Computational Statistics and Data Analysis 209 (2025) 108179

Contents lists available at ScienceDirect

Computational Statistics and Data Analysis

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

Sparse factor analysis for categorical data with the group-sparse generalized singular value decomposition

Ju-Chi Yu [a,*,1], Julie Le Borgne [b], Anjali Krishnan [c], Arnaud Gloaguen [d], Cheng-Ta Yang [e], Laura A. Rabin [c], Hervé Abdi [g,*,1], Vincent Guillemot [f,*,1]

[a] Campbell Family Mental Health Research Institute, Centre for Addiction and Mental Health, Toronto, Canada
[b] Université de Lille, INSERM, CHU Lille, Institut Pasteur Lille, U1167-riD-AGE, Facteurs de risque et déterminants moléculaires des maladies liées au vieillissement, Lille, France
[c] Department of Psychology, Brooklyn College of the City University of New York, Brooklyn, USA
[d] Centre National de Recherche en Génomique Humaine (CNRGH), Institut de Biologie François Jacob, CEA, Université Paris-Saclay, Évry, France
[e] Department of Psychology, National Cheng Kung University, Tainan, Taiwan
[f] Institut Pasteur, Université Paris Cité, Bioinformatics and Biostatistics Hub, F-75015 Paris, France
[g] School of Behavioral and Brain Sciences, The University of Texas at Dallas, Richardson, TX, USA

ARTICLE INFO

Keywords:
Sparsification
Multivariate analysis
Correspondence analysis
Discriminant correspondence analysis
Multiple correspondence analysis
Discriminant multiple correspondence analysis
Orthogonality

ABSTRACT

Correspondence analysis, multiple correspondence analysis, and their discriminant counterparts (i.e., discriminant simple correspondence analysis and discriminant multiple correspondence analysis) are methods of choice for analyzing multivariate categorical data. In these methods, variables are integrated into optimal components computed as linear combinations whose weights are obtained from a generalized singular value decomposition (GSVD) that integrates specific metric constraints on the rows and columns of the original data matrix. The weights of the linear combinations are, in turn, used to interpret the components, and this interpretation is facilitated when components are 1) pairwise orthogonal and 2) when the values of the weights are either large or small but not intermediate—a configuration called a simple or a sparse structure. To obtain such simple configurations, the optimization problem solved by the GSVD is extended to include new constraints that implement component orthogonality and sparse weights. Because multiple correspondence analysis represents qualitative variables by a set of binary columns in the data matrix, an additional group constraint is added to the optimization problem in order to sparsify the whole set of columns representing one qualitative variable. This method—called group-sparse GSVD (gsGSVD)—integrates these constraints in a new algorithm via an iterative projection scheme onto the intersection of subspaces where each subspace implements a specific constraint. This algorithm is described in details, and we show how it can be adapted to the sparsification of simple and multiple correspondence analysis (as well as their barycentric discriminant analysis versions). This algorithm is illustrated with the analysis of four different data sets—each illustrating the sparsification of a particular CA-based method.

* Corresponding authors.
E-mail addresses: Ju-Chi.Yu@camh.ca (J.-C. Yu), herve@utdallas.edu (H. Abdi), vincent.guillemot@pasteur.fr (V. Guillemot).
[1] These three authors contributed equally.

https://doi.org/10.1016/j.csda.2025.108179
Received 4 January 2024; Received in revised form 21 March 2025; Accepted 24 March 2025

Available online 25 March 2025

1. Introduction

In contemporary research, data sets routinely comprise large samples of observations (e.g., in the hundreds of thousands) often described by an even larger number of variables (e.g., in the millions or more) that can be quantitative, qualitative, or mixtures of these two data types. While principal component analysis (PCA) extracts components from *quantitative* data, typical component methods for qualitative data are correspondence analysis (CA) and multiple correspondence analysis (MCA, a generalization of CA). Like PCA, CA and MCA extract components that summarize the associations between qualitative variables by representing each variable with a set of binary columns corresponding to its levels——a coding scheme called *disjunctive coding* or *group coding* in multivariate statistics, or *one hot encoding* in machine learning (see, e.g., Abdi et al., 2024a).

Component methods create new variables (i.e., the *components*) obtained as linear combinations of the original variables, which optimally represent, on the one hand, the similarity structure of the observations by maps of the component space where the distance between observations approximates their similarity, and, on the other hand, the correlational structure of the variables by maps where the configurations of the variables approximates their similarity (as measured, e.g., by correlation, covariance, or cosine). Components——being linear combinations of the original variables—are often interpreted in terms of these variables (i.e., from the weights of the linear combinations) and are easy to interpret when each component is obtained from a small number of variables where each variable only contributes to a few (ideally one) components. Such an easily interpretable configuration is called a *simple* structure—a concept first elaborated for factor analysis methods by psychometricians (specifically "factorialists" such as, e.g., Thurstone, 1935; Cattell, 1978). By contrast, when the structure is not simple, components are hard to interpret because the contributing variables are hard to identify. To reach a simple solution and facilitate the interpretation of components, these early factorialists (Thurstone, 1935, 1947) developed heuristic procedures such as rotation. These procedures often simplify the interpretation (but at the cost of losing optimality and also sometimes orthogonality), particularly when the data fit the factorial hypotheses (e.g., a signal obtained as a mixture of few well-defined orthogonal dimensions and independent additive Gaussian noise), but large data sets collected without a clear construct are unlikely to naturally have a simple structure recoverable from standard rotation procedures (Cattell, 1978). However, rotation is rarely (if ever) used to facilitate interpretation in the methods from the correspondence analysis family, maybe because these methods were created as descriptive tools (with data unlikely to follow the factorialist model), but finding simple structures (especially for complex data) has recently become more relevant because of the large size of modern data sets.

To obtain a simple structure—rather than by using rotation—a traditional approach in the CA family (see, e.g., Saporta, 2011; Abdi and Béra, 2014) selects items (i.e., rows or columns) whose contributions exceed the average contribution (i.e., the inverse of the number of items), or exceed their a priori mass when the items have different masses (i.e., such as in CA). These descriptive approaches can be complemented by some inferential procedures such as test values (a cousin of Student's t statistics, see, e.g., Saporta, 2011). More recently, computer-based resampling techniques (e.g., bootstrap or permutation) provide non-parametric equivalents—such as bootstrap ratios—to these test values.

By contrast with these earlier heuristics, *sparsification*—the modern approach to "simplification" which originated in a multiple regression framework—reframes simplification as an optimization problem whose goal is to minimize the sum of squared residuals while simultaneously minimizing the sum of the absolute values of the coefficients (a procedure originally called *least absolute shrinkage and selection operator*, better known as LASSO, see, Hastie et al., 2009; Journée et al., 2010; Hastie et al., 2015; Efron and Hastie, 2016). In this context, when a model is sparsified, loadings below a specific threshold are shrunken to zero and are therefore eliminated from the model. Often, these redundant variables provide little *specific* information, and eliminating them from the model makes the prediction more reliable and easier to interpret (Efron, 2020). In the early 2000s, LASSO-based sparsification methods were extended to component methods, such as principal component analysis (PCA, see, e.g., Jolliffe and Uddin 2000; Jolliffe et al. 2003; Zou et al. 2006; Trendafilov 2014; for a recent review see Trendafilov and Gallo 2021, Chapter 4). Both sparsification and rotation try to find a compromise between simplicity and the amount of variance explained by the components. When used for PCA, sparsification appears as an alternative to rotation but—as argued by Trendafilov and Adachi (2015)—sparsification is to be preferred because 1) for big data analysis, ease of interpretation should be prioritized over maximizing explained variance; and 2) concluding that small loadings obtained after a rotation are negligible can be misleading because the rotated dimensions are interpreted by imposing an arbitrary threshold (as illustrated, e.g., by Cadima and Jolliffe, 1995).

Recently, several LASSO-based sparsification methods have been developed that integrate the LASSO with other constraints (Yuan and Lin, 2006; Zou and Hastie, 2005; Hastie et al., 2015). These sparsification methods concurrently maximize the explained variance while penalizing (i.e., reducing or eliminating) non-zero, intermediate loadings. In practice, sparsification is implemented by adding specific constraints to the component maximization problem. Current sparsification methods are largely dominated by extensions of the penalized matrix decomposition (PMD) introduced by Witten et al. (2009) and by sparse PCA from Mackey (2009). However, very few sparse versions of CA and MCA have been developed. As an early notable exception, a version of sparse MCA was introduced by Bernard et al. (2012), whose methodology was grounded in Witten et al. (2009)'s PMD. However, Bernard et al. (2012)'s approach did not rely on an explicit optimization problem nor did it guarantee the orthogonality of the resulting components. In a similar vein, Mori et al. (2016) proposed an alternative version of sparse MCA, based on an iterative algorithm that implements a decomposition of the dataset as an optimization problem under constraints, including orthogonality of loadings and sparsity. However, this method is mostly a heuristic because there is no convergence proof to indicate that this algorithm solves an explicit optimization problem (note that, as of 2023, the software associated with this publication is no longer available). In a very recent major advance, Liu et al. (2023, see also Abdi et al. 2024b), presented the first genuine sparse CA method rooted in PMD. In contrast to previous methods, Liu et al.

included a deflation step (i.e., BiOPD) to minimize the correlation between components (but cannot guarantee true orthogonality, because sparsification is not a linear problem).

All these recent sparse methods sacrifice orthogonality to sparsity, because it is difficult to concurrently obtain orthogonal components and orthogonal loadings (see, Journée et al. 2010; Trendafilov 2014; exceptions are Trendafilov and Jolliffe 2006, Qi et al. 2013, and Jolliffe et al. 2003 that can satisfy one of the constraints, but not both simultaneously). To concurrently obtain orthogonality and sparsification, Guillemot et al. (2019) developed a new SVD algorithm (called the constrained SVD, CSVD), which extends PMD to sparsify loadings while preserving orthogonality. Specifically, the CSVD imposes sparsification and orthogonality constraints by re-framing these constraints as convex spaces and solves the maximization problem by projecting the data onto the intersection of these spaces.

In this paper, we extend the CSVD—by integrating it with the generalized SVD (GSVD)—to sparsify CA-related methods while maintaining the orthogonality of both components and factor scores. To do so, we extend the CSVD algorithm to incorporate 1) the metric matrices specific to CA and MCA and 2) the group constraints necessary for MCA (because variables in MCA are represented by blocks of binary columns). Note that these group constraints can also be applied to groups of rows in CA or MCA. In this paper, we show how the CSVD can be applied to sparsifying CA and MCA as well as the barycentric discriminant analysis (Abdi, 2007a) versions of CA (DiSCA) and MCA (DiMCA).

In the following sections, we show how group sparsity can be used for CA-based methods, which properties are kept or lost, and how to compute optimal values for the sparsity parameters. We illustrate these procedures with four examples (one per method). The R packages for the methods are available on https://github.com/juchiyu/SPAFAC and https://github.com/vguillemot/sGSVD, and the reproducible R code of the analysis is accessible online in the vignette of the SPAFAC package.

2. Notations and definitions

Bold uppercase letters (e.g., $\mathbf{A}$) denote a matrix, bold lowercase letters (e.g., $\mathbf{a}$) denote a (column) vector, and italic lowercase letters (e.g., a) denote the elements of a matrix or of a vector; the indices for the elements of a set are denoted by italic lowercase letters (e.g., n) and the cardinal of a set by an italic uppercase letter (e.g., N). The matrix $\mathbf{I}$ is the identity matrix, $\mathbf{1}$ is a matrix (or vector) of ones, and $\mathbf{0}$ a matrix (or vector) of zeros (the dimensions of these matrices depend upon the context and are assumed to be conformable). The transpose operation for a matrix is denoted by the superscript $^\top$ (e.g., $\mathbf{A}^\top$) and the inverse of a square matrix (say $\mathbf{S}$) is denoted by the superscript $^{-1}$ (e.g., $\mathbf{S}^{-1}$). The Moore-Penrose pseudo-inverse of a matrix is denoted by the superscript $^+$ (e.g., $\mathbf{S}^+$ is the pseudo-inverse of $\mathbf{S}$). For a given data table, (except if stated otherwise) the number of rows is denoted I, and the number of columns is denoted J. The element stored in the ith row and jth column of matrix $\mathbf{X}$ is denoted $x_{i,j}$. For an $I \times J$ matrix, the minimum of I and J gives its largest possible *rank*, denoted L. The actual rank of a matrix is denoted by R (with $R \leq L$). The operator $\mathsf{diag}\,()$ applied to a (square) matrix gives a vector with all elements on the diagonal of this matrix [i.e., $\mathsf{diag}\,(\mathbf{X}) = \mathbf{x}$, with $x_i = x_{i,i}$] and when applied to a vector, $\mathsf{diag}\,()$ gives a diagonal matrix with the elements of this vector on the diagonal and 0s elsewhere [i.e., $\mathbf{X} = \mathsf{diag}\,(\mathbf{x})$, with $x_i = x_{i,i}$ and $x_{i,j} = 0$ when $i \neq j$]. The operator $\mathsf{trace}\,()$ applied to a (square) matrix gives the sum of all its diagonal elements. The sum of squares of all the elements of $\mathbf{X}$ is called the *inertia* of $\mathbf{X}$ and is equal to $\mathsf{trace}\left(\mathbf{X}^\top\mathbf{X}\right)$ and $\mathsf{trace}\left(\mathbf{X}\mathbf{X}^\top\right)$. The juxtaposition of two matrices indicates their standard product, when the product needs to be expressed explicitly a dot is used (e.g., $\mathbf{XY}$ and $\mathbf{X}\cdot\mathbf{Y}$ both denote the product of $\mathbf{X}$ and $\mathbf{Y}$). The Hadamar (i.e., element-wise) product between two matrices of same dimensions is denoted $\odot$ (i.e., $\mathbf{A}\odot\mathbf{B}$). A probability matrix contains only non-negative elements whose sum is equal to 1. A row (respectively column) profile denoted $\mathbf{i}$ (respectively $\mathbf{j}$) contains only non-negative elements whose sum is equal to 1. The operator

$$\underset{\mathbf{x}}{\arg\max}\,(f(\mathbf{x}))$$

identifies the argument $\mathbf{x}$ which maximizes the value of $f(\mathbf{x})$. Similarly, the operator

$$\underset{\mathbf{x}}{\arg\min}\,(f(\mathbf{x}))$$

identifies the argument $\mathbf{x}$ which minimizes $f(\mathbf{x})$. The $\mathcal{L}_1$ norm of vector $\mathbf{a}$ is denoted by $\|\mathbf{a}\|_1$, and is computed as the sum of all the absolute values of the elements of $\mathbf{a}$; the $\mathcal{L}_2$ norm of vector $\mathbf{a}$, denoted by $\|\mathbf{a}\|_2$, is computed as $\sqrt{\mathbf{a}^\top\mathbf{a}}$. The orthogonal complement of $\mathbf{A}$ is denoted by the superscript $^\perp$ (e.g., $\mathbf{A}^\perp$) and is the vector space of all the vectors orthogonal to the space spanned by $\mathbf{A}$. The space that contains all the vectors with an $\mathcal{L}_1$ norm inferior to s is called an $\mathcal{L}_1$-ball, and is denoted by $\mathscr{B}_1(s)$, with s being the radius of the $\mathcal{L}_1$-ball. The space that contains all the vectors with an $\mathcal{L}_2$ norm inferior to the scalar s is called an $\mathcal{L}_2$-ball; and is denoted by $\mathscr{B}_2(s)$, with s being the radius of the $\mathcal{L}_2$-ball. See also Appendix B for notation specific to the GSVD. Sparsified vectors, matrices, and their elements are denoted with a $\dot{}$ on top (e.g., $\dot{\mathbf{P}}$ is the sparsified version of $\mathbf{P}$).

3. Method

CA-related methods use the generalized singular value decomposition (GSVD, see, e.g., Abdi, 2007b) to compute their components. Therefore, to sparsify these methods, we developed a new sparsification algorithm for the GSVD called the sparse GSVD (sGSVD; Yu et al., 2023). Further, as some CA-related methods analyze categorical data where qualitative variables are represented by groups of (binary) columns, we also extended the sGSVD to create the group-sparse GSVD (gsGSVD) which, in addition, performs group sparsification where pre-defined groups of columns or rows are kept or eliminated together. In the following sections, we present the

optimization problem of the gsGSVD, and the algorithm we developed to solve it. Next, we show how to apply the gsGSVD to sparsify CA, DiSCA, MCA, and DiMCA.

3.1. The group-sparse generalized SVD (gsGSVD)

To sparsify variables as groups of levels for CA-related methods, we extend the algorithm of the sGSVD to develop the group-sparse generalized SVD (gsGSVD). This new approach sparsifies the elements of the generalized singular vectors of a given matrix $\mathbf{X}$ taking into account group constraints (on rows and columns) and metric constraints imposed by a metric matrix for the rows (denoted by $\mathbf{M}$) and by a metric matrix for the columns (denoted by $\mathbf{W}$). Specifically, the gsGSVD maximization problem is expressed as:

$$\underset{\dot{\mathbf{p}}_\ell, \dot{\mathbf{q}}_\ell}{\arg\max} \left(\dot{\delta}_\ell = \dot{\mathbf{p}}_\ell^\top \mathbf{M}^{\frac{1}{2}} \mathbf{X} \mathbf{W}^{\frac{1}{2}} \dot{\mathbf{q}}_\ell \right) \quad \text{subject to}$$

$$\dot{\mathbf{p}}_\ell \in \begin{cases} \mathscr{B}_{\mathcal{L}_2}(1) \\ \mathscr{B}_{\mathcal{G}_{\dot{\mathbf{p}}}}(s_{\dot{\mathbf{p}},\ell}) \\ \dot{\mathbf{P}}_\ell^\perp \end{cases}, \quad \dot{\mathbf{q}}_\ell \in \begin{cases} \mathscr{B}_{\mathcal{L}_2}(1) \\ \mathscr{B}_{\mathcal{G}_{\dot{\mathbf{q}}}}(s_{\dot{\mathbf{q}},\ell}) \\ \dot{\mathbf{Q}}_\ell^\perp \end{cases} \tag{1}$$

with respect to three constraints, involving: 1) the $\mathcal{L}_2$-ball that normalizes the singular vectors, 2) the orthogonal space (i.e., $\dot{\mathbf{P}}_\ell^\perp$ or $\dot{\mathbf{Q}}_\ell^\perp$) that ensures orthogonality between components, and 3) the $\mathcal{L}_\mathcal{G}$-ball that sparsifies the elements of the (generalized) singular vectors in whole groups. For Dimension 1 (i.e., when $\ell = 1$), there is no orthogonal constraint; for $\ell > 1$, then all the previously estimated pseudo singular vectors $\dot{\mathbf{p}}_\ell$ (resp. $\dot{\mathbf{q}}_\ell$) are concatenated into a matrix $\dot{\mathbf{P}}_\ell$ (resp. $\dot{\mathbf{Q}}_\ell$). Next the orthogonality constraint looks for possible solutions in the orthogonal complement of $\dot{\mathbf{P}}_\ell$ (resp. $\dot{\mathbf{Q}}_\ell$) noted $\dot{\mathbf{P}}_\ell^\perp$ (resp. $\dot{\mathbf{Q}}_\ell^\perp$). The ℓth generalized left and right pseudo-singular vectors of $\mathbf{X}$ are then computed as $\dot{\mathbf{u}}_\ell = \mathbf{M}^{-\frac{1}{2}} \dot{\mathbf{p}}_\ell$ and $\dot{\mathbf{v}}_\ell = \mathbf{W}^{-\frac{1}{2}} \dot{\mathbf{q}}_\ell$. We solve this maximization problem by modifying the algorithm of the sGSVD (see Algorithm 1; cf. Algorithm E.4 in Appendix F), where the three constraints are implemented with the Projection Onto Convex Sets (POCS) procedure (Combettes, 1993). In brief, to obtain the projection onto the intersection of several convex sets, POCS iteratively projects the input vector onto each of the sets until convergence. Specifically, the projection onto the $\mathcal{L}_2$-ball is equivalent to imposing constraints on the $\mathcal{L}_2$-norm, and the projection onto the $\mathcal{L}_\mathcal{G}$-norm ball is equivalent to imposing constraints on the group norm—defined as: $\|\mathbf{x}\|_\mathcal{G} = \sum_{g=1}^{G} \left\| \mathbf{x}_{t_g} \right\|_2$ (van den Berg et al., 2008). This group norm—called "the [1,2]-group norm" by van den Berg et al. (2008)—is the $\mathcal{L}_1$-norm of the vector containing the $\mathcal{L}_2$-norm of the sub-vectors defined by the groups. In this paper, we call this group norm the $\mathcal{L}_\mathcal{G}$-norm and its associated space the $\mathcal{L}_\mathcal{G}$-ball, denoted $\mathscr{B}_{\mathcal{G}_\cdot}(\cdot)$. Finally, we reorder the dimensions in the decreasing order of the derived pseudo-singular values ($\dot{\delta}_\ell$). This final step is necessary because, as is the case for the sparse SVD, there is no guarantee that the gsGSVD will estimate the pseudo-singular values in a predefined order, especially when the sparsity constraint is strong. The convergence of the algorithm behind the CSVD is studied in greater details in Guillemot et al. (2019). Because the gsGSVD is based on the CSVD, it keeps the same convergence properties: it will converge to a fixed point solution, but not necessarily to the global optimum, nor monotonously. In order to visualize the convergence properties of gsGSVD on a small realistic simulated dataset, we applied it to a 20×12 count matrix, in a setting similar to sparse CA. The results of this simulation study are detailed in Appendix G.

3.2. Sparse CA and related methods

In this paper, we present four sparse CA-related methods: 1) CA, 2) MCA, and their respective discriminant analysis versions: 3) DiSCA and 4) DiMCA. These methods are all specific cases of the gsGSVD, because they all use metric matrices and, in some cases, incorporate group structures on rows or columns. For example, CA (and DiSCA) analyzes a contingency table and includes metric matrices for both rows and columns, whereas MCA (and DiMCA) analyzes categorical variables represented with disjunctive coding (i.e., a block of binary columns) and thus, additionally, takes into account a group structure on the columns. In the following sections, we expose the algorithms for each method as specific cases of the gsGSVD.

3.2.1. CA and sparse CA (sCA): sparsification with metric matrices

CA analyzes an I by J contingency table (denoted $\mathbf{A}$) by first computing its associated probability matrix[2] (denoted $\mathbf{Z}$). To do so, given N observations (i.e., the grand total of $\mathbf{A}$), $\mathbf{Z}$ is computed as:

$$\mathbf{Z} = \frac{1}{N} \mathbf{A}. \tag{2}$$

[2] Note that this is one way of performing CA. Other ways based on row/column profiles are equivalent and described by Greenacre (1984) and Abdi and Williams, (2010).

Algorithm 1: General algorithm of group-sparse GSVD of $\mathbf{X}$.

Data: $\mathbf{X}$
ϵ (error), R (rank, > 1),
$\mathbf{M}$ and $\mathbf{W}$ (row and column metric matrices),
$s_{\mathbf{p},\ell}$ and $s_{\mathbf{q},\ell}$ (sparse parameters for singular vectors),
$\mathcal{G}_{\mathbf{p}}$, and $\mathcal{G}_{\mathbf{q}}$ (group arrangements of rows and columns)
Result: group-sparse GSVD of $\mathbf{X}$
$\widetilde{\mathbf{X}} = \mathbf{M}^{\frac{1}{2}}\mathbf{X}\mathbf{W}^{\frac{1}{2}}$
$\dot{\mathbf{P}} \leftarrow \dot{\mathbf{Q}} \leftarrow \dot{\mathbf{U}} \leftarrow \dot{\mathbf{V}} \leftarrow \emptyset$; ▷ $\dot{\mathbf{P}}$ and $\dot{\mathbf{Q}}$ store the left and right pseudo-singular vectors
▷ $\dot{\mathbf{U}}$ and $\dot{\mathbf{V}}$ store the *generalized* left and right pseudo-singular vectors
for $\ell = 1, \ldots, R$ **do**
  Initialize $\dot{\mathbf{p}}^{(0)}$ and $\dot{\mathbf{q}}^{(0)}$ ▷ Initialize $\dot{\mathbf{p}}$ and $\dot{\mathbf{q}}$ either from SVD or randomly
  $\dot{\delta}^{(0)} \leftarrow \dot{\mathbf{p}}^{(0)\top}\widetilde{\mathbf{X}}\dot{\mathbf{q}}^{(0)}$ ▷ Initialize $\dot{\delta}$ with $\dot{\mathbf{p}}^{(0)}$ and $\dot{\mathbf{q}}^{(0)}$
  $t \leftarrow 0$
  while $\left(\left\|\dot{\mathbf{p}}^{(t+1)} - \dot{\mathbf{p}}^{(t)}\right\|_2 \geq \epsilon\right)$ or $\left(\left\|\dot{\mathbf{q}}^{(t+1)} - \dot{\mathbf{q}}^{(t)}\right\|_2 \geq \epsilon\right)$ **do**
    $\dot{\mathbf{p}}^{(t+1)} \leftarrow \text{proj}(\widetilde{\mathbf{X}}\dot{\mathbf{q}}^{(t)}, \mathcal{B}_{\mathcal{G}_{\mathbf{p}}}(s_{\mathbf{p},\ell}) \cap \mathcal{B}_{\mathcal{L}_2}(1) \cap \dot{\mathbf{P}}^{\perp})$
    $\dot{\mathbf{q}}^{(t+1)} \leftarrow \text{proj}(\widetilde{\mathbf{X}}^{\top}\dot{\mathbf{p}}^{(t+1)}, \mathcal{B}_{\mathcal{G}_{\mathbf{q}}}(s_{\mathbf{q},\ell}) \cap \mathcal{B}_{\mathcal{L}_2}(1) \cap \dot{\mathbf{Q}}^{\perp})$
    $\dot{\delta}^{(t+1)} \leftarrow \dot{\mathbf{p}}^{(t+1)\top}\widetilde{\mathbf{X}}\dot{\mathbf{q}}^{(t+1)}$
    $t \leftarrow t + 1$ ▷ Iterate until $\dot{\mathbf{p}}$ and $\dot{\mathbf{q}}$ are stable
  $\dot{\delta}_\ell \leftarrow \dot{\delta}^{(t)}$; $\dot{\mathbf{p}}_\ell \leftarrow \dot{\mathbf{p}}^{(t)}$; $\dot{\mathbf{q}}_\ell \leftarrow \dot{\mathbf{q}}^{(t)}$ ▷ $\dot{\mathbf{p}}_\ell$ and $\dot{\mathbf{q}}_\ell$ are the left and right pseudo-singular vectors of $\widetilde{\mathbf{X}}$
  $\dot{\mathbf{u}}_\ell \leftarrow \mathbf{M}^{-\frac{1}{2}}\dot{\mathbf{p}}_\ell$
  $\dot{\mathbf{v}}_\ell \leftarrow \mathbf{W}^{-\frac{1}{2}}\dot{\mathbf{q}}_\ell$ ▷ $\dot{\mathbf{u}}_\ell$ and $\dot{\mathbf{v}}_\ell$ are the left and right generalized pseudo-singular vectors of $\widetilde{\mathbf{X}}$
  $\dot{\mathbf{P}} \leftarrow \left[\dot{\mathbf{P}} \mid \dot{\mathbf{p}}_\ell\right]$; $\dot{\mathbf{Q}} \leftarrow \left[\dot{\mathbf{Q}} \mid \dot{\mathbf{q}}_\ell\right]$
  $\dot{\mathbf{U}} \leftarrow \left[\dot{\mathbf{U}} \mid \dot{\mathbf{u}}_\ell\right]$; $\dot{\mathbf{V}} \leftarrow \left[\dot{\mathbf{V}} \mid \dot{\mathbf{v}}_\ell\right]$
Define $\dot{\boldsymbol{\delta}} = \left(\dot{\delta}_1, \cdots, \dot{\delta}_R\right)$
Reorder $\dot{\boldsymbol{\delta}}$ in decreasing order of its elements
$\dot{\boldsymbol{\Delta}} \leftarrow \text{diag}\left(\dot{\boldsymbol{\delta}}\right)$
Reorder the columns of $\dot{\mathbf{U}}$ and $\dot{\mathbf{V}}$ according to $\dot{\boldsymbol{\delta}}$

Note: The text colored in red is the sparsification constraints of the CSVD that are also used in the gsGSVD; the text colored in green is the metric constraints of the GSVD that are also used in the gsGSVD; and the text colored in blue is the group constraints that are specific to the gsGSVD. (For interpretation of the colors in the algorithm/figure(s), the reader is referred to the web version of this article.)

To analyze the pattern of associations between rows and columns, CA first computes the deviation from independence for $\mathbf{Z}$:

$$\mathbf{X} = \mathbf{Z} - \mathbf{r}\mathbf{c}^{\top} \quad \text{where} \quad \mathbf{r} = \mathbf{Z}\mathbf{1} \quad \text{and} \quad \mathbf{c} = \mathbf{Z}^{\top}\mathbf{1}. \tag{3}$$

Here, $\mathbf{r}$ and $\mathbf{c}$ store (respectively) the sums of the rows and of the columns of $\mathbf{Z}$. Therefore, taking this deviation from independence is equivalent to centering both the rows and the columns of $\mathbf{Z}$ (i.e., *double*-centering $\mathbf{Z}$). CA uses the GSVD to decompose this double centered probability matrix $\mathbf{X}$ into 3 matrices:

$$\mathbf{X} = \mathbf{U}\boldsymbol{\Delta}\mathbf{V}^{\top} \quad \text{under the constraints} \quad \mathbf{U}^{\top}\mathbf{D}_{\mathbf{r}}^{-1}\mathbf{U} = \mathbf{V}^{\top}\mathbf{D}_{\mathbf{c}}^{-1}\mathbf{V} = \mathbf{I}, \tag{4}$$

where $\mathbf{D}_{\mathbf{r}} = \text{diag}(\mathbf{r})$, $\mathbf{D}_{\mathbf{c}} = \text{diag}(\mathbf{c})$, and with $\boldsymbol{\Delta}$ being the diagonal matrix of the singular values (denoted δ) of $\mathbf{X}$ ordered by magnitude (recall that squared singular values are eigenvalues, denoted λ).

In this GSVD, the diagonal metric matrix $\mathbf{D}_{\mathbf{r}}^{-1}$ stores the row weights of $\mathbf{X}$, and the diagonal metric $\mathbf{D}_{\mathbf{c}}^{-1}$ stores the column weights of $\mathbf{X}$. With these metric matrices, CA gives the optimal decomposition for a given rank of the χ^2 associated to the contingency table $\mathbf{A}$ (see Appendix A). In addition, this GSVD is equivalent to the SVD of Equation (B.5) and solves the same maximization problem (cf. Equation (B.4)) with $\mathbf{M} = \mathbf{D}_{\mathbf{r}}^{-1}$ and $\mathbf{W} = \mathbf{D}_{\mathbf{c}}^{-1}$. As a result, this maximization can be solved by the alternating least squares (ALS) Algorithm (see Algorithm E.2 in Appendix F).

In CA, each column of $\mathbf{U}$ (respectively $\mathbf{V}$) stores the loadings for rows (respectively columns) that are used to compute the row factor scores (denoted $\mathbf{F}$) and the column factor scores (denoted $\mathbf{G}$) for each component, where

$$\mathbf{F} = \mathbf{D}_{\mathbf{r}}^{-1}\mathbf{U}\boldsymbol{\Delta} = \mathbf{D}_{\mathbf{r}}^{-1}\mathbf{X}\mathbf{D}_{\mathbf{c}}^{-1}\mathbf{V} \quad \text{and} \quad \mathbf{G} = \mathbf{D}_{\mathbf{c}}^{-1}\mathbf{V}\boldsymbol{\Delta} = \mathbf{D}_{\mathbf{c}}^{-1}\mathbf{X}^{\top}\mathbf{D}_{\mathbf{r}}^{-1}\mathbf{U}. \tag{5}$$

For each component, the contributions of the rows and the columns are computed from the squared weighted loadings:

$$\mathbf{ctr}_{\mathbf{r},\ell} = \left(\mathbf{D}_\mathbf{r}^{-\frac{1}{2}}\mathbf{u}_\ell\right) \odot \left(\mathbf{D}_\mathbf{r}^{-\frac{1}{2}}\mathbf{u}_\ell\right) \quad \text{and} \quad \mathbf{ctr}_{\mathbf{c},\ell} = \left(\mathbf{D}_\mathbf{c}^{-\frac{1}{2}}\mathbf{v}_\ell\right) \odot \left(\mathbf{D}_\mathbf{c}^{-\frac{1}{2}}\mathbf{v}_\ell\right), \tag{6}$$

where the contributions of a given component sum to 1 (i.e., $\mathbf{u}_\ell^\top \mathbf{D}_\mathbf{r}^{-1}\mathbf{u}_\ell = \mathbf{v}_\ell^\top \mathbf{D}_\mathbf{c}^{-1}\mathbf{v}_\ell = 1$).

For sCA, the gsGSVD is used to sparsify the generalized singular vectors $\mathbf{U}$ and $\mathbf{V}$ and gives generalized *pseudo*-singular vectors $\dot{\mathbf{U}}$ and $\dot{\mathbf{V}}$ with sparsified loadings shrunken to 0 to facilitate the interpretation of the components. The sparsification of CA is a specific case of the gsGSVD where the row metric matrices $\mathbf{M}$ and $\mathbf{W}$ are respectively equal to $\mathbf{D}_\mathbf{r}^{-1}$ and $\mathbf{D}_\mathbf{c}^{-1}$. When the data include group structures for rows or columns and should, therefore, be sparsified accordingly, the group constraints can be applied by specifying the sparsification parameters $s_{\mathbf{p},\ell}$ and $s_{\mathbf{q},\ell}$ to include group designs of rows and of columns. When group constraints are not included, the algorithm becomes the sGSVD (Algorithm E.4 in Appendix F).

In the sCA algorithm, the POCS procedure projects the data onto the intersection of three constrained spaces: 1) the $\mathcal{L}_2$-ball that normalizes the singular vectors, 2) the orthogonal space (i.e., $\dot{\mathbf{P}}_\ell^\perp$ or $\dot{\mathbf{Q}}_\ell^\perp$) that ensures orthogonality between components, and 3) the $\mathcal{L}_1$-ball that sparsifies the elements of the (generalized) singular vectors (i.e., $\mathbf{p}$, $\mathbf{q}$, $\mathbf{u}$, and $\mathbf{v}$) or the $\mathcal{L}_\mathcal{G}$-ball if they are sparsified in groups.

From the generalized pseudo-singular vectors, the factor scores are computed as:

$$\dot{\mathbf{F}} = \mathbf{D}_\mathbf{r}^{-1}\dot{\mathbf{U}}\dot{\mathbf{\Delta}} \quad \text{and} \quad \dot{\mathbf{G}} = \mathbf{D}_\mathbf{c}^{-1}\dot{\mathbf{V}}\dot{\mathbf{\Delta}}. \tag{7}$$

It is worth noting that, with sparsification,

$$\dot{\mathbf{F}} \neq \mathbf{D}_\mathbf{r}^{-1}\mathbf{X}\mathbf{D}_\mathbf{c}^{-1}\dot{\mathbf{V}} \quad \text{and} \quad \dot{\mathbf{G}} \neq \mathbf{D}_\mathbf{c}^{-1}\mathbf{X}^\top\mathbf{D}_\mathbf{r}^{-1}\dot{\mathbf{U}}. \tag{8}$$

Here, we choose to compute the factor scores as the scaled generalized *pseudo*-singular vectors (i.e., Equation (7)) over the linear combinations of the original data (i.e., Equation (8)) so that the sparsity of the generalized *pseudo*-singular vectors propagates to the factor scores. It is worth noting that, in plain CA, the inertia (i.e., the sum of the eigenvalues, denoted $\mathcal{I}$) is directly related to the independence χ^2 statistic (i.e., $\chi^2 = N \times \mathcal{I}$)— a consequence of both Equations (7) and (8) (see Appendix A). In sCA, by choosing Equation (7) over Equation (8), the inertia quantified by the *pseudo*-eigenvalues of sCA is no longer directly linked to the χ^2 statistic. However, these pseudo-eigenvalues can still be considered as an inertia. The loss and preservation of other properties are discussed in Section 3.3.

3.2.2. Discriminant simple CA (DiSCA) and sDiSCA: sparsification for discriminant analysis

The discriminant analysis of (simple) CA (as opposed to *multiple* CA) is used to analyze a contingency table whose rows are nested in groups; Here we seek to extract components that best explain the inertia *between* these groups of rows. To do so, DiSCA performs CA on the group summed across the rows. Formally, DiSCA computes the matrix of group sums (denoted by $\mathbf{A}_\mathcal{G}$) using an $I \times I_\mathcal{G}$ indicator matrix $\mathbf{H}$ where values of 1s (and 0s otherwise) indicate the group to which the rows belong. The matrix $\mathbf{A}_\mathcal{G}$ is therefore computed as:

$$\mathbf{A}_\mathcal{G} = \mathbf{H}^\top\mathbf{A}, \tag{9}$$

where the rows of $\mathbf{A}_\mathcal{G}$ store the $I_\mathcal{G}$ row groups and the columns of $\mathbf{A}_\mathcal{G}$ correspond to the J levels of the categorical variable on the columns. Similar to CA, DiSCA then computes the probability matrix by dividing $\mathbf{A}_\mathcal{G}$ by its total N:

$$\mathbf{Z}_\mathcal{G} = \frac{1}{N}\mathbf{A}_\mathcal{G}. \tag{10}$$

Next, this probability matrix $\mathbf{Z}_\mathcal{G}$ is double centered (cf. Equation (3)) to give the resulting matrix $\mathbf{X}_\mathcal{G}$, which is then decomposed by the GSVD (cf. Equation (4)). Then, we define $\mathbf{D}_{\mathbf{r},\mathcal{G}}^{-1}$ as the row metric matrix of $\mathbf{X}_\mathcal{G}$. Finally, the row and column factors are computed in the same way as in plain CA:

$$\mathbf{F} = \mathbf{D}_{\mathbf{r},\mathcal{G}}^{-1}\mathbf{U}\mathbf{\Delta} = \mathbf{D}_{\mathbf{r},\mathcal{G}}^{-1}\mathbf{X}_\mathcal{G}\mathbf{D}_\mathbf{c}^{-1}\mathbf{V} \quad \text{and} \quad \mathbf{G} = \mathbf{D}_\mathbf{c}^{-1}\mathbf{V}\mathbf{\Delta} = \mathbf{D}_\mathbf{c}^{-1}\mathbf{X}_\mathcal{G}^\top\mathbf{D}_{\mathbf{r},\mathcal{G}}^{-1}\mathbf{U}. \tag{11}$$

The contributions of rows and columns are also computed according to Equation (6).

In DiSCA, the row factor score $f_{i,\ell}$ represents the $i_\mathcal{G}$th group on the ℓth component, and the column factor score $g_{j,\ell}$ represents the jth column on the ℓth component. The individuals within each group—stored in the rows of $\mathbf{A}$—can be projected onto the component space as supplementary rows. Formally, to compute the projection (denoted $\mathbf{f}_i^*$) of a row profile (active or supplementary), denoted $\mathbf{i}$, onto the components we use this equation

$$\mathbf{f}_i^* = \mathbf{\Delta}^{-1}\mathbf{G}^\top\mathbf{i}. \tag{12}$$

To sparsify DiSCA, sDiSCA uses the gsGSVD to sparsify the generalized singular vectors $\mathbf{U}$ and $\mathbf{V}$ and computes the generalized *pseudo*-singular vectors $\dot{\mathbf{U}}$ and $\dot{\mathbf{V}}$. Just like sCA, sDiSCA is a specific case of the gsGSVD where the row metric matrix $\mathbf{M}$ is equal to $\mathbf{D}_{\mathbf{r},\mathcal{G}}^{-1}$ and the column metric matrix $\mathbf{W}$ is equal to $\mathbf{D}_\mathbf{c}^{-1}$ with optional group constraints that can be implemented by specifying the sparsification parameters $s_{\mathbf{p},\ell}$ and $s_{\mathbf{q},\ell}$. The only difference between DiSCA and sDiSCA is that the row metrics for sDisCA are now

computed from the row groups instead of being computed from the individual rows. From these generalized pseudo-singular vectors, the factor scores are computed (using Equation (7)) as in sCA.

Just like in sCA, the POCS procedure in the algorithm of sDiSCA projects the data onto the intersection of three constrained spaces: 1) the $\mathcal{L}_2$-ball that normalizes the singular vectors, 2) the orthogonal space (i.e., $\dot{\mathbf{P}}_\ell^\perp$ or $\dot{\mathbf{Q}}_\ell^\perp$) that ensures orthogonality between components, and 3) the $\mathcal{L}_1$-ball that sparsifies the elements of the (generalized) singular vectors (i.e., $\mathbf{p}$, $\mathbf{q}$, $\mathbf{u}$, and $\mathbf{v}$) or the $\mathcal{L}_\mathcal{G}$-ball if the data are sparsified in groups.

3.2.3. MCA and sparse MCA (sMCA): sparsification with metric matrices and group constraints for columns

MCA extends CA to analyze the pattern of associations between several categorical variables by using an *indicator matrix* to represent each variable with a group of 0/1 columns (Greenacre, 1984; Lebart et al., 1984; Abdi and Valentin, 2007), where each column codes a level of one variable with a value of 1 indicating the presence of this level (and 0 its absence). MCA concatenates the indicator matrices of all variables and then uses the same procedure as in CA to process and extract orthogonal components from this concatenated table.

Formally, consider a matrix $\mathbf{A}$ with I observations and K categorical variables, where J_k denotes the number of levels of the kth variable and with a total of $J = \sum_{k=1}^{K} J_k$ columns. Matrix $\mathbf{A}$ with a sub-table structure of indicator matrices can be expressed as

$$\mathbf{A} = \left[\mathbf{A}_1 \mid \mathbf{A}_2 \mid \mathbf{A}_3 \mid \cdots \mid \mathbf{A}_k \mid \cdots \mid \mathbf{A}_K\right], \tag{13}$$

where $\mathbf{A}_k$ is the indicator matrix for the k-th categorical variable. Note that the sum of each row of $\mathbf{A}$ is equal to K, the total sum of each $\mathbf{A}_k$ is equal to I, and, therefore, the sum of all entries of $\mathbf{A}$ is equal to $I \times K$. Just like in CA, MCA first computes the probability matrix $\mathbf{Z}$ by dividing $\mathbf{A}$ by the sum of all its entries (cf. Equation (2)):

$$\mathbf{Z} = \frac{1}{I \times K}\mathbf{A}, \tag{14}$$

and then double centers the probability matrix to obtain (cf. Equation (3)):

$$\mathbf{X} = \mathbf{Z} - \mathbf{r}\mathbf{c}^\top, \tag{15}$$

where $\mathbf{r}$ is the vector that stores the sums of each row (i.e., $\mathbf{r} = \mathbf{Z1}$) and $\mathbf{c}$ is the vector that stores the sums of each column (i.e., $\mathbf{c} = \mathbf{Z}^\top\mathbf{1}$). This double centered probability matrix $\mathbf{X}$ is then decomposed by the GSVD:

$$\mathbf{X} = \mathbf{U}\boldsymbol{\Delta}\mathbf{V}^\top \quad \text{subject to} \quad \mathbf{U}^\top\mathbf{D}_\mathbf{r}^{-1}\mathbf{U} = \mathbf{V}^\top\mathbf{D}_\mathbf{c}^{-1}\mathbf{V} = \mathbf{I}, \tag{16}$$

where $\mathbf{D}_\mathbf{r} = \mathsf{diag}(\mathbf{r})$, $\mathbf{D}_\mathbf{c} = \mathsf{diag}(\mathbf{c})$, and $\boldsymbol{\Delta}$ is the diagonal matrix storing the generalized singular values. In addition, just like in CA, such a GSVD is equivalent to the SVD from Equation (B.5) and solves the same maximization problem (cf. Equation (B.4)) with $\mathbf{M} = \mathbf{D}_\mathbf{r}^{-1}$ and $\mathbf{W} = \mathbf{D}_\mathbf{c}^{-1}$. As a result, this maximization problem can be solved by using the same Algorithm E.2 as for CA. Finally, the row and column factors are computed as in plain CA:

$$\mathbf{F} = \mathbf{D}_\mathbf{r}^{-1}\mathbf{U}\boldsymbol{\Delta} = \mathbf{D}_\mathbf{r}^{-1}\mathbf{X}\mathbf{D}_\mathbf{c}^{-1}\mathbf{V} \quad \text{and} \quad \mathbf{G} = \mathbf{D}_\mathbf{c}^{-1}\mathbf{V}\boldsymbol{\Delta} = \mathbf{D}_\mathbf{c}^{-1}\mathbf{X}^\top\mathbf{D}_\mathbf{r}^{-1}\mathbf{U}. \tag{17}$$

In MCA, the row factor score $f_{i,\ell}$ represents the ith observation on the ℓth component, and the column factor score $g_{j,k,\ell}$ represents the jth level of the kth variable on the ℓth component.

To sparsify MCA, we sparsify the left and right generalized singular vectors (i.e., $\mathbf{U}$ and $\mathbf{V}$) of MCA and derive *generalized sparsified pseudo-singular vectors* $\dot{\mathbf{U}}$ and $\dot{\mathbf{V}}$ where the sparse loadings are shrunken to 0. The sparsification of MCA is a specific case of the gsGSVD (Algorithm 1), similar to sCA, with the row metric matrix $\mathbf{D}_\mathbf{r}^{-1}$, the column metric matrix $\mathbf{D}_\mathbf{c}^{-1}$, and a (*non-optional*) group constraint $\mathcal{G}_\mathbf{q}$ on the columns. This group constraint ensures that all the levels of a variable are either discarded or kept. Such group sparsification can be optionally imposed on the rows (i.e., the observations) by specifying a row group matrix.

In the gsGSVD algorithm used for sMCA, the POCS procedure projects the data onto the intersection of three constrained spaces: 1) the $\mathcal{L}_2$-ball that normalizes the singular vectors, 2) the orthogonal space (i.e., $\dot{\mathbf{P}}_\ell^\perp$ or $\dot{\mathbf{Q}}_\ell^\perp$) that ensures orthogonality, and 3) the $\mathcal{L}_\mathcal{G}$-ball that sparsifies the elements of the right, and sometimes also the left, singular vectors (i.e., $\mathbf{p}$ and $\mathbf{q}$) and generalized singular vectors (i.e., $\mathbf{u}$ and $\mathbf{v}$) in groups.

3.2.4. Discriminant MCA (DiMCA) and sDiMCA: sparsification for discriminant analysis with group constraints for columns

The discriminant version of MCA (DiMCA) analyzes a matrix with variables coded as for MCA with rows nested in groups and extracts the components that best explain the inertia *between* these groups. To do so, DiMCA performs an MCA on the group sums computed across the rows of the concatenated table.

Formally, consider the same $I \times J$ data set $\mathbf{A}$ as used for MCA (see Equation (13)), if the observation groups are represented by an $I \times I_\mathcal{G}$ indicator matrix $\mathbf{H}$, the matrix of group sums across the observations (denoted by $\mathbf{A}_\mathcal{G}$) is computed as:

$$\mathbf{A}_\mathcal{G} = \mathbf{H}^\top\mathbf{A}, \tag{18}$$

where the rows of $\mathbf{A}_{\mathcal{G}}$ correspond to the $I_{\mathcal{G}}$ groups of observations and the columns of $\mathbf{A}_{\mathcal{G}}$ correspond to the levels of all K categorical variables (i.e., $J = \sum_{k=1}^{K} J_k$). Just like in MCA, the double centered probability matrix $\mathbf{X}_{\mathcal{G}}$ is obtained from the probability matrix $\mathbf{Z}_{\mathcal{G}}$ as (cf. Equations (14) and (15)):

$$\mathbf{Z}_{\mathcal{G}} = \frac{1}{IK}\mathbf{A}_{\mathcal{G}} \quad \text{and} \quad \mathbf{X}_{\mathcal{G}} = \mathbf{Z}_{\mathcal{G}} - \mathbf{r}\mathbf{c}^{\top}, \tag{19}$$

where $\mathbf{r}$ is a vector that stores the row sums (i.e., $\mathbf{r} = \mathbf{Z}_{\mathcal{G}}\mathbf{1}$) and $\mathbf{c}$ is a vector that stores the column sums (i.e., $\mathbf{c} = \mathbf{Z}_{\mathcal{G}}^{\top}\mathbf{1}$). $\mathbf{X}_{\mathcal{G}}$ is then analyzed by the GSVD:

$$\mathbf{X}_{\mathcal{G}} = \mathbf{U}\boldsymbol{\Delta}\mathbf{V}^{\top} \quad \text{subject to} \quad \mathbf{U}^{\top}\mathbf{D}_{\mathbf{r},\mathcal{G}}^{-1}\mathbf{U} = \mathbf{V}^{\top}\mathbf{D}_{\mathbf{c}}^{-1}\mathbf{V} = \mathbf{I}, \tag{20}$$

where $\mathbf{D}_{\mathbf{r},\mathcal{G}} = \mathsf{diag}(\mathbf{r})$, $\mathbf{D}_{\mathbf{c}} = \mathsf{diag}(\mathbf{c})$, and $\boldsymbol{\Delta}$ is the diagonal matrix of the generalized eigenvalues. In addition, just like in CA and MCA, this GSVD is equivalent to the SVD of Equation (B.5) and solves the same maximization problem (cf. Equation (B.4)) with $\mathbf{M} = \mathbf{D}_{\mathbf{r},\mathcal{G}}^{-1}$ and $\mathbf{W} = \mathbf{D}_{\mathbf{c}}^{-1}$. As a result, this maximization can also be solved by Algorithm E.2. Finally, the row and column factors are computed as for CA and MCA:

$$\mathbf{F} = \mathbf{D}_{\mathbf{r},\mathcal{G}}^{-1}\mathbf{U}\boldsymbol{\Delta} = \mathbf{D}_{\mathbf{r},\mathcal{G}}^{-1}\mathbf{X}\mathbf{D}_{\mathbf{c}}^{-1}\mathbf{V} \quad \text{and} \quad \mathbf{G} = \mathbf{D}_{\mathbf{c}}^{-1}\mathbf{V}\boldsymbol{\Delta} = \mathbf{D}_{\mathbf{c}}^{-1}\mathbf{X}^{\top}\mathbf{D}_{\mathbf{r},\mathcal{G}}^{-1}\mathbf{U}. \tag{21}$$

In DiMCA, the row factor scores $\mathbf{f}_{i,\ell}$ represent the $i_{\mathcal{G}}$th observation group on the ℓth component, and the column factor scores $\mathbf{g}_{j_k,\ell}$ represent the jth level of the kth variable on the ℓth component.

Finally, the original observations can be projected onto the component space as supplementary elements. Formally, a row profile denoted $\mathbf{i}$ (e.g., the ith row of $\mathbf{A}$) is projected onto the components to obtain the vector of its factor scores denoted $\mathbf{f}_i^*$ and computed as

$$\mathbf{f}_i^* = \boldsymbol{\Delta}^{-1}\mathbf{G}^{\top}\mathbf{i}. \tag{22}$$

To sparsify DiMCA, we sparsify the left and right generalized singular vectors (i.e., $\mathbf{U}$ and $\mathbf{V}$) of DiMCA and derive *generalized pseudo-singular vectors* $\dot{\mathbf{U}}$ and $\dot{\mathbf{V}}$ where the sparse loadings are shrunken to 0. Just like the other sparse methods, sDiMCA is a specific case of the gsGSVD (Algorithm 1) with row metric matrix $\mathbf{D}_{\mathbf{r},\mathcal{G}}^{-1}$, column metric matrix $\mathbf{D}_{\mathbf{c}}^{-1}$, and the group constraint $\mathcal{G}_{\dot{\mathbf{q}}}$ (on the columns). Here, the POCS procedure works the same way as in sMCA.

3.3. Sparsification: lost and found properties

Because of its specific preprocessing steps and metric constraints, CA (and therefore MCA, DiSCA, and DiMCA) possesses six essential properties: transition formulas, supplementary projections, asymmetric projection, distributional equivalence, barycentric projection, and embedded solutions (see Lebart et al., 1984, and Appendix C for details). Because sparsification introduces non-differentiable constraints into the optimization problem, the projection operators used for obtaining sparse solutions are non-linear. This non-linearity leads to a situation where the properties—which predominantly depend on linearity in standard CA-related methods—are either partially retained with minor or major modifications, or even completely lost. In this section, we identify these lost properties and evaluate possible solutions to restore them.

Property 3.1. *Transition formulas: from rows to columns and back.*

The transition formulas allow row factor scores ($\mathbf{f}_{\ell}$) to be computed from the column factor scores ($\mathbf{g}_{\ell}$), and vice versa (see, Escofier, 1979, the original transition formulas are also described in Property C.1). However, the original transition formulas no longer work with the gsGSVD because these formulas are linear projections whereas the projecting operator in Algorithm 1 is not linear. Using this non-linear projecting operator, we developed new transition formulas (with slight modifications from the original ones)

$$\begin{aligned}
\dot{\mathbf{f}}_{\ell} &= \mathbf{D}_{\mathbf{r}}^{-1}\dot{\mathbf{u}}_{\ell}\dot{\delta}_{\ell} \\
&= \mathbf{D}_{\mathbf{r}}^{-1}\mathbf{D}_{\mathbf{r}}^{\frac{1}{2}}\,\mathrm{proj}_{\dot{\mathcal{L}}_{\mathcal{G}_{\dot{\mathbf{u}}}}\cap\mathcal{L}_2\cap\dot{\mathbf{P}}}^{\perp}\left(\mathbf{D}_{\mathbf{r}}^{-\frac{1}{2}}\mathbf{X}\mathbf{D}_{\mathbf{c}}^{-1}\dot{\mathbf{v}}_{\ell}\right)\dot{\delta}_{\ell} \\
&= \mathbf{D}_{\mathbf{r}}^{-\frac{1}{2}}\,\mathrm{proj}_{\dot{\mathcal{L}}_{\mathcal{G}_{\dot{\mathbf{u}}}}\cap\mathcal{L}_2\cap\dot{\mathbf{P}}}^{\perp}\left(\mathbf{D}_{\mathbf{r}}^{-\frac{1}{2}}\mathbf{X}\mathbf{D}_{\mathbf{c}}^{-1}\dot{\mathbf{v}}_{\ell}\right)\dot{\delta}_{\ell} \\
&= \mathbf{D}_{\mathbf{r}}^{-\frac{1}{2}}\,\mathrm{proj}_{\dot{\mathcal{L}}_{\mathcal{G}_{\dot{\mathbf{u}}}}\cap\mathcal{L}_2\cap\dot{\mathbf{P}}}^{\perp}\left(\mathbf{D}_{\mathbf{r}}^{-\frac{1}{2}}\mathbf{X}\dot{\mathbf{g}}_{\ell}\dot{\delta}_{\ell}^{-1}\right)\dot{\delta}_{\ell},
\end{aligned} \tag{23}$$

$$
\begin{aligned}
\dot{\mathbf{g}}_\ell &= \mathbf{D}_\mathbf{c}^{-1}\dot{\mathbf{v}}_\ell\dot{\delta}_\ell \\
&= \mathbf{D}_\mathbf{c}^{-1}\mathbf{D}_\mathbf{c}^{\frac{1}{2}}\,\mathrm{proj}_{\dot{\mathcal{L}}_{\mathcal{G}_\mathbf{v}}\cap\mathcal{L}_2\cap\dot{\mathbf{Q}}^{\perp}}\left(\mathbf{D}_\mathbf{c}^{-\frac{1}{2}}\mathbf{X}^\top\mathbf{D}_\mathbf{r}^{-1}\dot{\mathbf{u}}_\ell\right)\dot{\delta}_\ell \\
&= \mathbf{D}_\mathbf{c}^{-\frac{1}{2}}\,\mathrm{proj}_{\dot{\mathcal{L}}_{\mathcal{G}_\mathbf{v}}\cap\mathcal{L}_2\cap\dot{\mathbf{Q}}^{\perp}}\left(\mathbf{D}_\mathbf{c}^{-\frac{1}{2}}\mathbf{X}^\top\mathbf{D}_\mathbf{r}^{-1}\dot{\mathbf{u}}_\ell\right)\dot{\delta}_\ell \\
&= \mathbf{D}_\mathbf{c}^{-\frac{1}{2}}\,\mathrm{proj}_{\dot{\mathcal{L}}_{\mathcal{G}_\mathbf{v}}\cap\mathcal{L}_2\cap\dot{\mathbf{Q}}^{\perp}}\left(\mathbf{D}_\mathbf{c}^{-\frac{1}{2}}\mathbf{X}^\top\dot{\mathbf{f}}_\ell\dot{\delta}_\ell^{-1}\right)\dot{\delta}_\ell.
\end{aligned}
\tag{24}
$$

When $\ell = 1$, $\dot{\mathbf{P}}^{\perp} = \dot{\mathbf{Q}}^{\perp} = \mathbf{0}$, and when $\ell > 1$, $\dot{\mathbf{P}}^{\perp} = \left[\dot{\mathbf{p}}_1 \mid \cdots \mid \dot{\mathbf{p}}_{\ell-1}\right]$ and $\dot{\mathbf{Q}}^{\perp} = \left[\dot{\mathbf{q}}_1 \mid \cdots \mid \dot{\mathbf{q}}_{\ell-1}\right]$.

These transition formulas for the gsGSVD are valid when (1) all the constraints are satisfied, and (2) the columns of the matrices of the estimated pseudo-singular vectors $\dot{\mathbf{P}}$ and $\dot{\mathbf{Q}}$ are in the order in which they were estimated by the ALS algorithm. Specifically, in our algorithm, the number of dimensions is specified, and the dimensions are reordered after all requested ones are estimated to give the pseudo-singular values in descending order. This reordering step is necessary because there is no guarantee that Algorithm 1 estimates the pseudo-singular values in a descending order. Remark (2) is especially important, because this order influences the definition of $\dot{\mathbf{P}}^{\perp}$ and $\dot{\mathbf{Q}}^{\perp}$ for a given value of ℓ. As a consequence, the number of dimensions to keep is one of the hyperparameters that should be evaluated and optimized (see Section 3.4 for details).

Property 3.2. *Supplementary projections.*

In CA/MCA/DiSCA/DiMCA, rows and columns can be projected as *supplementary elements* using equations from Property C.2. Of note, even though the equations for supplementary projection are directly derived from the transition formulas, in the sparsified versions of CA-related methods, the standard formulas for the projection of supplementary elements no longer work for dimensions extracted after the first dimension. The new transition formulas are defined in Equations (23) and (24). Note that when applied to a subset of existing rows or columns of $\mathbf{X}$, these transition formulas are not guaranteed to recover the correct sparse factor scores except for the first dimension. In fact, when applied to new data, the transition formulas could give incoherent results: Specifically, new data numerically close to existing data could be projected far away from each other. This problem occurs because of the iterative nature of POCS coupled with the orthogonality constraint.

To mitigate this issue, we propose to use a pseudo-inverse approach to project row or column profiles in certain cases. Specifically, with $\mathbf{R}$ and $\mathbf{C}$ denoting (respectively) the matrix of the row or column profiles, we define two projector operators denoted (respectively) $\mathbf{\Omega}_\mathbf{c}$ and $\mathbf{\Omega}_\mathbf{r}$ as:

$$
\mathbf{\Omega}_\mathbf{c} = \mathbf{R}^{+}\mathbf{F} \text{ and } \mathbf{\Omega}_\mathbf{r} = \mathbf{C}^{+}\mathbf{G} \quad \text{where } \mathbf{R}^{+} \text{ and } \mathbf{C}^{+} \text{ are the pseudo-inverse of (respectively) } \mathbf{R} \text{ and } \mathbf{C}. \tag{25}
$$

These projectors are then used to predict the factor scores from the profiles. Specifically, with $\mathbf{i}$ and $\mathbf{j}$ being (respectively) a row and column profile, their projections (denoted respectively $\mathbf{f}_{\text{sup}}$ and $\mathbf{g}_{\text{sup}}$) are obtained as

$$
\mathbf{f}_{\text{sup}} = \mathbf{i}\mathbf{\Omega}_\mathbf{c} \text{ and } \mathbf{g}_{\text{sup}} = \mathbf{j}\mathbf{\Omega}_\mathbf{r}\ . \tag{26}
$$

However, when there is no linear transformation from the data in $\mathbf{X}$ onto the sparse component space, these projectors provide only a (least squares optimal) approximation.

A linear projector approximation is informative when the sparsity parameters are near their maximum values—a pattern indicating low sparsity. For scenarios with high sparsity, we recommend using non-linear projectors without the orthogonality constraint. The intermediate sparsity cases are still a work in progress.

Property 3.3. *The asymmetric projection.*

The asymmetric projections of rows or columns give factor scores with unitary inertia (Property C.6). With sparsification, the asymmetric projections can be derived from the new transition formulas and are computed differently than the standard procedure described in Equations (C.11) and (C.13). The asymmetric projection are denoted by the subscript "asym" (e.g., the asymmetric projections of the row factor scores $\mathbf{F}$ is denoted $\mathbf{F}_{\text{asym}}$).

With sparsification, $\dot{\mathbf{F}}_{\text{asym}}$ (the matrix of asymmetric projections of the rows) is computed as:

$$
\dot{\mathbf{F}}_{\text{asym}} = \mathbf{D}_\mathbf{r}^{-1}\dot{\mathbf{U}} \quad \text{with inertia equal to} \quad \dot{\mathbf{F}}_{\text{asym}}^{\top}\mathbf{D}_\mathbf{r}\dot{\mathbf{F}}_{\text{asym}} = \mathbf{I}. \tag{27}
$$

Similarly, $\dot{\mathbf{G}}_{\text{asym}}$ (the matrix of asymmetric projections of the columns) is computed as:

$$
\dot{\mathbf{G}}_{\text{asym}} = \mathbf{D}_\mathbf{c}^{-1}\dot{\mathbf{V}} \quad \text{with inertia equal to} \quad \dot{\mathbf{G}}_{\text{asym}}^{\top}\mathbf{D}_\mathbf{c}\dot{\mathbf{G}}_{\text{asym}} = \mathbf{I}. \tag{28}
$$

These asymmetric projections of CA/MCA/DiSCA/DiMCA are preserved in the gsGSVD with sparsification.

Property 3.4. *Distributional equivalence: Rows (or columns) proportional to each other can be replaced by their sum without affecting the results of the analysis.*

Because these methods analyze frequencies of occurrences, two rows (or two columns) proportional to each other have identical profiles (e.g., see Escofier, 1969, for details.). Therefore, these rows (or columns) are represented by *two* points having the same coordinates in the component space, and so these two points can be merged into *one* whose mass is equal to the sum of the original masses (Escofier, 1969; Fichet, 2009; Benzécri et al., 1973; Greenacre, 1984). In addition, merging these two points does not change the geometry of the component space. This property is kept in gsGSVD with sparsification.

Property 3.5. *Barycentric projection: The barycenters of the row factor scores, and the column factor scores are equal to the null vector (Escofier, 1969).*

The row and the column factor scores of CA/MCA/DiSCA/DiMCA share a common barycenter equal to 0. Formally

$$\frac{1}{I}\mathbf{r}^\top \mathbf{f}_\ell = \frac{1}{J}\mathbf{c}^\top \mathbf{g}_\ell = 0. \tag{29}$$

Specifically, in MCA and DiMCA the barycenter of each variable (i.e., set of columns) is null:

$$\sum_{j=1}^{J_k} c_{j,k} g_{j,k,\ell} = 0, \tag{30}$$

where $c_{j,k}$ is the column weight for the jth level of the kth variable, and $g_{j,k,\ell}$ is the factor score of the jth level of the kth variable on the ℓth component. In addition, all observations that belong to a given variable level, have their mean row factor scores equal to the column factor score of this variable level. This property is lost in sCA and sDiSCA because the $\text{proj}_{\mathcal{L}_1}$ operator sparsifies the variable levels individually on rows and/or columns. However, this property is kept in sMCA and sDiMCA, because these methods use internally the $\text{proj}_{\mathcal{L}_\mathcal{G}}$ operator to keep or sparsify the levels of each variable in groups—a procedure that conserves their barycenters. Simulation results are illustrated in Fig. E.6.

Property 3.6. *Embedded solution: The GSVD of the non-centered matrix (i.e., $\mathbf{Z}$) has a first singular value of 1, a first left generalized singular vector equal to $\mathbf{r}$, and a first right generalized singular vector equal to $\mathbf{c}$. In addition, the subsequent components are the same as those of the GSVD of the double centered matrix $\mathbf{X}$.*

The embedded solution holds because the GSVD in CA (Equation (4)) can be rewritten as:

$$\begin{aligned} &\mathbf{X} = \mathbf{Z} - \mathbf{r}\mathbf{c}^\top = \mathbf{U}\boldsymbol{\Delta}\mathbf{V}^\top = \sum_{\ell=1}^{L} \delta_\ell \mathbf{u}_\ell \mathbf{v}_\ell^\top \\ &\text{under the constraints} \quad \mathbf{U}^\top \mathbf{D}_\mathbf{r}^{-1}\mathbf{U} = \mathbf{V}^\top \mathbf{D}_\mathbf{c}^{-1}\mathbf{V} = \mathbf{I}, \end{aligned} \tag{31}$$

which gives

$$\begin{aligned} &\mathbf{Z} = \mathbf{r}\mathbf{c}^\top + \mathbf{U}\boldsymbol{\Delta}\mathbf{V}^\top = 1 \times \mathbf{r}\mathbf{c}^\top + \sum_{\ell=2}^{L} \delta_\ell \mathbf{u}_\ell \mathbf{v}_\ell^\top \\ &\text{under the constraints} \quad \mathbf{U}^\top \mathbf{D}_\mathbf{r}^{-1}\mathbf{U} = \mathbf{V}^\top \mathbf{D}_\mathbf{c}^{-1}\mathbf{V} = \mathbf{I}. \end{aligned} \tag{32}$$

Therefore, when the non-centered data $\mathbf{Z}$ is analyzed, the first generalized singular value δ_1 equals 1, the first left generalized singular vector $\mathbf{u}_1$ equals $\mathbf{r}$, and the first right generalized singular vector $\mathbf{v}_1$ equals $\mathbf{c}$. With $\mathbf{r}\mathbf{c}^\top$ computing the *expected* probabilities of $\mathbf{Z}$ under independence, the CA of $\mathbf{X}$, where $\mathbf{X} = \mathbf{Z} - \mathbf{r}\mathbf{c}^\top$, analyzes the deviation of the *observed* data (i.e., $\mathbf{Z}$) from independence (i.e., as given by matrix $\mathbf{r}\mathbf{c}^\top$).

When the singular vectors are sparsified, the embedded solution of the CA/MCA/DiSCA/DiMCA (Property C.5) framework does not hold. Therefore, when the non-centered data $\mathbf{Z}$ is analyzed by sCA/sMCA/sDiSCA/sDiMCA, the first pseudo-singular-value $\dot{\delta}_1$ will be close to, but less than, 1, the first left (respectively right) generalized pseudo-singular-vector $\dot{\mathbf{u}}_1$ (respectively $\dot{\mathbf{v}}_1$) will be close to but not equal to $\mathbf{r}$ (respectively $\mathbf{c}$).

3.4. Evaluating the sparsification

The straightforward way to evaluate the sparsity of pseudo-generalized singular vectors is to express their number of zeros as a function of s (i.e., the sparsity parameter). Liu et al. (2023) noted that such a crude index of sparsity would not be useful for choosing an "optimal" value for the sparsity parameters. Recently, Trendafilov et al.'s (2017) proposed a new sparsity index that combines two measures: a measure of sparsity with a measure of how close the reduced rank sparse matrix is to the original data matrix.

In the following, L denotes the specific number of dimensions to estimate. To integrate these previous approaches, we measure sparsity with three different indices: 1) $\vartheta_{\dot{\mathbf{P}}}$, the ratio of the number of zeros (denoted $\#_0$) to the total number of coefficients in the

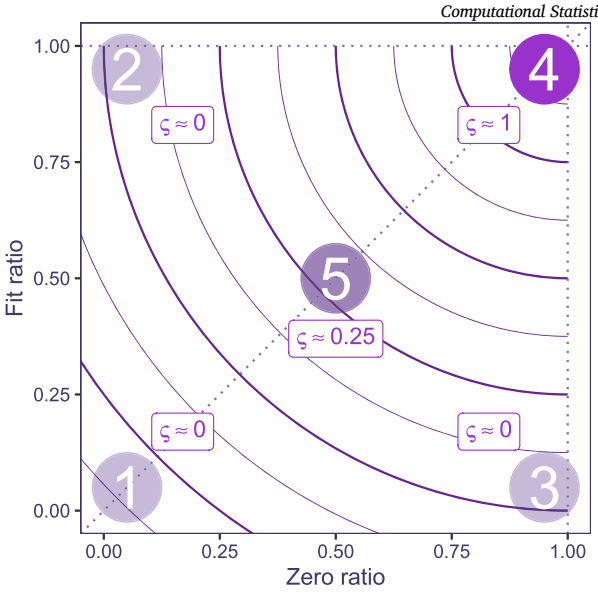

Fig. 1. Graph representing different possible values for the sparsity index on a map showing the "fit" as a function of the "zero ratio." The five zones represent five possible combinations of the two ratios, along with the corresponding values of the sparsity index ς.

loading matrix $\dot{\mathbf{P}}$ (equal to $I \times L$), and 2) its counterpart $\dot{\vartheta}_{\mathbf{Q}}$, the ratio of the number of zeros to the total number of coefficients in the loading matrix $\dot{\mathbf{Q}}$, and 3) ϑ, the ratio of the number of zeros in both $\dot{\mathbf{P}}$ and $\dot{\mathbf{Q}}$ to the total number of coefficients in $\dot{\mathbf{P}}$ and $\dot{\mathbf{Q}}$.

$$\begin{aligned}
\dot{\vartheta}_{\mathbf{P}} &= \frac{\#_0(\dot{\mathbf{P}})}{I \times L} \\
\dot{\vartheta}_{\mathbf{Q}} &= \frac{\#_0(\dot{\mathbf{Q}})}{J \times L} \\
\vartheta &= \frac{\#_0(\dot{\mathbf{P}}) + \#_0(\dot{\mathbf{Q}})}{(I+J) \times L}.
\end{aligned} \tag{33}$$

We measure the fit, denoted $\dot{\tau}$, as the ratio of the sum of the L squared pseudo-singular values to the sum of the first L squared singular values:

$$\dot{\tau} = \frac{\sum_{\ell=1}^{L} \dot{\delta}_\ell^2}{\sum_{\ell=1}^{L} \delta_\ell^2}. \tag{34}$$

Combining these two types of ratios (sparsity and fit), gives three different sparsity indices:

$$\begin{aligned}
\dot{\varsigma}_{\mathbf{P}} &= \dot{\vartheta}_{\mathbf{P}} \times \dot{\tau} \\
\dot{\varsigma}_{\mathbf{Q}} &= \dot{\vartheta}_{\mathbf{Q}} \times \dot{\tau} \\
\varsigma &= \vartheta \times \dot{\tau},
\end{aligned} \tag{35}$$

where $\dot{\varsigma}_{\mathbf{P}}$ is the compromise between fit and sparsity for the left generalized singular vectors, $\dot{\varsigma}_{\mathbf{Q}}$ is the compromise between fit and sparsity for the right generalized singular vectors, and ς is the compromise between fit and sparsity for both the left and the right generalized singular vectors.

Depending upon the application, the analyst can use one or more of these three sparsity indices to select the appropriate value of the sparsity parameters $s_{\mathbf{p}}$ and $s_{\mathbf{q}}$. We illustrate these indices in the result section.

Fig. 1 shows the range of possible values for the sparsity indices on a graph representing the ratio of zeros on the x-axis and the "fit" on the y-axis. In this figure, the result of a sparse analysis would be represented by a dot, according to how sparse the loadings are ("zero ratio") and how close the lower rank sparse decomposition of the data is to the original data ("fit"). We split the graph

into 5 zones: Zones 1 to 3 correspond to a low sparsity index because either or both "fit" and "zero ratio" are close to zero; Zone 4 corresponds to a sparsity index close to its maximum value of 1—a configuration that occurs when very few variables are selected that represent most of the information in the data—and, finally, Zone 5 corresponds to a middle ground, where a compromise is reached between sparsity and fit. In the following Results section, we provide such a map for each example. In these examples, the optimal value for the sparsity parameters and the number of dimensions are chosen by maximizing the sparsity index.

We recommend using both the sparsity index and the graph to choose an optimal set of sparsity parameters. Additionally, we propose (in Appendix H) an asymmetric version of the sparsity parameter—which is a modified version of Equation (35)—that is able to favor either fit or zero ratio.

4. Illustrations

4.1. sCA: death in the USA

4.1.1. Data

We applied sparse Correspondence Analysis to a contingency table storing the number of deaths in the USA in 2018 as a function of age and cause of death. The causes of death are categorized according to the International Statistical Classification of Diseases and Related Health Problems (World Health Organization, 2019). We categorized the age of death by groups of 5 years starting from the age of 1 and summing all deaths above 100 years old into a single group called "100+." We removed the deaths before the age of 1 because they mostly belong to the single and idiosyncratic category of perinatal causes. Therefore, including this age group would lead to this specific cause-age association dominating the results and masking other effects. The resulting data analyzed by CA and sCA were organized in a 21 (age groups) by 19 (causes of death) contingency table whose cells are storing the number of deaths of each cause at each age range.

4.1.2. Results

Fig. 2A shows the scree plot of the sCA results. Here, we only consider solutions with two or more components and found that sCA with 2 components gave the optimal result, which has the sparsest solution combined with the largest fit. These characteristics are shown in Fig. 2B, which plots the sparsity index for possible solutions with the chosen one being closest to the upper right corner where both fit and zero ratios are equal to 1. According to Fig. 2B, the sparsest solution with the largest fit gives an optimal sparsity index value of .473—a value that corresponds to optimal values for the sparsity parameters of $.51 \times \sqrt{21}$ (21 age groups) for the age groups and of $.31 \times \sqrt{19}$ (19 causes of death) for the causes of death.

In CA, the first component has an eigenvalue of .26 which explains 70% of the total inertia, and the second component has an eigenvalue of .08 which explains 21% of the total inertia. The row and the column factor scores show that the first component is characterized by deaths between the late teens and twenties that are related to childbirth and deaths at a young age ($<$ 35 years old) that are related to congenital and perinatal conditions (see the horizontal axis in Figs. 2C and E). The second component is characterized by deaths at older ages ($>$ 65 years old) which are mostly driven by malfunctions of the nervous system such as Alzheimer's and Parkinson's diseases (see the vertical axis in Figs. 2C and E). With sCA, the components give a clearer pattern that facilitates interpretation. The sparsified row and column factor scores show that the first component is characterized by deaths in early adulthood (between 20–35) that are related to childbirth and external causes (e.g., suicide). The second component differentiates deaths between 60–80 years old, which relate to the causes of neoplasms (e.g., cancer), from deaths at 90 years old and more, which relate to mental conditions (e.g., dementia).

4.2. sDiSCA: punctuation through the ages

4.2.1. Data

To illustrate sparse DiSCA, we used a data set that records the numbers of several punctuation marks in works of authors from different times and origins. The texts were extracted from the Gutenberg Project using the R-`gutenbergr` package (Robinson, 2021). In this data set, we included authors from France ($N = 63$), the United Kingdom ($N = 61$), and the United States ($N = 36$) and from different periods (see Table 1). The data counted the number of occurrences of the following nine punctuation marks: comma (,), period (.), question mark (?), exclamation mark (!), colon (:), semicolon (;) apostrophe ($'$), quotation marks for both English (" ") and French («»), hyphens (-), and M-dashes (—). We did not include the translated works and only considered the books written in the authors' original languages. We used DiSCA and sDiSCA to perform discriminant analysis on the eight author groups listed in Table 1.

4.2.2. Results

The sDiSCA results are shown in Fig. 3. Here, we consider a solution with 2 components because it gives the optimal results according to the sparsity index (Fig. 3B). According to Fig. 3B, the sparsest solution with the largest fit (i.e., closest to the upper right corner where both fit and zero ratios equal 1) gives an optimal sparsity index value of .335, which corresponds to the optimal sparsity parameter values of $.71 \times \sqrt{8}$ (8 author groups) for the individuals and of $.41 \times \sqrt{9}$ (9 punctuation marks) for the punctuation marks.

The original and the sparsified component spaces are shown in Figs. 3C to 3F. Fig. 3C shows the factor scores of the author groups from DiSCA, and Fig. 3D shows the same results from sDiSCA. The individual authors are projected as supplementary observations onto the same components and are used to illustrate the stability of these group means by estimating their 95% bootstrap confidence

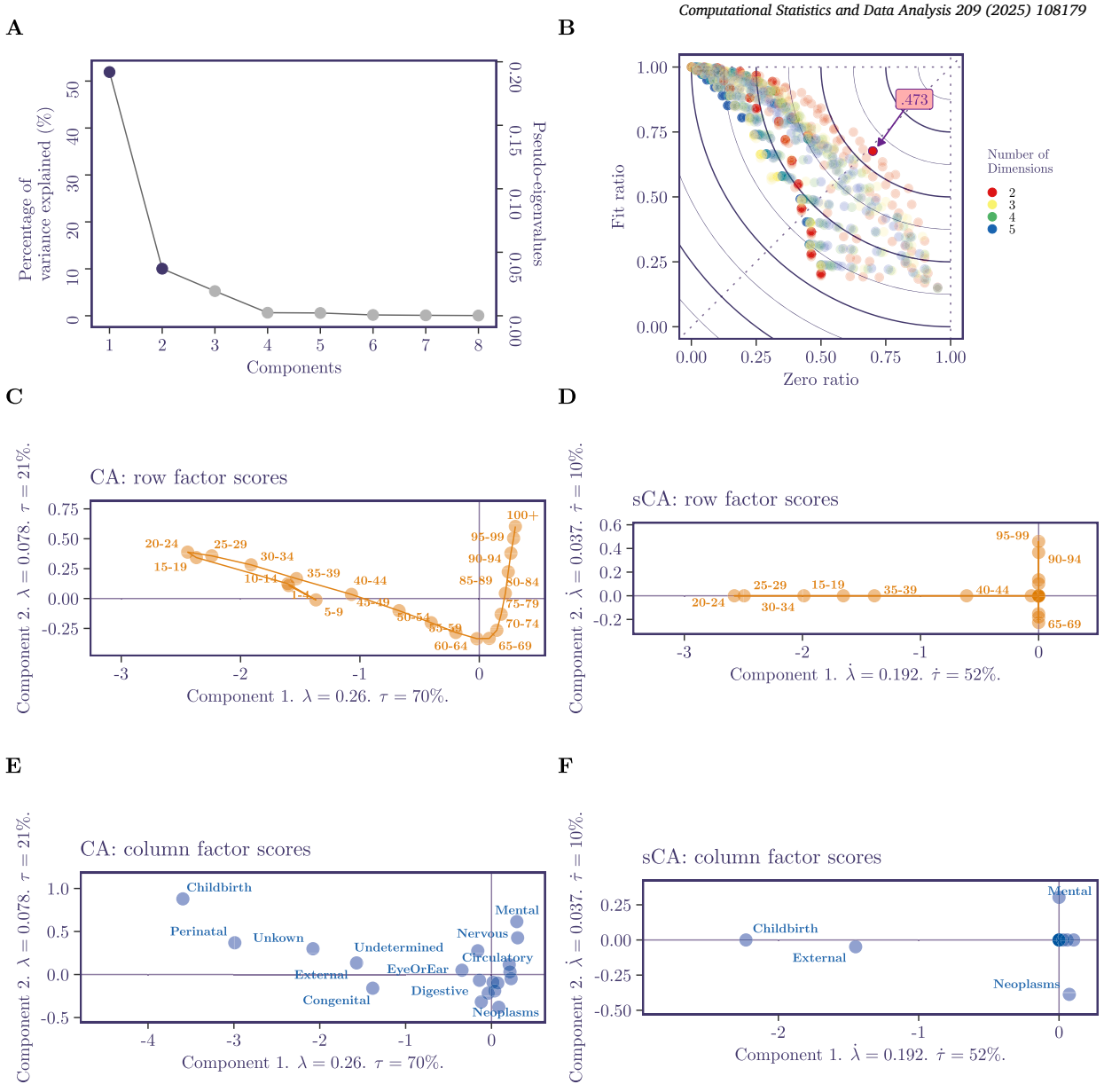

Fig. 2. Results of CA and sCA. (A) shows the scree plot of sCA with the optimal number of components colored in purple. (B) shows the fit-to-zero-ratio plot and highlights the optimal solution that has the maximum sparsity index. (C) and (D) show the row factor scores (which represent different age ranges) from CA and sCA. (E) and (F) show the column factor scores (which represent different causes of death) from CA and sCA.

Table 1
Numbers of authors in each group.

Origin	Time Period (century)		
	18th and before	19th	20th and after
France	16	28	19
UK	11	28	22
US	14		22

Note. UK: United Kingdom; US: United States.

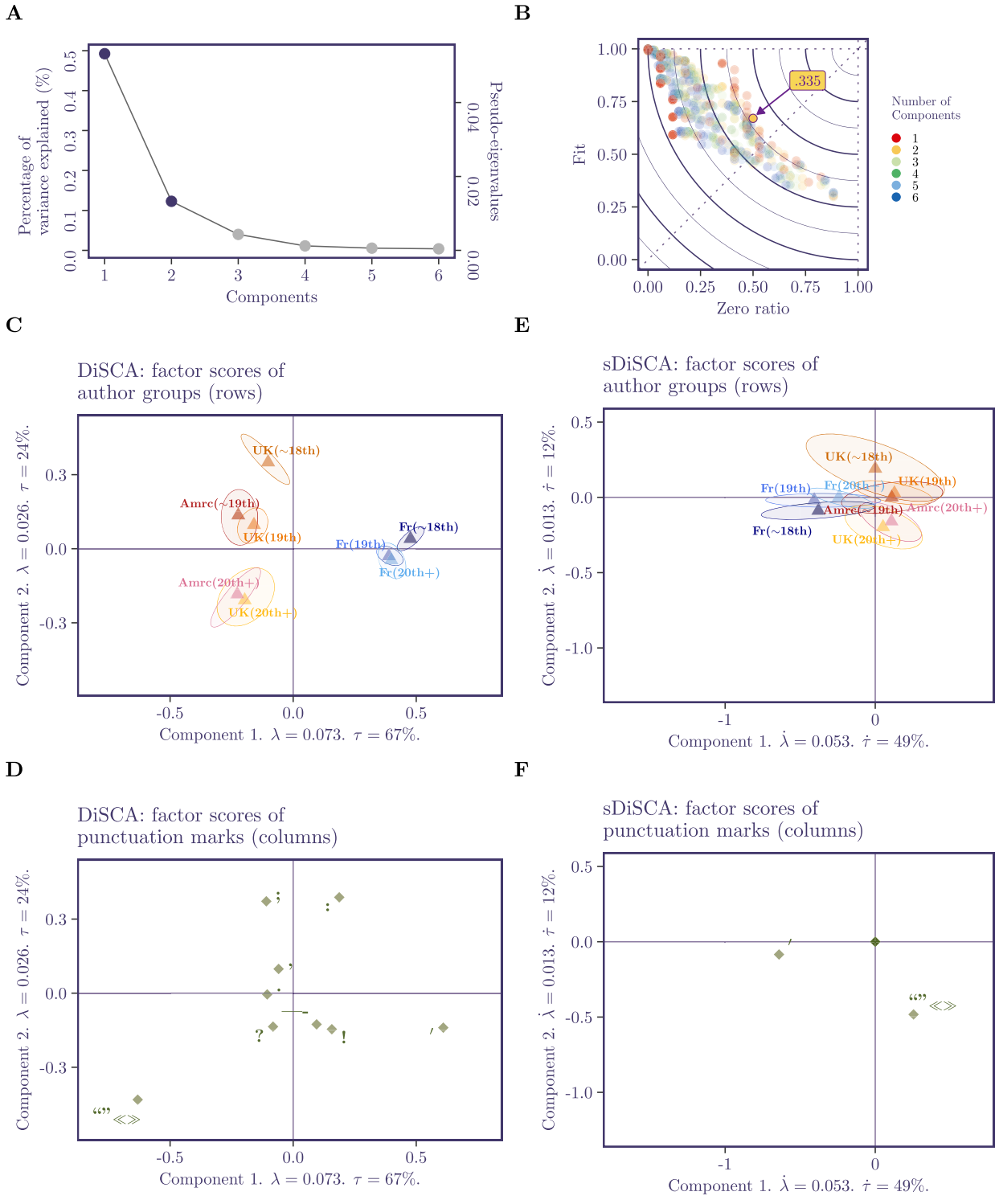

Fig. 3. Results of DiSCA and sDiSCA. (A) shows the scree plot of sDiSCA with the optimal number of components. (B) shows the fit-to-zero-ratio plot and highlighted the optimal solution with the maximum sparsity index. (C) and (D) show the row factor scores (which represent the groups of authors) from DiSCA and sDiSCA. (E) and (F) show the column factor scores (which represent the punctuation marks) from DiSCA and sDiSCA.

intervals. When two confidence intervals overlap, the difference between the two groups is considered non-significant (with $\alpha = .05$). In DiSCA, the first component distinguishes the French-speaking authors from the English-speaking authors, and the second component distinguishes the English-speaking authors from different periods of time. By contrast, the French authors are more consistent across time.

The results from sDiSCA show a similar pattern in Fig. 3D; the first component identifies only the three French groups of authors, and the second component identifies the oldest and the most recent groups of English authors. However, the bootstrap confidence intervals have quite a large variance—mostly because these intervals are derived from the supplementary projections of the authors with sparsification involved. While the sparsification algorithm sparsifies the number of contributing groups of each component, the group separations are, as a trade-off, no longer optimized, and thus results in supplementary projections are less segregated between groups.

The same trade-off is also reflected in the classification accuracy. In DiSCA, the discrimination between groups is measured by the accuracy of the correct classifications. To classify observations, their supplementary factor scores are compared to the group factor scores and classified as belonging to the closest group. The accuracy rate of DiSCA is .44 and the accuracy rate of sDiSCA is .16. Although lower than for DiSCA, the accuracy is still above chance level (.125). The accuracy rate is higher with respect to the origins (DiSCA: .74; sDiSCA: .47; chance level: .33) with the French authors being classified best (DiSCA: 1; sDiSCA: .68) as compared to the UK (DiSCA: .59; sDiSCA: .52) and the US (DiSCA: .56; sDiSCA: .27) authors.

Fig. 3E shows the factor scores of the punctuation marks from plain DiSCA, and Fig. 3F shows the same results from sDiSCA. The language effect on the first component is associated with the difference between how these authors used quotation marks and apostrophes. This language effect is therefore consistent with how apostrophes are used differently in English (to represent possession) and French (for grammatical purposes). The time effect on the second component is associated with how the authors used colons, semicolons, and commas (which connect sentences) versus how they used quotation marks, apostrophes, question, and exclamation marks (which end sentences). Therefore, the time effect could be related to how the style changes in English writing across time. Similar punctuation patterns are also shown in the results of sDiSCA, which identified the quotation marks and the apostrophes as contributing most to the group differences.

4.3. sMCA: self scale in Taiwan

4.3.1. Data

To illustrate sMCA, we used a data set using the Chinese version of the Independent and Interdependent Self Scale (C-IISS) developed by Lu and Gilmour (2007). This data set describes 130 undergraduate students (77 females and 53 Males; $M_{\text{age}} = 19.49$, and $sd_{\text{age}} = 1.52$) from National Cheng Kung University. These participants signed written informed consent and received NTD 120 at the end of the experiment. The C-IISS comprises forty-two items scored by the participants using a 7-point Likert scale (1 = strongly disagree; 7 = strongly agree). Among the 42 items, 21 of them measured independence (i.e., if one is aware of and values oneself as an individual) and the other 21 items measured interdependence (i.e., how one values oneself and acts based on one's cohort). Before we analyzed the data with sMCA, because the results of MCA (and sMCA) will be driven by the rarity of an event, we binned the responses of each item into categories of comparable sizes (see Supplementary Figure I1). The association patterns between these items are given in Supplementary Figure I2. We analyzed the binned data with both MCA and sMCA, where we only sparsified the items. Although sparsifying the individuals (or both) is also possible, we did not sparsify them because the separation of individuals was not of interest in this study.

4.3.2. Results

The sMCA results are shown in Fig. 4. For this data set, an sMCA with 9 components gives the optimal results according to the sparsity index (Fig. 4B). As shown in Fig. 4C, this result is also the sparsest solution with the largest fit (closest to the upper right corner where both fit and zero ratios equal 1). The sparsity parameter for the individuals was set to $\sqrt{130}$ (i.e., no sparsity; 130 individuals), and the optimal sparsity parameter for the items is $.29 \times \sqrt{42}$ (42 item levels). The sparsity index from this analysis equals .357.

The plain and sparsified component spaces are shown in Figs. 4D to 4I. In plain MCA—with Benzécri's (1979) eigenvalue correction, for details see also Beaton et al. (2014)—the first component eigenvalue of .08 explains 72% of the inertia, and the second component with an eigenvalue of .01 explains 9% of the inertia. Because it is unclear how such a correction should be applied to sMCA, we report here the uncorrected values directly from the GSVD for easier comparison between the two methods. Without Benzécri's correction, the first component from plain MCA with an eigenvalue of .30 explains 13.83% of the variance, and the second component with an eigenvalue of .12 explains 5.71% of the variance. The factor scores of the individuals are grouped according to the individual sex at birth, and the factor scores of the items are colored according to their corresponding category. The results showed that MCA generated components that could be difficult to explain given the complex pattern of loadings (see Fig. 4D), whereas the first two components from sMCA distinguished items from different categories (see Figs. 4F to 4H). Specifically, the first component of sMCA identifies linear level effects of 4 questions that measure the level of independence; the second component of sMCA identifies linear level effects of 4 questions that measure the level of interdependence. The first component from sMCA with an eigenvalue of .051 explains 2.36% of the variance, and the second component with an eigenvalue of .051 explains 2.34% of the variance. Figs. 4E and 4I show the factor scores of the individuals with the mean factor score of each sex group. To examine group effect post-hoc, the stability of these group means is illustrated by their 95% bootstrap confidence intervals; when two confidence intervals overlap, the

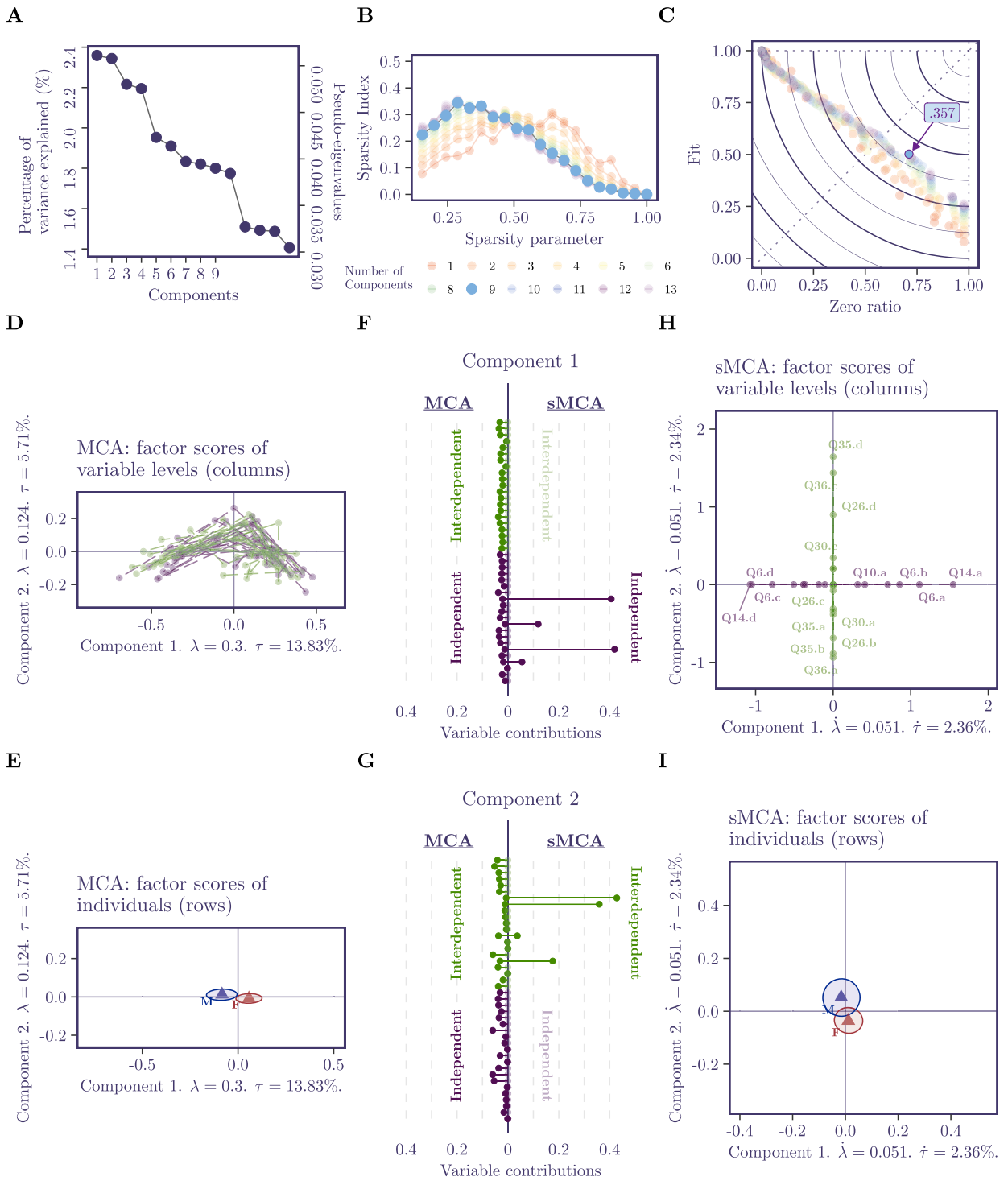

Fig. 4. Results from MCA and sMCA. (A) shows the scree plot of sMCA with the optimal number of components. (B) and (C) show the fit-to-zero-ratio plot and highlights the optimal solution with the maximum sparsity index. (D) and (H) show the column factor scores (which represent variable levels) from MCA and sMCA with the levels of the same variables connected from low to high to illustrate the trend. Items measuring interdependent self are colored in green, and items measuring independent self are colored in purple. (F) and (G) show the variables that contribute to the first two components of MCA and sMCA side by side. (E) and (I) show the mean row factor scores (which represent individuals) from MCA and sMCA with the blue triangle denoting the male group and the red triangle denoting the female group. The ellipses around the means illustrate the 95% bootstrap confidence intervals of the group means.

Table 2
Description of the groups for the Math questionnaire data.

Domain	Short Name	Number of Items
basic algebraic skills	Algebra	8
basic arithmetic skills	Arith	5
categorization and ranges	CatRang	3
decimals, fractions, and percentages	DecFracPerc	12
visual understanding	Visual	2

difference between the two groups is considered non-significant (at the α level of .05). In general, the individual factor scores from both analyses showed similar group separations with only marginal differences.

4.4. sDiMCA: MACS in stats

4.4.1. Data

The data we used to illustrate DiMCA and sDiMCA were collected from undergraduate students enrolled in an introductory psychology statistics course taught by various instructors, at an urban public college in the northeast of the United States. The course covered descriptive statistics, hypothesis testing, and an overview of advanced statistical procedures. Administered—for this illustration—to 460 participants across five semesters, the Math Assessment for College Students (MACS; Rabin et al., 2018) measures basic mathematics skills through a 30-item paper-and-pencil test, which covers five general content domains (see Table 2). Informed consent was obtained under an IRB-approved protocol and participants were not compensated. Demographic and academic performance data were also collected. The goal of the study was to examine the relationship between basic mathematics skills, demographic data, and academic performance.

Answering all for MACS questions for a participant took approximately 40 minutes and students completed the MACS during the first week of the semester. All 30 MACS items were graded with no partial credit, with each response recorded as 0 = incorrect and 1 = correct (scored by a single rater and re-scored by a second independent rater; Rabin et al., 2018). The course was computationally based, with students learning to perform statistical tests manually and use statistical software programs. Academic performance was evaluated based on the average score computed from three exams conducted in each semester. All exams included multiple-choice and problem-solving questions that covered basic statistics theory and applications. The average score was then categorized into a letter grade common to most undergraduate statistics courses, where: A = 90–100%; B = 80–89%; C = 70–79%; D = 60–69%; and F = below 60%. For this illustration, A and B grades were grouped together as were D and F grades, resulting in a total of 3 groups (i.e., AB, C, DF).

4.4.2. Results

DiMCA and sDiMCA results are shown in Fig. 5. For this data set, sDiMCA with 2 components gives the optimal non-unidimensional results according to the sparsity index (Fig. 5B). As shown in Fig. 5B, this result is also the sparsest solution with the largest fit (closest to the upper right corner where both fit and zero ratios equal 1). The sparsity parameter for the groups is set to equal $.79 \times \sqrt{5}$, and the optimal sparsity parameter for the items is $.57 \times \sqrt{60}$ (60 item levels). The sparsity index from this analysis equals .257.

As shown in Fig. 5, both DiMCA and sDiMCA differentiated between AB-level and DF-level students along the first dimension, and this difference was driven by the overall number of correct and incorrect responses on the MACS, where AB-level students had more correct responses and DF-level students had more incorrect responses. Specifically, DiMCA indicates that responses for all 30 MACS items reliably contribute to differences in performance—a pattern implying that DF-level students (compared to AB-level students) had an overall low performance across all basic domains of mathematics. In contrast, sDiMCA reveals that differences in performance on particular items related to basic algebraic skills and decimals, fractions, and percentages most discriminated between AB-level students and DF-level students. These topics are usually covered in elementary and middle school and may not have been reviewed prior to college-level statistics. Therefore, students could be encouraged to take statistics earlier (rather than later) in college when basic mathematics skills are more likely to still be remembered.

Based on MACS performance, certain students could be identified as “under-prepared” or “at risk” for poor outcomes in an introductory statistics course and might benefit from remediation before attempting courses that rely heavily on mathematics. Such remediation could be developed either as in-person or online formats with assignments completed independently or with a small learning group. Ideally, such efforts would provide opportunities for early success in introductory statistics courses through mastery of relevant prerequisite mathematics skills.

In addition, both DiMCA and sDiMCA differentiated the C-level students from other grade levels along the second dimension, but DiMCA, once again, showed that responses for all 30 MACS items reliably contributed to this difference. However, sDiMCA revealed that C-level students were able to correctly answer specific subsets of items in basic arithmetic (e.g., order of operations) and decimals, fractions, and percentages (e.g., adding fractions) that differentiated them from DF-level students, but incorrectly answered enough items (e.g., rounding to the nearest decimal, decimal and percentage conversions) to differentiate them from the AB-level students. These results suggest that while C-level students are slightly better than DF-level students on certain concepts, these C-level students might also benefit from remediation so that all students have the opportunity to begin the course with a similar level of mathematics proficiency.

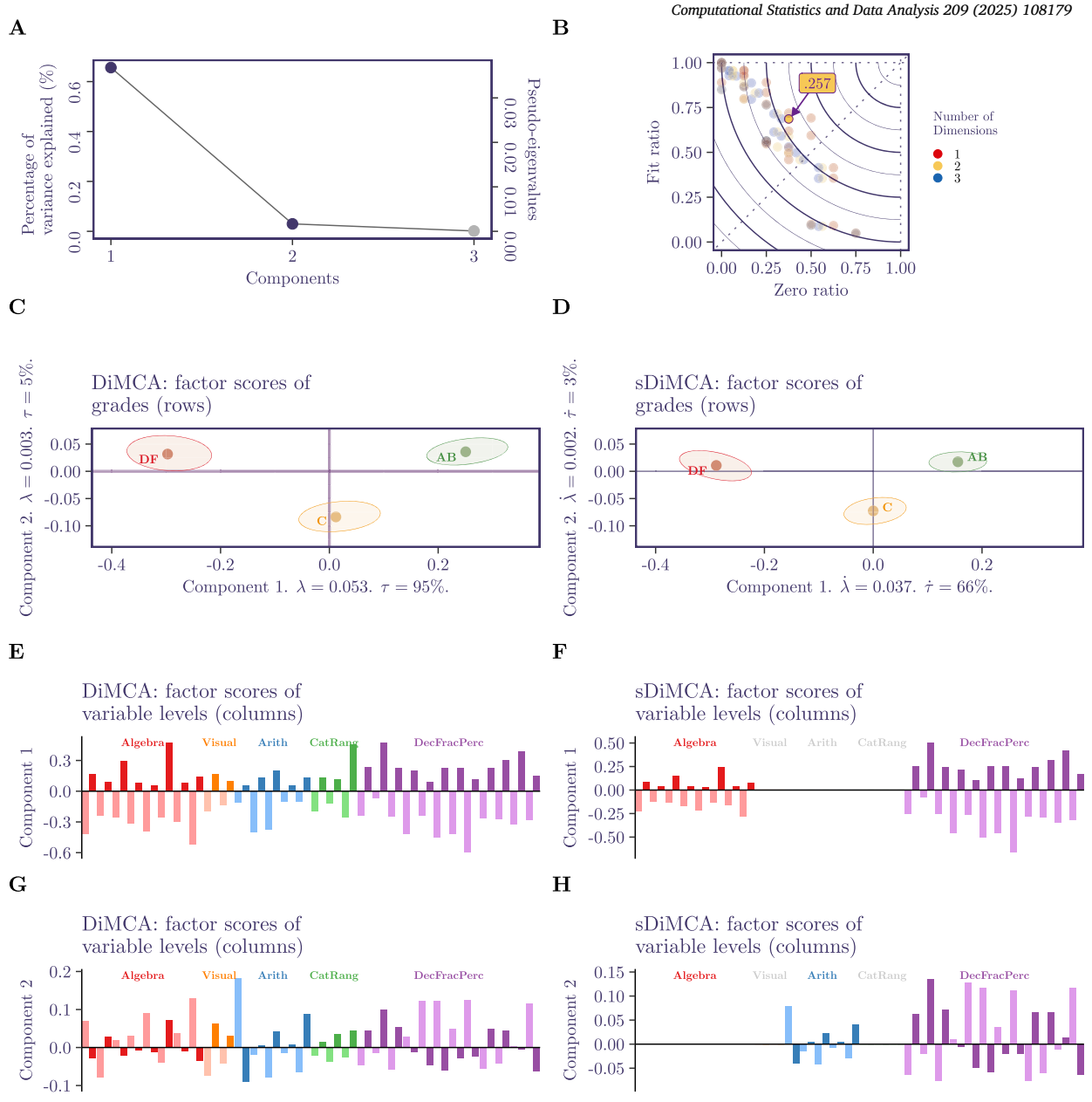

Fig. 5. Results from DiMCA and sDiMCA. (A) scree plot of DiMCA with components to keep (dark purple). (B) fit-to-zero-ratio plot with optimal solution highlighted. (C) and (D) row factor scores from (respectively) DiMCA and sDiMCA. Ellipsoids show bootstrap derived confidence intervals. (E) to (H) factor scores for the variables colored by category for DiMCA (on the left) and sDiMCA (on the right)—only non-null values are shown.

5. Conclusion and discussion

In this paper, we extended the sparse GSVD introduced by Yu et al. (2023) to create the group-sparse GSVD (gsGSVD) algorithm which keeps the orthogonality constraints of the GSVD while integrating sparsification with metric and group constraints. We applied the gsGSVD to sparsify CA, MCA, and their respective discriminant analysis versions: DiSCA and DiMCA. We illustrated these sparsification methods and compared their analytical merits on four real data sets. We also integrated group constraints into the sparsification problem—an essential property for categorical data analysis where variables are represented by blocks of columns. These group constraints could also be applied to different situations such as for data structured with *a priori* groups of variables.

This new gsGSVD algorithm seeks optimum solutions that satisfy strong constraints such as orthogonality and sparsity. Interestingly, our results also suggest that sparsification can be interpreted as a form of cluster analysis performed on the rows or the columns of the data matrix—an approach akin to, for example, spectral clustering which partitions the variables based on their singular vectors

(Kannan et al., 2000) or graph analysis (Spielman and Teng, 2011). Overall, we demonstrated how this new algorithm provides new multivariate tools to explore the complex structures of qualitative data.

To sparsify solutions while maintaining orthogonality between components, we used POCS to integrate two sets of strong constraints: one for orthogonality and one for sparsification. POCS is an iterative algorithm which loops over the projections (mentioned in Algorithm 1) to converge to the intersection between the spaces spanned by the constraints (for additional details, see Guillemot et al., 2019, Algorithm 3). However, in some cases, these constraints can be so strong that they may not be all satisfied together.

For example, to sparsify the solution while keeping the components orthogonal, we project the data onto the orthogonality constraints first and onto the sparsification constraints last. This approach guarantees that sparsity and orthogonality are satisfied to a high level of precision (i.e., smaller 10^{-10}). In practice, it is also possible to switch the order of projections and project onto the orthogonality constraints last. This order of projections prioritizes orthogonality over sparsity. It is worth noting that, sometimes, the solution of prioritizing different constraints could lead to different results. In certain circumstances, meeting both orthogonality and sparsity constraints is not feasible. These circumstances were rare but were often seen when the sparsify parameters were not adequately set given the data structure. This was the case, for example, when attempting to sparsify a contingency table too small to be sparsified (e.g. when measuring 6 observations on 3 variables). In such cases, the prioritized constraint will be satisfied at the expense of the other constraints.

It is worth noting that the dimensions derived from the CSVD and these proposed algorithms might not always be obtained in decreasing order of explained variance—a problem already present in Witten et al. (2009). The original order should be kept to implement the orthogonality constraints and for the transition formulas to work correctly. However, for convenience and to follow traditions, the current output of the R-implementation reorders the dimensions according to their variance.

Additionally—as is done by Witten et al. (2009) and Guillemot et al. (2019)—to make Algorithm 1 a convex optimization problem with convex constraints, the constraints on the (group-)$\mathcal{L}_1$- and $\mathcal{L}_2$-norms used are inequality constraints—an interpretation that could hinder the interpretability of the results. This could be the case, for example, when the $\mathcal{L}_2$-norm of some loadings is less than 1. In practice, however, these constraints are saturated, meaning that it is rarely the case that the $\mathcal{L}_2$-norms of the resulting pseudo generalized singular vectors are not exactly equal to 1, or that their $\mathcal{L}_1$-norm is not exactly equal to the given sparsity parameters. For cases when the constraints are unsaturated (i.e., they are not compatible), the solutions may not be unique and could be unstable.

The proposed sparse methods still require future developments to preserve more CA-MCA-related properties, to implement a comprehensive inference analysis, and to evaluate the stability and validity of the sparse solution. The current work also opens avenues to explore the properties of the POCS projecting operators such as the projection of supplementary elements, which currently uses (mostly linear) approximations.

Overall, this study presents an exciting foundation for refining and expanding the use of sparsification methods. Future avenues could extend the current methods to two data table analyses such as Partial Least Squares Correlation (PLSC, see, e.g., Krishnan et al., 2011) and its CA-like extension PLS-CA and PLS-MCA (Beaton et al., 2016). Future directions could also incorporate the consideration of hierarchical structures of variables or observations such as overlapping groups of grouped variables, exemplified in Single Nucleotide Polymorphism (SNP) data structured into pathways.

Acknowledgements

JCY receives funding from the Discovery Fund postdoctoral fellowship award of the Centre of Addiction and Mental Health.
LR and AK received funding from the PSC-CUNY TRADA Award (62058-00 50).
LR received funding from NSF Research Experiences for Undergraduates (REU) Award (2050755).

Appendix A. CA and inertia

Recall that, for an $I \times J$ contingency table, the independence χ^2 statistic is computed as:

$$\chi^2 = \sum_{i,j} \frac{(\text{Observed}_{i,j} - \text{Expected}_{i,j})^2}{\text{Expected}_{i,j}} \tag{A.1}$$

or, in matrix notation, the χ^2 is associated to the matrix $\mathbf{X}$ (with the notations from Equations (2) and (3)) and computed as:

$$\frac{1}{N}\chi^2 = \text{trace}\left(\mathbf{D_c}^{-\frac{1}{2}}\mathbf{X}^\top\mathbf{D_r}^{-1}\mathbf{X}\mathbf{D_c}^{-\frac{1}{2}}\right) = \text{trace}\left(\mathbf{D_r}^{-\frac{1}{2}}\mathbf{X}\mathbf{D_c}^{-1}\mathbf{X}^\top\mathbf{D_r}^{-\frac{1}{2}}\right). \tag{A.2}$$

From the GSVD, the inertia computed from the row factors $\mathbf{F}$ gives $\frac{1}{N}\chi^2$:

$$\begin{aligned}
&\text{trace}\left(\mathbf{F}^\top\mathbf{D_r}\mathbf{F}\right)\\
&= \text{trace}\left(\underbrace{\mathbf{V}^\top\mathbf{D_c}^{-1}\mathbf{X}^\top\mathbf{D_r}^{-1}}_{\text{Equation (5)}}\mathbf{D_r}\ \underbrace{\mathbf{D_r}^{-1}\mathbf{U}\boldsymbol{\Delta}}_{\text{Equation (5)}}\right)\\
&= \text{trace}\left(\mathbf{D_c}^{-1}\mathbf{X}^\top\mathbf{D_r}^{-1}\mathbf{U}\boldsymbol{\Delta}\mathbf{V}^\top\right)
\end{aligned} \tag{A.3}$$

$$
\begin{aligned}
&= \text{trace}\left(\mathbf{D_c}^{-1}\mathbf{X}^\top\mathbf{D_r}^{-1}\mathbf{X}\right)\\
&= \text{trace}\left(\mathbf{D_c}^{-\frac{1}{2}}\mathbf{X}^\top\mathbf{D_r}^{-1}\mathbf{X}\mathbf{D_c}^{-\frac{1}{2}}\right)\\
&= \frac{1}{N}\chi^2.
\end{aligned}
$$

Similarly, the inertia can be computed from the column factors $\mathbf{G}$:

$$
\text{trace}\left(\mathbf{G}^\top\mathbf{D_c}\mathbf{G}\right) = \text{trace}\left(\mathbf{D_r}^{-\frac{1}{2}}\mathbf{X}\mathbf{D_c}^{-1}\mathbf{X}^\top\mathbf{D_r}^{-\frac{1}{2}}\right) = \frac{1}{N}\chi^2. \tag{A.4}
$$

Appendix B. The SVD and the generalized SVD (GSVD)

The SVD decomposes a data matrix $\mathbf{X}$ into three matrices:

$$
\mathbf{X} = \mathbf{P}\boldsymbol{\Delta}\mathbf{Q}^\top \quad \text{such that} \quad \mathbf{P}^\top\mathbf{P} = \mathbf{Q}^\top\mathbf{Q} = \mathbf{I}, \tag{B.1}
$$

where $\mathbf{P}$ (respectively $\mathbf{Q}$) is the $I \times L$ (respectively $J \times L$) matrix of the left (respectively right) singular vectors, and $\boldsymbol{\Delta}$ is a diagonal matrix with the singular values (i.e., δ_ℓ) stored in its diagonal. The SVD solves the following maximization problem:

$$
\begin{aligned}
&\underset{\mathbf{p}_\ell,\mathbf{q}_\ell}{\arg\max}\left(\delta_\ell = \mathbf{p}_\ell^\top\mathbf{X}\mathbf{q}_\ell\right) \quad \text{subject to}\\
&\begin{cases} \mathbf{p}_\ell^\top\mathbf{p}_\ell &= 1,\\ \mathbf{q}_\ell^\top\mathbf{q}_\ell &= 1, \end{cases} \text{and, for any } \ell \neq \ell', \quad \begin{cases} \mathbf{p}_\ell^\top\mathbf{p}_{\ell'} &= 0.\\ \mathbf{q}_\ell^\top\mathbf{q}_{\ell'} &= 0. \end{cases}
\end{aligned} \tag{B.2}
$$

Here, δ_ℓ is the ℓth singular value and is associated to the ℓth left (respectively right) singular vector $\mathbf{p}_\ell$ and $\mathbf{q}_\ell$.

Similar to the SVD, the GSVD also decomposes $\mathbf{X}$ into three matrices but with row and column metric matrices included in the constraints:

$$
\mathbf{X} = \mathbf{U}\boldsymbol{\Delta}\mathbf{V}^\top \quad \text{such that} \quad \mathbf{U}^\top\mathbf{M}\mathbf{U} = \mathbf{V}^\top\mathbf{W}\mathbf{V} = \mathbf{I}, \tag{B.3}
$$

where $\mathbf{U}$ (respectively $\mathbf{V}$) is the matrix of the left (respectively right) generalized singular vectors, $\mathbf{M}$ (respectively $\mathbf{W}$) is the row (respectively column) metric matrix represented by a positive definite matrix. The GSVD solves the following maximization problem:

$$
\begin{aligned}
&\underset{\mathbf{p}_\ell,\mathbf{q}_\ell}{\arg\max}\left(\delta_\ell = \mathbf{u}_\ell^\top\mathbf{X}\mathbf{v}_\ell\right) \quad \text{subject to}\\
\text{subject to} \quad &\begin{cases} \mathbf{u}_\ell^\top\mathbf{M}\mathbf{u}_\ell &= 1,\\ \mathbf{v}_\ell^\top\mathbf{W}\mathbf{v}_\ell &= 1, \end{cases} \text{and, for any } \ell \neq \ell', \quad \begin{cases} \mathbf{u}_\ell^\top\mathbf{M}\mathbf{u}_{\ell'} &= 0.\\ \mathbf{v}_\ell^\top\mathbf{W}\mathbf{v}_{\ell'} &= 0. \end{cases}
\end{aligned} \tag{B.4}
$$

The maximization problem of the GSVD is equivalent to the following SVD of the weighted $\mathbf{X}$ (denoted by $\widetilde{\mathbf{X}}$), where

$$
\begin{aligned}
&\widetilde{\mathbf{X}} = \mathbf{M}^{\frac{1}{2}}\mathbf{X}\mathbf{W}^{\frac{1}{2}} = \mathbf{P}\boldsymbol{\Delta}\mathbf{Q}^\top\\
&\text{such that} \quad \mathbf{P}^\top\mathbf{P} = \mathbf{Q}^\top\mathbf{Q} = \mathbf{I},
\end{aligned} \tag{B.5}
$$

with

$$
\begin{cases} \mathbf{U} &= \mathbf{M}^{-\frac{1}{2}}\mathbf{P}\\ \mathbf{V} &= \mathbf{W}^{-\frac{1}{2}}\mathbf{Q} \end{cases} \quad \text{such that} \quad \mathbf{U}^\top\mathbf{M}\mathbf{U} = \mathbf{V}^\top\mathbf{W}\mathbf{V} = \mathbf{I}. \tag{B.6}
$$

Appendix C. Properties of CA, MCA, DiSCA, and DiMCA

Because of the specific preprocessing steps and the metric constraints, CA (and therefore MCA, DiSCA, and DiMCA) has several specific properties (Escofier, 1969; Greenacre, 1984; Abdi and Béra, 2014).

Property C.1. *Transition formulas: The row factor scores can be computed from the column factor scores and vice versa.*

CA/MCA/DiSCA/DiMCA analyze the rows and columns symmetrically; therefore, the row (respectively column) factors can be obtained from the data and the column (respectively row) factors by a transition formula. Transition formulas can be directly derived from Equation (5) that expresses the row factor scores $\mathbf{F}$ as

$$
\mathbf{F} = \mathbf{D_r}^{-1}\mathbf{X}\mathbf{D_c}^{-1}\mathbf{V} \quad \text{with} \quad \mathbf{D_c}^{-1}\mathbf{V} = \mathbf{G}\boldsymbol{\Delta}^{-1} \tag{C.1}
$$

and the column factor scores $\mathbf{G}$ as

$$\mathbf{G} = \mathbf{D}_\mathbf{c}^{-1}\mathbf{X}^\top\mathbf{D}_\mathbf{r}^{-1}\mathbf{U}, \quad \text{with} \quad \mathbf{D}_\mathbf{r}^{-1}\mathbf{U} = \mathbf{F}\boldsymbol{\Delta}^{-1}, \tag{C.2}$$

From these last two equations, by substitution, $\mathbf{F}$ can be computed from $\mathbf{G}$ by

$$\mathbf{F} = \mathbf{D}_\mathbf{r}^{-1}\mathbf{X}\mathbf{G}\boldsymbol{\Delta}^{-1} \tag{C.3}$$

and $\mathbf{G}$ can be computed from $\mathbf{F}$ by

$$\mathbf{G} = \mathbf{D}_\mathbf{c}^{-1}\mathbf{X}^\top\mathbf{F}\boldsymbol{\Delta}^{-1}. \tag{C.4}$$

Property C.2. *Supplementary projections: Equations (C.3) and (C.4) can be used to estimate the factor scores from a supplementary (also called out-of-sample) row (respectively column) that is represented by the same set of columns (respectively rows).*

The factor score (denoted $\mathbf{f}_{\text{sup}}$) of a supplementary row profile denoted $\mathbf{i}_{\text{sup}}$ is computed with an equation similar to Equation (C.3):

$$\mathbf{f}_{\text{sup}} = \mathbf{i}_{\text{sup}}^\top\mathbf{G}\boldsymbol{\Delta}^{-1}. \tag{C.5}$$

The factor score (denoted $\mathbf{g}_{\text{sup}}$ of a supplementary column denoted $\mathbf{j}_{\text{sup}}$ is computed by a similar equation to Equation (C.4):

$$\mathbf{g}_{\text{sup}} = \mathbf{j}_{\text{sup}}^\top\mathbf{F}\boldsymbol{\Delta}^{-1}. \tag{C.6}$$

Geometrically, these factor scores project the supplementary rows (or column) onto the component space built by the original data.

Property C.3. *Distributional equivalence: Identical rows (or columns) can be replaced by their sum without affecting the results.*

Because CA and MCA analyze the frequencies of the occurrences, two rows (or two columns) that are proportional to each other become identical after the preprocessing steps. These identical rows (or columns) can be represented by *two* coincident points in the component space, and the two points can be merged into *one* with the sum of the original weights (Fichet, 2009; Benzécri et al., 1973; Greenacre, 1984). In addition, merging these two points does not change the geometry of the component space.

Property C.4. *Barycentric projection: Row and column factor scores, have barycenters equal to 0.*

The row and the column factor scores of CA/MCA/DiSCA/DiMCA share a common barycenter (i.e., weighted mean) of 0; Formally

$$\frac{1}{I}\mathbf{r}^\top\mathbf{f}_\ell = \frac{1}{J}\mathbf{c}^\top\mathbf{g}_\ell = 0. \tag{C.7}$$

Specifically in MCA, for each of the K variables, the (column) factor scores of its levels will have a weighted mean of zero:

$$\sum_{j=1}^{J_k} c_{j,k}\, g_{j,k,\ell} = 0, \tag{C.8}$$

where $c_{j,k}$ is the column weight for the jth level of the kth variable, and $g_{j,k,\ell}$ is the factor scores of the jth level of the kth variable on the ℓth component. In addition, for all observations that belong to this variable level, their mean (row) factor score will equal the (column) factor score of this variable level.

Property C.5. *The embedded solution: in CA, the GSVD of the non-centered matrix (i.e., $\mathbf{Z}$) will have its first generalized singular value equal to 1, its first left generalized singular vector equal to $\mathbf{r}$, and its first right generalized singular vector equal to $\mathbf{c}$. In addition, the following components will be equivalent to the GSVD of the matrix $\mathbf{X}$.*

The embedded solution holds because the GSVD in CA (Equation (4)) can be rewritten as:

$$\begin{aligned} &\mathbf{X} = \mathbf{Z} - \mathbf{r}\mathbf{c}^\top = \mathbf{U}\boldsymbol{\Delta}\mathbf{V}^\top = \sum_{\ell=1}^{L}\delta_\ell\mathbf{u}_\ell\mathbf{v}_\ell^\top \\ &\text{under the constraints} \quad \mathbf{U}^\top\mathbf{D}_\mathbf{r}^{-1}\mathbf{U} = \mathbf{V}^\top\mathbf{D}_\mathbf{c}^{-1}\mathbf{V} = \mathbf{I} \end{aligned} \tag{C.9}$$

which gives

$$\begin{aligned} &\mathbf{Z} = \mathbf{r}\mathbf{c}^\top + \mathbf{U}\boldsymbol{\Delta}\mathbf{V}^\top = 1\times\mathbf{r}\mathbf{c}^\top + \sum_{\ell=2}^{L}\delta_\ell\mathbf{u}_\ell\mathbf{v}_\ell^\top \\ &\text{under the constraints} \quad \mathbf{U}^\top\mathbf{D}_\mathbf{r}^{-1}\mathbf{U} = \mathbf{V}^\top\mathbf{D}_\mathbf{c}^{-1}\mathbf{V} = \mathbf{I}. \end{aligned} \tag{C.10}$$

Therefore, when the non-centered data $\mathbf{Z}$ is analyzed, the first generalized singular value δ_1 equals 1, the first left generalized singular vector $\mathbf{u}_1$ equals $\mathbf{r}$, and the first right generalized singular vector $\mathbf{v}_1$ equals $\mathbf{c}$. With $\mathbf{rc}^\top$ computing the *expected* frequencies of $\mathbf{Z}$ under independence, the CA of $\mathbf{X}$, where $\mathbf{X} = \mathbf{Z} - \mathbf{rc}^\top$, analyzes the deviation of the *observed* data (i.e., $\mathbf{Z}$) from the independence (i.e., $\mathbf{rc}^\top$).

Property C.6. *The asymmetric projection: The row and column factor scores of each component can be scaled to have either a variance equal to 1 or a variance equal to the associated eigenvalue. When the row and the column factor scores are scaled differently (i.e., one set has a variance equal to 1 with the other set has a variance equal to the eigenvalue), these two sets are projected* asymmetrically.

In the CA/MCA/DiSCA framework, the rows and columns are analyzed symmetrically and the extracted components can be seen from the perspective of the rows or of the columns. From the perspective of the rows, the component space is defined by the asymmetric row factor scores (denoted by $\mathbf{F}_{\text{asym}}$) computed as:

$$\mathbf{F}_{\text{asym}} = \mathbf{D}_\mathbf{r}^{-1}\mathbf{U} \quad \text{with its variance} \quad \mathbf{F}_{\text{asym}}^\top \mathbf{D}_\mathbf{r} \mathbf{F}_{\text{asym}} = \mathbf{I}. \tag{C.11}$$

The columns can then be projected onto this component space as $\mathbf{G}$:

$$\mathbf{G} = \mathbf{D}_\mathbf{c}^{-1}\mathbf{V}\boldsymbol{\Delta} \quad \text{with its variance} \quad \mathbf{G}^\top \mathbf{D}_\mathbf{c} \mathbf{G} = \boldsymbol{\Lambda}. \tag{C.12}$$

Because the column factor scores are projected onto this space and scaled differently from the row factor scores, this projection is *asymmetric*.

From the perspective of the columns, the component space is defined by the column factor scores when the columns are projected asymmetrically (denoted by $\mathbf{G}_{\text{asym}}$) which are computed as:

$$\mathbf{G}_{\text{asym}} = \mathbf{D}_\mathbf{c}^{-1}\mathbf{V} \quad \text{with its variance} \quad \mathbf{G}_{\text{asym}}^\top \mathbf{D}_\mathbf{c} \mathbf{G}_{\text{asym}} = \mathbf{I}. \tag{C.13}$$

The asymmetric projection of the rows onto this space can then be computed as $\mathbf{F}$:

$$\mathbf{F} = \mathbf{D}_\mathbf{r}^{-1}\mathbf{U}\boldsymbol{\Delta} \quad \text{with its variance} \quad \mathbf{F}^\top \mathbf{D}_\mathbf{r} \mathbf{F} = \boldsymbol{\Lambda}. \tag{C.14}$$

According to Property C.5, because the first generalized singular value of the uncentered matrix equals 1, it is the maximum amount of variance any given component can have. When the factor scores of a component have a variance of 1, these factor scores span the entire available space—called a *simplex*—of the data. From Property C.4, the simplex defined by the row and the simplex defined by the column factors share the same *barycenter*. When one is used to define the simplex, the other can be projected asymmetrically onto this simplex where the distance between any two factor scores (including the distance between a row factor score and a column factor score) is meaningful. In contrast, when both the rows and the columns are symmetrically projected (as $\mathbf{F}$ and $\mathbf{G}$), only the distances within the same set are meaningful.

Appendix D. Transition formulas with and without sparsification

We derive the transition formulas for row and column factor scores, for regular and sparse CA-related methods. We provide a step-by-step breakdown of these derivations, highlighting specifically where the projection operators introduce non-linearity, leading to sparsity.

The original transition formulas are derived as follows for row ($\mathbf{f}$) and column ($\mathbf{g}$) factor scores:

$$\begin{aligned}
\mathbf{f}_\ell &= \mathbf{D}_\mathbf{r}^{-1}\mathbf{u}_\ell\delta_\ell \\
&= \mathbf{D}_\mathbf{r}^{-1}(\mathbf{U}\boldsymbol{\Delta}\mathbf{V}^\top)\mathbf{D}_\mathbf{c}^{-1}\mathbf{v}_\ell \\
&= \mathbf{D}_\mathbf{r}^{-1}\mathbf{X}\mathbf{D}_\mathbf{c}^{-1}\mathbf{v}_\ell \\
&= \mathbf{D}_\mathbf{r}^{-1}(\mathbf{Z} - \mathbf{rc}^\top)\mathbf{D}_\mathbf{c}^{-1}\mathbf{v}_\ell \\
&= \mathbf{D}_\mathbf{r}^{-1}\mathbf{Z}\mathbf{D}_\mathbf{c}^{-1}\mathbf{v}_\ell \\
&= \mathbf{R}\mathbf{D}_\mathbf{c}^{-1}\mathbf{v}_\ell \\
&= \mathbf{D}_\mathbf{r}^{-1}\mathbf{Z}\mathbf{g}_\ell\delta_\ell^{-1},
\end{aligned} \tag{D.1}$$

where $\mathbf{Z}$ is the contingency table matrix, $\mathbf{X}$ is the probability matrix, and $\mathbf{R}$ is the matrix of row profiles (i.e., $\mathbf{R} = \mathbf{D}_\mathbf{r}^{-1}\mathbf{Z}$):

$$\begin{aligned}
\mathbf{g}_\ell &= \mathbf{D}_\mathbf{c}^{-1}\mathbf{v}_\ell\delta_\ell \\
&= \mathbf{D}_\mathbf{c}^{-1}(\mathbf{V}\boldsymbol{\Delta}\mathbf{U}^\top)\mathbf{D}_\mathbf{r}^{-1}\mathbf{u}_\ell \\
&= \mathbf{D}_\mathbf{c}^{-1}\mathbf{X}^\top\mathbf{D}_\mathbf{r}^{-1}\mathbf{u}_\ell \\
&= \mathbf{D}_\mathbf{c}^{-1}(\mathbf{Z}-\mathbf{r}\mathbf{c}^\top)^\top\mathbf{D}_\mathbf{r}^{-1}\mathbf{u}_\ell \\
&= \mathbf{D}_\mathbf{c}^{-1}\mathbf{Z}^\top\mathbf{D}_\mathbf{r}^{-1}\mathbf{u}_\ell \\
&= \mathbf{C}\mathbf{D}_\mathbf{r}^{-1}\mathbf{u}_\ell \\
&= \mathbf{D}_\mathbf{c}^{-1}\mathbf{Z}^\top\mathbf{f}_\ell\delta_\ell^{-1},
\end{aligned} \tag{D.2}$$

where $\mathbf{C}$ is the matrix of column profiles (i.e., $\mathbf{C} = \mathbf{D}_\mathbf{c}^{-1}\mathbf{Z}^\top$):

With sparsification, the new transition formulas for CA/MCA/DiSCA require the projecting operators used in the gsGSVD algorithm:

$$\begin{aligned}
\dot{\mathbf{f}}_\ell &= \mathbf{D}_\mathbf{r}^{-\frac{1}{2}}\,\mathrm{proj}^{\cdot\perp}_{\mathcal{L}_{\mathcal{G},\dot{\mathbf{u}}}\cap\mathcal{L}_2\cap\dot{\mathbf{P}}}\left(\mathbf{D}_\mathbf{r}^{-\frac{1}{2}}\mathbf{X}\dot{\mathbf{g}}_\ell\dot{\delta}_\ell^{-1}\right)\dot{\delta}_\ell \\
&= \mathbf{D}_\mathbf{r}^{-\frac{1}{2}}\,\mathrm{proj}^{\cdot\perp}_{\mathcal{L}_{\mathcal{G},\dot{\mathbf{u}}}\cap\mathcal{L}_2\cap\dot{\mathbf{P}}}\left(\mathbf{D}_\mathbf{r}^{-\frac{1}{2}}(\mathbf{Z}-\mathbf{r}\mathbf{c}^\top)\dot{\mathbf{g}}_\ell\dot{\delta}_\ell^{-1}\right)\dot{\delta}_\ell \\
&= \mathbf{D}_\mathbf{r}^{-\frac{1}{2}}\,\mathrm{proj}^{\cdot\perp}_{\mathcal{L}_{\mathcal{G},\dot{\mathbf{u}}}\cap\mathcal{L}_2\cap\dot{\mathbf{P}}}\left(\mathbf{D}_\mathbf{r}^{-\frac{1}{2}}\mathbf{Z}\dot{\mathbf{g}}_\ell\dot{\delta}_\ell^{-1}\right)\dot{\delta}_\ell \\
&= \mathbf{D}_\mathbf{r}^{-\frac{1}{2}}\,\mathrm{proj}^{\cdot\perp}_{\mathcal{L}_{\mathcal{G},\dot{\mathbf{u}}}\cap\mathcal{L}_2\cap\dot{\mathbf{P}}}\left(\mathbf{D}_\mathbf{r}^{\frac{1}{2}}\mathbf{R}\dot{\mathbf{v}}_\ell\right)\dot{\delta}_\ell,
\end{aligned} \tag{D.3}$$

and

$$\begin{aligned}
\dot{\mathbf{g}}_\ell &= \mathbf{D}_\mathbf{c}^{-\frac{1}{2}}\,\mathrm{proj}^{\cdot\perp}_{\mathcal{L}_{\mathcal{G},\dot{\mathbf{v}}}\cap\mathcal{L}_2\cap\dot{\mathbf{Q}}}\left(\mathbf{D}_\mathbf{c}^{-\frac{1}{2}}\mathbf{X}^\top\dot{\mathbf{f}}_\ell\dot{\delta}_\ell^{-1}\right)\dot{\delta}_\ell \\
&= \mathbf{D}_\mathbf{c}^{-\frac{1}{2}}\,\mathrm{proj}^{\cdot\perp}_{\mathcal{L}_{\mathcal{G},\dot{\mathbf{v}}}\cap\mathcal{L}_2\cap\dot{\mathbf{Q}}}\left(\mathbf{D}_\mathbf{c}^{-\frac{1}{2}}(\mathbf{Z}-\mathbf{r}\mathbf{c}^\top)^\top\dot{\mathbf{f}}_\ell\dot{\delta}_\ell^{-1}\right)\dot{\delta}_\ell \\
&= \mathbf{D}_\mathbf{c}^{-\frac{1}{2}}\,\mathrm{proj}^{\cdot\perp}_{\mathcal{L}_{\mathcal{G},\dot{\mathbf{v}}}\cap\mathcal{L}_2\cap\dot{\mathbf{Q}}}\left(\mathbf{D}_\mathbf{c}^{-\frac{1}{2}}\mathbf{Z}^\top\dot{\mathbf{f}}_\ell\dot{\delta}_\ell^{-1}\right)\dot{\delta}_\ell \\
&= \mathbf{D}_\mathbf{c}^{-\frac{1}{2}}\,\mathrm{proj}^{\cdot\perp}_{\mathcal{L}_{\mathcal{G},\dot{\mathbf{v}}}\cap\mathcal{L}_2\cap\dot{\mathbf{Q}}}\left(\mathbf{D}_\mathbf{c}^{\frac{1}{2}}\mathbf{C}\dot{\mathbf{u}}_\ell\right)\dot{\delta}_\ell \quad .
\end{aligned} \tag{D.4}$$

Appendix E. Simulated experiment to illustrate the loss of centering after projection

Barycentric projections are based on the property that, when data are (double) centered, any linear combination of the items will also be centered—a property that makes factor scores and loadings also centered.

When sparsification is involved, however, this property no longer holds, because the way that the components and loadings are obtained is not based on linear combinations anymore. However, there is one special case where this property holds again: when applying group-sparsification with sparse MCA.

To observe this property on a simple simulated example, let us consider a normal random variable $\dot{\mathbf{P}}$, and $I = 100$ i.i.d. realizations of this random variable, stored into a vector $\dot{\mathbf{p}}$. We then define the vector $\dot{\mathbf{x}}$ of the centered values of $\dot{\mathbf{q}}$ and $\dot{\mathbf{x}}_\mathcal{G}$ the vector of the "group"-centered values of $\dot{\mathbf{p}}$, where $\mathcal{G}$ defines a partition of $1, \cdots, I$ into 5 randomly assigned groups of 20. We then apply the projection operators $\mathrm{proj}_{\mathcal{L}_1}$ and $\mathrm{proj}_{\mathcal{L}_\mathcal{G}}$ on these two vectors with varying degrees of sparsity. The goal of this simulation is to visualize the effect of these projections on centered data, especially on the mean of the projected vectors, which is represented in Fig. E.6. We show on this plot the mean of the resulting projected vectors, as a function of the value of the sparsity parameter.

First, we see that projecting a centered vector on an $\mathcal{L}_1$- or an $\mathcal{L}_\mathcal{G}$-ball results in a vector that is sparser, but not centered anymore. However, when the input vector is group-centered, then the resulting (group-)projected vector is centered. This effect is a consequence of the fact that the $\mathcal{L}_\mathcal{G}$-projection operator is scaling each group individually (therefore keeping each group centered) or eliminating them.

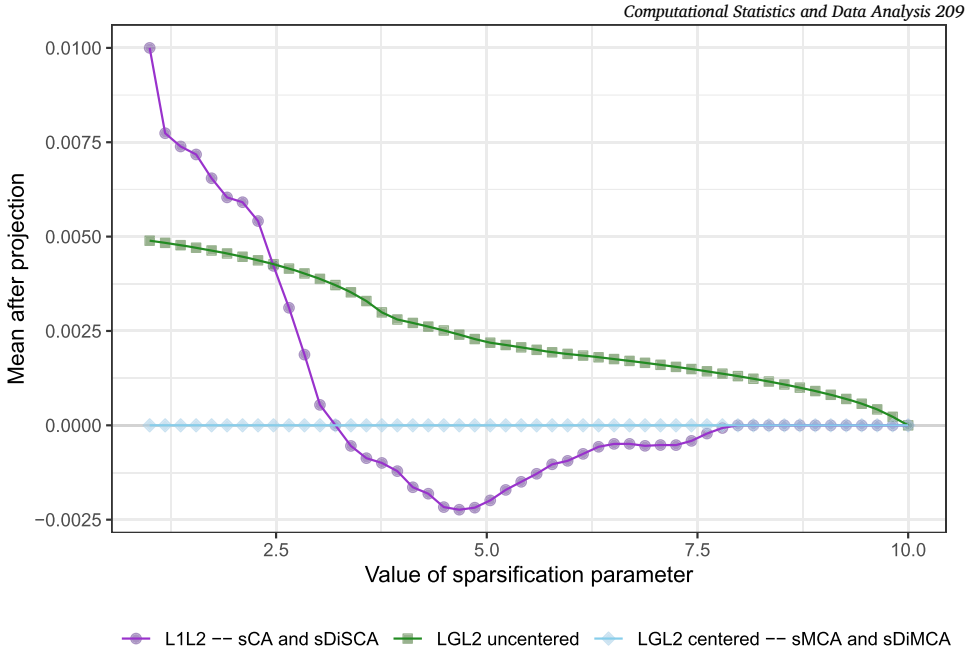

Fig. E.6. Projections onto the $\mathcal{L}_1$- or $\mathcal{L}_\mathcal{G}$- ball of a centered vector do not usually yield centered vectors, except in the case of the $\mathcal{L}_\mathcal{G}$ projection of a group-centered vector. Although the means of projections onto $\mathcal{L}_1$- and $\mathcal{L}_2$-balls (colored in purple) crossed 0, the mean is not exactly equal to, but only close to, 0 and does not satisfy the barycentric property.

Appendix F. Algorithms for plain SVD, GSVD, and sparse SVD (CSVD) and GSVD

Algorithm E.1: ALS algorithm of the SVD of $\mathbf{X}$.

Data: $\mathbf{X}$, ε, R ▷ Data $\mathbf{X}$, errors ε, and rank R
Result: SVD of $\mathbf{X}$
$\mathbf{P} \leftarrow \emptyset$; $\mathbf{Q} \leftarrow \emptyset$ ▷ $\mathbf{P}$ and $\mathbf{Q}$ store the left and right singular vectors
for $\ell = 1, \ldots, R$ **do**
  Initialize $\mathbf{p}^{(0)}$ and $\mathbf{q}^{(0)}$ ▷ Initialize $\mathbf{p}$ and $\mathbf{q}$
  $\delta^{(0)} \leftarrow 0$; $\delta^{(1)} \leftarrow \mathbf{p}^{(0)\top}\mathbf{X}\mathbf{q}^{(0)}$ ▷ Update δ with $\mathbf{p}^{(0)}$ and $\mathbf{q}^{(0)}$
  $t \leftarrow 0$
  while $\left(\left\|\mathbf{p}^{(t+1)} - \mathbf{p}^{(t)}\right\|_2 \geq \varepsilon\right)$ or $\left(\left\|\mathbf{q}^{(t+1)} - \mathbf{q}^{(t)}\right\|_2 \geq \varepsilon\right)$ **do**
    $\mathbf{p}^{(t+1)} \leftarrow \mathbf{X}\mathbf{q}^{(t)}$
    $\mathbf{q}^{(t+1)} \leftarrow \mathbf{X}^\top\mathbf{p}^{(t+1)}$
    $\delta^{(t+1)} \leftarrow \mathbf{p}^{(t+1)\top}\mathbf{X}\mathbf{q}^{(t+1)}$
    $t \leftarrow t + 1$ ▷ Iterate until $\mathbf{p}$ and $\mathbf{q}$ are stable
  $\delta_\ell \leftarrow \delta^{(t)}$; $\mathbf{p}_\ell \leftarrow \mathbf{p}^{(t)}$; $\mathbf{q}_\ell \leftarrow \mathbf{q}^{(t)}$
  $\mathbf{P} \leftarrow \left[\mathbf{P} \mid \mathbf{p}_\ell\right]$
  $\mathbf{Q} \leftarrow \left[\mathbf{Q} \mid \mathbf{q}_\ell\right]$
  $\mathbf{X} \leftarrow \mathbf{X} - \delta_\ell \mathbf{p}_\ell^\top \mathbf{q}_\ell$

Appendix G. Convergence analysis of the ALS algorithm

The proof of convergence of Algorithm 1 directly follows from the proof of the sparse SVD (E.3) from Guillemot et al. (2019), Appendix D. To go further, we show the results of the convergence analysis on a simple structured, small simulated dataset. Fig. G.7 shows both the data and the result of the convergence analysis of the alternating least squares. The convergence analysis consists in exploring the variation of $\delta = \mathbf{U}^\top\widetilde{\mathbf{X}}\mathbf{V}$ for each dimension, until convergence. First, we see that, as expected and previously observed by Witten et al. (2009), the dimensions are not always estimated in a monotonic order of the pseudo singular values. A consequence of including sparsity is that the dimensions need to be reordered once all the dimensions required by the user have been estimated by the ALS algorithm. Second, we see that, indeed, in this example, the algorithm converges, but the criterion is not necessarily strictly increasing (see the slight decrease in δ for the estimation of Dimension 4). For this analysis, the sparsity parameters were set to a third of their maximum value ($\sqrt{I}/3$ for the left parameters and $\sqrt{J}/3$ for the right parameters).

Algorithm E.2: ALS algorithm of the GSVD of $\mathbf{X}$.

Data: $\mathbf{X}$, $\mathbf{M}$, $\mathbf{W}$, ε, R ▷ Data $\mathbf{X}$, row metric matrix $\mathbf{M}$, column metric matrix $\mathbf{W}$, error ε, and rank R
Result: GSVD of $\mathbf{X}$
$\widetilde{\mathbf{X}} = \mathbf{M}^{\frac{1}{2}}\mathbf{X}\mathbf{W}^{\frac{1}{2}}$
$\mathbf{U} \leftarrow \emptyset$; $\mathbf{V} \leftarrow \emptyset$ ▷ $\mathbf{U}$ and $\mathbf{V}$ store the *generalized* left and right singular vectors
for $\ell = 1, \ldots, R$ **do**
  Initialize $\mathbf{p}^{(0)}$ and $\mathbf{q}^{(0)}$ ▷ Initialize $\mathbf{p}$ and $\mathbf{q}$
  $\delta^{(0)} \leftarrow 0$; $\delta^{(1)} \leftarrow \mathbf{p}^{(0)\top}\widetilde{\mathbf{X}}\mathbf{q}^{(0)}$ ▷ Update δ with $\mathbf{p}^{(0)}$ and $\mathbf{q}^{(0)}$
  $t \leftarrow 0$
  while $\left(\left\|\mathbf{p}^{(t+1)} - \mathbf{p}^{(t)}\right\|_2 \geq \varepsilon\right)$ or $\left(\left\|\mathbf{q}^{(t+1)} - \mathbf{q}^{(t)}\right\|_2 \geq \varepsilon\right)$ **do**
    $\mathbf{p}^{(t+1)} \leftarrow \widetilde{\mathbf{X}}\mathbf{q}^{(t)}$
    $\mathbf{q}^{(t+1)} \leftarrow \widetilde{\mathbf{X}}^{\top}\mathbf{p}^{(t+1)}$
    $\delta^{(t+1)} \leftarrow \mathbf{p}^{(t+1)\top}\widetilde{\mathbf{X}}\mathbf{q}^{(t+1)}$
    $t \leftarrow t + 1$ ▷ Iterate until $\mathbf{p}$ and $\mathbf{q}$ are stable
  $\delta_\ell \leftarrow \delta^{(t)}$; $\mathbf{p}_\ell \leftarrow \mathbf{p}^{(t)}$; $\mathbf{q}_\ell \leftarrow \mathbf{q}^{(t)}$
  $\mathbf{u}_\ell \leftarrow \mathbf{M}^{-\frac{1}{2}}\mathbf{p}_\ell$
  $\mathbf{v}_\ell \leftarrow \mathbf{W}^{-\frac{1}{2}}\mathbf{q}_\ell$
  $\mathbf{U} \leftarrow \left[\mathbf{U} \mid \mathbf{u}_\ell\right]$
  $\mathbf{V} \leftarrow \left[\mathbf{V} \mid \mathbf{v}_\ell\right]$
  $\widetilde{\mathbf{X}} \leftarrow \widetilde{\mathbf{X}} - \delta_\ell \mathbf{p}_\ell^{\top}\mathbf{q}_\ell$

Note: The text colored in green is specific to the GSVD as compared to the SVD.

Algorithm E.3: ALS algorithm of the CSVD of $\mathbf{X}$.

Data: $\mathbf{X}$, $s_{\cdot\,\mathbf{p},\ell}$, $s_{\cdot\,\mathbf{q},\ell}$, ε, R ▷ Data $\mathbf{X}$, errors ε, and rank R
▷ sparse parameters $s_{\cdot\,\mathbf{p},\ell}$ and $s_{\cdot\,\mathbf{q},\ell}$ for singular vectors
Result: CSVD of $\mathbf{X}$
$\dot{\mathbf{P}} \leftarrow \emptyset$; $\dot{\mathbf{Q}} \leftarrow \emptyset$ ▷ $\dot{\mathbf{P}}$ and $\dot{\mathbf{Q}}$ store the pseudo-left and right singular vectors
for $\ell = 1, \ldots, R$ **do**
  Initialize $\dot{\mathbf{p}}^{(0)}$ and $\dot{\mathbf{q}}^{(0)}$ ▷ Initialize $\dot{\mathbf{p}}$ and $\dot{\mathbf{q}}$ either from SVD or randomly
  $\dot{\delta}^{(0)} \leftarrow 0$; $\dot{\delta}^{(1)} \leftarrow \dot{\mathbf{p}}^{(0)\top}\mathbf{X}\dot{\mathbf{q}}^{(0)}$ ▷ Update $\dot{\delta}$ with $\dot{\mathbf{p}}^{(0)}$ and $\dot{\mathbf{q}}^{(0)}$
  $t \leftarrow 0$
  while $\left(\left\|\dot{\mathbf{p}}^{(t+1)} - \dot{\mathbf{p}}^{(t)}\right\|_2 \geq \varepsilon\right)$ or $\left(\left\|\dot{\mathbf{q}}^{(t+1)} - \dot{\mathbf{q}}^{(t)}\right\|_2 \geq \varepsilon\right)$ **do**
    $\dot{\mathbf{p}}^{(t+1)} \leftarrow \mathrm{proj}(\mathbf{X}\dot{\mathbf{q}}^{(t)}, \mathscr{B}_{\mathcal{L}_2}(1) \cap \mathscr{B}_{\mathcal{L}_1}(s_{\cdot\,\mathbf{p},\ell}) \cap \dot{\mathbf{P}}^{\perp})$ ▷ Projection of $\mathbf{X}\dot{\mathbf{q}}^{(t)}$ onto the intersection
    $\dot{\mathbf{q}}^{(t+1)} \leftarrow \mathrm{proj}(\mathbf{X}^{\top}\dot{\mathbf{p}}^{(t+1)}, \mathscr{B}_{\mathcal{L}_2}(1) \cap \mathscr{B}_{\mathcal{L}_1}(s_{\cdot\,\mathbf{q},\ell}) \cap \dot{\mathbf{Q}}^{\perp})$
    $\dot{\delta}^{(t+1)} \leftarrow \dot{\mathbf{p}}^{(t+1)\top}\mathbf{X}\dot{\mathbf{q}}^{(t+1)}$
    $t \leftarrow t + 1$ ▷ Iterate until $\dot{\mathbf{p}}$ and $\dot{\mathbf{q}}$ are stable
  $\dot{\delta}_\ell \leftarrow \dot{\delta}^{(t)}$; $\dot{\mathbf{p}}_\ell \leftarrow \mathbf{p}^{(t)}$; $\dot{\mathbf{q}}_\ell \leftarrow \dot{\mathbf{q}}^{(t)}$
  $\dot{\mathbf{P}} \leftarrow \left[\dot{\mathbf{P}} \mid \dot{\mathbf{p}}_\ell\right]$
  $\dot{\mathbf{Q}} \leftarrow \left[\dot{\mathbf{Q}} \mid \dot{\mathbf{q}}_\ell\right]$

Note: The text colored in red is specific to the CSVD as compared to the SVD.

Appendix H. Asymmetric alternative to the sparsity index

The sparsity index we proposed in section 3.4 assumes that we give as much weight to the fit ratio as to the zero ratio. To favor either fit or zero ratio, we recommend using the following modified sparsity index (defined as the weighted geometric mean of the two ratios) that includes an additional parameter:

$$\varsigma(w) = \vartheta^{w} \times \left(\dot{\tau}\right)^{1-w} \tag{H.1}$$

where $w \in [0; 1]$ is the weight that controls which ratio (either fit or zero) is favored. Fig. H.8 shows the values of the asymmetric sparsity index $\varsigma(w)$ as a function of the fit ratio and the zero ratio for 4 values of w.

Appendix. Supplementary material

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.csda.2025.108179.

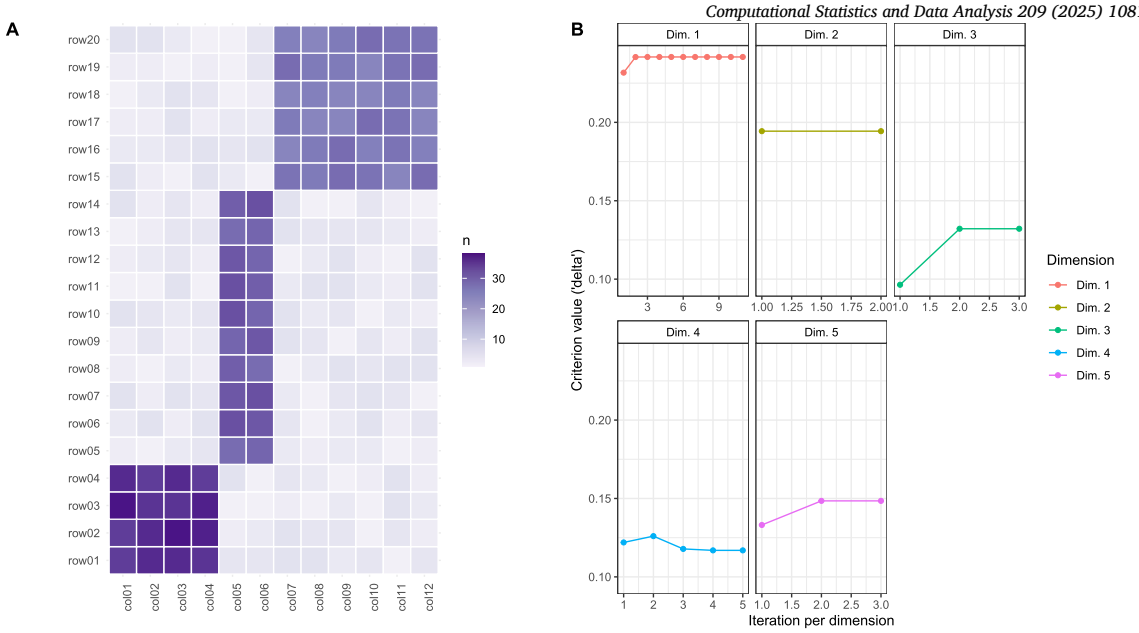

Fig. G.7. A - Simulated contingency table. B – Convergence of the ALS algorithm on the simulated data, on 5 dimensions.

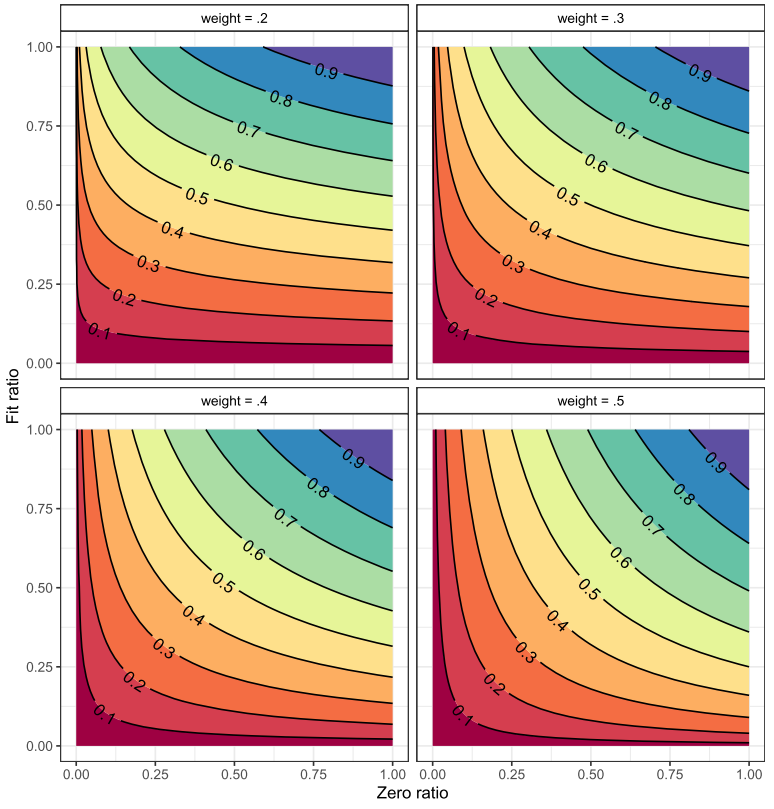

Fig. H.8. Four different fit ratio to zero-ratio space, for 4 values of the weight w. It depicts only values inferior to .5 because the behavior of the asymmetric sparsity index for values greater than .5 would correspond to a switch of the axes.

Algorithm E.4: ALS algorithm of sGSVD of $\mathbf{X}$.

Data: $\mathbf{X}$, $\mathbf{M}$, $\mathbf{W}$, $s_{\cdot_{\mathbf{p},\ell}}$, $s_{\cdot_{\mathbf{q},\ell}}$, ϵ, R

▷ Data $\mathbf{X}$, row metric matrix $\mathbf{M}$, column metric matrix $\mathbf{W}$, errors ϵ, and rank R

▷ sparse parameters $s_{\cdot_{\mathbf{p},\ell}}$ and $s_{\cdot_{\mathbf{q},\ell}}$ for singular vectors

Result: sGSVD of $\mathbf{X}$

$\widetilde{\mathbf{X}} = \mathbf{M}^{\frac{1}{2}}\mathbf{X}\mathbf{W}^{\frac{1}{2}}$

$\dot{\mathbf{U}} \leftarrow \emptyset$; $\dot{\mathbf{V}} \leftarrow \emptyset$ ▷ $\dot{\mathbf{U}}$ and $\dot{\mathbf{V}}$ store the pseudo-generalized left and right singular vectors

for $\ell = 1, \ldots, R$ **do**

  Initialize $\dot{\mathbf{p}}^{(0)}$ and $\dot{\mathbf{q}}^{(0)}$ ▷ Initialize $\dot{\mathbf{p}}$ and $\dot{\mathbf{q}}$ either from GSVD or randomly

  $\dot{\delta}^{(0)} \leftarrow 0$; $\dot{\delta}^{(1)} \leftarrow \dot{\mathbf{p}}^{(0)\top}\widetilde{\mathbf{X}}\dot{\mathbf{q}}^{(0)}$ ▷ Update $\dot{\delta}$ with $\dot{\mathbf{p}}^{(0)}$ and $\mathbf{q}^{(0)}$

  $t \leftarrow 0$ **while** $\left(\left\|\dot{\mathbf{p}}^{(t+1)} - \dot{\mathbf{p}}^{(t)}\right\|_2 \geq \varepsilon\right)$ or $\left(\left\|\dot{\mathbf{q}}^{(t+1)} - \dot{\mathbf{q}}^{(t)}\right\|_2 \geq \varepsilon\right)$ **do**

    $\dot{\mathbf{p}}^{(t+1)} \leftarrow \text{proj}(\widetilde{\mathbf{X}}\dot{\mathbf{q}}^{(t)}, \mathscr{B}_{\mathcal{L}_1}(s_{\cdot_{\mathbf{p},\ell}}) \cap \mathscr{B}_{\mathcal{L}_2}(1) \cap \dot{\mathbf{P}}^{\perp})$

    $\dot{\mathbf{q}}^{(t+1)} \leftarrow \text{proj}(\widetilde{\mathbf{X}}^{\top}\dot{\mathbf{p}}^{(t+1)}, \mathscr{B}_{\mathcal{L}_1}(s_{\cdot_{\mathbf{q},\ell}}) \cap \mathscr{B}_{\mathcal{L}_2}(1) \cap \dot{\mathbf{Q}}^{\perp})$

    $\dot{\delta}^{(t+1)} \leftarrow \dot{\mathbf{p}}^{(t+1)\top}\widetilde{\mathbf{X}}\dot{\mathbf{q}}^{(t+1)}$

    $t \leftarrow t + 1$ ▷ Iterate until $\dot{\mathbf{p}}$ and $\dot{\mathbf{q}}$ are stable

  $\dot{\delta}_\ell \leftarrow \dot{\delta}^{(t)}$; $\mathbf{p}_\ell \leftarrow \mathbf{p}^{(t)}$; $\dot{\mathbf{q}}_\ell \leftarrow \dot{\mathbf{q}}^{(t)}$

  $\dot{\mathbf{u}}_\ell \leftarrow \mathbf{M}^{-\frac{1}{2}}\dot{\mathbf{p}}_\ell$

  $\dot{\mathbf{v}}_\ell \leftarrow \mathbf{W}^{-\frac{1}{2}}\dot{\mathbf{q}}_\ell$

  $\dot{\mathbf{U}} \leftarrow \left[\dot{\mathbf{U}} \mid \dot{\mathbf{u}}_\ell\right]$

  $\dot{\mathbf{V}} \leftarrow \left[\dot{\mathbf{V}} \mid \dot{\mathbf{v}}_\ell\right]$

Note: The text colored in red describes the sparsification constraints of the CSVD that are also used in the sGSVD, and the text colored in green describes the metric constraints of the GSVD that are also used in the sGSVD.

References

Abdi, H., 2007a. Discriminant correspondence analysis. In: Salkind, N.J. (Ed.), Encyclopedia of Measurement and Statistics. Sage, Thousand Oaks, pp. 270–275.

Abdi, H., 2007b. Singular value decomposition (SVD) and generalized singular value decomposition (GSVD). In: Salkind, N.J. (Ed.), Encyclopedia of Measurement and Statistics. Sage, Thousand Oaks, pp. 907–912.

Abdi, H., Béra, M., 2014. Correspondence analysis. In: Alhajj, R., Rokne, J. (Eds.), Encyclopedia of Social Network Analysis and Mining. Springer, pp. 275–284.

Abdi, H., di Caccio, A., Saporta, G., 2024a. Old and new perspectives on optimal scaling. In: Beh Eric, J., Lombardo, R., Clavel Jose, G. (Eds.), Analysis of Categorical Data from Historical Perspectives. Springer, Singapore, pp. 1–24. chapter 9.

Abdi, H., Guillemot, V., Liu, R., Niang, N., Saporta, G., Yu, J.C., 2024b. From plain to sparse correspondence analysis: a generalized SVD approach. Stat. Appl. 35, 1–39. https://doi.org/10.26398/IJAS.0035-014.

Abdi, H., Valentin, D., 2007. Multiple correspondence analysis. In: Salkind, N. (Ed.), Encyclopedia of Measurement and Statistics. Sage, Thousand Oaks, CA, USA, pp. 651–666.

Abdi, H., Williams, L.J., 2010. Correspondence analysis. In: Salkind, N.J., Dougherty, D.M., B., F. (Eds.), Encyclopedia of Research Design. Sage, pp. 267–278.

Beaton, D., Chin Fatt, C.R., Abdi, H., 2014. An exposition of multivariate analysis with the singular value decomposition in R. Comput. Stat. Data Anal. 72, 176–189. https://doi.org/10.1016/j.csda.2013.11.006. https://www.sciencedirect.com/science/article/pii/S0167947313004441.

Beaton, D., Dunlop, J., Abdi, H., ADNI, 2016. Partial least squares correspondence analysis: a framework to simultaneously analyze behavioral and genetic data. Psychol. Methods 21, 621–651. https://doi.org/10.1037/met0000053.

Benzécri, J.P., 1979. Sur le calcul des taux d'inertie dans l'analyse d'un questionnaire. Cah. Anal. Données 4, 377–378.

Benzécri, J.P., et al., 1973. L'Analyse des Données, vol. 2. Dunod, Paris.

Bernard, A., Guinot, C., Saporta, G., 2012. Sparse principal component analysis for multiblock data and its extension to sparse multiple correspondence analysis. In: Proceedings of 20th International Conference on Computational Statistics (COMPSTAT 2012), pp. 99–106.

Cadima, J., Jolliffe, I.T., 1995. Loading and correlations in the interpretation of principal components. J. Appl. Stat. 22, 203–214.

Cattell, R.B., 1978. The Scientific Use of Factor Analysis in Behavioral and Life Sciences. Plenum Pr., New York.

Combettes, P., 1993. The foundations of set theoretic estimation. Proc. IEEE 81, 182–208.

Efron, B., 2020. Prediction, estimation, and attribution. J. Am. Stat. Assoc. 115, 636–655.

Efron, B., Hastie, T., 2016. Computer Age Statistical Inference. Cambridge University Press.

Escofier, B., 1969. L'analyse des correspondences. Bull. Bur. Univ. Rech. Opér. 11, 1–48.

Escofier, B., 1979. Une représentation des variables dans l'analyse des correspondances multiples. Rev. Stat. Appl. 27, 37–47.

Fichet, B., 2009. Metrics of L_p-type and distributional equivalence principle. Adv. Data Anal. Classif. 3, 305.

Greenacre, M., 1984. Theory and Applications of Correspondence Analysis. Academic Press, London (UK).

Guillemot, V., Beaton, D., Gloaguen, A., Löfstedt, T., Levine, B., Raymond, N., Tenenhaus, A., Abdi, H., 2019. A constrained singular value decomposition method that integrates sparsity and orthogonality. PLoS ONE 14, e0211463.

Hastie, T., Tibshirani, R., Friedman, J., 2009. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer Science & Business Media.

Hastie, T., Tibshirani, R., Wainwright, M., 2015. Statistical Learning with Sparcity the Lasso and Gneralizations. CRC Press, Boca Raton, FL.

Jolliffe, I.T., Trendafilov, N.T., Uddin, M., 2003. A modified principal component technique based on the LASSO. J. Comput. Graph. Stat. 12, 531–547.

Jolliffe, I.T., Uddin, M., 2000. The simplyfied component technique: an alternative to rotated principal components. J. Comput. Graph. Stat. 9, 689–710.

Journée, M., Nesterov, Y., Richtárik, P., Sepulchre, R., 2010. Generalized power method for sparse principal component analysis. J. Mach. Learn. Res. 11, 517–553.

Kannan, R., Vempala, S.S., Vetta, A., 2000. On clusterings-good, bad and spectral. In: Proceedings 41st Annual Symposium on Foundations of Computer Science, pp. 367–377. https://api.semanticscholar.org/CorpusID:61731027.

Krishnan, A., Williams, L.J., McIntosh, A.R., Abdi, H., 2011. Partial least squares (PLS) methods for neuroimaging: a tutorial and review. NeuroImage 56, 455–475.
Lebart, L., Morineau, A., Warwick, K.M., 1984. Multivariate Descriptive Statistical Analysis: Correspondence Analysis and Related Techniques for Large Matrices. Wiley, New York.
Liu, R., Niang, N., Saporta, G., Wang, H., 2023. Sparse correspondence analysis for large contingency tables. Adv. Data Anal. Classif. 1–20. https://doi.org/10.1007/S11634-022-00531-5/FIGURES/8. https://link.springer.com/article/10.1007/s11634-022-00531-5.
Lu, L., Gilmour, R., 2007. Developing a new measure of independent and interdependent views of the self. J. Res. Pers. 41, 249–257.
Mackey, L., 2009. Deflation methods for sparse PCA. In: Koller, D., Schuurmans, D., Bengio, Y., Bottou, L. (Eds.), Advances in Neural Information Processing Systems. Curran Associates, Inc, pp. 1017–1024. https://proceedings.neurips.cc/paper/2008/file/85d8ce590ad8981ca2c8286f79f59954-Paper.pdf.
Mori, Y., Kuroda, M., Makino, N., 2016. Sparse multiple correspondence analysis. In: Nonlinear Principal Component Analysis and Its Applications. Springer, Singapore, New York, pp. 47–56. chapter 5.
Qi, X., Luo, R., Zhao, H., 2013. Sparse principal component analysis by choice of norm. J. Multivar. Anal. 114, 127–160.
Rabin, L., Fink, L., Krishnan, A., Fogel, J., Berman, L., Bergdoll, R., 2018. A measure of basic math skills for use with undergraduate statistics students: the MACS. Stat. Educ. Res. J. 17, 179–195. https://doi.org/10.52041/SERJ.V17I2.165. https://iase-web.org/ojs/SERJ/article/view/165.
Robinson, D., 2021. gutenbergr: download and process public domain works from project Gutenberg. https://CRAN.R-project.org/package=gutenbergr. R package version 0.2.1.
Saporta, G., 2011. Probabilités, Analyse des Données et Statistique, 3rd ed. Technip, Paris, France.
Spielman, D.A., Teng, S.H., 2011. Spectral sparsification of graphs. SIAM J. Comput. 40, 981–1025.
Thurstone, L.L., 1935. The Vectors of Mind: Multiple-Factor Analysis for the Isolation of Primary Traits. University of Chicago Press.
Thurstone, L.L., 1947. Multiple-Factor Analysis; a Development and Expansion of the Vectors of Mind. University of Chicago Press.
Trendafilov, N., Gallo, M., 2021. Multivariate Data Analysis N Matrix Manifolds. Springer, Genève, Switzerland.
Trendafilov, N.T., 2014. From simple structure to sparse components: a review. Comput. Stat. 29, 431–454.
Trendafilov, N.T., Adachi, K., 2015. Sparse versus simple structure loadings. Psychometrika 80, 776–790.
Trendafilov, N.T., Fontanella, S., Adachi, K., 2017. Sparse exploratory factor analysis. Psychometrika 82, 778–794. https://doi.org/10.1007/s11336-017-9575-8.
Trendafilov, N.T., Jolliffe, I.T., 2006. Projected gradient approach to the numerical solution of the scotlass. Comput. Stat. Data Anal. 50, 242–253.
van den Berg, E., Schmidt, M., Friedlander, M.P., Murphy, K., 2008. Group sparsity via linear-time projection. Department of Computer Science. University of British Columbia. Technical Report.
Witten, D.M., Tibshirani, R., Hastie, T., 2009. A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. Biostatistics 10, 515–534.
World Health Organization, W., 2019. The ICD-10 Classification of Mental and Behavioural Disorders. World Health Organization, Genève, Switzerland. https://icd.who.int/browse10/2019/en/.
Yu, J.C., Gómez-Corona, C., Abdi, H., Guillemot, V., 2023. Sparse multiple factor analysis, sparse STATIS, and sparse DiSTATIS with applications to sensory evaluation. J. Chemom., e3443. https://doi.org/10.1002/cem.3443.
Yuan, M., Lin, Y., 2006. Model selection and estimation in regression with grouped variables. J. R. Stat. Soc., Ser. B, Stat. Methodol. 68, 49–67.
Zou, H., Hastie, T., 2005. Regularization and variable selection via the elastic net. J. R. Stat. Soc., Ser. B, Stat. Methodol. 67, 301–320.
Zou, H., Hastie, T., Tibshirani, R., 2006. Sparse principal component analysis. J. Comput. Graph. Stat. 15, 265–286.