Advances in Data Analysis and Classification (2022) 16:1001–1037
https://doi.org/10.1007/s11634-021-00475-2

REGULAR ARTICLE

Least-squares bilinear clustering of three-way data

Pieter C. Schoonees[1] · **Patrick J. F. Groenen**[2] · **Michel van de Velden**[2]

Received: 11 September 2020 / Revised: 16 August 2021 / Accepted: 1 October 2021 /
Published online: 15 November 2021

Abstract

A least-squares bilinear clustering framework for modelling three-way data, where each observation consists of an ordinary two-way matrix, is introduced. The method combines bilinear decompositions of the two-way matrices with clustering over observations. Different clusterings are defined for each part of the bilinear decomposition, which decomposes the matrix-valued observations into overall means, row margins, column margins and row–column interactions. Therefore up to four different classifications are defined jointly, one for each type of effect. The computational burden is greatly reduced by the orthogonality of the bilinear model, such that the joint clustering problem reduces to separate problems which can be handled independently. Three of these sub-problems are specific cases of k-means clustering; a special algorithm is formulated for the row–column interactions, which are displayed in clusterwise biplots. The method is illustrated via an empirical example and interpreting the interaction biplots are discussed. Supplemental materials for this paper are available online, which includes the dedicated R package, `lsbclust`.

Keywords Three-way data · Bilinear decomposition · k-Means cluster analysis · Least-squares estimation · Biplots

Mathematics Subject Classification 62H30 · 15A18

✉ Pieter C. Schoonees
schoonees@rsm.nl

Patrick J. F. Groenen
groenen@ese.eur.nl

Michel van de Velden
vandevelden@ese.eur.nl

[1] Department of Marketing Management, Rotterdam School of Management, Erasmus University, Rotterdam, The Netherlands

[2] Econometric Institute, Erasmus School of Economics, Erasmus University Rotterdam, Rotterdam, The Netherlands

1 Introduction

Multiway data, a generalization of the familiar two-way samples-by-variables data matrix, is becoming more common in a variety of fields. In computer vision applications, images are stored as three-way arrays, or third-order tensors, with rows and columns representing pixel locations and the third way (tubes) representing different color channels (e.g., Krizhevsky et al. 2012). Videos constitute fourth-order tensors, since they capture a sequence of such images over time (e.g., Abu-El-Haija et al. 2016). In the social sciences, a marketing research survey asking multiple individuals to rate several products on various characteristics using a Likert scale generates a three-way array of rating scores (e.g., DeSarbo et al. 1982). Similar research designs are commonly used in sensometrics (e.g., Cariou et al. 2021). Other applications include: high-throughput molecular data in bioinformatics (e.g., Lonsdale et al. 2013); spectroscopic data in chemometrics (e.g., Faber et al. 2003; Bro 2006); and neuroimaging data, collected using electroencephalography (EEG) or functional magnetic resonance imaging (fMRI), for example (e.g., Genevsky and Knutson 2015).

A taxonomy of measurement data is given by Carroll and Arabie (1980), where a *mode* is defined as "a particular class of entities" associated with a data array, such as stimuli, subjects or scale items; a three-way array may have up to three modes. Kiers (2000) introduced standardized notation and terminology for multiway analysis, while Kroonenberg (2008) is devoted to multiway data analysis methodology.

Performing cluster analysis on one or more of the modes of a three-way data array $\underline{\mathbf{X}}$ is an important problem in multiway data analysis. Statistical research into three-way clustering can be broadly divided into two streams. The application of finite mixture models for clustering three-way data has been studied in, amongst others, Basford and McLachlan (1985), Hunt and Basford (1999), Vermunt (2007), Viroli (2011), Meulders and De Bruecker (2018), Gallaugher and McNicholas (2020a, b). Whereas these rely on maximum likelihood estimation, another research stream have focused on nonparametric data approximation methods mainly via least-squares estimation. The papers of Vichi (1999), Rocci and Vichi (2005), Vichi et al. (2007), Papalexakis et al. (2013), Wilderjans and Ceulemans (2013), Llobell et al. (2019), Llobell et al. (2020) and Cariou et al. (2021) belong to this stream, as does our paper. An approach often used in this stream is to generalize a three-way data decomposition, such as Tucker's three-mode factor analysis (TUCKER3; Tucker 1966) or the CANDECOMP/PARAFAC decomposition (CP; Carroll and Chang 1970; Harshman 1970; Hitchcock 1927), by adding clustering over one or more of the modes.

Here we propose a new method—called least-squares bilinear clustering (or LSBCLUST)—derived in a similar manner but from a different starting point. In contrast to starting with a three-way decomposition, we start with a bilinear (or biadditive) decomposition of the matrix slices $\mathbf{X}_i$ ($i = 1, \ldots, I$) of $\underline{\mathbf{X}}$ for the mode (by convention, the first mode) over which we want to cluster (e.g., Denis and Gower 1994). An example of a bilinear approximation of a matrix $\mathbf{X}_i$ is

$$\mathbf{X}_i \approx m\mathbf{1}\mathbf{1}^{'} + \boldsymbol{a}\mathbf{1}^{'} + \mathbf{1}\boldsymbol{b}^{'} + \boldsymbol{c}\boldsymbol{d}^{'}, \tag{1}$$

where $\mathbf{1}$ denotes a column vector of ones. The terms on the right-hand side of (1) can be interpreted as the grand mean (m), row main effects ($\boldsymbol{a}$), column main effects

($\boldsymbol{b}$) and row–column interactions ($\boldsymbol{cd}'$), respectively, and are subject to appropriate identification restrictions.

By adding one or more sets of clusters to the effects on the right-hand side of Eq. (1), and by decomposing all matrix slices at the same time, the LSBCLUST method ensures that the estimated effects are equal for observations in the same cluster. The choice of bilinear decomposition has two interesting features. First, it allows the results to be displayed graphically using biplots (Gower et al. 2011; Gower and Hand 1996; Gabriel 1971), and other standard graphs. Second, it provides the possibility to have a different set of clusters for each of the four types of effects. This can be useful in cases where interesting differences can be expected not only with respect to the row–column interactions, but also with respect to the grand mean and main effects on the right-hand side of (1). We will discuss this aspect further in Sect. 3.

The main contribution of this paper is the introduction of the LSBCLUST model and loss function, together with an algorithm for estimating the parameters. We also provide an implementation of this algorithm and its related graphical procedures in an R (R Core Team 2020) package `lsbclust` (available on the Comprehensive R Archive Network, or CRAN), while supplemental materials illustrate its use. The method can be used as a complement to or replacement of other three-way data analysis procedures.

The remainder of the paper is structured as follows. Section 2 introduces the basic model and loss function used, which Sect. 3 augments with clustering to arrive at the LSBCLUST formulation. Section 4 shows how to simplify the loss function, and Sect. 5 discusses our algorithm for estimating the parameters. Enhancements to the basic method are discussed in Sect. 6, including using different bilinear models and aspects of biplot construction. The results of a simulation study is reported in Sect. 7. An empirical example is presented in Sects. 8, and 9 concludes.

Before we introduce LSBCLUST in more detail, we provide more detail on related methods in Sect. 1.1.

1.1 Related methods

The most prominent multiway data analysis methods, which include the TUCKER3 and CP decompositions, seek to generalize the singular value decomposition (SVD) of a matrix to the multiway case. Kolda and Bader (2009) provide an excellent review of these tensor decomposition methods and their applications across diverse fields. De Silva and Lim (2008) discusses mathematical aspects of generalizing the Eckart–Young theorem (Stewart 1993; Eckart and Young 1936; Schmidt 1907) to higher-order tensors, while Kiers and Van Mechelen (2001) discuss and illustrate the application of the TUCKER3 decomposition.

Three-mode factor analysis (Tucker 1966; Kroonenberg and de Leeuw 1980), commonly referred to as TUCKER3, is the most general widely-used three-way method. Let $\mathbf{X}_{J,KI} = \left[\mathbf{X}_1\ \mathbf{X}_2\ \cdots\ \mathbf{X}_I\right]$, and let the component matrices $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$ be low-dimensional columnwise orthonormal configurations for the first, second and third ways of $\underline{\mathbf{X}}$ with dimensions P, Q and R respectively. Also, let $\underline{\mathbf{H}} : P \times Q \times R$ be the so-called core array, which gives the interactions between the elements of the component matrices. The TUCKER3 model has loss function

$$L_{\text{T3}}(\mathbf{B}, \mathbf{C}, \mathbf{D}, \underline{\mathbf{H}}) = \left\| \mathbf{X}_{J,KI} - \mathbf{C}\mathbf{H}_{Q,RP}(\mathbf{B} \otimes \mathbf{D})' \right\|^2 = \sum_{i=1}^{I} \left\| \mathbf{X}_i - \sum_{p=1}^{P} b_{ip}\mathbf{C}\mathbf{H}_p\mathbf{D}' \right\|^2, \quad (2)$$

where b_{ip} denotes an element of $\mathbf{B}$, $\mathbf{H}_p : Q \times R$ is a slice of $\underline{\mathbf{H}}$ along the first mode, $\|\cdot\|$ is the Frobenius norm and $\otimes$ is the Kronecker product. Interpreting a TUCKER3 solution involves interpreting the three component matrices and the core. No clustering is performed, and (2) is typically minimized by alternating least squares (ALS; Kroonenberg and de Leeuw 1980).

The CP decomposition (Carroll and Chang 1970; Harshman 1970) is a restricted version of TUCKER3 where $P = Q = R$ and $\underline{\mathbf{H}}$ is replaced by $\underline{\mathbf{E}}$. Here $\underline{\mathbf{E}}$ contains elements $e_{pqr} = 1$ if $p = q = r$ and $e_{pqr} = 0$ otherwise. This implies that each component (column) in $\mathbf{B}$ is related to a single component in $\mathbf{C}$ and $\mathbf{D}$, and vice versa, while in TUCKER3 all components in one mode are related to all other components in the other modes. The CP loss function is

$$L_{\text{CP}}(\mathbf{B}, \mathbf{C}, \mathbf{D}) = \left\| \mathbf{X}_{J,KI} - \mathbf{C}\mathbf{E}_{P,PP}(\mathbf{B} \otimes \mathbf{D})' \right\|^2 = \sum_{i=1}^{I} \left\| \mathbf{X}_i - \mathbf{C}\,\text{diag}(\boldsymbol{b}_i)\mathbf{D}' \right\|^2. \quad (3)$$

Here $\text{diag}(\boldsymbol{b}_i)$ is the diagonal matrix with the ith row of $\mathbf{B}$ on the diagonal, and $\mathbf{E}_{P,PP} = \left[\mathbf{E}_1\ \mathbf{E}_2\ \cdots\ \mathbf{E}_p\right]$ with $\mathbf{E}_p$ being the pth matrix slice of $\underline{\mathbf{E}}$.

Neither of these methods employ clustering along any of the modes, but Rocci and Vichi (2005) formulate such a variant of TUCKER3, namely T3CLUS. The idea is to replace $\mathbf{B}$ by the indicator matrix $\mathbf{G} : I \times U$, which simply indicates which of U clusters each of the I observations belongs to. The core array $\underline{\mathbf{H}}$ now represents the three-way array of cluster centroids in the reduced component space. The T3CLUS loss function is

$$L_{\text{T3CLUS}}(\mathbf{G}, \mathbf{C}, \mathbf{D}, \underline{\mathbf{H}}) = \sum_{i=1}^{I}\sum_{u=1}^{U} g_{iu} \left\| \mathbf{X}_i - \mathbf{C}\mathbf{H}_u\mathbf{D}' \right\|^2. \quad (4)$$

Another related approach is the clusterwise CP method of Wilderjans and Ceulemans (2013)—CPCLUS hereafter. The loss function for this approach is

$$L_{\text{CPCLUS}}(\mathbf{G}, \mathbf{B}, \{\mathbf{C}_u\}_{u=1}^{U}, \{\mathbf{D}_u\}_{u=1}^{U}) = \sum_{i=1}^{I}\sum_{u=1}^{U} g_{iu} \left\| \mathbf{X}_i - \mathbf{C}_u\,\text{diag}(\boldsymbol{b}_i)\mathbf{D}_u' \right\|^2. \quad (5)$$

In contrast to T3CLUS, here the component matrices are cluster-specific.

Table 1 gives a summary of the loss functions of these three-way decompositions, together with the LSBCLUST loss function for the interaction effects (where $\mathbf{J}$ denotes

Table 1 A summary of loss functions for the three-way methods discussed in Sect. 1.1

Method	Clustered	Loss function
Tucker3	✗	$\sum_{i=1}^{I} \left\| \mathbf{X}_i - \sum_{p=1}^{P} b_{ip} \mathbf{C}\mathbf{H}_p \mathbf{D}' \right\|^2$
T3clus	✓	$\sum_{i=1}^{I} \sum_{u=1}^{U} g_{iu} \left\| \mathbf{X}_i - \mathbf{C}\mathbf{H}_u \mathbf{D}' \right\|^2$
CP	✗	$\sum_{i=1}^{I} \left\| \mathbf{X}_i - \mathbf{C}\,\mathrm{diag}(\boldsymbol{b}_i)\mathbf{D}' \right\|^2$
CPclus	✓	$\sum_{i=1}^{I} \sum_{u=1}^{U} g_{iu} \left\| \mathbf{X}_i - \mathbf{C}_u\,\mathrm{diag}(\boldsymbol{b}_i)\mathbf{D}_u' \right\|^2$
lsbclust	✓	$\sum_{i=1}^{I} \sum_{u=1}^{U} g_{iu}^{(\mathrm{i})} \left\| \mathbf{J}\left(\mathbf{X}_i - \mathbf{C}_u \mathbf{D}_u'\right)\mathbf{J} \right\|^2$

See the text for references and an explanation of the notation used. For LSBCLUST, we include only the part of the loss function which relates to the row–column interactions. Here $\mathbf{J}$ denotes a centring matrix of the appropriate size

a centring matrix of the appropriate size). From the table it is clear that each method approximates the matrix slices $\mathbf{X}_i$ ($i = 1, \ldots, I$) in different ways. TUCKER3 and CANDECOMP/PARAFAC are symmetric with respect to all three modes, but do not involve clustering. In case all $\mathbf{X}_i$ are double-centred, T3CLUS approximates $\mathbf{X}_i$ in cluster u by $\mathbf{C}\mathbf{H}_u\mathbf{D}'$, as compared to $\mathbf{C}_u\mathbf{D}_u'$ for LSBCLUST. T3CLUS therefore requires the clusterwise approximation of $\mathbf{X}_i$ to lie in the same row and column subspaces across clusters, while LSBCLUST has no such restrictions. The LSBCLUST loss function is clearly a constrained version of CPCLUS which replaces $\mathrm{diag}(\boldsymbol{b}_i)$ with the identity matrix. This reduces the number of parameters and simultaneously enforces the same interpretation for all members of the same cluster. The additional constraint means that a single biplot can be used to interpret the interaction effects in a cluster, instead of requiring one plot per cluster member.

In the special case where $\underline{\mathbf{X}}$ contains indicator matrices for each subject, LSBCLUST has a strong connection to the latent-class bilinear multinomial logit (LC- BML) model of van Rosmalen et al. (2010). Their model was developed specifically to deal with response styles when analyzing two-way self-report survey data. More details on the LC- BML model, and its connection to LSBCLUST, are given in the supplemental materials.

The next section discusses the basic building blocks of LSBCLUST.

2 Basic model and loss function

Consider real-valued data collected in a three-way array $\underline{\mathbf{X}}$. For the marketing research survey example, the entry x_{ijk} of $\underline{\mathbf{X}}$ is the rating by person i of product j on characteristic k, with $i = 1, \ldots, I$; $j = 1, \ldots, J$; and $k = 1, \ldots, K$, respectively. Let $\mathbf{X}_i$ denote the $J \times K$ matrix of responses for subject i, which is also the ith (frontal) slice

of the array $\underline{\mathbf{X}}$. By convention, we single out the first way of $\underline{\mathbf{X}}$ by treating the $\mathbf{X}_i$ as observations, but the method could also be applied to the second or third way of $\underline{\mathbf{X}}$.

We derive the proposed LSBCLUST formulation by augmenting (with clusters; Sect. 3) the following least-squares loss function:

$$L(m, \boldsymbol{a}, \boldsymbol{b}, \mathbf{C}, \mathbf{D}) = \sum_{i=1}^{I} \left\| \mathbf{X}_i - \left(m\mathbf{1}_J\mathbf{1}_K' + \boldsymbol{a}\mathbf{1}_K' + \mathbf{1}_J\boldsymbol{b}' + \mathbf{C}\mathbf{D}' \right) \right\|^2. \tag{6}$$

Here $\|\cdot\|$ is the Frobenius norm. Moreover, $\mathbf{1}_K$ is the length-K vector of ones; below we will drop the subscript—as in $\mathbf{1}$—for simplicity. Equation (6) approximates each $\mathbf{X}_i$ using the same bilinear (or biadditive) model. This model contains a grand mean m, the row and column main effects $\boldsymbol{a}$ and $\boldsymbol{b}$ respectively, and row–column interactions $\mathbf{C}\mathbf{D}'$.

The matrices $\mathbf{C}$ and $\mathbf{D}$ are low-dimensional representations of the rows (products) and columns (characteristics) of the $\mathbf{X}_i$ matrices respectively, but after adjusting for main effects. Representing the interaction effects using such inner products permit these to be displayed in biplots if the dimensionality of $\mathbf{C}$ and $\mathbf{D}$ are low enough so that displays can be made (Gower et al. 2011; Gower and Hand 1996; Gabriel 1971). Therefore, the dimensionality is typically set to two, although values up to $\min\{J, K\} - 1$ are possible. To ensure uniqueness of the model, the usual sum-to-zero constraints $\boldsymbol{a}'\mathbf{1} = \boldsymbol{b}'\mathbf{1} = 0$ and $\mathbf{1}'\mathbf{C} = \mathbf{1}'\mathbf{D} = \mathbf{0}$ must be imposed. Additionally, the columns of $\mathbf{C}$ and $\mathbf{D}$ are required to be orthogonal (for more information, see Denis and Gower 1994, as well as Sect. 6.3). Model (6) has an analytical solution.

3 Capturing heterogeneity with clusters

Equation (6) applies the same parameters to the entire data set. To capture potential heterogeneity, we allow for the existence of a prespecified number of clusters in a single mode between which the parameters in the bilinear decomposition of the $\mathbf{X}_i$ matrices may vary. Moreover, since the bilinear decomposition itself has four different sets of parameters, we introduce four sets of clusters—one for each type of parameter. This allows the model to recognize that, for example, although $\mathbf{X}_i$ and $\mathbf{X}_{i'}$ $(i \neq i')$ are similar with respect to main effects, they could differ in the interaction effects (or vice versa).

Let $\mathbf{G}^{(\mathrm{o})}$ be the $I \times R$ matrix of cluster memberships for the grand mean effect m, which has $g_{ir}^{(\mathrm{o})} = 1$ if person i belongs to cluster r and $g_{ir}^{(\mathrm{o})} = 0$ otherwise $(r = 1, 2, \ldots, R)$. Similarly, $\mathbf{G}^{(\mathrm{r})}$ is the $I \times S$ matrix of cluster memberships for the row effects, $\mathbf{G}^{(\mathrm{c})}$ the $I \times T$ matrix of cluster memberships for the column effects, and $\mathbf{G}^{(\mathrm{i})}$ the $I \times U$ matrix of cluster memberships for the interaction effects. Now, by incorporating the clustering, the least-squares loss function becomes

$$
\begin{aligned}
&L(\mathbf{G}^{(\mathrm{o})}, \mathbf{G}^{(\mathrm{r})}, \mathbf{G}^{(\mathrm{c})}, \mathbf{G}^{(\mathrm{i})}, \boldsymbol{m}, \mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}) \\
&\quad = \sum_{i,r,s,t,u} g_{ir}^{(\mathrm{o})} g_{is}^{(\mathrm{r})} g_{it}^{(\mathrm{c})} g_{iu}^{(\mathrm{i})} \left\| \mathbf{X}_i - \left(m_r \mathbf{1}\mathbf{1}^{'} + \boldsymbol{a}_s \mathbf{1}^{'} + \mathbf{1}\boldsymbol{b}_t^{'} + \mathbf{C}_u \mathbf{D}_u^{'} \right) \right\|^2 \\
&\quad = \sum_{i,r,s,t,u} g_{ir}^{(\mathrm{o})} g_{is}^{(\mathrm{r})} g_{it}^{(\mathrm{c})} g_{iu}^{(\mathrm{i})} L(i|r,s,t,u), \qquad (7)
\end{aligned}
$$

with the summation over all observations (I) and clusters (R, S, T and U). Here $\boldsymbol{m}^{'} = [m_1 \ \cdots \ m_R]$, $\mathbf{A} = [\boldsymbol{a}_1 \ \cdots \ \boldsymbol{a}_S]$, $\mathbf{B} = [\boldsymbol{b}_1 \ \cdots \ \boldsymbol{b}_T]$, $\mathbf{C}^{'} = [\mathbf{C}_1^{'} \ \cdots \ \mathbf{C}_U^{'}]$ and $\mathbf{D}^{'} = [\mathbf{D}_1^{'} \ \cdots \ \mathbf{D}_U^{'}]$. Note that it is assumed implicitly that all low-rank decompositions are of the same rank P for all clusters. For identifiability, the same sum-to-zero constraints now apply to each cluster-specific set of parameters, i.e., $\mathbf{D}_u^{'}\mathbf{1} = \mathbf{0}$ for all $u = 1, \ldots, U$. The columns of all $\mathbf{C}_u$ and $\mathbf{D}_u$ matrices are orthogonal.

Equation (7) allows for a total of $RSTU$ clusters by clustering each $\mathbf{X}_i$ on four different types of effects at the same time. However, we show next that the joint clustering problem can be simplified into four separate clustering problems, significantly reducing the computational complexity since only $R + S + T + U$ clusters will need to be found.

4 Decomposing the loss function

To simplify the exposition, note that mathematically we can drop the sum-to-zero constraints from the formulation by introducing centering matrices of the form

$$
\mathbf{J}_c = \mathbf{I}_c - \frac{1}{c}\mathbf{1}_c\mathbf{1}_c^{'}, \qquad (8)
$$

for some positive integer c controlling the size of the matrix (which we duly omit below for simplicity). This is done by redefining the terms in the summation in (7) as

$$
L(i|r,s,t,u) = \left\| \mathbf{X}_i - \left(m_r \mathbf{1}\mathbf{1}^{'} + \mathbf{J}\boldsymbol{a}_s \mathbf{1}^{'} + \mathbf{1}\boldsymbol{b}_t^{'}\mathbf{J} + \mathbf{J}\mathbf{C}_u\mathbf{D}_u^{'}\mathbf{J} \right) \right\|^2, \qquad (9)
$$

such that the sum-to-zero constraints on the columns of $\mathbf{C}_u$ and $\mathbf{D}_u$, and on $\boldsymbol{a}_s$ and $\boldsymbol{b}_t$, are automatically enforced. For example, estimating the parameters in $\mathbf{J}\boldsymbol{a}_s$ is equivalent to estimating $\boldsymbol{a}_s$ subject to $\mathbf{1}^{'}\boldsymbol{a}_s = 0$. To simplify the notation, we redefine $\mathbf{A} = [\mathbf{J}\boldsymbol{a}_1 \ \cdots \ \mathbf{J}\boldsymbol{a}_S]$ to avoid writing $\mathbf{JA}$. The matrices $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$ are also redefined analogously.

We proceed to simplify (9) by first expanding the double-centred $\mathbf{J}\mathbf{X}_i\mathbf{J}$ into separate terms. Then, by adding two additional centering operators for the row and column means, $\mathbf{X}_i$ can be rewritten as

$$
\mathbf{X}_i = \frac{\mathbf{1}^{'}\mathbf{X}_i\mathbf{1}}{JK}\mathbf{1}\mathbf{1}^{'} + \frac{1}{J}\mathbf{1}\mathbf{1}^{'}\mathbf{X}_i\mathbf{J} + \frac{1}{K}\mathbf{J}\mathbf{X}_i\mathbf{1}\mathbf{1}^{'} + \mathbf{J}\mathbf{X}_i\mathbf{J}. \qquad (10)
$$

We can now associate each of the terms in (10) with the corresponding terms in the model (9):

$$L(i|r,s,t,u) = \left\| \left(\tfrac{1}{JK}\mathbf{1}'\mathbf{X}_i\mathbf{1} - m_r \right) \mathbf{1}\mathbf{1}' + \mathbf{1}\left(\tfrac{1}{J}\mathbf{1}'\mathbf{X}_i - \boldsymbol{b}_t' \right)\mathbf{J} + \mathbf{J}\left(\tfrac{1}{K}\mathbf{X}_i\mathbf{1} - \boldsymbol{a}_s \right)\mathbf{1}' + \mathbf{J}\left(\mathbf{X}_i - \mathbf{C}_u\mathbf{D}_u' \right)\mathbf{J} \right\|^2. \tag{11}$$

It can be shown (see the appendix in the supplemental materials) that the decomposition (11) is orthogonal such that

$$\begin{aligned} L(i|r,s,t,u) &= JK \left\| \left(\tfrac{1}{JK}\mathbf{1}'\mathbf{X}_i\mathbf{1} - m_r \right) \right\|^2 + J \left\| \left(\tfrac{1}{J}\mathbf{1}'\mathbf{X}_i - \boldsymbol{b}_t' \right)\mathbf{J} \right\|^2 \\ &\quad + K \left\| \mathbf{J}\left(\tfrac{1}{K}\mathbf{X}_i\mathbf{1} - \boldsymbol{a}_s \right) \right\|^2 + \left\| \mathbf{J}(\mathbf{X}_i - \mathbf{C}_u\mathbf{D}_u')\mathbf{J} \right\|^2 \\ &= L_{(\mathrm{o})}(i|r) + L_{(\mathrm{r})}(i|s) + L_{(\mathrm{c})}(i|t) + L_{(\mathrm{i})}(i|u). \end{aligned} \tag{12}$$

This equality follows from the fact that all the cross-products are zero. Importantly, the orthogonality leads to a great simplification in the clustering, since now the loss function (7) equals

$$\begin{aligned} &L(\mathbf{G}^{(\mathrm{o})}, \mathbf{G}^{(\mathrm{r})}, \mathbf{G}^{(\mathrm{c})}, \mathbf{G}^{(\mathrm{i})}, \boldsymbol{m}, \mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}) \\ &= JK \sum_{i=1}^{I}\sum_{r=1}^{R} g_{ir}^{(\mathrm{o})} \left\| \left(\tfrac{1}{JK}\mathbf{1}'\mathbf{X}_i\mathbf{1} - m_r \right) \right\|^2 \\ &\quad + K \sum_{i=1}^{I}\sum_{s=1}^{S} g_{is}^{(\mathrm{r})} \left\| \mathbf{J}\left(\tfrac{1}{K}\mathbf{X}_i\mathbf{1} - \boldsymbol{a}_s \right) \right\|^2 \\ &\quad + J \sum_{i=1}^{I}\sum_{t=1}^{T} g_{it}^{(\mathrm{c})} \left\| \left(\tfrac{1}{J}\mathbf{1}_J'\mathbf{X}_i - \boldsymbol{b}_t' \right)\mathbf{J} \right\|^2 \\ &\quad + \sum_{i=1}^{I}\sum_{u=1}^{U} g_{iu}^{(\mathrm{i})} \left\| \mathbf{J}\left(\mathbf{X}_i - \mathbf{C}_u\mathbf{D}_u' \right)\mathbf{J} \right\|^2 \\ &= L_{(\mathrm{o})}\left(\mathbf{G}^{(\mathrm{o})}, \boldsymbol{m}\right) + L_{(\mathrm{r})}\left(\mathbf{G}^{(\mathrm{r})}, \mathbf{A}\right) + L_{(\mathrm{c})}\left(\mathbf{G}^{(\mathrm{c})}, \mathbf{B}\right) + L_{(\mathrm{i})}\left(\mathbf{G}^{(\mathrm{i})}, \mathbf{C}, \mathbf{D}\right). \end{aligned} \tag{13}$$

The main consequence of (13) is that the joint clustering reduces to separate clusterings on the grand means, row main effects, column main effects and interactions respectively. This implies that each of these four parts can be treated independently under this loss function and model. It also gives mathematical justification for the procedure whereby all $\mathbf{X}_i$ are first double-centred to remove the overall, row and column margins, and then analyzed by minimizing $L_{(\mathrm{i})}\left(\mathbf{G}^{(\mathrm{i})}, \mathbf{C}, \mathbf{D}\right)$ to study the row–column interactions. If the researchers are interested in the grand mean, row or column marginal effects, these can be analyzed separately by minimizing the corresponding loss functions. These are frequently omitted when only the row–column interactions

Algorithm 1: ALS algorithm for minimizing $L_{(\mathrm{i})}\left(\mathbf{G}^{(\mathrm{i})}, \mathbf{C}, \mathbf{D}\right)$

1. Set $k = 0$ and $\Delta_0 = I$. Randomly initialize $\mathbf{G}^{(\mathrm{i})}$ as $\mathbf{G}_0$.
2. While $\Delta_k > 0$:
   (a) Compute cluster sizes $I_1, \ldots, I_U$ using $\mathrm{diag}(\mathbf{G}_k'\mathbf{G}_k)$.
   (b) For $u = 1$ to U:
      i. Compute the cluster mean $\overline{\mathbf{X}}_u = \frac{1}{I_u}\sum_{i=1}^{I} g_{kiu}\mathbf{X}_i$.
      ii. Compute $\mathrm{SVD}(\mathbf{J}\overline{\mathbf{X}}_u\mathbf{J}) = \mathbf{U}_u\mathbf{\Gamma}_u\mathbf{V}_u'$.
      iii. Update $\mathbf{JC}_u = \mathbf{U}_u\mathbf{\Gamma}_u^{\alpha}\mathbf{L}_P$ and $\mathbf{JD}_u = \mathbf{V}_u\mathbf{\Gamma}_u^{1-\alpha}\mathbf{L}_P$.
   (c) Update $\mathbf{G}_k$ to $\mathbf{G}_{k+1}$ by assigning all observations to the closest (approximated) cluster mean:

$$\arg\min_{u=1}^{U} \left\| \mathbf{J}\left(\mathbf{X}_i - \mathbf{C}_u\mathbf{D}_u'\right)\mathbf{J}\right\|^2, \quad i = 1, \ldots, I.$$

   (d) Compute the number of reassignments

$$\Delta_{k+1} = I - \mathbf{1}'\,\mathrm{diag}(\mathbf{G}_k'\mathbf{G}_{k+1}).$$

   (e) Update $k \leftarrow k + 1$.
3. Output $\mathbf{G}_k$ for $\mathbf{G}^{(\mathrm{i})}$, as well as $\mathbf{C}$ and $\mathbf{D}$.

are of interest; otherwise, the researcher may prefer to jointly cluster on more than one of these effects at the same time (see Sect. 6.1).

Next, we discuss computational aspects of the proposed LSBCLUST method.

5 Estimation algorithms

Due to the form of the loss function (13), we can treat each of the components separately. Conveniently, the loss functions $L_{(\mathrm{o})}\left(\mathbf{G}^{(\mathrm{o})}, \boldsymbol{m}\right)$, $L_{(\mathrm{r})}\left(\mathbf{G}^{(\mathrm{r})}, \mathbf{A}\right)$ and $L_{(\mathrm{c})}\left(\mathbf{G}^{(\mathrm{c})}, \mathbf{B}\right)$ are specific instances of the well-known k-means loss function (e.g., Everitt et al. 2011). They differ only with respect to the data matrix $\mathbf{Y} : I \times d$ (say) on which k-means cluster analysis is to be applied, which can respectively be defined as follows:

- For minimizing $L_{(\mathrm{o})}\left(\mathbf{G}^{(\mathrm{o})}, \boldsymbol{m}\right)$, $\mathbf{Y}$ has a single column ($d = 1$) containing the overall means $\frac{1}{JK}\mathbf{1}'\mathbf{X}_i\mathbf{1}$ of the $\mathbf{X}_i$ ($i = 1, \ldots, I$);
- For minimizing $L_{(\mathrm{r})}\left(\mathbf{G}^{(\mathrm{r})}, \mathbf{A}\right)$, the rows of $\mathbf{Y}$ ($d = J$) consist of the row mean vectors $\frac{1}{K}\mathbf{JX}_i\mathbf{1}$ ($i = 1, \ldots, I$); and
- For minimizing $L_{(\mathrm{c})}\left(\mathbf{G}^{(\mathrm{c})}, \mathbf{B}\right)$, the rows of $\mathbf{Y}$ ($d = K$) are the column mean vectors $\frac{1}{J}\mathbf{JX}_i'\mathbf{1}$ ($i = 1, \ldots, I$).

Hence optimizing $L_{(\mathrm{o})}\left(\mathbf{G}^{(\mathrm{o})}, \boldsymbol{m}\right)$, $L_{(\mathrm{r})}\left(\mathbf{G}^{(\mathrm{r})}, \mathbf{A}\right)$ and $L_{(\mathrm{c})}\left(\mathbf{G}^{(\mathrm{c})}, \mathbf{B}\right)$ can resort to standard methods for k-means on the overall mean, row margins and column margins respectively. Also, there are a variety of tools available for selecting R, S and T.

Minimizing $L_{(\mathrm{i})}\left(\mathbf{G}^{(\mathrm{i})}, \mathbf{C}, \mathbf{D}\right)$ however requires a custom algorithm. The main steps of our algorithm, which is based on block-relaxation methods (see, for example, de Leeuw 1994), is outlined in Algorithm 1. It iterates over optimizing $\mathbf{C}$ and $\mathbf{D}$ in Step 2b while keeping $\mathbf{G}^{(\mathrm{i})}$ fixed, and vice versa in Step 2c. This approach is also known as alternating least squares (ALS).

Convergence is reached when no reassignment of a single observation to a different cluster will reduce the value of the loss function. This corresponds to $\Delta_k = 0$ in Algorithm 1. The algorithm is guaranteed to converge monotonically, but only to some accumulation points which are usually local minima. It must be initialized by a starting configuration for $\mathbf{G}^{(\mathrm{i})}$. To increase the likelihood of locating the global minimum, it is advisable to use multiple (random) starting values for $\mathbf{G}^{(\mathrm{i})}$.

We now describe the derivation of the key steps of Algorithm 1. Step 2b, where $L_{(\mathrm{i})}\left(\mathbf{G}^{(\mathrm{i})}, \mathbf{C}, \mathbf{D}\right)$ is minimized over $\mathbf{C}$ and $\mathbf{D}$ for fixed $\mathbf{G}^{(\mathrm{i})}$, relies on the decomposition

$$\begin{aligned} &L_{(\mathrm{i})}\left(\mathbf{G}^{(\mathrm{i})}, \mathbf{C}, \mathbf{D}\right) \\ &\quad = \sum_{i=1}^{I}\sum_{u=1}^{U} g_{iu}^{(\mathrm{i})} \left\|\mathbf{J}\left(\mathbf{X}_i - \overline{\mathbf{X}}_u\right)\mathbf{J}\right\|^2 + \sum_{u=1}^{U} I_u \left\|\mathbf{J}\left(\overline{\mathbf{X}}_u - \mathbf{C}_u\mathbf{D}_u^{'}\right)\mathbf{J}\right\|^2. \end{aligned} \tag{14}$$

Here $I_u = \sum_{i=1}^{I} g_{iu}^{(\mathrm{i})}$ is the cardinality of cluster u, and $\overline{\mathbf{X}}_u = \frac{1}{I_u}\sum_{i=1}^{I} g_{iu}^{(\mathrm{i})}\mathbf{X}_i$ is the cluster mean.

Since only the final term in (14) depends on $\mathbf{C}$ and $\mathbf{D}$, it is sufficient to minimize this term only. Let the singular value decomposition (SVD) of $\mathbf{J}\overline{\mathbf{X}}_u\mathbf{J}$ be $\mathbf{U}_u\boldsymbol{\Gamma}_u\mathbf{V}_u^{'}$, where $\mathbf{U}_u$ and $\mathbf{V}_u$ are orthonormal and $\boldsymbol{\Gamma}_u$ diagonal. The Eckart–Young theorem (Eckart and Young 1936; Schmidt 1907) establishes the best rank-P least-squares approximation of $\mathbf{J}\overline{\mathbf{X}}_u\mathbf{J}$ as the truncated SVD $\mathbf{U}_u\boldsymbol{\Gamma}_u\mathbf{L}_P\mathbf{V}_u^{'}$, where

$$\mathbf{L}_P = \begin{bmatrix} \mathbf{I}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}. \tag{15}$$

Multiplication by $\mathbf{L}_P$ sets all singular values except the first P equal to zero. Consequently, we can update $\mathbf{C}$ and $\mathbf{D}$ using

$$\begin{aligned} \mathbf{J}\mathbf{C}_u &= \mathbf{U}_u\boldsymbol{\Gamma}_u^{\alpha}\mathbf{L}_P \\ \mathbf{J}\mathbf{D}_u &= \mathbf{V}_u\boldsymbol{\Gamma}_u^{1-\alpha}\mathbf{L}_P, \end{aligned} \tag{16}$$

where $\boldsymbol{\Gamma}^{\alpha}$ denotes the diagonal matrix containing the singular values to the power α. The parameter $0 \leq \alpha \leq 1$ is typically taken to be 0.5, but can be set by the user to improve the interpretability of the graphical output. See Sect. 6.3 for a description of the heuristic rule used in our implementation.

Now in Step 2c, $\mathbf{G}^{(\mathrm{i})}$ is updated while regarding $\mathbf{C}$ and $\mathbf{D}$ as fixed. The updated $\mathbf{G}^{(\mathrm{i})}$ is constructed by simply assigning each i to the cluster with the closest mean, hence minimizing $L_{(\mathrm{i})}\left(\mathbf{G}^{(\mathrm{i})}, \mathbf{C}, \mathbf{D}\right)$ for each individual in a greedy manner. This entails assigning observation i to cluster

$$\arg\min_{u=1}^{U} \left\| \mathbf{J}\left(\mathbf{X}_i - \mathbf{C}_u \mathbf{D}_u^{'}\right)\mathbf{J} \right\|^2. \tag{17}$$

A few details remain. When an empty cluster is encountered, that cluster is reinitialized using the worst-fitting observation across all clusters. Label-switching is countered by reassigning cluster labels between iterations when necessary. Finally, we prefer reporting the minimized value of $L_{(\mathrm{i})}\left(\mathbf{G}^{(\mathrm{i})}, \mathbf{C}, \mathbf{D}\right)$ after division by $\sum_{i=1}^{I} \left\|\mathbf{J}\mathbf{X}_i\mathbf{J}\right\|^2$, since this standardized value lies in [0, 1].

In the next section, we briefly discuss alternative model formulations.

6 Enhancements

Here we first consider accommodating alternative bilinear model specifications in the LSBCLUST loss function (7). Thereafter, we discuss restricted model formulations for the interaction effects that reduce the number of parameters and aid interpretation. Finally, we briefly consider scaling options used in biplot construction. Some practical guidelines on performing model selection are provided in Sect. 8, where an application is discussed.

6.1 Using other bilinear decompositions

In certain situations, a different bilinear model may be more appropriate than the one used in the loss function (7). For example, the researcher may not want to treat the grand, row and column means separately, in which case a more appropriate loss function would simply be

$$L(\mathbf{G}^{(\mathrm{i})}, \mathbf{C}, \mathbf{D}) = \sum_{i,u} g_{iu}^{(\mathrm{i})} \left\| \mathbf{X}_i - \mathbf{C}_u \mathbf{D}_u^{'} \right\|^2, \tag{18}$$

with the only constraint required being orthogonality of the columns of $\mathbf{C}_u$ and $\mathbf{D}_u$. Yet a situation may arise where the row means (but not the column means) are themselves interesting, leading instead to the following loss function:

$$L(\mathbf{G}^{(\mathrm{r})}, \mathbf{G}^{(\mathrm{i})}, \mathbf{A}, \mathbf{C}, \mathbf{D}) = \sum_{i,s,u} g_{ir}^{(\mathrm{o})} g_{is}^{(\mathrm{r})} \left\| \mathbf{X}_i - \left(\boldsymbol{a}_s \mathbf{1}^{'} + \mathbf{C}_u \mathbf{D}_u^{'} \mathbf{J}\right) \right\|^2. \tag{19}$$

Here the centering matrix enforces sum-to-zero constraints on the columns of $\mathbf{D}_u$, which allows $\boldsymbol{a}_s$ to be estimated.

In fact, a variety of different bilinear models can be specified by dropping one or more constraints, and hence cluster sets, from the formulation (7). An exhaustive list of the nine possibilities are given in Table 2, with Model 9 being the original formulation in (7). With the exception of Model 6, these models are orthogonal as in Sect. 4. Hence the algorithms in Sect. 5 also apply for these bilinear models (except Model 6), after minor adjustments to account for the different centering options.

Table 2 A summary of the models implied by different choices of $\boldsymbol{\delta}$ in the generalized LSBCLUST formulation

Model	δ_1	δ_2	δ_3	δ_4	Model for $\mathbf{X}_i$
1	0	0	0	0	$\mathbf{C}_u\mathbf{D}_u^{'}$
2	0	1	0	0	$\boldsymbol{a}_s\mathbf{1}^{'} + \mathbf{C}_u\mathbf{D}_u^{'}\mathbf{J}$
3	0	1	0	1	$m_r\mathbf{1}\mathbf{1}^{'} + \mathbf{J}\boldsymbol{a}_s\mathbf{1}^{'} + \mathbf{C}_u\mathbf{D}_u^{'}\mathbf{J}$
4	1	0	0	0	$\mathbf{1}\boldsymbol{b}_t^{'} + \mathbf{J}\mathbf{C}_u\mathbf{D}_u^{'}$
5	1	0	1	0	$m_r\mathbf{1}\mathbf{1}^{'} + \mathbf{1}\boldsymbol{b}_t^{'}\mathbf{J} + \mathbf{J}\mathbf{C}_u\mathbf{D}_u^{'}$
6	1	1	0	0	$-m_r\mathbf{1}\mathbf{1}^{'} + \boldsymbol{a}_s\mathbf{1}^{'} + \mathbf{1}\boldsymbol{b}_t^{'} + \mathbf{J}\mathbf{C}_u\mathbf{D}_u^{'}\mathbf{J}$
7	1	1	0	1	$\mathbf{J}\boldsymbol{a}_s\mathbf{1}^{'} + \mathbf{1}\boldsymbol{b}_t^{'} + \mathbf{J}\mathbf{C}_u\mathbf{D}_u^{'}\mathbf{J}$
8	1	1	1	0	$\boldsymbol{a}_s\mathbf{1}^{'} + \mathbf{1}\boldsymbol{b}_t^{'}\mathbf{J} + \mathbf{J}\mathbf{C}_u\mathbf{D}_u^{'}\mathbf{J}$
9	1	1	1	1	$m_r\mathbf{1}\mathbf{1}^{'} + \mathbf{J}\boldsymbol{a}_s\mathbf{1}^{'} + \mathbf{1}\boldsymbol{b}_t^{'}\mathbf{J} + \mathbf{J}\mathbf{C}_u\mathbf{D}_u^{'}\mathbf{J}$

Note that Model 6 is not orthogonal and is only included for completeness

In Table 2, and in our software, we characterize the different models using a vector of four binary indicators, $\boldsymbol{\delta}^{'} = (\delta_1, \delta_2, \delta_3, \delta_4)$. Each of these correspond to the presence or absence of one of the centering matrices in the model formulation. Specifically, $\delta_1 = 1$ and $\delta_2 = 1$ indicate centering the columns of $\mathbf{C}_u$ and $\mathbf{D}_u$ respectively (as in $\mathbf{J}\mathbf{C}_u$ and $\mathbf{J}\mathbf{D}_u$), while $\delta_3 = 1$ and $\delta_4 = 1$ in addition implies respectively centering the column and row means—as in $\mathbf{J}\boldsymbol{b}_t$ and $\mathbf{J}\boldsymbol{a}_s$. Additionally, it is only possible to have $\delta_3 = 1$ if $\delta_1 = 1$; an equivalent relationship holds between δ_4 and δ_2.

6.2 Common row or column coordinates

The formulation in (7) can contain a large number of parameters. This is mainly because the interaction approximations $\mathbf{C}_u\mathbf{D}_u^{'}$ require the row and column representations, $\mathbf{C}_u$ and $\mathbf{D}_u$ respectively, to be different for each cluster. We can counter this by restricting either the rows or columns to have a common representation across clusters, which has the added benefit of making the biplots based on $\mathbf{C}_u$ and $\mathbf{D}_u$ easier to interpret. We see this interpretability as the main reason to elect such a constraint; in a practical application this must be weighed against the resulting reduction in goodness-of-fit.

Consequently, we allow the following three options for modelling the interactions:

- **(I)** $\mathbf{C}_u\mathbf{D}_u^{'}$: both $\mathbf{C}_u$ and $\mathbf{D}_u$ are specific to the interaction cluster (as above); or
- **(II)** $\mathbf{C}_1\mathbf{D}_u^{'}$: a common row representation $\mathbf{C}_1$ for all interaction clusters, and a differential column representation $\mathbf{D}_u$ for each interaction cluster; or
- **(III)** $\mathbf{C}_u\mathbf{D}_1^{'}$: differential row representations $\mathbf{C}_u$ but a common column representation $\mathbf{D}_1$.

If the alternative specifications (II) or (III) are used, $\mathbf{C}_u\mathbf{D}_u^{'}$ in (7) should be replaced by $\mathbf{C}_1\mathbf{D}_u^{'}$ or $\mathbf{C}_u\mathbf{D}_1^{'}$ respectively. These restricted formulations also require adjustments to Algorithm 1. To facilitate this, we define the following block matrices by stacking either row- or column-wise:

$$\begin{aligned}
\mathbf{C}_* &= \left[\sqrt{I_1}\,\mathbf{C}_1^{'}\mathbf{J}\ \sqrt{I_2}\,\mathbf{C}_2^{'}\mathbf{J}\ \cdots\ \sqrt{I_U}\,\mathbf{C}_U^{'}\mathbf{J}\right]^{'};\\
\mathbf{D}_* &= \left[\sqrt{I_1}\,\mathbf{D}_1^{'}\mathbf{J}\ \sqrt{I_2}\,\mathbf{D}_2^{'}\mathbf{J}\ \cdots\ \sqrt{I_U}\,\mathbf{D}_U^{'}\mathbf{J}\right]^{'};\\
\overline{\mathbf{X}}_{(c)} &= \left[\sqrt{I_1}\,\overline{\mathbf{X}}_1\mathbf{J}\ \sqrt{I_2}\,\overline{\mathbf{X}}_2\mathbf{J}\ \cdots\ \sqrt{I_U}\,\overline{\mathbf{X}}_U\mathbf{J}\right];\\
\overline{\mathbf{X}}_{(r)} &= \left[\sqrt{I_1}\,\overline{\mathbf{X}}_1^{'}\mathbf{J}\ \sqrt{I_2}\,\overline{\mathbf{X}}_2^{'}\mathbf{J}\ \cdots\ \sqrt{I_U}\,\overline{\mathbf{X}}_U^{'}\mathbf{J}\right]^{'}.
\end{aligned} \tag{20}$$

The final term in (14) can then be rewritten in the respective formulations as

$$\begin{aligned}
&\left\|\mathbf{J}\left(\overline{\mathbf{X}}_{(c)} - \mathbf{C}_1\mathbf{D}_*^{'}\right)\right\|^2 \quad \text{in case (II), or}\\
&\left\|\left(\overline{\mathbf{X}}_{(r)} - \mathbf{C}_*\mathbf{D}_1^{'}\right)\mathbf{J}\right\|^2 \quad \text{in case (III).}
\end{aligned}$$

Step 2b in Algorithm 1 is subsequently amended to perform a single SVD on either $\mathbf{J}\overline{\mathbf{X}}_{(c)}$ to update $\mathbf{C}_1$ and $\mathbf{D}_*$ under (II), or on $\overline{\mathbf{X}}_{(r)}\mathbf{J}$ to update $\mathbf{C}_*$ and $\mathbf{D}_1$ under (III). From these, updates to $\mathbf{J}\mathbf{D}_u$ or $\mathbf{J}\mathbf{C}_u$ are derived for $u = 1, \ldots, U$ by extracting the relevant block matrices from $\mathbf{D}_*$ or $\mathbf{C}_*$ respectively.

6.3 Biplot interpretability

Under case (I), where there is no requirement for the interaction decompositions to be similar across clusters, it can aid interpretation to rotate the configurations so that the biplot axes lie more or less in the same direction. For any orthogonal matrix $\mathbf{Q}_u$, it holds for the inner product matrices that $\mathbf{C}_u\mathbf{D}_u^{'} = \left(\mathbf{C}_u\mathbf{Q}_u\right)\left(\mathbf{D}_u\mathbf{Q}_u\right)^{'}$, and hence these are invariant to orthogonal rotations. The problem of finding orthogonal matrices $\mathbf{Q}_u$, $u = 1, 2, \ldots, U$, such that either the row or column configurations match each other as closely as possible is known as the generalized orthogonal Procrustes problem (Gower 1975; Gower and Dijksterhuis 2004). We apply this by default in our software implementation.

Besides this adjustment, two types of scalings can be used to make the biplot displays more attractive, namely so-called α- and λ-scaling. First, since our choice of α in (16) does not change the inner product approximations, we are free to choose it such that the resulting biplots are easy to interpret. In our software implementation we use as a heuristic method the value of α which maximizes the minimum Euclidean distance over all row and column points to the origin. Alternatively, the user can choose any other quantile of these distances, such as the median, or specify the desired value of α explicitly.

Second, note that for matrices $\mathbf{A}$ and $\mathbf{B}$ it holds that $\mathbf{A}\mathbf{B}^{'} = (\lambda\mathbf{A})(\mathbf{B}^{'}/\lambda)$, so that λ can also be freely chosen. Following Gower et al. (2011, Section 2.3.1), we choose λ such

that the average squared Euclidean distances from the two sets of points represented by the rows of the matrices in (16) to the origin are equal. For case (I) in (16), for example, this amounts to choosing

$$\lambda = \left(\frac{J \|\mathbf{V}\boldsymbol{\Gamma}_u^{1-\alpha}\mathbf{L}\|^2}{K \|\mathbf{U}\boldsymbol{\Gamma}_u^{\alpha}\mathbf{L}\|^2} \right)^{1/4} = \left(\frac{J \operatorname{tr} \boldsymbol{\Gamma}_u^{1-\alpha}\mathbf{L}}{K \operatorname{tr} \boldsymbol{\Gamma}_u^{\alpha}\mathbf{L}} \right)^{1/4}.$$

The appendix in the supplemental materials provides information on goodness-of-fit indices to quantify the quality of the within-cluster approximations.

Next, we report the results of a stimulation study.

7 Simulation study

A simulation study was conducted to assess cluster membership recovery of LSBCLUST. Since the separate clustering problems for the overall mean, row means and column means are simply k-means problems, we focus on reporting the results for the row–column interactions. Simulation studies for ordinary k-means can be found, for example, in Milligan (1980).

The results of LSBCLUST are compared to that of T3CLUS, as well as to two different versions of k-means clustering. The first version of k-means, which we will call vectorized k-means or VECKMEANS, merely vectorizes the matrix slices by stringing them out as long vectors and then applies ordinary k-means to the matrix having these vectors as rows. The second variation first obtains the best P-dimensional approximation to each matrix slice using the SVD, and then applies VECKMEANS. This variant we call DIMKMEANS. Both these methods are to be compared to the LSBCLUST interaction clustering, hence the matrix slices $\mathbf{X}_i$ are double-centred before being submitted to these procedures.

Simulated data from the LSBCLUST model are constructed according to the following steps:

1. Simulate the cluster membership matrix $\mathbf{G}^{(\mathrm{i})}$ given the required number of clusters U and the number of observations I. The proportion of observations attributed to each class, π_u, $u = 1, \ldots, U$, must be specified. Similarly, generate $\mathbf{G}^{(\mathrm{o})}, \mathbf{G}^{(\mathrm{r})}, \mathbf{G}^{(\mathrm{c})}$ with R, S and T clusters respectively. The same cluster membership probabilities are used for these clusters.
2. Simulate overall means $m_r, r = 1, \ldots, R$, from a standard normal (Gaussian) distribution, as well as row means $\boldsymbol{a}_s, s = 1, \ldots, S$, and column means $\boldsymbol{b}_t, t = 1, \ldots, T$. These are also drawn from standard normal distributions, and subsequently centred.
3. Simulate matrices $\overline{\mathbf{X}}_u, u = 1, \ldots, U$, representing the cluster means in (14), as follows:

   (a) Generate two random orthogonal matrices using Stewart (1980)'s method, representing the rows and columns in P-dimensional space. Column-centre both these matrices to arrive at $\mathbf{U}_u$ and $\mathbf{V}_u$, say. If case (a) or (b) applies, either the same $\mathbf{U}_u = \mathbf{U}$ or $\mathbf{V}_u = \mathbf{V}$ is used for all clusters.

(b) Generate P singular values as random numbers on [0.5, 5], and sort these in decreasing order in the vector $\boldsymbol{\gamma}_u$. If case (a) or (b) applies, the same singular values $\boldsymbol{\gamma}_u = \boldsymbol{\gamma}$ are used for all clusters.
(c) Construct $\boldsymbol{\Gamma}_u = \text{diag}\left(\boldsymbol{\gamma}_u\right)$ and consequently $\overline{\mathbf{X}}_u = \mathbf{U}_u\boldsymbol{\Gamma}_u\mathbf{V}_u^{'}$.

4. Finally, construct the simulated $\underline{\mathbf{X}}$ by using the relevant simulated cluster means of each type for each matrix slice, and by adding random noise simulated from a normal (Gaussian) distribution with zero mean and standard error σ. Here σ controls the signal-to-noise ratio: larger σ makes it harder to estimate the parameters used in the simulation steps.

In our simulation study, the following factors were varied:

- The number of observations, $I \in \{100, 500\}$;
- The interaction decomposition: (a) $\mathbf{C}_1\mathbf{D}_u^{'}$ (rows fixed across clusters), (b) $\mathbf{C}_u\mathbf{D}_1^{'}$ (columns fixed) or (c) $\mathbf{C}_u\mathbf{D}_u^{'}$ (neither fixed);
- The error standard deviation, $\sigma \in \{0.5, 1, 1.5\}$;
- The cluster proportions, $\boldsymbol{\pi}_u$, being either balanced ($\boldsymbol{\pi}_u^{'} = (0.2, 0.2, 0.2, 0.2, 0.2)$) or unbalanced ($\boldsymbol{\pi}_u^{'} = (0.1, 0.15, 0.2, 0.25, 0.3)$); and
- The dimensionality of the interaction decomposition, $P \in \{2, 5\}$.

This translates to a $3^2 \times 2^3$ design, with 72 conditions. For simplification, the following were kept fixed: the model simulated from, namely model (9)—see Table 2; the size of the matrix slices, $J = K = 8$; and the number of clusters, $R = S = T = U = 5$.

For each of the 72 conditions, we generate 100 parameter sets. In turn, for each of these parameter sets, we generate 50 randomly sampled data sets, resulting in 5000 simulated data sets per condition, or 360,000 in total. The results can be assessed both on clustering quality and estimation accuracy, where the latter includes clustering quality as well as parameter recovery. Here we discuss clustering quality only.

7.1 Clustering quality

Cluster recovery is measured by the adjusted Rand index (ARI; Hubert and Arabie 1985), which in our case quantifies the similarity between the actual, simulated clustering and that recovered by an algorithm. It improves on simple cluster agreement by adjusting for the chance of a randomly chosen pair of subjects falling in the same class. The ARI takes a value of one when the cluster recovery is perfect, and zero when the algorithm performs similarly to random assignment. The ARI can also take negative values, which indicate worse performance than random assignment. We first report the performance profiles of the ARI, which assesses how well each method performs relative to all the other methods. Thereafter, we consider the absolute performance of the methods.

7.1.1 Performance profiles

Performance profiles are used to compare multiple algorithms on a chosen performance metric (Dolan and Moré 2002; van den Burg and Groenen 2016). The basic idea is

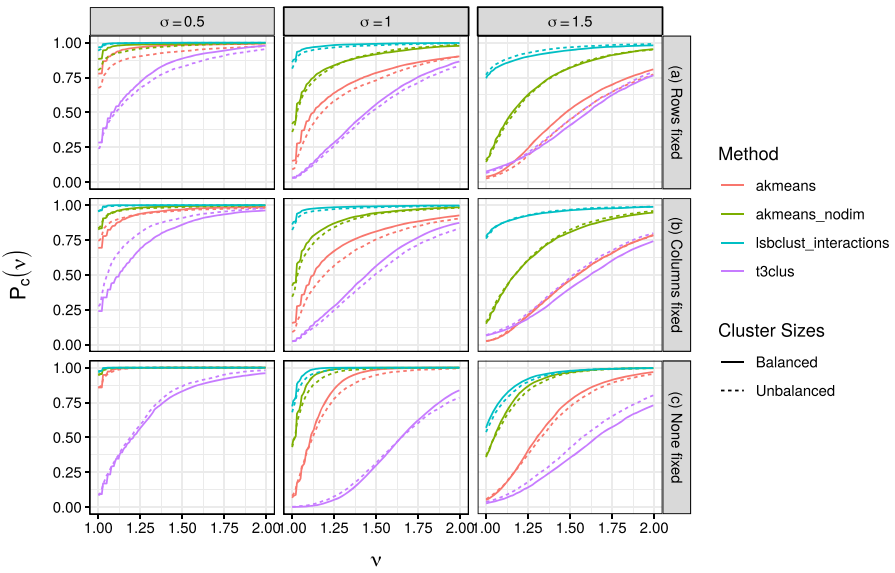

Fig. 1 Performance profiles based on the adjusted Rand index, for $P = 2$ dimensions and a sample size of $I = 100$

to express the performance of each algorithm relative to that of the best performing algorithm on each particular data set, and then to plot the cumulative distribution of this relative performance. Denote by $\mathcal{D}$ the set of data sets and by $\mathcal{C}$ the set of algorithms. Suppose that $p_{d,c}$ is the ARI for $d \in \mathcal{D}$ and $c \in \mathcal{C}$. The performance ratio $r_{d,c}$ is then defined as the ratio of the best performing ARI on data set d to $p_{d,c}$:

$$r_{d,c} = \frac{\max_{c \in C} p_{d,c}}{p_{d,c}}. \tag{21}$$

Typically, the best performing method has a performance ratio of one, with other methods having larger performance ratios, indicating how close these methods came to the best method. However, for the ARI, there is one caveat: the ARI may be negative or even exactly zero, in which case (21) do not work as intended. We circumvent this problem by adding a small positive constant to both the numerator and denominator in (21).

The performance profile for algorithm c is simply the empirical cumulative distribution function (ecdf) of the performance ratios. This can be calculated as

$$P_c(\nu) = \frac{|\{d \in \mathcal{D} : \nu_{d,c} \leq \nu\}|}{|\mathcal{D}|} \tag{22}$$

where $|\cdot|$ denotes the cardinality of a set. Therefore, $P_c(\nu)$ is simply the empirical probability of algorithm c having an performance ratio of at most ν, and by extension $P_c(1)$ is the empirical probability that algorithm c achieves the best performance.

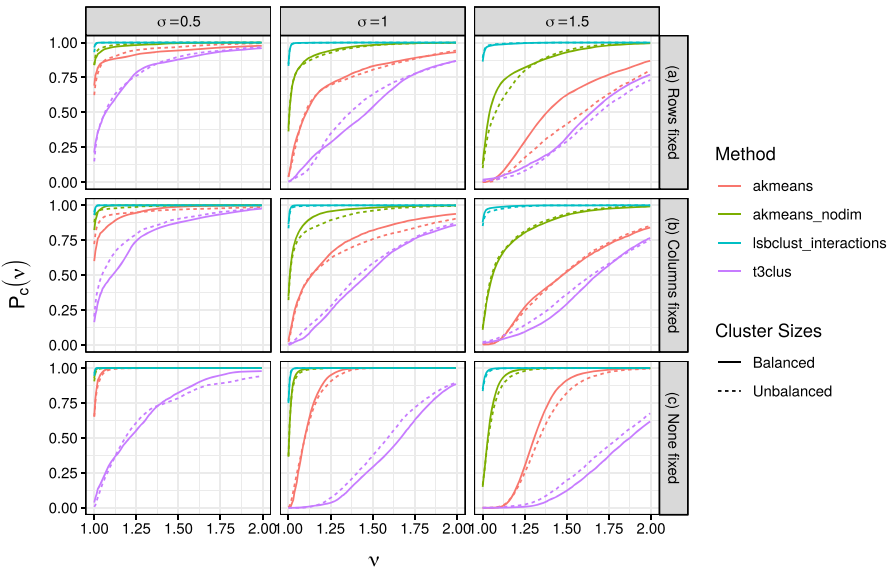

Fig. 2 Performance profiles based on the adjusted Rand index, for $P = 2$ dimensions and a sample size of $I = 500$

We calculate performance profiles for each combination of the five factors manipulated in the simulation study. The results are shown in Figs. 1 and 2. Both of these figures pertain to $P = 2$ dimensions, with Figs. 1 and 2 purporting to data sets with $I = 100$ and $I = 500$ observations respectively. For the sake of brevity, we omit the figures for $P = 5$ dimensions: in these cases, all algorithms achieve close to optimal results.

The results in Figs. 1 and 2 show that LSBCLUST generally outperforms the other methods, since its performance profiles in general are larger than that of the other methods. The next best method is VECKMEANS, with T3CLUS coming in last. It should be expected that LSBCLUST should perform well, since it was used to generate the data. VECKMEANS is quite competitive when interaction decomposition (c) is used (where neither component matrices are fixed across clusters). When the actual model do include restrictions on the interaction decomposition, LSBCLUST performs much better than the other methods.

In terms of factors manipulated in the study, the error standard deviation (σ), the sample size I and the interaction decomposition is most important. Whether the clusters are balanced or not has very little bearing on the results.

Having assessed the relative performance of the methods, we turn our attention to the absolute ARI achieved on the simulated data.

7.1.2 Adjusted Rand index

Table 3 reports the average ARI for the different methods. We first calculate the average ARI for the 50 data sets generated for each set of parameters, and then average

Table 3 The mean adjusted Rand indices for the different methods, averaged over data and parameter sets

		LSBCLUST			T3CLUS			VECKMEANS			DIMKMEANS		
		$\sigma = 0.5$	$\sigma = 1$	$\sigma = 1.5$	$\sigma = 0.5$	$\sigma = 1$	$\sigma = 1.5$	$\sigma = 0.5$	$\sigma = 1$	$\sigma = 1.5$	$\sigma = 0.5$	$\sigma = 1$	$\sigma = 1.5$
$I = 100$	(a) Rows fixed	0.977	0.788	0.494	0.847	0.543	0.287	0.960	0.678	0.334	0.972	0.748	0.426
	(b) Columns fixed	0.974	0.824	0.496	0.829	0.570	0.287	0.941	0.709	0.321	0.964	0.782	0.421
	(c) None fixed	0.997	0.871	0.520	0.800	0.518	0.303	0.992	0.778	0.406	0.996	0.853	0.499
$I = 500$	(a) Rows fixed	0.959	0.841	0.624	0.849	0.590	0.360	0.931	0.730	0.454	0.954	0.821	0.591
	(b) Columns fixed	0.979	0.838	0.594	0.845	0.575	0.337	0.950	0.717	0.416	0.976	0.818	0.553
	(c) None fixed	0.997	0.916	0.694	0.803	0.548	0.355	0.992	0.828	0.524	0.997	0.908	0.668

The results here considers only $P = 2$ dimensions and balanced clusters: for $P = 5$ clusters, all methods achieved nearly optimal ARI on most data sets, there is little difference in results for unbalanced cluster sizes

the resultant 100 average ARI's over the 100 different parameter sets. The table only includes results for $P = 2$ dimensions, and for the case where the cluster sizes are balanced. The latter does not affect the results significantly, so has been omitted. The former does have an important but uninteresting effect: when $P = 5$, all methods achieved near optimal ARI. Overall, an increase in the error standard deviation degrades the model performance the most. With enough samples, LSBCLUST can however still achieve decent clustering performance when σ is high.

In the next section, we consider an illustrative empirical example.

8 Application

The data comes from a brand positioning study where 187 consumers evaluated 10 car manufacturers on a set of 8 attributes (Bijmolt and van de Velden 2012). It was collected via an online survey using a representative Dutch sample from the CentERpanel of Tilburg University in the Netherlands, with the aim of studying how consumers perceive different car brands. The data $\underline{\mathbf{X}}$ is therefore of size $187 \times 10 \times 8$.

The 10 international car brands considered were: Citroën, Fiat, Ford, Opel, Peugeot, Renault, Seat, Toyota, Volkswagen and Volvo. Respondents rated each of these brands on 8 different attributes using a 10-point rating scale. For 6 out of the 8 items, namely Affordability, Attractiveness, Safety, Sportiness, Reliability and Features, a score of 10 is the most desirable outcome. However, for the items Size and Operating Cost, a score of 10 reflects small cars and those with high operating costs, respectively. Hence, higher ratings on these two attributes reflect increasingly negative sentiment, in contrast to the remaining six items.

We fit an LSBCLUST model with $\boldsymbol{\delta} = (1, 1, 1, 1)$—Model 9 in Table 2—so that the overall means, row means, column means and interactions are estimated separately. Also, we select $P = 2$ dimensions for display purposes, and we fix the coordinates of the 10 car brands across all interaction biplots—using case (II) from Sect. 6.2—to aid with interpretation.

Besides these choices, the number of clusters for each of the four components must be determined. For simplicity, we opt to do this separately for each of the four sets of clusters, although it is possible to make a joint selection. Selecting the number of clusters is a common problem, and many procedures and criteria have been proposed in the literature. Milligan and Cooper (1985), Hardy (1996) and Everitt et al. (2011) provide an assessment of some of these criteria and additional references. The simplest approach, and the one we use here for illustration, is probably the scree test (Cattell 1966). This method involves running the algorithm for several values of k and plotting the loss function against k. The user must then choose a value for k based on this so-called scree plot, such that the chosen k is close to an "elbow" in the plot. This indicates that adding additional groups to the analysis does not significantly increase how well the results describe the data.

Table 4 Segments based on overall mean ratings as detected in the cars data

Cluster	Size	Mean
O1	105	5.94
O2	45	7.04
O3	26	4.55
O4	8	9.80
O5	3	1.00

Here we fit LSBCLUST models for 1 to 15 clusters and inspect the resulting scree plots to select R, S, T and U.[1] Based on these plots, we selected $R = 5$, $S = 5$, $T = 6$ and $U = 8$ clusters. The number of row clusters was reduced to $S = 5$ from an initial choice of $S = 8$ to avoid clusters contain a single observation only. We note that these choices are subjective, should take into account the aims of the research and that alternative selection criteria can also be used. The number of random starts used for the interaction and k-means clustering were 100 and 1000 respectively.

The cluster sizes and centres (i.e., m_r in Eq. (7)) for each of the five overall mean clusters are shown in Table 4. Noticeable here are that the small clusters O4 (8 observations) and O5 (3 observations) identified persons who invariably used very high and very low scores, respectively. These respondents obviously do not provide very interesting information in their answers. But since their corresponding row means, column means and interactions do not differ from the overall mean, they are merely assigned to the row, column and interaction segments containing negligible effects (see below). LSBCLUST has therefore been able to identify the 11 persons in clusters O4 and O5 who provide very little sensible information.

Figure 3 displays the means of the eight car brand (row) clusters across all attributes. Effect sizes can be read off on the horizontal axis. The first two clusters, Segments R1 (98 observations) and R2 (59 observations), contain no pronounced large effects, which indicates that these consumers do not have strong, consistent opinions on any of the brands across all attributes. The 23 observations in Segment R3 do assign lower scores to Volvo and somewhat higher scores to Peugeot across all items. These effects amount to (approximately) -1.4 and 0.7 for Volvo and Peugeot, respectively. Even larger effects are observed in the smaller segments R4 (4 observations) and R5 (3 observations), but these comprise a relatively small part of the data.

The attribute (column) mean effects for all six clusters are displayed in Fig. 4. Here all segments contain at least 13 observations. Again, the largest cluster, Segment C1 (69 observations), contains no large effects. Segments C2 (35 observations), C3 (33 observations), and C4 (22 observations) are similar in that they display an inclination to assign higher scores on Reliability and Safety, and lower scores on Affordability, irrespective of the car brand being assessed. The magnitude of these effects vary greatly over these clusters though, and Segments C2 and C4 also display negative effects for Size. Segment C5 (15 observations) is dominated by a tendency to assign low scores

[1] Runtime is roughly one minute on a laptop with an Intel Core i7-6560U processor with 16 GB RAM running Microsoft Windows 10.

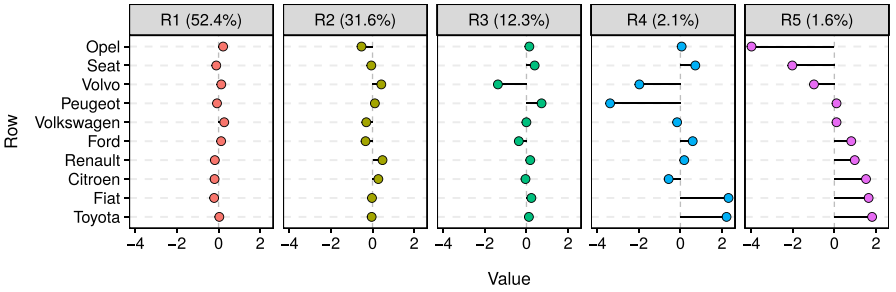

Fig. 3 Car manufacturer (row) cluster means detected in the cars data. The size of the effects can be read off from the horizontal axis

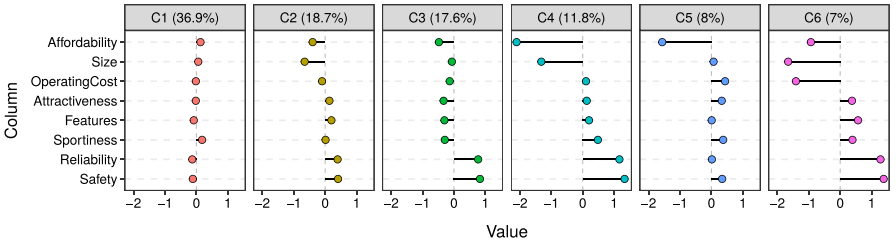

Fig. 4 Attribute (column) cluster means detected in the cars data

on Affordability. Several large effects are seen in Segment C6 (13 observations). These indicate a generally positive assessment of all brands on Size, Operating Cost, Reliability and Size, bearing in mind that for the former two attributes lower scores are better.

The most interesting results can be found among the interactions, which is where respondents distinguish between different car manufacturers on the measured attributes. Figs. 5 and 6 shows the biplots for the eight interaction segments. The car manufacturers are represented by points, and the attributes by arrows. The labels, points and arrows are shaded according to their goodness-of-fit, with well-fitting points being darker. All car brands, except Ford with a fit of only 0.09, fit reasonably well—see Table 5. The locations of the car brands are fixed across all biplots to make them easier to interpret. It is immediately apparent that the French manufacturers (Peugeot, Citroën and Renault) are judged to be similar, while the German brands Opel and Volkswagen are also located close together. Volvo, the Swedish car manufacturer, is somewhat isolated towards the right of the biplots, in contrast with Fiat at the opposite side of the plot. Fiat and Toyota are judged to be somewhat similar to the French and German brands respectively. Seat in turn are most similar to Toyota. As a result of its low fit, Ford is hardly visible and lies near the origin.

The fit for the eight attributes vary per segment, and is summarized in Table 6. Typically only a subset of items fit well in each segment, and only the better-fitting ones are adorned with calibrated axes in Figs. 5 and 6. For any manufacturer, the estimated within-cluster interaction effects can be read off from the orthogonal projection of its representing point onto the respective biplot axes. For example, Volvo

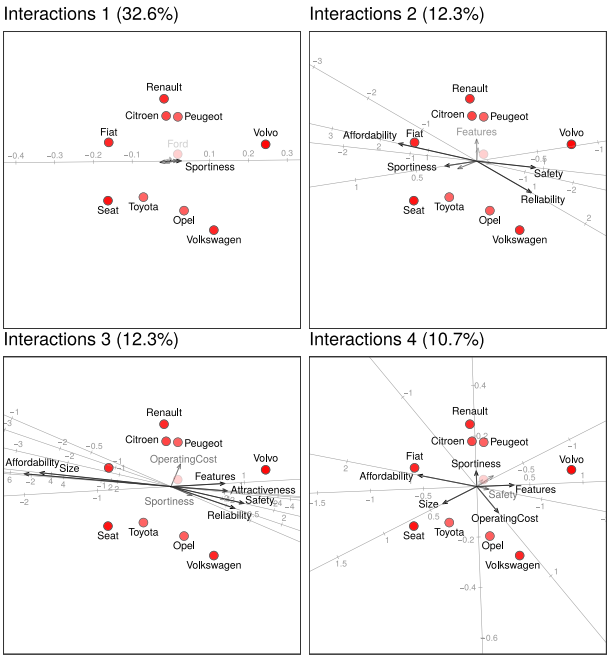

Fig. 5 Biplots for the interaction clusters I1 to I4 detected in the cars data. The relative cluster sizes are displayed in the titles of the different panels. Each attribute is represented by a vector, and those which fit best in each panel also by calibrated axes. The car manufacturers are represented by points, which has identical locations across panels. The orthogonal projections of the car manufacturer points onto the attribute axes give the estimated mean effects. The colors and labels are faded according to how well they fit into the display: solid colors fit well and transparent ones fit badly

scores approximately 0.25 points above that predicted by the overall mean, row mean and column mean on Sportiness in Segment I1, and Fiat score about 2 rating points above the overall and marginal effects on Affordability in the Segment I6. The overall variance accounted for in approximating the cluster means is 82.8% in two dimensions, with 64.1% and 18.7% attributed to dimensions 1 and 2, respectively. Hence two dimensions are a reasonable choice.

The effects in the interaction clusters should be interpreted as deviations from that expected from the overall and marginal means alone. Here are summaries of the interaction segments:

- Segment I1 (32.6%) has no large effects. Hence for this sizable group of individuals, the overall and marginal means contain most of the information provided.
- Segment I2 (12.3%) interprets Fiat and Seat to be more affordable than expected from the overall and marginal effects alone. The opposite applies to Volvo and Volkswagen. In terms of safety and reliability, however, the roles are reversed: Fiat and Seat score lower than expected, while Volvo and Volkswagen excel on these items. The effect sizes are roughly the same on the aforementioned three items.

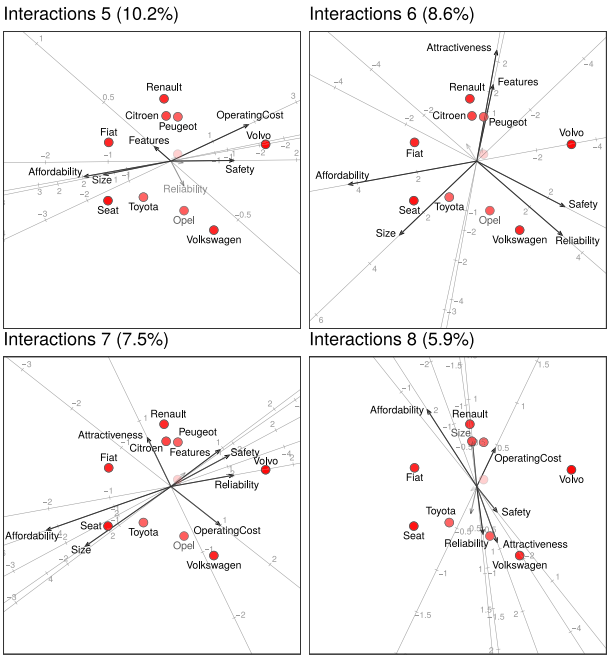

Fig. 6 Biplots for the interaction clusters I5 to I8 detected in the cars data

- Segment I3 (12.3%) contrasts positive interaction effects on Affordability and Size with negative effects on Safety, Reliability, Attractiveness and Features, and vice versa. Taking into account that Size is reverse-coded, this segment considers Fiat and Seat to be more affordable but smaller cars. At the same time, they are considered less reliable, attractive and safe than especially Volvo and Volkswagen. This segment is not dissimilar to Segment I2, except that effect sizes are larger and the inclusion of Attractiveness and Features on the right side of the plot.
- Segment I4 (10.7%) contains smaller effects than Segment I3. It distinguishes somewhat between Affordability and Size, in contrast to Segment I3. Whereas Fiat still better than expected in terms of Affordability, Seat is perceived to perform worse than Fiat on Size (after adjusting for the overall and marginal means). There is also now a much bigger difference between Volvo and Volkswagen in terms of size, with the latter scoring worse than expected compared to Renault, Peugeot and Citroën.
- Segment I5 (10.2%) again interprets the left of the plot as better in terms of Affordability and worse in terms of Size. But it also associates this with lower operating costs (since the latter is reverse-coded). Volvo and Volkswagen are scores higher than expected in terms of safety, in contrast to Seat and Fiat.
- Segment I6 (8.6%) contains large interaction effects on three pairs of items. Renault, Citroën and Peugeot score higher on Attractiveness and Features than expected. This is in contrast to Seat and Volkswagen, who exhibit negative interaction effects with these items. Safety and Reliability again favour Volkswagen

Table 5 Brand fit for the cars data across all clusters. Higher values indicate better fit, with a maximum of one and minimum of zero

Brand	Fit
Volvo	0.96
Seat	0.92
Volkswagen	0.84
Fiat	0.83
Renault	0.82
Citroen	0.70
Opel	0.60
Peugeot	0.58
Toyota	0.56
Ford	0.09

and Volvo, while Affordability and Size have similar but larger effects than in Segment I4.

- Segment I7 (7.5%) have similar interaction effects for Affordability and Size as with Segments I4 and I5. But these individuals also consider Volkswagen, Volvo and Opel to have higher than expected operating costs, in contrast to Fiat, Renault and Citroën. The latter brands, together with Peugeot are also considered the most attractive, with Volkswagen being considered less attractive than expected using only the overall and main effects.
- Segment I8 (5.9%) contains individuals who consider Renault, Fiat, Citroën and Peugeot to be more affordable than expected using only the overall and main effects. At the same time, these brands are considered to have higher expected operating costs and smaller cars. The direction of Attractiveness, Safety and Reliability indicate that the combination of these aspects are considered to correlate with higher prices.

Clearly, there are strong similarities but also interesting differences in how these groups of individuals interpreted the performance of the brands on the respective items after adjusting for overall and main effects. In the context of this application, these insights can provide valuable input to a brand positioning strategy, for example. Code reproducing our analysis of these data appears in the appendix of the supplemental materials.

9 Conclusions

This paper introduces LSBCLUST, a modelling framework for three-way data, where one of the three ways is clustered over whilst the corresponding matrix slices are approximated by low-rank decompositions. The clustering is done simultaneously with respect to up to four different aspects of these matrix slices, namely the overall

Table 6 Attribute fit for the cars data, for all eight interaction segments

	Interaction segment							
	I1	I2	I3	I4	I5	I6	I7	I8
Affordability	0.58	0.91	0.98	0.77	0.85	0.84	0.96	0.88
Attractiveness	0.21	0.19	0.80	0.33	0.24	0.90	0.70	0.66
Safety	0.57	0.89	0.95	0.50	0.87	0.92	0.76	0.63
OperatingCost	0.49	0.46	0.49	0.62	0.90	0.09	0.67	0.53
Sportiness	0.71	0.60	0.52	0.56	0.05	0.13	0.20	0.36
Size	0.23	0.26	0.93	0.83	0.90	0.94	0.87	0.66
Reliability	0.39	0.95	0.90	0.12	0.37	0.92	0.89	0.70
Features	0.18	0.35	0.73	0.94	0.72	0.83	0.91	0.06

mean responses, the row means, the column means, and the row–column interactions. These are the four elements of the biadditive (or bilinear) model used to approximate each of the matrix slices. Which of these terms are included in the model depends on the choice of identifiability constraints, as parametrized by $\boldsymbol{\delta}$. We show that in eight out of nine unique choices for $\boldsymbol{\delta}$, the combination of the bilinear model and least-squares loss allows the four clustering problems to be treated independently. This important property greatly simplifies the complexity of the clustering problem, which also has positive implications for model selection and the interpretation of the results. The low-rank decompositions of the interaction cluster means lead to readily interpretable biplots which aid in the interpretation of the results.

As illustrated in an application, LSBCLUST is a useful and natural alternative to more traditional three-way matrix decomposition methods such as PARAFAC/CANDECOMP and TUCKALS3. Our method uses a combination of well-known multivariate statistical methods, namely k-means cluster analysis, low-rank decompositions of two-way matrices as well as biplots, whereas traditional three-way methods require domain-specific expertise. Since least-squares loss functions are used, the problems can be treated very efficiently in software. Such software implementing LSBCLUST has been developed in the form of an eponymous R (R Core Team 2020) package. The package, lsbclust (Schoonees 2019), is available for download from the Comprehensive R Archive Network (CRAN, http://cran.r-project.org).

There are some points that require further research. The treatment of missing values have not been discussed, and should be investigated in the future. In terms of model selection, a wide variety of alternatives to the scree test can and should be investigated. There are a number of promising methods available in the literature, including using multiple criteria and taking a vote to determine the most attractive choice. We note that the rank of the low-rank decomposition can also be considered as a model selection step. Furthermore, it would be possible to add case weights to the methodology. An

advantage of case weights is that it allows a mechanism for implementing the bootstrap (e.g. Efron and Tibshirani 1994) to assess the variability of any given solution.

Author Contributions All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by Pieter Schoonees, Patrick Groenen and Michel van de Velden. The first draft of the manuscript was written by Pieter Schoonees and Patrick Groenen, and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding None.

Availability of data and material All data required to reproduce the results in the manuscript are available in the **lsbclust** (Schoonees 2019) package for R (R Core Team 2020), which is freely available on the Comprehensive R Archive Network (CRAN).

Declarations

Conflicts of interest None.

Code availability The reported analyses were conducted using the **lsbclust** package (Schoonees 2019). See Sect. D in the Supplemental Material for more information.

Appendix:

The appendix contains additional materials cited in the main paper. The first section discusses orthogonality, as referenced in Sect. 4. The second section discusses goodness-of-fit indices for biplots, as mentioned in Sect. 6. Additional information on the LC- BML model is given in the third section. The final section shows how to reproduce the analysis of Sect. 8 using the dedicated R package. (PDF document)

R package `lsbclust`:

The R package `lsbclust` contains the code to perform the methods described in the article. The package also contains the data set used in the article. The most up-to-date version can be found on CRAN at http://cran.r-project.org. (GNU zipped tar file)

A Orthogonality

Here we discuss the orthogonality of the decomposition (13), using the notation of Sect. 6.1. Equation (13) represents a special case of the more general notation. For the decomposition (13) to be orthogonal, it must be shown that all six cross-products occurring among the terms in a generalized version of (10) are zero. We treat each of these cross-products in turn.

1. For the cross-product between the interaction term and the row term, it holds that

$$\begin{aligned}
&\operatorname{tr}\left(\mathbf{J}_J^{(\delta_1)}\left(\mathbf{X}_i-\mathbf{C}_u\mathbf{D}_u^{'}\right)\mathbf{J}_K^{(\delta_2)}\right)\left(\delta_2\mathbf{J}_J^{(\delta_4)}\left(\tfrac{1}{K}\mathbf{X}_i\mathbf{1}_K-\boldsymbol{a}_s\right)\mathbf{1}_K^{'}\right)^{'}\\
&\quad=\delta_2\operatorname{tr}\mathbf{J}_J^{(\delta_1)}\left(\mathbf{X}_i-\mathbf{C}_u\mathbf{D}_u^{'}\right)\mathbf{J}_K^{(\delta_2)}\mathbf{1}_K\left(\tfrac{1}{K}\mathbf{1}_K^{'}\mathbf{X}_i^{'}-\boldsymbol{a}_s^{'}\right)\mathbf{J}_J^{(\delta_4)}\\
&\quad=0.
\end{aligned}\tag{23}$$

   The last equality follows since when $\delta_2=1$, $\mathbf{J}_K^{(\delta_2)}\mathbf{1}_K=\mathbf{J}_K\mathbf{1}_K=\mathbf{0}$. When $\delta_2=0$, the equality is trivial.
2. For the cross-product between the interaction term and the column term, the result is analogous to the above, except that now the equality $\mathbf{1}_J^{'}\mathbf{J}_J=\mathbf{0}^{'}$ is used.
3. For the cross-product between the interaction term and the term for the overall mean, we have that

$$\begin{aligned}
&\operatorname{tr}\left(\mathbf{J}_J^{(\delta_1)}\left(\mathbf{X}_i-\mathbf{C}_u\mathbf{D}_u^{'}\right)\mathbf{J}_K^{(\delta_2)}\right)\left(\delta^*\left(\tfrac{1}{JK}\mathbf{1}_J^{'}\mathbf{X}_i\mathbf{1}_K-m_r\right)\mathbf{1}_J\mathbf{1}_K^{'}\right)^{'}\\
&\quad=\delta^*\left(\mathbf{1}_J^{'}\mathbf{J}_J^{(\delta_1)}\left(\mathbf{X}_i-\mathbf{C}_u\mathbf{D}_u^{'}\right)\mathbf{J}_K^{(\delta_2)}\mathbf{1}_K\right)\left(\tfrac{1}{JK}\mathbf{1}_J^{'}\mathbf{X}_i\mathbf{1}_K-m_r\right)\\
&\quad=0.
\end{aligned}\tag{24}$$

   This cross-product equals zero whenever at least one of the following is true: $\delta_1=1$, $\delta_2=1$ or $\delta^*=0$. But whenever both $\delta_1=\delta_2=0$, $\delta^*=\delta_1\delta_3+\delta_2\delta_4-\delta_1\delta_2=0$ irrespective of δ_3 and δ_4. Hence the cross-product always equals zero.
4. For the cross-product between the row and column terms, we have

$$\begin{aligned}
&\operatorname{tr}\left(\delta_1\mathbf{1}_J\left(\tfrac{1}{J}\mathbf{1}_J^{'}\mathbf{X}_i-\boldsymbol{b}_t^{'}\right)\mathbf{J}_K^{(\delta_3)}\right)\left(\delta_2\mathbf{J}_J^{(\delta_4)}\left(\tfrac{1}{K}\mathbf{X}_i\mathbf{1}_K-\boldsymbol{a}_s\right)\mathbf{1}_K^{'}\right)^{'}\\
&\quad=\delta_1\delta_2\left(\tfrac{1}{J}\mathbf{1}_J^{'}\mathbf{X}_i-\boldsymbol{b}_t^{'}\right)\mathbf{J}_K^{(\delta_3)}\mathbf{1}_K\left(\tfrac{1}{K}\mathbf{1}_K^{'}\mathbf{X}_i^{'}-\boldsymbol{a}_s^{'}\right)\mathbf{J}_J^{(\delta_4)}\mathbf{1}_J\\
&\quad=\begin{cases}\left(\tfrac{1}{J}\mathbf{1}_J^{'}\mathbf{X}_i\mathbf{1}_K-\boldsymbol{b}_t^{'}\mathbf{1}_K\right)\left(\tfrac{1}{K}\mathbf{1}_K^{'}\mathbf{X}_i^{'}\mathbf{1}_J-\boldsymbol{a}_s^{'}\mathbf{1}_J\right) & \text{if } \boldsymbol{\delta}=(1,1,0,0)^{'}\\ 0 & \text{otherwise.}\end{cases}
\end{aligned}\tag{25}$$

   Deducing when the cross-product equals zero uses the same concepts as above, but when $\boldsymbol{\delta}=(1,1,0,0)^{'}$ none of these apply and the cross-product is not necessarily equal to zero.

5. The cross-product between the column term and the term for the overall mean also does not necessarily equal zero. Here we can derive the following:

$$\begin{aligned}
&\operatorname{tr}\left(\delta_1 \mathbf{1}_J \left(\tfrac{1}{J}\mathbf{1}_J'\mathbf{X}_i - \boldsymbol{b}_t'\right)\mathbf{J}_K^{(\delta_3)}\right)\left(\delta^*\left(\tfrac{1}{JK}\mathbf{1}_J'\mathbf{X}_i\mathbf{1}_K - m_r\right)\mathbf{1}_J\mathbf{1}_K'\right)' \\
&\quad = J\delta_1\delta^*\left(\tfrac{1}{J}\mathbf{1}_J'\mathbf{X}_i - \boldsymbol{b}_t'\right)\mathbf{J}_K^{(\delta_3)}\mathbf{1}_K\left(\tfrac{1}{JK}\mathbf{1}_J'\mathbf{X}_i\mathbf{1}_K - m_r\right) \\
&\quad = \begin{cases} -J\left(\tfrac{1}{J}\mathbf{1}_J'\mathbf{X}_i\mathbf{1}_K - \boldsymbol{b}_t'\mathbf{1}_K\right)\left(\tfrac{1}{JK}\mathbf{1}_J'\mathbf{X}_i\mathbf{1}_K - m_r\right) & \text{if } \boldsymbol{\delta} = (1,1,0,0)' \\ 0 & \text{otherwise.} \end{cases}
\end{aligned} \tag{26}$$

The last line follows from the fact that the crossproduct equals zero if $\delta_1 = 0$, if $\delta_1 = 1$ and $\delta_3 = 1$, and when $\delta^* = 0$. Hence consideration must be given to the four cases $\delta_1 = 1, \delta_2 \in \{0,1\}, \delta_3 = 0, \delta_4 \in \{0,1\}$. It is easy to see that $\delta^* = \delta_1\delta_3 + \delta_2\delta_4 - \delta_1\delta_2 = \delta_2\delta_4 - \delta_2 = 0$ in all these cases except when $\boldsymbol{\delta} = (1,1,0,0)'$.

6. Analogously to the above, consider the cross-product between the row term and the term for the overall mean:

$$\begin{aligned}
&\operatorname{tr}\left(\delta_2\mathbf{J}_J^{(\delta_4)}\left(\tfrac{1}{K}\mathbf{X}_i\mathbf{1}_K - \boldsymbol{a}_s\right)\mathbf{1}_K'\right)\left(\delta^*\left(\tfrac{1}{JK}\mathbf{1}_J'\mathbf{X}_i\mathbf{1}_K - m_r\right)\mathbf{1}_J\mathbf{1}_K'\right)' \\
&\quad = K\delta_2\delta^*\mathbf{1}_J'\mathbf{J}_J^{(\delta_4)}\left(\tfrac{1}{K}\mathbf{X}_i\mathbf{1}_K - \boldsymbol{a}_s\right)\left(\tfrac{1}{JK}\mathbf{1}_J'\mathbf{X}_i\mathbf{1}_K - m_r\right) \\
&\quad = \begin{cases} -K\left(\tfrac{1}{K}\mathbf{1}_J'\mathbf{X}_i\mathbf{1}_K - \mathbf{1}_J'\boldsymbol{a}_s\right)\left(\tfrac{1}{JK}\mathbf{1}_J'\mathbf{X}_i\mathbf{1}_K - m_r\right) & \text{if } \boldsymbol{\delta} = (1,1,0,0)' \\ 0 & \text{otherwise.} \end{cases}
\end{aligned} \tag{27}$$

The expression equals zero when $\delta_2 = 0$, when $\delta_2 = 1$ and $\delta_4 = 1$ or when $\delta^* = 0$. Considering the cases $\delta_1 \in \{0,1\}, \delta_2 = 1, \delta_3 \in \{0,1\}, \delta_4 = 1$ then, it can be seen that $\delta^* = \delta_1\delta_3 + \delta_2\delta_4 - \delta_1\delta_2 = \delta_1\delta_3 - \delta_1 = 0$ except when $\boldsymbol{\delta} = (1,1,0,0)'$.

Consequently the decomposition in (13) is valid for all $\boldsymbol{\delta}$, except for $\boldsymbol{\delta} = (1,1,0,0)'$—Model 6 in Table 2.

B Fit diagnostics

The quality of the solution can be assessed using a variety of metrics. We focus on measures for investigating the quality of the interaction approximations.

The interaction approximations—case (I), (II), or (III)—allow biplots to be used for visualizing the relationships between the J rating categories and K items for each of the clusters. Biplots generalize scatterplots of two variables to multiple variables (Gower and Hand 1996; Gower et al. 2011), and rely on low-rank inner product approximations. These are most useful when the number of dimensions is low: for example, $P \in \{1,2,3\}$. Constructing biplots for the interactions simply entails plotting the approximation of $\overline{\mathbf{X}}_u$ for each cluster. For example, under (III) the rating categories

are represented in P dimensions by the rows in $\mathbf{C}_u$, while the items are represented in the same space by the rows of $\mathbf{D}_u$. The inner products between the pairs of rows in these matrices are rank-P approximations of the corresponding entries in $\overline{\mathbf{X}}_u$. We defer discussion of the interpretation of these biplots to Sect. 8, where empirical examples are examined.

Goodness-of-fit measures for the biplots—and hence, the interactions—are based on the proportion of variation accounted for by the model. A fit value of one indicates perfect fit, while low fit values imply that a substantial amount of variation occurs in the subspace orthogonal to that identified by the model. These are calculated separately for each interaction cluster, as appropriate. Again, we must distinguish between the three cases (I), (II) and (III). We discuss first case (I) here. Cases (II) and (III) are briefly treated after. We do not explicitly include any eventual centering matrices to keep the notation simple.

Measures can be defined for (a) the overall fit, (b) the J rating categories in the rows of $\mathbf{X}_i$, as well as for (c) the K items in the columns of $\mathbf{X}_i$. For case (I), these are:

(a) The overall quality of fit for the interactions within cluster u for P dimensions is

$$o_{\text{fit}}(u) = \frac{\|\mathbf{C}_u\mathbf{D}_u'\|^2}{\|\overline{\mathbf{X}}_u\|^2} = \frac{\operatorname{tr}\boldsymbol{\Gamma}_u^2\mathbf{L}}{\operatorname{tr}\boldsymbol{\Gamma}_u^2}. \tag{28}$$

This is just the proportion of the variation in the cluster mean explained by the model. Here $\operatorname{tr}\mathbf{A}$ denotes the trace of the matrix $\mathbf{A}$, which is just the sum of its diagonal elements.

(b) The proportion of the variation explained by each of the rows (rating categories), also known as sample predictivities (Gower et al. 2011), is calculated as

$$\begin{aligned}\boldsymbol{r}_{\text{fit}}(u) &= \left[\operatorname{diag}\mathbf{C}_u\mathbf{D}_u'\mathbf{D}_u\mathbf{C}_u'\right]\left[\operatorname{diag}\overline{\mathbf{X}}_u\overline{\mathbf{X}}_u'\right]^{-1}\mathbf{1}_J \\ &= \left[\operatorname{diag}\mathbf{U}_u\boldsymbol{\Gamma}_u^2\mathbf{L}\mathbf{U}_u'\right]\left[\operatorname{diag}\mathbf{U}_u\boldsymbol{\Gamma}_u^2\mathbf{U}_u'\right]^{-1}\mathbf{1}_J,\end{aligned} \tag{29}$$

with each element bounded on [0, 1]. In this context, $\operatorname{diag}\mathbf{A}$ denotes the diagonal matrix constructed from the main diagonal of $\mathbf{A}$.

(c) The column fit can be defined analogously for each of the K items as

$$\begin{aligned}\boldsymbol{c}_{\text{fit}}(u) &= \left[\operatorname{diag}\mathbf{D}_u\mathbf{C}_u'\mathbf{C}_u\mathbf{D}_u'\right]\left[\operatorname{diag}\overline{\mathbf{X}}_u'\overline{\mathbf{X}}_u\right]^{-1}\mathbf{1}_K \\ &= \left[\operatorname{diag}\mathbf{V}_u\boldsymbol{\Gamma}_u^2\mathbf{L}\mathbf{V}_u'\right]\left[\operatorname{diag}\mathbf{V}_u\boldsymbol{\Gamma}_u^2\mathbf{V}_u'\right]^{-1}\mathbf{1}_K.\end{aligned} \tag{30}$$

These quantities are also known as axis predictivities (Gower et al. 2011).

Next, we briefly consider goodness-of-fit measures for each $\mathbf{X}_i$. The loss contribution for person i towards the interactions is defined as

$$L_{(\mathrm{i})}(i) = \sum_{u=1}^{U} g_{iu}^{(\mathrm{i})} L_{(\mathrm{i})}(i|u) = \sum_{u=1}^{U} g_{iu}^{(\mathrm{i})} \left\| \left(\mathbf{X}_i - \mathbf{C}_u\mathbf{D}_u'\right) \right\|^2. \tag{31}$$

This gives an indication of badness-of-fit, and the sum over all persons gives the minimized value of $L_{(\mathrm{i})}\left(\mathbf{G}^{(\mathrm{i})}, \mathbf{C}, \mathbf{D}\right)$. These loss contributions account for possible differences in origin, scale and/or rotation between a person's interactions and the modelled cluster mean $\mathbf{C}_u\mathbf{D}_u^{'}$. A more informative manner of presenting these loss contributions may be as percentage contributions to $L_{(\mathrm{i})}\left(\mathbf{G}^{(\mathrm{i})}, \mathbf{C}, \mathbf{D}\right)$.

An alternative measure of person fit which is bounded on $[-1, 1]$ is given by

$$p_{\mathrm{fit}}(i) = \sum_{u=1}^{U} g_{iu}^{(\mathrm{i})} \frac{\operatorname{tr} \mathbf{X}_i \mathbf{D}_u \mathbf{C}_u^{'}}{\left\|\mathbf{X}_i\right\| \left\|\mathbf{C}_u \mathbf{D}_u^{'}\right\|}. \tag{32}$$

This only takes into account differences in rotation and origin, and high values indicate good fit whilst negative values indicate poor fit. When the origins coincide, the quantity (32) can be interpreted as a product-moment correlation coefficient between $\operatorname{Vec} \mathbf{X}_i$ and $\operatorname{Vec} \mathbf{C}_u\mathbf{D}_u^{'}$. The notation $\operatorname{Vec} \mathbf{A}$ denotes the vector formed by concatenating the columns of a matrix $\mathbf{A}$ into a single vector.

Finally, we briefly note fit diagnostics for cases (II) and (III). For case (II), we again have an orthogonal decomposition, namely

$$\overline{\mathbf{X}}_{(c)}\overline{\mathbf{X}}_{(c)}^{'} = \left(\mathbf{C}_1\mathbf{D}_*^{'}\right)\left(\mathbf{C}_1\mathbf{D}_*^{'}\right)^{'} + \left(\overline{\mathbf{X}}_{(c)} - \mathbf{C}_1\mathbf{D}_*^{'}\right)\left(\overline{\mathbf{X}}_{(c)} - \mathbf{C}_1\mathbf{D}_*^{'}\right)^{'}.$$

The row fit can therefore be defined as

$$\boldsymbol{r}_{\mathrm{fit}} = \left[\operatorname{diag} \mathbf{C}_1\mathbf{D}_*^{'}\mathbf{D}_*\mathbf{C}_1^{'}\right]\left[\operatorname{diag} \overline{\mathbf{X}}_{(c)}\overline{\mathbf{X}}_{(c)}^{'}\right]^{-1} \mathbf{1}_J$$

and the column fit as

$$\boldsymbol{c}_{\mathrm{fit}} = \left[\operatorname{diag} \mathbf{D}_*\mathbf{C}_1^{'}\mathbf{C}_1\mathbf{D}_*^{'}\right]\left[\operatorname{diag} \overline{\mathbf{X}}_{(c)}^{'}\overline{\mathbf{X}}_{(c)}\right]^{-1} \mathbf{1}_{UK}.$$

For case (III), a similar decomposition is available, and we have

$$\boldsymbol{r}_{\mathrm{fit}} = \left[\operatorname{diag} \mathbf{C}_*\mathbf{D}_1^{'}\mathbf{D}_1\mathbf{C}_*^{'}\right]\left[\operatorname{diag} \overline{\mathbf{X}}_{(r)}\overline{\mathbf{X}}_{(r)}^{'}\right]^{-1} \mathbf{1}_{UJ}$$

for the rows, and similarly

$$\boldsymbol{c}_{\mathrm{fit}} = \left[\operatorname{diag} \mathbf{D}_1\mathbf{C}_*^{'}\mathbf{C}_*\mathbf{D}_1\right]\left[\operatorname{diag} \overline{\mathbf{X}}_{(r)}^{'}\overline{\mathbf{X}}_{(r)}\right]^{-1} \mathbf{1}_K$$

for the columns.

C The latent-class bilinear multinomial logit model

In this appendix, we briefly expand on the LC- BML model of van Rosmalen et al. (2010) mentioned in Sect. 1.1, which was developed to analyze survey data contaminated with response styles.

Response styles occur when respondents use rating scales heterogeneously (e.g., Schoonees et al. 2015; Baumgartner and Steenkamp 2001). The LC- BML model is a parametric finite mixture of multinomial logit models which models the responses to all items jointly. It simultaneously segments respondents into two types of clusters, namely response style and substantive item segments. Similarly to LSBCLUST, the LC- BML model produces biplots describing the relationship between the values and the rating categories within each item segment. The response styles are modelled as marginal effects for the rating categories. A nonparametric equivalent of the LC- BML model can be formulated within the LSBCLUST framework. The resulting model is, except for the inclusion of demographic variables in the LC- BML model, equivalent to the LC- BML model. It has the distinct advantage of being much faster to compute, as least-squares estimation and crisp clustering are used instead of maximum likelihood and finite mixture models.

Whereas LSBCLUST models entries in $\underline{\mathbf{X}}$ directly, the LC- BML model focuses on modelling the probability of a certain response pattern across a number of items measured on a common rating scale. Mathematically, we have

$$\mathsf{P}(X_{ijk} = 1) = \sum_{s=1}^{S} \sum_{u=1}^{U} \pi_{su} \mathsf{P}(X_{ijk} = 1 \mid s, u),$$

where $\{X_{ijk}\}$ are the random variables whose realizations are the entries in $\underline{\mathbf{X}}$. The mixing proportions, or a priori class membership probabilities, are denoted by π_{su}, where s and u indexes the two types of latent classes analogous to the two types of clusters in LSBCLUST. The cluster-specific probabilities are modelled using multinomial logits such that

$$\mathsf{P}(X_{ijk} = 1 \mid s, u) = \frac{\exp\left(\eta_{jk \mid s,u}\right)}{\sum_{j=1}^{J} \exp\left(\eta_{jk \mid s,u}\right)},$$

where $\eta_{jk \mid s,u}$ is a segment-specific linear predictor. The basic form of the linear predictor is

$$\eta_{jk \mid s,u} = \alpha_{j \mid s} + \gamma_{jk \mid u}. \tag{33}$$

Here $\alpha_{j \mid s}$ is the attractiveness of rating category j under response style s, and $\gamma_{jk \mid u}$ captures the joint effect of rating category j for item k under interaction cluster u, after adjusting for $\alpha_{j \mid s}$. To reduce the large number of parameters and to facilitate the use of biplots for interpreting the results, Van Rosmalen et al. (2010) further restrict

(33) to be of the form

$$\eta_{jk\,|\,s,u} = \alpha_{j\,|\,s} + \boldsymbol{c}_{1j}^{'}\boldsymbol{d}_{k\,|\,u}. \tag{34}$$

Here $\boldsymbol{c}_{1j}^{'}$ and $\boldsymbol{d}_{k\,|\,u}^{'}$ form the rows of $\mathbf{C}_1$ and $\mathbf{D}_u$ respectively. Identifiability restrictions are applicable to these parameters—see the original paper for more information.

The LC- BML model specification is completed by the likelihood function:

$$L\,(\mathbf{\Pi}, \mathbf{A}, \mathbf{C}, \mathbf{D}) = \prod_{i=1}^{I}\sum_{s=1}^{S}\sum_{u=1}^{U}\pi_{su}\prod_{j=1}^{J}\prod_{K=1}^{K}\mathsf{P}(X_{ijk} = 1\,|\,s, u).$$

This likelihood function is optimized using an EM-algorithm (Dempster et al. 1977), which requires significantly more computation time than our algorithm in Sect. 5.

A nonparametric equivalent of the LC- BML model can be formulated within the LSBCLUST framework. The data array is constructed by transforming each observation into an indicator matrix, with the rows representing the respective rating categories and the columns the survey items. Each column contains a single one indicating which rating was used to answer that item. In effect, we therefore consider the rating scale as one of the modes in our three-way data set. Choosing Model 2 in Table 2 fits a model containing only row, or, in this context, response style effects, and interactions. Additionally, we use the case (II) from Sect. 6.2 ($\mathbf{C}_1\mathbf{D}_u^{'}$) as a model for the interactions so that the coordinates for the rating categories are fixed across biplots. The resulting model is, except for the inclusion of demographic variables in the LC- BML model, equivalent to the LC- BML model. The similarity between LC- BML and LSBCLUST is apparent from comparing (34) and (7). As for LSBCLUST under case (I), the matrix $\mathbf{C}_1$ contains the coordinates of the rating category effects across all interaction segments in a P-dimensional space, while $\mathbf{D}_u$ contains the item coordinates for interaction cluster u in that same space.

Note that Van Rosmalen et al. (2010) also include effects for demographic variables in the linear predictor. This is not currently possible for LSBCLUST.

An empirical comparison of LC- BML and LSBCLUST, based on an analysis of the list-of-values data conducted in van Rosmalen et al. (2010), is available from the authors upon request.

D Reproducing the empirical example

All computations in this paper were carried out with the **lsbclust** package (Schoonees 2019) for the open-source statistical software environment R (R Core Team 2020). Version 1.1 of **lsbclust** is available for download from the Comprehensive R Archive Network (CRAN) and can be installed from within an R session with the command

```
R> install.packages("lsbclust")
```

The cars data used in Section 4.1 is available as part of **lsbclust** as the `dcars` data set. This is a three-way `array` containing the data for 187 respondents. We can load the package and the data, and inspect the dimensions of `dcars` with the commands

```
R> library("lsbclust")
R> data("dcars")
R> dim(dcars)
[1]  10   8 187
```

The first observation is

```
R> dcars[, , 1]
          Afford    Attr       Safety  Operating Sport  Size  Reli    Features
          ability    activeness          Cost    iness       ability
Citroen     3       6           4        6       6     4     5        5
Fiat        7       6           7        6       6     7     6        6
Ford        6       6           6        6       6     6     6        6
Opel        7       4           4        5       5     5     3        4
Peugeot     6       6           5        6       4     5     5        5
Renault     5       6           6        6       6     6     7        6
Seat        6       5           6        5       4     4     5        5
Toyota      6       6           6        6       6     6     6        6
Volkswagen  3       4           3        5       5     3     3        5
Volvo       5       7           7        6       6     4     6        7
```

Model 9 in Table 2 is fitted ($\boldsymbol{\delta} = c(1, 1, 1, 1)$) with the number of clusters $R = 5$, $S = 5$, $T = 6$ and $U = 8$. Two dimensions are used for the low-rank decomposition of the interaction cluster means (`ndim = 2`) and the coordinates of the car brands are fixed across all interaction biplots (case (II), `fixed = "rows"`). We use 100 random starts (`nstart = 100`) for the interaction clustering and 1000 random starts (`nstart.kmeans = 1000`) for the three k-means parts. The code is

```
R> set.seed(5448)
R> fit <- lsbclust(data = dcars, ndim = 2, delta = c(1, 1, 1, 1),
                   nclust = c(5, 5, 6, 8), nstart = 100,
                   nstart.kmeans = 1000, parallel = TRUE,
                   verbose = -1, fixed = "rows")
K-means on overall means... DONE
K-means on row margins...   DONE
K-means on column margins...    DONE
Interaction clustering (100 starts)...  DONE
```

The object `fit` can now be queried and plotted to produce the results in the accompanying paper. To create the figures, which are returned as graphical objects created by the **ggplot2** package (Wickham 2009), the following code can be used:

- A plot containing the information in Table 4:

```
R> plot(fit, type = "overall")
```

- Figure 3:

```
R> plot(fit, type = "rows")
```

- Figure 4:

```
R> plot(fit, type = "columns")
```

- Figures 5 and 6:

```
R> plot(fit, type = "interactions", legend.position = "none",
          segments = c(FALSE, TRUE))
```

The information in Tables 5and 6 can be accessed by the `summary()` method for the interactions as

```
R> summary(fit$interactions)
```

Note that the rows in Table 5 have been reordered such that the fit statistics are in descending order.

The three persons in Segment O5 can be identified by looking at the cluster membership vector for the overall means. This can be accessed as part of the `"overall"` slot of our fitted object

```
R> which(fit$overall$cluster == 5)
[1] 50 66 85
```

The observations for these persons can be viewed by

```
R> dcars[, , which(fit$overall$cluster == 5)]
```

We can verify that the three persons in Segment O5 also fall in Segment R1 by cross-tabulating the cluster assignment for these two components:

```
R> table(fit$overall$cluster, fit$rows$cluster)

     1  2  3  4  5
  1 55 31 15  2  2
  2 19 20  5  1  0
  3 14  8  3  0  1
  4  7  0  0  1  0
  5  3  0  0  0  0
```

Finally, the code we used for selecting the number of clusters are as follows:

```
R> set.seed(43235)
R> ms <- step.lsbclust(dcars, margin = 3, nclust = 1:15,
                fixed = "rows", nstart = 100,
                parallel = TRUE, nstart.kmeans = 1000)
R> plot(ms, which = 1:5)
```

Information on the R session is as follows:

```
R> sessionInfo()
R version 4.1.0 (2021-05-18)
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: Ubuntu 18.04.5 LTS

Matrix products: default
BLAS:   /usr/lib/x86_64-linux-gnu/blas/libblas.so.3.7.1
LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.7.1

locale:
 [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
 [3] LC_TIME=nl_NL.UTF-8        LC_COLLATE=en_US.UTF-8
 [5] LC_MONETARY=nl_NL.UTF-8    LC_MESSAGES=en_US.UTF-8
 [7] LC_PAPER=nl_NL.UTF-8       LC_NAME=C
 [9] LC_ADDRESS=C               LC_TELEPHONE=C
[11] LC_MEASUREMENT=nl_NL.UTF-8 LC_IDENTIFICATION=C

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
[1] stringr_1.4.0    xtable_1.8-4     lsbclust_1.1     ggplot2_3.3.5
[5] memisc_0.99.27.3 MASS_7.3-54      lattice_0.20-44  knitr_1.33

loaded via a namespace (and not attached):
 [1] Rcpp_1.0.7         pillar_1.6.2       compiler_4.1.0
 [4] plyr_1.8.6         iterators_1.0.13   base64enc_0.1-3
 [7] tools_4.1.0        digest_0.6.27      clue_0.3-59
[10] jsonlite_1.7.2     evaluate_0.14      lifecycle_1.0.0
[13] tibble_3.1.3       gtable_0.3.0       pkgconfig_2.0.3
[16] rlang_0.4.11       foreach_1.5.1      DBI_1.1.1
[19] parallel_4.1.0     mvtnorm_1.1-2      xfun_0.25
[22] gridExtra_2.3      withr_2.4.2        cluster_2.1.2
[25] repr_1.1.3         dplyr_1.0.7        generics_0.1.0
[28] vctrs_0.3.8        tidyselect_1.1.1   grid_4.1.0
[31] glue_1.4.2         data.table_1.14.0  R6_2.5.0
[34] fansi_0.5.0        purrr_0.3.4        reshape2_1.4.4
[37] magrittr_2.0.1     codetools_0.2-18   scales_1.1.1
[40] ellipsis_0.3.2     htmltools_0.5.1.1  assertthat_0.2.1
[43] colorspace_2.0-2   utf8_1.2.2         stringi_1.7.3
[46] doParallel_1.0.16  munsell_0.5.0      crayon_1.4.1
```

References

Abu-El-Haija S, Kothari N, Lee J, Natsev P, Toderici G, Varadarajan B, Vijayanarasimhan S (2016) Youtube-8m: a large-scale video classification benchmark. arXiv:1609.08675

Basford KE, McLachlan GJ (1985) The mixture method of clustering applied to three-way data. J Classif 2(1):109–125

Baumgartner H, Steenkamp JBEM (2001) Response styles in marketing research: a cross-national investigation. J Mark Res 38(2):143–156

Bijmolt TH, van de Velden M (2012) Multiattribute perceptual mapping with idiosyncratic brand and attribute sets. Mark Lett 23(3):585–601

Bro R (2006) Review on multiway analysis in chemistry–2000–2005. Crit Rev Anal Chem 36(3–4):279–293

Cariou V, Alexandre-Gouabau M, Wilderjans TF (2021) Three-way clustering around latent variables approach with constraints on the configurations to facilitate interpretation. J Chemom 35(2):e3269

Carroll JD, Arabie P (1980) Multidimensional scaling. Annu Rev Psychol 31:607–649
Carroll JD, Chang J-J (1970) Analysis of individual differences in multidimensional scaling via an n-way generalization of Eckart–Young decomposition. Psychometrika 35(3):283–319
Cattell RB (1966) The scree test for the number of factors. Multivar Behav Res 1(2):245–276
de Leeuw J (1994) Block-relaxation algorithms in statistics. In: Bock H-H, Lenski W, Richter MM (eds) Inf Syst Data Anal. Springer, pp 308–324
De Silva V, Lim L-H (2008) Tensor rank and the ill-posedness of the best low-rank approximation problem. SIAM J Matrix Anal Appl 30(3):1084–1127
Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the em algorithm. J R Stat Soc Ser B (Methodol) 39(1):1–22
Denis J-B, Gower JC (1994) Asymptotic covariances for the parameters of biadditive models. Utilitas Math 46:193–205
DeSarbo WS, Carroll JD, Lehmann DR, Oshaughnessy J (1982) Three-way multivariate conjoint analysis. Market Sci 1(4):323–350
Dolan ED, Moré JJ (2002) Benchmarking optimization software with performance profiles. Math Program 91(2):201–213
Eckart C, Young G (1936) The approximation of one matrix by another of lower rank. Psychometrika 1(3):211–218
Efron B, Tibshirani RJ (1994) An introduction to the bootstrap. Chapman & Hall/CRC, Boca Raton
Everitt B, Landau S, Leese M, Stahl D (2011) Cluster analysis, 3rd edn. Wiley, Chicester
Faber NKM, Bro R, Hopke PK (2003) Recent developments in candecomp/parafac algorithms: a critical review. Chemom Intell Lab Syst 65(1):119–137
Gabriel KR (1971) The biplot graphic display of matrices with application to principal component analysis. Biometrika 58(3):453–467
Gallaugher MP, McNicholas PD (2020a) Mixtures of skewed matrix variate bilinear factor analyzers. Adv Data Anal Classif 14(2):415–434
Gallaugher MP, McNicholas PD (2020b) Parsimonious mixtures of matrix variate bilinear factor analyzers. In: Imaizumi T, Nakayama A, Yokoyama S (eds) Advanced studies in behaviormetrics and data science: essays in honor of Akinori Okada. Springer, pp 177–196
Genevsky A, Knutson B (2015) Neural affective mechanisms predict market-level microlending. Psychol Sci 26(9):1411–1422
Gower JC (1975) Generalized Procrustes analysis. Psychometrika 40(1):33–51
Gower JC, Dijksterhuis GB (2004) Procrustes problems. Oxford University Press, Oxford
Gower JC, Hand DJ (1996) Biplots. Chapman & Hall, London
Gower JC, Lubbe SG, Le Roux NJ (2011) Understanding biplots. Wiley, Chicester
Hardy A (1996) On the number of clusters. Comput Stat Data Anal 23(1):83–96
Harshman RA (1970) Foundations of the PARAFAC procedure: models and conditions for an "explanatory" multimodal factor analysis. UCLA working papers in phonetics, vol 16, pp 1–84
Hitchcock FL (1927) The expression of a tensor or a polyadic as a sum of products. J Math Phys 6(1–4):164–189
Hubert L, Arabie P (1985) Comparing partitions. J Classif 2(1):193–218
Hunt LA, Basford KE (1999) Fitting a mixture model to three-mode three-way data with categorical and continuous variables. J Classif 16(2):283–296
Kiers HA (2000) Towards a standardized notation and terminology in multiway analysis. J Chemom 14(3):105–122
Kiers HA, Van Mechelen I (2001) Three-way component analysis: principles and illustrative application. Psychol Methods 6(1):84
Kolda TG, Bader BW (2009) Tensor decompositions and applications. SIAM Rev 51(3):455–500
Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. Adv Neural Inf Process Syst 25:1097–1105
Kroonenberg PM (2008) Applied multiway data analysis, vol 702. Wiley, New York
Kroonenberg PM, de Leeuw J (1980) Principal component analysis of three-mode data by means of alternating least squares algorithms. Psychometrika 45(1):69–97
Llobell F, Cariou V, Vigneau E, Labenne A, Qannari EM (2019) A new approach for the analysis of data and the clustering of subjects in a cata experiment. Food Qual Prefer 72:31–39

Llobell F, Cariou V, Vigneau E, Labenne A, Qannari EM (2020) Analysis and clustering of multiblock datasets by means of the statis and clustatis methods. application to sensometrics. Food Qual Prefer 79:103520

Lonsdale J, Thomas J, Salvatore M, Phillips R, Lo E, Shad S, Hasz R, Walters G, Garcia F, Young N et al (2013) The genotype-tissue expression (gtex) project. Nat Genet 45(6):580–585

Meulders M, De Bruecker P (2018) Latent class probabilistic latent feature analysis of three-way three-mode binary data. J Stat Softw 87(1):1–45

Milligan GW (1980) An examination of the effect of six types of error perturbation on fifteen clustering algorithms. Psychometrika 45(3):325–342

Milligan GW, Cooper MC (1985) An examination of procedures for determining the number of clusters in a data set. Psychometrika 50(2):159–179

Papalexakis EE, Sidiropoulos ND, Bro R (2013) From k-means to higher-way co-clustering: Multilinear decomposition with sparse latent factors. IEEE Trans Signal Process 61(2):493–506

R Core Team (2020) R: a language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria

Rocci R, Vichi M (2005) Three-mode component analysis with crisp or fuzzy partition of units. Psychometrika 70(4):715–736

Schmidt E (1907) Zur Theorie der linearen und nichtlinearen Integralgleichungen. I Teil. Entwicklung willkürluchen Funktionen nach System vorgeschriebener. Math Ann 63:433–476

Schoonees PC (2019) lsbclust: Least-squares bilinear clustering for three-way data. R package version 1.1. http://CRAN.R-project.org/package=lsbclust

Schoonees PC, van de Velden M, Groenen PJ (2015) Constrained dual scaling for detecting response styles in categorical data. Psychometrika 80(4):968–994

Stewart G (1980) The efficient generation of random orthogonal matrices with an application to condition estimators. SIAM J Numer Anal 17(3):403–409

Stewart GW (1993) On the early history of the singular value decomposition. SIAM Rev 35(4):551–566

Tucker LR (1966) Some mathematical notes on three-mode factor analysis. Psychometrika 31(3):279–311

van den Burg GJ, Groenen PJ (2016) Gensvm: a generalized multiclass support vector machine. J Mach Learn Res 17(225):1–42

van Rosmalen J, van Herk H, Groenen PJF (2010) Identifying response styles: a latent-class bilinear multinomial logit model. J Mark Res 47:157–172

Vermunt JK (2007) A hierarchical mixture model for clustering three-way data sets. Comput Stat Data Anal 51(11):5368–5376

Vichi M (1999) One-mode classification of a three-way data matrix. J Classif 16(1):27–44

Vichi M, Rocci R, Kiers HA (2007) Simultaneous component and clustering models for three-way data: within and between approaches. J Classif 24(1):71–98

Viroli C (2011) Finite mixtures of matrix normal distributions for classifying three-way data. Stat Comput 21(4):511–522

Wickham H (2009) ggplot2: elegant graphics for data analysis. Springer, New York

Wilderjans TF, Ceulemans E (2013) Clusterwise parafac to identify heterogeneity in three-way data. Chemom Intell Lab Syst 129:87–97