Computational Statistics and Data Analysis 52 (2008) 4923–4938

Contents lists available at ScienceDirect

Computational Statistics and Data Analysis

journal homepage: www.elsevier.com/locate/csda

Algorithms for additive clustering of rectangular data tables

Dirk Depril [a,*], Iven Van Mechelen [a], Boris Mirkin [b]

[a] KULeuven, Department of Psychology, Tiensestraat 102 - b3713, 3000 Leuven, Belgium
[b] Birkbeck College, Department of Computer Science, Malet Street, London WC1E 7HX, UK

ARTICLE INFO

Article history:
Received 23 October 2007
Received in revised form 12 April 2008
Accepted 12 April 2008
Available online 18 April 2008

ABSTRACT

The overlapping additive clustering model or principal cluster model is a model for two-way two-mode object by variable data that implies an overlapping clustering of the objects and a set of profiles (characteristic variable values for each cluster). The model values of the variables of an object are the sum of the profiles of its corresponding clusters. In the associated data analysis the data matrix at hand is approximated by an overlapping additive clustering model of a prespecified rank by minimizing a least squares loss function. Recently an algorithm has been proposed for this purpose. This algorithm is a sequential fitting strategy, also called the method of principal clusters (PCL). Theoretical and empirical evidence that the PCL algorithm may have problems in revealing the true structure underlying a data set will be presented. As a way out, three new algorithms to fit the principal cluster model to empirical data will be presented: two of an alternating least squares (ALS) type, orthogonally combined with two different starting strategies, and one based on simulated annealing (SA). In a simulation study it is demonstrated that all three new algorithms outperform the existing PCL algorithm. The amount of objects that belong to more than one cluster (the overlap) is further found to have a considerable influence on the algorithmic performance of the ALS algorithms, with low amounts of overlap requiring a different starting strategy than high ones. As a consequence, for the analysis of real data sets in practice, a hybrid approach will be presented consisting of one of the ALS algorithms initialized by means of the two starting strategies under study.

© 2008 Elsevier B.V. All rights reserved.

1. Introduction

The family of additive clustering models is a broad class of models that imply an overlapping clustering of one or more of the modes of the data set at hand. The archetypical model of this family is the ADCLUS model proposed by Shepard and Arabie (1979). ADCLUS has been developed to represent symmetric similarity data matrices, the two ways of the matrices referring to the same set of objects. In ADCLUS, the objects are grouped into overlapping clusters, which may be interpreted as features, the reconstructed similarity of two objects then equaling the sum of the weights of their common features. As such ADCLUS is a special case of Tverky's contrast model for similarities (Tversky, 1977).

Given a two-way two-mode object by variable data set, one may also be interested in finding overlapping clusters of the objects. A possible solution to this problem can be to preprocess the data by converting them into two-way one-mode similarities and to subsequently fit the ADCLUS model to them. However, there is an almost infinite number of ways to derive one-mode similarities from two-way two-mode data. Therefore, as an alternative, one may go for an additive clustering that represents the two-way two-mode data directly. For this purpose, Mirkin (1987) proposed a one-mode additive clustering model. The model proposed by Mirkin (1987) is similar to ADCLUS in that the objects are grouped into overlapping clusters;

* Corresponding author. Tel.: +32 16 32 58 32; fax: +32 16 32 59 93.
E-mail address: dirk.depril@psy.kuleuven.be (D. Depril).

0167-9473/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.csda.2008.04.014

each cluster further takes a value for each variable, yielding a cluster-specific variable profile. The reconstructed values on the variables for an object then are summations of the profiles of the clusters that object belongs to.

The potential usefulness of the one-mode additive clustering model for two-mode data can be illustrated by means of the following hypothetical medical example. Suppose a patient by symptom data set is given. Cluster-specific symptom profiles then can be interpreted as underlying diseases or syndromes. A patient further can suffer from more than one syndrome (symptom co-morbidity) and therefore can be assigned to multiple syndrome clusters. The final symptom profile of a patient then can be obtained by summing the symptom profiles of the syndromes he or she suffers from.

Given an empirical two-mode data set under study, one may wish to fit a one-mode additive clustering model to it by minimizing a least squares loss function. The relevant part of the solution space for this minimization problem can be shown to be finite, implying that the minimization comes down to a combinatorial problem. In theory it is therefore possible to find the global optimum enumeratively; however, an enumerative strategy becomes quickly infeasible in case of a growing number of objects and clusters. Therefore, suitable optimization heuristics are needed. For this purpose Mirkin (1987, 1990) has developed an algorithm called Principal Cluster Analysis (PCL), which sequentially extracts clusters from a two-mode data set. However, little is yet known about the performance of this algorithm. Moreover, we will show in this paper that PCL has some problems on a theoretical level. As a way out, three new algorithms will be proposed, two of an alternating least squares type and one simulated annealing approach. All algorithms will be evaluated in an extensive simulation study.

The remainder of this manuscript is structured as follows. In Section 2, the one-mode additive cluster model and the associated data analysis are presented. Next, in Section 3, Mirkin's PCL algorithm will be recapitulated. In addition, the theoretical issues with PCL we discovered, will be explained. In Section 4, our three new algorithms will be proposed. A simulation study to explore the performance of the algorithms will be presented in Section 5. Finally, in Section 6, some concluding remarks are given.

2. The One-mode additive clustering model

2.1. The model

The one-mode additive clustering model, as proposed by Mirkin (1987), is a model for an $I \times J$ two-way two-mode data matrix $\mathbf{X}$, of which the entry of the ith row and the jth column is denoted with x_{ij}. The data matrix $\mathbf{X}$ is approximated by an $I \times J$ model matrix $\mathbf{M}$ that, in turn, can be decomposed into an $I \times K$ binary (0/1) matrix $\mathbf{A}$ and a $K \times J$ real-valued matrix $\mathbf{P}$:

$$\mathbf{M} = \mathbf{AP}, \qquad m_{ij} = \sum_{k=1}^{K} a_{ik} p_{kj}, \tag{1}$$

with m_{ij} the element of ith row and jth column of $\mathbf{M}$ and K being the smallest number for which such a decomposition is possible. The columns of $\mathbf{A}$ define K clusters and constitute the *membership* or *cluster matrix*; the entries a_{ik} denote whether object i belongs to cluster k ($a_{ik} = 1$) or not ($a_{ik} = 0$). Zero columns are not allowed since then K would not be minimal. No further restrictions are put on the matrix $\mathbf{A}$, implying that the clustering may be an overlapping one that, moreover, does not have to imply a cover of the object set. The rows of $\mathbf{P}$ contain the *variable profiles* for each cluster and $\mathbf{P}$ is therefore called the *profile matrix*, with entries p_{kj} denoting the value of the jth variable for cluster k. (We use the term profile instead of the more classical term centroid, to point out that the rows of $\mathbf{P}$ do not have to coincide with the mean vectors of the objects in the corresponding clusters.) Expression (1) then means that the values of row i of $\mathbf{M}$ are the sum of the profiles of the clusters object i belongs to.

One may note that in the signal processing domain a model similar to (1) has been proposed (see, e.g. Talwar et al. (1996)). In that case, however, a binary matrix $\mathbf{A}$ is used with entries 1 or -1. More in particular, it is assumed that K signals are arriving at I sensors in a discrete time space of J time points, each of the I sensors detecting a superposition of these K signals. The purpose of the analysis is to recover the K separate signals from the matrix $\mathbf{M}$. This is done by decomposing $\mathbf{M}$ into a real-valued $I \times K$ sensor by signal matrix $\mathbf{P}$ and a $K \times J$ binary ($-1/1$) signal by timepoint matrix $\mathbf{A}$. The sensor by signal matrix $\mathbf{P}$ contains the response intensity of each sensor for each signal (how well the signal is transmitted to the electric circuit behind the sensor) and the signal by timepoint matrix $\mathbf{A}$ contains for each signal the implied information (in $1/-1$ bit format) at each time point. The model is then written as

$$\mathbf{M} = \mathbf{PA}, \tag{2}$$

which upon transposition looks similar to model (1). One should note that, in spite of the apparent similarity between models (1) and (2), from a mathematical viewpoint they are not equivalent. For example, unlike model (1), model (2) can at best be identifiable only upon a reflection of $\mathbf{A}$ and $\mathbf{P}$. In the remainder of this paper we will limit ourselves to model (1) with 0/1 binary matrices.

2.2. Illustrative application

To further clarify the one-mode additive clustering model, we will now present an illustrative application from the domain of the psychology of emotions. Within this domain, one of the currently dominant theoretic approaches is the

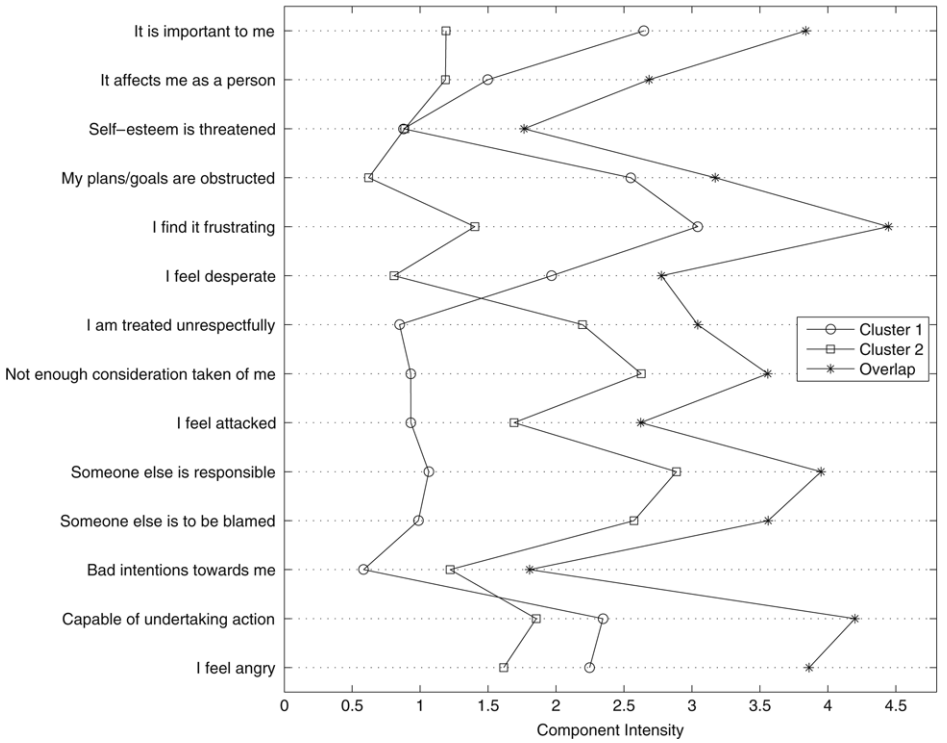


Fig. 1. Component profiles of the situation clusters.

componential one. According to this approach, emotions are conceived as integrated wholes of sets of components (Smith and Lazarus, 1993). Such components may include interpretations or appraisals of the situation, action tendencies, as well as emotional meta-experiences (i.e., the subjective feeling of being anxious, happy, etc.). An important general psychological question in componential emotion research pertains to understanding the mechanisms through which in the average person a subset of emotion components is elicited by a particular situation. To answer this question, one may assume that the elicitation of emotion components is linked to underlying sources or triggers as included in situations. In general, a particular situation may include no, one or several such triggers. Every trigger can further be assumed to elicit each emotional component with a specific level of intensity, resulting in a component profile for the trigger in question. When more than one trigger is present in a given situation, the component intensities as associated with the different triggers can be assumed to combine in an additive way. Given these assumptions, the one-mode additive clustering model seems to be most suitable to induce unknown triggers underlying a given set of emotion component data.

To illustrate, we will analyze a data set of Kuppens and Van Mechelen (2007), which consists of average ratings of 24 unpleasant situations on 14 anger components on a scale ranging from 0 to 6. Those data were obtained by taking the average of component ratings by the individual answers of 357 first year psychology students. A description of the components can be found in Fig. 1. An additive clustering model with $K = 2$ situation clusters was fitted to these data, making use of a least squares loss function (see further Section 2.3); the resulting model accounts for 62% of the total variance in the data. A first cluster consists of situations in which a number of anger components are triggered by the fact that things do not go as planned (e.g., you miss a party because you fell asleep; your floppy disk containing a school assignment gets destroyed in the computer). A second cluster consists of situations in which problems are caused by someone else (e.g., you are hit on your bike by another biker; you are not being served in a restaurant). The intersection of the two clusters consists of situations in which both triggers are present: things are not going as planned due to someone else (e.g., you agree with a friend to go out together, but in the end you don't hear from him/her; a fellow student fails to return your notes when you needed them for studying). The component profiles of the two clusters (as well as of their intersection) are plotted in Fig. 1. As can be seen in this figure, the situations that belong to the first cluster only, unlike those belonging to the second cluster only, score high on appraisals such as "It is important to me", "My plans/goals are being obstructed", and "I find it frustrating". Interestingly, these are all so-called primary appraisals, which concern the relevance of the situation to the subject's personal well-being (Smith and Lazarus, 1993). The situations belonging to the second cluster only in their turn score high on appraisals such as "Not enough consideration taken of me", "I think someone else is to blame" and "I think someone else is responsible". Interestingly, most of these are so-called secondary appraisals, which concern attributions and coping-related interpretations. Situations that belong to both clusters score high on all primary and secondary appraisals as

highlighted above. Moreover, the additivity property can be nicely observed on the components "I am capable of undertaking action against it" and "I feel angry".

The model was fitted with all three new algorithms proposed in Section 4: both alternating least squares algorithms ALS_{lf_2} and ALS_{lf_1} and simulated annealing (SA). Concerning the ALS algorithms, a multistart procedure was used, using both the data based and the random starting strategy (see Sections 4.1 and 4.2). ALS_{lf_2} took 1500 random starts and 1500 data based starts; ALS_{lf_1} took 20 random starts and 20 data based starts. This is exactly the same as in the simulation study (see Section 5.3). Each algorithm (and, moreover, each starting strategy) yielded the same result, most likely because of the relatively small size of the solution space.

2.3. Data analysis: Loss functions

Given an empirical $I \times J$ data array $\mathbf{X}$, a model matrix $\mathbf{M}$ is looked for to optimally approximate $\mathbf{X}$. Note that every data set $\mathbf{X}$ can be exactly represented by a model matrix $\mathbf{M}$ provided that K is sufficiently large (given, e.g., that it always holds that $\mathbf{X} = \mathbf{I}_I\mathbf{X}$ with $\mathbf{I}_I$ denoting the $I \times I$ identity matrix). In practice, however, one may prefer to capture the structure within a data matrix with a relatively small K, rather than to go for a perfect data reconstruction. We therefore allow for discrepancies between $\mathbf{X}$ and $\mathbf{M}$, which means that $\mathbf{X} = \mathbf{M} + \mathbf{E}$. A model matrix $\mathbf{M} = \mathbf{AP}$, with some prespecified number $K < I$ of clusters, is then to be estimated by minimizing the least squares loss function

$$lf(\mathbf{A}, \mathbf{P}) = \|\mathbf{X} - \mathbf{AP}\|_F^2 = \sum_{i=1}^{I}\sum_{j=1}^{J}\left(x_{ij} - \sum_{k=1}^{K} a_{ik}p_{kj}\right)^2, \tag{3}$$

over $\mathbf{A}$ and $\mathbf{P}$, with $\|.\|_F$ denoting the Frobenius norm of a matrix, that is, the square root of the sum of the squared entries. If a membership matrix $\mathbf{A}$ is given, the conditionally optimal $\mathbf{P}$ for that $\mathbf{A}$ is the least squares multiple regression estimator

$$\mathbf{P}_{LS} = (\mathbf{A}'\mathbf{A})^{-1}\mathbf{A}'\mathbf{X}. \tag{4}$$

Plugging this expression into (3) leads to an alternative loss function with one argument only:

$$lf(\mathbf{A}) = lf(\mathbf{A}, \mathbf{P}_{LS}) = \|\mathbf{X} - \mathbf{A}(\mathbf{A}'\mathbf{A})^{-1}\mathbf{A}'\mathbf{X}\|_F^2. \tag{5}$$

In the remainder of this paper we will refer to loss function (3) as lf_2 and to loss function (5) as lf_1, with in each case the subscript denoting the number of arguments. Minimizing lf_2 over $\mathbf{A}$ and $\mathbf{P}$ is equivalent to minimizing lf_1 over $\mathbf{A}$. Concerning the discrepancy or error terms e_{ij}, no distributional assumptions are made on them, meaning that we are taking a deterministic approach. Note, however, that the least squares loss functions (3) and (5) can also be obtained by a maximum likelihood approach assuming independent and identically distributed (iid) normally distributed error terms.

In practice, the number of clusters K is usually unknown. A common strategy then is to fit one-mode additive clustering models for increasing numbers of clusters K and then to select a final value of K by means of some rank selection heuristic (like, e.g., a scree-plot). An alternative, as used by Mirkin (1990), is to take the number of clusters that explains a prespecified proportion of the total variance in the data, or that is such that the difference in explained variance with the subsequent number of clusters is less than a prespecified proportion. Also, one could decide on K on the basis of substantive considerations.

Upon closer examination of loss function (5), it can be easily seen that the relevant part of the solution space for the minimization problem at hand is finite and consists of all 2^{IK} possible binary membership matrices $\mathbf{A}$. In principle it is thus possible to find the global optimum enumeratively. However, since the size of the relevant part of the solution space is exponential in I and K, this strategy will become rapidly computationally infeasible. Therefore suitable algorithms or heuristics need to be developed. Both new and existing algorithms will be presented and discussed in the next sections.

3. Principal cluster analysis

3.1. PCL algorithm

To minimize loss function (3), Mirkin (1987, 1990) proposed a sequentially fitting (SEFIT) algorithm, called Principal Cluster Analysis (PCL). This algorithm extracts clusters one by one from a data set, and is inspired by the stepwise extraction of principal components from a data set. The clusters are sought on residual data, defined as follows:

$$x_{ij}^1 = x_{ij} \quad \text{and} \quad x_{ij}^m = x_{ij} - \sum_{k<m} a_{ik}p_{kj}, \qquad \text{for } m = 2, \ldots, K, \tag{6}$$

meaning that the first $m-1$ clusters are subtracted from the data to yield the mth residual data x_{ij}^m. The first cluster is then sought on the data x_{ij}^1 and, in general, the mth cluster is sought on x_{ij}^m, leaving the previous $m-1$ clusters as they are. The quest for the mth cluster is done by minimizing the residual loss function

$$\sum_{i=1}^{I}\sum_{j=1}^{J}(x_{ij}^m - a_{im}p_{mj})^2. \tag{7}$$

Note that whenever 0/1 cluster memberships a_{im} $(i = 1, \ldots, I)$ are given, the conditionally optimal profile values p_{mj} $(j = 1, \ldots, J)$ are given by

$$p_{mj} = \frac{\sum_{i=1}^{I} a_{im} x_{ij}^{m}}{\sum_{i=1}^{I} a_{im}}, \quad \text{for } j = 1, \ldots, J, \tag{8}$$

which is the residual data average of the objects in the cluster. The full estimation procedure for a_{im} $(i = 1, \ldots, I)$ and p_{mj} $(j = 1, \ldots, J)$ then reads as follows. First start with an empty cluster (i.e., with $a_{im} = 0$ for $i = 1, \ldots, I$) and sequentially add objects to it in a greedy way, that is, add each time the object that yields the smallest residual loss value (7). (Note that for every object that is considered for joining the cluster, a new profile and a new loss value have to be calculated.) Continue, until there is no further decrease in the loss function. Subsequently, the product $a_{im}p_{mj}$ of the newly obtained membership vector and cluster profile is subtracted from x_{ij}^{m} yielding the new residual data $x_{ij}^{m+1} = x_{ij}^{m} - a_{im}p_{mj}$. The entire procedure is repeated on x_{ij}^{m+1} and the algorithm stops when the prespecified number of clusters K is reached. The pseudocode for PCL can be found in Appendix A.1.

This sequential updating procedure of a cluster is in line with the original version of PCL as proposed by Mirkin (1990). As an alternative, one could try to find the conditionally optimal cluster memberships by enumeratively evaluating all 2^I possible 0/1 binary memberships a_{im} $(i = 1, \ldots, I)$. This would, however, become quickly computationally infeasible for increasing number of objects I.

3.2. Some problems with PCL

The PCL algorithm can be shown to suffer from problems with regard to recovering the true model underlying a data set whenever the clustering of the true model is an overlapping one. Consider, as an illustration, a data set $\mathbf{X} = \mathbf{M} + \mathbf{E}$ with the following underlying structure M:

$$\mathbf{M} = \mathbf{AP} = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 1 \\ 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \end{pmatrix} = \begin{pmatrix} p_{11} & p_{12} & p_{13} \\ p_{11} & p_{12} & p_{13} \\ p_{11} & p_{12} & p_{13} \\ p_{11} + p_{21} & p_{12} + p_{22} & p_{13} + p_{23} \\ p_{11} + p_{21} & p_{12} + p_{22} & p_{13} + p_{23} \\ p_{21} & p_{22} & p_{23} \end{pmatrix}. \tag{9}$$

For convenience we put $\mathbf{E} = 0$ (or, equivalently, $\mathbf{X} = \mathbf{M}$). Suppose now that the first membership column $(1, 1, 1, 1, 1, 0)'$ has correctly been recovered by the PCL algorithm. In the next step of PCL, the profile values p_{1j} $(j = 1, 2, 3)$ are calculated by means of Eq. (8). The profile reads

$$p_{1j} = p_{1j} + 2p_{2j}/5, \quad \text{for } j = 1, 2, 3,$$

which is not equal or even close to the true profile p_{1j} $(j = 1, 2, 3)$ since an extra term $2p_{2j}/5$ is added to each entry p_{1j} $(j = 1, 2, 3)$. Clearly a bias has been introduced due to the cluster overlap. All further estimates may be influenced by this bias since the wrong profile will be subtracted from the objects of the first cluster in the next step of the algorithm.

4. Three new algorithms

4.1. ALS$_{lf_2}$

A first new algorithm, called ALS$_{lf_2}$ is of an alternating least squares (ALS) type and optimizes loss function (3) (lf_2). The algorithm alternatingly estimates the membership matrix $\mathbf{A}$ and the profile matrix $\mathbf{P}$, conditionally optimal upon the other. In particular, the algorithm starts from a randomly drawn or an a priori given membership matrix $\mathbf{A}_0$. (As an alternative, one could also start the algorithm with an initial profile matrix $\mathbf{P}_0$) Next, the conditionally optimal $\mathbf{P}_0$ upon $\mathbf{A}_0$ is calculated, making use of Eq. (4): $\mathbf{P}_0 = (\mathbf{A}_0'\mathbf{A}_0)^{-1}\mathbf{A}_0'\mathbf{X}$. Given this new $\mathbf{P}_0$, a new membership matrix $\mathbf{A}_1$ is estimated conditionally optimal upon $\mathbf{P}_0$. For this estimation a separability property (Chaturvedi and Carroll, 1994) of loss function lf_2 is used. Indeed, this loss function can be written as

$$lf_2(\mathbf{A}, \mathbf{P}) = \sum_j \left(x_{1j} - \sum_{k=1}^{K} a_{1k}p_{kj} \right)^2 + \cdots + \sum_j \left(x_{Ij} - \sum_{k=1}^{K} a_{Ik}p_{kj} \right)^2. \tag{10}$$

This implies that the contribution of the ith row of $\mathbf{A}_1$ to the loss function can be separated from the contributions of the other rows. As a consequence $\mathbf{A}_1$ can be estimated row by row; this reduces the work to evaluating $I2^K$ possible membership rows instead of the full number of 2^{IK} membership matrices. The estimation of each row of $\mathbf{A}_1$ is done enumeratively. Next, a new profile matrix $\mathbf{P}_1$ is estimated by means of (4) and so on, until there is no more decrease in the loss function.

Since the alternating procedure provides consecutive conditionally optimal estimates, it generates a nonincreasing row of loss values bounded below by zero, which necessarily converges. Moreover, convergence is reached in a finite number of steps since the solution space is finite.

To generate an initial membership matrix $\mathbf{A}_0$, we propose two strategies. A first one consists of drawing the entries of $\mathbf{A}_0$ iid from a Bernoulli distribution with parameter $\pi = 0.5$. A second one consists of taking the conditionally optimal memberships upon K randomly drawn data points.

The result of the algorithm strongly depends on the seeded matrix $\mathbf{A}_0$ and in general the ALS_{lf_2} algorithm will yield a local rather than a global optimum. To remedy for this, a multistart procedure is recommended with the solution with the lowest loss value to be retained. The pseudocode for the algorithm, seeded with one membership matrix $\mathbf{A}_0$, is given in Appendix A.2.

It should be noted that, strictly speaking, loss function lf_2 (3) is not fully separable as matrix $\mathbf{A}_1$ should be of full column rank (Van Mechelen and Schepers, 2007). Otherwise, if in the course of the iteration process a zero column shows up in $\mathbf{A}_1$, a problem is encountered for the updating of $\mathbf{P}_1$ by means of (4) as in that case, the inverse of $\mathbf{A}_1'\mathbf{A}_1$ does not exist. We will deal with this problem by using the Moore–Penrose pseudo-inverse to calculate the least squares profiles (4). In the subsequent iteration the algorithm will then most often return to a full rank matrix $\mathbf{A}_1$, especially when the values of those memberships that do not affect the loss value are put equal to 1.

4.2. ALS_{lf_1}

A second new algorithm, called ALS_{lf_1} is also of an alternating least squares type and minimizes loss function (5) (lf_1). Note that this loss function, unlike loss function lf_2, is not separable. The algorithm is seeded with a randomly drawn or a priori given membership matrix $\mathbf{A}_0$. This membership matrix is updated on the basis of a partitioning of the set of its elements into a series of subsets, with each subset being alternatingly re-estimated conditionally upon the other subsets. The subsets of the partitioning can be, for example, all rows or all separate entries. Here, we choose to update $\mathbf{A}_0$ row by row, analogously to ALS_{lf_2}.

One full update of $\mathbf{A}_0$ then looks as follows. The algorithm starts with the first row and looks for the binary row pattern that yields the lowest loss value (5) given the other rows of $\mathbf{A}_0$. The search for the best binary row pattern is done enumeratively and the new row, which can be either the same or an actual new one, is immediately plugged into $\mathbf{A}_0$. The algorithm then repeats this procedure for the second row and so on. After a pass through all rows, the loss value of the newly obtained membership matrix $\mathbf{A}_1$ is compared to the loss value of $\mathbf{A}_0$. If a decrease has been established, the rowwise updating procedure is repeated on $\mathbf{A}_1$. The process stops when there is no further decrease in the loss function. This algorithm differs from the one in the previous section in that for each row update the conditionally optimal profiles are recalculated immediately, whereas in ALS_{lf_2} the profiles are only updated at the end of the rowwise membership updating.

To start the algorithm, the two strategies proposed in the previous section can be used here as well. In particular, the entries of the initial membership matrix $\mathbf{A}_0$ can be drawn randomly out of a Bernoulli distribution with parameter $\pi = 0.5$ or A_0 can be the conditionally optimal membership matrix upon K randomly drawn data points.

The algorithm generates a sequence of membership matrices each with the same or a lower loss value than its predecessor. A nonincreasing row of positive loss values is thus obtained, which consequently converges. Moreover, convergence is reached in a finite number of steps, since there are only a finite number of membership matrices.

Starting from only one single $\mathbf{A}_0$ the algorithm may yield a local rather than the global optimum. To remedy for this, a multistart procedure is to be used and the solution with the lowest loss value across all runs is to be selected. The pseudocode for ALS_{lf_1} can be found in Appendix A.3.

Whenever a rank deficient membership matrix $\mathbf{A}$ is encountered during the updating of the rows, the inverse of $\mathbf{A}'\mathbf{A}$ does not exist and the Moore–Penrose inverse will be calculated instead. Usually, in the next step the membership matrix will no longer be rank-deficient.

4.3. SA_{lf_1}

A third algorithm, called SA_{lf_1}, aims at minimizing loss function lf_1 and is of a simulated annealing (SA) type (Aarts et al., 1997), which implies a walk through the solution space of all possible membership matrices. This algorithm starts with a randomly drawn membership matrix $\mathbf{A}_0$, the entries of which are iid drawn from a Bernoulli distribution with parameter $\pi = 0.5$, together with its loss value $lf^0 = lf_1(\mathbf{A}_0)$. It then randomly chooses a new membership matrix $\mathbf{A}_1$ out of the set of *neighbour* membership matrices of $\mathbf{A}_0$. Neighbour matrices are defined to be close to the original matrix in the sense that only a relatively small number of entries differ. In particular, here we define the neighbours of a binary membership matrix as all matrices that have exactly one row with different entries. To generate a neighbour, a row is chosen at random and replaced by a randomly chosen 0/1 binary pattern out of all 2^K possible membership row patterns. (Note that the definition of neighbours and the associated generation procedure implies an algorithmic analogy with the rowwise updating of the membership matrices in the ALS algorithms.) Given a neighbour $\mathbf{A}_1$, the loss value $lf^1 = lf_1(\mathbf{A}_1)$ is calculated and when smaller than lf^0 the matrix $\mathbf{A}_1$ is accepted (i.e., $\mathbf{A}_0$ is replaced by $\mathbf{A}_1$). If, however, the loss value is higher, $\mathbf{A}_1$ is accepted with a probability p_a, defined as

$$p_a = \exp((lf^0 - lf^1)/T), \tag{11}$$

with T being called the *temperature*. The whole procedure is then repeated on $\mathbf{A}_0$, which is thus either the newly accepted matrix or the old matrix. During the process of generating and accepting membership matrices, the temperature is slowly decreased. This implies that at the beginning of the algorithmic process the probability of accepting worse solutions is higher than at the end. According to the recommendations in Aarts et al. (1997), we keep the temperature constant for $I2^K$ consecutively generated membership matrices, such a sequence being called a chain. The temperature is then lowered to $T = \alpha T, \alpha \in (0, 1)$. We chose to set $\alpha = 0.975$ since pilot studies revealed that this implies a good trade-off between speed and accuracy of the annealing process. When a final temperature T_f close to zero is reached (set here to $T_f = 10^{-5}$), the algorithm stops and the best encountered solution is reported. To determine the initial temperature, a full chain is run in which all solutions are accepted; the initial temperature T_0 is then set equal to $T_0 = -m/\log(0.8)$, where m is the mean of all subsequent absolute differences in loss values encountered during this chain. This initial temperature T_0 is an estimate of the temperature at which 80% of worse solutions is expected to be accepted. As a full simulated annealing run can be very time consuming, we further built in two acceleration strategies. First, when the number of accepted solutions within one chain reaches 10% of the predefined chain's length of $I2^K$ membership generations, the chain is stopped and a new chain with a lower temperature is started. Second, when 10 consecutive chains end with the same solution, the algorithm stops. One run of the thus resulting SA_{lf_1} algorithm typically generates about 50 000 solutions. The pseudocode for the SA algorithm, is given in Appendix A.4.

5. Simulation study

5.1. Introduction

In the previous sections, four algorithms were presented to estimate the best fitting one-mode additive clustering model for a given data set: the existing PCL algorithm and three new ones. In this section, we will present a simulation study to evaluate the performance of these algorithms. In this regard, we are interested in two aspects of algorithmic performances: goodness of fit and goodness of recovery. With regard to goodness of fit, we will examine whether an algorithm finds the global optimum of the loss function. Concerning goodness of recovery, we will investigate to what extent each algorithm succeeds in recovering the true structure underlying a given data set. Algorithmic performance will be evaluated on a global level as well as a function of data characteristics. Furthermore, it will be examined both from the viewpoint of the four algorithms as a whole as from the viewpoint of algorithmic differences.

In the next subsections we will outline the design of the simulation study on the level of data generation (Section 5.2), the algorithms (Section 5.3) and the specific evaluation criteria (Section 5.4). In Section 5.5 the results will be presented, and in Section 5.6 follows a discussion.

5.2. Data generation

To generate data sets $\mathbf{X}$ of size $I \times J$, we will independently generate matrices $\mathbf{A}$, $\mathbf{P}$ and $\mathbf{E}$. The matrix $\mathbf{A}$ is a binary $I \times K$ matrix the columns of which define K overlapping clusters; the rows of $\mathbf{A}$ are independently drawn from a multinomial distribution on all possible 2^K binary row patterns (with a probability of 0.05 for the zero pattern). The matrix $\mathbf{P}$ is a real-valued $K \times J$ matrix the rows of which contain the K profiles of the clusters implied by $\mathbf{A}$; the columns of $\mathbf{P}$ are independently drawn from an equicorrelated K-variate normal distribution $N(\mathbf{0}, \Sigma_{\mathbf{P}})$, with equal variances. The matrix $\mathbf{E}$ is a real-valued $I \times J$ matrix containing error terms; its rows are independently drawn from an equicorrelated J-variate normal distribution $N(\mathbf{0}, \Sigma_{\mathbf{E}})$ with equal variances. A data set $\mathbf{X}$ is then obtained as $\mathbf{X} = \mathbf{AP}+\mathbf{E}$. The matrices $\mathbf{A}$ and $\mathbf{P}$ constitute the true one-mode additive clustering model underlying $\mathbf{X}$ and their product $\mathbf{AP}$ will further be denoted as $\mathbf{M}$.

The following design factors are manipulated on the level of the data generation.

- *Data shape:* The data shape of $\mathbf{X}$ is defined as the ratio I/J and will take three different levels: 4:1, 1:1 and 1:4. The number of entries of $\mathbf{X}$ will always be equal to 1024, implying three different values for $I \times J$: 64×16, 32×32 and 16×64.
- *Number of clusters K*: 3, 4, 5.
- *Amount of cluster overlap:* This is defined as the probability of belonging to more than one cluster and it is put equal to 25%, 50% or 75%; for this purpose, the multinomial probabilities of all row patterns in $\mathbf{A}$ that contain more than one 1 were put equal to one another with a total equal to the percentage in question.
- *Relative cluster sizes:* This is defined in terms of the probability distribution over the distinct parts of the different clusters which may be uniform or nonuniform; for this purpose, the multinomial probabilities of the patterns in $\mathbf{A}$ that contain a single 1 are either equal or unequal, with in the latter case the largest and lowest probabilities having a ratio 4:1, and with all other probabilities being in a ratio 2:1 to the smaller one.
- *Profile correlation:* This is defined as the expected correlation between the rows of $\mathbf{P}$ and is controlled through the off-diagonal elements of $\Sigma_{\mathbf{P}}$. Those are put equal to either 0 or 0.5.
- *Noise level* ε: This is defined as the proportion ε of the total variance in the data $\mathbf{X}$ accounted for by $\mathbf{E}$. It is controlled through the value of the variances on the diagonal of $\Sigma_{\mathbf{E}}$. This proportion ε will be either of 0, 0.05, 0.10, 0.20 or 0.40.
- *Noise correlation:* The correlations of the equicorrelated distribution $N(\mathbf{0}, \Sigma_{\mathbf{E}})$ are either 0 or 0.3.

All design factors were fully crossed. This yields 3 (*Data shape*) × 3 (*Number of clusters*) × 3 (*Amount of cluster overlap*) × 2 (*Relative cluster sizes*) × 2 (*Profile correlation*) × 5 (*Noise level*) × 2 (*Noise correlation*) = 1080 combinations for each of which 20 replicates were generated yielding 20 × 1080 = 21 600 simulated data sets.

5.3. Algorithms

Four algorithms will be evaluated in the simulation study: PCL, ALS_{lf_2}, ALS_{lf_1} and SA_{lf_1}. The two ALS algorithms will make use of a multistart procedure and will be seeded with both starting strategies we proposed, namely with random membership matrices, denoted by the superscript *random*, and with conditionally optimal membership matrices derived on the basis of K randomly drawn data points, denoted by the superscript *data*. As such we have an algorithmic factor in our design with six different levels: PCL, $\text{ALS}_{lf_2}^{random}$, $\text{ALS}_{lf_2}^{data}$, $\text{ALS}_{lf_1}^{random}$, $\text{ALS}_{lf_1}^{data}$, SA_{lf_1}.

Concerning the number of starts, it was decided to run the simulated annealing algorithm SA_{lf_1} only once since it is fairly time consuming. For the ALS algorithms, a multistart procedure will be used, the best solution of which will be reported and analyzed. The number of starts for those algorithms is chosen such that they perform the same amount of work as the SA_{lf_1} algorithm expressed in terms of arithmetic operations needed to calculate profiles and/or loss function values. To determine these, a small pilot study was run in which for several settings of I, J and K the number of starts needed for each ALS algorithm was dynamically determined, such that on average an equal number of arithmetic operations was performed as the number of such operations in one run of the SA_{lf_1} algorithm. It was found that one run of SA_{lf_1} is equivalent to approximately 1500 ALS_{lf_2} runs and approximately 20 ALS_{lf_1} runs. With these numbers of starts, the number of arithmetic operations needed is about 10^9 for each algorithm, with our implementations of the ALS_{lf_2}, ALS_{lf_1} and SA_{lf_1} algorithms taking about 40 CPU seconds each (on a 2.80 GHz single core Pentium IV desktop pc with 1 GB of RAM). The single started PCL takes about 0.02 seconds and cannot take any more time.

Every data set is analyzed with the same number of clusters as the true underlying model. On the one hand, this can be considered as a limitation since in practice one may also want to analyze data sets in other ranks than the true one. On the other hand, analyzing a simulated data set in its true rank allows us to use the true underlying model to determine an upper bound on the loss value and to fully investigate the recovery of the membership and profile matrices.

5.4. Evaluation criteria

In the subsequent subsections, the measures to evaluate the algorithms will be explained. In line with the two evaluation aspects explained in Section 5.1 we will distinguish between two types of measures: the first type pertains to the minimization of the loss function and the second to the recovery of the underlying clustering model.

5.4.1. Minimization measure

On each data set we want to determine for each algorithm whether it reached the global optimum of the loss function. However, when adding error **E** to a true matrix **M**, we do not know this global optimum. Because of this we introduce the concept of *proxy* or *pseudo-optimum*. This proxy is determined for each data set separately and acts as an approximation of the global optimum. For each data set we will determine whether or not each algorithm reached the proxy.

In particular, the proxy for each data set is determined as follows:

1. First of all, an *upper bound* (UB) on the loss value is determined. An obvious candidate upper bound is the loss of the true underlying clustering model $\mathbf{M} = \mathbf{AP}$. However, we will use a better upper bound by running the ALS_{lf_2} and ALS_{lf_1} algorithms seeded with both the true memberships **A** and the conditionally optimal memberships upon the true profiles **P**; the best of the four resulting loss values will be taken as the upper bound UB.
2. All six algorithms are run on the data set which yields six loss values $L_1^2, \ldots, L_6^2$.
3. The value of the proxy is then given by $\min(\text{UB}, L_1^2, \ldots, L_6^2)$.

Note that either no, one or several algorithms can reach the proxy for a given data set at hand.

5.4.2. Recovery measures

Each algorithm will yield estimates $\hat{\mathbf{A}}$, $\hat{\mathbf{P}}$, which combine to $\hat{\mathbf{M}}$ by $\hat{\mathbf{M}} = \hat{\mathbf{A}}\hat{\mathbf{P}}$. Recovery now can be measured on the level of **A** (cluster recovery), **P** (profile recovery) and **M** (model recovery).

Clustering recovery

To evaluate the quality of the clustering $\hat{\mathbf{A}}$ found by an algorithm, we will calculate the expression

$$\left(1 - \frac{\sum_{i,k=1}^{I,K} |a_{ik} - \hat{a}_{ik}|}{IK}\right) 100. \tag{12}$$

Table 1
Average results for each algorithm on each evaluation criterion

Algorithm	Proxy reached %	GOC	GOP	GOM
$ALS_{lf_2}^{random}$	74.58	94.20	73.46	74.55
$ALS_{lf_2}^{data}$	66.27	93.58	70.08	71.47
$ALS_{lf_1}^{random}$	77.66	93.31	69.38	72.91
$ALS_{lf_1}^{data}$	64.07	92.16	63.40	69.38
SA_{lf_1}	66.48	92.16	59.15	69.65
PCL	0.03	76.39	14.12	20.29

To take the permutational freedom of the order of clusters into account, we will define the Goodness of Clustering (GOC) as the minimum value of (12) over all column permutations of $\hat{\mathbf{A}}$. The GOC takes values in the interval [0, 100], with a value of 100 meaning perfect recovery.

Profile recovery

To evaluate the quality of the resulting profiles $\hat{\mathbf{P}}$ found by an algorithm, we will calculate the expression

$$\left(1 - \frac{\sum_{k,j=1}^{K,J} (p_{kj} - \hat{p}_{kj})^2}{\sum_{k,j=1}^{K,J} (p_{jk} - \bar{p})^2}\right) 100, \tag{13}$$

where $\bar{p}$ is the average of all the entries of the true profile matrix $\mathbf{P}$. Note that Eq. (13) is a rescaling of the sum of the Euclidean distances between corresponding profiles. To take the permutational freedom of the order of clusters into account, we will define the Goodness of Profiles (GOP) as the minimum value of (13) over all row permutations of $\hat{\mathbf{P}}$. The GOP takes values in the interval $(-\infty, 100]$, with a value of 100 meaning perfect recovery.

Model recovery

To evaluate the quality of the resulting model $\hat{\mathbf{M}}$ found by an algorithm, we calculated the Goodness of the Model (GOM), defined as

$$\text{GOM} = \left(1 - \frac{\sum_{i,j=1}^{I,J} (m_{ij} - \hat{m}_{ij})^2}{\sum_{i,j=1}^{I,J} (x_{ij} - m_{ij})^2}\right) 100. \tag{14}$$

The GOM expresses the difference between the estimated and the true model relative to the amount of error, or, stated differently, how much the estimation comes closer to the truth than the data. The GOM takes values in the interval $(-\infty, 100]$, with a value of 100 indicating perfect recovery, and with negative values meaning that the estimated model is further from the truth than the data is.

5.5. Results

Minimization of the loss function

With regard to the minimization of the loss function, the proxy was on average reached on 58.19% of the data sets (with on 98.42% of the generated data sets the proxy being reached by one or more of the algorithms). The averages for each algorithm separately can be found in Table 1. The most striking algorithmic difference we observed, was the poor performance of the PCL algorithm which reached the proxy on only 0.03% of the data sets. If the PCL algorithm is left out of the simulation results, the proxy is on average reached on 69.81% of the data sets.

Furthermore, considerable differences were found between the different cells of the design of the study and between the different algorithms. To capture these differences, we calculated for each cell of the design and for each individual algorithm the percentage of data sets on which the proxy had been reached. We then analyzed these percentages by means of a factorial analysis of variance with all data characteristics and the algorithmic factor as independent variables (with the algorithm being treated as a repeated measures factor). Note that we left PCL out from the algorithmic factor, since otherwise the algorithmic main effect would account for the main proportion of the variance. Since we have only one observation per algorithm in each cell, we fitted a reduced model by omitting the highest order interaction term. Below, we will only focus on main effects with an effect size (η^2) of 0.05 or higher. For the minimization performance, these are given in the upper panel of Table 2.

As can be observed in this table, minimization performance is on average influenced by three different design characteristics. The first one is the true number of clusters K, which has a negative influence on the results: An increasing

Table 2
Most important effects in analyses of variance for each evaluation criterion

Criterion	Effect	η^2
Reaching proxy	Number of clusters K	.20
	Algorithms * Amount of overlap	.16
	Amount of noise	.15
	Algorithms * Amount of noise	.09
	Profile correlation	.06
Cluster recovery: GOC	Amount of noise	.23
	Algorithms * Amount of overlap	.07
	Profile correlation	.06
	Number of clusters K	.05
Profile recovery: GOP	Amount of noise	.21
	Data shape	.08
	Number of clusters K	.07
Model recovery: GOM	Number of clusters K	.13
	Amount of noise	.12
	Data shape	.09
	Noise correlation	.06

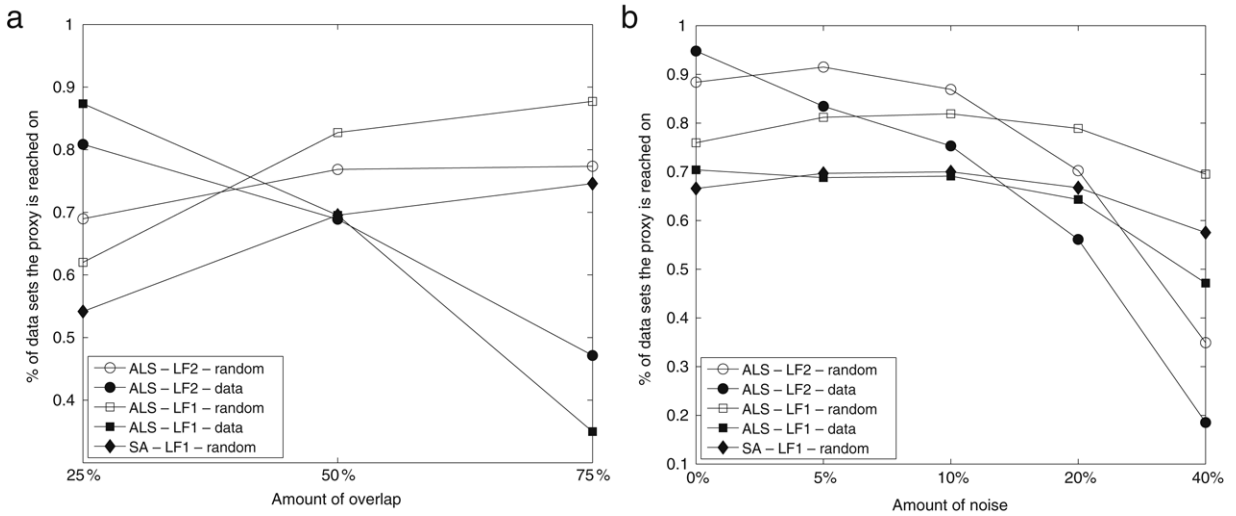


Fig. 2. Interaction between the algorithms and the amount of overlap (a) and the amount of noise in the data (b) on the percentages of reaching the proxy (all sub- and superscripts have been put behind the algorithm's name).

number of clusters in the model makes it more difficult for an algorithm to reach the proxy. A second influential factor is the amount of noise, with the more noise on the data, the more difficult it is for an algorithm to reach the proxy. The third important factor is the profile correlation: If the rows of the profile matrix P are correlated, it is more difficult for an algorithm to reach the proxy.

With regard to algorithmic differences, first, there appeared to be no sizeable main effect of the algorithmic factor. Second, two sizeable interactions involving the algorithmic factor were found. The first of these is the interaction between the algorithmic factor and the amount of overlap in the data (see Fig. 2(a)). From this figure it can be seen that the interaction is of a disordinal type with for low amounts of cluster overlap the ALS algorithms seeded with the data based starting strategy outperforming the ones seeded with random membership matrices and with the reverse holding for high amounts of overlap. The second sizeable interaction is that between the algorithmic factor and the amount of noise on the data. This interaction is depicted in Fig. 2(b). In this figure it can be seen that the interaction in question is again disordinal and stems from the fact that the ALS_{lf_2} algorithm does not perform very well at high amounts of error, regardless of the type of starting strategy.

Recovery

The average values for the three recovery measures GOC, GOP and GOM are 90.12, 58.27 and 56.28 respectively. The averages per algorithm for each recovery criterion can be found in Table 1, in which the poor performance of PCL is striking. If the PCL algorithm is left out, the average recoveries of the other algorithms were 92.87, 67.09 and 71.59 for

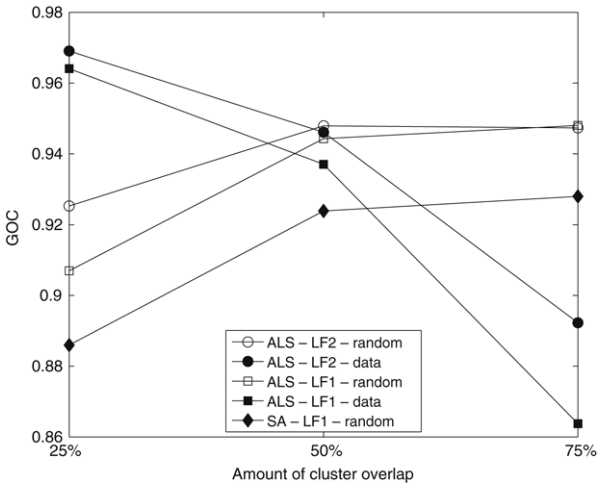


Fig. 3. Interaction between the algorithms and the amount of overlap for the Goodness of Clustering (GOC) (all sub- and superscripts have been put behind the algorithm's name).

the clustering (GOC), profiles (GOP) and the model (GOM) respectively. Furthermore, considerable differences between the different cells of the simulation design as well as between the different algorithms were observed. To examine the remaining differences between design cells and between the algorithms in more detail, factorial analyses of variance were conducted on each recovery criterion separately with the data factors of the simulation design and the algorithmic factor acting as the independent variables (with the latter further being treated as a repeated measures factor). Analogously to the analysis of the minimization results, PCL is left out of the analyses to avoid the fact that the main effect of the algorithmic factor would account for the main proportion of the variance. Significant effects with an effect size (η^2) of 0.05 or greater are tabulated in the lower three panels of Table 2 together with their effect size η^2.

As can be seen in this table, two data characteristics have on average an effect on all three recovery measures: the true number of clusters K and the amount of noise on the data, with an increasing number of clusters and an increasing amount of noise yielding worse recoveries. Furthermore, the data shape was found to have a sizeable influence on profile and model recovery, with relatively lower number of rows yielding worse recoveries (especially in cases with more variables than data points). Finally, profile correlation and noise correlation were also found to yield worse cluster and model recoveries, respectively.

With regard to algorithmic differences, first, we observed no sizeable algorithmic main effect. Second, a sizeable interaction showed up between the algorithmic factor and the amount of cluster overlap (for cluster recovery) which is depicted in Fig. 3. The interaction appeared to be disordinal with for low amounts of overlap the ALS algorithms seeded with the data based starting strategy outperforming the ones seeded with random membership matrices and with the reverse holding for high amounts of overlap.

5.6. Discussion

Two types of factors of the simulation design were found to have a sizeable effect on algorithmic performance: factors that have an effect on average and factors that interact with algorithmic differences. In the subsequent paragraphs these two types of factors will be discussed more in detail.

Four factors were found to influence minimization and/or recovery results on average: the true number of clusters K, the amount of noise on the data, the profile correlation and the data shape. An increasing level of each of these factors appears to be associated with worse algorithmic performance (with increasing data shape level meaning a decreasing number of objects I and an increasing number of variables J). This can be explained as follows: First, the minimization results and all recoveries are worse for an increasing number of clusters K in the model, since the solution space for the minimization of the loss functions (3) (lf_2) and (5) (lf_1) grows exponentially with increasing K. Secondly, the minimization results and all recoveries are worse for an increasing amount of noise on the data (which is natural). Third, the presence of profile correlation has a negative influence on cluster recovery and on reaching the proxy because of a multicollinearity problem that is present when estimating a membership matrix upon (an estimate of) a row correlated matrix P. Moreover, when the rows of P are correlated, the individual profiles will lie closer to each other, making the separate clusters less distinct from each other (which may further imply a harder recovery). Fourth, data shape (and in particular a smaller number of rows I) influences the profile and model recovery because of the alternating nature of our algorithms. In case of a small number of objects I, the ALS structure will imply the bottleneck that a large number of JK profile values need to be estimated on the basis of only a small number of IK pieces of binary membership information.

Table 3
Percentage of data sets per algorithm for which the proxy was found for the 20 data sets of size 64 × 16, with 5 clusters, 75% of cluster overlap, equal cluster sizes, 40% of error, no error correlation and no profile correlation

Algorithm	Number of multistarted runs[a]	
	1	51
$ALS_{lf_2}^{random}$	0	10
$ALS_{lf_2}^{data}$	0	5
$ALS_{lf_1}^{random}$	35	100
$ALS_{lf_1}^{data}$	40	50
SA_{lf_1}	65	95

[a] One run of $ALS_{lf_2}^{random}$ and $ALS_{lf_2}^{data}$ takes 1500 starts; one run of $ALS_{lf_1}^{random}$ and $ALS_{lf_1}^{data}$ takes 20 starts and one run of SA_{lf_1} takes a single start.

Regarding algorithmic differences, a first observation pertains to the rather poor performance of the PCL algorithm. A possible explanation for this might be that in its original version Mirkin (1990), PCL was applied to column-centered data. To test this explanation, we conducted alternative PCL analyses on the centered data sets of our simulation study, the performance of which was evaluated in terms of the original data. This, however, yielded only a slight improvement in the minimization results (with the proxy being reached for 0.07% of the data sets in contrast to 0.03% for the uncentered PCL algorithm). This implies that other explanations for the performance of the PCL algorithm have to be looked for. One possible such explanation might be the fact that PCL is a single start algorithm and as such delivers only one single solution. Another explanation might be the fact that this single solution is the only one within the entire solution space that has been inspected by the PCL algorithm (as opposed to our ALS algorithms which inspect more than one solution, even when only one start is taken). To overcome this latter limitation, one could develop new algorithmic strategies that take the PCL solution as an intial estimate. In our case, one could consider to use the PCL estimates as a start of one of the alternating ALS algorithms. To test this explanation we reran each of our two ALS algorithms four times on the data sets of our simulation study by seeding them with either the memberships or the profiles as obtained from a PCL analysis on either the regular data or the column centered data. From the eight resulting algorithmic strategies the best performance was observed for ALS_{lf_1} seeded with the memberships found by PCL on the column centered data which reached on 25% of the data sets the proxy. For a single data set, this algorithmic strategy required on average only 4% of the total amount of iterations that were needed for 20 randomly started runs of the $ALS_{lf_1}^{random}$ strategy. The performance of this algorithmic strategy is much better than that of PCL, but does still not match the multistart ALS performance level (and cannot be improved by allowing it more time). As a conclusion, the performance of PCL, when looking for overlapping clusters, might be attributed to both the fact that it generates a single solution and the fact that this single solution is the only one inspected within the entire solution space. In contrast, its "non-overlapping" version performs rather well in experiments described by Ming-Tso Chiang and Mirkin (2006).

Concerning algorithmic differences between the novel ALS strategies as proposed in this paper, first, an algorithmic main effect of them was not observed, which means that the strategies marginally perform equally well. Second, however, a sizeable interaction between the algorithmic factor and the amount of overlap was observed with regard to both reaching the proxy and cluster recovery. More in particular, as appears from Fig. 2(a), seeding the ALS algorithms with a data based start appears to be the best option when there is only a small amount of cluster overlap, whereas in case of a high amount of overlap, seeding with random memberships appears to be preferable. This can be explained by the fact that in case of small amounts of cluster overlap a lot of objects belong to a single cluster only. Hence, when drawing K data points in that case, the probability of ending up with K points that stem from K different single clusters is rather high. However, when the amount of overlap increases, the probability of drawing such a set of points will be much smaller, suggesting that a random start is relatively more beneficial.

Up to now we have discussed, on a theoretical level, the different factors that influence algorithmic performance. One may, however, also wish to know on a practical level how the results of our simulation study can be translated into guidelines for the analysis of real life data sets. In the following two such guidelines will be given.

A first guideline pertains to the factors that affect the algorithmic performance in general (viz., the number of clusters, the amount of noise, the profile correlation and the data shape). The more of these factors take on values associated with less satisfying algorithmic performance (viz., a higher number of clusters, a higher amount of noise, the presence of profile correlation and more variables than objects) an increasing algorithmic effort may be required, which can be done by taking a larger amount of starts in the multistart procedures. The question then arises how many such starts would be desirable. To illustrate how an answer to this question could be obtained we reexamined the data sets in one cell of the simulation design that was characterized by problematic values for the two factors with the largest impact on algorithmic performance (namely number of clusters and amount of noise which were equal to 5 clusters and 40% of error, respectively) and with non-problematic values for the other factors (viz., no profile correlation, data size 64 × 16, 75% of cluster overlap, equal cluster sizes and no error correlation). Although, in this cell, the proxy was found for each data set by at least one algorithm, the performance of each algorithm separately was far from satisfactory, as can be seen in the second column of Table 3. We therefore subjected each data set to 50 additional multistart runs (with for each algorithm and each run the same number of starts as in the simulation study). The percentage of data sets for which the proxy was found during the whole of the 51

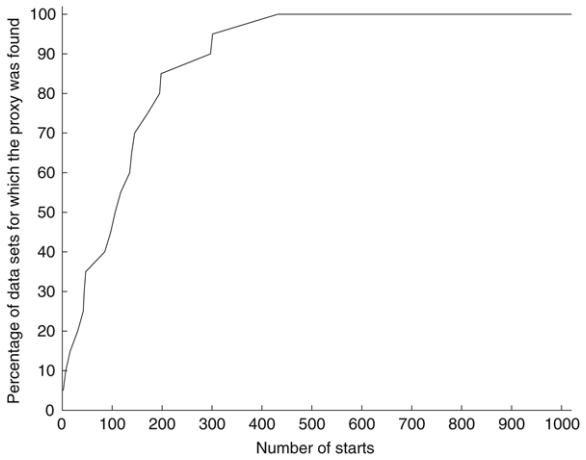


Fig. 4. Percentage of data sets on which the proxy is found as a function of the number starts for the $ALS_{lf_1}^{random}$ algorithm (data sets of size 64×16, with 5 clusters, 75% of cluster overlap, equal cluster sizes, 40% of error, no error correlation and no profile correlation)

multistarted runs can be found in the rightmost column of Table 3. Two algorithms now show a satisfactory performance: simulated annealing (SA_{lf_1}) and randomly started ALS ($ALS_{lf_1}^{random}$). For the latter algorithm we further examined more in detail how many starts were at least needed to find the proxy on all data sets. For this purpose we plotted the percentage of data sets for which the proxy was found as a function of the number of multistarts (Fig. 4). As shown in this Figure, after 432 starts the proxy was found for all data sets.

A second guideline pertains to the factor involved in a sizeable interaction with algorithmic differences, namely the amount of cluster overlap. Our simulation results imply that in case of a large amount of cluster overlap random starts are to be preferred as a seeding strategy, whereas for a low amount, data based strategies are superior. When analyzing a data set in practice, however, the amount of cluster overlap is in general unknown. As a way out, we propose to use a hybrid starting strategy based on a number of random starts and a number of data based starts, the best result of which is finally reported. Such a hybrid strategy appears very effective indeed: when retaining in the simulation study for each data set the best result of the first 10 randomly and the first 10 data based started ALS_{lf_1} runs (respectively the best results of the first 750 randomly and first 750 data based started ALS_{lf_2} runs), it was found that for 85% (respectively 80%) of the data sets the proxy was reached. As a consequence, in practice, the use of the ALS_{lf_1} algorithm with a hybrid starting strategy can be recommended. In the simulation study, the average recoveries for this hybrid algorithm are 94.67, 68.68 and 75.18 for the cluster, profiles and model values respectively. Optimally, one could also add one more start in this hybrid ALS_{lf_1} starting strategy based on the memberships as yielded by PCL on the column centered data; in the simulation study this start was found to lead to the proxy on 151 (0.70%) data sets while no other start did, which can be be considered as a slight improvement at a very low computational cost only. Moreover, as a comparison, taking either the eleventh random start or the elevent data based start in the hybrid ALS_{lf_1} starting strategy, yields an extra of only 147 (0.68%) or 86 (0.40%) data sets respectively on which the proxy was found.

6. Concluding remarks

Mirkin's (1987) one-mode additive clustering model is a very flexible model that may be especially useful when a mechanism is hypothesized to underly data under study that implies discrete structures that combine in an additive way. Examples of such mechanisms include syndromes in medicine and sources of states such as frustration or stress in psychology. In the present paper, this was illustrated by fitting the one-mode additive clustering model to situations by emotions data which successfully revealed latent triggers of anger and anger components.

In this paper we focused on algorithms for fitting the one-mode additive clustering model on a given data set in terms of minimizing a least squares loss function. Several new algorithms to minimize this loss function were proposed and examined in a simulation study. For the data-analytic practice this resulted in a proposal of one overall strategy based on a hybrid starting procedure. This strategy was shown to be very efficient in terms of both optimization and recovery performance.

In the present paper, we focused on one-mode additive clustering for a two-way two-mode data set. Given such data, however, one might wish to cluster the second unclustered mode as well. For this purpose one could consider the use of a so-called two-mode additive clustering model. Several such models are available along with several associated algorithmic strategies (see Van Mechelen et al. (2004) for an overview). However, not much is known about the performance of the algorithms in question. An interesting direction for further research would be to examine both the existing algorithms as well as extensions of the algorithmic strategies as proposed in the present paper in terms of optimization and recovery

performance. At this point one may for instance wish to know whether the results as reported in the present paper would generalize to the two-mode additive clustering case.

Acknowledgement

The research in this paper was supported by IAP P6/03, awarded to Iven Van Mechelen.

Appendix. Pseudocodes

A.1. PCL pseudocode

Algorithm 1 Principal cluster analysis: Pseudocode

Input: Data matrix $\mathbf{X}$, number of clusters K.
$m = 1$; $x_{ij}^m = x_{ij}$;
while $m \leq K$ **do**
  $\mathbf{a}_m = (0, \ldots, 0)'$; {length I}
  $\mathbf{p}_m = (0, \ldots, 0)'$; {length J}
  Calculate loss value l^2 as in (7).
  do
    $l_{old}^2 = l^2$; $\mathbf{a}_m^{old} = \mathbf{a}_m$; $\mathbf{p}_m^{old} = \mathbf{p}_m$; {Values from the previous step}
    for all i such that $\mathbf{a}_m(i) = 0$ **do**
      $\mathbf{a}_m^i = \mathbf{a}_m$; $\mathbf{a}_m^i(i) = 1$;
      Calculate $\mathbf{p}_m^i$ as in (8) with memberships $\mathbf{a}_m^i$.
      Calculate loss value l_i^2 as in (7) with $\mathbf{a}_m^i$ and $\mathbf{p}_m^i$.
    end for
    $i^* = \text{argmin}_i l_i^2$;
    $l^2 = l_{i^*}^2$;
    $\mathbf{a}_m = \mathbf{a}_m^{i^*}$; $\mathbf{p}_m = \mathbf{p}_m^{i^*}$;
  while $l^2 < l_{old}^2$ and $\mathbf{a}_m \neq (1, \ldots, 1)'$
  $\mathbf{a}_m = \mathbf{a}_m^{old}$; $\mathbf{p}_m = \mathbf{p}_m^{old}$;
  $x_{ij}^{m+1} = x_{ij}^m - \mathbf{a}_m \mathbf{p}_m$;
  $m = m + 1$;
end while
Output: Matrix $\mathbf{A}$ with vectors $\mathbf{a}_m$ in its columns.
Output: Matrix $\mathbf{P}$ with vectors $\mathbf{p}_m$ in its rows.

A.2. ALS_{lf_2} Pseudocode

Algorithm 2 ALS_{lf_2} pseudocode (seeded with one membership matrix)

Input: Data matrix $\mathbf{X}$, number of clusters K.
Initialize $\mathbf{A}_0 = \mathbf{A}_1$: randomly drawn or user specified.
$\mathbf{P}_0 = \mathbf{P}_1 = (\mathbf{A}_0'\mathbf{A}_0)^{-1}\mathbf{A}_0'\mathbf{X}$;
$lf_2^0 = lf_2^1 = \|\mathbf{X} - \mathbf{A}_0\mathbf{P}_0\|_F^2$;
do
  $\mathbf{A}_0 = \mathbf{A}_1$; $\mathbf{P}_0 = \mathbf{P}_1$; $lf_2^0 = lf_2^1$;
  for $i = 1, \ldots, I$ **do**
    Estimate row i of $\mathbf{A}_1$ by minimizing enumeratively
    $\sum_j (x_{ij} - \sum_{k=1}^K a_{ik} p_{kj}^0)^2$
    over all possible 0/1 combinations for a_{ik}, $k = 1, \ldots, K$.
  end for
  $\mathbf{P}_1 = (\mathbf{A}_1'\mathbf{A}_1)^{-1}\mathbf{A}_1'\mathbf{X}$;
  $lf_2^1 = \|\mathbf{X} - \mathbf{A}_1\mathbf{P}_1\|_F^2$;
while $lf_2^1 < lf_2^0$
Output: $\mathbf{A}_0$, $\mathbf{P}_0$.

A.3. ALS_{lf_1} Pseudocode

Algorithm 3 ALS_{lf_1} pseudocode (seeded with one membership matrix)

Input: Data matrix $\mathbf{X}$, number of clusters K.
Initialize $\mathbf{A}_0 = \mathbf{A}_1$: randomly drawn of user specified.
$\mathbf{P}_0 = \mathbf{P}_1 = (\mathbf{A}_0'\mathbf{A}_0)^{-1}\mathbf{A}_0'X$;
$lf_1^0 = lf_1^1 = \|\mathbf{X} - \mathbf{A}_0\mathbf{P}_0\|_F^2$;
do
    $\mathbf{A}_0 = \mathbf{A}_1$; $\mathbf{P}_0 = \mathbf{P}_1$; $lf_1^0 = lf_1^1$;
    for $i = 1, \ldots, I$ **do**
        for $j = 0, \ldots, 2^K - 1$ **do**
            Convert j into a binary (column) vector $\mathbf{b}$ of length K.
            $\mathbf{A}^j = \mathbf{A}_1$;
            $\mathbf{A}^j(i, :) = \mathbf{b}'$; {Row i of $\mathbf{A}^j$}
            Calculate loss $lf_{1;j} = lf(\mathbf{A}^j)$ as in (5).
        end for
        $j^* = \text{argmin}_j lf_{1;j}$;
        $\mathbf{A}_1 = \mathbf{A}^{j^*}$;
    end for
    $\mathbf{P}_1 = (\mathbf{A}_1'\mathbf{A}_1)^{-1}\mathbf{A}_1'X$;
    $lf_1^1 = \|\mathbf{X} - \mathbf{A}_1\mathbf{P}_1\|_F^2$;
while $lf_1^1 < lf_1^0$

Output: $\mathbf{A}_0$, $\mathbf{P}_0$.

A.4. Simulated annealing pseudocode

Algorithm 4 SA_{lf_1} pseudocode (seeded with one random membership matrix A_0)

Input: Data matrix $\mathbf{X}$, number of clusters K.
Randomly draw initial membership matrix $\mathbf{A}_0$
Calculate loss loss value $lf_1^0 = lf(\mathbf{A}_0)$ as in (5).
Calculate initial temperature T starting from $\mathbf{A}_0$.
Initialize $\mathbf{A}_b = \mathbf{A}_0$; $lf_1^b = lf_2^0$; {Best solution}
Initialize $lf_1^p = lf_2^0$; {Solution of previous chain}
$T_f = 10^{-5}$; $eqCh = 1$;
while $T > T_f$ and $eqCh < 10$ **do**
    $acc = 1$; $sim = 1$;
    while $sim \leq I2^K$ AND $acc \leq I2^K/10$ **do**
        Draw a neighbour $\mathbf{A}_1$ of $\mathbf{A}_0$.
        Calculate loss value $lf_1^1 = lf(\mathbf{A}_1)$ as in (5).
        if $lf_1^1 < lf_1^0$ **then**
            $\mathbf{A}_0 = \mathbf{A}_1$; $lf_1^0 = lf_1^1$; $acc = acc + 1$;
            if $lf_1^1 < lf_1^b$ **then**
                $\mathbf{A}_b = \mathbf{A}_1$; $lf_1^b = lf_1^1$;
            end if
        else
            Draw p_a uniformly random on the interval (0,1).
            if $p_a < \exp((lf_1^0 - lf_1^1)/T)$ **then**
                $\mathbf{A}_0 = \mathbf{A}_1$; $lf_1^0 = lf_1^1$; $acc = acc + 1$;
            end if
        end if
        $sim = sim + 1$;
    end while
    $T = \alpha T$;

if $lf_1^0 \neq lf_1^p$ **then**
  $lf_1^p = lf_1^0$;
  $eqCh = 1$;
else
  $eqCh = eqCh + 1$;
end if
end while
Calculate $\mathbf{P}_b = (\mathbf{A}_b'\mathbf{A}_b)^{-1}\mathbf{A}_b'X$.
Output: $\mathbf{A}_b$, $\mathbf{P}_b$.

References

Aarts, E.H.L., Korst, J.H.M., van Laarhoven, P.J.M., 1997. Simulated annealing. In: Aarts, E. H.L., Lenstra, J.K. (Eds.), Local Search in Combinatorial Optimization. Wiley, Chichester, UK, pp. 91–120.

Chaturvedi, A., Carroll, J.D., 1994. An alternating combinatorial optimization approach to fitting the INDCLUS and generalized INDCLUS models. J. Classif. 11, 155–170.

Kuppens, P., Van Mechelen, I., 2007. Determinants of the anger appraisals of threatened self-esteem, other-blame, and frustration. Cogn. Emot. 21, 56–77.

Ming-Tso Chiang, M., Mirkin, B.G., 2006. Determining the number of clusters in the straight *k*-means: Experimental comparison of eight options. In: Wang, X., Li, R. (Eds.), Proceedings of the UK Workshop on Computational Intelligence 2006. University of Leeds, Leeds, UK, pp. 119–126.

Mirkin, B.G., 1987. The method of principal clusters. Autom. Remote Control 10, 131–143.

Mirkin, B.G., 1990. A sequential fitting procedure for linear data analysis models. J. Classif. 7, 167–195.

Shepard, R.N., Arabie, P., 1979. Additive clustering representations of similarities as combinations of discrete overlapping properties. Psychol. Rev. 86 (2), 87–123.

Smith, C.A., Lazarus, K.A., 1993. Appraisal components, core relational themes, and the emotions. Cogn. Emot. 7, 151–162.

Talwar, S., Viber, M., Paulraj, A., 1996. Blind separation of synchronous co-channel digital signals using an antenna array, part I: Algorithms. IEEE Trans. Signal Process. 44 (5), 1184–1197.

Tversky, A., 1977. Features of similarity. Psychol. Rev. 84, 327–352.

Van Mechelen, I., Bock, H.-H., De Boeck, P., 2004. Two-mode clustering methods: A structured overview. Stat. Methods Med. Res. 13, 363–394.

Van Mechelen, I., Schepers, J., 2007. A unifying model involving a categorical and/or dimensional reduction for multimode data. Comput. Stat. Data Anal. 52 (1), 537–549.