PERGAMON

Pattern Recognition 36 (2003) 2955–2966

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

A computational study of several relocation methods for k-means algorithms

Agostino Tarsitano*

Dipartimento di Economia e Statistica, Università della Calabria, 87030, Arcavacata di Rende (Cs), Italy

Received 20 December 2002; accepted 22 April 2003

Abstract

The core of a k-means algorithm is the reallocation phase. A variety of schemes have been suggested for moving entities from one cluster to another and each of them may give a different clustering even though the data set is the same. The present paper describes shortcomings and relative merits of 17 relocation methods in connection with randomly generated data sets.

Keywords: Non-hierarchical classification; Iterative partitioning; Combinatorial optimization

1. Introduction

The goal of cluster analysis is to verify the presence (or the absence) of natural clusters in a given set of entities. The data set D consists of n distinct m-dimensional points $D = \{\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_n\}^m$ where, for each r, $\boldsymbol{X}_r$ gives the observed values of m real-valued characteristics. Relative geometric arrangements, causing concentration and dispersion of the points in different regions, produce clusters.

There is a wide choice of clustering methods which have different adaptability to the data and different requirements of computer resources. The k-means algorithm usually start with an arbitrary choice of a feasible classification of the entities into clusters. Then, keeping the same number of clusters, a sequence of possible reassignments is considered. The reassignment that yields the maximum benefit is made and the process is repeated until an optimum of the criterion is reached. The core of a k-means algorithm is the reallocation phase. A variety of schemes have been suggested for moving entities from one cluster to another [1] and each of them may give a different clustering of the data set. However, little consensus exists with respect to which scheme is most indicated as being effective (i.e. maximizes the number of proper assignments) and efficient (i.e. evaluates the minimum number of membership changes). The current research study will focus on k-means algorithms based on the Friedman–Rubin criterion. In particular, questions regarding both the computational efficiency and clustering efficacy of several reassignment methods are addressed.

The contents of the various sections are as follows. Section 2 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. Section 3 is devoted to relocation schemes which can be used by a k-means algorithm to determine the best cluster assignment. Some of the methods have never been previously tested—as far as the author is aware—and in some cases some surprising results have been noted. Section 4 will focus on the reallocation of pairs of entities. Finally, Section 5 describes shortcomings and relative merits of 17 relocation methods in connection with randomly generated data sets.

2. K-means algorithms

A partition is a collection of k subsets such that

$$C_j \neq \emptyset, \quad 1 \leqslant k \leqslant n, \quad \bigcup_{j=1}^{k} C_j = D,$$
$$C_i \cap C_j = \emptyset \quad (i \neq j), \tag{1}$$

* Tel.: +39-0984-49246; fax: +39-0984-492468.
E-mail address: agotar@unical.it (A. Tarsitano).

0031-3203/03/$30.00
doi:10.1016/S0031-3203(03)00190-0

where $\emptyset$ is an empty set and the cardinalities $n_1, n_2, \ldots, n_k$ of the clusters satisfy

$$\text{(a) } n_i \geqslant 1, \quad \text{(b) } \sum_{i=1}^{k} n_i = n. \tag{2}$$

This implies that each entity is assigned to exactly 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 $\gamma = (\gamma_1, \gamma_2, \ldots, \gamma_n)$ where γ_r denotes the cluster membership assigned to $\boldsymbol{X}_r$ that is $\gamma_r = j$ if $\boldsymbol{X}_r \in C_j$. The requirement of exclusive assignment 2b is particularly stringent since every entity is constrained to join a cluster, including outliers (entities that consistently emerge as singletons or very small clusters) and intermediate entities linking two or more otherwise isolated clusters. 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 problem addressed by k-means algorithms is to find, for each cluster C_j, $j = 1, 2, \ldots, k$ a representative entity or centroid $\boldsymbol{\mu}_j$ for whom is minimized a known function of the dissimilarities between an entity in the cluster and the centroid. When $\boldsymbol{\mu}_j$, $j = 1, 2, \ldots, k$ are defined the classification vector γ is determined by assigning each entity to the cluster with the nearest centroid. The ability of $\boldsymbol{\mu}_j$ to summarize the information content of C_j 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. Since each partition may provide a reasonable solution to the clustering problem, some selection is necessary. A comparison can be accomplished using an objective function $L(\gamma) : \gamma \in P(n,k) \rightarrow [0, \infty)$ which measures the quality of different partitions of the same data set. More specifically, $L(\gamma) < L(\boldsymbol{\delta})$ means that γ provides better estimates than $\boldsymbol{\delta}$. The symbol $P(n,k)$ denotes the set of partitions of n entities into k clusters. This paper uses the criterion proposed by Friedman and Rubin [2] $L(\gamma_q) = \text{Min}\{|\boldsymbol{W}(\gamma_q)|\}$ where

$$\boldsymbol{W}(\gamma_q) = \sum_{j=1}^{k} \boldsymbol{W}_j^q, \qquad \boldsymbol{W}_j^q = \sum_{r=1}^{n} (\boldsymbol{X}_r - \boldsymbol{\mu}_{\gamma_r}^q)(\boldsymbol{X}_r - \boldsymbol{\mu}_{\gamma_r}^q)^t,$$

$$\boldsymbol{\mu}_j^q = \sum_{\gamma_r = j} \boldsymbol{X}_r / n_j^q, \qquad j = 1, 2, \ldots, k. \tag{3}$$

$\boldsymbol{W}(\gamma_q)$ is the pooled dispersion matrix across the k clusters (or within-cluster dispersion matrix) for the qth classification vector. It is assumed that $n \geqslant m + 1$ otherwise the estimate is always singular regardless the true value of $\boldsymbol{W}(\gamma)$. The use of (3) implies:

(1) the dissimilarities are measured by Mahalanobis distances;

(2) each centroid coincides with the average of all entities within the cluster;

(3) the clusters have the same variance–covariance matrix.

The criterion $\text{Min}\{|\boldsymbol{W}(\gamma)|\}$ is invariant under the affine transformations $\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X} + \boldsymbol{b}$ where $\boldsymbol{A}$ is non-singular (this allows the question of standardization of the variables to be overcome); additionally, it is appropriate when the variables are correlated because it takes into account the variability of the values in all dimensions. However, since $\boldsymbol{W}(\gamma)$ is an average of the variance–covariance matrices of the clusters, correlated variables in the clusters generate multicollinearity in $\boldsymbol{W}(\gamma)$, that is $|\boldsymbol{W}(\gamma)|$ will approach to zero as correlations grow stronger.

If $\boldsymbol{W}(\gamma)$ is ill-conditioned and one supposes that the clusters lie in the same subspace, redundant features can be eliminated by applying techniques such as principal component analysis. This has two positive implications. Firstly, for the computational effort because reduced data require less storage space and can be manipulate more quickly than the complete data. Secondly, a limited set of selected features may alleviate the influence of irrelevant information.

Since the cardinality of $P(n,k)$ is finite, it exists at least one partition γ^* such that $|\boldsymbol{W}(\gamma^*)| < |\boldsymbol{W}(\gamma)|$ for $\gamma \in P(n,k)$. The most straightforward way to find γ^* is to evaluate $|\boldsymbol{W}(\gamma^*)|$, for all possible partitions generating, randomly or sequentially, each partition at most once. It is well known, though, that a complete search is possible in principle, but not in practice since the average number of partitions to be considered grows rapidly and becomes prohibitively high for even moderate values of n and k. It is necessary instead to apply techniques of local optimization.

Given an initial partition γ^q with $q=0$, k-means algorithms compute the criterion value $|\boldsymbol{W}(\gamma^q)|$. Another partition γ^{q+1} is obtained by transferring a single entity (or block of them) from one cluster to another. The new partition is accepted if $|\boldsymbol{W}(\gamma^{q+1})| < |\boldsymbol{W}(\gamma^q)|$ and the procedure is repeated until no further reduction of $|\boldsymbol{W}(\gamma^q)|$ can be obtained. Of course, there is no absolute guarantee in terms of solution quality and running time.

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. In general, the repetition of the algorithm starting from independent partitions and saving the best results is considered a good protection against this sensitivity. It should be emphasized that $|\boldsymbol{W}(\gamma^0)| < |\boldsymbol{W}(\boldsymbol{\delta}^0)|$ does not necessarily imply that $|\boldsymbol{W}(\gamma^*)| < |\boldsymbol{W}(\boldsymbol{\delta}^*)|$; therefore, the k-means algorithms must consider each initialization as a separate basis for the subsequent phases.

Less obvious, but often crucial to performance, is the sequence of the entities within the data set. A k-means algorithm is said to be combinatorial [3] if the criterion, centroids, cardinalities, and the within-group scatter matrix 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 non-combinatorial k-means algorithm [4], 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, the calculations are substantially simplified (but not necessarily reduced) and the iterative process does not suffer from ordering effects. However, unless certain conditions are satisfied [5], there is no guarantee of a net improvement in $L(\gamma)$, no guarantee that the k-means process converges, and no guarantee that k non-empty clusters are produced. In spite of this 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 [6, p. 166], and will not adopted in the present study.

3. Relocation of the entities

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 local minimum and the accuracy of this minimum. The schemes considered here 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.

3.1. Transferring entities

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).

$$\Delta_q(r,j,i) = \frac{|\mathbf{W}_{q+1}|}{|\mathbf{W}_q|} = (1+\alpha_i \mathbf{y}_i^t \mathbf{W}_q^{-1}\mathbf{y}_i)(1-\alpha_j \mathbf{y}_j^t \mathbf{W}_q^{-1}\mathbf{y}_j) + \alpha_i\alpha_j(\mathbf{y}_i^t \mathbf{W}_q^{-1}\mathbf{y}_j)^2,$$

$$\alpha_i = n_i^q/(n_i^q+1), \quad \alpha_j = n_j^q/(n_j^q-1),$$
$$\boldsymbol{y}_i = \mathbf{X}_r - \boldsymbol{\mu}_i^q, \quad \boldsymbol{y}_j = \mathbf{X}_r - \boldsymbol{\mu}_j^q. \tag{4}$$

If $\Delta_q(r,j,i)=\rho < 1$ 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. A threshold lower than one (e.g. $\rho = 0.9999$) prevents cycling divergence due to numerical problems; additionally, it may help to regulate the running time of the algorithm. The transferring pass can be carried out in three different ways.

3.1.1. Transferring pass-First-improving (TFI)

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 $i=1,2,\ldots,k;\ i\neq j;\ r=1,2,\ldots,n$, where the sequence in which the cluster are tried can also be natural or randomly generated. If $\Delta_q(r,j,i)\leqslant\rho$ 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 change in the scatter matrix, its inverse, centroids and cardinalities is easily computed from the following relations

$$\mathbf{W}_{q+1} = \mathbf{W}_q - \alpha_j \mathbf{y}_j\mathbf{y}_j^t + \alpha_i \mathbf{y}_i\mathbf{y}_i^t, \quad 1-\alpha_j\mathbf{y}_j^t\mathbf{Z}_q^{-1}\mathbf{y}_j \neq 0,$$
$$1+\alpha_i y_i^t \mathbf{W}_q^{-1}\mathbf{y}_i \neq 0,$$
$$\mathbf{W}_{q+1}^{-1} = \mathbf{Z}_q^{-1} + \frac{\alpha_j(\mathbf{Z}_q^{-1}\mathbf{y}_j)(\mathbf{Z}_q^{-1}\mathbf{y}_j)^t}{1-\alpha_j\mathbf{y}_j^t\mathbf{Z}_q^{-1}\mathbf{y}_j},$$
$$\mathbf{Z}_q^{-1} = \mathbf{W}_q^{-1} - \frac{\alpha_i(\mathbf{W}_q^{-1}\mathbf{y}_i)(\mathbf{W}_q^{-1}\mathbf{y}_i)^t}{1+\alpha_i y_i^t\mathbf{W}_q^{-1}\mathbf{y}_i},$$
$$\mu_i^{q+1} = \frac{n_i^q\mu_i^q + \mathbf{X}_r}{n_i^q+1}, \quad \mu_j^q = \frac{n_j^q\mu_j^q - \mathbf{X}_r}{n_j^q-1}, \quad n_i^{q+1} = n_i^q+1,$$
$$n_j^{q+1} = n_j^q - 1. \tag{5}$$

To avoid the accumulation of rounding errors, the quantities in (5) should be computed directly from the data after a number of transfers depending on the data set (e.g. $200(\mathrm{nm})^{0.5}$). The n entities are then checked in turn to see if another transfer decreases the criterion. For each entity, TFI examines at most $(k-1)$ partitions derived from the current partition by moving an entity from one cluster to another (that is, the neighborhood set).

3.1.2. Transferring pass-Local best-improving (TLBI)

A first-improving policy may lead to premature convergence of the k-means process. This motivated the development of several search methods to solve the problem of $\mathrm{Min}\{|\boldsymbol{W}(\gamma)|\}$. Rubin [7] suggested examining the potential effect of switching $\boldsymbol{X}_r$ from the cluster it occupies to each other cluster and finding the value satisfying $\mathrm{Min}\,\Delta(r,j,i)|\Delta(r,j,i)\leqslant\rho,\ i=1,2,\ldots,k;\ j\neq i\}$. If such a 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 may be 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 TLBI is more computer demanding that TFI since the latter is interrupted if also the former is interrupted, but this may continue to evaluate transfers also when the TFI does not.

3.1.3. Transferring pass-Global best-improving (TGBI)

Local improvements may impede or postpone more effective transfers; therefore, it seems plausible to devise a scheme which moves down the steepest permissible direction globally. TGBI performs a complete scanning of the entities and produces the set of candidate transfers $E = \{\Delta(r_h, j_h, i_h) \leqslant \rho,\ h = 1, 2, \ldots, \}$. Then the elements of E are sorted in ascending order and the corresponding transfers executed (provided that $\Delta(r, j, i) \leqslant \rho$ after each transfer) starting from the first, but discarding those affecting clusters already involved in a reassignment.

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 algorithm. However, global strategies may be questionable under the request of computer resources. In fact, for each pass through the data set TGBI moves at most $[k/2]$ entities which could seem unsatisfactory compared with the number of relocations performed by TFI or TLBI. It should be pointed out, though, that after some initial iterations characterized by many refinements, both TFI and TLBI 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 TGBI are invariant with respect of the entity order, whereas the ordering of the entities can have a significant impact upon the final solution of TFI and TLBI.

3.2. Swapping entities

Banfield and Bassil [8] proposed that the interchange of cluster membership between entities is a useful tool for reassigning entities.

3.2.1. Swapping pass-First-improving (SFI)

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

$$\mathbf{W}_{q+1} = \mathbf{W}_q - \beta(\mathbf{X}_r - \mathbf{X}_s)(\mathbf{X}_r - \mathbf{X}_s)^{\mathrm{t}} + (\mathbf{X}_r - \mathbf{X}_s) \times (\boldsymbol{\mu}_j - \boldsymbol{\mu}_i)^{\mathrm{t}} + (\boldsymbol{\mu}_j - \boldsymbol{\mu}_i)(\mathbf{X}_r - \mathbf{X}_s)^{\mathrm{t}}, \tag{6}$$

where $\beta = (n_i + n_j)/(n_i n_j)$. The determinant of (6) and its inverse can be computed by repeated applications of the Sherman–Morrison formula:

$$|\boldsymbol{W}_{q+1}| = |\boldsymbol{W}_q|^{*} \Delta(r, s, j, i) = |\boldsymbol{W}_q|^{*}(1 - \beta \boldsymbol{f}^{\mathrm{t}} \boldsymbol{W}_q^{-1} \boldsymbol{f}) \times (1 + \boldsymbol{g}^{\mathrm{t}} \boldsymbol{B}_q^{-1} \boldsymbol{f})(1 + \boldsymbol{f}^{\mathrm{t}} \boldsymbol{A}_q^{-1} \boldsymbol{g})$$

$$\boldsymbol{f} = (\mathbf{X}_r - \mathbf{X}_s), \quad \boldsymbol{g} = (\boldsymbol{\mu}_j - \boldsymbol{\mu}_i). \tag{7}$$

$$\boldsymbol{B}_q^{-1} = \boldsymbol{W}_q^{-1} + \frac{\beta(\boldsymbol{W}_q^{-1}\boldsymbol{f})(\boldsymbol{W}_q^{-1}\boldsymbol{f})^{\mathrm{t}}}{1 - \beta \boldsymbol{f}^{\mathrm{t}} \boldsymbol{W}_q^{-1} \boldsymbol{f}},$$

$$A_q^{-1} = \boldsymbol{B}_q^{-1} - \frac{(\boldsymbol{B}_q^{-1}\boldsymbol{f})(\boldsymbol{B}_q^{-1}\boldsymbol{g})^{\mathrm{t}}}{1 + \boldsymbol{g}^{\mathrm{t}} \boldsymbol{B}_q^{-1} \boldsymbol{f}},$$

$$\boldsymbol{W}_{q+1}^{-1} = \boldsymbol{A}_q^{-1} - \frac{(\boldsymbol{A}_q^{-1}\boldsymbol{g})(\boldsymbol{A}_q^{-1}\boldsymbol{f})^{\mathrm{t}}}{1 + \boldsymbol{f}^{\mathrm{t}} \boldsymbol{A}_q^{-1} \boldsymbol{g}},$$

$$\mu_i^{q+1} = \mu_i^q + n_i^{-1} \boldsymbol{f}, \quad \mu_j^{q+1} = \mu_j^q - n_j^{-1} \boldsymbol{f}. \tag{8}$$

The set of the combinations without replacement of n entities taken two at a time is tested—sequentially or at random—and the first swap for which $\Delta(r, s, j, i) \leqslant \rho$ is immediately executed. The centroids and the dispersion matrix are updated after each interchange (the cardinalities of course remain unchanged). The number of potential partitions examined by a SFI pass has an upper bound of $n(n-1)/2$.

3.2.2. Swapping pass-Global best-improving (SGBI)

For each scan of all possible interchanges between different clusters SGBI implements the swaps (if any) which most reduce the criterion, provided that no cluster is involved in more than one swap and that $\Delta(r, s, j, i) \leqslant \rho$ after each swap.

The cardinality of the neighborhood set of a SGBI pass is exactly $n(n-1)/2$ which makes it difficult to apply SGBI to large data sets. It should be noted, however, that SGBI is immune to the ordering problem whereas the order in which the pairs are considered has a direct effect upon the clustering quality of SFI.

The swapping pass can be combined with the transferring pass 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 may feel the impact of order dependency (with the exception of TGBI+SGBI and SGBI+TGBI).

4. Reassigning pairs of entities

Most of the k-means algorithm currently in use maintains the principle that only one entity is to be moved at a time, even though a "block move", i.e. transferring several entities simultaneously, could rescue an algorithm that has became trapped at a local minimum significantly less good than a global minimum. Block moves have been discouraged by several authors (e.g. Anderberg, [6, p. 45], Rubin [7] and Späth [9, p. 30]). Firstly, it is not proven that the final partition found with a sequence of block moves is a better partition than one obtained with single-move passes. Secondly, the computer time and memory storage needed to examine the effects of switching vector of entities is still exorbitant in the age of parallel processors. However, the simultaneous reallocation of only two entities is not more

computer demanding than an interchange of their cluster membership and the additional operations that can now be performed can balance the extra time requirement. Moreover, a double transfer includes as special cases all the schemes previously discussed so that the k-means algorithm can choose at each stage the operation which best meets the criterion.

Let $(\boldsymbol{X}_r, \boldsymbol{X}_s)$ be a pair of distinct entities with $\gamma_r = i$ and $\gamma_s = j$ and consider a pair of clusters (C_g, C_h). The analysis of a double-transfer must consider $2^6 = 64$ patterns involving clusters i, j, h, g. The patterns can be characterized by the following binary variables:

$$D_1 = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{if } i \neq j, \end{cases} \quad D_2 = \begin{cases} 1 & \text{if } h = g, \\ 0 & \text{if } h \neq g, \end{cases}$$

$$D_3 = \begin{cases} 1 & \text{if } i = g, \\ 0 & \text{if } i \neq g, \end{cases}$$

$$D_4 = \begin{cases} 1 & \text{if } j = h, \\ 0 & \text{if } j \neq h, \end{cases} \quad D_5 = \begin{cases} 1 & \text{if } i = h, \\ 0 & \text{if } i \neq h, \end{cases}$$

$$D_6 = \begin{cases} 1 & \text{if } j = g, \\ 0 & \text{if } j \neq g. \end{cases} \tag{9}$$

Some variants are inadmissible. There may be no (i, j) and (g, h) yielding sextuples of the type $(0, 1, 1, 1, x, x)$. Furthermore, sextuples like $(1, 1, 1, 1, x, x)$ leave unchanged the cluster membership and several others are equivalent. For example (1,0,1,0,0,1), (0,1,1,0,1,0) and (0,0,1,0,0,0) generate the same sequence of operations which moves $\boldsymbol{X}_s$ from cluster C_i to cluster C_h whereas $\boldsymbol{X}_r$ remains where it is. Due to the nature of the relocation phase it is possible to make certain a priori reductions and deductions so that only seven basic operations must be examined for each pair of entities.

The effect of a double-transfer can be expressed by $|\boldsymbol{W}_{q+1}| = \delta_2(i, j, g, h)|\boldsymbol{W}_q|$ where $\delta_2(i, j, g, h)$ depends on the particular operation being executed.

The k-means algorithm investigates, either in natural or in a random sequence, all possibilities of moving two entities at the same time to ascertain whether a reduction in $|\boldsymbol{W}(\boldsymbol{\gamma})|$ can be achieved. A pattern is implemented if and only if $\delta_2(i, j, g, h) \leqslant \rho$ and if no clusters will become empty after the move. This implies that the criterion will decrease monotonically at each iteration and the algorithms will converge to a local, thought not necessarily global minimum.

The upgrade of centroids and scatter matrix after a double-transfer can be calculated using Eqs. (6)–(8) for almost all the operations. Only circular shifts require new formulae

$$\boldsymbol{W}_{q+1} = \begin{cases} \boldsymbol{W}_q - \dfrac{1}{n_i}\boldsymbol{f}\boldsymbol{f}^{\mathrm{t}} + \boldsymbol{h}_{ri}\boldsymbol{h}_{ri}^{\mathrm{t}} - \boldsymbol{h}_{si}\boldsymbol{h}_{si}^{\mathrm{t}} \\ \quad -\alpha_j \mathbf{y}_j \mathbf{y}_j^{\mathrm{t}} + \alpha_h \mathbf{y}_h \mathbf{y}_h^{\mathrm{t}}, \\ \boldsymbol{W}_q - \dfrac{1}{n_j}\boldsymbol{f}\boldsymbol{f}^{\mathrm{t}} + \boldsymbol{h}_{sj}\boldsymbol{h}_{sj}^{\mathrm{t}} - \boldsymbol{h}_{rj}\boldsymbol{h}_{rj}^{\mathrm{t}} \\ \quad -\alpha_i \mathbf{y}_i \mathbf{y}_i^{\mathrm{t}} + \alpha_g \mathbf{y}_g \mathbf{y}_g^{\mathrm{t}}, \end{cases} \tag{10}$$

Obviously, circular shifts involve more arithmetic than any of the other operations.

An application rarely calls of all of the tentative placements in their full generality, so we can find an optimal implementation of a double-transfer schemes depending on the operations which are to be done most frequently. In this sense, single transfers are of principal importance since they are executed by far the most often: more than 75% in a number of experiments (not reported here) with $k \geqslant 4$, whereas the pattern involving two separate transfers and the split pattern were the least frequent. Fortunately, the upgrading formulae after a single-transfer pass are the quickest among the operations included in Table 1.

The neighborhood set of a given pair of entities is formed by all partitions that derive from the current partition by changing the membership of two entities. As mentioned earlier, the k-means algorithm can be interrupted after the first improvement (T2FI) found in the neighborhood set or after examining the whole neighborhood set (T2LBI). In the former case, a maximum of k^2 candidate partitions are evaluated while, in the latter, exactly k^2 alternatives must be analyzed. The process is carried out until no partition in the neighborhood set of any pair of entities causes the criterion to reduce. Both T2FI and T2LBI suffer from ordering effects.

It would also be possible to evaluate $|\boldsymbol{W}(\boldsymbol{\gamma})|$ for all pairs of entities and all pairs of clusters thus examining $n(n-1)k^2/2$ distinct partitions at each iteration (in this case the entity order would have no influence on the final partition). Such a large neighborhood set makes it more likely that the partition achieved will in fact be a global minimum, but the computation time which is spent for at most two changes in cluster membership dissuades from using it, particularly for data sets in which a large value of n is coupled with a large values of k.

5. Experimental investigation

This paper deals with the problem of determining which type of relocation scheme yields a better solution when inserted in a k-means algorithm based on the Friedman–Rubin criterion. To this end, the performance of the 17 methods discussed in the previous section have been evaluated in terms of quality of the clustering obtained and the time taken to achieve the result for several test data sets.

All the software has been written in Future Basic 3 language running on G4 computers using MacOS 9.0.4. Although it may be possible to appreciably improve on any one program, approximately the same effort was devoted to coding each one so that comparisons of the computational times should be quite meaningful.

5.1. Data sets

The experiments were constructed by simulating points from k m-dimensional random variables belonging to the

Table 1
Patterns of cluster membership changes in a double-transfer pass

Single transfers	$X_s \in C_i = C_j = C_g \rightarrow C_h,\ X_r \in C_j = C_i = C_h \rightarrow C_g$
Binary transfer	$X_s, X_r \in C_j \rightarrow C_g = C_h$
Confluence	$X_s \in C_i,\ X_r \in C_j \rightarrow C_h = C_g$
Swap	$X_s \in C_i \rightarrow C_j,\ X_r \in C_j \rightarrow C_i$
Split	$X_s \in C_i = C_j \rightarrow C_h,\ X_r \in C_j \rightarrow C_g$
Two transfers	$X_s \in C_i \rightarrow C_h,\ X_r \in C_j \rightarrow C_g$
Circular shifts	$X_r \in C_j \rightarrow C_g = C_i,\ X_s \in C_i \rightarrow C_h,\ X_s \in C_i \rightarrow C_h = C_j,\ X_r \in C_j \rightarrow C_g$

same family of multivariate uniform distributions having different locations but a common shape. The mean vectors are $\boldsymbol{\mu}_i = d_i \boldsymbol{c}_m,\ i = 1, 2, \ldots, k$ where $\boldsymbol{c}_m$ is a vector $(m \times 1)$ of "1". This choice eliminates the bias of criterion (3) toward a single strongly grouped variable [10]. The dependency between the variables of the clusters is specified by the variance–covariance matrix

$$\Sigma = (\sigma_{ij}) = \begin{cases} 9 & \text{if } i = j, \\ 3 & \text{if } i \neq j, \end{cases} \tag{11}$$

which ensures an adequate cohesiveness for each cluster. The reason for including uniform samples is explained by Art et al. [11] to find out how the lack of a high-density region in the middle of each cluster affects k-means algorithms. The Mahalanobis distance between $\boldsymbol{\mu}_i$ and $\boldsymbol{\mu}_j$ is proportional to $|d_i - d_j|/2$ allowing for a check whether the true centroids provide a satisfactory separation between clusters. The following values were used for the simulations: $d_1 = 74, d_2 = 62, d_3 = 49, d_4 = 35, d_5 = 20, d_6 = 4$ which determine a sufficient isolation of the clusters and prevent atypical entities such as outliers which are unduly emphasized by the Mahalanobis metric and borderline entities which are difficult to classify in a hard clustering context. From a geometrical point of view, the clusters tend to take the form of a hyper-rectangle which calls in to question the performance of k-means algorithms based on $\mathrm{Min}\{|\boldsymbol{W}(\boldsymbol{\gamma})|\}$ which is oriented to finding hyper-ellipsoidal clusters.

Thirty data sets were generated by varying k, m and the cardinalities $\{n_i\}$ covering most of the different parameters that might affect the performance of a k-means algorithm. In particular, the number of clusters was set to $2, 3, 4, 5, 6$; the number of variables ranged over $4, 6, 8$. Two patterns were selected for the cardinalities: clusters varying in size $(n_1 = 15, n_2 = 25, n_3 = 35, n_4 = 45, n_5 = 55, n_6 = 65)$ and equal sized clusters $(n_i = 40,\ i = 1, 2, \ldots, k)$.

5.2. Ordering of the entities

Since the test 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 [3]. 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. [12] suggested trying many runs with different arrangements to marginalize out ordering effects, but the number of repetitions deserves further exploration. Fisher et al. [13] argued that arrangements so that consecutive entities are dissimilar lead to good clustering. In this sense, each test data set, prior to clustering, has been reordered by using the furthest-neighbor procedure [14]. In the absence of ties the procedure of Kennard and Stone determines a uniquely defined sorting $P(r),\ r = 1, 2, \ldots, n$ which reduces the impact of the entity order upon the relocation schemes. It goes without saying that this sorting strategy is arbitrary since other methods may give similar or better orderings. Further work remains to be done on connections between sorting strategies of the data and recovery rate of combinatorial k-means algorithm based on $\mathrm{Min}\{|\boldsymbol{W}(\boldsymbol{\gamma})|\}$.

5.3. Starting partitions

The data sets used as test problems are well-structured so that every good (resp. bad) initial partition gives rise, almost without exception, to a global (resp. local) minimum and the influence of the initial conditions increases with the number of clusters. Since the goal of this study is to test the efficacy of relocation schemes rather than the adequacy of initialization methods, the k-means algorithms obtained their initial configuration by a stratified random sampling based on the true classification $\boldsymbol{\delta}$. According to this procedure, the cluster $C_i \in \boldsymbol{\gamma}^0$ contains a minimum of $n_i - [n_i/k]$ entities randomly selected from $C_i \in \boldsymbol{\delta}$ for $i = 1, 2, \ldots, k$. The other assignments are generated giving a random weight to each cluster. $N = 200$ independent random partitions were tried as starting configuration for each data set for a total of 6000 distinct experiments. The number of entities in $\boldsymbol{\gamma}^0$ which have the true membership increases with m to balance the loss of cohesiveness due to the decrease of the percentage of total variance $(100/m)[1 + (m-1)/3]$ accounted for the first principal component of (11). This, however, does not necessarily provide a systematically good starting partition even with the strongly structured data sets used in our simulations.

5.4. Computational efficiency

Table 2 reports the average Cpu-time in seconds consumed by each k-means algorithm to reach a solution.

Table 2
Computational times in seconds recorded for various relocation methods

m	Method	Equal size					Disparate size				
		2	3	4	5	6	2	3	4	5	6
4	TFI	0.0	0.0	0.2	0.4	1.0	0.0	0.0	0.1	0.3	0.8
	TLBI	0.0	0.0	0.1	0.3	0.7	0.0	0.0	0.1	0.3	0.6
	TGBI	0.0	0.0	0.1	0.1	0.2	0.0	0.0	0.0	0.1	0.2
	TFI+SFI	0.1	0.6	1.5	2.8	5.3	0.0	0.2	1.0	2.3	5.2
	TLBI+SFI	0.1	0.5	1.4	2.6	4.5	0.0	0.2	0.9	2.0	4.6
	TGBI+SFI	0.2	0.6	0.9	1.3	1.9	0.0	0.2	0.4	1.0	1.7
	TFI+SGBI	0.4	1.6	4.1	7.9	11.0	0.1	0.5	2.2	7.3	17.1
	TLBI+SGBI	0.4	1.6	4.1	7.8	11.0	0.1	0.5	2.1	7.1	16.8
	TGBI+SGBI	0.4	1.4	2.5	4.0	4.9	0.1	0.4	1.2	3.1	6.1
	SFI+TFI	0.1	0.5	1.3	2.6	4.7	0.0	0.2	0.9	2.1	4.7
	SFI+TLBI	0.1	0.5	1.2	2.4	4.0	0.0	0.2	0.8	1.8	3.8
	SFI+TGBI	0.2	0.5	0.7	1.0	1.5	0.0	0.2	0.4	0.8	1.3
	SGBI+TFI	0.4	1.5	4.0	7.7	10.6	0.1	0.5	2.1	7.1	16.8
	SGBI+TLBI	0.4	1.5	3.9	7.7	10.6	0.1	0.5	2.1	7.0	16.5
	SGBI+TGBI	0.4	1.3	2.4	3.8	4.8	0.1	0.4	1.2	3.0	5.9
	T2FI	0.2	1.5	6.0	15.5	34.3	0.1	0.6	3.5	13.4	37.7
	T2LBI	0.2	1.5	5.7	13.7	30.1	0.1	0.6	3.4	10.4	33.9
6	TFI	0.0	0.1	0.4	1.0	1.7	0.0	0.0	0.2	0.8	1.8
	TLBI	0.0	0.1	0.3	0.8	1.4	0.0	0.0	0.2	0.7	1.4
	TGBI	0.0	0.1	0.2	0.2	0.3	0.0	0.0	0.1	0.2	0.3
	TFI+SFI	0.3	1.1	3.3	5.0	8.8	0.1	0.5	2.2	5.4	13.4
	TLBI+SFI	0.3	1.0	3.1	4.9	7.9	0.1	0.5	2.1	4.6	11.5
	TGBI+SFI	0.4	1.1	1.9	2.7	3.2	0.1	0.4	1.0	2.0	3.7
	TFI+SGBI	0.8	3.4	8.3	15.8	22.8	0.1	1.0	4.4	14.0	34.4
	TLBI+SGBI	0.8	3.3	8.3	15.9	22.4	0.1	1.0	4.4	14.0	34.0
	TGBI+SGBI	0.7	2.8	6.1	7.9	15.3	0.1	0.8	2.6	6.3	12.3
	SFI+TFI	0.2	0.9	3.6	4.4	9.1	0.1	0.5	2.1	5.0	12.7
	SFI+TLBI	0.2	0.9	2.8	4.2	7.3	0.1	0.5	1.8	4.0	10.2
	SFI+TGBI	0.3	0.9	2.0	2.1	4.3	0.1	0.4	0.9	1.6	2.9
	SGBI+TFI	0.8	3.2	5.7	15.2	13.9	0.1	1.0	4.3	13.6	33.7
	SGBI+TLBI	0.8	3.2	7.9	15.2	22.1	0.1	1.0	4.3	13.3	33.2
	SGBI+TGBI	0.7	2.7	6.0	7.8	15.0	0.1	0.8	2.5	6.2	11.8
	T2FI	0.4	2.9	11.1	35.2	71.1	0.1	1.2	6.5	25.0	71.0
	T2LBI	0.4	3.0	10.0	31.7	56.6	0.1	1.2	6.1	23.7	60.5
8	TFI	0.0	0.1	0.6	1.7	3.5	0.0	0.1	0.4	1.4	3.7
	TLBI	0.0	0.1	0.5	1.5	2.8	0.0	0.1	0.3	1.1	2.9
	TGBI	0.0	0.1	0.2	0.4	0.5	0.0	0.1	0.1	0.3	0.5
	TFI+SFI	0.6	1.9	6.1	11.7	21.1	0.1	0.9	3.4	9.4	21.2
	TLBI+SFI	0.6	1.8	5.2	10.4	18.0	0.1	0.9	3.1	8.1	17.7
	TGBI+SFI	0.7	2.0	3.2	4.8	6.0	0.1	0.8	1.9	3.7	6.5
	TFI+SGBI	1.3	5.8	15.5	29.3	41.5	0.2	1.7	7.6	25.3	61.9
	TLBI+SGBI	1.3	5.7	15.6	29.6	41.3	0.2	1.7	7.4	24.8	60.9
	TGBI+SGBI	1.2	4.7	8.8	14.1	18.0	0.2	1.4	4.4	10.6	20.7
	SFI+TFI	0.5	1.5	5.5	10.6	19.3	0.1	0.8	3.1	9.2	20.4
	SFI+TLBI	0.5	1.5	4.7	9.8	16.6	0.1	0.8	2.6	7.3	16.5
	SFI+TGBI	0.6	1.8	2.5	3.8	4.5	0.1	0.6	1.6	3.0	5.0
	SGBI+TFI	1.2	5.6	14.8	28.3	39.9	0.2	1.6	7.3	24.3	60.7
	SGBI+TLBI	1.2	5.5	14.8	28.3	39.7	0.2	1.6	7.1	24.2	59.8
	SGBI+TGBI	1.1	4.5	8.5	13.7	17.0	0.2	1.3	4.2	10.3	20.0
	T2FI	0.6	4.5	19.0	53.0	112.2	0.2	1.9	11.2	43.6	126.1
	T2LBI	0.6	4.5	18.5	48.3	97.3	0.2	1.9	11.3	38.4	125.4

The scheme TGBI scores top marks in terms of convergence speed significantly better than any other scheme and on all 30 data sets. In this sense, it is, unexpectedly, a natural candidate for clustering large data sets, at least for applications where a reasonably good initial classification is available.

The other findings in Table 2 mirror the increasing complexity of the local search inserted in the k-means algorithm. 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 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. Interestingly enough, the results of combinations S+T compare favorably with those of the reverse combinations T+S. Banfield and Bassil [8] have ignored mixed methods of the type S+T which, on the contrary, seem to generate efficient schemes.

Not surprisingly, the most time consuming strategies are those using a double-transfer scheme because of the considerable space of partitions they explore. T2FI and T2LBI are more than two time slower than the slowest among all the other procedures (with the exception of $k = 2$ where their neighborhood set is smaller than that of TGBI+SGBI or SGBI+TGBI). In spite of this, the average duration of T2LBI in the largest dimensional problem ($k = 6$, $m = 8$, disparate sized clusters) has an average running time of two minutes which could be judged feasible in some situations, should the quality of the results be satisfactory. The execution time spent by T2FI was roughly the same as the time spent by T2LBI because both the first and the best local operation was quite often a single-transfer pass. As k increases T2LBI tends to be faster than T2FI.

As it may be expected, the duration of computer runs increases with k and m, but for double-transfer schemes the growth is exponential. It is also confirmed the noticeable difference in Cpu times for algorithms running on data sets with unequal cluster sizes which were much faster than running on data sets with clusters of equal size.

Schemes based on a "first-improving" policy are almost always slower than schemes based on a "local best-improving" policy both in the tandem S+ and in the tandem T+; in addition, the divergence enlarges as k and m increase. Analogous results are obtained by schemes using a double-transfer pass. This is surprising since a substantial fraction of the computation time required by any of these schemes is spent in determining the centroid closest to a particular entity and a LFI slower than a LBI disagrees strongly with our intuition. One possible reason for this results is that schemes enforcing a "best-improving" principle perform a smaller number of iterations which overcompensates the higher speed of schemes driven by a "first-improving" principle [15].

5.5. Clustering efficacy

The effectiveness of the relocation procedures has been measured by comparing the final partition $\boldsymbol{\gamma}^+$ generated by k-means algorithms with the prior knowledge of the true classification $\boldsymbol{\delta}$ (which, for the chosen data sets, it is a global minimum). In particular we used the Hubert-Arabie [16] statistic

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

$$a = \sum_{i=1}^{k} n_i^+(n_i^+ - 1), \qquad b = \sum_{i=1}^{k} n_i(n_i - 1),$$

$$c = \sum_{r=1}^{n-1} \sum_{s=r+1}^{n} \varepsilon(\gamma_r^+ = \gamma_s^+ \cap \delta_r = \delta_s), \tag{12}$$

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 $E(R_{HA}) = 0$ under the hypothesis that $\boldsymbol{\gamma}^+$ and $\boldsymbol{\delta}$ are picked at random subject to having the true number of clusters and objects in each.

Table 3, in two parts, shows the results of the simulations. In these tables are displayed the percentage T_c of runs in which $R_{HA} = 1$ and the average deviation of the criterion at $\boldsymbol{\gamma}^+$ from the value of the criterion at the true classification vector $\boldsymbol{\delta}$:

$$\frac{1}{N} \sum_{i=1}^{N} \left(\frac{|\boldsymbol{W}(\boldsymbol{\gamma}^+)| - |\boldsymbol{W}(\delta)|}{|\boldsymbol{W}(\delta)|} \right) \times 100. \tag{13}$$

Although the cluster structure in the data was reasonably clear-cut, the k-means algorithms are not equally appropriate and the quality of the solution differs as a function of m, k and the $\{n_i\}$. The key findings are listed below.

1. 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. This is apparent from Table 3a and b that clearly show the consistent superiority of TGBI+ over TFI+ or TLBI+, and +TGBI over +TFI or +TLBI.

2. Mixed schemes T+ obtain (but not always, see Ismail and Kamel [17]) 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 small 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.

Table 3
Simulation results

m	Method	$k=2$		$k=3$		$k=4$		$k=5$		$k=6$	
		Rmd	T_c	Rmd	T_c	Rmd	T_c	Rmd	T_c	Rmd	T_c
(a) *for equal size clusters*											
4	TFI	28.6	0.95	65.6	0.91	289.0	0.66	283.6	0.68	401.7	0.62
	TLBI	28.6	0.95	81.1	0.89	291.4	0.65	306.8	0.66	399.1	0.62
	TGBI	0	1.00	0.0	1.00	0.0	1.00	0.0	1.00	0.0	1.00
	TFI+SFI	8.9	0.99	2.7	1.00	21.4	0.98	20.6	0.98	5.7	1.00
	TLBI+SFI	8.9	0.99	6.1	0.99	16.6	0.98	9.8	0.99	0.0	1.00
	TBGI+SFI	3	1.00	0.0	1.00	0.0	1.00	0.0	1.00	0.0	1.00
	TFI+SGBI	2.8	1.00	3.5	1.00	10.3	0.99	8.1	0.99	0.0	1.00
	TLBI+SGBI	2.8	1.00	3.5	1.00	5.1	1.00	13.1	0.99	0.0	1.00
	TGBI+SGBI	0	1.00	0.0	1.00	0.0	1.00	0.0	1.00	0.0	1.00
	SFI+TFI	8.2	0.99	0.0	1.00	20.1	0.98	9.1	0.99	0.0	1.00
	SFI+TLBI	8.2	0.99	6.1	0.99	13.3	0.99	8.7	0.99	0.0	1.00
	SFI+TGBI	2.5	1.00	0.0	1.00	0.0	1.00	0.0	1.00	0.0	1.00
	SGBI+TFI	0	1.00	3.6	1.00	10.0	0.99	4.6	1.00	0.0	1.00
	SGBI+TLBI	0	1.00	3.6	1.00	5.1	1.00	4.6	1.00	0.0	1.00
	SGBI+TGBI	0	1.00	0.0	1.00	0.0	1.00	0.0	1.00	0.0	1.00
	T2FI	2.2	0.99	0.0	1.00	9.5	0.96	57.5	0.75	75.2	0.71
	T2LBI	1.4	0.99	3.3	0.98	4.9	0.98	5.9	0.97	9.6	0.97
6	TFI	92.2	0.71	118.5	0.88	230.7	0.79	538.7	0.50	388.7	0.63
	TLBI	92.2	0.71	138.8	0.85	230.8	0.79	553.2	0.48	378.5	0.65
	TGBI	4.6	0.98	0.0	1.00	0.0	1.00	0.0	1.00	0.0	1.00
	TFI+SFI	15.3	0.95	18.6	0.98	0.0	1.00	5.7	1.00	0.0	1.00
	TLBI+SFI	15.3	0.95	13.6	0.99	0.0	1.00	0.0	1.00	0.0	1.00
	TGBI+SFI	25.2	0.92	0.0	1.00	0.0	1.00	0.0	1.00	0.0	1.00
	TFI+SGBI	59.5	0.82	4.6	1.00	0.0	1.00	0.0	1.00	0.0	1.00
	TLBI+SGBI	59.5	0.82	4.7	1.00	0.0	1.00	0.0	1.00	0.0	1.00
	TGBI+SGBI	24.4	0.92	0.0	1.00	0.0	1.00	0.0	1.00	0.0	1.00
	SFI+TFI	16.2	0.95	13.9	0.99	0.0	1.00	10.1	0.99	5.0	1.00
	SFI+TLBI	16.2	0.95	4.6	1.00	0.0	1.00	0.0	1.00	0.0	1.00
	SFI+TGBI	39.4	0.87	0.0	1.00	0.0	1.00	0.0	1.00	0.0	1.00
	SGBI+TFI	60.2	0.81	9.6	0.99	5.7	1.00	5.5	1.00	0.0	1.00
	SGBI+TLBI	60.2	0.81	5.1	1.00	5.7	1.00	5.7	1.00	0.0	1.00
	SGBI+TGBI	24.3	0.93	0.0	1.00	4.2	1.00	0.0	1.00	4.5	1.00
	T2FI	5.6	0.92	1.3	1.00	18.6	0.93	96.3	0.64	106.8	0.58
	T2LBI	7.5	0.90	7.1	0.97	2.7	0.99	36.7	0.87	13.4	0.95
8	TFI	153	0.64	69.5	0.94	253.6	0.71	653.9	0.34	609.3	0.44
	TLBI	153	0.64	86.5	0.92	257.5	0.71	675.8	0.33	621.8	0.42
	TGBI	16.2	0.96	0.0	1.00	0.0	1.00	0.0	1.00	0.0	1.00
	TFI+SFI	60.8	0.85	0.0	1.00	18.4	0.98	24.4	0.98	12.3	0.99
	TLBI+SFI	60.8	0.85	0.0	1.00	0.0	1.00	17.6	0.98	3.1	1.00
	TGBI+SFI	64	0.84	0.0	1.00	0.0	1.00	0.0	1.00	0.0	1.00
	TFI+SGBI	79.9	0.81	0.0	1.00	22.1	0.98	10.0	0.99	3.1	1.00
	TLBI+SGBI	79.9	0.81	0.0	1.00	13.0	0.99	10.3	0.99	3.1	1.00
	TGBI+SGBI	56.4	0.85	0.0	1.00	0.0	1.00	0.0	1.00	0.0	1.00
	SFI+TFI	58.5	0.85	4.1	1.00	4.9	1.00	33.5	0.97	3.1	1.00
	SFI+TLBI	58.5	0.85	9.3	0.99	8.5	0.99	23.0	0.98	3.1	1.00
	SFI+TGBI	71.8	0.82	0.0	1.00	0.0	1.00	0.0	1.00	0.0	1.00
	SGBI+TFI	91.3	0.78	0.0	1.00	18.9	0.98	18.6	0.98	3.1	1.00
	SGBI+TLBI	91.3	0.78	0.0	1.00	18.2	0.98	13.9	0.99	0.0	1.00
	SGBI+TGBI	54.1	0.86	0.0	1.00	0.0	1.00	0.0	1.00	0.0	1.00
	T2FI	2	0.98	2.3	0.99	27.4	0.88	197.0	0.18	208.4	0.25
	T2LBI	2.2	0.97	1.3	1.00	12.3	0.95	118.1	0.52	66.9	0.76

Table 3 (*continued*)

m	Method	$k=2$		$k=3$		$k=4$		$k=5$		$k=6$	
		Rmd	T_c	Rmd	T_c	Rmd	T_c	Rmd	T_c	Rmd	T_c
(b) *for disparate sized clusters*											
4	TFI	87.2	0.85	63.1	0.90	172.6	0.65	384.5	0.33	305.3	0.21
	TLBI	87.2	0.85	77.6	0.88	179.8	0.64	382.9	0.34	299.3	0.23
	TGBI	11.7	0.98	0.0	1.00	0.0	1.00	0.0	1.00	0.0	1.00
	TFI+SFI	48.2	0.92	44.0	0.94	32.8	0.92	103.5	0.79	106.8	0.69
	TLBI+SFI	48.2	0.92	27.6	0.96	32.1	0.93	70.1	0.85	94.3	0.73
	TBGI+SFI	51.1	0.91	0.0	1.00	0.0	1.00	0.0	1.00	0.0	1.00
	TFI+SGBI	58.7	0.90	34.9	0.94	58.8	0.89	38.7	0.95	26.9	0.94
	TLBI+SGBI	58.7	0.90	24.8	0.96	54.0	0.90	38.2	0.95	26.7	0.95
	TGBI+SGBI	17.3	0.97	5.3	0.99	0.0	1.00	0.0	1.00	0.0	1.00
	SFI+TFI	47	0.92	44.2	0.94	37.5	0.91	98.1	0.81	98.9	0.72
	SFI+TLBI	47	0.92	23.6	0.97	34.8	0.92	75.5	0.84	70.8	0.80
	SFI+TGBI	42.9	0.93	0.0	1.00	0.0	1.00	0.0	1.00	0.0	1.00
	SGBI+TFI	44	0.92	26.7	0.95	65.8	0.88	41.9	0.95	34.9	0.94
	SGBI+TLBI	44	0.92	29.6	0.95	54.3	0.90	33.9	0.95	31.1	0.94
	SGBI+TGBI	23.3	0.96	0.0	1.00	0.0	1.00	0.0	1.00	0.0	1.00
	T2FI	2.4	0.99	15.5	0.90	28.0	0.82	124.3	0.12	143.9	0.05
	T2LBI	2.7	0.98	4.9	0.97	6.4	0.95	87.0	0.57	54.1	0.63
6	TFI	167.2	0.52	259.7	0.59	421.4	0.36	616.4	0.31	411.3	0.31
	TLBI	167.2	0.52	264.0	0.59	405.0	0.39	636.5	0.31	404.5	0.33
	TGBI	78.6	0.79	2.7	1.00	0.0	1.00	0.0	1.00	0.0	1.00
	TFI+SFI	144.3	0.59	77.9	0.87	80.5	0.88	120.1	0.86	95.7	0.77
	TLBI+SFI	144.3	0.59	67.4	0.89	76.3	0.89	61.7	0.93	81.1	0.79
	TGBI+SFI	142.2	0.64	13.7	0.98	0.0	1.00	0.0	1.00	0.0	1.00
	TFI+SGBI	169.1	0.56	72.7	0.87	68.4	0.90	141.1	0.87	21.2	0.97
	TLBI+SGBI	169.1	0.56	76.1	0.87	64.7	0.91	130.6	0.87	17.1	0.98
	TGBI+SGBI	126.8	0.67	2.4	1.00	6.1	0.99	0.0	1.00	0.0	1.00
	SFI+TFI	148.1	0.60	87.6	0.86	69.8	0.89	117.2	0.86	103.2	0.77
	SFI+TLBI	148.1	0.60	62.9	0.90	85.9	0.87	73.8	0.92	69.9	0.83
	SFI+TGBI	158.2	0.59	23.3	0.97	6.1	0.99	0.0	1.00	0.0	1.00
	SGBI+TFI	167.5	0.56	76.9	0.86	71.6	0.90	130.5	0.87	29.0	0.96
	SGBI+TLBI	167.5	0.56	69.6	0.88	65.4	0.91	132.3	0.87	20.0	0.97
	SGBI+TGBI	127.2	0.66	4.5	0.99	6.3	0.99	0.0	1.00	0.0	1.00
	T2FI	9.6	0.87	28.3	0.82	110.8	0.33	190.2	0.09	136.8	0.18
	T2LBI	13	0.85	8.5	0.95	30.3	0.82	106.9	0.51	9.8	0.93
8	TFI	316.5	0.28	327.1	0.41	995.6	0.27	441.1	0.29	702.9	0.19
	TLBI	316.5	0.28	329.1	0.41	1017.4	0.26	417.6	0.31	721.4	0.21
	TGBI	140.8	0.68	7.4	0.99	0.0	1.00	0.0	1.00	0.0	1.00
	TFI+SFI	199.7	0.54	126.8	0.76	279.0	0.80	69.5	0.85	96.2	0.75
	TLBI+SFI	199.7	0.54	108.5	0.79	190.3	0.87	46.3	0.90	56.0	0.85
	TGBI+SFI	193	0.54	20.9	0.96	0.0	1.00	0.0	1.00	0.0	1.00
	TFI+SGBI	215.8	0.52	126.9	0.76	220.3	0.84	92.3	0.87	54.5	0.94
	TLBI+SGBI	215.8	0.52	135.2	0.74	199.8	0.86	89.4	0.88	49.4	0.95
	TGBI+SGBI	178.3	0.60	30.7	0.94	6.9	1.00	0.0	1.00	0.0	1.00
	SFI+TFI	196.8	0.54	141.9	0.73	237.5	0.83	74.7	0.87	105.9	0.77
	SFI+TLBI	196.8	0.54	115.7	0.77	167.6	0.88	40.4	0.92	61.5	0.84
	SFI+TGBI	191.1	0.54	27.5	0.94	0.0	1.00	0.0	1.00	0.0	1.00
	SGBI+TFI	206.9	0.53	117.8	0.76	176.9	0.87	93.3	0.87	55.8	0.94
	SGBI+TLBI	206.9	0.53	117.5	0.76	193.4	0.86	87.6	0.87	50.0	0.95
	SGBI+TGBI	162.2	0.62	42.3	0.92	13.7	0.99	0.0	1.00	0.0	1.00
	T2FI	42.6	0.54	67.9	0.39	265.0	0.19	158.9	0.09	205.3	0.04
	T2LBI	40.2	0.57	40.8	0.65	174.5	0.49	55.5	0.66	76.3	0.59

3. As can be inferred from Tables 3a and b, 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 exploit the fact that most of the changes in cluster membership occurs at the first few iterations [6, p. 163].

4. 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 $\mathrm{Min}\{|\boldsymbol{W}(\gamma)|\}$ encourages the formation of partitions with clusters of equal size when the separation between the clusters is not large [18,19, p. 94].

5. The double-transfer schemes detect the true cluster structure only to a limited extent. According to the figures in the last two rows of Table 3a and b, the recovery rate of T2FI and T2BLI is consistently lower than the corresponding single-transfer schemes. A possible explanation for this unexpected lack of accuracy is the dominant frequency of the relatively inefficient and restrictive single-transfer pass as the most convenient operation among the ones allowed by a double-transfer pass.

6. The global effectiveness of k-means solutions becomes admittedly worse as the number of variables increases. This is somewhat unusual since previous clustering research (e.g. Milligan [20]) indicate that cluster recovery tends to increase with increasing dimensionality. There are two possible reasons for this. The first one is that the decline of the percentage of total variation explained by the first principal component of (11) determines progressively (though slightly) less compact clusters as m becomes larger. Secondly, the rapid diminution of the percentage of the cluster variance explained by the other components adds poor discriminatory variables to the representation of the entities. This reduces the accuracy of k-means algorithm because the fixed number of entities in the clusters tends to concentrate in the corners as the dimensionality of the data sets increases.

6. Conclusion

The purpose of the present study was to analyze 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 methods are based on a combination of transfers and swaps. The transfers are either considered as a stand-alone tool or alternate with the swaps as part of a mixed scheme. In the latter case, the swaps are used both as pre-processing stage determining profitable initial partitions for the transfers and as a post-processing stage to improve the quality of the solution determined by transfers. In addition, strategies moving two entities at a time were investigated in detail.

The best results in terms of clustering quality and robustness were obtained by TGBI, that is a relocation method which, for each scan of the data set, executes only the best transfers involving pairs of distinct clusters. 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 (for the present technology) computer resources.

The complexity of mixed schemes is higher than pure schemes due to their more intricate search, but the limited impact on the classification adequacy does not compensate the extra energy expended for these procedures. The experiments of Section 5 indicate that combinations of different strategies may provide significantly less good performance than do their isolate application. The swaps phase should be essentially considered as a way of getting out of a local minimum.

A major gulf between demand of computer resources and quality of the outcome is noticed for double-transfer schemes which, despite their enormous potential impact and their formidable computational cost, produced poor-quality results and do not seem worthy of further consideration for the Friedman–Rubin criterion (at least for local search strategies).

An inherent limitation of the k-means algorithms tried in this paper is that their final partition 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 contribution.

7. Summary

The purpose of this paper is to report and discuss the results of an empirical investigation of several techniques used by k-means algorithms (based on the Friedman–Rubin approach) to move entities from one cluster to another. Most of these procedures differ basically in the number of criterion evaluations required to reach an optimum and the accuracy of this optimum. The prime objective of the current research study has been to establish the relative merits of 17 combinatorial passes by comparing them across a variety of artificial data sets. The experimental results suggest that a direct and efficient search which moves down the steepest permissible direction globally outperforms both simple and more sophisticated reassignment methods in terms of grouping efficacy and numerical efficiency.

References

[1] A.D. Gordon, Classification, 2nd Edition, Chapman & Hall, London, 1999.

[2] H.P. Friedman, J. Rubin, On some invariant criteria for grouping data, J. Am. Stat. Assoc. 62 (1967) 1159–1178.
[3] J. MacQueen, Some methods for classification and analysis of multivariate observations, in: L.M. LeCam, J. Neyman (Eds.), Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Vol. 1—Statistics, UCLA Press, Berkeley, USA, 1967, pp. 281–297.
[4] E. Forgy, Cluster analysis of multivariate data. Efficiency vs. Interpretability of classification. WNAR Meetings, UCLS Riverside, CA, June 22–23, 1965. Abstract in Biometrics 21 (1965) 768.
[5] S.Z. Selim, M.A. Ismail, *K*-means type algorithm: generalized convergence theorem and characterization of local optimality, IEEE Trans. Pattern Anal. Mach. Intell. 6 (1984) 81–87.
[6] M.R. Anderberg, Cluster Analysis for Applications, Academic Press, New York, 1973.
[7] J. Rubin, Optimal classification into groups: an approach for solving the taxonomy problem, J. Theoret. Biol. 15 (1967) 103–144.
[8] C.F. Banfield, L.C. Bassil, Algorithm AS113. A transfer algorithm for non-hierarchical classification, Appl. Stat. 36 (1977) 206–210.
[9] H. Späth, Cluster Dissection and Analysis, Theory, FORTRAN Programming, Examples, Ellis Horward Limited, Chichester, UK, 1985.
[10] F.H.C. Marriott, Practical problems in a method of cluster analysis, Biometrics 27 (1971) 501–514.
[11] D. Art, R. Gnanadesikan, J.R. Kettenring, Data-based metrics for cluster analysis, Utilitas Math. 21A (1982) 75–99.
[12] J.M Peña, J.A. Lozano, P. Larrañaga, An empirical comparison of four initialization methods for the *k*-means algorithm, Pattern Recognition Lett. 20 (1999) 1027–1040.
[13] D. Fisher, L. Xu, N. Zard, Ordering effects in clustering, Proceedings of the Nineth International Conference on Machine Learning, Morgan Kauffman, San Mateo, USA, 1992, pp. 163–168.
[14] R.W. Kennard, L.A. Stone, Computer aided design of experiments, Technometrics 11 (1969) 137–148.
[15] E.J. Anderson, Mechanisms for local search, Eur. J. Oper. Res. 88 (1996) 139–151.
[16] L.J. Hubert, P. Arabie, Comparing partitions, J. Classification 2 (1985) 193–218.
[17] M.A. Ismail, M.S. Kamel, Multidimensional data classification utilizing hybrid search strategies, Pattern Recognition 22 (1989) 75–89.
[18] B.S. Everitt, S. Landau, M. Lase, Cluster Analysis, 4th Edition, Arnold, London, 2001.
[19] A.J. Scott, M.J. Symons, Clustering method based on likelihood ratio criteria, Biometrics 27 (1971) 387–397.
[20] G.W. Milligan, An examination of the effect of six types of error perturbation on fifteen clustering algorithms, Psychometrika 45 (1980) 325–342.

About the Author—AGOSTINO TARSITANO received his degree in economics from Calabria University (Italy) in 1978. He has been on the Faculty of economics at the Calabria University since 1978, an Associate Professor since 1992. He was with the University of Utah of Salt Lake City during the academic year 1986–1987. His present research interests are in the area of multivariate analysis, remote sensing, statistics for the regional sciences.