Chemometrics and Intelligent Laboratory Systems 129 (2013) 87–97

Contents lists available at ScienceDirect

Chemometrics and Intelligent Laboratory Systems

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

Clusterwise Parafac to identify heterogeneity in three-way data

Tom F. Wilderjans *, Eva Ceulemans [1]

Methodology of Educational Sciences Research Group, Faculty of Psychology and Educational Sciences, KU Leuven, Andreas Vesaliusstraat 2, Box 3762, 3000 Leuven, Belgium

ARTICLE INFO

Article history:
Received 24 December 2012
Received in revised form 19 August 2013
Accepted 20 September 2013
Available online 18 October 2013

Keywords:
Parafac
Candecomp
Clusterwise analysis
Sensory profiling data
Population heterogeneity
Three-way data
Qualitative (and quantitative) differences between elements

ABSTRACT

In many research areas, the *Parafac* model is adopted to disclose the underlying structure of three-way three-mode data. In this model, a set of latent variables, called components, that captures the complex interaction between the elements of the three modes is sought. An important assumption of this model is that these components are the same for all elements of the three modes. In many cases, however, it makes sense to assume that the components may differ (i.e., qualitative differences in underlying component structure) across groups of elements of one of the modes. Therefore, in this paper, we present *Clusterwise Parafac*. In this new model, the elements of one of the three modes are assigned to a limited number of mutually exclusive clusters and, simultaneously, the data within each cluster are modeled with *Parafac*. As such, elements that belong to the same cluster are assumed to be governed by the same components, whereas elements that are assigned to different clusters have a different underlying component structure. To evaluate the performance of the new *Clusterwise Parafac* strategy, an extensive simulation study is conducted. Moreover, the strategy is applied to sensory profiling data regarding different cheeses.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

In many research areas, the *Parafac* model [1–3] is used to summarize the covariation in three-way three-mode data. One such research domain is sensory profiling, where one often lets a number of panelists rate a set of food samples (e.g., different cheeses) on a number of sensory attributes [4]. In chemometrics, this model is a popular method to explore the structure of fluorescence spectroscopy data [see, 5–12] or of chromatography results [see, a.o., 13–15]. Still other application areas include mass spectrometry [16,17], second order calibration [18,19], and (on-line) batch process monitoring [20].

In *Parafac*, components (i.e., latent variables) are sought that capture the complex interaction between the elements of the three modes. In the sensory profiling example, the components represent the most important dimensions for distinguishing between the cheese samples, for example, the taste and the texture. Each element of each mode receives a score on these components. The attribute scores reflect to which extent the attributes measure the underlying dimensions, whereas the sample scores indicate the location of each sample on these dimensions (e.g., the texture of a particular sample). The panelist scores represent the extent to which the panelists rely on a particular dimension when rating the samples (i.e., saliency).

An important assumption of this model is that the components are the same for all elements of the three modes. In many cases, however, it makes sense to assume that the underlying dimensions may differ across the elements of one of the modes [i.e., qualitative differences in underlying component structure may exist between elements of one mode; for a discussion of qualitative and quantitative differences in three-way data, see 21]. For the sensory profiling example, it is reasonable to assume that the dimensions that are important for capturing the quality of French soft cheeses differ from those used for evaluating other cheese types, such as blue cheeses. To complicate things further, it is often not known beforehand which elements are scored in a similar way and thus have a similar component structure and which are not. In order to account for such qualitative differences between the elements of a particular mode, a data analysis technique is needed that induces which elements are similar in that their ratings can be summarized by means of the same components, and which elements are not.

Therefore, we will introduce a new modeling strategy, called *Clusterwise Parafac*. In this new model, the elements of one of the three modes are assigned to a limited number of mutually exclusive clusters and, simultaneously, the data within each cluster are modeled with *Parafac*. Hence, qualitative differences between elements are modeled by assigning elements that have a similar component structure to the same cluster and elements with different structures to different clusters.

The remainder of this paper is organised in five main sections: In Section 2, the *Parafac* model is recapitulated and the new *Clusterwise Parafac* model is presented; finally, its relations to existing cluster models for three-way data are discussed. In Section 3, we will discuss an alternating least squares algorithm to estimate the parameters of the new model, along with a model selection procedure. In Section 4, we will present an extensive simulation study to evaluate the performance

* Corresponding author. Tel.: +32 16 32 61 23, +32 16 32 62 01; fax: +32 16 32 62 00.
E-mail addresses: tom.wilderjans@ppw.kuleuven.be (T.F. Wilderjans), eva.ceulemans@ppw.kuleuven.be (E. Ceulemans).
[1] Tel.: +32 16 32 58 81, +32 16 32 62 01; fax: +32 16 32 62 00.

0169-7439/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.chemolab.2013.09.010

of the *Clusterwise Parafac* algorithm in terms of optimizing the loss function and recovering the underlying parameters. In Section 5, an illustrative application of the new model to sensory profiling data of cheese samples will be discussed. Finally, in Section 6, we will conclude with a general discussion.

2. Model

2.1. The Parafac model for three-way three-mode data

In *Parafac* [1–3], the $I \times J \times K$ variable by source by object model array $\underline{\mathbf{M}}$ is fitted to the $I \times J \times K$ data array $\underline{\mathbf{X}}$. The model array $\underline{\mathbf{M}}$ is further decomposed into an $I \times R$ variable component matrix $\mathbf{A}$, a $J \times R$ source component matrix $\mathbf{B}$, and a $K \times R$ object component matrix $\mathbf{C}$. In particular, the model matrix $\mathbf{M}_k$ (of size $I \times J$) of object k ($k = 1 \ldots K$) is decomposed as follows:

$$\mathbf{M}_k = \mathbf{A}\mathbf{D}_k\mathbf{B}^T \tag{1}$$

where $\mathbf{D}_k$ is a $R \times R$ diagonal matrix that has the component scores for object k (i.e., the k-th row of $\mathbf{C}$) on its diagonal ($k = 1 \ldots K$), T denotes matrix transposition, and R denotes the number of components. As $\mathbf{A}$ and $\mathbf{B}$ have no index k, they are assumed to be the same for all objects. Under mild conditions, the *Parafac* decomposition is unique, except for a permutation and a scaling of the components [22–26].

2.2. The Clusterwise Parafac model

In order to allow clusters of objects[2] to be governed by (qualitatively) different components, *Clusterwise Parafac* partitions the K objects into Q mutually exclusive and non-empty clusters. Objects that belong to the same cluster have the same underlying components, whereas objects that belong to different clusters are assumed to have a different component structure. Specifically, in *Clusterwise Parafac* the model matrix $\mathbf{M}_k$ for object k (that belongs to cluster q) is decomposed as follows:

$$\mathbf{M}_k = \sum_{q=1}^{Q} p_{kq}\left(\mathbf{A}_q\mathbf{D}_k\mathbf{B}_q^T\right) \tag{2}$$

where Q is the number of clusters, p_{kq} are the entries of the binary partition matrix $\mathbf{P}$ ($K \times Q$) which indicates whether object k belongs to cluster q ($p_{kq} = 1$) or not ($p_{kq} = 0$), $\mathbf{A}_q$ ($I \times R$) is the variable component matrix and $\mathbf{B}_q$ ($J \times R$) the source component matrix for cluster q ($q = 1 \ldots Q$). Note that the number of components R is assumed to be equal for all objects, which in some cases may be too restrictive (for a further discussion of this topic, we refer to Section 6).[3]

From the *Clusterwise Parafac* decomposition rule in Eq. (2), it can be seen that this model is a generic modeling strategy that encompasses different models as special cases, two of which will be outlined here. First, when all objects are assigned to the same cluster (i.e., Q equals one), the standard *Parafac* model is obtained. A second special case is encountered when all objects are assigned to a different cluster (i.e., Q equals K), which implies that each object is modeled by a separate set of components. In this case, the new model boils down to performing a separate *PCA* to each $I \times J$ data matrix $\mathbf{X}_k$, which consists of the data slice of $\underline{\mathbf{X}}$ that pertains to object k.

Note that a *Clusterwise Parafac* model with Q clusters and R components can also be conceived as a constrained *Parafac* model with $Q \times R$ components. Specifically, in the latter model, the component matrix for the objects is constrained to have zero entries for all the components that are not associated to the cluster to which the object in question belongs.

2.3. Relations with other models

In this section, we describe the differences and similarities between *Clusterwise Parafac* and other models for three-way data that combine clustering with dimension reduction. Within these models, based on the work of [28], a distinction can be drawn between models that use dimension reduction to shed light on between-cluster differences in mean level (i.e., the components model the differences in the mean profiles of the different clusters) and models that use components to gain insight into between-cluster differences in covariation (i.e., the covariation within the clusters is described through cluster-specific components).

As the proposed model clusters the elements of one mode and models the data in different clusters with separate *Parafac* models, our new model clearly belongs to the second type of models. To the best of our knowledge, only two other deterministic models belong to this category. The first one is the *ParaFac with Optimally Clustered Variables* (*PFOCV*, [29]) model. In *PFOCV*, an optimal partitioning of the elements of one mode is identified, together with the *Parafac* components that optimally correspond to the resulting clusters: Each *Parafac* component corresponds to one cluster only, implying that the number of *PFOCV* components equals the number of clusters.[4] It can be concluded that *Clusterwise Parafac* is a generalization of *PFOCV* in that the cluster-specific *Parafac* models may consist of more than one component (and that the number of components per cluster needs to be determined during the analysis). The second model is *Clusterwise SCA* of which different variants have been proposed[5] (see [30,27,31,32]). Like *Clusterwise Parafac*, the latter model clusters the elements of one mode. Within each cluster, however, *Clusterwise SCA* models the data with *Simultaneous Component Analysis* (*SCA*; [33–41]), whereas *Clusterwise Parafac* uses *Parafac* instead; the former model is a more flexible model than the latter one because within a cluster, *Clusterwise Parafac* restricts the component scores to be equal across the elements of the clustered mode whereas *Clusterwise SCA* allows the component scores to differ across all elements of the clustered mode.

Deterministic models that focus on between-cluster differences in mean level were proposed by [42,43]. In these models, which can be conceived as three-way generalizations of Reduced K-means [44], also called Projection Pursuit Clustering [45,46], or of Factorial K-means [47,48], a clustering of the elements of one mode is combined with a dimensional reduction of the elements of the other modes. The main difference between these models and the models that focus on between-cluster differences in covariation is that in the first the clustering determines which elements of the clustered mode have identical scores on all extracted components, whereas in the latter the clustering indicates which subset of the components is relevant for summarizing the data of a particular element of the clustered elements (while allowing the scores on these components to differ across elements of the same cluster).

[2] Without loss of generality, the elements of the object mode (i.e., objects) will be clustered.

[3] However, allowing the number of components to vary across clusters implies a non-trivial model extension. In particular, using an *ALS* algorithm to fit the loss function (Eq. (3)) to the model in Eq. (1) with a varying number of components may not give the required result. In this regard, [27] show that an *ALS* algorithm for a *Clusterwise SCA* model with cluster-specific R values has the tendency to erroneously assign most of the data blocks to the cluster(s) with the largest number of components as this yields a larger increase in model fit (i.e., more components yield more modeling freedom). This phenomenon especially occurs when the clusters are relatively difficult to distinguish, which, among others, may be caused by a high amount of noise variance. Moreover, allowing the number of components to vary across clusters also complicates model selection as the optimal number of components needs to be determined for each cluster separately.

[4] Note that [29] also present the *ParaFac with Clustered Variables* (*PFCV*) model in which the clustering under consideration is given beforehand.

[5] Note that *Clusterwise SCA* was proposed to model multivariate multiblock data and that three-way data can be conceived as a special case of this type of data [21].

3. Data analysis

3.1. Aim

The aim of a *Clusterwise Parafac* analysis with Q clusters and R components of a data array $\underline{\mathbf{X}}$ is to estimate $\mathbf{P}$, $\mathbf{A}_q$, $\mathbf{B}_q$, and $\mathbf{D}_k$ ($q = 1 \ldots Q$ and $k = 1 \ldots K$) such that (1) the value of the loss function

$$f = \sum_{k=1}^{K} \|\mathbf{X}_k - \mathbf{M}_k\|_F^2, \quad f = \sum_{k=1}^{K} \|\mathbf{X}_k - \sum_{q=1}^{Q} p_{kq}\left(\mathbf{A}_q \mathbf{D}_k \mathbf{B}_q^T\right)\|_F^2 \tag{3}$$

is minimized, with $\|\ldots\|_F$ denoting the Frobenius norm (i.e., the square root of the sum of squared values) and (2) the model array $\underline{\mathbf{M}}$ can be perfectly represented by a *Clusterwise Parafac* model with Q clusters and R components.

3.2. Algorithm

In order to fit a *Clusterwise Parafac* model with Q clusters and R components to a given $I \times J \times K$ data array $\underline{\mathbf{X}}$ at hand, the loss function in (Eq. (3)) is minimized. To this end, an alternating least squares (*ALS*) algorithm will be adopted (for an *ALS* algorithm for the original *Parafac* model, see [49]). Because an *ALS* algorithm may end up in a local rather than the global optimum, it is advised to complement the algorithm with a multi-start procedure, along with a smart way to obtain initial parameter estimates that are already of a "high quality".

3.2.1. ALS algorithm

Because there exist many different ways of assigning the K objects to Q clusters [i.e., a partition problem is NP-hard, see, e.g., 50,51], determining the global optimum of the *Clusterwise Parafac* loss function in Eq. (3) is not an easy task, except in those cases where K is small (i.e., $K<20$). To this end, we will use an *ALS* procedure [for more information on *ALS*, see 52,53] that alternates between updating the cluster memberships of the objects (in partition matrix $\mathbf{P}$) and re-estimating the (cluster-specific) *Parafac* parameters (i.e., $\mathbf{A}_q$, $\mathbf{B}_q$, and $\mathbf{C}$), until none of these updates result in an improvement in the loss function value. In particular, the *Clusterwise Parafac ALS* algorithm consists of the following four steps:

1. Generate an initial partition matrix $\mathbf{P}$ by assigning the K objects to one of the Q clusters, without allowing clusters to be empty.
2. Estimate the object component matrix $\mathbf{C}$ and the cluster-specific variable and source component matrices $\mathbf{A}_q$ and $\mathbf{B}_q$. To this end, perform a separate *Parafac* on each cluster-specific data matrix $\underline{\mathbf{X}}_q$ (of size $I \times J \times K_q$) that is obtained by only retaining the data slices of $\underline{\mathbf{X}}$ of the objects that belong to cluster q. To estimate these cluster-specific *Parafac* models, an *ALS* algorithm is used [49] in which the component matrices $\mathbf{A}_q$, $\mathbf{B}_q$, and the relevant part of $\mathbf{C}$ (i.e., only the component scores of the objects that belong to cluster q are updated) are alternatingly re-estimated conditionally upon the other component matrices until no improvement in the loss function is observed [see 54–56].[6] At the end, compute the loss function value (Eq. (3)).
3. Update the partition matrix $\mathbf{P}$ row-wise (i.e., object by object) and re-estimate $\mathbf{C}$, $\mathbf{A}_q$, and $\mathbf{B}_q$ ($q = 1, \ldots, Q$) as in step 2. To determine the optimal cluster membership of object k, for each cluster q, the optimal object component scores for object k given $\mathbf{A}_q$ and $\mathbf{B}_q$ are computed by means of linear regression (for more information, see [20]). Next, for each cluster q, the partition criterion $L_{kq} = \|\mathbf{X}_k - \mathbf{A}_q \mathbf{D}_k^q \mathbf{B}_q^T\|_F^2$ is computed, with $\mathbf{D}_k^q$ being a diagonal matrix that contains on its diagonal the component scores for object k when assigned to cluster q. The value for the partition criterion L_{kq} denotes the extent to which object k does not fit in cluster q. Finally, object k is re-assigned to the cluster q for which L_{kq} is minimal (i.e., $p_k^* = arg\ min_q\ L_{kq}$, with p_k^* denoting the cluster to which object k is assigned to). After updating the cluster membership of each object, it may occur that one (or more) cluster(s) is empty, which may indicate that there are less clusters in the data than the pre-specified number Q. In the case where one wants to avoid empty clusters, one may optionally apply the following procedure (a similar procedure is also used to avoid empty clusters in k-means clustering, see [58]): (*a*) assign the object that fits its current cluster the worst to (one of) the empty cluster(s) and re-estimate the component matrices as in step 2, and (*b*) repeat the procedure in (*a*) until all clusters are non-empty. When adopting this procedure to avoid empty clusters in cases in which Q is specified too large, the worst fitting object(s) from the *Q-1* (*Q-2*, …) solution will be assigned to a separate cluster.[7]
4. Compute the loss function value (Eq. (3)). When it has improved (i.e., decreased), return to step 3, otherwise the algorithm has converged.

3.2.2. Multi-start procedure

Because the above described *ALS* algorithm may end up in a local rather than the global optimum, a multi-start procedure is adopted. This procedure consists of running the *Clusterwise Parafac ALS* algorithm with different (pseudo-) random and rational initializations of the partition matrix $\mathbf{P}$ [59,60]; the solution with the lowest value on loss function (Eq. (3)) is retained. Because many suboptimal solutions for the *Clusterwise Parafac* loss function may exist, determining a set of good ("high quality") initial object partitions is of utmost importance. In particular, V high quality object partitions can be generated by means of the following procedure:

1. Obtain a rational initial object partition matrix $\mathbf{P}^{rational}$ by, first, performing a *PCA* with R components on each $\mathbf{X}_k$ (resulting in a variable and a source component matrix $\mathbf{A}_k$ and $\mathbf{B}_k$ for each object k). Next, determine the similarity between each pair of obtained variable and source component matrices $\mathbf{A}_k$ and $\mathbf{B}_k$ ($k = 1 \ldots K$).[8] Finally, perform an (unweighted) average linkage hierarchical cluster analysis [62] on the $K \times K$ matrix of similarity coefficients, and obtain a (rational) object partition by cutting the resulting tree at Q clusters.[9]
2. Starting from $\mathbf{P}^{rational}$, generate $V \times 5$ different pseudo-random partition matrices $\mathbf{P}^{pseudo\text{-}random}$ (with no empty clusters) by re-assigning each object to another cluster with a probability equal to .15 (with all 'other' clusters having the same probability of being assigned to). Additionally, generate X random object partitions

[6] To minimize the risk of retaining a suboptimal solution for the (cluster-specific) *Parafac* parameters, a multi-start procedure is used in which the alternating *Parafac* algorithm is repeated many times, each time starting with other rationally or randomly determined parameter values, and the best encountered solution is selected as the final one (for more information, see [57]).

[7] In these cases, the following features of the obtained solution may indicate that Q is specified too large: (1) the obtained solution has a singleton cluster(s), and/or (2) for two (or more) clusters the obtained $\mathbf{A}_q$ and $\mathbf{B}_q$ component matrices are very similar. Note, however, that a singleton cluster will also occur when the *Parafac* components underlying a single object are very distinct from the components underlying the other (clusters of) objects.

[8] In particular, to determine the similarity between object k and object k', first, $\mathbf{A}_k$ ($\mathbf{B}_k$) is rotated obliquely towards $\mathbf{A}_{k'}$ ($\mathbf{B}_{k'}$) in a least squares sense, resulting in $\mathbf{A}_{k'}^{rotated}$ ($\mathbf{B}_{k'}^{rotated}$). Next, for each associated component in $\mathbf{A}_k$ and $\mathbf{A}_{k'}^{rotated}$ ($\mathbf{B}_k$ and $\mathbf{B}_{k'}^{rotated}$) the Tucker congruence coefficient [61] is calculated and the average of all these coefficients is computed.

[9] Average linkage hierarchical clustering was adopted because (1) it is a standard method for clustering objects on the basis of similarities, (2) it consistently performs among the best when comparing different hierarchical clustering methods [63–69], (3) solutions for all possible numbers of clusters are obtained at once (i.e., by cutting the tree at different places), and (4) it is computationally fast. Regarding the latter, note that average linkage hierarchical clustering is often used to obtain in a fast way a good initial start for other clustering procedures, like k-means [70,65].

$\mathbf{P}^{random}$ (with no empty clusters), with each object having the same probability of being assigned to each cluster.

3. Estimate for $\mathbf{P}^{rational}$, each pseudo-random $\mathbf{P}^{pseudo\text{-}random}$, and each random $\mathbf{P}^{random}$, the associated *Clusterwise Parafac* model parameters (as in step 2, see above) and calculate the resulting loss function value (Eq. (3)) for each starting solution. Next, sort all $1 + (V \times 5) + X$ solutions according to their loss function value and retain the best V solutions.

The *Clusterwise Parafac* algorithm has been implemented in MATLAB-code (version 8) and is available upon request from the first author. For estimating the cluster-specific *Parafac* models, code from the *N*-way Toolbox in MATLAB [57] has been used.

3.3. Model selection

Usually the optimal number of clusters Q and the optimal number of components R of a *Clusterwise Parafac* model underlying a data array at hand are unknown in advance. As a way out, different *Clusterwise Parafac* analyses with increasing numbers of clusters q (i.e., $q = 1, \ldots, Q^{max}$) and components r (i.e., $r = 1, \ldots, R^{max}$) are conducted. An optimal model is selected by looking for the model with the best balance between, on the one hand, fit to the data (i.e., the loss function value), and, on the other hand, the complexity of the model (i.e., the number of clusters and components). To this end, different model selection strategies may be used, like *CORCONDIA* [71] and *CHull*, a numerical convex hull based method [72–74].

Comparing different model selection strategies (a.o., *CHull*), [27] shows that a specific sequential method clearly outperforms the others. This sequential method, which may be conceived as a generalization of the well-known scree test [75], consists of the following two-step procedure [31]: (1) determine the optimal number of clusters Q, and (2) given the selected optimal number of clusters Q, identify the optimal number of components R (for a similar model selection approach in the context of *Clusterwise HICLAS*, see [76]). In particular, the first step of this procedure consists of computing for each value q ($q = 2, \ldots, Q^{max} - 1$), given different numbers of components r ($r = 1, \ldots, R^{max}$), the following scree ratio $sr_{q|r}$:

$$sr_{q|r} = \frac{SSQ_{q,r} - SSQ_{q-1,r}}{SSQ_{q+1,r} - SSQ_{q,r}}, \tag{4}$$

with $SSQ_{q,r}$ denoting the loss function value (Eq. (3)) for a *Clusterwise Parafac* model with q clusters and r components. Note that for the smallest (i.e., $q = 1$) and the largest (i.e., $q = Q^{max}$) number of clusters q, the scree ratio $sr_{q|r}$ is not defined (implying that the procedure cannot select these numbers of clusters). Next, for each value q ($q = 2, \ldots, Q^{max} - 1$), the mean scree ratio sr_q is calculated by averaging $sr_{q|r}$ over the different number of components (i.e., $sr_q = \frac{\sum_{r=1}^{R^{max}} sr_{q|r}}{R^{max}}$), and the optimal number of clusters Q is determined as the q for which sr_q is maximal.

In the second step of the procedure, the optimal number of components R is determined by calculating for different values of r ($r = 2, \ldots, R^{max} - 1$) the following scree ratio $sr_{r|Q}$, with Q being the optimal number of clusters as selected in the first step:

$$sr_{r|Q} = \frac{SSQ_{r,Q} - SSQ_{r-1,Q}}{SSQ_{r+1,Q} - SSQ_{r,Q}}. \tag{5}$$

Note that the smallest (i.e., $r = 1$) and largest (i.e., $r = R^{max}$) value of r cannot be selected. Next, the optimal number of components R is identified as the value r for which $sr_{r|Q}$ is maximal. It should be noted that the final decision regarding the model to be retained should also be based on the interpretability and stability of the *Clusterwise Parafac* solution.

4. Simulation study

4.1. Problem

In this section, we will present a simulation study to evaluate the performance of the *Clusterwise Parafac* algorithm. At this point, we are interested in five performance aspects: optimization, recovery, computation time, number of iterations performed, and degeneracy of the obtained solutions. With regard to optimization, we will examine how sensitive the algorithm is to local minima. Concerning recovery, we will determine the degree to which the algorithm succeeds in disclosing the true object partition and cluster-specific component structure underlying the data. Moreover, we will investigate whether and how both aspects depend on five data characteristics. The first three characteristics that will be investigated pertain to the partition of the objects: (1) the number of underlying clusters, (2) the cluster sizes, and (3) the degree of congruence (i.e., similarity) between the cluster-specific variable and source components. Regarding these three characteristics, we expect that algorithmic performance will deteriorate: (1) when the number of underlying clusters increases [77,78,30,27], (2) when the clusters are of different sizes [79,78,60,27], and/or (3) when there is a large amount of congruence between the cluster-specific component matrices [80,32]. The fourth manipulated factor is the number of underlying components in the cluster-specific *Parafac* models. Regarding this factor, which reflects the complexity of the underlying *Parafac* model(s), we expect that the algorithmic performance will decrease with an increasing number of components [81–83,30,27]. Finally, the fifth factor pertains to the amount of noise in the data, for which we hypothesize that algorithmic performance will decrease when the data become more noisy [77,83,30,27].

4.2. Design and procedure

In order to not have an overly complex design, in the simulation study, the size of the data set was kept constant. In particular, the number of variables was fixed at 15 ($I = 15$), the number of sources at 30 ($J = 30$), and the number of objects at 120 ($K = 120$). Further, the five factors that were introduced above were systematically manipulated in a completely randomized five-factorial design in which all factors were considered random:

1. the number of clusters, Q, at 2 levels: 2 and 4;
2. the cluster sizes, at 3 levels [see 78]: equal (objects are equally distributed across clusters); unequal with minority (one small cluster having 10% of the objects and the remaining objects being equally distributed over the other clusters); unequal with majority (one large cluster containing 70% of the objects and the remaining objects being equally distributed over the other clusters);
3. the degree of congruence between the cluster-specific component matrices $\mathbf{A}_q$ and $\mathbf{B}_q$ ($q = 1, \ldots, Q$). The congruence for $\mathbf{A}_q$ and $\mathbf{B}_q$ was manipulated independently at 2 levels (i.e., low congruence and moderate congruence), resulting in the following 4 levels for the congruence factor: (1) low congruence for $\mathbf{A}_q$ and low congruence for $\mathbf{B}_q$, (2) low congruence for $\mathbf{A}_q$ and moderate congruence for $\mathbf{B}_q$, (3) moderate congruence for $\mathbf{A}_q$ and low congruence for $\mathbf{B}_q$, and (4) moderate congruence for $\mathbf{A}_q$ and moderate congruence for $\mathbf{B}_q$;
4. the number of components of the cluster-specific *Parafac* models, R, at 2 levels: 2 and 4;
5. the amount of noise in the data, ε, at 5 levels: .10, .30, .50, .70, and .90.

For each combination of the levels of the five manipulated factors (i.e., cell of the design), 10 data sets $\underline{\mathbf{X}}$ were constructed by means of the following procedure: First, a true partition matrix $\mathbf{P}^{true}$ was constructed by computing the number of objects that belong to each cluster (given the number of clusters and the cluster sizes) and assigning at random the correct number of objects to each cluster. Next, a true object component matrix $\mathbf{C}^{true}$, and true cluster-specific

variable $\mathbf{A}_q^{true}$ and source $\mathbf{B}_q^{true}$ component matrices ($q = 1 \ldots Q$) were generated. The true object component matrix $\mathbf{C}^{true}$ was obtained by independently drawing entries from a uniform distribution from the interval [−1 1]. To manipulate the amount of congruence between the different $\mathbf{A}_q^{true}$ and $\mathbf{B}_q^{true}$ ($q = 1 \ldots Q$), first, a (common) base variable $\mathbf{A}^{base}$ and source $\mathbf{B}^{base}$ component matrix was generated by drawing entries from $U(-1,1)$. Next, matrices $\mathbf{A}_q^{temp}$ and $\mathbf{B}_q^{temp}$ were simulated by drawing entries from $U(-2.50, 2.50)$ or $U(-.60, .60)$[10], and $\mathbf{A}_q^{temp}$ and $\mathbf{B}_q^{temp}$ were added to $\mathbf{A}^{base}$ and $\mathbf{B}^{base}$, respectively (e.g., $\mathbf{A}_q^{true} = \mathbf{A}^{base} + \mathbf{A}_q^{temp}$). As a consequence, a set of $\mathbf{A}_q^{true}$ and $\mathbf{B}_q^{base}$ matrices ($q = 1 \ldots Q$) that are moderately and lowly congruent were obtained.[11] Next, true matrices $\mathbf{T}_k$ ($k = 1 \ldots K$) were computed by combining $\mathbf{P}^{true}$, $\mathbf{A}_q^{true}$, $\mathbf{B}_q^{true}$, and $\mathbf{C}^{true}$ by the *Clusterwise Parafac* decomposition rule in Eq. (2). Finally, for each true matrix $\mathbf{T}_k$, a data matrix $\mathbf{X}_k$ was constructed by adding a noise matrix $\mathbf{E}_k$ (of size $I \times J$) to $\mathbf{T}_k$. To manipulate the amount of noise in the data, each noise matrix $\mathbf{E}_k$ was constructed by, first, generating a noise matrix $\mathbf{E}_k^{start}$ (of size $I \times J$) by independently drawing numbers from $N(0,1)$, and, next, rescaling $\mathbf{E}_k^{start}$ to ensure $\frac{\sum_{k \in C_{q(k)}} \|\mathbf{E}_k\|_F^2}{\sum_{k \in C_{q(k)}} \|\mathbf{X}_k\|_F^2}$ being equal to ε for each cluster q, with $C_{q(k)}$ denoting the set of objects that belong to the same cluster to which object k belongs.

Fully crossing all design factors, 10 (replication) × 2 (number of clusters) × 3 (cluster sizes) × 4 (congruence) × 2 (number of components) × 5 (amount of noise) = 2400 different data arrays $\underline{\mathbf{X}}$ were obtained. Subsequently, a *Clusterwise Parafac* analysis with the true number of clusters Q (not allowing clusters to be empty) and the true number of components R was performed on each data array $\underline{\mathbf{X}}$ with 25 starts. The starts were obtained by selecting the best 25 initial partition matrices $\mathbf{P}$ among (1) a rationally determined $\mathbf{P}$, (2) 125 pseudo-random $\mathbf{P}$ (with the probability of re-allocating an object to another cluster being equal to .15), and (3) 25 random $\mathbf{P}$ (see Section 3.2). The simulation study was run on a supercomputer consisting of INTEL XEON L5420 processors with a clock frequency of 2.5 GHz and with 8 GB RAM.

4.3. Results

4.3.1. Optimization performance: sensitivity to local minima

The aim of this section is to investigate whether the *Clusterwise Parafac* algorithm is capable of identifying the global optimum of the loss function (Eq. (3)). From the moment, however, that noise is added to the data, the global optimum of the loss function is unknown. A way out consists of defining a proxy for the global optimum, which can be conceived as our best guess of the loss function value of the global optimal solution. We define the proxy as the lowest loss function value encountered among the following two *Clusterwise Parafac* solutions: (1) the true solution underlying the data (i.e., $\mathbf{P}^{true}$, $\mathbf{A}_q^{true}$, $\mathbf{B}_q^{true}$, and $\mathbf{C}^{true}$), because this true solution is always a valid solution with Q clusters and R components for the data at hand, and (2) the solution obtained by running the *Clusterwise Parafac* algorithm using the true clustering $\mathbf{P}^{true}$ as start. Note that in most cases the second option will lead to a tighter proxy (i.e., a lower loss function value) than the first one. We consider a *Clusterwise Parafac* solution as suboptimal when its loss value exceeds the proxy. This appeared to be the case for 578 out of the 2400 data sets (i.e., 24.08%). In Table 1, which presents the percentage of data sets for which a suboptimal solution is encountered for all levels of the five manipulated factors, one can see that local minima mostly occur in the conditions with four components, when there is a large amount of noise, and when there is a large cluster together with one (or more) small cluster(s).

4.3.2. Recovery performance

In this subsection, we will evaluate the recovery performance of the *Clusterwise Parafac* algorithm regarding (1) the object partition, and (2) the cluster-specific variable and source component matrices.

4.3.2.1. Recovery of the true partition of the objects. To determine the degree to which the underlying object partition has been recovered by the *Clusterwise Parafac* algorithm, we calculated the Adjusted Rand Index [*ARI*; 84] between the true object partition (i.e., $\mathbf{P}^{true}$) and the object partition retained by the algorithm (i.e., $\mathbf{P}$). An *ARI* value of 1 is encountered when both partitions are identical, whereas *ARI* equals 0 when recovery is at chance level.

Across all 2400 data sets, the mean *ARI* equals .9315 with a standard deviation of .1553. Moreover, a perfect recovery of the true object partition (i.e., $ARI = 1$) was encountered for 1545 out of the 2400 data sets (64.38%). This implies that the *Clusterwise Parafac* algorithm recovers the underlying object partition to a very large extent. To investigate whether the recovery of the true partition depends on the manipulated data characteristics, for all levels of the five manipulated factors, the mean *ARI* value is presented in Table 2. In this table, one can see that the cluster recovery decreases when the amount of noise increases, the number of clusters increases, the degree of congruence among cluster-specific component matrices increases, and when there is one small cluster with one (or more) large(r) cluster(s). Because there is only a limited variation in *ARI* values, no analysis of variance has been performed.

4.3.2.2. Recovery of the true cluster-specific variable and source component matrices. The extent to which the *Clusterwise Parafac* algorithm recovers the true cluster-specific variable and source component matrices $\mathbf{A}_q$ and $\mathbf{B}_q$ was determined by, first, calculating, for each cluster q, the Tucker congruence coefficient [61] θ_q^A (θ_q^B) between the true $\mathbf{A}_q^{true}$ ($\mathbf{B}_q^{true}$) and the estimated $\mathbf{A}_q$ ($\mathbf{B}_q$). To account for the within-cluster permutation freedom of the *Parafac* components, for each cluster q, that permutation

Table 1
Percentage of data sets for which a suboptimal solution was encountered (i.e., proxy miss) for all levels of the manipulated factors.

Factor	Levels	Percentage proxy miss
Number of clusters	2	22.75
	4	25.42
Cluster sizes	Equal sized clusters	24.00
	One small cluster	17.13
	One large cluster	31.13
Number of components	2	14.00
	4	34.17
Amount of noise in the data	.10	4.79
	.30	10.62
	.50	17.08
	.70	46.46
	.90	41.46
Degree of congruence between the cluster-specific component matrices	Low $\mathbf{B}_q$–low $\mathbf{C}_q$	24.67
	Low $\mathbf{B}_q$–moderate $\mathbf{C}_q$	23.67
	Moderate $\mathbf{B}_q$–low $\mathbf{C}_q$	22.67
	Moderate $\mathbf{B}_q$–moderate $\mathbf{C}_q$	25.33

[10] In a pilot study it was found that for the data characteristics used in the simulation study adopting a value of 2.50 for c yields a mean Tucker congruence [61] value around .15 (i.e., a low congruence), whereas a c-value of .60 results in a mean Tucker congruence value around .75 (i.e., a moderate congruence).

[11] To determine the amount of congruence between the resulting cluster-specific component matrices $\mathbf{A}_q^{true}$ and $\mathbf{B}_q^{true}$ at the four levels of the congruence factor, the Tucker congruence coefficient (with 1 indicating perfect congruence/similarity; see [61]) between the columns of each couple of $\mathbf{A}_q^{true}$ ($\mathbf{B}_q^{true}$) matrices was computed, after taking the permutation freedom of the *Parafac* components into account. When averaging this Tucker congruence coefficient over all components and all couples of cluster-specific variable (source) component matrices, the following mean congruence scores are obtained for the four levels of the congruence factor (for $\mathbf{A}_q$ and $\mathbf{B}_q$, respectively): (1) .1879 ($SD = .1072$) and .1401 ($SD = .0793$) for the low $\mathbf{A}_q$ congruence–low $\mathbf{B}_q$ congruence condition, (2) .1265 ($SD = .1201$) and .7328 ($SD = .0369$) for the low $\mathbf{A}_q$ congruence–moderate $\mathbf{B}_q$ congruence condition, (3) .7331 ($SD = .0561$) and .1379 ($SD = .0887$) for the moderate $\mathbf{A}_q$ congruence–low $\mathbf{B}_q$ congruence condition, and (4) .7287 ($SD = .0535$) and .7347 ($SD = .0374$) for the moderate $\mathbf{A}_q$ congruence–moderate $\mathbf{B}_q$ congruence condition.

Table 2
Mean *ARI* value (and standard deviation) and mean $\theta_{all}^{\mathbf{AB}}$ value for all levels of the manipulated factors.

Factor	Levels	Mean *ARI* (standard deviation of *ARI*)	Mean $\theta_{\mathbf{all}}^{\mathbf{AB}}$
Number of clusters	2	.9638 (.1107)	.9893
	4	.8991 (.1841)	.9532
Cluster sizes	Equal sized clusters	.9683 (.0829)	.9907
	One small cluster	.8855 (.1968)	.9465
	One large cluster	.9406 (.1524)	.9764
Number of components	2	.9221 (.1592)	.9770
	4	.9408 (.1507)	.9655
Amount of noise in the data	.10	.9836 (.0829)	.9929
	.30	.9632 (.1276)	.9852
	.50	.9592 (.1340)	.9824
	.70	.9444 (.1293)	.9770
	.90	.8070 (.2063)	.9186
Congruence between the cluster-specific component matrices	Low $\mathbf{B}_q$–low $\mathbf{C}_q$	.9366 (.1482)	.9651
	Low $\mathbf{B}_q$–moderate $\mathbf{C}_q$	.9383 (.1419)	.9733
	Moderate $\mathbf{B}_q$–low $\mathbf{C}_q$	.9379 (.1546)	.9720
	Moderate $\mathbf{B}_q$–moderate $\mathbf{C}_q$	.9131 (.1732)	.9744

of the components was selected that optimizes ${\theta_q}^{\mathrm{A}} + {\theta_q}^{\mathrm{B}}$. Next, an overall ${\theta_{all}}^{\mathrm{A}}$ (${\theta_{all}}^{\mathrm{B}}$)-statistic was obtained by averaging the cluster-specific ${\theta_q}^{\mathrm{A}}$ (${\theta_q}^{\mathrm{B}}$) across all clusters:

$$\theta_{all}^{\mathbf{A}} = \frac{\sum_{q=1}^{Q}\theta\left(\mathbf{A}_q^{true},\mathbf{A}_q\right)}{Q} = \frac{\sum_{q=1}^{Q}\theta_q^{\mathbf{A}}}{Q},$$
$$\theta_{all}^{\mathbf{B}} = \frac{\sum_{q=1}^{Q}\theta\left(\mathbf{B}_q^{true},\mathbf{B}_q\right)}{Q} = \frac{\sum_{q=1}^{Q}\theta_q^{\mathbf{B}}}{Q}. \quad (6)$$

The between-cluster permutation freedom was taken into account by trying all cluster permutations and selecting the cluster permutation that yields the largest value for $\theta_{all}^{\mathbf{AB}} = \frac{\theta_{all}^{\mathbf{A}}+\theta_{all}^{\mathbf{B}}}{2}$. A $\theta_{all}^{\mathbf{AB}}$ value of 1 indicates that the recovery of the true cluster-specific variable and source components is perfect, whereas a value of 0 implies that there is no recovery at all.

Across all data sets, the mean $\theta_{all}^{\mathbf{AB}}$ equals .9712 with a standard deviation of .0663, indicating that the *Clusterwise Parafac* algorithm recovers the cluster-specific variable and source components to a very large extent. In order to study how the recovery of $\mathbf{A}_q^{true}$ and $\mathbf{B}_q^{true}$ varies as a function of the manipulated data characteristics, the mean $\theta_{all}^{\mathbf{AB}}$ value for each level of the five manipulated factors is displayed in Table 2. From this table, although the differences are small, it appears that the recovery of the cluster-specific variable and source component matrices decreases when the optimization problem becomes more complex (i.e., a larger number of clusters and a larger amount of noise in the data) and when there is one small and one (or more) large(r) cluster(s) in the data. Because there was too little variation among the $\theta_{all}^{\mathbf{AB}}$ values (i.e., more than 86% of the values is larger than .95), no analysis of variance has been performed.

4.3.3. Computation time, number of performed iterations, and the occurrence of degenerate Parafac solutions

In the simulation study, the mean computation time for one simulated data set equals 3.23 h with a standard deviation of 2.41 h. To investigate how the computation time depends on the manipulated data characteristics, an analysis of variance was performed with the computation time as the dependent variable and the manipulated factors as independent variables. Only taking effects into account with an intraclass correlation $\hat{p}_I$ [85,86] larger than .10, it appears that the computation time increases with increasing numbers of clusters ($\hat{p}_I = .28$), amounts of noise in the data ($\hat{p}_I = .20$), and numbers of components ($\hat{p}_I = .14$). In particular, taking four instead of two clusters implies the computation time to double, whereas going from two to four components lengthens the computation time with 70%. For the amount of noise ε equaling .10, .30, .50, .70, and .90, the mean computation time equals 2.22, 2.23, 2.29, 3.60, and 5.81 h, respectively. These main effects, however, are qualified by a considerable interaction between the number of clusters and the amount of noise ($\hat{p}_I = .11$). In particular, as one can see in Table 3, the increase in computation time for higher number of clusters is more pronounced when the data contain larger amounts of noise.

The number of iterations that is performed by the algorithm is mainly influenced by the amount of noise ($\hat{p}_I = .22$) and the number of clusters ($\hat{p}_I = .10$). In particular, the algorithm iterates more when the amount of noise (i.e., on average, 2.89, 3.30, 3.86, 4.56, and 6.56 iterations for $\varepsilon = .10, .30, .50, .70$, and .90, respectively) and the number of clusters (i.e., on average, 3.54 and 4.93 iterations for 2 and 4 clusters, respectively) increase.

To determine whether some of the obtained cluster-specific *Parafac* models are degenerate (i.e., some components are proportional to each other), we based ourselves on the degeneracy criteria discussed by [87,88]: (1) the triple cosine product [89,29,90–92] is smaller than −.85, (2) the smallest eigenvalue of the triple-cosine matrix (i.e., the $R \times R$ matrix with the triple cosine products between the R components as entries) is smaller than .50, and (3) the condition number (i.e., the ratio of the largest eigenvalue to the smallest eigenvalue) of the triple-cosine matrix is larger than 5. More information on the degeneracy of a *Parafac* solution and ways to deal with this problem can be found in [2,93,24,94–97,49]. Applying these criteria in our simulation study reveals that almost all our obtained solutions are non-degenerate. In particular, only for 14 data sets (.58%) the minimal triple cosine product across the clusters is smaller than −.85. Further, only for 35 data sets (1.46%) is the smallest eigenvalue smaller than .50, whereas a condition number larger than 5 is only encountered for 31 data sets (1.29%). Note that these few degenerate solutions all belong to the simulation conditions with four clusters.

4.3.4. Discussion of the results

From the simulation results presented above, it appears that, in general, the *Clusterwise Parafac* algorithm performs well with respect to optimizing the loss function when using 25 "high quality" multi-starts (see Section 3.2), although local minima problems may be encountered when the data contain an excessive amount of noise (i.e., more than 70%). Therefore, when the optimization problem becomes more difficult (i.e., larger number of clusters/components and larger amounts of noise), we advise to increase the number of starts that is used throughout the algorithm. Moreover, the algorithm recovers both the true object partition and the true cluster-specific variable and source components to a very large extent. As such, it can be concluded that, when the optimal number of clusters and components is known, the *Clusterwise Parafac* algorithm with 25 starts delivers solutions of a good quality. One may wonder what would happen if we carry out the analysis with too few or too many clusters. In this regard, [32] show for *Clusterwise SCA* that when too few clusters are estimated, two true clusters get cleanly fused into a single cluster, whereas extracting too

Table 3
Mean computation time (and standard deviation) in hours for *Clusterwise Parafac* analyses (with 25 multi-starts, which are the best ones out of 151 initial partitionings) for different amounts of noise in the data and varying numbers of clusters.

Amount of noise ε	2 clusters	4 clusters	Overall
.10	1.55 (.41)	2.88 (.89)	2.22 (.96)
.30	1.56 (.39)	2.90 (.91)	2.23 (.97)
.50	1.57 (.39)	3.01 (.96)	2.29 (1.03)
.70	2.24 (.65)	4.97 (1.77)	3.60 (1.91)
.90	3.51 (1.29)	8.11 (3.65)	5.81 (3.58)
overall	2.09 (1.05)	4.37 (2.81)	3.23 (2.41)

many clusters results in one true cluster being split into two smaller clusters. Given the close relationship between both methods (see Section 2.3 and [21]), a similar result may be expected for *Clusterwise Parafac*.

5. Illustrative application

In this section, *Clusterwise Parafac* will be illustrated by re-analyzing sensory profiling data. The goal of analyzing such data is disclosing the experience dimensions that underlie the attribute ratings of a set of food samples by different panelists (i.e., QDA; Quantitative Descriptive Analysis with fixed vocabulary profiling). Herewith, the similarities and differences among the food samples can be uncovered by studying the sample scores on the dimensions [4]. However, as argued in the Introduction, it is very likely that the ratings of different groups of food samples may rely on different experience dimensions. Such qualitative differences between sample groups can be revealed by applying *Clusterwise Parafac* while clustering the sample mode.

The data set that we will analyze was collected by [98] and pertains to 10 different cheese types (with one cheese type being used twice – as an internal reference – to test the stability of the evaluations of the panelists across the different sessions). For each of the 10 cheese types, 3 samples were created, resulting in 30 cheese samples. Each of the 30 cheese samples was rated by 8 trained panelists on 23 attributes, which refer to textural, appearance, and taste related features that can be observed by smelling, by sight, by tasting, and by touching (an overview of the attributes can be found in Table 7). To score the attributes, a horizontal 15 cm unstructured line scale was used, resulting in scores between 0 and 15. More information regarding the data collection can be found in [98] and [4]; the data were retrieved from http://www.models.kvl.dk/Cream.

The data were pre-processed by, first, centering the data across the cheese samples (i.e., for each combination of an attribute and a panelist the mean across the cheese samples equals zero) and, next, scaling the data per attribute (i.e., across all combinations of a cheese sample and a panelist) to a sum of squares of 1 (for more information regarding pre-processing of three-way data, see [6,99,93]). Next, different *Clusterwise Parafac* analyses (with 25 starts, being the best ones out of 1 rational start, 125 pseudo-random starts, and 25 random starts) were performed on the pre-processed data, with the number of clusters ranging from one to six (no empty clusters allowed) and the number of components from one to four. In Table 4, one can see that a *Clusterwise Parafac* analysis gets computationally cumbersome when the number of clusters and especially the number of components becomes large. Some of the latter solutions (i.e., many clusters and/or components) suffer from degeneracy problems.

In order to select an optimal model, we applied the generalized scree test [31], as explained in Section 3.3, to the obtained loss function values (see Fig. 1). In Table 5, in which the $sr_{q|r}$ values (i.e., second to fifth column) are displayed, together with the sr_q values (i.e., final column), one can see that the optimal number of clusters equals three as the sr_q value is the largest for $Q = 3$. When only considering the solutions

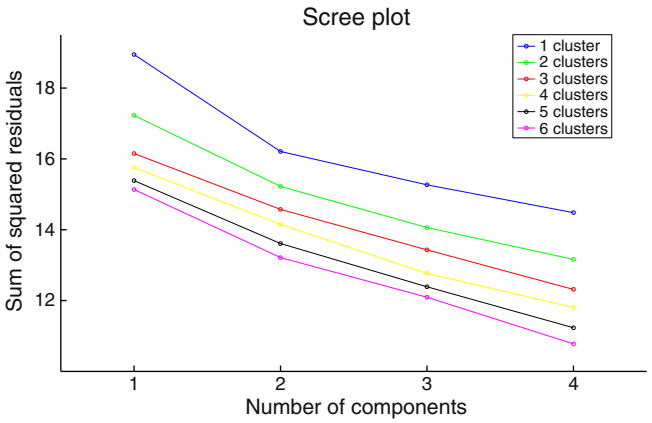

Fig. 1. Loss function values for the fitted *Clusterwise Parafac* solutions with the number of clusters varying from one to six and the number of components ranging from one to four for the cheese data.

with $Q = 3$ clusters, the scree ratio values $sr_{q|Q=3}$ equal 1.3800 and 1.0237 for the solution with 2 and 3 components, respectively. Hence, we selected the solution with three clusters and two components. Each cluster-specific *Parafac* model of this solution is non-degenerate (i.e., across all clusters, the smallest triple cosine product equals −.0806, the smallest eigenvalue is .9194, and the largest condition number equals 1.18).

In Table 6, in which the obtained cheese sample partition is presented, one can see that the first cluster consists of (all the samples of) the Standard full fat cream cheese (34%) and the Commercial cream cheese D (*D-CHO*). The second cluster subsumes the Medium fat reduced (24%) and the Maximum fat reduced cream cheese (16%), together with the Commercial cream cheese B (*B-Prot*), the Prototype cream cheese (*P*), and the Prototype cream cheese with added Butter Aroma (*P + Aroma*). The third cluster, finally, collects both Commercial cream cheeses A (*A-Prot*) and the Commercial cream cheese C (*C-CHO*). The three samples of the same cheese type are always assigned to the same cluster, except for one sample of the Medium fat reduced cream cheese 24% (i.e., assigned to cluster 3 instead of cluster 2) and one sample of the Commercial cream cheese A (i.e., assigned to cluster 2 instead of cluster 3).

When studying the cluster-specific attribute component scores, which are presented in Table 7, it appears that the texture (i.e., Hand- and Mouth-resistance and Mouth-firm versus Mouth-melt down) and the appearance (i.e., Eyes-white versus Eyes-yellow) of the cheeses are the most important dimensions to discriminate among the 33% and the *D-CHO* samples (i.e., first cluster). In particular, in Table 6 and Fig. 2, in which the sample component scores are displayed (for each cluster), one can see that the 33% cheese is considered soft (i.e., Mouth-melt down) and has a yellow appearance (i.e., large negative scores on both components), whereas the *D-CHO* cheese is white and has a firm texture (i.e., large positive component scores). For the second cluster (i.e., 16%, 24%, *B-Prot*, *P*, *P + Aroma*), the texture of the products is again

Table 4
Computation time in hours for different *Clusterwise Parafac* analyses of the cheese data (with 25 starts, which are the best ones out of 151 initial partitionings).

Number of clusters	1 component	2 components	3 components	4 components
1	.00	.01	.02	.03
2	.48	2.47	4.68	5.43
3	.85	3.09	5.38	5.75
4	1.11	3.55	5.53	9.41
5	1.26	3.75	7.34	9.49
6	1.66	3.82	7.36	8.23

Table 5
Scree ratios $sr_{q|r}$ for the number of clusters q ($q = 2,\ldots, Q^{max} - 1$) given the number of components r ($r = 1, \ldots, R^{max}$) and average scree ratio sr_q over the numbers of components for the cheese data. The largest scree ratio in each column is highlighted in boldface.

Number of clusters (Q)	1 component	2 components	3 components	4 components	Mean over components (sr_q)
2	1.60	1.52	**1.92**	1.56	1.32
3	**2.71**	**1.53**	.95	**1.66**	**1.37**
4	1.08	.79	1.78	.89	.91
5	1.47	1.36	1.27	1.25	1.07

Table 6
Cheese sample (object) component scores for the *Clusterwise Parafac* solution with three clusters and two components for the cheese data (s1–s3: sample 1–sample 3); loadings larger (in absolute value) than .30 are indicated in boldface.

Label	Cheese sample (object)	Loadings		Cluster
1	Standard full fat cheese (34%) s1	**−.32**	**.59**	1
1	Standard full fat cheese (34%) s2	**−.39**	**.49**	1
1	Standard full fat cheese (34%) s3	**−.44**	**.60**	1
2	Medium fat reduced cheese (24%) s1	.20	.29	3
2	Medium fat reduced cheese (24%) s2	.19	**.34**	2
2	Medium fat reduced cheese (24%) s3	.25	**.42**	2
3	Maximum fat reduced cheese (16%) s1	−.02	**−.35**	2
3	Maximum fat reduced cheese (16%) s2	.02	**−.35**	2
3	Maximum fat reduced cheese (16%) s3	−.17	**−.53**	2
4	Commercial cheese D (D-CHO) s1	**−.69**	−.10	1
4	Commercial cheese D (D-CHO) s2	**−.64**	−.10	1
4	Commercial cheese D (D-CHO) s3	**−.59**	−.14	1
5	Commercial cheese A (A-Prot) s1	**.53**	−.25	3
5	Commercial cheese A (A-Prot) s2	**−.51**	.14	2
5	Commercial cheese A (A-Prot) s3	**.51**	−.15	3
6	Commercial cheese B (B-Prot) s1	**−.70**	−.28	2
6	Commercial cheese B (B-Prot) s2	**−.56**	−.21	2
6	Commercial cheese B (B-Prot) s3	**−.53**	−.22	2
7	Prototype cheese (P) s1	.24	**−.32**	2
7	Prototype cheese (P) s2	.23	−.28	2
7	Prototype cheese (P) s3	.12	−.24	2
8	Commercial cheese C (C-CHO) s1	**−.60**	.29	3
8	Commercial cheese C (C-CHO) s2	**−.62**	−.04	3
8	Commercial cheese C (C-CHO) s3	**−.57**	−.03	3
9	Commercial cheese A (A-Prot) s1	**.38**	−.23	3
9	Commercial cheese A (A-Prot) s2	**.54**	**−.38**	3
9	Commercial cheese A (A-Prot) s3	**.57**	−.22	3
10	Prototype cheese + Butter (P + Aroma) s1	.22	**−.33**	2
10	Prototype cheese + Butter (P + Aroma) s2	.24	−.22	2
10	Prototype cheese + Butter (P + Aroma) s3	.15	−.28	2

important to differentiate among the samples. However, the other dimension pertains to the differences between sour taste and chalky sensation on the one hand and fat-related flavor properties (i.e., M-fat, M-cream, and M-butter) on the other hand. Fig. 2 shows that the *B-Prot* cheese samples are considered as soft and having a sour taste and chalky sensation, whereas the 16%, *P*, and *P + Aroma* cheeses also taste soft, but have all the fat-related flavor properties. The two samples of the 24% cheese are firm and have a fat/creamy taste. In the third cluster, finally, which consists of two commercial cheeses, the creaminess of these cheeses (i.e., second component) together with their salty taste versus firm and chalky sensation (i.e., first component) are the key features that panelists use to differentiate between these samples. The *A-Prot* cheeses are considered chalky/firm without a creamy taste, whereas the *C-CHO* samples have a salty taste and are a bit creamy.

Table 7
Attribute (variable) component scores for the *Clusterwise Parafac* solution with three clusters and two components for the cheese data; loadings larger (in absolute value) than .30 are indicated in boldface.

Attribute	Cluster 1		Cluster 2		Cluster 3	
Nose-cream	−.10	−.01	.14	−.00	−.06	**.42**
Nose-acidic	.04	.00	−.04	.12	.06	−.18
Nose-butter	−.15	−.16	.24	−.03	−.08	−.07
Nose-old milk	.14	.05	−.20	.06	.14	.15
Eyes-white	−.13	**−.49**	−.19	.03	.26	−.19
Eyes-gray	.04	−.11	−.06	.10	.19	.22
Eyes-yellow	.07	**.55**	.25	.01	−.28	.11
Eyes-green	−.02	.10	.05	−.01	.04	**.36**
Hand-resistance	**−.39**	.05	−.06	**.44**	.20	.12
Eyes-grainy	.18	.25	−.04	.15	.18	**.37**
Eyes-shiny	**.41**	−.22	.12	**−.40**	−.14	−.08
Mouth-firm	**−.35**	−.09	−.09	**.44**	**.32**	−.00
Mouth-melt down	**.30**	.21	.14	**−.41**	−.24	.02
Mouth-resistance	**−.33**	−.23	−.15	**.41**	.25	−.08
Mouth-creaminess	−.22	−.04	−.03	.13	−.21	**.37**
Mouth-grainy	.15	.27	.20	−.11	.07	.17
Mouth-chalky	.19	.02	−.23	.14	**.40**	−.04
Mouth-cream	−.12	.15	**.35**	−.06	−.09	**.42**
Mouth-fat	−.22	−.03	**.34**	.01	−.21	.13
Mouth-butter	−.17	.10	**.46**	−.05	−.27	.07
Mouth-salt	.14	.23	.22	−.03	**−.33**	.08
Mouth-sour	.15	−.08	**−.32**	−.06	.14	−.04
Mouth-sweet	−.11	.18	.10	−.03	−.11	−.02

In Table 8, the cluster-specific panelist scores, which reflect the salience of the cluster-specific dimensions for each panelist when discriminating the samples within each cluster, are displayed. When discriminating the samples of the first cluster, all panelists use both features to more or less the same extent. For the second sample cluster, small differences exist regarding the saliencies of the second component (i.e., the texture component), whereas larger differences are encountered for the sour/chalky versus fat flavor component. Interestingly, the saliencies of the texture dimension, which underlies both the samples of the first and second clusters, differ depending on the sample group that is judged. In the third cluster, all panelists rely (almost) to the same extent on the salty versus chalky/firm component, whereas large differences in saliency are encountered for the cream feature (i.e., only the fourth panelist heavily makes use of this dimension).

Comparing the *Clusterwise Parafac* solution to the standard *Parafac* solution (with two components) as reported in [4], reveals that a standard *Parafac* results in a more simplistic picture of the experience dimensions that drive the discrimination of the cheese samples. Specifically, the variable components for the three groups get mixed up, with the largest group of samples (i.e., the second cluster) dominating the solution. Therefore, it is no surprise that the obtained *Parafac* components are strongly related to the *Clusterwise Parafac* components for the second cluster (i.e., the texture component and the sour versus fat flavor component). However, we have revealed that other (qualitatively different) dimensions come into play when we focus on the finer discriminations among subgroups of cheese samples.

Of course, one should bear in mind that *Clusterwise Parafac* will almost always (i.e., as soon as the data contain noise and at least two clusters are used) fit a given data set better than *Parafac*, because *Clusterwise Parafac* has more modeling freedom. Specifically, for the cheese data, a *Parafac* solution with two components has 118 free parameters (i.e., $(I+J+K-2)\times R$, see [100,72]), whereas a *Clusterwise Parafac* solution with three clusters and two components uses 264 free parameters (i.e., $K+(I+Q+J+Q+K)\times R-2\times R\times Q$).[12] However, because both models are deterministic (i.e., no explicit distributional assumptions about the noise in the data are made), no standard statistical procedure exists to test whether *Clusterwise Parafac* significantly fits the data better than *Parafac*.

As a way out, one may construct a likelihood ratio test on the basis of a minimal stochastic extension of both models. This extension builds on the assumption that all noise entries e_{ijk} (i.e., $e_{ijk}=d_{ijk}-m_{ijk}$) are identically and independently distributed as $N(0,\sigma^2)$, which implies that the associated (negative) log likelihood equals $\log(2\pi)\frac{IJK}{2}+\log(\sigma)IJK+\frac{\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K}e_{ijk}^2}{2\sigma^2}$. Note that the numerator of the ratio in the last term of the likelihood equals loss function (Eq. (3)) and that the noise variance σ^2 can be estimated by $\hat{\sigma}^2=\frac{\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K}e_{ijk}^2}{IJK}$ (for more information, see [38,39]). Thus, the maximum likelihood solution for the minimal stochastic extension equals the least-squares solution for the associated deterministic model [102,103]. For the cheese data, the thus obtained log likelihood for the

[12] Note that in these calculations it is assumed that each parameter, regardless of its type (i.e., component loading or cluster membership), should be weighted equally when determining the model complexity (see [101]).

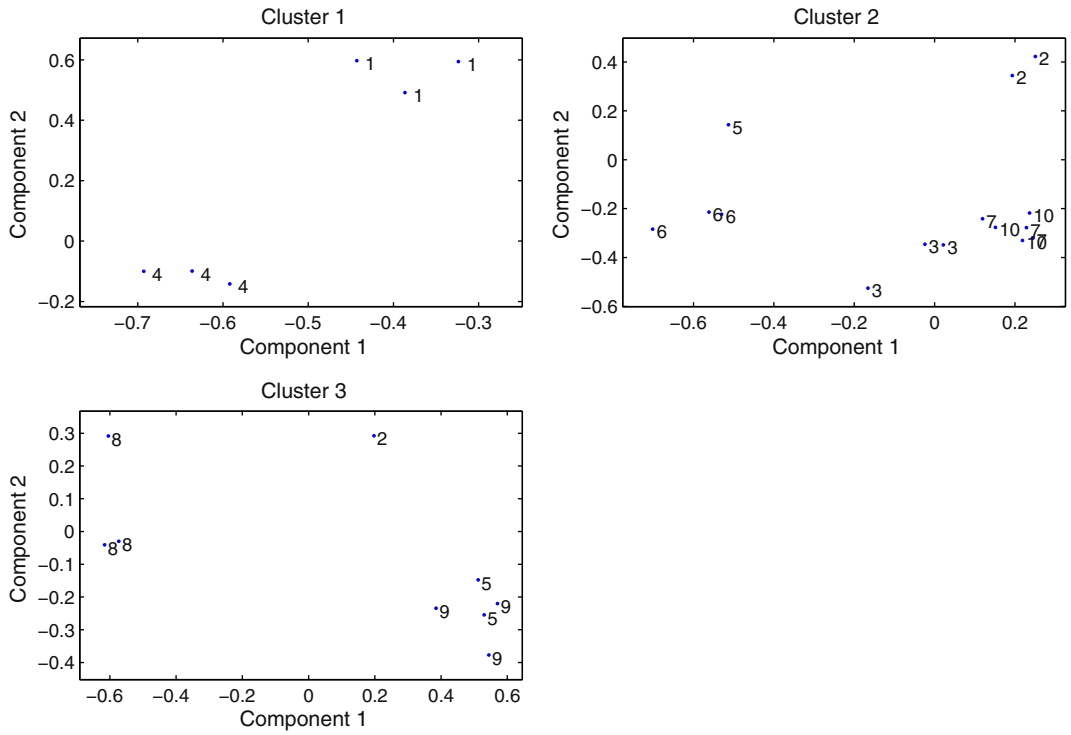

Fig. 2. Cluster-specific cheese sample (object) component scores for the retained *Clusterwise Parafac* solution with three clusters and two components for the cheese data.

Parafac and *Clusterwise Parafac* solutions equals −8259.87 and −8553.25, respectively (i.e., the least-squares fit for both models equals 16.2088 and 14.5742, respectively). Computing twice the difference of both values yields the test statistic (i.e., 586.76) of the likelihood ratio test, which follows a chi-square distribution with 146 degrees of freedom (i.e., the difference in the number of free parameters between both models), resulting in the rejection of the null hypothesis that all cluster-specific parameters are equal across clusters ($p < .0001$). Consequently, we conclude that the *Clusterwise Parafac* model significantly fits the data better than the *Parafac* model.

The same reasoning can be applied to compare the *Clusterwise Parafac* solution to other nested solutions, such as the *PFOCV* solution (see Section 2.3) with three components (c.q., three clusters with one component each). As the latter model has a log likelihood value of −8269.62 and 147 free parameters (and a least-squares fit of 16.1516), we conclude that it fits the data worse than the *Clusterwise Parafac* solution ($p < .0001$).

6. Concluding remarks

In this paper, *Clusterwise Parafac* was introduced as a generic modeling strategy for uncovering heterogeneity in three-way three-mode data. To this end, the elements of one of the three modes are clustered and the covariation within each cluster is summarized by means of a *Parafac* model. In an extensive simulation study it was demonstrated that the *Clusterwise Parafac* algorithm performs well in terms of optimizing the loss function and recovering the true object partition and cluster-specific (variable and source) components. Further, in the context of sensory profiling data, *Clusterwise Parafac* appeared to be successful in disclosing the qualitative differences in underlying component structure between groups of cheese samples.

When adopting *Clusterwise Parafac*, the researcher has to make an important decision, which clearly may influence the obtained results, regarding which data mode should be clustered. Indeed, for our cheese data example, different decisions reveal different types of heterogeneity. A first alternative, which has been adopted in Section 5, is to cluster the cheese samples. As such, qualitative differences in the dimensions that the panelists take into account when rating the cheese samples in a specific cluster can be disclosed. A second choice consists of clustering the panelists. When taking this option, differences in the product space (i.e., component structure) that is underlying the ratings of subgroups of panelists can be studied. For instance, whereas the ratings of one group of panelists may solely depend on the taste of the cheese samples, other panelists may take their appearance into account as well. A final option is to cluster the attributes, which may give insight into the different psychological experience dimensions that underlie different groups of attributes. The final decision about the clustered mode should thus be based on the kind of differences one is interested in.

Obviously, the *Clusterwise Parafac* model may further be extended or restricted in many different ways. A first possible extension relaxes the assumption that the number of components for the cluster-specific *Parafac* models should be the same across clusters, which in some instances may be a too restrictive assumption. For example, there are no substantive reasons to expect that the dimensionality of the product space should be identical across different groups of panelists. Therefore, it might make sense to allow the number of components for the cluster-specific *Parafac* models to differ over clusters. A useful point of departure for this extension could be the work of [27] in the context of clusterwise simultaneous component analysis.

Table 8
Panelist (source) component scores for the *Clusterwise Parafac* solution with three clusters and two components for the cheese data; loadings larger (in absolute value) than .30 are indicated in boldface.

	Cluster 1		Cluster 2		Cluster 3	
Panelist 1	.29	**.41**	**.34**	**.30**	**.41**	−.07
Panelist 2	.22	**.40**	**.56**	.28	**.38**	.16
Panelist 3	.26	**.36**	.25	**.32**	**.31**	−.08
Panelist 4	**.59**	**.33**	**.52**	**.42**	**.31**	**.93**
Panelist 5	**.30**	.19	.18	**.38**	.29	−.12
Panelist 6	**.32**	**.40**	**.31**	**.41**	**.37**	−.25
Panelist 7	**.40**	**.42**	**.34**	**.38**	**.43**	−.12
Panelist 8	**.32**	.26	.08	**.31**	**.30**	.00

A second extension of the *Clusterwise Parafac* model pertains to the type of differences that can be modeled. In our analysis of the cheese data, the components are allowed to differ across the sample clusters (i.e., qualitative differences between samples). However, as each cluster is modeled with *Parafac*, within a particular sample cluster, the scores of the samples on the extracted dimensions are assumed to be the same across panelists.[13] For some data sets at hand, however, this assumption may be far from satisfied, resulting in *Clusterwise Parafac* not fitting the data well. To solve this problem, one may adopt the *Clusterwise SCA* approach [21] in which sample scores may differ over panelists. For instance, one may develop a *Clusterwise SCA-PF2* model, a *Clusterwise SCA* variant that coincides with a clusterwise version of the *Parafac2* model [104,105].

Finally, in our application, the component scores for the attributes and the panelists are allowed to vary across the sample clusters. In some cases, however, it may be assumed that the covariation for different sample groups can be summarized quite well by means of the same experience dimensions, but that the weight that the panelists give to these dimensions depends on the sample cluster under study. This hypothesis can be incorporated in the analysis by restricting the attribute component scores to be the same across the clusters. To fit the resulting model, one has to adapt the estimation of the component scores, which can no longer be achieved by fitting separate *Parafac* models to the data within each cluster. To this end, the *Linked-mode Parafac* algorithm [93,106,107], in which a three-way data array and a two-way data matrix are analyzed that share one mode (i.e., the LMPCA model, see [108]), may be extended to the case of different three-way data arrays that have a single mode in common.

Acknowledgments

The first author is a post-doctoral researcher at the Fund of Scientific Research (FWO)—Flanders. The research reported in this article was partially supported by the Fund for Scientific Research (FWO)—Flanders (Belgium), Project G.0477.09 awarded to Eva Ceulemans, Marieke Timmerman, and Patrick Onghena, and by the Research Council of KU Leuven (GOA/2010/02).

References

[1] F.L. Hitchcock, The expression of a tensor or a polyadic as a sum of products, J. Math. Phys. 6 (1927) 164–189.
[2] R.A. Harshman, Foundations of the PARAFAC procedure: models and conditions for an explanatory multi-modal factor analysis, UCLA Working Papers in Phonetics, 16, 1970, pp. 1–84.
[3] J.D. Carroll, J.J. Chang, Analysis of individual differences in multidimensional scaling via an n-way generalization of "Eckart–Young" decomposition, Psychometrika 35 (1970) 283–319.
[4] R. Bro, E.M. Qannari, H.A.L. Kiers, T. Naes, M.B. Frost, Multi-way models for sensory profiling data, J. Chemom. 22 (2008) 36–45.
[5] R. Bro, PARAFAC: tutorial and applications, Chemom. Intell. Lab. Syst. 38 (2) (1997) 149–171.
[6] A.K. Smilde, R. Bro, P. Geladi, Multi-way Analysis With Applications in the Chemical Sciences, Wiley, Chichester, UK, 2004.
[7] E. Acar, B. Yener, Unsupervised multiway data analysis: a literature survey, IEEE Trans. Knowl. Data Eng. 21 (1) (2009) 6–20.
[8] J. Christensen, E.M. Becker, C.S. Frederiksen, Fluorescence spectroscopy and PARAFAC in the analysis of yogurt, Chemom. Intell. Lab. Syst. 75 (2005) 201–208.
[9] Y.N. Ni, D.Q. Lin, S. Kokot, Synchronous fluorescence, UV–visible spectrophotometric, and voltammetric studies of the competitive interaction of bis(1,10-phenanthroline) copper(ii) complex and neutral red with DNA, Anal. Biochem. Methods Biol. Sci. 352 (2) (2006) 231–242.
[10] Y.N. Ni, D.Q. Lin, S. Kokot, Synchronous fluorescence and UV–vis spectroscopic studies of interaction between the tetracycline antibiotic, aluminium ions and DNA with the aid of the methylene blue dye probe, Anal. Chim. Acta. 606 (1) (2008) 19–25.
[11] A. Sahar, T. Boubellouta, S. Portanguen, A. Kondjoyan, E. Dufour, Synchronous front-face fluorescence spectroscopy coupled with parallel factors (PARAFAC) analysis to study the effects of cooking time on meat, J. Food Sci. 74 (9) (2009) 534–539.
[12] L.G. Larsen, G.R. Aiken, J.W. Harvey, G.B. Noe, J.P. Crimaldi, Using fluorescence spectroscopy to trace seasonal dom dynamics, disturbance effects, and hydrologic transport in the Florida everglades, J. Geophys. Res. Biogeosci. 115 (G3) (2010) 2005–2012.
[13] D. Arroyo, M.C. Ortiz, L.A. Sarabia, F. Palacios, Advantages of PARAFAC calibration in the determination of malachite green and its metabolite in fish by liquid chromatography–tandem mass spectrometry, J. Chromatogr. A 1187 (1–2) (2008) 1–10.
[14] R. Bro, N. Viereck, M. Toft, H. Toft, P.I. Hansen, S.B. Engelsen, Mathematical chromatography solves the cocktail party effect in mixtures using 2D spectra and PARAFAC, Trends Anal. Chem. 29 (4) (2010) 281–284.
[15] L.A.F. de Godoy, M.P. Pedroso, L.W. Hantao, F. Augustoa, R.J. Poppi, Determination of fuel origin by comprehensive 2D GC-FID and parallel factor analysis, J. Braz. Chem. Soc. 24 (4) (2013) 645–650.
[16] W.P. Gardner, R.E. Shaffer, J.E. Girard, J.H. Callahan, Application of quantitative chemometric analysis techniques to direct sampling mass spectrometry, Anal. Chem. 73 (2001) 596–605.
[17] A. Moreda-Pineiro, A. Marcos, A. Fisher, S.J. Hill, Parallel factor analysis for the study of systematic error in inductively coupled plasma atomic emission spectrometry and mass spectrometry, J. Anal. At. Spectrom. 16 (2001) 360–369.
[18] B. Hemmateenejad, Z. Rezaei, S. Zaeri, Second-order calibration of excitation–emission matrix fluorescence spectra for determination of glutathione in human plasma, Talanta 79 (3) (2009) 648–656.
[19] D.-Z. Tu, H.-L. Wu, Y.-N. Li, J. Zhang, Y. Li, C.-C. Nie, X.-H. Zhang, R.-Q. Yu, Measuring estriol and estrone simultaneously in liquid cosmetic samples using second-order calibration coupled with excitation–emission matrix fluorescence based on region selection, Anal. Methods 4 (2012) 222–229.
[20] X. Meng, A.J. Morris, E.B. Martin, On-line monitoring of batch processes using a PARAFAC representation, J. Chemom. 17 (1) (2003) 65–81.
[21] K. De Roover, M.E. Timmerman, I. Van Mechelen, E. Ceulemans, On the added value of multiset methods for three-way data analysis, Chemom. Intell. Lab. Syst. 129 (2013) 98–107.
[22] R.L. Harshman, Determination and proof of minimum uniqueness conditions for PARAFAC1, UCLA Working Papers in Phonetics, 22, 1972, pp. 111–117.
[23] J.B. Kruskal, Three-way arrays: rank and uniqueness of trilinear decompositions, with applications to arithmetic complexity and statistics, Linear Algebra Appl. 18 (1977) 95–138.
[24] J.B. Kruskal, Rank, decomposition, and uniqueness for 3-way and *n*-way arrays, in: R. Coppi, S. Bolasco (Eds.), Multiway Data Analysis, Elsevier, Amsterdam, 1989, pp. 8–18.
[25] N.D. Sidiropoulos, R. Bro, On the uniqueness of multilinear decomposition of *n*-way arrays, J. Chemom. 14 (2000) 229–239.
[26] J.M.F. ten Berge, N.D. Sidiropoulos, On uniqueness in CANDECOMP/PARAFAC, Psychometrika 67 (2002) 399–409.
[27] K. De Roover, E. Ceulemans, M.E. Timmerman, J.B. Nezlek, P. Onghena, Modeling differences in the dimensionality of multiblock data by means of clusterwise simultaneous component analysis, Psychometrika 78 (2013) 648–668, http://dx.doi.org/10.1007/S11336-013-9318-4.
[28] M.E. Timmerman, E. Ceulemans, K. De Roover, K. Van Leeuwen, Subspace k-means clustering, Behav. Res. Methods (2013), http://dx.doi.org/10.3758/s13428-013-0329-y.
[29] W.P. Krijnen, The Analysis of Three-way Arrays by Constrained PARAFAC Methods, DSWO Press, Leiden, 1993.
[30] K. De Roover, E. Ceulemans, M.E. Timmerman, P. Onghena, A clusterwise simultaneous component method for capturing within-cluster differences in component variances and correlations, Br. J. Math. Stat. Psychol. 66 (2013) 81–102.
[31] K. De Roover, E. Ceulemans, M.E. Timmerman, How to perform multiblock component analysis in practice, Behav. Res. Methods 44 (2012) 41–56.
[32] K. De Roover, E. Ceulemans, M.E. Timmerman, K. Vansteelandt, J. Stouten, P. Onghena, Clusterwise simultaneous component analysis for analyzing structural differences in multivariate multiblock data, Psychol. Methods 17 (2012) 100–119.
[33] R.E. Millsap, W. Meredith, Component analysis in cross-sectional and longitudinal data, Psychometrika 53 (1988) 123–134.
[34] H.A.L. Kiers, J.M.F. ten Berge, Alternating least squares algorithms for simultaneous components analysis with equal component weight matrices for all populations, Psychometrika 54 (1989) 467–473.
[35] J.M.F. ten Berge, H.A.L. Kiers, V. Van der Stel, Simultaneous components analysis, Stat. Appl. 4 (1992) 377–392.
[36] K. Van Deun, A.K. Smilde, M.J. van der Werf, H.A.L. Kiers, I. Van Mechelen, A structured overview of simultaneous component based data integration, BMC Bioinforma. 10 (2009) 246.
[37] H.A.L. Kiers, J.M.F. ten Berge, Hierarchical relations between methods for simultaneous component analysis and a technique for rotation to a simple simultaneous structure, Br. J. Math. Stat. Psychol. 47 (1994) 109–126.
[38] T.F. Wilderjans, E. Ceulemans, I. Van Mechelen, R.A. van den Berg, Simultaneous analysis of coupled data matrices subject to different amounts of noise, Br. J. Math. Stat. Psychol. 64 (2011) 277–290.
[39] R.A. van den Berg, I. Van Mechelen, T.F. Wilderjans, K. Van Deun, H.A.L. Kiers, A.K. Smilde, Integrating functional genomics data using maximum likelihood based simultaneous component analysis, BMC Bioinforma. 10 (2009) 340.
[40] M.E. Timmerman, H.A.L. Kiers, Four simultaneous component models for the analysis of multivariate time series from more than one subject to model intraindividual and interindividual differences, Psychometrika 68 (2003) 105–121.

[13] To be precise, *Parafac* allows the sample scores to vary proportionally (i.e., taking the saliencies into account) across the panelists, which can be considered as (a) rather restrictive (type of quantitative differences between samples).

[41] K. Van Deun, T.F. Wilderjans, R.A. van den Berg, A. Antoniadis, I. Van Mechelen, A flexible framework for sparse simultaneous component based data integration, BMC Bioinforma. 12 (2011) 448.
[42] R. Rocci, M. Vichi, Three-mode component analysis with crisp or fuzzy partition of units, Psychometrika 70 (4) (2005) 715–736.
[43] M. Vichi, R. Rocci, H.A.L. Kiers, Simultaneous component and clustering models for three-way data: within and between approaches, J. Classif. 24 (2007) 71–98.
[44] G. De Soete, J.D. Carroll, K-means clustering in a low-dimensional Euclidean space, in: E. Diday, Y. Lechevallier, M. Schader, P. Bertrand, B. Burtschy (Eds.), New Approaches in Classification and Data Analysis, Springer Verlag, Heidelberg, 1994, pp. 212–219.
[45] H.H. Bock, On the interface between cluster analysis, principal component analysis, and multidimensional scaling, in: H. Bozdogan, A.K. Gupta (Eds.), Multivariate Statistical Modeling and Data Analysis, Reidel Publishing Company, Dordrecht, 1987, pp. 17–34.
[46] W. Stute, L.X. Zhu, Asymptotics of k-means clustering based on projection pursuit, Sankhya, Indian J. Stat. Ser. A 57 (3) (1995) 462–471.
[47] M. Vichi, H.A.L. Kiers, Factorial k-means analysis for two-way data, Comput. Stat. Data Anal. 37 (2001) 49–64.
[48] M.E. Timmerman, E. Ceulemans, H.A.L. Kiers, M. Vichi, Factorial and reduced k-means reconsidered, Comput. Stat. Data Anal. 54 (2010) 1858–1871.
[49] P.M. Kroonenberg, J. de Leeuw, Principal component analysis of three-mode data by means of alternating least squares algorithms, Psychometrika 45 (1980) 69–97.
[50] M.J. Brusco, A repetitive branch-and-bound algorithm for minimum within-cluster sums of squares partitioning, Psychometrika 71 (2006) 347–363.
[51] B.J. van Os, J.J. Meulman, Improving dynamic programming strategies for partitioning, J. Classif. 21 (2004) 207–230.
[52] J.M.F. ten Berge, Least Squares Optimization in Multivariate Analysis, DSWO Press, Leiden, 1993.
[53] J. de Leeuw, Block-relaxation algorithms in statistics, in: H. Bock, W. Lenski, M.M. Richter (Eds.), Information Systems and Data Analysis, Springer-Verlag, Berlin, 1994, pp. 308–325.
[54] A.K. Smilde, H.A.L. Kiers, Multiway covariates regression models, J. Chemom. 13 (1999) 31–48.
[55] G. Tomasi, R. Bro, A comparison of algorithms for fitting the PARAFAC model, Comput. Stat. Data Anal. 50 (2006) 1700–1734.
[56] N.M. Faber, R. Bro, P.K. Hopke, Recent developments in CANDECOMP/PARAFAC algorithms: a critical review, Chemom. Intell. Lab. Syst. 65 (2003) 119–137.
[57] C.A. Andersson, R. Bro, The *N*-way toolbox for MATLAB, Chemom. Intell. Lab. Syst. 52 (2000) 1–4.
[58] P.S. Bradley, U.M. Fayyad, Refining initial points for k-means clustering, in: J. Shavlik (Ed.), Proceedings of the 15th International Conference on Machine Learning (ICML98), Morgan Kaufman, San Francisco, CA, 1998, pp. 91–99.
[59] E. Ceulemans, I. Van Mechelen, I. Leenen, The local minima problem in hierarchical classes analysis: an evaluation of a simulated annealing algorithm and various multistart procedures, Psychometrika 72 (2007) 377–391.
[60] D. Steinley, Local optima in k-means clustering: what you don't know may hurt you, Psychol. Methods 8 (2003) 294–304.
[61] L.R. Tucker, A method for synthesis of factor analysis studies, Personnel Research Section Rapport # 984, Department of the Army, Washington, DC, 1951.
[62] A.D. Gordon, Classification, Chapman and Hall, 1981.
[63] F.K. Kuiper, L.A. Fisher, A Monte Carlo comparison of six clustering procedures, Biometrics 31 (1975) 777–783.
[64] R.K. Blashfield, Mixture model tests of cluster analysis: accuracy of four agglomerative hierarchical methods, Psychol. Bull. 83 (1976) 377–388.
[65] G.W. Milligan, An examination of the effect of six types of error perturbation on fifteen clustering algorithms, Psychometrika 45 (1980) 325–342.
[66] S. Hands, B. Everitt, A Monte Carlo study of the recovery of cluster structure in binary data by hierarchical clustering techniques, Multivar. Behav. Res. 22 (1987) 235–243.
[67] G.W. Milligan, M.C. Cooper, A study of standardization of variables in cluster analysis, J. Classif. 5 (1988) 181–204.
[68] L. Ferreira, D.B. Hitchcock, A comparison of hierarchical methods for clustering functional data, Commun. Stat. Simul. Comput. 38 (2009) 1925–1949.
[69] S. Saracli, N. Dogan, I. Dogan, Comparison of hierarchical cluster analysis methods by cophenetic correlation, J. Inequal. Appl. 2013 (2013) 203.
[70] P. Arabie, L.J. Hubert, Combinatorial data analysis, Annu. Rev. Psychol. 43 (1992) 169–203.
[71] R. Bro, H.A.L. Kiers, A new efficient method for determining the number of components in PARAFAC models, J. Chemom. 17 (2003) 274–286.
[72] E. Ceulemans, H.A.L. Kiers, Selecting among three-mode principal component models of different types and complexities: a numerical convex hull based method, Br. J. Math. Stat. Psychol. 59 (2006) 133–150.
[73] E. Ceulemans, H.A.L. Kiers, Discriminating between strong and weak structures in three-mode principal component analysis, Br. J. Math. Stat. Psychol. 62 (2009) 601–620.
[74] T.F. Wilderjans, E. Ceulemans, K. Meers, CHull: a generic convex hull based model selection method, Behav. Res. Methods 45 (2013) 1–15.
[75] R.B. Cattell, The meaning and strategic use of factor analysis, in: R.B. Cattell (Ed.), Handbook of Multivariate Experimental Psychology, Rand McNally, Chicago, 1966, pp. 174–243.
[76] T.F. Wilderjans, G. Lambrechts, B. Maes, E. Ceulemans, Revealing interdyad differences in naturally occurring staff reactions to challenging behavior of clients with severe or profound intellectual disabilities by means of Clusterwise HICLAS, J. Intellect. Disabil. Res. (2013), http://dx.doi.org/10.1111/jir.12076(in press).
[77] M.J. Brusco, J.D. Cradit, Conpar: a method for identifying groups of concordant subject proximity matrices for subsequent multidimensional scaling analyses, J. Math. Psychol. 49 (2005) 142–154.
[78] G.W. Milligan, S.C. Soon, L.M. Sokol, The effect of cluster size, dimensionality, and the number of clusters on recovery of true cluster structure, IEEE Trans. Pattern Anal. Mach. Intell. 5 (1983) 40–47.
[79] M.J. Brusco, J.D. Cradit, A variable selection heuristic for k-means clustering, Psychometrika 66 (2001) 249–270.
[80] T.F. Wilderjans, E. Ceulemans, P. Kuppens, Clusterwise HICLAS: a generic modeling strategy to trace similarities and differences in multi-block binary data, Behav. Res. Methods 44 (2012) 532–545.
[81] T.F. Wilderjans, E. Ceulemans, I. Van Mechelen, The CHIC model: a global model for coupled binary data, Psychometrika 73 (2008) 729–751.
[82] T.F. Wilderjans, E. Ceulemans, I. Van Mechelen, Simultaneous analysis of coupled data blocks differing in size: a comparison of two weighting schemes, Comput. Stat. Data Anal. 53 (2009) 1086–1098.
[83] T.F. Wilderjans, E. Ceulemans, I. Van Mechelen, The SIMCLAS model: simultaneous analysis of coupled binary data matrices with noise heterogeneity between and within data blocks, Psychometrika 77 (2012) 724–740.
[84] L. Hubert, P. Arabie, Comparing partitions, J. Classif. 2 (1985) 193–218.
[85] E.A. Haggard, Intraclass Correlation and the Analysis of Variance, Dryden, New York, 1958.
[86] R.E. Kirk, Experimental Design: Procedures for the Behavioral Sciences, 2nd edition Brooks/Cole, Belmont, CA, 1982.
[87] P.M. Kroonenberg, R.A. Harshman, T. Murakami, Analysing three-way profile data using the PARAFAC and TUCKER3 models, Appl. Multivar. Res. 13 (1) (2009) 5–41.
[88] P.M. Kroonenberg, Three-mode Principal Component Analysis: Theory and Applications, DSWO, Leiden, 1983.
[89] R.A. Harshman, M.E. Lundy, The PARAFAC model for three-way factor analysis and multidimensional scaling, in: H.G. Law, J.C.W. Snyder, J. Hattie, R.P. McDonald (Eds.), Research Methods for Multimode Data Analysis, Praeger, New York, 1984, pp. 122–215.
[90] W.P. Krijnen, P.M. Kroonenberg, Degeneracy and PARAFAC, Technical Report, Department of Psychology, University of Groningen, 2000.
[91] B.C. Mitchell, D.S. Burdick, An empirical comparison of resolution methods for three-way arrays, Chemom. Intell. Lab. Syst. 20 (2) (1993) 149–161.
[92] B.C. Mitchell, D.S. Burdick, Slowly converging PARAFAC sequences: swamps and two-factor degeneracies, J. Chemom. 8 (2) (1994) 155–168.
[93] R.A. Harshman, M.E. Lundy, Data preprocessing and the extended PARAFAC model, in: H.G. Law, C.W.J. Snyder, J. Hattie, R.P. McDonald (Eds.), Research Methods for Multimode Data Analysis, Praeger, New York, 1984, pp. 216–284.
[94] J.B. Kruskal, R.A. Harshman, M.E. Lundy, How 3-MFA data can cause degenerate PARAFAC solutions, among other relationships, in: R. Coppi, S. Bolasco (Eds.), Multiway Data Analysis, Elsevier, Amsterdam, 1989, pp. 115–122.
[95] A. Stegeman, Degeneracy in CANDECOMP/PARAFAC explained for $p \times p \times 2$ arrays of rank $p + 1$ or higher, Psychometrika 71 (3) (2006) 483–501.
[96] A. Stegeman, Degeneracy in CANDECOMP/PARAFAC and INDSCAL explained for several three-sliced arrays with a two-valued typical rank, Psychometrika 72 (4) (2007) 601–619.
[97] P.M. Kroonenberg, Applied Multiway Data Analysis, Wiley, Hoboken, NJ, 2008.
[98] M.B. Frost, The influence of fat content on sensory properties and consumer perception of dairy products, (Ph.D. thesis) The Royal Veterinary and Agricultural University, Center for Advanced Food Studies, Department of Dairy and Food Science, 2002.
[99] R. Bro, A.K. Smilde, Centering and scaling in component analysis, J. Chemom. 17 (2003) 16–33.
[100] T.J. Wansbeek, J. Verhees, Models for multidimensional matrices in econometrics and psychometrics, in: R. Coppi, S. Bolasco (Eds.), Multiway Data Analysis, Elsevier, Amsterdam, 1989, pp. 543–552.
[101] K. Bulteel, T.F. Wilderjans, F. Tuerlinckx, E. Ceulemans, CHull as an alternative to AIC and BIC in the context of mixtures of factor analyzers, Behav. Res. Methods 45 (2013) 782–791, http://dx.doi.org/10.3758/s13428-012-0293-y.
[102] R. Bro, N.D. Sidiropoulos, A.K. Smilde, Maximum likelihood fitting using ordinary least squares algorithms, J. Chemom. 16 (2002) 387–400.
[103] D.T. Andrews, L. Chen, P.D. Wentzell, D.C. Hamilton, Comments on the relationship between principal components analysis and weighted linear regression for bivariate data sets, Chemom. Intell. Lab. Syst. 34 (1996) 231–244.
[104] R.L. Harshman, PARAFAC2. Mathematical and technical notes, UCLA Working Papers in Phonetics, 22, 1972, pp. 30–44.
[105] H.A.L. Kiers, J.M.F. ten Berge, R. Bro, PARAFAC2—part I. a direct fitting algorithm for the PARAFAC2 model, J. Chemom. 13 (1999) 275–294.
[106] R.A. Harshman, M.E. Lundy, PARAFAC: parallel factor analysis, Comput. Stat. Data Anal. 18 (1994) 39–72.
[107] A.K. Smilde, J.A. Westerhuis, R. Boqué, Multiway multiblock component and covariates regression models, J. Chemom. 14 (2000) 301–331.
[108] T.F. Wilderjans, E. Ceulemans, H.A.L. Kiers, K. Meers, The LMPCA program: a graphical user interface for fitting the linked-mode PARAFAC-PCA model to coupled real-valued data, Behav. Res. Methods 41 (2009) 1073–1082.