Iterations are also stopped if $|\boldsymbol{W}| < 10^{-20}$ to avoid looping and overflowing. Clearly, global strategies are expected to give better results than local ones since an improvement of the local search do not necessarily mean an improvement of the total k-means algorithm. However, global strategies may be questionable under the request of computer resources. In fact, for each pass through the data set ***DetClus*** moves at most *[k/2]* entities which could seem unsatisfactory compared with the number of potential relocations considered by a local search. It should be pointed out, though, that after some initial iterations characterized by quick refinements, local searches tend to settle into sequences of very few and often ineffective moves even when the process is not in the vicinity of a minimum partition. In addition, the results of ***DetClus*** are invariant with respect of the entity order (except when multiple equivalent solution exist), whereas the final solution of combinatorial algorithms incorporating local searches may feel the impact of order dependency.

To avoid array overflow errors the number of transfers between cluster *i* and *j* to be retained should have a fixed upper bound because the number of potential moves (its maximum is *(k-1)n*) could be greater then the available temporary storage). ***DetClus*** considers a maximum of 1'124'250 moves. The option of retaining only the best transfer for each entity although parsimonious in terms of memory storage and execution time, has been proved much less efficient than considering all the transfers (allowed by the memory size of the program).

.

2.4 Swapping entities

Banfield and Bassil (1977) proposed that the interchange of cluster membership between entities is a useful tool for reassigning entities. ***DetClus*** offers the opportunity of a mixed scheme alternating transfers with swaps.

Consider the swap of entity $\boldsymbol{X}_r$ with $\gamma_r=i$ and entity $\boldsymbol{X}_s$ with $\gamma_s=j$, $i\neq j$. The effect on the dispersion matrix is

$$\mathbf{W}_{q+1} = \mathbf{W}_q - \beta(\mathbf{X}_r - \mathbf{X}_s)(\mathbf{X}_r - \mathbf{X}_s)^t + (\mathbf{X}_r - \mathbf{X}_s)(\mu_j - \mu_i)^t + (\mu_j - \mu_i)(\mathbf{X}_r - \mathbf{X}_s)^t \qquad (13)$$

where $\beta=(n_i+n_j)/(n_i n_j)$. The inverse of (13) and its determinant can be computed by repeated applications of the Sherman-Morrison formula.

$$\begin{aligned} &|\boldsymbol{W}_{q+1}| = |\boldsymbol{W}_q| * \Delta(r,s,j,i) = |\boldsymbol{W}_q| * \left(1-\beta \boldsymbol{f}^t \boldsymbol{W}_q^{-1} \boldsymbol{f}\right)\left(1+\boldsymbol{g}^t \boldsymbol{B}_q^{-1} \boldsymbol{f}\right)\left(1+\boldsymbol{f}^t \boldsymbol{A}_q^{-1} \boldsymbol{g}\right) \\ &\boldsymbol{f} = (\mathbf{X}_r - \mathbf{X}_s),\ \boldsymbol{g} = \left(\mu_j^q - \mu_i^q\right) \end{aligned} \qquad (14)$$

$$\boldsymbol{B}_q^{-1} = \boldsymbol{W}_q^{-1} + \frac{\beta\left(\boldsymbol{W}_q^{-1}\boldsymbol{f}\right)\left(\boldsymbol{W}_q^{-1}\boldsymbol{f}\right)^t}{1-\beta \boldsymbol{f}^t \boldsymbol{W}_q^{-1}\boldsymbol{f}};\ \boldsymbol{A}_q^{-1} = \boldsymbol{B}_q^{-1} - \frac{\left(\boldsymbol{B}_q^{-1}\boldsymbol{f}\right)\left(\boldsymbol{B}_q^{-1}\boldsymbol{g}\right)^t}{1+\boldsymbol{g}^t \boldsymbol{B}_q^{-1}\boldsymbol{f}};$$

$$\boldsymbol{W}_{q+1}^{-1} = \boldsymbol{A}_q^{-1} - \frac{\left(\boldsymbol{A}_q^{-1}\boldsymbol{g}\right)\left(\boldsymbol{A}_q^{-1}\boldsymbol{f}\right)^t}{1+\boldsymbol{f}^t \boldsymbol{A}_q^{-1}\boldsymbol{g}};\ \mu_i^{q+1} = \mu_i^q + n_i^{-1}\boldsymbol{f};\ \mu_j^{q+1} = \mu_j^q - n_j^{-1}\boldsymbol{f} \qquad (15)$$

For each scan of all possible interchanges between different clusters ***DetClus*** implements the swaps (if any) which most reduce the criterion, provided that no cluster is involved in more than one swap and that $\Delta(r,s,j,i)\leq\rho$ after each swap. This condition ensures that the procedure does indeed produce progressively better partitions. Since the criterion $Min\{W(\gamma)\}$ corresponds to a sum of squares, the process of relocating only those entities which yield a reduction must converge because a sum of squares cannot be indefinitely reduced.

As was previously noted, the k-means algorithm can be interrupted after the first improvement found in the neighborhood set or after examining the whole neighborhood set. In the former case, a maximum of $n(n-1)/2$ candidate partitions are evaluated while, in the latter, exactly $n(n-1)/2$ alternatives must be analyzed. In the first case, an order dependency may be introduced which can be ameliorated randomizing the choice of the pairs to be swapped.

Option 1: Post processing stage

With this option the user activates a hybrid process oscillating between the transfers stage and the swaps stage. The whole data set is reprocessed until there is no further improvement in the quality of the clustering by means of a transfer; only then the swaps stage is executed repeatedly for all pairs of entities until a new convergence occurs. If one or more swaps are found beneficial, then the transfers stage is restarted. The iterations continue until the membership of the clusters stop changing. The hybrid scheme denoted as option 1 should be essentially considered as a way of overcoming a local minimum. The swopping, is a heuristic technique in the sense that its failure to produce a better solution does not mean that the actual partition is the best. However, it reinforces our confidence in it.

Option 2,3. Mixed strategies: transfer+swaps

The transfers are applied for the first pass across all entities then the swaps for the second, and proceed in this fashion until a minimum of the criterion is reached. Banfield and Bassil (1977) considered a single search of the *n(n-1)/2* pairs of entities although further repetitions (after recomputing the centroids) could led to better partitions.

Option 4,5. mixed strategies: swaps+transfers

In this case the swaps are used for the first stage then transfers for the second and continue oscillating until there are no entities that change their cluster membership. The mixed strategies should help in applying k-means with inadequate starting partitions.

The swapping pass (options 2-4) can be combined with the transferring pass (option 1-3) generating 12 mixed schemes: TFI+SFI, TFI+SGBI, TBLI+SFI, TLBI+SGBI, TGBI+SFI, TGBI+SGBI, SFI+TFI, SGBI+TFI, SFI+TLBI, SGBI+TLBI, SFI+TGBI, SGBI+TGBI. The pure schemes: TFI, TLBI, TGBI reprocess the whole data set and terminates when there are no entities that change their cluster membership. The mixed schemes have two distinct alternating strategies: either the transfers are applied for the first pass across all entities then the swaps for the second, and proceed in this fashion until a minimum of the criterion is reached or the swaps are used for the first stage then transfers for the second and continue oscillating until convergence occurs. In both cases, mixed schemes suffer from ordering effects, with the exception of TGBI+SGBI and SGBI+TGBI.

2.5 Simulation results

Tarsitano (2002) has analyzed and compare 17 different relocation methods for the k-means algorithm implementing the Friedman-Rubin criterion (given that the number of natural clusters is known and the order of entities within the data set is fixed).

The key findings are listed below.

1. The scheme TGBI, unexpectedly, scores top marks in terms of convergence speed significantly better than any other scheme. In this sense, it is a natural candidate for clustering large data sets, at least for applications where a reasonably good initial classification is available.

2. The mixed schemes are uniformly less rapid than pure schemes and the difference between execution times reaches a maximum -as it should be suspected- when the globally best transfer is coupled with the globally best swap. On the other hand, when swaps are performed, TGBI+ are faster than TLFI+ which are, in turn, faster than TFI+. The same ranking is found for the tandems lead by SFI and for those lead by SGBI. The durations of TGBI+ and SGBI+ are higher than any other mixed scheme by orders of magnitude. It is evident that the swapping stage is a time-consuming task because it compares an entity with the entire data set. Worth of note is that the results of combinations S+T compares favorably with those of the reverse combinations T+S. Banfield and Bassil (1977) have ignored mixed methods of the type S+T which, on the contrary, seems to generate efficient schemes.

3. Coleman *et al.* (1999) argue that a TFI strategy seems to be preferred to a TLBI strategy for the problem of classification to minimize the determinant criterion. Ismail and Kamel (1984) indicate that TLBI is more susceptible to being trapped at a local minimum than TFI, at least for algorithm guided by $Min\{Tr(\boldsymbol{W}(\gamma))\}$. On the other hand, Zhang and Boyle (1991) found that TFI and TLBI are indistinguishable. These findings were not confirmed by my experiments. For k-means algorithms based on $Min\{|\boldsymbol{W}(\gamma)|\}$, TGBI outperforms all the other methods, regardless the number of variables, the number of clusters and the structure of the cardinalities. The inclusion of a global search determining a chain of reassignments each of which is the best taken from among the available reassignments is generally beneficial for improving both the rate of convergence and the accuracy of the final partition. Moreover, TGBI is indifferent to the order of data whose influence on other schemes is complex and unpredictable. For medium sized data sets the algorithm runs quite efficiently. Huge data sets are precluded because the large values of nm would require excessive computer resources.

4. The mixed schemes T+ obtain (but non always) some refinement of the final partition over the respective pure schemes. Similar results are found for SFI+ and SGBI+. Nonetheless the impact of the swaps over the quality of the solution is limited and the time needed for each convergence may not be worth the extra computation. Mixed schemes are more likely to work better for poor starting conditions, but the limited impact on the classification adequacy does not compensate the extra energy expended for these procedures. The experiments indicate that com-

binations of different strategies may provide significantly less good performance than do their isolate application. In particular, the swaps, not only are time consuming, but also tends to block the process after very few iterations. In practice, the swaps should be essentially considered as a way of getting out of a local minimum.

5. Schemes of the type S+ tend to yield better solutions in terms of stability and accuracy than T+. Most likely the phenomenon is due to the major ability of the swaps to use more productively the fact that most of the changes in cluster membership occurs at the first few iterations (Anderberg, 1973, p. 163)

6. For data sets divided into even clusters, the recovery rate is steadily higher than for disparate sized clusters and the differences becomes more pronounced as the number of clusters increases. This is aligned with the conjecture that $Min\{|W(\gamma)|\}$ encourages the formation of partitions with clusters of equal size if the separation between the clusters is not large (Scott and Symons, 1971; Everitt *et al.* 2001, p. 94).

7. An interesting point is that the dimension of the problems and the number of clusters did not affect the convergence of either of the algorithms implemented in **DetClus.**.

Overcoming local minima

The problem of IPM's is that the local minimum γ^* may not be the global minimum. Rubin (1967) remarked that two type of problems cause local minima:

1) Two homogeneous but unrelated clusters are united while other clusters may be well formed;
2) The centres of the clusters do not allow a very stable classification of hybrid entities.

The first situation affers directly with the problem of the number of cluster and will be discussed in section 4. **DetClus.** attempts to circumvent local minima due to the second situation by swaps. The swopping, as many other techniques for overcoming a local minimum, is heuristic in the sense that its failure to produce a better solution does not mean that the actual partition is the best. However, it reinforces our confidence in it. It must be said that the swopping phase, rarely provides an improvement and may be ignored if the algorithm starts from a good configuration.

Limitations

An inherent limitation of the k-means algorithms included in **DetClus** is that their final configuration does not necessarily coincide with one of the desired global minima. Since all the schemes do only descent moves, they are not able to force the process out of the current valley and eventually fall into a deeper one. The development of mixed algorithms which combine the best elements of the transfers/swaps with a non descent technique would be a significant contribu-

tion. Additional work is needed to determine the most appropriate strategy of alternating transfers and swaps and to keep the algorithms from taking too many iterations in regions where insufficient progress is being made. For instance, hybrid processes which oscillate between a trasferring stage which reprocess the whole data set until there is no further improvement in the quality of the clustering and only then a swapping stage is used repeatedly for all pairs of entities, should help in applying k-means with inadequate starting partitions. Furthermore, two-phase strategies in which a first-improving transfer pass is applied if the entity number is odd otherwise a best-improving pass is executed (or *vice versa*) can be considered (either as isolate application or combined with swaps) to devise a better k-means algorithm. Moreover, strategies in which the swopping phase is done periodically or randomly could be devised.

The performances of iterative partitioning methods are mainly affected by the intensity of the clustering. The procedures described in this section are all appropriate when the clusters form essentially compact clouds that are fairly well separated from one another. If the clusters are close to one another (even by outliers or hybrids), or if their shapes are not hyper-ellipsoidal, the results of clustering can vary quite dramatically. In fact for poorly defined clusters the misclassification rate reaches unacceptable levels even if the method is valid and consistent with the data-generating process. Furthermore, as Mineo (1986) pointed out, it is more difficult to determine a good starting point and, as a consequence, the algorithm is more likely to stop on local minima

3. Initialization methods

The k-means algorithms described in the previous section converge finitely to a partition γ^* that is locally minimal for $|W(\gamma)|$. The convergence is deterministic given the initial configuration, but the quality of the minimum is not guaranteed. The efficacy of a k-means algorithm is influenced by many factors. Most obvious is the starting partition. In fact, k-means algorithms have differential recovery rates depending on the quality of the starting configuration. So far no attempt has been made to set up a procedure that works well on every occasion.

The reason for this is simply that what is most appropriate for one data set may not be so for another and, unfortunately, there is no simple, universally good solution to this problem (Duda et al, 2001, p. 550). In some case it is possible to obtain excellent results by taking the first k entities as typical representatives; in others, only sophisticated and computationally expensive methods may provide an initial partition acceptably close to the final solution.

Furthermore, the concept of "best" is a compromise between accuracy and computation cost which, for this reason, cannot lead to an initialization method that outperforms all the others on all the data sets. Many available procedures invite the user to have a hieratic confidence in a built-in initialization method (no further details) which regularly finds good clusters provided that they exist and the user gives the correct number of clusters to detect. Such a guarantee cannot be given.

As the cluster analysis has evolved, a a wide variety of techniques has emerged for choosing the first k centroids (or, alternatively, for specifying an appropriate starting partition $\gamma^{(0)}$). Anderberg (1973), Hartigan (1975 Blashfield *et al.* (1982), and Peña *et al.* give a brief summary of a number of different procedures by which an iterative partitioning could be triggered and more can be found (*e.g.* Mineo,1985, Al-Daoud and Roberts,1996).

If an inadequate initialization is performed two puzzling phenomena tend to appear. First, the algorithm may be interrupted at a lower value of the criterion not corresponding to a greater recovery rate (Coleman *et al.*, 1999). Second, the minimum partition found may not be unique as other partitions may give the same criterion value which, in addition, may be associated with a different degree of clustering effectiveness. There is little chance to avoid these problems because the surface defined by $|W(\gamma^0)|$ is usually flat and contains many local minima.

Repetition of the procedure with different partitions appear to be a reasonable method to face this problem. Moreover, it can give good indication of the sensitivity of the final solution γ^* to the starting partition. It should be emphasized, though, that $|W(\gamma^0)| < |W(\delta^0)|$ does not necessarily imply that $|W(\gamma^*)| < |W(\delta^*)|$. Therefore, a user is advised to try several initialization methods on a given data set. In fact, **DetClus** uses each starting partition as a separate basis for the subsequent phases and the classification vector corresponding to the lowest value of $|W(\gamma)|$ is chosen as final clustering. It hardly need adding that the search of the initial configuration takes very much longer then the entire algorithm (the problem is even serious when n, k, and m are large). However, the advantages in terms of partitional adequacy of the final solution far outweigh the consumption of computer time. Of course, multiple restarts may be incompatible with large data sets, at least for the actual technology of the combination hardware/software.

DetClus determines γ^0 by trying several effective techniques which can be classified in two categories: deterministic and random.

3.1 Deterministic techniques.

The deterministic techniques yield an initial partition which is unique in that it is found optimizing a suitable objective function. The partition for which ***DetClus*** obtains the best results is written in the output file as optimal solution for the given value of k.

3.1.1 Best among naive methods

This command calls three different procedures characterized by rapid movements of the entities and quick computations. To avoid generating unfeasible partitions all clusters with no entities assigned to them receive a randomly chosen entity from the largest cluster.

Option 1: mean entity

Hartigan (1975, p. 88) proposed a quick initial clustering based on the simple arithmetic mean of the variables for each entity. ***DetClus*** uses a weighted average of the variables which standardizes the variables by their sample variances. In particular, the observed value of the *m* variables for the r-th entity is summarized by

$$S_r = \sum_{j=1}^{m} w_j x_{r,j}; \qquad w_j = \frac{\sigma_j^2}{\sum_{i=1}^{m} \sigma_i^2}, \quad j = 1,2,\ldots,m \tag{16}$$

where σ_j^2 is the sample variance of X_j. The r-th entity is assigned to the cluster C_i if

$$\left[(k-1)\left(\frac{S_r - S_{min}}{S_{max} - S_{min}} \right) + 1 \right] = i = \gamma_r, \quad r = 1,2,\ldots,n \tag{17}$$

Option 2: leading component

Hartigan (1975, p.102). Let w_j for *j= 1,2,...,m* be the factor loadings of the first principal component of $(n\text{-}1)^{-1}\boldsymbol{T}$ and let

$$S_r = \sum_{j=1}^{m} w_j x_{rj}, \quad r = 1,2,\ldots,n \tag{18}$$

The cluster membership is given by (17). Of course, if the variables have been expressed as factor score, the rule (17) applies to the first variable of the transformed data set. It must be said that the averaging features applied by option 1 and 2 could destroy information contained in the multivariate data.

Option 3: quantiles

The ordered scores S_r, *r= 1,2,...,n* of the dominant factor of $(n\text{-}1)^{-1}\boldsymbol{T}$ are divided into *k* slices with approximately the same number of entities. The membership of the entities is determined according to the rule

$$\gamma_r = j, \ \text{if} \ \ S_{\left(\frac{j-1}{k}\right)} \le S_r \le S_{\left(\frac{j}{k}\right)}; \ S_{0.0} = S_{min}, \ \ S_{1.0} = S_{max} \tag{19}$$

~~where $i = [nt + 0.5]$; $\alpha = nt + 0.5 - i$~~

3.1.2 Best among built-in techniques (simple methods)

These methods are considered "simple" because they perform a single pass through the data set, but require an estimate $\boldsymbol{W}^*$ of $\boldsymbol{W}$. Since the cluster membership is unknown before the analysis some approximate procedure must be used (see section 3.1.3)

Option 1: Sequential splitting

Let g be the number of clusters already formed and let h and i indicate, respectively, the cluster and the variable where the coefficient

$$CD = \frac{2\sqrt{\dfrac{\sum_{\gamma_r=h}\left[x_{ri}-\mu_{ih}\right]^2}{n_h}}}{\left[\left|L_{ih}\right|+\left|U_{ih}\right|\right]};\quad L_{ih}=\underset{\gamma_r=h}{Min}\left\{x_{ri}, i=1,\ldots,n_h\right\};\quad U_{ih}=\underset{\gamma_r=h}{Max}\left\{x_{ri}, i=1,\ldots,n_h\right\} \tag{20}$$

is higher. Index (20) increases as the relative variability increases and it is immune from standardization bias; moreover, the denominator does not vanish unless $X_{ri} \cong 0$ (in this case $CD=0$). The denominator of CD is not an average since its value may fall outside the sample range. Suppose that the current number of entities in cluster C_h is $n_h > 1$ and that X_i is not constant in C_h. Then cluster C_h can be splitted as follows

$$\gamma_r^0 = \begin{cases} g & if \quad x_{ri} \le M \\ g+1 & if \quad x_{ri} > M \end{cases};\quad \gamma_r = h;\quad M = \underset{1\le t<n_h}{Max}\left\{\frac{\left[\sum_{i=1}^{t} x_{(t)i}\right]^2}{t}+\frac{\left[\sum_{i=t+1}^{n_h} x_{(t)i}\right]^2}{n-t}\right\} \tag{21}$$

Formula (21) maximizes the between-group sum of squares for the i-th variable (Engelman and Hartigan, 1969; Anderberg, 1973, pp. 45-46). The split separates the cluster of points above the mean μ_i from the cluster of points below the mean. Centroids are then computed for each cluster by averaging coordinates of its members. In practice, a Forgy step though the data is executed, that is the entities do not change to their new cluster membership until all assignment have been evaluated. The assignment of the entities to the clusters is based on Mahalanobis distance with metric $(\boldsymbol{W}^*)^{-1}$. At the end of step the cluster centroids are updated to be the averages of entities contained within them. No further iterations are performed. Splitting continue until $g+1=k$. A clusters that has less than one entities as its members id discarded.

Option 2: ordered distance from the total mean

This procedure was proposed by Hartigan and Wong (1979). The entities are first sorted by their distances to the overall mean vector μ of the data set; then, the cluster centroids P_j, $j=1,2,\ldots,k$,

are chosen to be the entities labelled *1+(j-1)b, j=1,2,...,k* with *b=[n/k]*. The classification vector γ is derived according to

$$\gamma_r = j \;\; if \;\; \left(X_r - P_j\right)' W^{*-1} \left(X_r - P_j\right) \le \left\{\left(X_r - P_i\right)' W^{*-1} \left(X_r - P_i\right)\right\};\;\; i = 1,2,\ldots,g \tag{22}$$

Formula (22) implies that an entity which need to be assigned to one of the clusters is identified with the cluster to which it is closest as judged by the Mahalanobis distance based on the metric W^*. If an entity is at the same distance from several centroids it is by convention assigned to the cluster C_j with the smallest index *j* among the competitors.

Option 3-6: farthest neighbor

These procedures consist of 3 steps.

Step_1. Determine the first centroid P_1. Two alternatives may be considered:

a) the first centroid is the entity which is nearest to μ, the mean vector of the data set.

$$P_1 = X_s \Rightarrow \left(X_s - \mu\right)' W^{*-1} \left(X_s - \mu\right) \le \left(X_r - \mu\right)' W^{*-1} \left(X_r - \mu\right) \;\; r = 1,2,\ldots,n \tag{23}$$

b) the first centroid is the entity which has the greatest distance from μ.

$$P_1 = X_s \Rightarrow \left(X_s - \mu\right)' W^{*-1} \left(X_s - \mu\right) \ge \left(X_r - \mu\right)' W^{*-1} \left(X_r - \mu\right) \;\; r = 1,2,\ldots,n \tag{24}$$

Step_2. Let $P=\{P_1, P_2, ..., P_g\}$ be the current set of centroids. The (g+1)-st centroid may be chosen according two alternative rules

$$P_{g+1} = X_s \Rightarrow \underset{P_j \in P}{Min} \left(X_s - P_j\right)' W^{*-1} \left(X_s - P_j\right) \ge \underset{P_j \in P}{Min} \left(X_r - P_j\right)' W^{*-1} \left(X_r - P_j\right) \tag{25}$$

$$P_{g+1} = X_s \Rightarrow \underset{P_j \in P}{Min} \sum_{j=1}^{g} \left(X_s - P_j\right)' W^{*-1} \left(X_s - P_j\right) \ge \underset{P_j \in P}{Min} \sum_{j=1}^{g} \left(X_r - P_j\right)' W^{*-1} \left(X_r - P_j\right) \tag{26}$$

Step_3. Execute a Forgy steps through the data set *i.e.* the changes caused by each entities are accumulated and executed at the end of the cycle. The classification vector is determined by assigning all entities to the most similar centroid. Only one complete pass through all the entities is executed. The assignment of the entities to the clusters with the nearest centroid is based on (22). Step_2 and Step_3 are repeated until *k* centroids have been selected.

Methods implementing the farthest-neighbor policy have the advantage of ensuring that extreme entities appear in the initial configuration, but have the drawback of including as centroids atypical entities such as outliers which are unduly emphasized by the Mahalanobis norm.

DetClus uses the following labels:

Option_3: (total mean, maximum distance) *Option_4: (total mean, mean distance)*
Option_5: (farthest entity, maximum distance) *Option_6: (farthest entity, mean distance).*

3.1.3 Preliminary estimation of the within-clusters matrix

Art *et al.* (1982) proposed an algorithm to compute an estimate of **W** without knowing the cluster structure but assuming that the clusters have different means and a common covariance matrix. The standard multivariate analysis decomposition (10) can also be made in terms of pairwise differences:

$$\frac{1}{n}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\left(\boldsymbol{X}_i-\boldsymbol{X}_j\right)\left(\boldsymbol{X}_i-\boldsymbol{X}_j\right)^t=\underbrace{\frac{1}{n}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\left(\boldsymbol{X}_i-\boldsymbol{X}_j\right)\left(\boldsymbol{X}_i-\boldsymbol{X}_j\right)^t}_{Within}+\underbrace{\frac{1}{n}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\left(\boldsymbol{X}_i-\boldsymbol{X}_j\right)\left(\boldsymbol{X}_i-\boldsymbol{X}_j\right)^t}_{Between} \tag{27}$$

$$=W^*+B^*$$

The first term on the right side of (27) involves all the pairs which belongs to the same clusters, and the second term involves all the distance measurement occurring between those pairs where one entity comes from cluster *i* and the other entity comes from cluster *j* with $i \neq j$. No explicit indication is made to the classification vector. The left sides of (10) and (28) are equal. Therefore $\mathbf{T}=\mathbf{W}^*+\mathbf{B}^*$. Under normal sampling assumptions, with $X_{ij} \tilde{} N(\mu_i, \Omega)$ the expected values of $\boldsymbol{W}$ and $\boldsymbol{W}^*$ are

$$E(\boldsymbol{W})=(n-k)\boldsymbol{\Omega}, \quad E\left(\boldsymbol{W}^*\right)=\left(\frac{\sum_{i=1}^{k} n_i^2-n}{n}\right)\boldsymbol{\Omega} \tag{28}$$

Hence $\boldsymbol{W}$ and $\boldsymbol{W}^*$ can be used to construct an unbiased estimate for $\boldsymbol{\Omega}$, but $\boldsymbol{W}^*$ gives relatively more weight to large clusters than does $\boldsymbol{W}$. Naturally, since the cluster structure is unknown neither $\boldsymbol{W}$ nor $\boldsymbol{W}^*$ can be computed. The initialization of an iterative partitioning, however, requires something of less stringent and even a a crude estimate of $\boldsymbol{\Omega}$ can be very helpful. Generalizing the idea of Art et al (1982) a first approximation to $\boldsymbol{W}^*$ can be obtained by

$$\boldsymbol{W}_{(1)}^*=\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\delta_h\left(\boldsymbol{X}_i-\boldsymbol{X}_j\right)\left(\boldsymbol{X}_i-\boldsymbol{X}_j\right)^t; \quad h=j+(i-1)n-\frac{i(i+1)}{2} \tag{29}$$

$$with \quad \delta_h = \begin{cases} f\left[\left(\boldsymbol{X}_i - \boldsymbol{X}_j\right)^t \boldsymbol{M}^{-1}\left(\boldsymbol{X}_i - \boldsymbol{X}_j\right)\right] & if \quad h \le q \\ 0 & if \quad h > q \end{cases}; \quad \delta_h \ge 0, \ \sum_{h=1}^{q} \delta_h = 1 \tag{30}$$

where $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$ are among the closest q pairs in terms of the metric $\boldsymbol{M}$. The weight function is such that $f'(x)<0$ *for* $x>0$, that is, the weight d_b is a non increasing function of the distance between the pair *(i,j)*: the larger the distance is the smaller is the weight attached to the pair. Then (n-1)/2 possible pairs of entities need not to be sorted as long as it can be established that $h>q$ or not. The integer q is chosen conservatively small to avoid contamination by between-cluster pairs.

Next $\boldsymbol{W}^*_{(2)}$ is formed in the same manner except that a new squared generalized distance is used to define the coefficients d_h, that is $\boldsymbol{M}=\boldsymbol{W}^*_{(1)}$

$$\delta_h = f\left[\left(\boldsymbol{X}_i - \boldsymbol{X}_j\right)^t \left[\boldsymbol{W}^*_{(1)}\right]^{-1}\left(\boldsymbol{X}_i - \boldsymbol{X}_j\right)\right] \tag{31}$$

The algorithm continues in a like manner until the process stabilizes, which it usually does rather quickly.

The estimation procedure is controlled by the following parameters:

1) The first metric. Art *et al.* (1982) used $\boldsymbol{M}=\boldsymbol{I}$, that is the first allocation is made by using Euclidean distance, although $\boldsymbol{M}=\boldsymbol{T}$ seems a more plausible choice when the data consist of a number of variables measured in different scales and $\boldsymbol{T}$ is well-conditioned. Using the total covariance matrix as the first estimate, while simple and obvious, also ends up ignoring possible clusters in the data. Another plausible choice is $\boldsymbol{M}=\boldsymbol{V}=diag(v_1, v_2, ..., v_m)$. It should be noted that choosing a diagonal is justified only when the variables are uncorrelated or weakly correlated. If this fact is not taken into account, the measure of closeness of the entities suffers.

2) The number of pairs. Art *et al.* (1982) and Gnanadesikan *et al.* (1993) suggested $q=(n/3)(n/k-1)$ neglecting the number of dimensions. More reasonable values can be found in the range $4[m^2+m(2k-1)] \le q \le (n-k)(n-k-1)/3$.

3) The weights. Art *et al. used* $d_h=1/q$ which have, undoubtedly, some advantages from a computational point of view. In fact, the sorting of the distances is not necessary because it is sufficient to determine the smallest q distance, and these may be unsorted if the scope is their unweighted sum. However, the partial sorting involved in this approach has an high cost in term of storage and the gain in execution time is irrelevant. Moreover, the heapsort suggested by Art *et al.* has a mean time which is inferior to the recursive quicksort implemented by **DetClus**.

After some experiments the following formula has given better results

$$\delta_h = \frac{\alpha(1-\alpha)^h}{1-(1-\alpha)^{q+1}}, \quad h = 1, 2, \ldots, q, \quad 0 < \alpha < 1$$

where *h* indicates the h-th closest pairs.

4) The measure of closeness. Art *et al.* defined

$$\varepsilon_{i+1} = trace\left[\left(\boldsymbol{W}^{*}_{(i)}\left[\boldsymbol{W}^{*}_{(i+1)}\right]^{-1} - \boldsymbol{I}\right)\left(\boldsymbol{W}^{*}_{(i)}\left[\boldsymbol{W}^{*}_{(i+1)}\right]^{-1} - \boldsymbol{I}\right)\right] \tag{32}$$

and convergence is considered satisfactory if $\varepsilon_{i+1} \leq \varepsilon$. An alternative measure is the Jeffreys divergence

$$\varepsilon_{i+1} = \frac{1}{2} trace\left[\left(\boldsymbol{W}^{*}_{(i)}\left[\boldsymbol{W}^{*}_{(i+1)}\right]^{-1} - \boldsymbol{I}\right)\left(\left[\boldsymbol{W}^{*}_{(i)}\right]^{-1}\boldsymbol{W}^{*}_{(i+1)} - \boldsymbol{I}\right)\right] \tag{33}$$

which allow us to measure the distance between two hypothesis $\boldsymbol{W}=\boldsymbol{W}_{i}$ *vs* $\boldsymbol{W}=\boldsymbol{W}_{i+1}$ in the case of a multidimensional sample stemming from one of two schemes relative to normal distribution in R^{m}. However, (32) is less computer demanding than (33).

DetClus implements the procedure with **M=I**, $\delta_{h} = 1/q$, $q = min\{5[m^{2} + m(2k_{2} - 1)], n(n-1)/2\}$ and (32) with $\varepsilon = 0.001$. Iterations are also stopped after 30 iterations.

The fact that W* needs a multiplicative constant to make it an unbiased estimator of W is not relevant since a clustering based on $\boldsymbol{W}^{*}$ is invariant with respect of the transformation $\boldsymbol{W}^{*} = a\boldsymbol{W}^{+}$ with $a > 0$. The main drawback of this procedure is that becomes inapplicable for large data sets both for the storage and for the sorting of the distances. For $n > 1500$ ***DetClus*** chooses the closest q pairs in a random sample (with replacement) of pairs of size 1'124'250. The weights are given by (32) with $\alpha = 0.0001$. The procedure is stopped after ten iterations or if $\varepsilon_{i+1} \leq \varepsilon$.

A similar method for obtaining an estimate of W is available in the Acelus procedure implemented in the SAS procedure **Fastclus.** However, if the population clusters have very different covariances matrices the procedures outlined above is of no avail.

3.1.4 Best among built-in techniques (elaborate methods)

Under this command are comprised four procedures which are computational expensive in that consider, repeatedly, the Mahalanobis distance among all the pairs of entities. None of them is practical when it comes to solving large problems as all of them can become prohibitively expensive even with present-day high-speed computers solution. Moreover, some of them require a large amount of space for storage purposes.

Option_1: complete link centroids

Kennard and Stone, (1969) proposed a sequential method to select initial centroids having as even a spread as possible over the variable space. The first two tentative centroids are selected by choosing the two entities that are farthest apart

$$P_1 = X_s,\ P_2 = X_t \Rightarrow \left(X_s - X_t\right)' W^{*-1}\left(X_r - X_w\right) \geq \left(X_r - X_w\right)' W^{*-1}\left(X_r - X_w\right) \tag{34}$$

The entities are assigned to the nearest cluster according to (22). Then a Forgy pass is applied for the reassignment of all entities until all entities. Let $P=\{P_1, P_2, ..., P_g\}$ be the current set of centroids. The (g+1)-st leader is chosen according to (25). The procedures continue until $g=k$.

Option_2: average link centroids

Same as option 1, but the (g+1)-st leader may be chosen according to (26). This alternative was suggested by Sadocchi (1977).

Option_3: representative istances

Kaufman and Rouseeuw (1990). The first centroid is the most typical member of the data set, that is, the entity P_1 such that

$$\sum_{r=1}^{n}\left(X_r - P_1\right)' W^{*-1}\left(X_r - P_1\right) \leq \sum_{r=1}^{n}\left(X_r - X_s\right)' W^{*-1}\left(X_r - X_s\right);\ s=1,\ldots,n \qquad (35)$$

Let $P=\{P_1, P_2, ..., P_g\}$ be the set of the current centroids. A new centroid is chosen among the not yet selected entities according to

$$\begin{aligned} &P_{k+1} = X_r \ \Rightarrow \sum_{j=1}^{n} C_{jr} \geq \sum_{j=1}^{n} C_{ji},\ i=1,2,\ldots,n;\ C_{ji} = Max\left\{D_j - d_{ij};0\right\} \\ &d_{ij} = \left(X_i - X_j\right)' W^{*-1}\left(X_i - X_j\right);\ D_j = \underset{P_s \in P}{Min}\left\{\left(X_j - P_s\right)' W^{*-1}\left(X_i - P_s\right)\right\} \end{aligned} \qquad (36)$$

The procedures continues until $g+1=k$. By construction, each cluster has at least one entity. Peña *et al. (1999)* state that (36) chooses as leaders the entities that promise to have around them a higher number of other entities.

Option_4: divisive analysis (Di.Ana)

An iterative divisive technique is applied. In practice the Diana algorithm of Kaufman and Rousseeuw (1990, ch. 6) has been extended to rectangular matrices.

The essence of this methods consecutive partition into clusters. Initially, set $C_1=D$. **DetClus** searches for the entity X_r which has the largest average Mahalanobis distance $d(X_r,C_1)$ from all other entities belonging to cluster C_1. The entity X_r is discarded from C_1 and considered the first entity of the new cluster C_2. Let $d(X_s,C_1)$ and $d(X_s,C_2)$ be, respectively, the average distance from the entities in C_1 and the average distance in C_2. For X_s, $s=1,2,\ldots,n$ is left in C_1 if $d(X_s,C_1) < d(X_s,C_2)$ otherwise is moved to C_2. If $k>2$ then the cluster with the largest diameter (5) is splitted until k clusters have been created.

3.2. Random procedures

The random technics generate an initial partition which is independent of the data set. in particular, a pseudorandom sample of ν partitions is considered and the algorithm run for every single partition.

The size ν is crucial. A larger set of test partitions will make it more likely that $\gamma^{(0)}$ is near to γ^*, but will also increases the time taken to carry out the search. As an example, the well-known package MIKCA constructed by McRae (1971) starts by analyzing $\nu=3$ different set of randomly chosen leaders; Symons (1981) selected the initial solution from among $\nu=32$ randomly-generated partitions. Casgrain (Le Progiciel R v4.0d6, 2001) has a default value of 100 for ν. Späth (1985, p. 155) criticized heuristic and more elaborate methods for finding a single "good" starting partition and preferred repeating (in his examples, for 20 times) the entire process choosing at random the initial configuration. These values are too small to be really useful. Peña et al. (1999) used $\nu=1000$ initial partitions which is perhaps too large for many data sets. If our objective is to find a partition that is in the top α% of $P(n,k)$ and we test a random sample without repetitions of ν partitions belonging to $P(n,k)$ then the probability of getting such a partition is $p=1-(1-\alpha)^{\nu}$ which implies $\nu=[Ln(1-p)/Ln(1-\alpha)]$. If $\alpha=0.01$ and $p=0.99$ then there is a better than 99% chance that $\nu=458$ will provide a partition which lies in the top 1% of $P(n,k)$. Of course, the top percentile may include highly unsatisfactory partitions.

In a sense $[\sqrt{(nmk)}]$ represents a reasonable compromise between the accuracy of the preliminary search and the duration of a computer run. It hardly need adding that the search of the initial configuration takes very much longer then the entire algorithm (the problem is even serious when n, k and m are large). However, the advantages in terms of partitional adequacy of the final solution far outweigh the consumption of computer time.

In **DetClus** the number of partitions to be tried is supplied by the user (the default value is $\left[\sqrt{nmk_2}\right]$).

3.2.1 Random points methods

These method sample the space of the variables determining a centroid as a random point in the convex hull defined by the observed values of the variables.

Option_1:

Anderberg (1973, p. 157) suggested the following method to determine the first centroids. Let L_i and U_i represent the minimum and maximum values of the i-th variable for the given data set. Then $R_i=U_i-L_i$ is the sample range of X_i.

The coordinates of the leader P_j for the i-th variable are given by

$$m_{ij} = L_i + u_{ij} R_i; \quad i = 1,\ldots,m; \;\; j = 1,\ldots,k \tag{37}$$

where u_{ij} is a uniform random number from *[0,1]*. The starting classification vector is determined according to (22).

Option_2:

The total mean of the data set $\mu=(\mu_1,\mu_2,\ldots,\mu_m)$ is chosen as reference point a randomly perturbed to define the centroids of the clusters. More specifically, the coordinates of the k m-dimensional centroids are given by

$$m_{ij} = \begin{cases} \mu_i + u_{ij}(U_i - \mu_i) & if \;\; e_{ij} < 0.5 \\ \mu_i - u_{ij}(\mu_i - L_i) & if \;\; e_{ij} \geq 0.5 \end{cases}; \qquad i = 1,\ldots,m; \;\; j = 1,\ldots,k \tag{38}$$

where u_{ij} and z_{ij} are independent uniform random number from (0,1).

The difficult with these schemes is that the resulting centroids are different estimates of the total mean vector and their separateness is questionable. Moreover, unless the data set "fills" the m-dimensional space, some of the centroids may be quite distant from any of the entities and the clusters built around them will have no members. This problem can be attenuated by considering more centroids and eliminating the candidates that are too close. To this end, ***DetClus*** generates *4k* candidates and, among these, selects the best k centroids by applying the Kennard-Stone procedure of section 3.1.2 (*option5)* but ignoring the Forgy step which would be scarcely useful in this context. All clusters with no entities assigned to them receive a randomly chosen entity from the largest cluster

3.2.2 Random permutation of representative values

The range of each variable X_j, *j=1,2,...,m* is divided into k group. With the i-th group associate a value m_{ij} and imagine that each entity put in the i-th group is given the value m_{ij} for the j-th variable. Then we have a matrix (kxm) of representative values which express the peculiarities of the data set.

Consider a random integer $1 \leq s \leq k^m$ and convert s into the subscript vector $\boldsymbol{I}=(i_1,i_2,\ldots,i_k)$ with $1 \leq i_h \leq k$, *h=1,2,...,k* (see O'Flaherty and MacKenzie,1982).

Then the i-th coordinate of g-th centroid is defined $P_{gj} = m_{i_g,j}$ for *j=1,2,...,m*. Finally, a random

samples without replacement of k vector $\boldsymbol{I}$ from the set of k^m possibilities (cf. Bissell, 1976) is generated to define the k centroids. The initial classification vector γ^0 is obtained by applying (22). The number of repetitions v of this procedure is specified by the user. If $v > C(k^m,k)$ then all the combinations are considered as candidate block of centroids.

Option 1: uniform distributions

The values of each variable are arranged in ascending order and divided into k blocks. The first $(k-1)$ blocks include $n_j = b = [n/k]$, $j = 1,2,\ldots,k-1$ whereas the remaining $n_k = n-(k-1)b$ entities are allocated to the last block. Suppose that $n_0 = 1$ and let m_{ij} be the partial mean of the block

$$m_{ij} = \frac{\sum_{r=n_{i-1}}^{n_i} x_{(r),j}}{n_i}; \quad i = 1,2,\ldots,k; \quad j = 1,2,\ldots,m \tag{39}$$

Option 2: partial medians

It is similar to the first option, but the centroids (31) are replaced by the medians of the blocks.

$$m_{ij} = x_{\left[n_{j-1}+0.5\left(n_j - n_{j-1}\right)\right],i}; \quad i = 1,2,\ldots,m; \quad j = 1,2,\ldots,k \tag{40}$$

Option 3: Gaussian distributions.

For $n \to \infty$ with one Gaussian variable the cut points for the optimal partition of a data set into $k=2,3,\ldots,6$ clusters have been computed by Cox (1967) under the condition that

$$\sum_{i=1}^{k} p_i \left(\frac{\mu_i - \mu_j}{\sigma_j} \right)^2 = maximum \tag{41}$$

where μ_j and σ_j are, respectively, the mean and the standard deviation of the j-th variable, μ_i is the i-th conditional mean of X_j given $L_i \le X_j \le U_i$, $i=1,2,\ldots,k$ and p_i denotes the probability of an observation falling in the i-th group. I have extended the work of Cox for $2 \le k \le 25$.

k											
2	(-∞,0)	(0,∞)									
3	(-∞,-0.61)	(-0.61,0.61)	(0.61,∞)								
4	(-∞,-0.98)	(-0.98,0)	(0,0.98)	(0.98,∞)							
5	(-∞,-1.24)	(-1.24,-0.38)	(-0.38,0.38)	(0.38,1.24)	(1.24,∞)						
6	(-∞,-1.45)	(-1.45,-0.66)	(-0.66,0)	(0,0.66)	(0.66,1.45)	(1.45,∞)					
7	(-∞,-1.61)	(-1.61,-0.87)	(-0.87,-0.28)	(-0.28,0.28)	(0.28,0.87)	(0.87,1.61)	(1.61,∞)				
8	(-∞,-1.75)	(-1.75,-1.05)	(-1.05,-0.50)	(-0.50,0)	(0,0.50)	(0.50,1.05)	(1.05,1.75)	(1.75,∞)			
9	±0.22	±0.68	±1.20	±1.87							
10	0.00	±0.40	±0.83	±1.32	±1.96						
11	±0.01	±0.41	±0.84	±1.33	±1.97						
12	0.00	±0.01	±0.41	±0.84	±1.33	±1.97					
13	±0.01	±0.02	±0.42	±0.85	±1.34	±1.98					
14	0.00	±0.01	±0.02	±0.42	±0.85	±1.34	±1.98				
15	±0.01	±0.02	±0.03	±0.43	±0.85	±1.34	±1.98				
16	0.00	±0.01	±0.02	±0.03	±0.43	±0.85	±1.34	±1.98			
17	±0.01	±0.02	±0.03	±0.04	±0.44	±0.86	±1.35	±1.99			
18	0.00	±0.01	±0.02	±0.03	±0.04	±0.44	±0.86	±1.35	±1.99		
19	±0.01	±0.02	±0.03	±0.04	±0.05	±0.45	±0.87	±1.35	±1.99		
20	0.00	±0.01	±0.02	±0.03	±0.04	±0.05	±0.45	±0.87	±1.35	±1.99	
21	±0.01	±0.02	±0.03	±0.04	±0.05	±0.06	±0.46	±0.88	±1.36	±2.00	
22	0.00	±0.01	±0.02	±0.03	±0.04	±0.05	±0.06	±0.46	±0.88	±1.36	±2.00
23	±0.01	±0.02	±0.03	±0.04	±0.05	±0.06	±0.07	±0.46	±0.88	±1.36	±2.00
24	0.00	±0.01	±0.02	±0.03	±0.04	±0.05	±0.06	±0.07	±0.46	±0.88	±1.36	±2.00
25	±0.01	±0.02	±0.03	±0.04	±0.05	±0.06	±0.07	±0.08	±0.46	±0.88	±1.36	±2.00

Table 1: optimal grouping for a gaussian random variable

The values in table 1 are accurate to two significant digits. By using these cut points the partial means of the *m* variables are computed and inserted in a (kxm) matrix of typical values. If the first or the last interval were left empty then their mean is set equal, respectively, to the maximi and to the minimum of the variables.

Option 4: "natural classes"

Mineo (1985). Let $\{x_{j,r}, n_{j,r}, r=1,2,\ldots,m_j\}$ be the frequency distribution of the j-th variable where the values are sorted by size and m_j denotes the number of distinct values observed for X_j. Determine the minimum of

$$D_r = \left(\frac{n_{j,r} n_{j,r+1}}{n_{j,r} + n_{j,r+1}} \right) \left(x_{j,r+1} - x_{j,r} \right)^2, \quad r = 1,2,\ldots,m_j - 1 \tag{42}$$

The value of $x_{j,r}$ is replaced by

$$x^*_{r,j} = \frac{x_{j,r} n_{j,r} + x_{j,r+1} n_{j,r+1}}{n_{j,r} + n_{j,r+1}} \qquad with \quad n^*_{r,j} = n_{j,r} + n_{j,r+1} \tag{43}$$

The value $x_{j,r+1}$ is eliminated and the values above are shifted back to form a new frequency distribution with m_j *-1* values. These two steps are iterated until the frequency distribution has only *k* distinct values. The same procedure is repeated for the *m* variables to define the matrix of *(kxm)* representative values. A serious drawback of all these methods is that, as dimensionality increases, the volume of the data concentrates in the external boundary with the consequence that high dimensional space is mostly empty which, in turn, implies that these methods are doomed to find most of the partitions invalid because one or more clusters have no entities in it. To avoid invalid partitions, random entities are selected from the largest clusters and placed in the empty clusters.

3.2.3 Random combinations of entities.

Let $P=\{P_1, P_2, ..., P_k\}$ a random samples without replacement of *k* entities from the data set of *n* entities; then each entity is assigned to its closest centroid according to (22) (this ensures that each cluster contains at least one entity). The procedure is repeated until $Min\{v, C(n,k)\}$ partitions are examined. In particular, if $v > C(n,k)$ then all possible combinations of *n* entities taken *k* at a time are considered as initial centroids. However there is no guarantee that a "true" centroid coincides with one of the entities to cluster so that even a complete enumeration of all combinations may result in an inappropriate initial partition. This method presents a similar problem to that examined in section 3.2.1. In fact, when two or more of the selected entities are close together so that there will be two or more cluster close together which not necessarily are present in the data set. In addition, if clusters are of unequal size, the small cluster have lower chances to generated a centroid and tend do be absorbed by the larger ones.

To remedy this shortcoming ***DetClus*** considers $Min\{4k, n/2\}$ randomly selected and distinct entities and choice the best *k* centroids by applying the Kennard_Stone procedure (*option 5).*

3.2.4 Random partitions

A set of *n* integers is chosen as follows

$$\gamma_r^0 = j \quad if \quad P_{j-1} \le \varpi_r \le P_j, \quad j = 1,2,\ldots,k; \qquad r = 1,2,\ldots,n \tag{44}$$

where ω_r is a pseudorandom numbers from *[0,1]*. The quantities $P_0, P_1, ..., P_k$ are given by

$$P_0 = 0, \quad P_t = \sum_{i=0}^{t} v_{ri}; \qquad v_{ri} = \frac{v_{ri}^*}{\sum_{j=1}^{k} v_{rj}^*}, \quad i = 1,2,\ldots,k \tag{45}$$

where v_{ri}^* $i=1,2,...k$ are random numbers from $[0,1]$. The previous expressions ensure that each cluster always contains at least one entity.

3.2.5 Random shuffling

Let $\gamma=(\gamma_1, \gamma_2, ..., \gamma_n)$ be a vector of integers between 1 and k. By using the technique suggested by Knuth (1981, p. 139) random permutations of the γ_r's are considered. The set of numbers to be shuffled is chosen as follows

$$\gamma_r^0 = j \quad for \;\; r = N_{j-1}, N_{j-1}+1, \ldots, N_j - 1; \quad where \;\; N_j = \sum_{i=0}^{j} n_j; \;\; n_0 = 1, \;\; j = 1,2,\ldots,k$$

The user must specify the cardinalities of the clusters. This option allows the algorithm to explore the partitions with the same number of members per cluster.

3.3 Applications of the Indifference Principle

Since we ignore the real cluster membership of the entities, each entity should have the same chances of joining one of the k cluster. An initial configuration based on this policy is free of overt biases. Le $b=[n/k]$ and $s=n-b*k$;

Option_1: equal membership partition

Each cluster has the same number of entities except the last group which is assigned all extra entities. To obtain such a partition the first b entities are assigned to cluster C_1; entities labelled from $b+1$ to $2b$ to cluster C_2 and so on. The last s entities are added to the last cluster.

Option_2: discrete uniform distribution

For each entity r a random number j is generated from the discrete uniform $[1,k]$ distribution and $\gamma_r^{(0)}=j$ for $r=1,2,\ldots,n$. The empty clusters receive an entity from a regular cluster

Option_3: random blocks

Step_1. Set $\gamma_r=0$ for $r=1,2,\ldots,n$. Set $h=1$.

Step_2. Generate b distinct random integers u_i, $i=1,2,\ldots,b$ in the interval $[1,n]$.

Step_3. Set $r=u_i$. If $\gamma_r=0$ then assign entity X_r to cluster C_h . Set $\gamma_r=1$.

Step_4. If $h<k-1$ then set $h=h+1$ and go to Step_2.

Step_5. If $\gamma_r=0$ then assign entity X_r to cluster C_k for $r=1,2,\ldots,n$.

Option_4: nested loops

The entities labelled $\{j, k+j, 2k+j, \ldots, (b-1)k+j\}$ are assigned to the cluster $C_j=j=1,2,\ldots,k$. The last s entities are added, one for each, to the first s clusters.

3.4 Read centroids from file

The user can provide the estimated centroids from a text file in which the rows are the centroids and the columns are the variables. This options is allowed for a fixed number of clusters. The program checks the internal conditions: $L_j \leq \mu_{ij} \leq U_j$, $j=1,2,\ldots,m$; $i=1,2,\ldots,k$. If this condition is not satisfied then each invalid entry is replaced by a uniform random number in the interval $[L_j,U_j]$. The corresponding classification vector is obtained by applying (22). The partition is discarded if some cluster is empty.

3.5 Read partition from file

Sometimes the entities to be clustered have *a-priori* labels and one is investigating wether the cluster membership that can be obtained by the algorithm is consistent with the known labels (supposing these to be a plausible classification of the data set). Moreover, this option allows the user to start **DetClus** from the configuration achieved by another procedure (internal or external to the program).

The program accepts the proposed partition only if each clusters has at least one empty. The number of clusters is derived from the number of distinct labels found in the file.

$$\begin{bmatrix} \gamma_1 & \gamma_2 & \cdots & \gamma_i & \cdots & \gamma_n \end{bmatrix}$$

4.The quality of a partition

Any clustering algorithm constructs a partition $\gamma^* \in P(n,k)$ which is optimal in terms of the stated criterion and the initial solution, with as many clusters as desired (virtually every definition of optimal clustering does not depend on the number of clusters). However, the clustering found will be useful only if the classes can be substantively interpreted. Fisher and Van Ness (1971) observed that the main objective of a clustering is to condense information by reducing the individual description of all X's to a relatively few general description of *k* typical representatives, one for each cluster. Paradoxically, if the variables were constant within the clusters, one entity per cluster would suffice to express any detail of the data set. As a rule, the lower *k* is the stronger is the partition since less information is needed to summarize the data; hence, when there is more than one optimal solution, the one with the lower number of clusters should be chosen. Castagnoli (1977) has shown that such a partition always exists. The problem is further compounded by the fact that, as we have seen in the previous section, the number and the type of clusters in the data may depend on the resolution with which we look at the data.

One major problem shared by all methods of cluster analysis is that an optimal partition of the data set into a certain number of nonempty subsets with pairwise empty intersections will be developed whether or not a natural clustering exists and whether or not it is possible to select plausible centroids among the data set.

Example:

In dissection, the data set comprises entities whose distribution into the space of variables is uniform; the aim is to subdivide the entities into sectors (e.g. Policy precincts, voting districts, school districts and so forth). Nevertheless, it is legitimate to wonder whether entities in different sectors of Figure 11 are heterogeneous and whether the clusters obtained have a real existence.

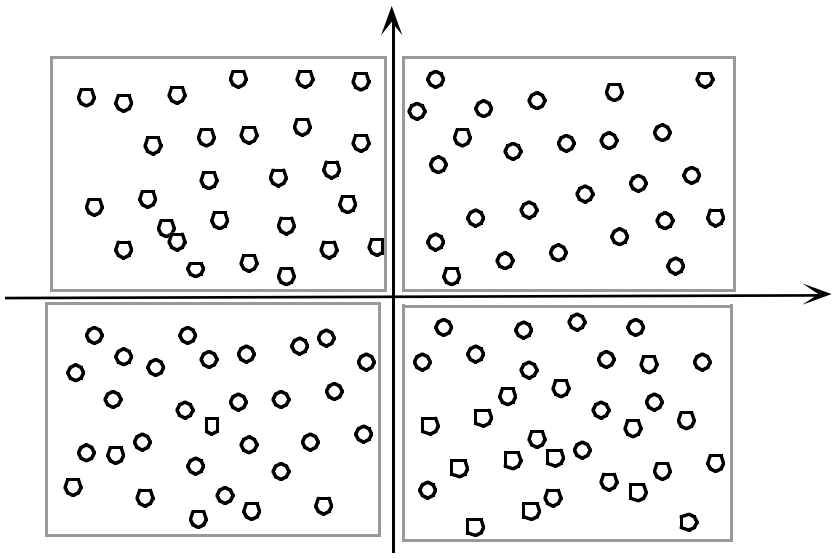


Figure_11: artificial clustering

The quality of a clustering is partly intrinsic to the data-generating process, the data collection equipment, the choice of the variables, and a possible selective identification of the entities to be clustered. These issues are, however, outside the scope of the present section; here the intent is to devise extrinsic aids (graphical and numerical) for distinguishing meaningful partitions from those artificially imposed on the entities.

Example:

This experiment was constructed by simulating points from 3-dimensional random variables having uniform marginal distributions. Let $\boldsymbol{u}_i$ be a vector of *m* independent uniform random variables on *(0-1)* with $E(\boldsymbol{u}_i)=0.5\,\boldsymbol{c}_m$ and $E(\boldsymbol{u}_i\boldsymbol{u}_i^t)=(12)^{-1}\boldsymbol{I}$ where $\boldsymbol{I}$ is the identity matrix of order *m*; then $\boldsymbol{Y}_i=\sqrt{12}(\boldsymbol{u}_i-0.5\boldsymbol{c}_m)$ is a vector of independent uniform random variables with $E(\boldsymbol{Y}_i)=\boldsymbol{0}$ and $E(\boldsymbol{Y}_i\boldsymbol{Y}_i^t)=\boldsymbol{I}$. Consider now the affine transformation $\boldsymbol{X}_i=\boldsymbol{H}\boldsymbol{Y}_i+\boldsymbol{d}_i$ where $\boldsymbol{H}\boldsymbol{H}^t$ is the Cholesky factorization of $\sum$; then $\boldsymbol{X}_i$ *is a a* m-dimensional random variables having uniform marginal distributions with $E(\boldsymbol{X}_i)=\boldsymbol{d}_i$ and variance-covariance matrix $E(\boldsymbol{X}_i\boldsymbol{X}_i^t)=\boldsymbol{H}\boldsymbol{H}^t=\sum$ Usually, only synthetic data sets include “natural” clusters exhibiting high level of external isolation and internal cohesion. However, if one encounters such data (and there is no reason to suspect an happenstance, an error or a joke), it would not be hard to find a convincing *post hoc* rational explanation which legitimates the empirical results.

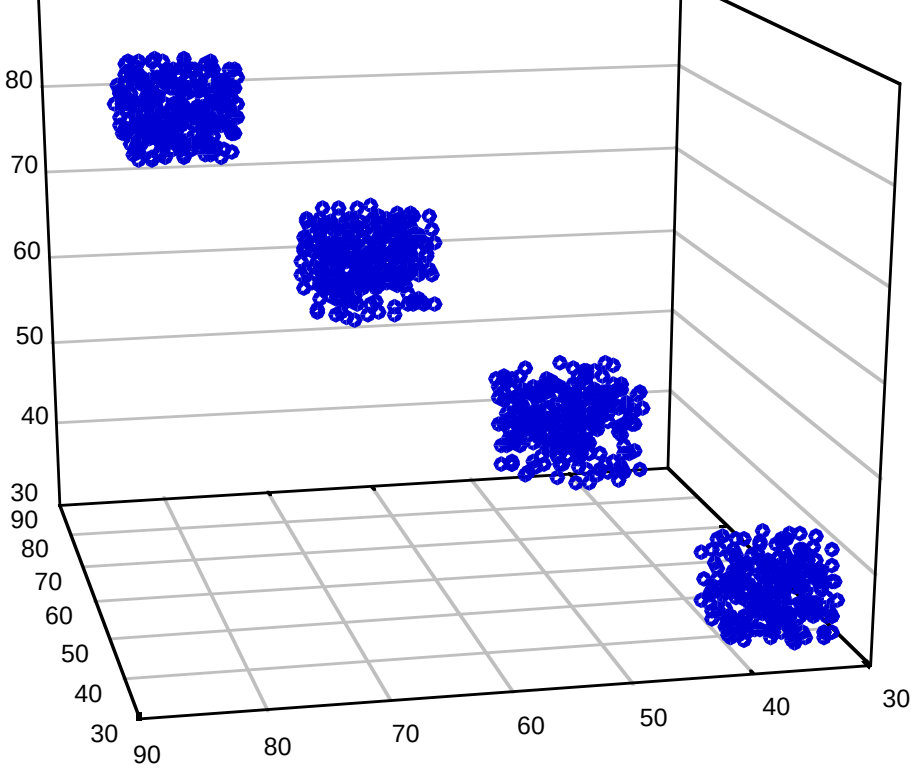


Figure 12: ideal data set

Though the partitioning of natural clusters appears meaningful and potentially useful, the partitioning of a unimodal or uniformly distributed entities does not appear to have the same basis (Arnold, 1979).

Example

Späth data set (Späth, 1985, p.144).
Two variables for 41 entities randomly scattered over the variable space without accumulation zones; none of the interpoint distances is significant; there is no natural grouping within the data so that any rule proposing a "plausible" partition into *k* groups should be critically exhamined. ***DetClus*** for *k=3*, has produced an arbitrary dissection (figure13) along the axis of maximal dispersion. Marriott (1971) noted that minimization of *Min{|W(γ)| }"... searches for any natural grouping, not necessarily one based on all the measurement"*.

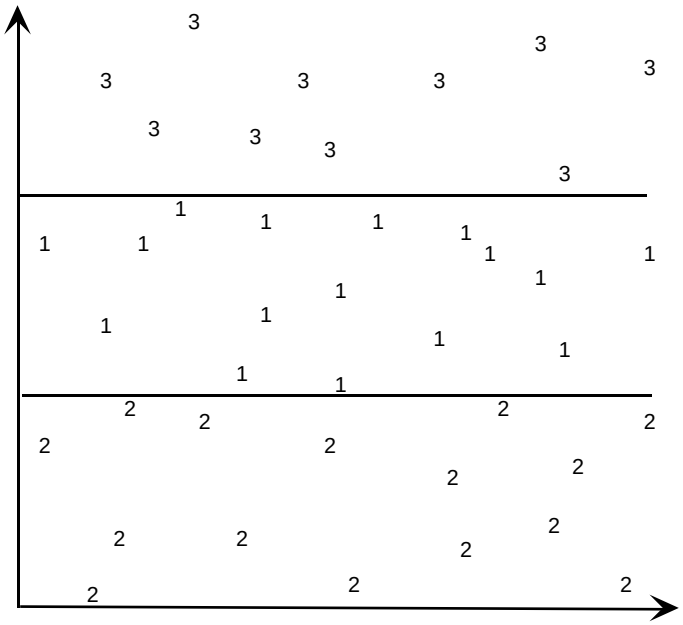


Figure 13: results for the Späth data set

Example

Unimodal data set. A sample of 250 bidimensional entities uniformly distributed within the ellispoidal region $x'\Omega^{-1}x \leq 1$ where

$$\Omega = \begin{bmatrix} 9 & 3 \\ 3 & 9 \end{bmatrix}$$

DetClus has no protection against finding groups in the data when in effect none is present. For *k=3*, it finds a seemingly plausible partition which is actually unexplicable in terms of what is known on the data. The fact that the clustering algorithm has found a structure doed not imply that ther sdtructure is real.

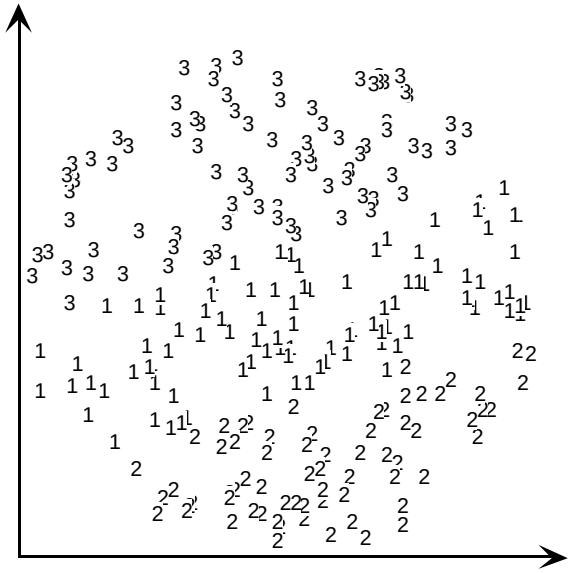


Figure 14: results for the unimodal data set

In both examples it appears difficult to argue that one particular solution has more meaning or stability in either a logicakl or theoreticalk sense that any other clustering that can be randomly generated.

In general, each clustering is a good clustering if there is theoretical and circumstantial evidence that may convincingly explain the structure obtained; conversely, any clustering, optimal though it may be, lacking an explanation as to how the member of a group came to be described as similar, and how these members differ from those of other groups, is merely an artifact of the algorithm.

Any expert or practitioner of cluster analysis knows that the output of a clustering procedure is not the end of the story, but several questions must be answered. Bock (1995) suggests the following

1) What is the relevance and significance of the resulting classes?
2) Do they reflect a "true" or "natural" grouping structure of the data or just an artifact of the method selected?
3) How does the clustering perform when compared to random classifications?
4) Which are the strongest or the most doubtful classes?

In general, procedures used to evaluate clusters determined by a clustering method are of two types. The first one includes procedures for testing the resultant clusters against the null hypothesis that the clusters were randomly determined. Procedures of the second type are based on the assumption that the clustering method in use has attained an optimal partition which is compared with a given partition for comparison purposes.

4.1 External indices of validation

The effectiveness of relocation procedures can be measured by comparing the final partition γ^+ generated by the algorithm with the prior knowledge of the true classification δ. Sometimes the iterative scheme is starting from δ which should be, hopefully, in the domain of attraction of a global minimum. This situation is very unrealistic in that it tacitly assumes that not only the number of clusters, but also the true cluster membership of all *n* entities is known. However, such idealized setting offers a simple benchmark against which the results can easily be compared. In particular ***DetClus*** computes the Hubert-Arabie (1985) statistic

$$R_{HA} = \frac{nc(n-1)(c-1)-ab}{n(n-1)b-ab};$$

$$a = \sum_{i=1}^{k} n_i^+\left(n_i^+ - 1\right);\ b = \sum_{i=1}^{k} n_i\left(n_i - 1\right);\ c = \sum_{r=1}^{n-1}\sum_{s=r+1}^{n} \varepsilon\left(\gamma_r^+ = \gamma_s^+ \cap \delta_r = \delta_s\right) \tag{46}$$

where $\varepsilon(x)$ is one if *x* is true and zero otherwise. The statistic R_{HA} has a fixed upper bound $R_{HA} = 1$ indicating perfect clustering recovery and takes the value zero under the hypothesis that γ^+ and δ are picked at random subject to having the true number of clusters and objects in each. In

addition, ***DetClus*** computes a naive index of clustering efficacy

$$Q = \frac{\pi_C - \pi_L}{1 - \pi_L}; \qquad \pi_C = \sum_{i=1}^{k} \binom{n_i(\gamma^+)}{2} \Big/ \binom{n}{2} \quad ; \quad \pi_L = \sum_{i=1}^{k} \binom{n_L(\delta)}{2} \Big/ \binom{n}{2} \tag{47}$$

where percentage π_C is the proportion of pairs in which the two entities are in the same clusters both in γ^+ and δ while π_L is the percentage of pairs of entities belonging to the largest cluster of δ. In pracgtice, the statistic$Q$ compares the goodness of the classification resulting from a k-means algorithm and the naive classification obtained putting all the entities in one cluster. A negative value of Q indicates that ***DetClus*** was not able to detect any clustering in the data set, at least for the given starting partition. The user must be aware that (46) and (47), as well as, many other external indices of agreement, are not a naturally increasing function of the quality of the partition found by the procedure.

Example:

Ruspini data set. (Kaufman and Rousseeuw, 1990, p. 100). This is a standard example consisting of 75 two-dimensional points making up *k=4* natural groups including 23, 20, 17, 15 entities . Actually the these data are different form the original data used by Ruspini (Rasson e Kukbushi-shi,1994, p. 191).

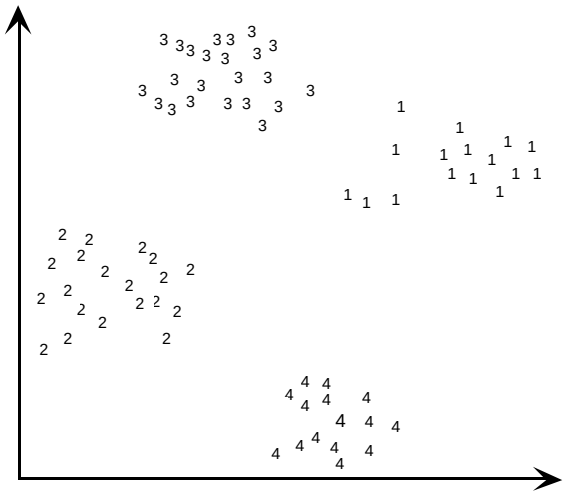


Figure 15: results for the Ruspini data

In this example the groups are well-structured and any reasonable method of cluster analysis can isolate them. ***DetClus*** does not fail to retriew this obvious structure: R_{HA}*=1* and *Q=1*.

Example:

Storm survival of sparrows (Bumpus data set). After a severe storm on 1 February 1898, a total of 136 sparrows *(Passer domesticus)* were taken to Bumpus's laoratory. Bumpus took m=9 morphological measurement on each bird and also weighted them. Manly (1985) reproduced his data classified according to sex and the age of males for a total of six clusters having the cardinalities: young males that survided=16; young males that died=12; adult males tha survided=35; adult males that died=24; adult and young females thad survided=21; adult and young females that died=28.

The correlation matrix is positive (each elements is greater than zero) so that the first principal component is an index of size (factor loadings having the same sign and roughly equal magnitude) whereas the other components are contrast or shape components (at least one factor loading has a sign different form the others).

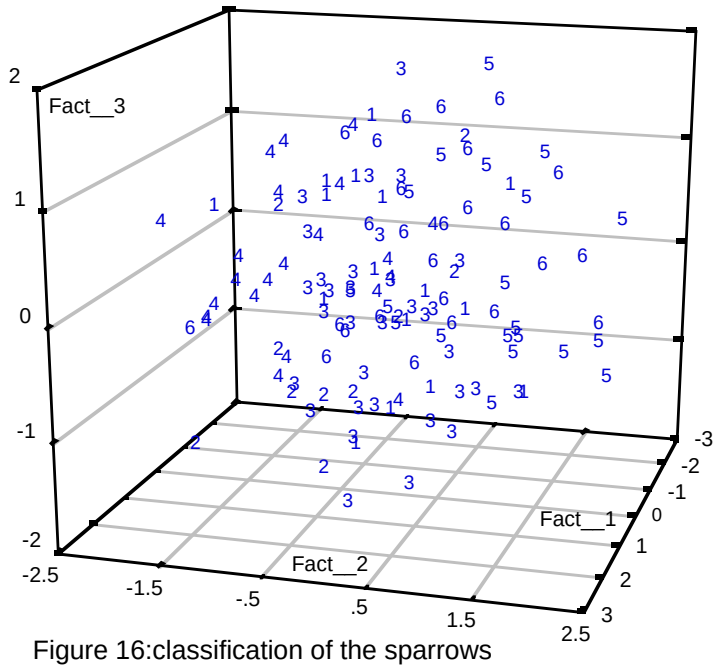


Figure 16:classification of the sparrows

The space of the first three PC's, which explains the 76.6% of total variation , does not show any particular structure. For *k=6* **DetClus** found R_{HA}*=0.075* and *Q=-0.012* The quality of the results does not improve when the final partition of **DetClus** is compared with the subdvision of the data set according to the sex or according to survivors/non survivors sparrows.

4.2 Estimation of the number of clusters

There is no standard way of statistically evaluating the adequacy of the obtained sequence of partitions. The vagueness of the theoretical basis makes it difficult to achieve analytical results in this area and preference should be given to graphic displays. These techniques are very useful in the validation of a clustering even though it has proven unreliable to trust intuition or visual perception alone. Blanshfield *et al* (1982) have observed that iterative partitioning algorithms are much better than hierarchical algorithms concerning output descriptive statistics to making it possible to obtain many graphic views for more intimately inspecting the clustering process.

4.2.1 Complete clustering characteristic graph

One of the most popular methods of choosing the appropriate number of clusters is to plot the objective function against the number of clusters *k* for a range of values of *k*. The true number of clusters is found by considering those values of k fro which the plot shows a sharp in/decrease of the criterion. ***DetClus*** considers two indicators

$$Friedman-Rubin: \quad C = \frac{Min\{|\boldsymbol{W}(i)|\}}{|\boldsymbol{T}|} * 100; \qquad i = k_1,\ldots,k_2 \tag{48}$$

The criterion is normalized by the corresponding value for *i=1* so that (48) lies in the interval *(0,100)*. Expression C is a decreasing function of the number of clusters and an increasing function of the number of entities and dimensions. Undoubtedly, with every increase in *i* there will be a decrease in (48), but the change should be irrelevant for *i>k* when *k* is the number of cluster which best fits the data. In practice, a discontinuity in slope should correspond to the true number of clusters, otherwise there no justification for having more than one class (Hardy,1996).

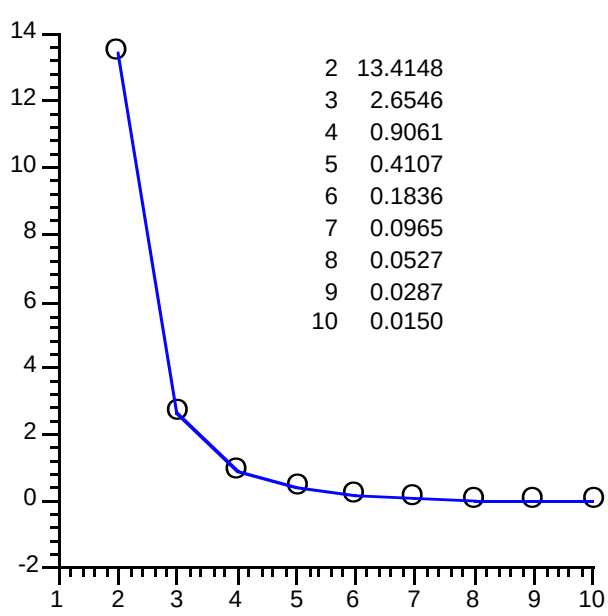
Arnold (1979) proposed the following test statistic

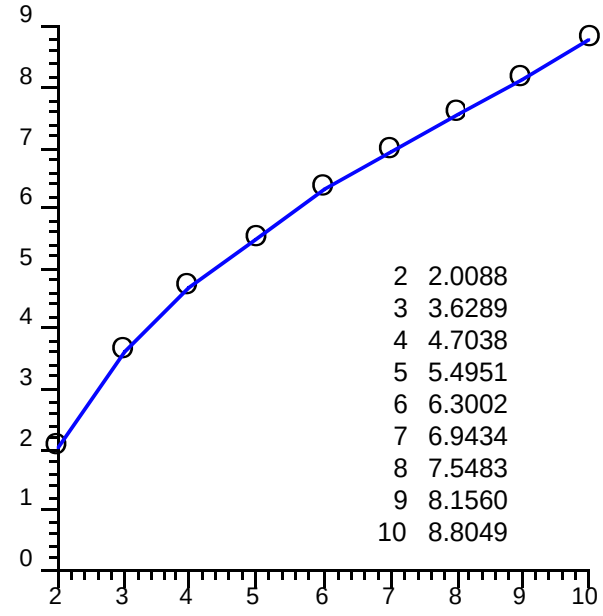
$$\alpha = Ln\left\{\frac{|\boldsymbol{T}|}{|\boldsymbol{W}(\boldsymbol{i})|}\right\}; \qquad i = k_1,\ldots,k_2 \tag{49}$$

for testing the null hypothesis that the entities are either uniformly distributed or grouped into clusters. The method of deriving the distribution of α was based on Monte Carlo techniques, but the results are not satisfactory. However, the plot of (49) can be used as (48) to correctly estimate the number of clusters. ***DetClus*** writes (48) in the output file, but the user can easily compute (49) by $\alpha=Ln(100/C)$. The user must be aware that highly collinear variables can create problem to the Anderson statistic.

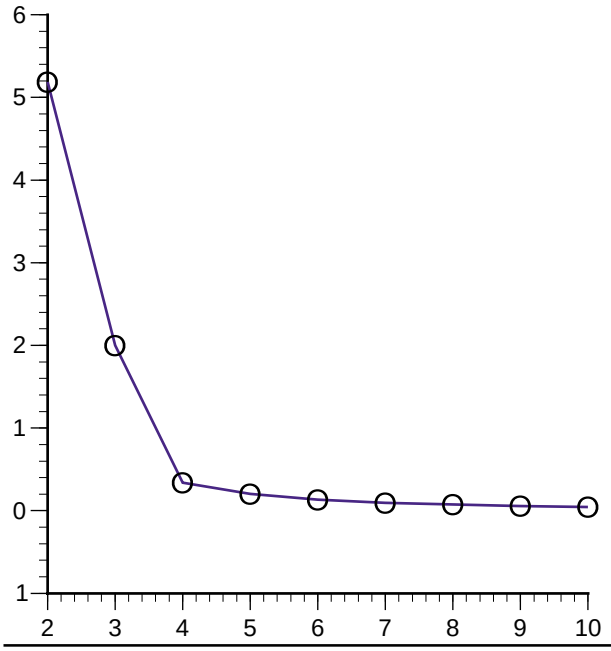
Example

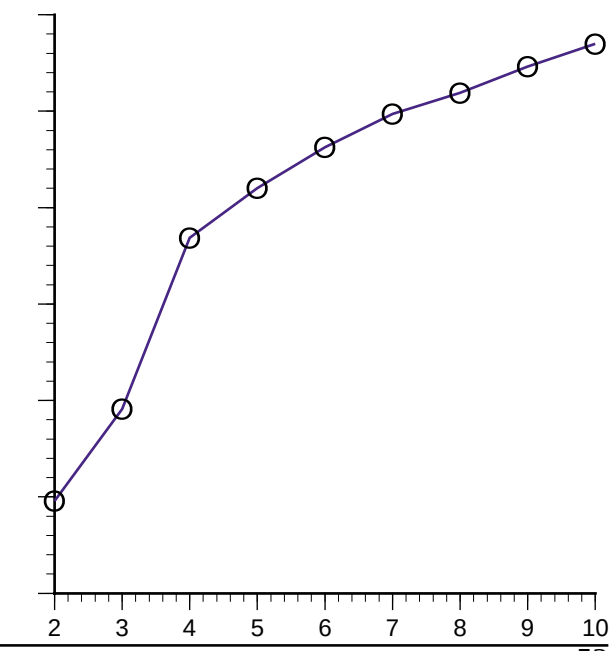
The statistics of poverty and inequality (Rouncefield, 1995). For $n=97$ countries in the world, data are given for birth rates, death rates, infant death rates, life expectancies for males and females, and GNP. For this example the first four principal components were used (98.9% of total variation explained). The clustering of the data set appear to be weak and does not correspond to the classification in $k=6$ clusters proposed by the geographical grouping included in the data. The value $k=4$ is a plausible choice because of the sharp decrease noted in (48) and the progressively reduced increments in (49) after $i=3$, but other choices can easily be made.





For the Ruspini data sets, both the indices perform well.



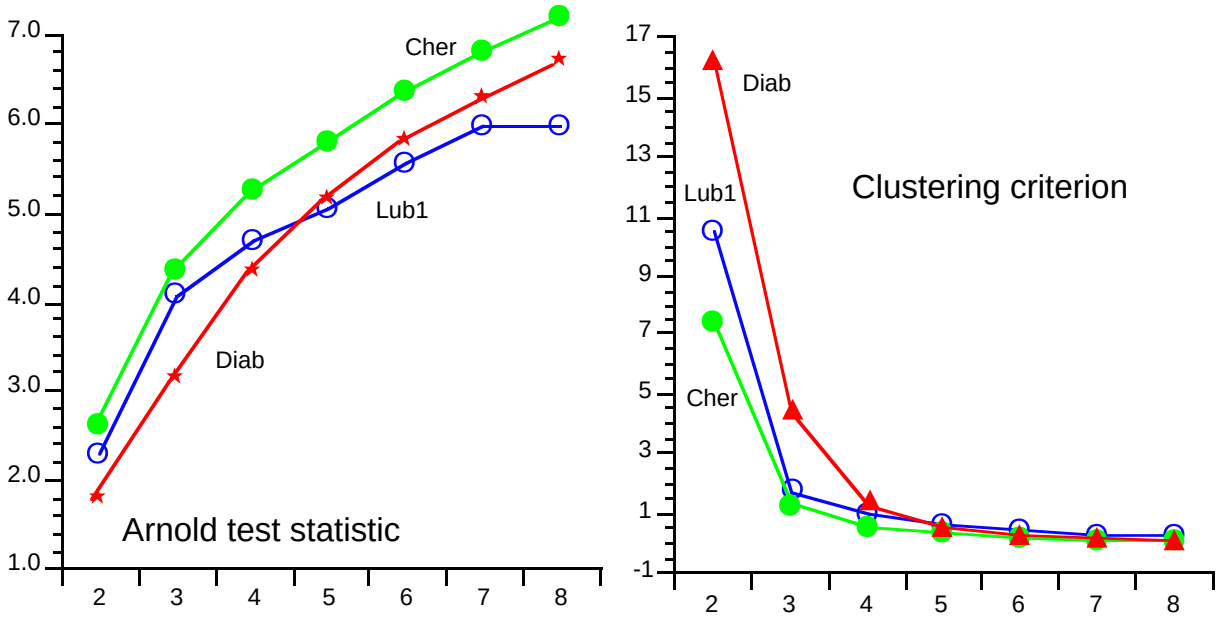


Examples:

1) Lubishew data set 1 (Lubishew, 1962). Measurements were made of six variables in the males of three species *Chaetocnema concinna, Ch. heikertingeri, and Ch. heptapotamica,* The real composition of the groups is (21, 31, 22). ***DetClus*** correctly assigned to the appropriate cluster all the entities even though only the first three principal components (89.3% of total variation) were used to identify the specimen.

2) Fossils data (Chernoff, 1973). Six variables were measured on each of nummulited specimens from Eocene Yellow Limestone formation of Northwestern Jamaica. According to Chernoff the entities divide into three distinct clusters: {40, 34, 13} with one or two specimen which can be regarded as singleton or borderline. ***DetClus*** has been applied to the first four principal components (accounting for 94.6% of the variability contained in the data set) providing perfect recovery of all the entities. However, the largest cluster can be separated into subclusters, but their number is undeterminate.

3) Chemical and overt Diabetes (Andrews and Herzberg,1985). This data set consists of five variables (insulin area, glucose area, and steady-state plasma glucose response) measured on *n=145* non obese adult subjects. The subjects were clinically classified as normal (76), Chemical diabetes (36) and overt diabetes (33). The clusters have various sizes and different non-ellipsoidal dispersion matrices.



For the Lubishew1 data set *k=3* is an evident point of inflection. The Chernoff data set shows a drop (or a jump if you are looking to the Arnold statistic) at *k=3* and at *k=4* but it is non easy to make a decision without further analysis. The graph for the diabetes data set is confused. However, a partition in *k=3* or *k=4* cluster is deemed to be plausible.

DetClus provides a very raw graph of (48), but the value of the criterion can be copied from the output file and pasted in one's favorite plotting program (Excel, Statistic, Deltagraph, etc.£

Hall and Khanna (1977) have great confidence on this type of graph: a knee (i.e. a sharp step from *i* to *(i+1)* followed by a marked flattening of the curve suggests that *k=i+1* is a good choice. Other authors (*e.g.* Everitt, 1979; Gordon, 1999, p. 61) do not recommend great reliance on this graph. Three good reasons for such reservations are:

a) A data set often exhibits more than one point of diminishing return (that is, the value of *k* at which the rate of decrease in the slope starts to diminish) and it is difficult to tell which indicates the correct number of clusters.

b) Frequently the plot has a knee even if the conjoint cluster solution might be considered the best partition.

c) It may be difficult to locate the critical point in the graph for large values of *k* where the variations are small anyway.

The above-mentioned problems are frequent when the structure of the data set is very complicated. Unfortunately, these are just the occasions when an effective means for comparing alternative clusterings becomes more acutely necessary.

The plots by themselves do not rigorously reveal how many clusters are actually present. Rather, they are useful guidelines in selecting an appropriate number of clusters in a context where developing inferential methods has proved difficult. The defects of a subjective estimation of *k* result, in the main, from uncertainty which is demonstrated when different observers have to decide on the same plot and obtain different answers. In fact, Milligan and Cooper (1985) excluded from their review any technique requiring human judgement, but took into consideration analogous procedures, based on "difference scores", that are not so different from visual assessments.

Example:

Egyptian skulls data set (Hand *et al*. 1994). Four measurements of male Egyptian skulls from five different time periods. Thirty skulls are measured from each time period (*n=150*). The elimination of the point corresponding to *k=2* evidentiates the knee at *k=5* (the true value of the number of clusters). However, the recovery rate is extremely poor: R_{HA}*=0.0041* and *Q=-0.09.*