Available online at www.sciencedirect.com
ScienceDirect
Computational Statistics & Data Analysis 51 (2006) 1623–1642
COMPUTATIONAL STATISTICS & DATA ANALYSIS
www.elsevier.com/locate/csda

Three-mode partitioning

Jan Schepers*, Iven van Mechelen, Eva Ceulemans

Department of Psychology, Katholieke Universiteit Leuven, Tiensestraat 102, Leuven, Belgium

Received 24 November 2005; received in revised form 1 June 2006; accepted 1 June 2006
Available online 30 June 2006

Abstract

The three-mode partitioning model is a clustering model for three-way three-mode data sets that implies a simultaneous partitioning of all three modes involved in the data. In the associated data analysis, a data array is approximated by a model array that can be represented by a three-mode partitioning model of a prespecified rank, minimizing a least squares loss function in terms of differences between data and model. Algorithms have been proposed for this minimization, but their performance is not yet clear. A framework for alternating least-squares methods is described in order to offset the performance problem. Furthermore, a number of both existing and novel algorithms are discussed within this framework. An extensive simulation study is reported in which these algorithms are evaluated and compared according to sensitivity to local optima. The recovery of the truth underlying the data is investigated in order to assess the optimal estimates. The ordering of the algorithms with respect to performance in finding the optimal solution appears to change as compared to the results obtained from the simulation study when a collection of four empirical data sets have been used. This finding is attributed to violations of the implicit stochastic model underlying both the least-squares loss function and the simulation study. Support for the latter attribution is found in a second simulation study.
© 2006 Elsevier B.V. All rights reserved.

Keywords: Three-way data; Clustering; *k*-means; Alternating least-squares algorithms

1. Introduction

N -way data in general offer a wealth of information to the data analyst. However, this information is usually very complex and hard to grasp. A way out for this can be to reduce one or more of the involved modes to a small number of classes or dimensions. In this paper, we are concerned with a simultaneous reduction of the three modes of a three-way three-mode data set. More in particular, we will focus on the most simple case: the simultaneous partitioning of all three modes involved in the data. Recently, the three-mode partitioning model was proposed independently by several authors (Rocci and Vichi, 2003; Kiers, 2004; Schepers and Van Mechelen, 2004). The model is in fact a direct generalization of the simultaneous two-mode partitioning models proposed by Gaul and Schader (1996), Baier et al. (1997), Govaert (1995) and Vichi (2002). For a comprehensive overview of two-mode clustering methods, see Van Mechelen et al. (2004).

A problem that is still left unresolved is the estimation of the three-mode partitioning model. More in particular, a number of *k*-means type algorithms have been proposed by the different authors of the model, but so far the performance of these algorithms is not yet clear, let alone, what is the preferred one. One of the reasons to be careful in this regard is

* Corresponding author. Tel.: +32 16 326095;fax: +32 16 325993.
E-mail address: jan.schepers@psy.kuleuven.be (J. Schepers).

0167-9473/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.csda.2006.06.002

the well-known fact that k-means type algorithms suffer from local minima problems (Selim and Ismail, 1984; Steinley, 2003). Considering the increased complexity when clustering more than a single mode, there seems to be an obvious need to assess the performance of three-mode partitioning algorithms. The present paper will examine the performance of several such algorithms by means of a simulation study and by means of a test on four empirical data sets.

The remainder of the paper is organized as follows: Section 2 recapitulates the three-mode partitioning model. Section 3 presents a framework for alternating least-squares methods for estimating the model. In Section 4, a simulation study is presented in which several existing and novel algorithms are evaluated in terms of their capability of minimizing the loss function and in terms of their capability to recover the truth underlying the data. In Section 5, we report the results of the application of the same algorithms to four empirical data sets. Unexpectedly, the latter results will appear not to be in line with those of the simulation study. An explanation of this finding will be looked for by means of parametric bootstrap tests. This explanation will further be tested in a second simulation study that will be presented in Section 6. Section 7 will present a few concluding remarks.

2. Three-mode partitioning

2.1. Data

A three-way data set defines a mapping from the Cartesian product of three sets of entities to some value set (say the set of reals $\mathbb{R}$). If the three sets of the Cartesian product are all distinct, the data set is referred to as three-way three-mode (Carroll and Arabie, 1980). In many areas of sciences, this type of data often occurs. Examples include data in personality psychology pertaining to the intensity of different behaviors as elicited by various situations for different persons, and data in marketing research pertaining to the perceived usefulness of different products for different goals as judged by diverse age groups. In the remainder of this paper, we will refer to the horizontal slices of three-way three-mode data as objects, to the lateral slices as attributes and to the frontal slices as sources. We will denote the data points by d_{ijk}, referring to the value of the ith object on the jth attribute according to the kth source.

2.2. Models

A three-mode partitioning implies a decomposition of an $I \times J \times K$ real-valued model or reconstructed data array $\underline{\mathbf{M}}$ into an $I \times P$ object partition matrix $\mathbf{A}$, a $J \times Q$ attribute partition matrix $\mathbf{B}$, a $K \times R$ source partition matrix $\mathbf{C}$ and a $P \times Q \times R$ real-valued array $\underline{\mathbf{W}}$, with (P, Q, R) being the rank of the model. More in particular the model array can be written as

$$m_{ijk} = \sum_{p=1}^{P} \sum_{q=1}^{Q} \sum_{r=1}^{R} a_{ip} b_{jq} c_{kr} w_{pqr} \quad \forall i, j, k, \tag{1}$$

where a_{ip}, b_{jq} and c_{kr} indicate whether or not object i, attribute j and source k belong to cluster p, q and r, respectively, and where w_{pqr} is a real-valued number associated with the data cluster indexed by p, q, r, which in turn equals the Cartesian product of object cluster p, attribute cluster q, and source cluster r. The rows of the matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ are restricted to sum to 1 and no empty clusters (zero columns) are allowed so as to represent proper partitioning structures. Eq. (1) implies that m_{ijk} equals w_{pqr} iff (i, j, k) belongs to the data cluster indexed by p, q, r. Of course, small deviations between the actual and reconstructed data points are to be expected in practice. In order to link the model to a given data set, an error term e_{ijk} is therefore assumed,

$$d_{ijk} = m_{ijk} + e_{ijk} \quad \forall i, j, k. \tag{2}$$

In absence of further assumptions on e_{ijk}, (1) can be considered to denote a deterministic model. Optionally, however, a stochastic version of (1) can be obtained by making additional assumptions about the distribution of the error component in (2), for example, the e_{ijk} can be assumed to be iid normally distributed with zero mean, yielding

$$d_{ijk} \overset{\text{iid}}{\sim} N\left(m_{ijk}, \sigma^2_{ijk}\right), \tag{3}$$

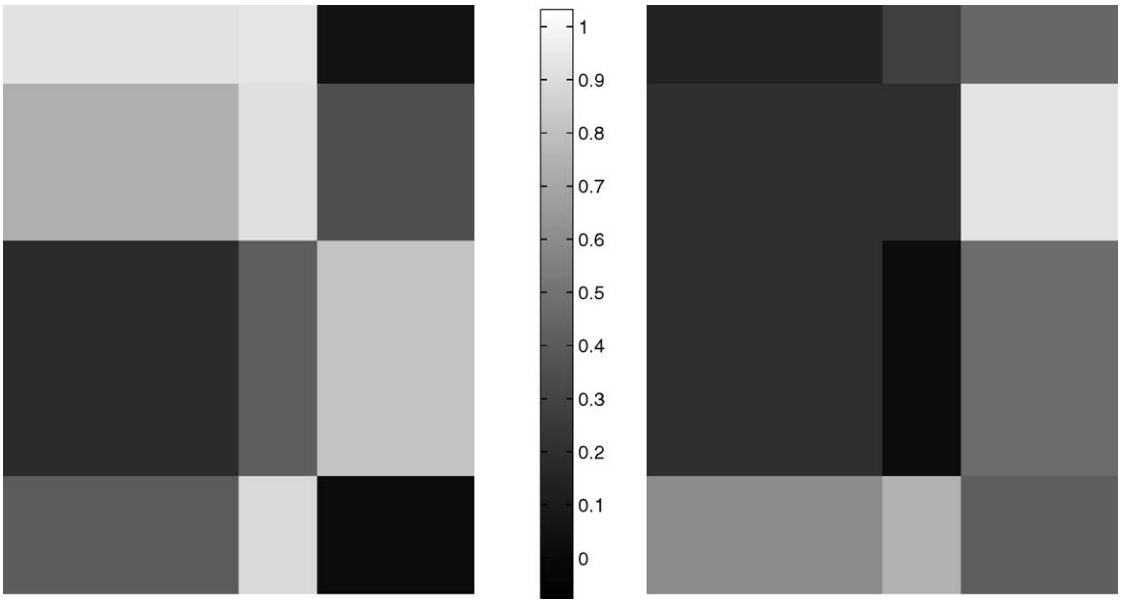

Fig. 1. Two heat maps corresponding to two reconstructed data slices (i.e., two source elements) that belong to different source clusters. The objects and attributes are permuted such that, for these two modes, entities assigned to the same cluster are displayed adjacent to one another. In this representation, four object and three attribute clusters are clearly visible.

where the mean m_{ijk} is defined as in (1) and the variance σ^2_{ijk} is the variance of the error component. Typically, further restrictions may be put on the error distribution. For example, all σ^2_{ijk} can be assumed to be equal (homoscedastic case), yielding

$$d_{ijk} \overset{\text{iid}}{\sim} N\left(m_{ijk}, \sigma^2\right). \tag{4}$$

A less restrictive assumption could read that σ^2_{ijk} depends on the data cluster to which the data point (i, j, k) is assigned. Note that in all cases as considered in the present paper, the entries of the **A**, **B** and **C** matrices are considered fixed constants (rather than as realizations of latent, e.g., multinomially distributed, variables), implying that the stochastic models under study can be considered so-called fixed-partition models (Bock, 1996).

2.3. Graphical representation

A three-mode partitioning model can be given a graphical representation in the form of a heat map that visualizes the reconstructed data values in such a way that, after permuting the entities of all three modes in such a way that objects (resp. attributes, sources) assigned to the same cluster are positioned adjacent to one another, darker grey values represent lower values and lighter grey values higher ones. A hypothetical example of two heat maps for two reconstructed data slices (i.e., source elements) of a three-mode partitioning is given in Fig. 1. This type of visualization nicely illustrates the data compression that is achieved by a three-mode partitioning: whereas the original data array $\underline{\mathbf{D}}$ consists of $I \times J \times K$ possibly different data values, the heat map of the reconstructed data array includes at most $P \times Q \times R$ different values.

In an alternative heat map representation, one may display the actual data values rather than the reconstructed ones, after permuting the entities of all three modes in such a way that objects (resp. attributes, sources) assigned to the same cluster are positioned adjacent to one another. The advantage of the latter type of visualization is that, apart from the partitioning structure, it also allows one to capture how well the model explains the data. Note that this type of visualization fits within a non-destructive approach to data analysis (Murtagh, 1989). Fig. 2 illustrates this type of representation for the same hypothetical example as in Fig. 1.

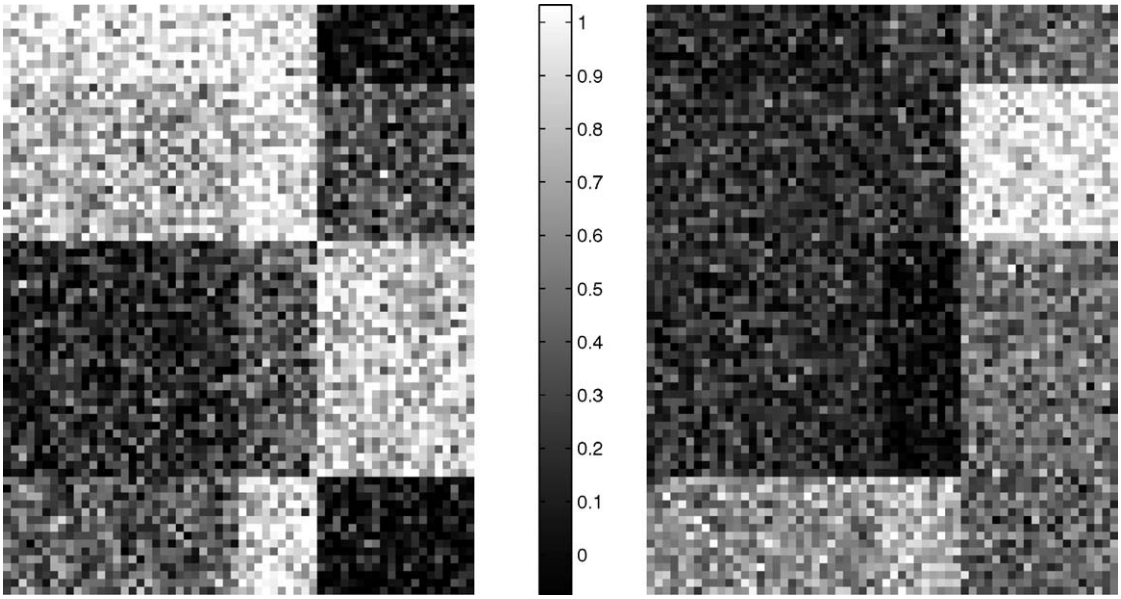

Fig. 2. Non-destructive heat maps of the same two data slices that are depicted in Fig. 1. The noise present in this case equals to 30% of the total variance in the data.

2.4. Data analysis

In order to fit the deterministic model in (1) to a given data set $\underline{\mathbf{D}}$, the following loss function is minimized with respect to $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\underline{\mathbf{W}}$:

$$f\left(\mathbf{A}, \mathbf{B}, \mathbf{C}, \underline{\mathbf{W}}\right) = \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K}\left(d_{ijk} - m_{ijk}\right)^2 = \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K}\left(d_{ijk} - \sum_{p=1}^{P}\sum_{q=1}^{Q}\sum_{r=1}^{R} a_{ip}b_{jq}c_{kr}w_{pqr}\right)^2, \tag{5}$$

where d_{ijk} is the data value for object i, attribute j and source k. Expression (5) shows that the model is fitted in terms of a least-squares loss function, implying that one tries to find data clusters that are as homogeneous as possible in the least-squares sense. One may note that, given partition matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$, the conditionally optimal $\underline{\mathbf{W}}$ follows directly from (1) and (5):

$$w_{pqr} = \frac{\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K} a_{ip}b_{jq}c_{kr}d_{ijk}}{\left(\sum_{i=1}^{I} a_{ip}\right)\left(\sum_{j=1}^{J} b_{jq}\right)\left(\sum_{k=1}^{K} c_{kr}\right)} \quad \forall p, q, r, \tag{6}$$

implying that the best possible estimate of w_{pqr} is the mean of all data points for which it holds that the corresponding objects, attributes and sources belong to clusters p, q and r, respectively. By plugging this conditionally optimal estimate of $\underline{\mathbf{W}}$ into (1) we obtain:

$$f'(\mathbf{A}, \mathbf{B}, \mathbf{C}) = \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K}\left(d_{ijk} - \sum_{p=1}^{P}\sum_{q=1}^{Q}\sum_{r=1}^{R} a_{ip}b_{jq}c_{kr}\frac{\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K} a_{ip}b_{jq}c_{kr}d_{ijk}}{\sum_{i=1}^{I} a_{ip}\sum_{j=1}^{J} b_{jq}\sum_{k=1}^{K} c_{kr}}\right)^2. \tag{7}$$

In this way the data-analytic problem in (5) is recast into a minimization problem with respect to three arguments only: $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$, which shows that the minimization problem at hand inherently has a finite solution space. It must be noted

that the solution that minimizes (5) is necessarily also the one that minimizes (7). However, further in this paper, it will become clear that this difference in perspective on the minimization problem at hand is not trivial since it implies the use of different optimization algorithms, some of which are more prone to end up in locally optimal solutions than others.

In order to estimate stochastic model variations such as (3) and (4) one may adopt a classification likelihood approach (see e.g. John, 1970; Bryant and Williamson, 1978; McLachlan, 1982; Celeux and Govaert, 1992; Banfield and Raftery, 1993; Govaert and Nadif, 2003). Note that it can easily be shown that maximizing the classification likelihood for the homoscedastic case (4) is equivalent to minimizing (5).

2.5. Uniqueness

The solution with the smallest possible numbers of partition classes that decomposes a particular model array $\underline{\mathbf{M}}$ according to (1) is unique, apart from column permutations of the partition matrices **A**, **B** and **C**.

Proof. Suppose that two different decompositions exist that reproduce the same $\underline{\mathbf{M}}$. If the decompositions are different, this implies that they include for at least one of the modes (say, the object mode) partition matrices **A** and $\mathbf{A}^*$ that differ more than upon a permutation of their columns only. This implies that there exist at least two objects that belong to the same cluster c in one decomposition and to different clusters c' and c'' in the other one. From the former, it follows that the two objects in question, as well as all other objects of c' and c'', must have identical slices in $\underline{\mathbf{M}}$. This implies that c' and c'' can be merged into a single cluster without affecting $\underline{\mathbf{M}}$ and, hence, that a decomposition of $\underline{\mathbf{M}}$ exists that includes a smaller number of classes; this further violates the assumption that the number of partition classes in the decompositions under study was minimal. □

3. Framework for ALS procedures

In order to obtain a solution to the minimization problem in (5) or (7), one can either use a global optimization technique (one that necessarily yields the global optimum) or some kind of heuristic procedure. Examples of algorithms that belong to the former class of techniques (which is only feasible for very small data sets) include complete enumeration and dynamic programming (see e.g. Hubert et al., 2001). Examples within the latter group of techniques include simulated annealing, tabu search, genetic algorithms, differential evolution, particle swarm optimization, greedy algorithms, alternating least-squares algorithms, etc. (see e.g. Al-Sultan and Maroof Khan, 1996; Paterlini and Krink, 2006).

In this paper we will only deal with heuristic procedures. In particular, we will focus on one specific group within these heuristics, namely ALS (Alternating Least Squares) procedures to estimate the three-mode partitioning model and we present a framework for them.

ALS procedures are iterative, alternating algorithms that require the total set of parameters to be partitioned into two or more subsets. By this partitioning, the estimation problem is turned into a sequence of smaller, more manageable subproblems. After the assignment of initial values to the parameter subsets, ALS algorithms conditionally update each subset, keeping the other subsets fixed. Within every such conditional update the loss function decreases (or at least stays the same). As the loss function (5) is positive and the solution space is finite, convergence is reached within a finite number of steps. However, this may occur at a local optimum in the solution space.

Below, we will first discuss the initialization of the parameter values in Section 3.1. Second, we will discuss the general updating scheme of the distinct parameter subsets, taking into account the different possible formalizations of the loss function that may be considered ((5) versus (7)) in Section 3.2. Finally, in the process of estimating the three-mode partitioning model for a given data set, empty clusters may show up. A variety of strategies can be thought of as to how to deal with such empty clusters. This topic will be discussed in Section 3.3.

3.1. Initialization of the parameter subsets

Initial values are required for **A**, **B** and **C**, and, depending on the type of loss function one considers, possibly also $\underline{\mathbf{W}}$. Since a closed form solution for $\underline{\mathbf{W}}$, given **A**, **B** and **C**, is readily available through (6), we will not consider the initialization of $\underline{\mathbf{W}}$ further.

We identify two aspects on which any type of initialization procedure may be qualified. First, a distinction can be drawn between rationally (also called smart) and randomly chosen starts. Regarding the rational-random aspect, in between the two extremes, perturbed rational starts may be situated. As a second qualitative aspect, we may distinguish between single and multistarts. Note that it is not possible to perfectly disentangle the two aspects in question since a rationally chosen start often implies the use of a single start only.

One may note that the qualitative aspects as outlined above are addressed in discussions on the quality of the final solutions as obtained from ALS algorithms. For example, the idea behind rational starts is that such starts may lie close to the globally optimal solution and that ALS algorithms therefore, when making use of such starts, will be less likely to get stuck in locally optimal solutions. Simulation results indeed suggest that better one-mode *k*-means solutions can be obtained by using a rational start (Milligan, 1980). Steinley (2003) from his part, however, investigated in a more thorough way the local minima problem for one-mode *k*-means, as implemented within three well-known general purpose statistical software packages (SPSS, SYSTAT and SAS), which all rely on a single rational start. By comparing the results provided by these packages with those of a *k*-means with a random multistart procedure, he concluded that the methods implemented in the commercial packages are very likely to end up in local optima. The author therefore suggests that, when it comes down to finding the optimal *k*-means clustering, the "quality of the starts" is less important than the "quantity of the starts". A similar conclusion was later drawn by Hand and Krzanowski (2005) on the basis of their simulation results regarding one-mode *k*-means clustering.

In what follows we will discuss more in detail three types of starts, distinguished on the basis of the first aspect (rational vs. random): rational, perturbed rational, and random starts.

3.1.1. Rational start

In principle, one could consider an almost unlimited number of techniques to obtain a rational start. Here, however, we will limit the discussion to two techniques that are based on one-mode *k*-means clustering on the one hand (*independent* and *sequential k-means*), and a single technique that is based on a full three-way method on the other hand (*Tucker3*).

Independent k-means: A smart initial solution for ALS procedures for three-mode partitioning may be obtained by performing independent one-mode *k*-means analyses on the three possible matricized forms of the three-way data array and by using their outcomes as initial estimates of **A**, **B** and **C**, respectively. For example, to obtain a rational start for **A**, one may first "flatten out" the three-mode data array $\underline{\mathbf{D}}$ to a data matrix **D** with dimensions $I \times JK$, and then search for the optimal one-mode *k*-means partitioning of the objects, based on their values with regard to all elements of the Cartesian product of attributes and sources. Note that the use of one-mode *k*-means requires suitable choices with regard to all three aspects that are to be dealt with for three-mode partitioning as well (choice of initialization method, choice of ALS updating scheme, choice of empty cluster procedure).

Sequential k-means: Similarly to *independent k-means*, this type of rational start is based on one-mode *k*-means clustering (with all implied choices). More in particular, a one-mode *k*-means partitioning is first obtained for one of the modes, say, the objects. Then, a partitioning of the attributes is obtained using the matrix of centroids from the first clustering procedure, weighted by the cardinalities of the object clusters, as the input data matrix. Finally, a partitioning of the sources is found using the centroid matrix from the second clustering procedure, weighted by the cardinalities of the object and attribute clusters.

Tucker3: A Tucker3 rational start is obtained by making use of a technique that is specifically designed to deal with three-way data, namely Tucker3 analysis (see e.g. Kroonenberg and De Leeuw, 1980). This method, which implies an approximate componential decomposition of three-way data, leads to column-wise orthonormal component matrices for the three modes. Any procedure that transforms these component matrices into partition matrices then returns a rational start for the three-mode partitioning model. In this regard, Kiers (2004) suggested that (a) performing a VARIMAX rotation of all three component matrices (which is possible because of the rotational freedom of the Tucker3 model), (b) multiplying columns having negative sums by -1, and (c) replacing all row-wise highest elements with 1 and all other elements of the same row by 0, may provide a good rational start for the three-mode partitioning model.

3.1.2. Perturbed rational start

A perturbed rational start for three-mode partitioning algorithms is one in which a randomly chosen (small) fraction of the cluster assignments, as obtained by applying a rational procedure (i.e., one of the procedures discussed above), is changed. As the fraction is chosen by some random process, one may consider this type of start as being situated somewhere in between rational and random starting procedures. More in particular, one may use the following procedure

to obtain a perturbed rational start: given a rational start $\mathbf{A}^R$, $\mathbf{B}^R$ and $\mathbf{C}^R$ for each of the partition matrices, randomly choose a proportion of the objects (resp. attributes, sources), identify to which cluster these elements have been assigned (according to the original rational start), and reassign them at random to one of the other clusters for that same mode. If necessary this procedure may be repeated (again starting from the original rational start) until a solution without empty clusters in any of the matrices $\mathbf{A}^R$, $\mathbf{B}^R$ and $\mathbf{C}^R$ is obtained.

3.1.3. Random start

In general, a random start can be characterized as a randomly drawn solution out of the total solution space for the model being estimated. For the three-mode partitioning case, this implies that, for each of the partition matrices separately, a random partition matrix is drawn from the set of proper partition matrices. More in particular, one may proceed in the following way: randomly assign each object (resp. attribute, source) to a cluster p (resp. q, r), with the probability of being assigned to a cluster p (resp. q, r) equal for all $p = 1, \ldots, P$ (resp. $q = 1, \ldots, Q$, $r = 1, \ldots, R$). If necessary this procedure may be repeated until a solution without empty clusters in any of the partition matrices is obtained.

3.2. ALS-scheme

Assuming initial estimates for the partition matrices, a further choice has to be made concerning the update mechanism of the ALS algorithm. We will discuss two approaches, the distinction between which relates to the perspective on the loss function. In the first approach loss function (5) is considered (which includes $\underline{\mathbf{W}}$ as an argument); the total set of parameters is therefore partitioned into four different subsets. The second approach is based on loss function (7) and implies a partition into three parameter sets (**A**,**B**,**C**). Below these two approaches will be discussed more in detail.

With regard to the approach based on loss function (5), two questions are to be dealt with. A first question pertains to when and how often $\underline{\mathbf{W}}$ should be updated. More in particular, the core $\underline{\mathbf{W}}$ can be updated after the estimation of all three partition matrices as in the sequence **ABC**$\underline{\mathbf{W}}$**AB**..., or after the estimation of each individual partition matrix as in the sequence **A**$\underline{\mathbf{W}}$**B**$\underline{\mathbf{W}}$**C**$\underline{\mathbf{W}}$**A**$\underline{\mathbf{W}}$**B**$\underline{\mathbf{W}}$.... The second question concerns the procedure for updating the partition matrices **A**, **B** and **C**. Consider, without loss of generality, the estimation of **A**. An analysis of the loss function in this case shows that it satisfies a separability property (Chaturvedi and Carroll, 1994). This separability means that the contribution of the cluster pattern of object i ($a_{i.}$) to the loss function can be separated from the contribution of the cluster patterns of the other objects. For the conditional estimation of **A** (keeping **B**, **C**, and $\underline{\mathbf{W}}$ fixed), this property is implied by the following decomposition of the loss function (5) in I separate terms:

$$\begin{aligned}
&\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K}\left(d_{ijk}-\sum_{p=1}^{P}a_{ip}\sum_{q=1}^{Q}\sum_{r=1}^{R}b_{jq}c_{kr}w_{pqr}\right)^2\\
&\quad=\sum_{j=1}^{J}\sum_{k=1}^{K}\left(d_{1jk}-\sum_{p=1}^{P}a_{1p}\sum_{q=1}^{Q}\sum_{r=1}^{R}b_{jq}c_{kr}w_{pqr}\right)^2\\
&\qquad+\cdots+\sum_{j=1}^{J}\sum_{k=1}^{K}\left(d_{Ijk}-\sum_{p=1}^{P}a_{Ip}\sum_{q=1}^{Q}\sum_{r=1}^{R}b_{jq}c_{kr}w_{pqr}\right)^2. \qquad (8)
\end{aligned}$$

Because of the separability the estimation can proceed by independently optimizing the pattern $a_{i.}$ $(i = 1 \ldots I)$ in each of the I terms of (8). It is straightforward to see that a similar separability property holds for the conditional estimations of **B** and **C**. In order to obtain conditional estimates for any of the partition matrices, one can for example reassign each object (resp. attribute, source) to another cluster and accept this change if the loss is improved. A conditionally optimal procedure is obtained by evaluating the assignment of each object (resp. attribute, source) to every possible cluster and accepting the cluster for which it holds that its corresponding i'th term in (8) is minimal.

With regard to the second approach, based on loss function (7), only three parameter subsets (**A**,**B**,**C**) are distinguished. This approach may therefore be represented by the sequence **ABCAB**... In this case, when a conditional estimate of **A** is desired (holding **B** and **C** fixed), the loss function does not satisfy separability with respect to each of the individual rows of **A**. This means that the conditional estimate of **A** now has to be obtained as a whole. As in the one-mode k-means

situation, finding such an optimal partition is only feasible when the number of elements to be clustered is very small. In other cases, heuristic procedures must be applied. Within these heuristic procedures, one possibility is to reassign every object (resp. attribute, source) to the best cluster, starting from the first object and proceeding in a one by one fashion to the last object. This procedure may then be repeated until no more improvement in the loss function (7) can be accomplished. Alternatively, one may in each step reassign the object for which reassignment yields the highest improvement in the loss function and continue doing this until no more improvement is found.

3.3. Empty clusters

Empty clusters imply models that are most likely locally optimal since at least one other solution exists, without empty clusters, that approximates the data at least as good. For the ALS-schemes based on loss function (5), the occurrence of empty clusters is a fatal problem because in this case the part of $\underline{\mathbf{W}}$ that corresponds to these empty clusters is undefined and will stay so during all subsequent iterations, the final estimate necessarily being a solution with empty clusters as well. For the ALS-scheme based on loss function (7), the occurrence of empty clusters is not as detrimental because in subsequent iterations, the ALS-procedure may still jump to a model without empty clusters. Moreover, as the standard ALS-schemes based on loss function (7) all include a row-wise reallocation of elements, the empty cluster problem can simply be avoided by not reallocating elements that already constitute a singleton cluster. It is easy to verify that the loss function (7) in such cases is not affected by this restriction.

In the case of the ALS-schemes based on loss function (5), we may distinguish between two classes of strategies to remove empty clusters. First, one may alter the cluster assignments in the partition matrix at hand. Second, one may also start by altering the cluster centroids $\underline{\mathbf{W}}$. Although many possibilities may be considered within each class of strategies, we will only discuss two procedures in which the cluster assignment is altered (*singleton* and *splitting*) and one procedure in which $\underline{\mathbf{W}}$ is altered (*mirror*).

A first way to adjust the cluster assignments is to reallocate the worst fitting element in the mode with an empty cluster to a singleton cluster. If more than one empty cluster is present, the element with the second worst fit is also reallocated to a singleton cluster, and so on, until no further empty clusters are present. We will further refer to this procedure as the *singleton* procedure.

A second way to adjust the cluster assignments is to split the most heterogeneous cluster into two or more smaller clusters. In this regard, Rocci and Vichi (2003) proposed to use a one-mode k-means clustering with $k=2$ to the reduced data array containing only the elements of the most heterogeneous cluster. If there is more than one empty cluster, the procedure may be repeated until no more empty clusters are present. We will further refer to this procedure as the *splitting* procedure.

A second class of strategies consists of altering the values in the core that correspond to the empty clusters. The idea behind this alteration follows from the observation that these particular elements of the core $\underline{\mathbf{W}}$ do not contribute to the loss function. As a result, they can be replaced by any other value without affecting the loss function. In this regard, Kiers (2004) suggested to multiply the core elements in question by -1, whereafter the partition matrix $\mathbf{A}$ is reestimated using the adjusted core. If this procedure again yields empty clusters in $\mathbf{A}$, Kiers further suggested to reset both $\mathbf{A}$ and the core to their previous values, implying that no update of $\mathbf{A}$ takes place. We will further refer to this procedure as the *mirror* procedure.

4. Simulation study

4.1. Problem and design

In this section, we present a simulation study to examine the performance of several ALS-algorithms. With this simulation study we will try to answer two specific groups of questions:

1. Goodness of fit: What is the performance, in terms of minimizing the loss function, of the ALS-algorithms under study? What is the effect of the size of the data set, the underlying true rank, the error level and/or the number of clusters?
2. Goodness of recovery: To which extent does, for each given data set, the optimal solution (across all ALS-algorithms) resemble the true structure underlying the data?

Table 1
Seven types of starting procedures situated within three different categories

Category	Starting procedure
Rational	(1) Independent *k*-means (2) Sequential *k*-means (3) Tucker3
Perturbed rational	(4) Perturbed independent *k*-means (5) Perturbed sequential *k*-means (6) Perturbed Tucker3
Random	(7) Random

Below, we successively discuss the design of our simulation study on the level of the algorithms (4.1.1) and on the level of the data (4.1.2).

4.1.1. Algorithms

Forty-nine different ALS-algorithms, composed using the elements of the framework presented in Section 3, are used in this simulation study. These algorithms were obtained by orthogonally combining seven types of initializations with seven combinations of an ALS-scheme and a procedure to deal with empty clusters.

With regard to initialization we used seven types of starting procedures, listed in Table 1, from the three categories described in Section 3.1 (rational, perturbed rational, and random). *Independent k-means*, *sequential k-means* and *Tucker3* were used to obtain rational starts. For the first two methods, which rely on one-mode *k*-means, a choice has to be made regarding all three aspects ((a) initialization, (b) updating scheme and (c) procedure to deal with empty clusters) covered in this paper. In our implementation, (a) 10 random multistart one-mode *k*-means analyses were run, retaining the best solution as the rational initial start for $\mathbf{A}^R$ (resp. $\mathbf{B}^R$, $\mathbf{C}^R$) for the three-mode partitioning algorithm, (b) the ALS-scheme within both one-mode *k*-means procedures was based on a loss function with the unknown partition matrix as the only argument: starting with the first element, each element was (re)allocated to the closest cluster centroid, the centroids being updated continuously; if a pass through all elements did not improve the loss, the procedure stopped, in the other case it was repeated, (c) elements already constituting a singleton cluster were never reallocated. In order to obtain the rational *Tucker3* start, the rationally started TUCKALS3 algorithm, as described by Kroonenberg and De Leeuw (1980), was applied. Three perturbed rational starts were generated by randomly selecting 20% of the rows of the partition matrices as provided by each of the three rational initialization procedures and by altering their cluster assignments through randomly reallocating them to one of the remaining clusters. Note that the value of 20% was chosen on the basis of a pretesting phase. Finally, a random start as described in Section 3.1 was used. Note that in the case of a rational start, only a single start is used; in all other cases (i.e., perturbed rational starts and random start) 50 multiple starts were obtained by repeating the corresponding generating procedure.

We further chose seven combinations of an ALS-scheme and an empty clusters procedure, making use of the two categories described in Section 3.2 and the two categories described in Section 3.3. These seven combinations are depicted in Fig. 3. More in particular, we proceeded in the following way:

With regard to the ALS schemes, we selected two procedures based on loss function (5) and one based on loss function (7). The ALS-schemes based on loss function (5) were **ABCWAB**... and **AWBWCWAWBW**... respectively; conditionally optimal estimates for the partition matrices were obtained as described in Section 3.2; the ALS procedures further stopped when no further decrease in the loss function (5) is observed after a successive estimation of all three partition matrices. With regard to the ALS-scheme based on loss function (7), conditional estimates of **A**, **B** and **C** were obtained as follows: starting with the first element of a mode, each element was (re)allocated to the cluster for which it holds that (7) is minimal. If a pass through all elements of a mode did not improve the loss, the procedure moved on to the estimation of the other partition matrices; in the other case the procedure was repeated. The procedure stopped when no further decrease in the loss function (7) was observed after a successive estimation of all three partition matrices.

With regard to the procedures for dealing with empty clusters, we note that in case of the ALS-scheme based on loss function (7) no additional empty cluster procedure is required. For the ALS-schemes based on loss function (5) however, we used the three procedures discussed in Section 3.3 for dealing with empty clusters (*singleton*, *splitting* and

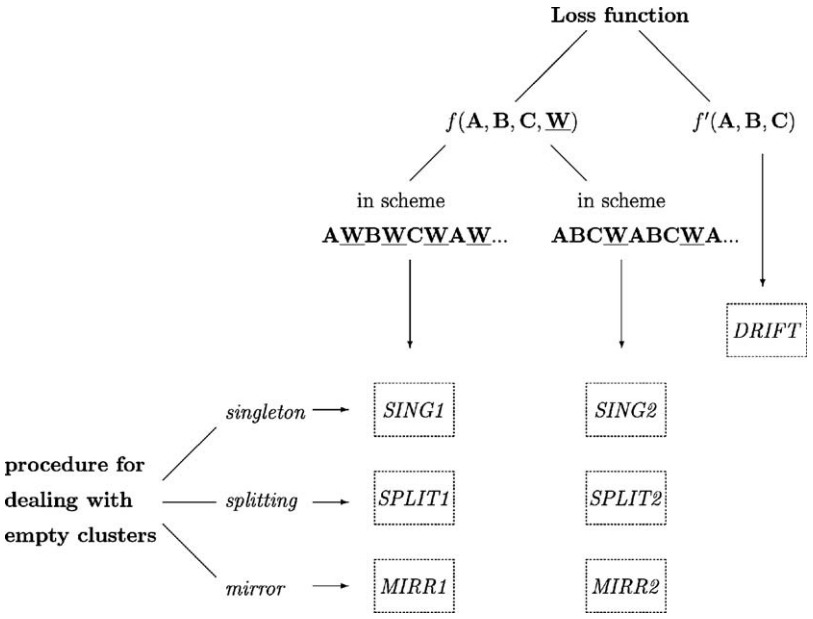

Fig. 3. Graphical representation of seven combinations of an *ALS-scheme* and a *Procedure to deal with empty clusters* (in the dashed boxes) as used in our first simulation study, situated within the ALS framework.

mirror). For the splitting of the most heterogeneous cluster into two clusters (as included in the *splitting* procedure), a one-mode k-means with $k = 2$ was applied with one single start (i.e., an assignment of the first half of the elements to the first cluster and the second half to the other one) and with the same ALS updating scheme as the one we used to obtain the one-mode k-means rational starts. Fig. 3 illustrates how the three ALS-schemes and the three types of empty clusters procedures are combined to yield seven *ALS-scheme/empty clusters procedure* combinations, denoted by *MIRR1*, *MIRR2*, *SPLIT1*, *SPLIT2*, *SING1*, *SING2* and *DRIFT*. The combinations *MIRR1*, *SPLIT2* and *DRIFT* have already been proposed by Kiers (2004), Rocci and Vichi (2003) and Schepers and Van Mechelen (2004), respectively. The remaining ones are novel. Note that all algorithms described in this paper have been implemented in MATLAB and are freely available from the first author.

4.1.2. Data

To explain the design of the data generation, three different types of real-valued $I \times J \times K$ arrays must be distinguished: a true array $\underline{\mathbf{T}}$, which can be represented by a three-mode partitioning model of a certain rank; a data array $\underline{\mathbf{D}}$, which is $\underline{\mathbf{T}}$ perturbed with error; and the model array $\underline{\mathbf{M}}$ yielded by an ALS-algorithm, which can be represented by a three-mode partitioning model of the same rank as the true array $\underline{\mathbf{T}}$.

Four design factors were fully crossed on the level of the data generation (arrays $\underline{\mathbf{T}}$ and $\underline{\mathbf{D}}$):

(1) the *size*, $I \times J \times K$, of $\underline{\mathbf{T}}$, $\underline{\mathbf{D}}$, and $\underline{\mathbf{M}}$, at three levels: $48 \times 48 \times 48$, $96 \times 48 \times 48$, $96 \times 96 \times 48$;
(2) the *rank* of the three-mode partitioning model for $\underline{\mathbf{T}}$, at five levels: $2 \times 2 \times 2$, $2 \times 2 \times 4$, $4 \times 2 \times 2$, $2 \times 4 \times 4$, $4 \times 4 \times 4$;
(3) the *error*, ε, which is the expected proportion of variance in $\underline{\mathbf{D}}$ due to error, at five levels: 0.00, 0.01, 0.05, 0.20, 0.60;
(4) the *equality of cluster sizes*, the number of modes that contain one cluster with a cardinality that is five times smaller than the cardinalities of the other clusters pertaining to that same mode. At level zero, for each mode all clusters corresponding to that same mode contain the same number of elements, whereas at level one (resp. two, three), for one (resp. two, three) mode(s) the cardinality of one cluster of that mode is five times smaller than the cardinality of the other clusters for that mode while for the remaining modes all clusters are of equal size.

For each combination of these four independent variables, 10 replicates were simulated, yielding 10 × 3 (*Size*) × 5 (*Rank*) × 5 (*Error*) × 4 (*Equality of Cluster Sizes*) = 3000 simulated data sets. For each combination of the levels of *Size*, *Rank* and *Equality of Cluster Sizes*, ten sets of partition matrices **A**, **B** and **C** were randomly drawn. For each set a true array $\underline{\mathbf{T}}$ was then obtained by (1) generating entries of a core $\underline{\mathbf{W}}$ as independent realizations of a uniformly distributed variable in [0, 1], (2) combining **A**, **B**, **C** and $\underline{\mathbf{W}}$ by expression (1). Next, for each true array $\underline{\mathbf{T}}$ a corresponding data array $\underline{\mathbf{D}}$ was obtained by adding error using the following expression:

$$d_{ijk} = \sum_{p=1}^{P}\sum_{q=1}^{Q}\sum_{r=1}^{R} a_{ip}b_{jq}c_{kr}w_{pqr} + e_{ijk} \quad \forall i, j, k, \tag{9}$$

where e_{ijk} was sampled from $N(0, \sigma_e^2)$ with $\sigma_e^2 = (\varepsilon/(1-\varepsilon))\sigma_{\underline{\mathbf{T}}}^2$, and, hence, $\varepsilon = \sigma_e^2/\sigma_{\underline{\mathbf{D}}}^2$ ($\sigma_{\underline{\mathbf{T}}}^2$ and $\sigma_{\underline{\mathbf{D}}}^2$ denoting the variances across all array entries as well as across all possible replications of the error process for $\underline{\mathbf{T}}$ and $\underline{\mathbf{D}}$, respectively).

4.2. Results

4.2.1. Goodness of fit

In this section we report the results on the performance of the different ALS algorithms under study in minimizing the loss function both in general, and as a function of the manipulated data characteristics *Size*, *Rank*, *Error* and *Equality of cluster sizes*. For this purpose, we introduce the measures Badness of data (*BOD*) and Badness of fit (*BOF*):

$$BOD = \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K}\left(d_{ijk} - t_{ijk}\right)^2, \quad BOF = \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K}\left(d_{ijk} - m_{ijk}\right)^2,$$

where d_{ijk}, t_{ijk} and m_{ijk} denote the entries of the arrays $\underline{\mathbf{D}}$, $\underline{\mathbf{T}}$ and $\underline{\mathbf{M}}$, respectively. We will focus on the analyses in the true rank, because in that specific case, the value of *BOD* is an upper bound for the *BOF*-value of the global optimum.

First, we examined for how many data sets by at least one of the combinations of the algorithms a solution was found with a *BOF* less than or equal to *BOD*. It turns out that for all data sets, this was the case. As a consequence, for each given data set, we will further use the *BOF* of the best solution across all analyses of that data set as a proxy for the *BOF* of the corresponding globally optimal solution.

Second, we identified for each algorithm and each data set whether the *BOF* of the final estimate (which equals the final *BOF* resulting from the one single run in case of rational starts and the *BOF* of the best final solution in case of multistarts, i.e., the perturbed rational starts and the random starts) equals the proxy for the *BOF* of the global optimum. Table 2 shows the proportions of data sets for which this holds. It appears that the proxy of the global optimum is (almost) always found with a *Tucker3* rational start and with any multistart *SPLIT* or *DRIFT* procedure.

Third, considering only multistart procedures, we counted for each data set and each algorithm how many times the optimal solution was found within the 50 multistarts. The higher this number, the more reliable the algorithm is in minimizing the loss function. An analysis of variance was conducted with the frequency of optimally ending runs as dependent variable, with *Size*, *Rank*, *Error* and *Equality of Cluster Sizes* as between-subject independent variables, and with *Type of Start* and *ALS-scheme/Empty cluster procedure* as within-subject independent variables. Considering only effects with an effect size larger than 0.10 as sizeable, the analysis revealed sizeable effects of *ALS-scheme/Empty cluster procedure* $(\rho_I = 0.46)$, *Rank* $(\rho_I = 0.28)$ and interaction between *Rank* and *ALS-scheme/Empty cluster procedure* $(\rho_I = 0.11)$. More in particular, as shown in Fig. 4, the *SPLIT* procedures are most likely to return the optimal solution within one single run, and their performance is in this respect near perfect; also, the optimal solution is less likely to be found within one single run when *Rank* increases, and this is more pronounced for the *MIRR* procedures, whereas the *SPLIT* procedures are far less sensitive to an increase in *Rank*.

4.2.2. Goodness of recovery

In this section we examine the extent to which the optimal partitions as obtained from the data analyses resemble the true partitions underlying the data. In particular, we measured the agreement between the true underlying partition of the set of objects (resp. attributes, sources) and the estimated corresponding partition for the best solution obtained across all algorithms (i.e., the solution with a *BOF*-value equal to the proxy of the global optimum), making use of

Table 2
Proportion of data sets for which it holds that a particular combination of *ALS-scheme/empty cluster procedure* and *type of start* returns a solution with a *BOF* value equal to the proxy of the global optimum

Type of start	$f(\mathbf{A}, \mathbf{B}, \mathbf{C}, \underline{\mathbf{W}})$						$f(\mathbf{A}, \mathbf{B}, \mathbf{C})$
	Mirror		Splitting		Singleton		
	MIRR1	MIRR2	SPLIT1	SPLIT2	SING1	SING2	DRIFT
Rational							
Independent *k*-means	0.22	0.25	0.76	0.73	0.55	0.53	0.63
Sequential *k*-means	0.69	0.71	0.75	0.76	0.75	0.74	0.71
Tucker3	0.99	1.00	1.00	1.00	1.00	1.00	0.98
Perturbed rational							
Perturbed independent *k*-means	0.91	0.88	1.00	1.00	0.94	0.95	0.99
Perturbed sequential *k*-means	0.92	0.89	1.00	1.00	0.96	0.98	1.00
Perturbed Tucker3	0.94	0.95	1.00	1.00	0.98	0.97	1.00
Random							
Random	0.99	0.98	1.00	1.00	0.99	0.99	1.00

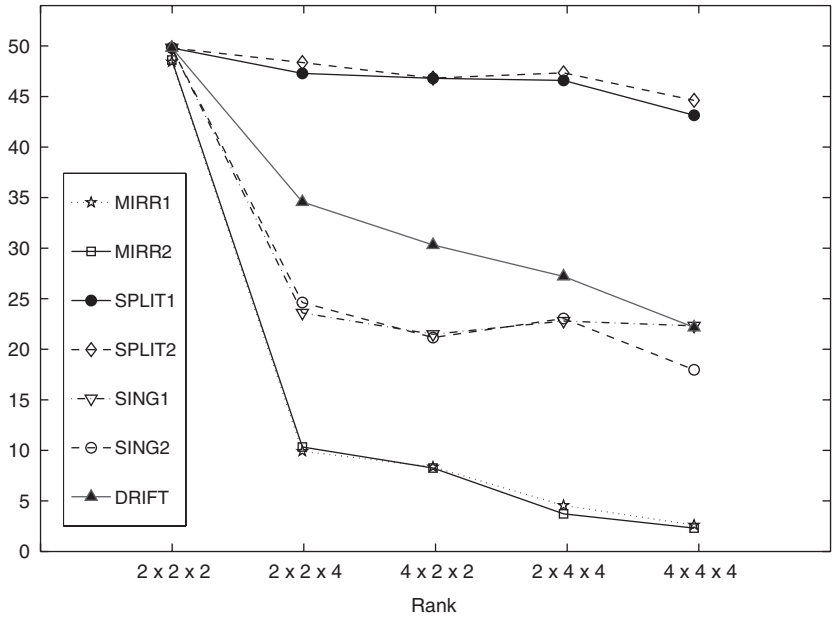

Fig. 4. Line plots of the mean number of times (*Y*-axis) the optimal solution is found within 50 runs, for each combination of an *ALS-scheme* and a *Procedure for dealing with empty clusters* and for each level of *Rank*.

the corrected Rand index (Hubert and Arabie, 1985). This index equals 1 if the two partitions are identical and 0 if the two partitions do not correspond more than expected by chance. A combined corrected Rand index (c-CRI) was calculated by taking the average corrected Rand index for the object, the attribute and the source partitions, weighted by the number of objects, attributes and sources, respectively.

For all 3000 data sets, the c-CRI between the true partitions and the partitions corresponding to the best solution obtained across all algorithms was equal to 1, implying that the true and the estimated partitions are identical in every case. Note that this implies that even when 60% of the variance in the data is due to error a perfect recovery of the underlying partitions was found.

Table 3
Proportion of ranks for which it holds that a particular combination of *ALS-scheme/empty cluster procedure* and *type of start* returns a solution with a *BOF* value equal to the proxy of the global optimum for the anger-consequence data

Type of start	f(**A**, **B**, **C**, **W**)						f(**A**, **B**, **C**)
	Mirror		Splitting		Singleton		
	MIRR1	MIRR2	SPLIT1	SPLIT2	SING1	SING2	DRIFT
Rational							
Independent *k*-means	0.05	0.05	0.05	0.05	0.05	0.05	0.06
Sequential *k*-means	0.05	0.05	0.05	0.05	0.05	0.05	0.05
Tucker3	0.06	0.02	0.05	0.02	0.05	0.02	0.06
Perturbed rational							
Perturbed independent *k*-means	0.06	0.08	0.06	0.08	0.09	0.09	0.69
Perturbed sequential *k*-means	0.11	0.11	0.11	0.11	0.13	0.11	0.64
Perturbed Tucker3	0.11	0.11	0.09	0.11	0.09	0.11	0.64
Random							
Random	0.11	0.09	0.11	0.09	0.11	0.09	0.75

5. Four empirical data sets

5.1. Performance of different algorithms

In this section we report the results of analyses of four empirical three-way three-mode data sets for each of the combinations of an *ALS-scheme/Empty cluster procedure* and a *Type of Start*. The data sets were the following ones:

1. Anger-consequence data (Van Coillie and Van Mechelen, 2006): a set of 139 respondents indicated for a set of eight consequences to what extent they expected them to change after they would have executed each behavior out of a set of 16 anger-related behaviors.
2. Anger data (Kuppens and Van Mechelen, in press): respondents indicated to what degree they experienced a list of 24 anger related responses in 14 different situations. Leaving out all subjects for which missing values were observed, 357 respondents remained.
3. Chopin data (Murakami and Kroonenberg, 2003): 38 Japanese university students rated 24 short piano solo pieces composed by Frederik Chopin on a set of 20 bipolar scales.
4. Archetypal patients (Mezzich and Solomon, 1980): each of 22 psychiatrists was invited to think of a typical patient for each of four diagnostic categories and to characterize each patient in terms of severity on 17 psychiatric symptoms.

All data sets were analyzed with as *Rank* all possible combinations of 2–5 clusters for each mode (except for the patient mode in the archetypal patient data set for which the highest number of clusters in the analyses was limited to 3, because of the small number of elements in this mode). For each rank, every combination of *ALS-scheme/Empty cluster procedure* and *Type of Start* was run, retaining only the best solution out of 50 in the case of multistarts. The *BOF* of the best overall solution found for every level of *Rank* was taken as a proxy for the global optimum for that data set and level of *Rank*. Proportions of ranks in which the globally optimal solution was found by each algorithm are reported in Tables 3–6. From these tables, the multistart *DRIFT* procedure clearly has superior performance over the others (Note that, in the earlier study it was already among the best, but others were similar).

Second, considering only those algorithmic combinations with multistarts, we counted how many times the optimal solution was found within these 50 multistarts. An analysis of variance performed per data set, with the frequency of optimally ended runs as dependent variable, the levels of *Rank* as blocks, and *Type of Start* and *ALS-scheme/Empty cluster procedure* as fully crossed independent variables, only shows for each data set sizeable effects of *ALS-scheme/Empty*

Table 4
Proportion of ranks for which it holds that a particular combination of *ALS-scheme/empty cluster procedure* and *type of start* returns a solution with a *BOF* value equal to the proxy of the global optimum for the Anger data

Type of start	$f(\mathbf{A}, \mathbf{B}, \mathbf{C}, \underline{\mathbf{W}})$						$f\prime(\mathbf{A}, \mathbf{B}, \mathbf{C})$
	Mirror		Splitting		Singleton		
	MIRR1	MIRR2	SPLIT1	SPLIT2	SING1	SING2	DRIFT
Rational							
Independent *k*-means	0.00	0.00	0.00	0.00	0.00	0.00	0.09
Sequential *k*-means	0.00	0.00	0.00	0.00	0.00	0.00	0.14
Tucker3	0.00	0.00	0.00	0.00	0.00	0.00	0.08
Perturbed rational							
Perturbed independent *k*-means	0.16	0.14	0.20	0.19	0.20	0.17	0.80
Perturbed sequential *k*-means	0.22	0.23	0.22	0.23	0.22	0.23	0.83
Perturbed Tucker3	0.25	0.23	0.25	0.23	0.27	0.23	0.81
Random							
Random	0.14	0.06	0.17	0.09	0.17	0.08	0.83

Table 5
Proportion of ranks for which it holds that a particular combination of *ALS-scheme/empty cluster procedure* and *type of start* returns a solution with a *BOF* value equal to the proxy of the global optimum for the Chopin data

Type of start	$f(\mathbf{A}, \mathbf{B}, \mathbf{C}, \underline{\mathbf{W}})$						$f\prime(\mathbf{A}, \mathbf{B}, \mathbf{C})$
	Mirror		Splitting		Singleton		
	MIRR1	MIRR2	SPLIT1	SPLIT2	SING1	SING2	DRIFT
Rational							
Independent *k*-means	0.00	0.00	0.00	0.00	0.00	0.00	0.06
Sequential *k*-means	0.00	0.00	0.00	0.00	0.00	0.00	0.06
Tucker3	0.00	0.00	0.00	0.00	0.00	0.00	0.13
Perturbed rational							
Perturbed independent *k*-means	0.13	0.17	0.14	0.14	0.13	0.16	0.81
Perturbed sequential *k*-means	0.19	0.13	0.19	0.13	0.20	0.13	0.97
Perturbed Tucker3	0.13	0.16	0.11	0.16	0.11	0.16	0.94
Random							
Random	0.13	0.11	0.13	0.11	0.14	0.14	0.86

cluster procedure (i.e., with ρ_I equalling 0.26, 0.19, 0.31 and 0.63 for the Anger-Consequence, Anger, Chopin and Archetypal patients data sets, respectively; *Type of Start* and the interaction between *Start* and *ALS-scheme/Empty cluster procedure* always accounted for less than 1% of the total variances). In order to capture the meaning of this effect, one may look at Fig. 5 which contains boxplots of the number of times the best solution was found by each *ALS-scheme/Empty cluster procedure* combination, aggregated over all levels of *Rank* and *Type of Start*, for each data set. From this figure, it again appears that *DRIFT* outperforms all other procedures.

Taken as a whole, the results of the simulation study that indicated a superior performance for *Tucker3* rational starts and for the *SPLIT* procedures are not in line with the analyses of all four empirical data sets. In the latter case the multistart *DRIFT* algorithms performed more or less as to be expected from the simulation study, whereas

Table 6
Proportion of ranks for which it holds that a particular combination of *ALS-scheme/empty cluster procedure* and *type of start* returns a solution with a *BOF* value equal to the proxy of the global optimum for the Archetypal patients data

Type of start	$f(\mathbf{A}, \mathbf{B}, \mathbf{C}, \underline{\mathbf{W}})$						$f(\mathbf{A}, \mathbf{B}, \mathbf{C})$
	Mirror		Splitting		Singleton		
	MIRR1	MIRR2	SPLIT1	SPLIT2	SING1	SING2	DRIFT
Rational							
Independent *k*-means	0.00	0.00	0.00	0.00	0.00	0.00	0.06
Sequential *k*-means	0.00	0.00	0.00	0.00	0.00	0.00	0.06
Tucker3	0.00	0.00	0.00	0.00	0.00	0.00	0.16
Perturbed rational							
Perturbed independent *k*-means	0.19	0.19	0.25	0.19	0.28	0.22	1.00
Perturbed sequential *k*-means	0.31	0.16	0.31	0.16	0.31	0.16	0.97
Perturbed Tucker3	0.22	0.13	0.22	0.13	0.19	0.13	0.97
Random							
Random	0.25	0.22	0.25	0.19	0.22	0.19	0.97

all other algorithms performed much worse. In the next subsection, we will try to identify possible reasons for this finding.

5.2. Parametric bootstrapping

In this section we will examine in what way(s) the empirical data sets differ from the data sets generated in the simulation study. This may indicate possible causes of the reversal of the ordering of the combinations of ALS-schemes and empty clusters procedures in terms of performance in finding the optimal solution, when the analyses are performed on simulated as compared to empirical data.

In the simulation study, the data were generated in accordance to the stochastic model formulation in (4) that also implicitly underlies the least-squares loss function minimized by the algorithms. This implies that these data sets meet the assumptions of independently and identically normally distributed residuals. One may wonder whether these assumptions also hold for all of our empirical data sets. In the following, we will investigate this, more in particular by examining (i) whether the within-data cluster variances (i.e., the variances of the data values across all triplets of an object, an attribute and a source that belong to the data cluster in question) are identical, (ii) whether the residuals are normally distributed, and (iii) whether the independence assumption holds. For this purpose we will make use of a parametric bootstrap procedure. For matters of simplicity, we will only focus on one of the data sets of the previous section, namely the Anger-Consequence data (Van Coillie and Van Mechelen, 2006), and on only one value of *Rank*, namely $2 \times 3 \times 4$. Note that the results of the analyses discussed below are the same when other empirical data sets and other levels of *Rank* are considered. For the parametric bootstrap test, we made use of the optimal $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\underline{\mathbf{W}}$, as obtained from the whole of our analyses in *Rank* $2 \times 3 \times 4$; furthermore, we calculated the pooled within-cluster residual variance $\hat{\sigma}_\varepsilon^{\,2}$ as an estimate of σ_ε^2. Next, we drew 1000 bootstrap replicated data sets by generating residuals independently from $N\left(0, \hat{\sigma}_\varepsilon^{\,2}\right)$.

In order to test the above assumptions we computed, for both the Anger-Consequence data set and for all bootstrap replicates, (i) the ratio of the highest within-data cluster variance over the smallest within-data cluster variance, (ii) the value of the Kolmogorov–Smirnoff statistic of the residuals with respect to deviations from the normal distribution with the same mean and variance, and (iii) the sum of the squared covariances between the residuals for all pairs of elements from the Cartesian product of attributes and sources across all objects. Fig. 6 shows the values of the three statistics for the bootstrap data sets (in the form of boxplots), and for the empirical data set (in the form of asterisks).

Clearly, the values of all three statistics calculated for the observed Anger-Consequence data are highly unlikely under the null model. We therefore must conclude that for the Anger-Consequence data the within-cluster variances are

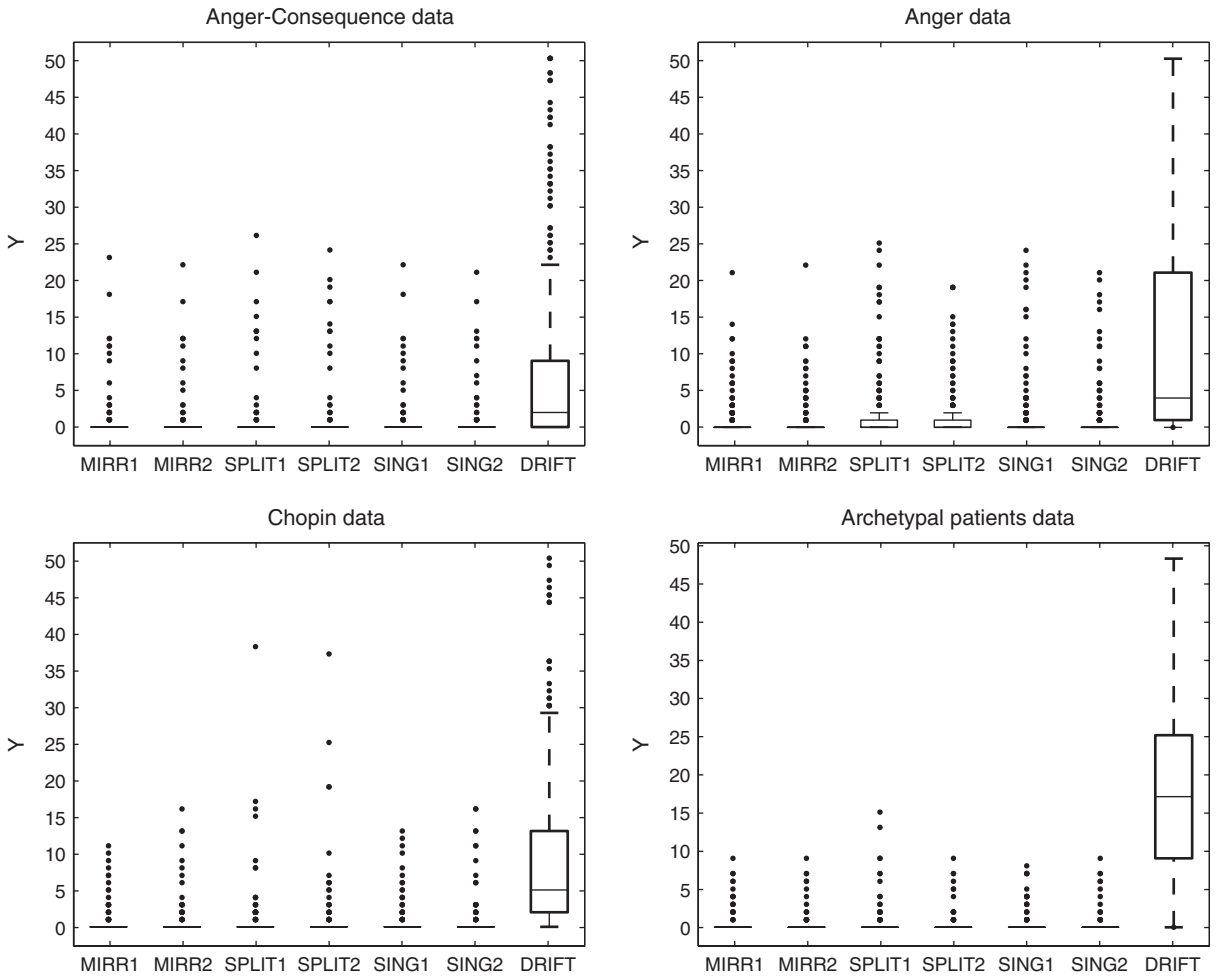

Fig. 5. Boxplots of the number of times (*Y*) the best solution was found, aggregated over all combinations of ranks, by each of the different *ALS-scheme/Procedure for dealing with empty clusters* combinations, for each empirical data set.

not equal, that the residuals are not normally distributed, and that there is a dependency between the residuals within a row slice of the data. This raises the suspicion that one of these violations, or a combination of several of them, may affect algorithmic performance. In order to get more insight into this, a second simulation study was set up which is discussed in the next section.

6. Second simulation study

In order to test whether violations of the implicit stochastic model as observed in the previous section could affect algorithmic performance, we simulated two new groups of data sets, each of them violating in some way the assumptions of model (4). All data sets had $Size = 20 \times 20 \times 20$, $Rank = 4 \times 4 \times 4$, and $Error = 0.3$.

In the first part of the simulation study we dealt with the assumptions of equal variance and normality. For this purpose we orthogonally combined two design factors in the data generation: *Error distribution* (7 levels) and *Within-cluster variance inequality* (3 levels). Table 7 shows the seven error distributions, and the values of their respective parameters. The distributions in question all were centered and rescaled in line with the chosen error level in the data.

Three different levels of *within-cluster variance inequality* were obtained by making the within-cluster variance of 0, 16 or 32 out of the $4 \times 4 \times 4 = 64$ data clusters 9 times as large as the within-cluster variance of the remaining data clusters. For each combination of *Error distribution* and *Within-cluster variance inequality*, 10 data sets were

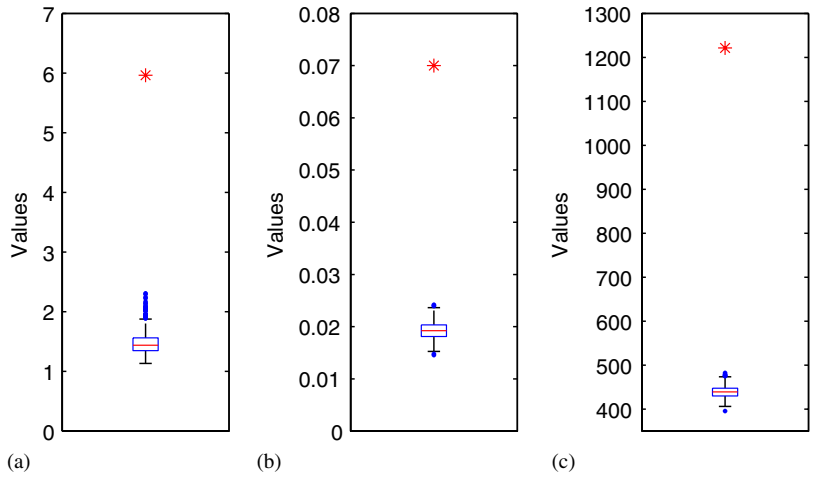

Fig. 6. Boxplots for the values of (a) the ratio of the largest over the smallest within-cluster variance, (b) the Kolmogorov–Smirnoff statistic of the residuals and (c) the sum of squared residual covariances, for 1000 parametric bootstrap replicates of the Anger-consequence data. The asterisks indicate the corresponding value of the test statistic for the real data.

Table 7
Error distributions used in the second simulation study

Distribution	Parameters	Form
t	$\nu = 5$	Symmetric, thicker tails than normal distribution
Beta	$\alpha = 2, \beta = 3$	Skewed to the right
Lognormal	$\mu = 0.5, \sigma = 0.3$	Skewed to the right
Poisson	$\lambda = 2$	Skewed to the right, non-continuous
Uniform	$\alpha = -0.25, \beta = 0.25$	Symmetric
Weibull	$\beta = 1, \gamma = 2$	Skewed to the right
χ^2	$\nu = 4$	Skewed to the right

generated. Each resulting data set was then analyzed with 50 random multistarts and a single rational *Tucker3* start by each of the seven *ALS-scheme/Empty cluster procedure* combinations. Regarding the multistarts, we only used random starts in order to reduce computational load and because, when considering only the multistart procedures, we did not find, in Simulation study 1, an effect of *Type of Start*. We also only used a single rational start, *Tucker3*, because for this type of start the difference in performance in minimizing the loss function was most salient when comparing the results of the original simulation study (4.2.1) and those of the analyses of the empirical data sets (5.1).

First, we identified in the same way as discussed in Section 4.2.1 the proxy for the *BOF* of the global optimum for each data set. Then, for each algorithm and each data set, we (a) investigated whether the *BOF* of the final estimate equals the proxy for the global optimum, and (b) considering only the random multistart procedure, calculated the frequency of optimally ending runs. In all conditions, the same pattern of performance as in the first simulation study was found, and we will therefore only give a brief summary of the results. With respect to (a), we found perfect performance of the *Tucker3* start and near perfect performance for the random multistart procedure; with respect to (b), the *SPLIT* combinations outperformed all other combinations, that is, these combinations showed the largest frequencies of optimally ending runs. These results suggest that the performance of the *Tucker3* start and the ordering of the *ALS-scheme/Empty cluster procedure* combinations in terms of minimizing the loss function are robust against violations of the implicit stochastic model pertaining to different within-cluster variances and deviations from normality.

In the second part of the simulation we dealt with the independence assumption. For this purpose, 100 data sets were generated with residuals being drawn from the normal distribution as in (9), but now with a nondiagonal covariance matrix. More in particular, a variance–covariance matrix for the attribute-source combinations was used with anti-Robinson form, with constant variances of 1, and with decreasing covariances with a constant step of $1/(J \times K) - 1$

Table 8
Proportion of data sets for which it holds that a particular combination of *ALS-scheme/empty cluster procedure* and two levels of *type of start* returns a solution with a *BOF* value equal to the proxy of the global optimum in the second simulation study (covariance condition)

Type of start	$f(\mathbf{A}, \mathbf{B}, \mathbf{C}, \underline{\mathbf{W}})$						$f(\mathbf{A}, \mathbf{B}, \mathbf{C})$
	Mirror		Splitting		Singleton		
	MIRR1	MIRR2	SPLIT1	SPLIT2	SING1	SING2	DRIFT
Tucker3	0.46	0.46	0.46	0.46	0.47	0.46	0.71
Random	0.55	0.54	0.70	0.70	0.64	0.65	0.93

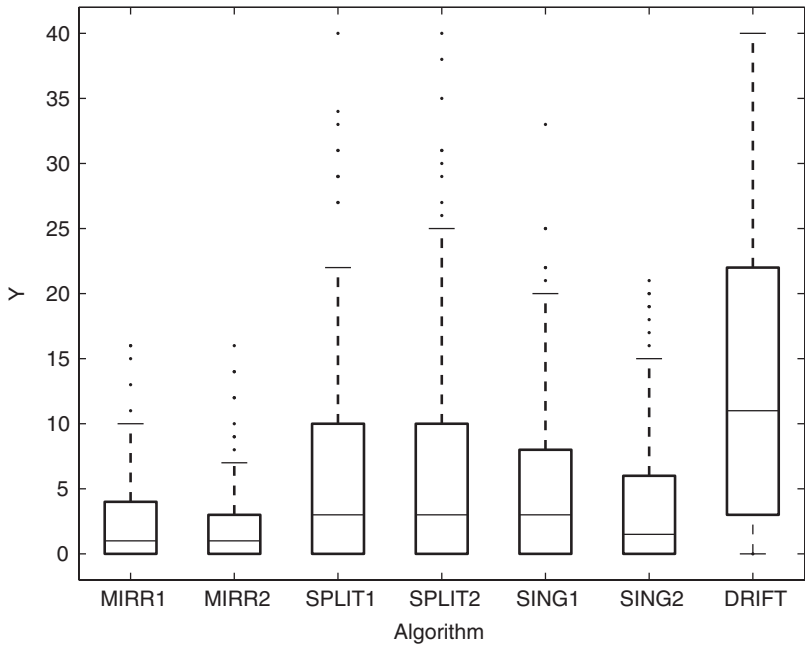

Fig. 7. Box plots of the number of times (*Y*) the best solution was found by each of the different *ALS-scheme/Procedure for dealing with empty clusters* combinations in the second simulation study (condition with dependencies between the residuals).

when moving away from the main diagonal (in any horizontal or vertical direction). After generation of the residuals, their values all were rescaled in line with the chosen error level in the data. Subsequently, we again identified, in the same way as discussed in Section 4.2.1, the proxy for the *BOF* of the global optimum for each data set. Then, for each algorithm and for each data set, we (a) investigated whether the *BOF* of the final estimate equals the proxy for the global optimum, and (b) considering only the random multistart procedure, calculated the frequency of optimally ending runs.

Regarding (a), Table 8 shows for all algorithms the proportions of data sets for which a solution with *BOF*-value equal to the proxy was returned. The *Tucker3* initialization appears to be less effective than the random multistart procedure and the *DRIFT* procedure now outperforms the *SPLIT* combinations. Regarding (b), an analysis of variance with the frequency of optimally ended runs as dependent variable, the individual data sets in this condition as blocks, and the combination *ALS-scheme/Empty cluster procedure* as independent variable shows a sizeable effect $(\rho_I = 0.17)$ of *ALS-scheme/Empty cluster procedure*. From Fig. 7, which displays the boxplots for the dependent measure for each *ALS-scheme/Empty cluster procedure*, it appears that *DRIFT* returns the optimal solution more frequently than the other *ALS-scheme/Empty cluster procedure* combinations. Particularly striking is also the large decrease in performance

(as compared to the results of the first simulation study) of the *SPLIT* procedures. All this lines up with the results of the analyses of the four empirical data sets.

7. Concluding remarks

This paper recapitulated the three-mode partitioning model, which yields a simultaneous partitioning of all three modes of a three-way three-mode data array. We further presented an overall framework for alternating least-squares algorithms for fitting the three-mode partitioning model to a data set.

In a first simulation study in which forty-nine algorithms representing different cells of the overall framework were evaluated according to their performance in minimizing the loss function, it was found that any procedure using the rational *Tucker3* initialization, any multistart *SPLIT* algorithm and, to a somewhat lesser extent, any multistart *DRIFT* algorithm show superior performance. When applying the same algorithms to four empirical data sets, however, a different result was obtained as compared to the simulation study: the multistart *DRIFT* algorithms now outperformed all other algorithms. By means of parametric bootstrap tests, it was found that the real data sets differed in several respects from the artificially generated data sets of the simulation study, which were generated according to the stochastic model assumptions in (4). In a second simulation study in which data sets were generated under different modeling assumptions, we found that the discrepancy between the results of the first simulation study and the analyses of the real data sets could be attributed to dependencies between the residuals. In case of dependent residuals, indeed, similar to the analyses of the real data sets, the multistart *DRIFT* procedure outperformed all other procedures. In order to better understand this finding, we note that the conditional updating of the partition matrices in the ALS-scheme in *DRIFT* goes along with a continuous updating of $\underline{\mathbf{W}}$. As such, this approach may allow for a permanent interaction between the different elements of the same mode during the optimization process; in this way, covariances might be better taken into account; in the conditional updating step of the other ALS-schemes, however, every (re)estimation of a row of a partition matrix is independent from the estimation of the other rows of that same partition matrix.

The fact that nonindependent errors lead to problems for some of the algorithms could perhaps also be attributed to the fact that too few clusters were used. Indeed, in the nonindependence condition some structural information remains in the residuals, which might be captured by adding additional clusters to the model (although this is not too straightforward). In practical applications structure in the residuals can be expected to show up very often. Support for this may be found in the observation that the multistart *DRIFT* algorithms outperform all others for almost all levels of *Rank* when applied to the four empirical data sets (see Section 5.1).

Taken as a whole, the findings in this paper suggest that the best choice of an algorithm to estimate a three-mode partitioning could be a combination of a multistart with the procedure *DRIFT*. An important justification of this choice is that dependencies between residuals may often be expected in empirical practice. In addition to a multistart-*DRIFT*, one could finally also apply one of the *SPLIT* procedures combined with either a rational *Tucker3* start or a multistart. Although computationally more expensive such a multi-algorithmic approach may imply a better guarantee to yield a globally optimal solution for a broad scope of data sets.

Acknowledgments

The authors would like to thank the referees for their valuable comments and suggestions, which helped improve the original manuscript. They also would like to thank Henk Kiers, Maurizio Vichi and Roberto Rocci for kindly making available their programs. J. Schepers and I. van Mechelen were supported by the Fund for Scientific Research-Flanders (Belgium), Project No. G.0146.06 awarded to Iven van Mechelen and by the Research Council KULeuven (GOA/2005/04). E. Ceulemans is a post-doctoral fellow of the Fund for Scientific Research-Flanders (Belgium).

References

Al-Sultan, K.S., Maroof Khan, M., 1996. Computational experience on four algorithms for the hard clustering problem. Pattern Recognition Lett. 17, 295–308.

Baier, D., Gaul, W., Schader, M., 1997. Two-mode overlapping clustering with applications to simultaneous benefit segmentation and market structuring. In: Klar, R., Opitz, O. (Eds.), Classification and Knowledge Organization. Springer, Berlin, Germany, pp. 557–566.

Banfield, J.D., Raftery, A.E., 1993. Model-based Gaussian and non-Gaussian clustering. Biometrics 49, 803–821.

Bock, H.H., 1996. Probabilistic models in cluster analysis. Comput. Statist. Data Anal. 23, 5–28.

Bryant, P., Williamson, J.A., 1978. Asymptotic behaviour of classification maximum likelihood estimates. Biometrika 65, 273–281.
Carroll, J.D., Arabie, P., 1980. Multidimensional scaling. Annu. Rev. Psychol. 31, 607–649.
Celeux, G., Govaert, G., 1992. Comparison of the mixture and the classification maximum likelihood in cluster analysis. J. Statist. Comput. Simulation 14, 315–332.
Chaturvedi, A., Carroll, J.D., 1994. An alternating combinatorial optimization approach to fitting the INDCLUS and generalized INDCLUS models. J. Classification 11, 155–170.
Gaul, W., Schader, M., 1996. A new algorithm for two-mode clustering. In: Bock, H., Polasek, W. (Eds.), Classification and Knowledge Organization. Springer, Berlin, Germany, pp. 15–23.
Govaert, G., 1995. Simultaneous clustering of rows and columns. Control Cybernet. 24, 437–458.
Govaert, G., Nadif, M., 2003. Clustering with block mixture models. Pattern Recognition 36, 463–473.
Hand, D.J., Krzanowski, W.J., 2005. Optimising *k*-means clustering results with standard software packages. Comput. Statist. Data Anal. 49, 969–973.
Hubert, L., Arabie, P., 1985. Comparing partitions. J. Classification 2, 193–218.
Hubert, L., Arabie, P., Meulman, J., 2001. Combinatorial Data Analysis: Optimization by Dynamic Programming. SIAM, Philadelphia, PA.
John, S., 1970. On identifying the population of origin of each observation in a mixture of observations from two normal populations. Technometrics 12, 553–563.
Kiers, H.A.L., 2004. Clustering all three modes of three-mode data: computational possibilities and problems, COMPSTAT 2004, Charles University, Prague.
Kroonenberg, P.M., De Leeuw, J., 1980. Principal component analysis of three-mode data by means of alternating least squares algorithms. Psychometrika 45, 69–97.
Kuppens, P., Van Mechelen, I., in press. Determinants of the anger appraisals of threatened self-esteem, other-blame, and frustration. Cognition Emotion.
McLachlan, G., 1982. The classification and mixture maximum likelihood approaches to cluster analysis. In: Krishnaiah, P.R., Kanal, L.N. (Eds.), Handbook of Statistics, vol. 2. North-Holland, Amsterdam, pp. 199–208.
Mezzich, J.E., Solomon, H., 1980. Taxonomy and Behavioral Science: Comparative Performance of Grouping Methods. Academic Press, London.
Milligan, G.W., 1980. An examination of the effect of six types of error perturbation on fifteen clustering algorithms. Psychometrika 45, 325–342.
Murakami, T., Kroonenberg, P.M., 2003. Three-mode models and individual differences in semantic differential data. Multivariate Behav. Res. 38, 247–283.
Murtagh, F., 1989. (Review of the book Data, Expert Knowledge and Decisions). J. Classification 6, 129–132.
Paterlini, S., Krink, T., 2006. Differential evolution and particle swarm optimisation in partitional clustering. Comput. Statist. Data Anal. 50, 1220–1247.
Rocci, R., Vichi, M., 2003. Three-mode clustering of a three-way data set, CLADAG 2003. University of Bologna, Bologna.
Schepers, J., Van Mechelen, I., 2004. Three-mode partitioning: model and algorithm, GfKl 2004. University of Dortmund, Dortmund.
Selim, S., Ismail, M., 1984. *K*-means type algorithms: a generalized convergence theorem and characterization of local optimality. IEEE Trans. Pattern Anal. Mach. Intell. 6, 81–87.
Steinley, D., 2003. Local optima in *K*-means clustering: what you don't know may hurt you. Psychol. Methods 8, 294–304.
Van Coillie, H., Van Mechelen, I., 2006. Expected consequences of anger-related behaviors. European J. Personality 20, 137–154.
Van Mechelen, I., Bock, H.-H., De Boeck, P., 2004. Two-mode clustering methods: a structured overview. Statist. Methods Med. Res. 13, 363–394.
Vichi, M., 2002. Double *k*-means clustering for simultaneous classification of objects and variables. In: Borra, S., Rocci, R., Schader, M. (Eds.), Advances in Classification and Data Analysis. Springer, Heidelberg, Germany, pp. 43–51.