Chemometrics and Intelligent Laboratory Systems 213 (2021) 104310

Contents lists available at ScienceDirect

Chemometrics and Intelligent Laboratory Systems

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

ONVAR: A simultaneous component analysis approach for disentangling outlying and non-outlying variables

Sopiko Gvaladze [a,*], Kim De Roover [a,b], Eva Ceulemans [a]

[a] Faculty of Psychology and Educational Sciences, KU Leuven, Leuven, Belgium
[b] Department of Methodology and Statistics, Tilburg University, Tilburg, Netherlands

ARTICLE INFO

Keywords:
Multi-block data
PCA
Outlying variable
Simultaneous component analysis

ABSTRACT

Multivariate column-coupled multi-block data are collected in many fields of science. Such data consist of multiple object-by-variable blocks, which all share the variable mode. To summarize the main information in such data, principal component analysis per block (separate PCAs) and simultaneous component analysis (SCA) across blocks are highly popular. The main difference is that, with separate PCAs, the loadings can differ from block to block, whereas SCA restricts the loadings to be the same across the blocks. However, one often sees that most variables have similar correlation patterns across the blocks, whereas a few variables behave differently. We consider those variables as 'non-outlying' and 'outlying' respectively. For various research goals, it makes sense to model the data in blocks simultaneously but to disentangle the outlying and non-outlying variables and impose different restrictions on them. To this end, we present the new Outlying and Non-outlying Variable (ONVar) method. ONVar models all variables and all blocks simultaneously but allows to capture for which variables the loadings are the same and for which they differ. In a simulation study, we show that ONVar flags the outlying variables correctly most of the time and outperforms existing outlying variable detection heuristics. Finally, we illustrate ONVar by applying it to sensory ratings of multiple bread samples.

1. Introduction

Multivariate column-coupled multi-block data are collected in many fields of science [1]. Such data consist of multiple object-by-variable blocks, which all share the variable mode. In behavioral studies, the blocks might represent, for example, different experimental groups [2] or countries [3–5] to which the participants belong. In metabolomics, studies on the expression of multiple genes in a few participant groups (e.g., groups that are vaccinated with different vaccines; [6]) would yield such multi-block multivariate data as well. Depending on the study design, sensometric studies may also produce these type of data. For example, a study in which different panelists rate some food samples using a set of sensory attributes, will also yield multi-block multivariate data, in that the data of each panelist could be considered to constitute a different block [7,8].

When analyzing multivariate column-coupled data, it is often of prime interest to extract a few constructs that summarize the linear relations between the variables. This can be done for each block separately by running a standard principal component analysis (PCA) [9] on each block. Alternatively, one can use simultaneous component analysis (SCA) [10,11], a multi-block extension of PCA, which analyses all blocks simultaneously. PCA and SCA both decompose the data into component scores and loading matrices. The loadings reflect the linear relationships between the variables and the extracted constructs, called components, and are used to interpret these components. The component scores quantify the position of the objects under study on the components. The main difference between separate PCAs and SCA pertains to the loading matrices. In particular, SCA restricts the loading matrices to be the same for all blocks, while with separate PCAs, the loading matrices can differ from block to block. Thus, SCA restricts the extracted constructs to be the same across the blocks, whereas with separate PCAs, they may differ.

This SCA restriction on the loadings is a strong one, but makes perfect sense in many cases. In sensory profiling studies, for instance, the assumption that different assessors interpret and use the different sensory variables in a similar way is crucial [12]. Since these assessors are extensively trained and calibrated beforehand, individual differences in the correlation structure of the variables can indeed be considered nuisance effects. In the behavioral sciences, equivalence of the loadings across the blocks is a necessary requirement for investigating whether the constructs behave the same across the blocks [13–15].

* Corresponding author. Faculty of Psychology and Educational Sciences, KU Leuven, Leuven, Tiensestraat 102, 3000, Leuven, Belgium.
E-mail address: sofiagvaladze13@gmail.com (S. Gvaladze).

https://doi.org/10.1016/j.chemolab.2021.104310
Received 18 June 2020; Received in revised form 16 March 2021; Accepted 3 April 2021
Available online 29 April 2021
0169-7439/© 2021 Elsevier B.V. All rights reserved.

But what to do if such correlation or loading differences occur nevertheless? Depending on the extent of the differences, researchers might take one of the following two routes. If most of the correlations are very different across the blocks, it is recommended to switch to running separate PCAs. However, if the differences are concentrated around a few variables only, it may be better to flag these variables, which we call *outlying*, in line with [16–18] and continue with a simultaneous analysis approach on the subset of variables that behave (virtually) the same across the blocks and are therefore called non-outlying variables. Note that our definition of outlying variables differs from the one given by Hahs-Vaughn [19]; this author uses the term outlying variables to denote variables that hardly correlate with any of the other variables.

Obviously, the question then becomes: how one should decide which variables are the outlying ones? De Roover, Timmerman and Ceulemans [16] developed a heuristic to just do that, which was later improved by Gvaladze et al. [18]. The latter heuristic, called LRUBCM, outperforms the first, called LBCM, in that it yields less false positives. However, unlike LBCM, LRUBCM is rather restrictive in that it is applicable to two-block cases only. Therefore, we focus on LBCM here. The main idea of this heuristic is that one runs separate PCAs on the different blocks, rotates the obtained components to make them maximally similar, and assesses the similarity of the resulting loadings by means of Tucker's congruence coefficient [20–23]. Next, one removes the most outlying variables (as indicated by Tucker's congruence) one by one from the data until the similarity of the loadings of the remaining variables is satisfactory. The components that can be extracted from the remaining variables can be interpreted in the same way across the blocks, although the block-specific loadings might still differ slightly. This stepwise procedure showed promising results in simulation studies and produced meaningful outlying variables when applied to real data.

This heuristic has a major drawback, however. First, the procedure is sequential and greedy, implying that once labeled outlying and removed, a variable cannot re-enter the analyses. This might hinder correct retrieval of all outlying variables. Second, by simply removing the outlying variables, no insight is obtained in the correlation structure of the outlying variables. Indeed, after removing the outlying variables, the final loading structure only expresses the associations among the retained non-outlying variables. However, inspecting the loadings of the outlying variables on the constructs defined by the non-outlying ones may be very informative since it provides insight into the differential behavior of these variables across the blocks[1] (for examples, see Ref. [24]).

To deal with this drawback, this paper proposes a new simultaneous component analysis approach, called Outlying and Non-outlying Variable analysis (ONVar). ONVar models all variables and all blocks simultaneously but allows to disentangle for which variables the loadings are the same (non-outlying variables) and for which they differ (outlying variables); it therefore bridges the gap between SCA and separate PCAs. To this end, ONVar includes, on top of component scores and loadings, a partition vector that clusters the variables into outlying and non-outlying sets where non-outlying variables are restricted to have the same loadings across the blocks, and outlying ones are modeled with block-specific loadings. By means of these loading structures, ONVar gives a full yet parsimonious picture of the linear relations among all the variables and how they possibly differ across the data blocks. In order to find the best fitting solution for a given data set, we present an alternating least squares (ALS) algorithm.

The remainder of this paper is organized as follows: First, we describe the data structure and preprocessing in section 2. In section 3, we start with recapitulating the separate PCAs (section 3.1) and SCA-ECP approaches (section 3.2), followed by a description of the ONVar model and algorithm (section 3.3). Section 4 presents an extensive simulation study to evaluate the ability of ONVar to correctly detect the outlying variables and to compare ONVar with LBCM in terms of outlying variable detection and with separate PCAs and SCA-ECP in terms of the recovery of the loading matrices. In section 5, ONVar is applied to an empirical data set on sensory bread ratings. In section 6, we discuss our findings and directions for future work.

2. Data structure and pre-processing

In this paper, we assume that the data consists of I data blocks, where each data block $\mathbf{X}_i$ $(i = 1,\ldots,I)$ consists of the scores of N_i objects on the same J variables. The number of objects in each block N_i is allowed to differ between blocks. The full data set $\mathbf{X}$ is obtained by vertically concatenating the $\mathbf{X}_i$ blocks and thus counts $N = \sum_{1}^{I} N_i$ rows.

The data are preprocessed before the analysis. As we are interested in between-block differences in correlation structure, the variables are centered in each block, discarding between-block differences in means which may otherwise distort the obtained loadings, and each variable is rescaled so that its variance equals one per block [25,26]. Thus preprocessing boils down to standardizing all variables per block.

3. Separate PCAs, SCA and ONVar

In this section, we recapitulate the two existing approaches – separate PCAs [9] and SCA-ECP [11,27], and present the novel ONVar model and algorithm. The main difference between these three approaches pertains to the restrictions they put on the loading structures across the blocks. In this regard, SCA-ECP and separate PCAs can be considered as two opposite ends of a scale, as SCA-ECP fixes the loading structures to be entirely the same across blocks while separate PCAs allow them to differ for all variables. ONVar is in between these two extremes as it only restricts the loadings of the non-outlying variables to be the same across the blocks, whereas the loadings of the outlying variables are allowed to differ. Note that we restrict the number of components,Q, to be the same across the blocks. This restriction is quite reasonable given that we expect that most of the variables behave similarly across the blocks and, thus, that similar dimensions are extracted for all blocks.

3.1. Separate PCAs

Separate PCAs reconstruct each data block, $\mathbf{X}_i$ $(i = 1\ldots I)$, as follows:

$$\mathbf{X}_i = \mathbf{F}_i\mathbf{B}_i^{'} + \mathbf{E}_i \tag{1}$$

The loadings in the block-specific loading matrices $\mathbf{B}_i(J \times Q)$ represent the relations between the observed variables and the components and can be used to label the obtained components. Note that these labels may thus differ across the blocks. The component score matrices $\mathbf{F}_i$ $(N_i \times Q)$ hold the scores of the N_i objects on the Q block-specific components. Moreover, to partially identify the model (partially, because rotation is still allowed), the variance of each column of $\mathbf{F}_i$ is fixed to one and, due to the centering of the data blocks, the mean of each column of $\mathbf{F}_i$ equals zero. The matrix $\mathbf{E}_i$ denotes the residual matrix.

3.2. SCA-ECP

SCA-ECP stands for simultaneous component analysis with Equal Cross-Product constraints [11,27]. SCA-ECP yields a single loading matrix $\mathbf{B}$ $(J \times Q)$ for both blocks and models them as follows:

$$\mathbf{X}_i = \mathbf{F}_i\mathbf{B}^{'} + \mathbf{E}_i \tag{2}$$

[1] Note that applying separate PCAs on all variables does not offer a solution, as this approach allows the loadings of all the variables, outlying and non-outlying, to vary from block to block. As a result, the distinction between both sets of variables will often be blurry as the block-specific loadings of the non-outlying variables might differ across the blocks as well due to error fitting, potentially complicating the interpretation of the constructs.

This means that SCA-ECP extracts a single set of components only. As was the case for separate PCAs, SCA-ECP model sets the variances of the component scores (i.e., the variance of each column of $\mathbf{F}_i$) to one. Moreover, this SCA variant also restricts the correlations of the component scores to be equal across the blocks.

3.3. ONVar

3.3.1. Model

On top of the component scores and loading matrices $\mathbf{F}_i$ and $\mathbf{B}_i$, the ONVar model includes a binary partition vector $\mathbf{p}$ $(J \times 1)$ that splits the variables into two groups: a non-outlying group that adheres to the regular SCA-ECP restrictions and an outlying group for which the loadings may differ across the blocks. If a variable j is non-outlying, the corresponding entry p_j of $\mathbf{p}$ will equal 0, otherwise it will equal 1. Thus, if p_j equals zero, this implies that $b_{jq}^{i} = b_{jq}^{i'}$ $(i, i' = 1, \ldots, I)$. When all the entries of $\mathbf{p}$ equal 0 (resp. 1), the ONVar model boils down to SCA-ECP (resp. separate PCAs). To obtain stable estimates, we recommend to have at least $3 \times Q$ non-outlying variables, even though the model can technically be estimated when the data contain as few as Q non-outlying variables. Given what is technically possible, we impose the restriction that the sum of the entries of $\mathbf{p}$ should not exceed J-Q.

Given $\mathbf{F}_i$, $\mathbf{B}_i$ and $\mathbf{p}$, the data blocks $\mathbf{X}_i$ $(i = 1, \ldots, I)$, are reconstructed as follows:

$$\mathbf{X}_i = \mathbf{F}_i \mathbf{B}_i^{'} + \mathbf{E}_i \tag{3}$$

In ONVar, the variance of each column of $\mathbf{F}_i$ is fixed to one as well. To better grasp the implications of the ONVar model, we rearrange the columns of the data blocks $\mathbf{X}_i$ so that the non-outlying variables come first and the outlying variables come last. To this end, we define the reorder function $reorder\{\mathbf{X}, \mathbf{p}\}$ that reorders the $\mathbf{X}_i$ matrices according to the information in $\mathbf{p}$: $\mathbf{X}_i^{reordered} = [\mathbf{X}_i^{non-outlying} \mathbf{X}_i^{outlying}]$. This reordering does not affect the model fit, but simplifies notation.

Equation (3) can then be rewritten as follows:

$$\mathbf{X}_i^{reordered} = \mathbf{F}_i \begin{bmatrix} \mathbf{B}^{non-outlying} \\ \mathbf{B}_i^{outlying} \end{bmatrix}^{'} + \mathbf{E}_i \tag{4}$$

In Equation (4), the reordered loading matrices $\mathbf{B}_i^{reordered}$ are represented as partitioned matrices, $\begin{bmatrix} \mathbf{B}^{non-outlying} \\ \mathbf{B}_i^{outlying} \end{bmatrix}$, where $\mathbf{B}^{non-outlying}$ is the same across the groups and $\mathbf{B}_i^{outlying}$ differs across the groups. The $\mathbf{B}_i^{reordered}$ matrices have rotational freedom, which means that they can be rotated to a more interpretable solution, without affecting the fit. To preserve the equality of the non-outlying loadings across the blocks, we apply Varimax rotation [28] to $\mathbf{B}^{non-outlying}$, which results in the rotation matrix $\mathbf{T}$. Afterwards, we rotate the $\mathbf{B}_i^{outlying}$ parts using the same rotation matrix $\mathbf{T}$.

3.3.2. Estimation

Loss function. Given a prespecified number of components, Q, the aim of the analysis is to find the partition vector $\mathbf{p}$, the component score matrices $\mathbf{F}_i$ and the loading matrices $\mathbf{B}_i$, that minimize the following penalized loss function:

$$L = (1-\alpha)\frac{\sum_{i=1}^{I} \mathbf{X}_i - \mathbf{F}_i \mathbf{B}_{i2}^{'2}}{N} + \alpha \mathbf{p}_1 = (1-\alpha)\frac{\sum_{i=1}^{I} \mathbf{E}_{i2}^{2}}{N} + \alpha \mathbf{p}_1 \tag{5}$$

Here 1 and 2 denote the L1-norm and the Euclidean norm respectively, where the L1-norm is the sum of the absolute values of the entries and the Euclidean norm equals the square root of the sum of the squared entries. We divide $\sum_{i=1}^{I} \mathbf{E}_{i2}^{2}$ by N to obtain a value between 0 and J, assuming the data are preprocessed as discussed in Section 2. $\mathbf{p}_1$ will yield a value between 0 and J-Q that corresponds to the number of outlying variables. The tuning parameter α varies between 0 and 1. When α is set to 0, the algorithm will focus on minimizing the sum of the squared residuals and, therefore, up to J-Q variables will most likely be labeled as outlying. When α is set to 1, L is minimized by setting all variables as non-outlying. In order to identify the optimal α value for a given dataset, we propose to use the CHull heuristic [29,30], as will be discussed in section 3.3.3 on model selection.

Algorithm: In order to find the solution with the minimal L value for a given α and Q, we developed an alternating least squares (ALS) algorithm, in which the partition, $\mathbf{p}$, of the variables into outlying and non-outlying variables, the component scores matrices $\mathbf{F}_i$ and the loading matrices $\mathbf{B}_i$ are updated cyclically until the algorithm converges. To avoid obtaining a local minimum, we run the algorithm 20 times, each time starting from a different random start.[2] Out of the 20 resulting solutions, the one with the lowest L value is retained. The algorithm consists of the following steps:

1. Initialize the partition vector $\mathbf{p}$: The entries of the partition vector are independently drawn from a Bernoulli distribution with the Bernoulli parameter being set to $1/3$. Note that users can obviously specify a different parameter value. If the resulting vector contains less than Q zeroes, it is discarded and new entries are sampled.
2. Given the partition vector $\mathbf{p}$ and the associated reordering of $\mathbf{X}$, estimate the component scores and the loading matrices: To this end, we use an ALS procedure, comprising the following steps:
   a. Based on $\mathbf{X}^{non-outlying}$ compute $\mathbf{F}$ and $\mathbf{B}^{non-outlying}$: Decompose $\mathbf{X}^{non-outlying}$ into $\mathbf{U}$, $\mathbf{S}$ and $\mathbf{V}$, with $\mathbf{X}^{non-outlying} = \mathbf{USV}^{'}$, using singular value decomposition. Least squares estimates of $\mathbf{F}$ (which is a vertical concatenation of $\mathbf{F}_i$, $i = 1, \ldots, I$) and $\mathbf{B}^{non-outlying}$ can then be obtained as $\mathbf{F} = \sqrt{N}\mathbf{U}_{(q)}$ and $\mathbf{B}^{non-outlying} = \frac{1}{\sqrt{N}}\mathbf{V}_{(q)}\mathbf{S}_{(q)}$, where $\mathbf{U}_{(q)}$ and $\mathbf{V}_{(q)}$ are the first Q columns of $\mathbf{U}$ and $\mathbf{V}$ respectively and $\mathbf{S}_{(q)}$ consists of the first Q rows and columns of $\mathbf{S}$.
   b. Based on $\mathbf{X}_i^{outlying}$ and $\mathbf{F}_i$ compute the block-specific parts of $\mathbf{B}_i$, denoted by $\mathbf{B}_i^{outlying}$. These parts are estimated by a regression step: $\mathbf{B}_i^{outlying} = ((\mathbf{F}_i^{'}\mathbf{F}_i)^{-1}\mathbf{F}_i^{'}\mathbf{X}_i^{outlying})^{'}$, where $i = 1, \ldots, I$.
   c. Based on the full $\mathbf{B}_i$, which is a vertical concatenation of $\mathbf{B}^{non-outlying}$ and $\mathbf{B}_i^{outlying}$, re-estimate $\mathbf{F}_i$: To obtain $\mathbf{F}_i$ for $i = 1, \ldots, I$, decompose $\mathbf{X}_i^{reordered}\mathbf{B}_i$ into $\mathbf{U}_i$, $\mathbf{S}_i$ and $\mathbf{V}_i$ through singular value decomposition and set $\mathbf{F}_i = \sqrt{N}\mathbf{U}_i\mathbf{V}_i^{'}$. $\mathbf{F}$ is obtained by vertically concatenating the $\mathbf{F}_i$ matrices.
   d. Re-estimate $\mathbf{B}^{non-outlying}$ and $\mathbf{B}_i^{outlying}$ through regression steps: $\mathbf{B}^{non-outlying}$ and $\mathbf{B}_i^{outlying}$ are respectively computed as $\mathbf{B}^{non-outlying} = ((\mathbf{F}^{'}\mathbf{F})^{-1}\mathbf{F}^{'}\mathbf{X}^{non-outlying})^{'}$ and $\mathbf{B}_i^{outlying} = ((\mathbf{F}_i^{'}\mathbf{F}_i)^{-1}\mathbf{F}_i^{'}\mathbf{X}_i^{outlying})^{'}$. Compute $SSQ^Z = \sum_{i=1}^{I} \mathbf{X}_i - \mathbf{F}_i\mathbf{B}_i^{'}$, where z indicates the iteration index.
   a. Repeat steps c and d until convergence, where convergence has been reached if $\frac{SSQ^z - SSQ^{z-1}}{NJ} < .001$.
3. Re-estimate the partition vector $\mathbf{p}$: For each of the variables, it is evaluated how the L value would change if the variable is reassigned to the outlying group (if it was non-outlying) or vice versa (and step 2

[2] In a pilot study, we performed 100 starts, but for the selected α value, all the 100 resulting solutions were almost always the same. Therefore, we reduced computational effort by decreasing the number of random starts to 20. Note that the sufficient number of starts might obviously depend on specific data characteristics, such as the number of variables and the degree of outlyingness of the outlying variables. Therefore, we recommend users to always check how often the same solution was obtained across multiple starts and to increase the number of starts if needed.

is performed to update the component score and loading matrices accordingly). If the L value decreases, the variable is reassigned; otherwise, it is not.
4. Steps 2 and 3 are repeated until convergence of the loss function L, where convergence has been reached if $L_{previous} - L_{current} < \varepsilon$, with ε a pre-specified small number (e.g., 10^{-6}).

The ONVar model is implemented in Matlab R2019a and the code can be acquired from the first author on request.

3.3.3. Model selection

In the previous subsection, we described how to fit an ONVar model given a specific α value and number of components Q. We now focus on how these values should be selected.

With respect to the number of components Q, this value can be selected based on previous studies or a priori hypotheses about the correlation structure of the variables. Alternatively, building on the assumption that the data contain a few outlying variables only, one can apply SCA-ECP to the data using different numbers of components and assess which number of components yields an optimal balance between fit to the data and model complexity. To this end, one can conduct a scree test [31–33] or run parallel analysis [34].

Next, we propose to tune the α value using the CHull procedure [29,30]. The key idea of this procedure is to base model selection on the complexity versus fit balance of multiple solutions. To this end, one plots the badness-of-fit of all solutions considered as a function of the complexity of these solutions. Next, one retains the solutions on the lower boundary of the convex hull of this plot and performs a scree test on these hull solutions. To implement this idea for ONVar, one first fits models with α values that increase from 0.00001 to 1, in small steps. Note that, for computational efficiency, higher α values can be discarded as soon as a specific α value results in a solution with no outlying variables. Next, for each retained α value, one plots the corresponding badness-of-fit value f (i.e., the unweighted first term of L in equation (3)) against the number of outlying variables v (i.e., the unweighted second term of L in equation (3)), as well as the solution obtained when running separate PCAs on the data (i.e., an optimal estimate of the fit-value for $v = J$), and retrieves the hull solutions. This step automatically implies that, in case more than one α value yields the same number of outlying variables, one retains the one leading to the lowest badness-of-fit. The hull solutions should show that the badness-of-fit decreases when the number of outlying variables increases. Next, one implements the scree test to pick the α value associated with the solution after which the decrease in badness-of-fit levels off, implying that allowing for more outlying variables does not yield a clearly better description of the data. The CHull procedure computes the following scree test values st_s for the $s = 1, \ldots, S$ hull solutions, except for the one with no outlying variables (i.e., $s = 1$) and the one with the highest amount of outlying variables (i.e., $s = S$):

$$st_s = \frac{\left(\frac{f_{s-1} - f_s}{v_s - v_{s-1}}\right)}{\left(\frac{f_s - f_{s+1}}{v_{s+1} - v_s}\right)} \tag{6}$$

In principle, one should retain the α value that yields the highest st_s value. However, the CHull procedure has a weakness which has been pointed out before [18,30] in that small differences in f values can still lead to large st_s values, because dividing a small difference such as 0.01 by an even smaller difference, for example, 0.001 leads to an value of 10, if the v-differences in the formula equal one. To deal with this weakness, we first calculate the 'overall slope' of the scree plot $\frac{f_1 - f_S}{v_S - v_1}$ and compare it to the slope of each individual solution s, calculated as $\frac{f_{s-1} - f_s}{v_s - v_{s-1}}$. In case the sth individual slope value is not larger than the overall slope value, we discard this solution when comparing the scree test values, st_s.

4. Simulation study

4.1. Problem

In this section, we present a simulation study in which we evaluated how well the ONVar algorithm succeeds in finding the outlying variables. Running these analyses, we fixed Q to the correct number of components, but optimized the α value using the procedure described above. We investigated the impact of the following data characteristics: (1) the number of non-outlying variables, (2) the number of outlying variables, (3) the outlyingness type, where we distinguish between high outlyingness, medium outlyingness, and a mixed form, (4) the amount of error in the data, (5) the number of observations per data block, and (6) the number of data blocks. The outlying variables had a different loading pattern in each of these data blocks. We hypothesized that a larger sample size, higher number of non-outlying variables, lower number of outlying variables, higher outlyingness, larger number of data blocks, and lower amounts of error will affect performance positively.

Additionally, we compared the performance of ONVar to that of already existing methods. Specifically, we investigated how well the LBCM outlying variable heuristic [16] succeeds in retrieving the outlying variables and compare numbers of false positives and negatives for ONVar and LBCM.[3] Finally, we compared ONVar to SCA-ECP and separate PCAs, focusing on the recovery of the true loadings. We hypothesized that ONVar will outperform SCA-ECP as SCA-ECP is more restrictive and does not allow the loadings of the outlying variables to differ across the blocks. We expected the performance differences between ONVar and separate PCAs to be smaller, since with separate PCAs all loadings may differ across the blocks. Yet, since separate PCAs are less parsimonious than ONVar, this approach may be more prone to error fitting.

4.2. Design and procedure

All the data sets were simulated with three components. The following six design factors were systematically varied in a complete factorial design:

1. The number of non-outlying variables: 9 and 21.
2. The number of outlying variables: 2 and 4.
3. The outlyingness type: all outlying variables have a high level of outlyingness, all outlying variables have a medium level of outlyingness, half of the outlying variables have a high level of outlyingness and the other half has a medium level of outlyingness.
4. The amount of error in the data, which is the expected proportion of error variance e in the data blocks: .20 and .40.
5. The number of observations per data block: 100,250.
6. The number of blocks: 2, 3 and 4.

The factorial design thus contains $2 \times 2 \times 3 \times 2 \times 2 \times 3 = 144$ cells. For each cell of this design, 20 data matrices $\mathbf{X}$ were generated, yielding 2880 datasets. For each data block of these datasets, the component scores in $\mathbf{F}_i$ were randomly sampled from a standard normal distribution. To generate the two block-specific loading matrices $\mathbf{B}_i$, the following steps were performed. First, the non-outlying variables were distributed equally over the three components. If the non-outlying variable was assigned to the component, the variable has a loading of one on that component and zero loadings on the other components.

[3] We cannot compare OnVar with LRUBCM across the full simulation design, due to LRUBCM being restricted to two blocks only. However, we applied both methods to all 960 two-block cases. Based on the simulation results reported in Ref. [18], we set the perturbation parameter of LRUBCM to 0.05. OnVar clearly outperformed LRUBCM, in that OnVar and LRUBCM correctly detected all outlying variables for 753 and 718 datasets respectively."

Next, the outlying variables were each assigned to one component for the first block, with a loading of one. In case of two outlying variables, two of the three components contain one outlying variable each and the third component contains none, while in case of four outlying variables two out of three components contain one outlying variable each and the third component contains two outlying variables. For the second block, the outlying variables received a loading b_{outl1} on the component to which it was assigned in block 1, but also a loading b_{outl2} on another component - where this other component differed between the second and the third block. The values of b_{outl1} and b_{outl2} depend on the outlyingness type, and amount to $\sqrt{.50}$ and $\sqrt{.50}$ and $\sqrt{.75}$ and $\sqrt{.25}$, respectively, for high and medium outlyingness. The outlying part of the fourth block was a mix of that of the first and second block, in that the loading patterns of the second and third outlying variable were taken from the first block and the loading patterns of the other two - from the second. Afterwards the loading matrices were rescaled by $\sqrt{1-e}$, to take the amount of error e into account. For both blocks, the error matrices, $\mathbf{E}_i$, were randomly sampled from a multivariate normal distribution with zero means and an identity covariance matrix and were rescaled by multiplying it with $\sqrt{e}$ to obtain the desired error level. Finally, each $\mathbf{X}_i$ is computed as $\mathbf{F}_i\mathbf{B}_i^{'} + \mathbf{E}_i$. Afterwards, the data were preprocessed as described in section 2.

4.3. Results

4.3.1. Outlying variable detection - comparison of ONVar and LBCM

We start by looking at how often ONVar and LBCM were able to find the correct partition of variables into outlying and non-outlying ones, [4] [5]. In Table 1, we see that, for ONVar, this was the case for 90% of the simulated data sets, while LBCM had a success rate of 88%. We also conclude that, while the number of non-outlying variables hardly matters (90 vs. 90% for 9 and 21 non-outlying variables) for OnVar, LBCM clearly benefits from more non-outlying variables (86 vs. 90% for 9 and 21 non-outlying variables). This was expected as LBCM is a stepwise procedure that removes one variable in each step and re-estimates the loadings. The accuracy of these updated loadings obviously depends on the number of variables that remains. Both methods yield somewhat better results when the number of outlying variables equals two (ONVar: 92%; LBCM: 90%) rather than four (ONVar: 89%; LBCM: 86%). Higher outlyingness (ONVar: 98%; LBCM: 95%) is easier to detect than medium (ONVar: 87%; LBCM: 84%) and mixed (ONVar: 87%; LBCM: 85%) forms for both methods.

Performance clearly depends on the sample size per block, as it drops from 98% to 84% for ONVar and from 96% to 80% for LBCM, when going from 250 to 100 observations. Also, the amount of error has a large effect (96% vs. 85% for ONVar and 96% vs. 80% for LBCM - for e = 0.20 and e = 0.40, respectively).

Overall, ONVar was able to find the correct variable partition for 2603 out of the 2880 simulated data sets. To gain insight into what went wrong for the rest of the 277 datasets, we checked the differences between the true partition vector and the estimated one. This comparison showed that ONVar resulted in 128 false positives for 70 data sets and 282 false negatives for 205 data sets. For the remaining two datasets, the obtained partition was completely incorrect. LBCM detected a lot more outlying variables than ONVar – 1602 false positives spread over 273 datasets. False negatives is less of an issue for LBCM, as it yielded only 79 false negatives over 69 datasets.

4.3.2. Recovery of loadings: comparison of ONVar and SCA-ECP

In order to compare the performance of ONVar and SCA-ECP, we looked at the recovery of the true loadings across all the simulated data sets. We first obliquely rotated the estimated loading matrices to the true ones. If we take two-block case as an example, we obliquely rotated the estimated $\begin{bmatrix}\mathbf{B}_1^{OnVar}\\ \mathbf{B}_2^{OnVar}\end{bmatrix}$ and $\begin{bmatrix}\mathbf{B}^{SCA-ECP}\\ \mathbf{B}^{SCA-ECP}\end{bmatrix}$, respectively, towards $\begin{bmatrix}\mathbf{B}_1^{True}\\ \mathbf{B}_2^{True}\end{bmatrix}$. These rotations do not affect the restrictions that ONVar and SCA-ECP impose on the loadings. Afterwards, we computed the mean Tucker's congruence [23] between these rotated loadings and the true ones. The Tucker's congruence coefficient is a similarity measure between two vectors of loadings, b_i^1 and b_i^2, of the same length. It is calculated as follows $\phi = \frac{\sum_i b_i^1 b_i^2}{\sqrt{\sum_i (b_i^1)^2 \sum_i (b_i^2)^2}}$. The obtained value varies from −1 to 1. A value close to zero means that the vectors are very dissimilar, whereas absolute values close to one indicate strong similarity.

The obtained congruence values were averaged per design cell and are shown in Table 2. We conclude that SCA-ECP generally performed worse than ONVar. This was expected as SCA-ECP is more restrictive than ONVar and does not allow the loadings of the outlying variables to differ across the blocks. Therefore, the less non-outlying variables, the more outlying variables, and the more outlying variables have a high level of outlyingness (i.e., outlyingness type), the more difficult it becomes for SCA-ECP, whereas ONVar is less affected by these manipulations (see Table 2).

4.3.3. Comparison of ONVar and separate PCAs

When comparing ONVar and separate PCAs we again focused on the recovery of the loadings. To this end, we rotated, for each block separately, the block-specific loading matrices towards their true counterparts. This rotation does not account for the restrictions that ONVar imposes on the loadings of the non-outlying variables, implying that the loadings of these variables might not be the same across the blocks after this rotation. However, this rotation approach ensures a fair comparison between the two methods. Afterwards, we again computed the mean Tucker's congruence between the vertically concatenated and rotated ONVar and PCA loading matrices and the vertically concatenated true

Table 1
Percentage of data sets for which ONVar and LBCM yielded correct partitions, across the manipulated data characteristics in the simulation.

Data characteristics	OnVar	LBCM
9 non-outlying	91	86
21 non-outlying	90	90
2 outlying	92	89
4 outlying	89	87
High outlyingness	98	95
medium outlyingness	87	84
Mixed outlyingness	87	85
20% error	96	96
40% error	85	80
100 obs. per block	84	80
250 obs. per block	98	96
2 block	78	76
3 block	96	95
4 block	97	94

[4] We also investigated how often ONVar found the correct number of outlying variables, irrespective of the partition being correct or not. There were only two datasets out of 2880 simulated ones, for which ONVar found the correct number of outlying variables, but not the correct partition.

[5] We also investigated in a pilot study focusing on the two blocks settings how sensitive ONVar is to incorrect specification of the Bernoulli parameter in the first step of the algorithm. To this end, we ran the algorithm twice, once setting this parameter to 1/3, and once setting it correctly to #outlying/(#outlying+#non-outlying). The number of correct partitions per design cell was not affected by this change. Therefore, as the correct number of outlying variables is generally not known beforehand, we stuck to 1/3.

Table 2
Recovery of the true loadings across the simulated data sets comparing ONVar and SCA-ECP, and ONVar and separate PCAs, respectively. Recovery is measured by means of the Tucker's congruence coefficient.

Data characteristics	ONVar	SCA-ECP	ONVar	Sep. PCAs
9 non-outlying	.9972	.9799	.9977	.9967
21 non-outlying	.9981	.9894	.9982	.9965
2 outlying	.9978	.9884	.9982	.9967
4 outlying	.9974	.9808	.9977	.9965
High outlyingness	.9980	.9795	.9981	.9966
medium outlyingness	.9973	.9897	.9979	.9966
Mixed outlyingness	.9976	.9846	.9979	.9966
20% error	.9986	.9852	.9989	.9983
40% error	.9967	.9840	.9970	.9950
100 observations per block	.9964	.9841	.9970	.9951
250 observations per block	.9988	.9852	.9989	.9981
2 block	.9968	.9883	.9974	.9967
3 block	.9979	.9819	.9981	.9966
4 block	.9981	.9837	.9983	.9966

loading matrices. The results in Table 2 show that ONVar performs slightly better than separate PCAs. These results are in line with our hypothesis that separate PCAs are more prone to error fitting, because these models contain more parameters than ONVar.

5. Illustrative application

In this section, we re-analyze the bread data that were reported on before [7,8,18]. The data can be downloaded from http://www.models.kvl.dk/Sensory_Bread. In this study, two replicates of five different breads were baked, resulting in ten different samples. These bread samples were then evaluated on eleven attributes by eight trained panelists in a fixed vocabulary profiling analysis. As indicated in the introduction, it makes sense to assume that the different panelists interpret and use the eleven sensory variables (see Table 3) in a similar way [12].

De Roover et al. [8] investigated this assumption further. To this end, they considered the samples by attributes data of each panelist as a different data block and applied clusterwise SCA [35–37]. This technique is related to separate PCAs as it partitions the blocks into mutually exclusive clusters, according to the correlation structure of the variables, and fits a separate SCA model to the data within each cluster. The analysis yielded two clusters of panelists and associated components. As holds for separate PCAs, this analysis sheds no direct light on which variables (i.e., attributes) are used differently in the two clusters, and to what extent. To answer this question, Gvaladze et al. [17] did a follow up analysis that resulted in a ranking of the variables from most to least outlying across these two clusters of panelists. Albeit interesting, the latter analysis suffers from two limitations: First, it does not draw a clear line between outlying and non-outlying variables. Second, it does not yield invariant loadings for the non-outlying variables and variant ones for the outlying variables. To address these issues, we applied the ONVar model to the bread data. The first set of analyses builds on the analyses of De Roover et al. [8] and Gvaladze et al. [17] as it uses the same partitioning of the data blocks. Thus, these analyses determine the variables that behave differently in the two clusters of panelists and shed light on their loading patterns. The second set of analyses treats the data of each panelist as a different data block and thus yields a more fine-grained search for outlying variables.

5.1. Outlying variables across two clusters of panelists

To obtain two data blocks, we split the data according to the clusters of panelists identified by De Roover et al. [8]. Specifically, the first data block is the vertical concatenation of the data of the panelists that belong to the first cluster, whereas the second data block compiles the data of the panelists in the second cluster. Like De Roover et al. [8], we preprocessed the data by standardizing the attributes of each panelist and extracted two components.

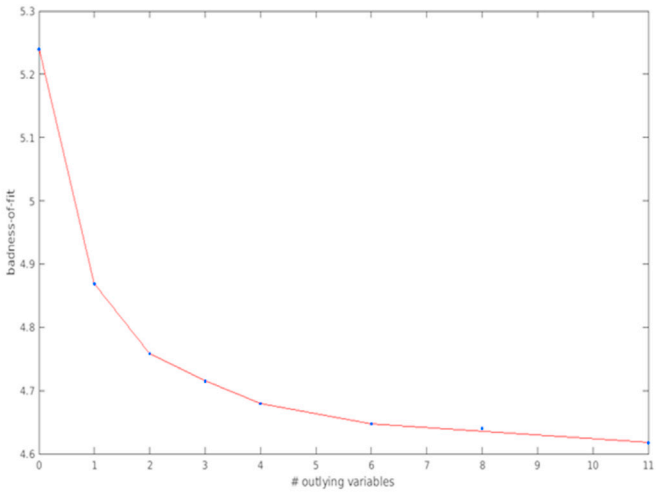
To decide on the α value and thus on the number of outlying variables, we applied the model selection procedure described in section 3.3.3. This resulted in the comparison of the 8 solutions shown in panel (a) of Fig. 1. In panel b of this Figure, we see that the badness-of-fit value decreases when the number of outlying variables increases, as expected, and that this decrease begins to level off after allowing for one outlying variable. Indeed, after discarding the solutions for which the individual slopes are less steep than the overall slope, only the solutions with 2 and 1 outlying variables remain. Out of these two solutions, the solution with one outlying variable has the highest *st* value; therefore, we retain this solution.

Table 3 shows the Varimax rotated ONVar loadings (see section 3.3.1), as well as the Varimax rotated SCA-ECP ones. Table 3 also presents the separate PCAs loadings, that were orthogonally Procrustes rotated towards the SCA-ECP ones. The non-outlying attributes 'color', 'moisture', 'tough' and 'salt-t' have strongly positive loadings on the first ONVar component, while 'sweet-t' and 'other-t' have strongly negative loadings. This loading pattern makes sense as salt gives a tougher structure to bread, causes moisture retention, and masks bad flavors. Also, it is quite intuitive that the 'salt-t' and 'sweet-t' attributes have loadings with opposite signs. Therefore this component can be labeled 'salty'. We label he second component of the ONVar solution 'yeast' as it is characterized by strongly positive loadings of 'yeast-od' and 'yeast-t'

Table 3
The orthogonally rotated SCA-ECP loadings, block-specific PCA and block-specific ONVar loadings with 1 outlying variable for the bread data, where the first block consists of the first, second and seventh panelists and the second block comprises the rest of the panelists. The block -specific PCA solutions are orthogonally procrustes rotated towards the varimax rotated SCA-ECP loadings. The loadings with an absolute value higher than .40 are printed in bold. The outlying variable detected by ONVar is highlighted with a grey background.

Attributes	SCA-ECP		PCA				ONVar			
	B		B1		B2		B1		B2	
Bread-od	.31	.06	-.20	-.20	**−50**	.16	.27	.06	.27	.06
Yeast-od	.10	**.73**	-.05	**.87**	.15	**.68**	.06	**.74**	.06	**.74**
Off-flav	**-.50**	**-.51**	-.31	**-.43**	**-.52**	**-.50**	**-.46**	**-.53**	**-.46**	**-.53**
Color	**.41**	.12	.38	-.19	.35	**.42**	**.40**	.14	**.40**	.14
Moisture	**.79**	.21	**.80**	.24	**.75**	.19	**.78**	.25	**.78**	.25
Tough	**.81**	-.23	**.78**	-.37	**.85**	-.10	**.82**	-.19	**.82**	-.19
Salt-t	**.80**	-.38	**.76**	**-.43**	**.81**	-.35	**.80**	-.34	**.80**	-.34
Sweet-t	**-.81**	-.15	**-.80**	-.08	**-.82**	-.14	**-.82**	-.18	**-.82**	-.18
Yest-t	-.03	**.80**	.02	**.84**	-.08	**.75**	-.04	**.80**	-.04	**.80**
Other-t	**-.66**	**-.44**	**-.60**	**-.67**	**-.68**	-.24	-.63	**-.47**	**-.63**	**-.47**
Total	.25	.08	**-.69**	.05	**.75**	-.01	**-.64**	.02	**.74**	.09

Alpha	#outl.	Fit value	Partition vector	st
.00	11	4.61	1 1 1 1 1 1 1 1 1 1 1	0
.00	8	4.64	1 1 1 1 1 1 0 0 0 1 1	
.01	6	4.65	1 1 1 1 0 0 0 0 0 1 1	2.64
.02	4	4.68	1 0 0 1 0 0 0 0 0 1 1	2.25
.04	3	4.72	1 0 0 1 0 0 0 0 0 0 1	1.20
.05	*2*	4.76	1 0 0 0 0 0 0 0 0 0 1	2.56
.10	***1***	**4.87**	**0 0 0 0 0 0 0 0 0 0 1**	**3.39**
.28	0	5.24	0 0 0 0 0 0 0 0 0 0 0	0



a) b)

Fig. 1. ONVar results for the cluster-based analysis of the bread data, where the first block consists of the first, second and seventh panelists and the second block comprises the rest of the panelists. a) For each number of outlying variables, we show the α-value that yields the lowest badness-of-fit value, and the corresponding partition vector. After removing the solutions with a smaller-than-overall slope, only the solutions with 2 and 1 outlying variables will be considered, leading to the detection of one outlying variable (solution printed in bold). b) CHull plot of the solutions in panel a. The red line denotes the lower boundary of the convex hull. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

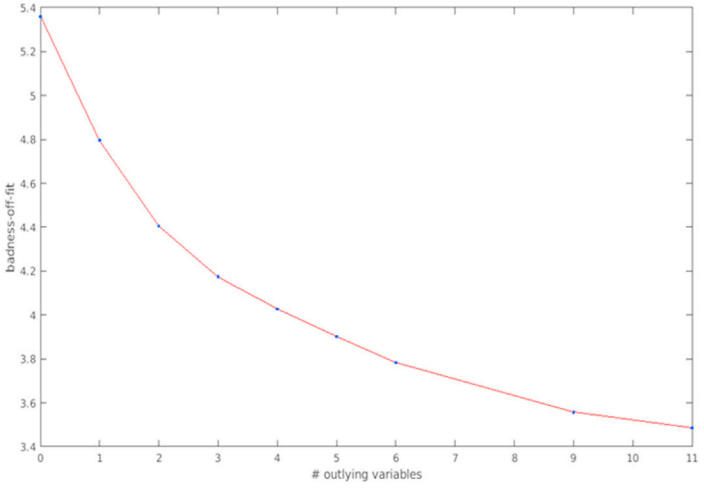
and by strongly negative loadings of 'off-flav' and 'other-t'. The latter might indicate that yeast has such a dominant taste and odor that other tastes and smells are hard to notice. The outlying attribute 'total' has a *negative* loading on the first component in the first block, while it has a *positive* loading on the very same component in the second block. While most attributes measure specific qualities of the bread, the attribute 'total' is more general and thus leaves more room for variation and personal interpretation of the panelist. Specifically, our results might indicate that when it comes to generally liking the bread, the main difference between the two groups of panelists depends on the saltiness of the bread, which might be a personal preference rather than conveying the general quality of the bread. One can speculate that this is the reason why the attribute 'total' is outlying across the two groups of the panelists.

The separate PCAs and SCA-ECP components can be interpreted in a similar manner as the ONVar components as they strongly resemble the ONVar loadings. Yet, using SCA-ECP we miss out on the different use of the attribute 'total'; note that the SCA-ECP loadings of this attribute lie in between the ONVar ones. On the other hand, the separate PCA loadings do not clearly show that the attribute 'total' stands out, as one discerns multiple loading differences. Thus, the advantage that ONVar offers over these other approaches lies in identifying and modeling the outlying variable in a clear and concise manner.

5.2. Outlying variables across all the panelists

The previous analyses are somewhat limited in case one is interested in individual differences between the panelists as it ignores potential, more subtle differences between the panelists in the same cluster. Since

Alpha	#outl.	Fit value	Partition vector	st
0.00	11	3.49	1 1 1 1 1 1 1 1 1 1 1	0
0.00	9	3.56	1 1 1 1 1 1 0 0 1 1 1	2.10
0.07	6	3.78	1 0 1 1 0 0 0 0 1 1 1	1.58
0.11	5	3.90	1 0 1 0 0 0 0 0 1 1 1	1.05
0.12	4	4.03	1 0 1 0 0 0 0 0 0 1 1	1.18
0.13	*3*	4.17	1 0 1 0 0 0 0 0 0 0 1	1.57
0.19	***2***	**4.41**	**1 0 0 0 0 0 0 0 0 0 1**	**1.69**
0.29	*1*	4.80	0 0 0 0 0 0 0 0 0 0 1	1.44
0.37	*0*	5.36	0 0 0 0 0 0 0 0 0 0 0	0



a) b)

Fig. 2. a) ONVar results for the bread data, when treating the data of each panelist as a separate block. For each number of outlying variables, we show the α-value that yields the lowest badness-of-fit value, and the corresponding partition vector. After removing the solutions with a smaller-than-overall slope, only the solutions with 3, 2 and 1 outlying variables will be considered, leading to the detection of two outlying variables (solution printed in bold). b) CHull plot of the solutions in panel a. The red line denotes the lower boundary of the convex hull. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

OnVar can handle more than two blocks, we can easily run a second set of analyses, considering the data of each panelist as a separate data block. In order to decide on the number of outlying variables, we again applied the model selection procedure (see section 3.3.3), which resulted in the comparison of the 9 solutions shown in panel (a) of Fig. 2. The solutions with 3, 2 and 1 outlying variables have an individual slope that is steeper than the overall slope and will thus be considered. The solution with 2 outlying variables is picked as the solution with the best fit-complexity balance. The corresponding panel-specific loading structures (Table 4) show that the attributes 'bread odor' and 'total' are considered outlying. This panelist-based ONVar solution is consistent with the cluster-based ONVar solution, in that the additional outlying attribute 'bread odor' was the second most outlying variable in the cluster-based analyses (panel a of Fig. 1). Further, the loadings of the non-outlying variables in Table 4 are very similar to the non-outlying variable patterns displayed in Table 3. Regarding the outlying variables, we see that 'bread odor' is associated with yeast odor, for some of the panelists, and with the 'salty' component for others. Since we do not expect the trained panelists to interpret the attributes this differently, OnVar can be used to pinpoint potential points to improve in further training. For example, in this case one can try to clarify more what is meant by the 'bread odor' and 'total' attributes.

6. Discussion

We started this paper by describing the state-of-the art when it comes to modeling multivariate multi-block data with component-analysis based techniques. Specifically, existing component analysis approaches for such data fall into two categories: on the one hand, we have simultaneous approaches that restrict the loading matrices to be the same across the blocks [10,11] and, on the other hand, separate PCAs that allow the loadings of all the variables to vary across the blocks [9]. As both approaches make it hard to pinpoint outlying variables, we proposed the new ONVar approach, which partitions the variables into outlying and non-outlying ones and accounts for their status in the loadings. ONVar bridges the gap between SCA-ECP and separate PCAs in that the loadings of the non-outlying variables are restricted to be the same across the blocks, whereas those of the outlying variables may differ.

The results of the simulation study revealed that ONVar performs well in finding the outlying variables and outperforms LBCM. The effects of the manipulated data are in line with existing simulation studies on component analysis methods [16,18,35], as they indicated that the problem becomes more difficult as the number of outlying variables and the amount of error increases and the number of non-outlying variables, the number of observations, and the outlyingness decreases. In addition, the simulation study showed that increasing the number of blocks also increased the likelihood that all the outlying variables will be detected correctly. Moreover, we compared how well ONVar, SCA-ECP and separate PCAs recovered the loadings, again confirming our predictions: SCA-ECP is too strict if the data contain outlying variables, while PCA slightly overfits the data because this approach focuses on a single data block each time.

We see at least five points of discussion. First, one might wonder how applied researchers should use the ONVar results. What to do if outlying variables are detected in the data and one is not just interested in modeling the data in the blocks correctly, but also wants to further correlate the component scores with external variables, for instance, to establish construct validity [5,38]? This brings us back to approaches like LBCM that suggest to remove the outlying variables from the data until the components extracted from the remaining variables are the same across the blocks, before proceeding with further analyses [16–18]. This strategy is defendable if we have a relatively large number of non-outlying variables and a few outlying ones only. However, if the number of outlying variables grows large, results may become quite unstable as only a few variables remain on which to base the estimation of the shared loadings. Further research on this topic is needed. Such research might give specific guidelines about how to handle outlying variables given specific data characteristics. We think that in some scenarios it is even possible to have the best of both worlds. In particular, if we could indicate that some of the variables are outlying, but that taking this into account does not significantly alter the interpretation of the components, this would definitely make some applications less complicated.

Second, the current ONVar algorithm does not allow the number of non-outlying variables to be smaller than the number of components. One may wonder if this restriction can be relaxed. In fact, from a technical point of view, the algorithm can be adjusted so that this is not an issue any more. The latter could be achieved by switching steps 2.a and 2.b in the algorithm, and thus by first modeling the outlying part of the data, followed by estimating the non-outlying part. However, we think this would be unwise, as it will raise conceptual issues from a substantive point of view. In particular, if the components differ across the blocks (i.e., they are now defined by the outlying part), why would it be useful to restrict some variables to have the same loadings, if these loadings express relations to components that are themselves different? In the end, we think that the concept of an outlying variable only makes sense in those settings where most of the variables have a similar correlation structure across the blocks. However, in this paper we have not studied the critical ratio of outlying to non-outlying variables. Building on concepts from robust statistics, it would be beneficial to study the breakdown point of ONVar in more detail [39,40]. For those datasets where one expects most variables to have different loadings across the blocks, we recommend to simply run separate PCAs.

Third, one might think of alternative model estimation and selection procedures that circumvent the regularization approach with a penalized number of outlying variables. This would imply developing a different algorithm in which the number of outlying variables v is fixed and the algorithm estimates which v variables are then singled out as outlying. Developing such an algorithm requires some thinking however, because the possible ways of choosing v from J variables can quickly become very large. Model selection would be simpler, however, because tuning the α value is not necessary anymore. In some cases, researchers might even have an a priori idea regarding the number of outlying variables. One should keep in mind, however, that the current model selection procedure performs very well and is efficient in terms of the number of estimated models, even for a large number of variables, when the non-outlying/outlying distinction is clear-cut (i.e., when raising α quickly makes all variables non-outlying).

Fourth, to handle a high number of blocks, it might make sense to develop a clusterwise ONVar extension, building on the work of [35,41,42]. Such a model would simultaneously perform a clustering of the blocks, partition the variables into nonoutlying and outlying ones, and estimate the associated loadings of the variables. Such an analysis would focus on the biggest loadings differences for the outlying variables. While such a model would be more parsimonious, model selection would be intricate, as one should also decide on the number of clusters for the blocks.

Finally, in the current implementation of ONVar, all variables are treated equally in that they a priori have an equal chance to be labeled as outlying. In practice, applied researchers might have hypotheses about which variables might behave differently, however. For instance, in sensory studies, there might be attributes that are notoriously hard to train. ONVar can be easily adjusted to take such a priori suspicions into account. Indeed, one can for instance simply fix the status of the variable in the analysis, implying that this variable is disregarded when re-estimating the partition vector in step 3 of the algorithm. To test whether it is reasonable to fix a specific variable as non-outlying or as outlying, one could potentially look at fit indices as is done in confirmatory factor analysis when running measurement invariance tests [43–45].

Table 4
ONVar results for the bread data, considering the data of each panelist as a separate data block. The loadings with an absolute value higher than 0.40 are printed in bold. The two outlying variables detected by ONVar are highlighted with a grey background.

Attributes	B1		B2		B3		B4		B5		B6		B7		B8	
Bread-od	**-.64**	**-.65**	**.59**	**-.41**	**.35**	**-.15**	**.50**	**.61**	**.56**	**-.06**	**.11**	**.83**	**-.36**	**.51**	**.82**	**-.17**
Yeast-od	.06	**.75**	.06	**.75**	.06	**.75**	.06	**.75**	.06	**.75**	.06	**.75**	.06	**.75**	.06	**.75**
Off-flav	**-.42**	**-.51**	**-.42**	**-.51**	**-.42**	**-.51**	**-.42**	**-.51**	**-.42**	**-.51**	**-.42**	**-.51**	**-.42**	**-.51**	**-.42**	**-.51**
Color	**.40**	.15	**.40**	.15	**.40**	.15	**.40**	.15	**.40**	.15	**.40**	.15	**.40**	.15	**.40**	.15
Moisture	**.77**	.23	**.77**	.23	**.77**	.23	**.77**	.23	**.77**	.23	**.77**	.23	**.77**	.23	**.77**	.23
Tough	**.84**	-.17	**.84**	-.17	**.84**	-.17	**.84**	-.17	**.84**	-.17	**.84**	-.17	**.84**	-.17	**.84**	-.17
Salt-t	**.81**	-.30	**.81**	-.30	**.81**	-.30	**.81**	-.30	**.81**	-.30	**.81**	-.30	**.81**	-.30	**.81**	-.30
Sweet-t	**-.81**	-.21	**-.81**	-.21	**-.81**	-.21	**-.81**	-.21	**-.81**	-.21	**-.81**	-.21	**-.81**	-.21	**-.81**	-.21
Yest-t	-.07	**.74**	-.07	**.74**	-.07	**.74**	-.07	**.74**	-.07	**.74**	-.07	**.74**	-.07	**.74**	-.07	**.74**
Other-t	**-.62**	**-.46**	**-.62**	**-.46**	**-.62**	**-.46**	**-.62**	**-.46**	**-.62**	**-.46**	**-.62**	**-.46**	**-.62**	**-.46**	**-.62**	**-.46**
Total	**-.38**	**-.64**	**-.63**	**-.21**	**.95**	**.09**	**.38**	**.76**	**.48**	**.47**	**.72**	**-.13**	**-.72**	**.56**	**.82**	**-.34**

Author statement

Dr. Sopiko Gvaladze: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft, Writing – review & editing. Dr.Kim De Roover: Conceptualization, Methodology, Software, Validation, Formal analysis, Data curation, Writing – review & editing. Prof. Dr. Eva Ceulemans: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft, Writing – review & editing, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The research leading to the results reported in this paper was supported by the Research Fund of KU Leuven (C14/19/054). For the simulations, we used the infrastructure of the VSC—Flemish Supercomputer Center, funded by the Hercules Foundation and the Flemish Government—department EWI.

References

[1] I. Van Mechelen, E. Ceulemans, Component-and factor-based models for data fusion in the behavioral sciences, Proc. IEEE 103 (9) (2015) 1621–1634.
[2] J. Stouten, E. Ceulemans, M.E. Timmerman, A. Van Hiel, Tolerance of justice violations: the effects of need on emotional reactions after violating equality in social dilemmas, J. Appl. Soc. Psychol. 41 (2) (2011) 357–380.
[3] E. Diener, C. Kim-Prieto, C. Scollon, Colleagues, International Collage Survey 2001, 2001.
[4] P. Kuppens, E. Ceulemans, M.E. Timmerman, E. Diener, C.H.U. Kim-Prieto, Universal intracultural and intercultural dimensions of the recalled frequency of emotional experience, J. Cross Cult. Psychol. 37 (5) (2006) 491–515.
[5] F. Ozcanli, E. Ceulemans, D. Hermans, L. Claes, B. Mesquita, Obsessions across two cultures: a comparison of Belgian and Turkish non-clinical samples, Front. Psychol. 10 (2019) 657.
[6] K. Van Deun, A.K. Smilde, L. Thorrez, H.A.L. Kiers, I. Van Mechelen, Identifying common and distinctive processes underlying multiset data, Chemometr. Intell. Lab. Syst. 129 (2013) 40–51.
[7] R. Bro, Multi-way Analysis in the Food Industry-Models, Algorithms, and Applications, Ph.D. Thesis, University of Amsterdam, 1997.
[8] K. De Roover, M.E. Timmerman, I. Van Mechelen, E. Ceulemans, On the added value of multiset methods for three-way data analysis, Chemometr. Intell. Lab. Syst. 129 (2013) 98–107.
[9] I.T. Jolliffe, Principal component analysis, in: Encyclopedia of Statistics in Behavioral Science, Americal Cancer Society, 2005.
[10] J.M. Ten Berge, H.A. Kiers, V. Van der Stel, Simultaneous components analysis, Statis. Appl. 4 (4) (1992) 277–392.
[11] M.E. Timmerman, H.A. 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 (1) (2003) 105–121.
[12] T. Næs, P.B. Brockhoff, O. Tomic, Statistics for Sensory and Consumer Science, John Wiley & Sons, 2011.
[13] B.M. Byrne, R.J. Shavelson, B. Muthén, Testing for the equivalence of factor covariance and mean structures: the issue of partial measurement invariance, Psychol. Bull. 105 (2) (1989) 456–466.
[14] W. Meredith, J.A. Teresi, An essay on measurement and factorial invariance, Med. Care (2006) S69–S77.
[15] K.F. Widaman, S.P. Reise, Exploring the Measurement Invariance of Psychological Instruments: Applications in the Substance Use Domain, 1997.
[16] K. De Roover, M.E. Timmerman, E. Ceulemans, How to detect which variables are causing differences in component structure among different groups, Behav. Res. Methods 49 (1) (2017) 216–229.
[17] S. Gvaladze, K. De Roover, F. Tuerlinckx, E. Ceulemans, Detecting outlying variables in multi-block data: a comparison of different loading similarity coefficients, J. Chemom. 35 (2) (2021).
[18] S. Gvaladze, K. De Roover, F. Tuerlinckx, E. Ceulemans, Detecting which variables alter component interpretation across multiple groups: a resampling-based method, Behav. Res. Methods (2019) 1–28.
[19] H. Dl, Applied Multivariate Statistical Concepts, Routledge, New York, NY, 2016.
[20] H. Abdi, Congruence: Congruence Coefficient, RV Coefficient, and Mantel Coefficient, 2010, pp. 222–229.
[21] E.C. Davenport Jr., Significance testing of congruence coefficients: a good idea? Educ. Psychol. Meas. 50 (2) (1990) 289–296.
[22] W.J. Krzanowski, Between-groups comparison of principal components, J. Am. Stat. Assoc. 74 (367) (1979) 703–707.
[23] L.R. Tucker, "A Method for Synthesis of Factor Analysis Studies," (No. PRS-984), EDUCATIONAL TESTING SERVICE, PRINCETON NJ, 1951.
[24] K. De Roover, J.K. Vermunt, E. Ceulemans, Mixture multigroup factor analysis for unraveling factor loading noninvariance across many groups, Psychol., 2020.
[25] E. Ceulemans, T.F. Wilderjans, H.A. Kiers, M.E. Timmerman, MultiLevel simultaneous component analysis: a computational shortcut and software package, Behav. Res. Methods 48 (3) (2016) 1008–1020.
[26] K. De Roover, E. Ceulemans, M.E. Timmerman, How to perform multiblock component analysis in practice, Behav. Res. Methods 44 (1) (2012) 41–56.
[27] H.A. Kiers, J.M. 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 (1) (1994) 109–126.
[28] H.F. Kaiser, The varimax criterion for analytic rotation in factor analysis, Psychometrika 23 (3) (1958) 187–200.
[29] E. Ceulemans, H.A. 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 (1) (2006) 133–150.
[30] T.F. Wilderjans, E. Ceulemans, K. Meers, CHull: a generic convex-hull-based model selection method, Behav. Res. Methods 45 (1) (2013) 1–15.
[31] R.B. Cattell, The scree test for the number of factors, Multivariate Behav. Res. 1 (2) (1966) 245–276.
[32] E. Ceulemans, H.A. 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 (1) (2006) 133–150.
[33] T.F. Wilderjans, E. Ceulemans, K. Meers, CHull: a generic convex-hull-based model selection method, Behav. Res. Methods 45 (1) (2013) 1–15.
[34] J.L. Horn, A rationale and test for the number of factors in factor analysis, Psychometrika 30 (2) (1965) 179–185.
[35] 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 (1) (2013) 81–102.
[36] 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 (1) (2012) 100–119.
[37] 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 (4) (2013) 648–668.
[38] F. Ozcanli, Cultural Differences in Obsessions: Empirical Studies with Belgian and Turkish Samples, Belgium, Leuven, 2019.

[39] M. Daszykowski, K. Kaczmarek, Y. Vander Heyden, B. Walczak, Robust statistics in data analysis—a review: basic concepts, Chemometr. Intell. Lab. 85 (2) (2007) 203–219.
[40] P.J. Huber, Robust Statistics, John Wiley & Sons, Hoboken, New Jersey, 2004.
[41] 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 (1) (2012) 100–119.
[42] K. De Roover, M.E. Timmerman, J. De Leersnyder, B. Mesquita, E. Ceulemans, What's hampering measurement invariance: detecting non-invariant items using clusterwise simultaneous component analysis, Front. Psychol. 5 (2014) 604.
[43] G.W. Cheung, R.B. Rensvold, Evaluating goodness-of-fit indexes for testing measurement invariance, Struct. Equ. Model. 9 (2) (2002) 233–255.
[44] T.L. Milfont, R. Fischer, Testing measurement invariance across groups: applications in cross-cultural research, Int. J. Psychol. Stud. 3 (1) (2010) 111–130.
[45] R. Van de Schoot, P. Lugtig, J. Hox, A checklist for testing measurement invariance, Eur. J. Dev. Psychol. 9 (4) (2012) 486–492.