Article

A New Algorithm for Computing Disjoint Orthogonal Components in the Parallel Factor Analysis Model with Simulations and Applications to Real-World Data

Carlos Martin-Barreiro ^{1,2} , John A. Ramirez-Figueroa ^{1,2} , Xavier Cabezas ² , Victor Leiva ^{3,*} ,
Ana Martin-Casado ¹ and M. Purificación Galindo-Villardón ¹

¹ Department of Statistics, Universidad de Salamanca, 37008 Salamanca, Spain; cmmartin@espol.edu.ec (C.M.-B.); jramirez@espol.edu.ec (J.A.R.-F.); ammc@usal.es (A.M.-C.); pgalindo@usal.es (M.P.G.-V.)

² Faculty of Natural Sciences and Mathematics, Universidad Politécnica ESPOL, Guayaquil 090902, Ecuador; joxacabe@espol.edu.ec

³ School of Industrial Engineering, Pontificia Universidad Católica de Valparaíso, Valparaíso 2362807, Chile

* Correspondence: victor.leiva@pucv.cl or victorleivasanchez@gmail.com

Abstract: In this paper, we extend the use of disjoint orthogonal components to three-way table analysis with the parallel factor analysis model. Traditional methods, such as scaling, orthogonality constraints, non-negativity constraints, and sparse techniques, do not guarantee that interpretable loading matrices are obtained in this model. We propose a novel heuristic algorithm that allows simple structure loading matrices to be obtained by calculating disjoint orthogonal components. This algorithm is also an alternative approach for solving the well-known degeneracy problem. We carry out computational experiments by utilizing simulated and real-world data to illustrate the benefits of the proposed algorithm.

Keywords: degeneracy; disjoint components; heuristic algorithms; Parafac model; PCA; R software; three-way tables

Citation: Martin-Barreiro, C.; Ramirez-Figueroa, J.A.; Cabezas, X.; Leiva, V.; Martin-Casad, A.; Galindo-Villardón, M.P. A New Algorithm for Computing Disjoint Orthogonal Components in the Parallel Factor Analysis Model with Simulations and Applications to Real-World Data. *Mathematics* **2021**, *9*, 2058. <https://doi.org/10.3390/math9172058>

Academic Editor: Alma Y. Alanis

Received: 10 July 2021

Accepted: 24 August 2021

Published: 26 August 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction, Notations, and Objectives

1.1. Introduction and Bibliographical Review

A third-order tensor (from now on, a three-way table for the applications in multivariate statistics) is an $I \times J \times K$ array $\underline{X}$ that stores observations on I individuals for J variables in K situations. For each way, we have a mode (set of entities), where the individuals are at the first mode, the variables in the second mode, and the situations in the third mode.

A popular multivariate statistical method is the principal component analysis (PCA), which allows for classification of variables or individuals [1,2]. When performing PCA on three-way tables, the aim is to reduce the dimension of at least one of the modes. A popular model for analyzing a three-way table is the parallel factor analysis, abbreviated as “Parafac”, where the three modes are reduced to R components, with $R < I, J, K$. This trilinear model has its origins in the research carried out by Hitchcock in 1927 [3,4]. In 1970, Harshman [5] and, Carroll and Chang [6], based on Hitchcock’s ideas, proposed the Parafac model, as it is known today. A method for determining the number of components in the Parafac model can be seen in [7].

It is important that the loading matrices A (first mode), B (second mode) and C (third mode) in the Parafac model are easily interpretable. Scaling techniques, orthogonality constraints, and non-negativity constraints, are generally used to obtain simple structure loading matrices. It is known that the rows of a loading matrix are the original entities, while the columns are latent entities or components. For the loading matrix to be interpretable, in terms of determining the original entities that are represented by a particular component, the above techniques do not guarantee that a simple structure can be achieved

in the matrix. In practice, note that some original entities in the loading matrices have similar loads on two or more components.

In the Parafac model, the components do not need to be orthogonal. However, the orthogonality constraint on at least one of the loading matrices allows the possibility of obtaining an interpretable solution [8,9]. The degeneracy problem is well known in the Parafac model [10]. This problem occurs when the inputs of the loading matrices diverge (matrix inputs are unstable), despite an apparent convergence in the fit of the model (as the algorithm iterates, the fit tends to stabilize at a value). A solution obtained in this way is called a degenerate solution and should not be interpreted. The degeneracy problem arises because the three-way data do not fit with the Parafac model, in other words, the model is not appropriate for the data. For a more detailed discussion of the degeneracy problem, see [11]. The degeneracy problem can be detected when two of the components have a high inverse correlation. For this reason, the component correlation matrix Ψ is usually calculated. If at least one of the inputs of Ψ is a value close to -1 , it is suspected that the solution found is degenerate [12]. To avoid the degeneracy problem, orthogonality constraints are usually incorporated into the model in at least one of the loading matrices. Degeneracy can also be avoided by adding non-negativity constraints [13,14]. Imposing non-negativity constraints is also a technique used to obtain interpretable loading matrices. The use of an R package called *ThreeWay* that provides computational support for the Parafac model is illustrated in [15].

Sparse techniques consist of locating some zeros in the inputs of a matrix to obtain simple structure loading matrices. However, these techniques decrease the fit quality [16–18]. In a recent research [19], algorithms are proposed to build sparse loading matrices. Other authors [20,21] have used clusters in the Parafac model to obtain a better interpretation of the results. In all the above proposals, constraints are imposed on one mode. In general, the known techniques do not guarantee that a simple structure can be obtained for the loading matrices. However, the disjoint approach that we propose in this work for the Parafac model guarantees that a loading matrix is interpretable when computing disjoint orthogonal components. This is because each original entity of the matrix will belong to a unique component. Additionally, each component must represent at least one original entity.

In two-way tables, disjoint orthogonal components have proven to be a successful technique for obtaining simple structure loading matrices [22–24]. Calculation of disjoint orthogonal components for two-way tables by using an algorithm that is based on particle swarm optimization was proposed in [25]. A heuristic algorithm for calculating disjoint orthogonal components in the Tucker3 model is derived in [26], which is the first known algorithm that computes disjoint orthogonal components for three-way tables.

1.2. Acronyms, Notations, and Symbols

Next, Table 1 presents the acronyms, notations and math symbology considered in this paper to facilitate its reading.

1.3. Objectives and Description of Sections

This work is the result of an effort to incorporate disjoint orthogonal components in the Parafac model. Then, the objective of this paper is to propose a heuristic algorithm that allows the disjoint orthogonal components of the Parafac model to be computed. In addition, we present a procedure that provides a recommendation for the use of this algorithm, named here as *DisjointParafacALS*, because this is based on the alternating least squares (ALS) method. The *DisjointParafacALS* algorithm solves the degeneracy problem and eases the interpretation of loading matrices. We illustrate the algorithm with simulated and real-world data to show its potential applications.

The rest of the paper is organized as follows. Section 2 defines a disjoint orthogonal matrix and presents the mathematical model that must be optimized to calculate the disjoint orthogonal components in the Parafac model. In Section 3, we introduce the *DisjointParafacALS* algorithm. In Section 4, the computational aspects are presented while

Section 5 reports the numerical experiments based on simulated and real-world data. Finally, some conclusions, other potential applications, and ideas for further research are provided in Section 6.

Table 1. Acronyms, notations, and symbols employed in the present document.

Acronym / Notation / Symbol	Definition
PCA	Principal component analysis.
Parafac	Parallel factor analysis.
$\underline{X}$	Tensor or three-way table (three dimensional).
$\mathbf{X}$	Data matrix (two dimensional).
I, J, K	Number of entities in the first, second and third mode of the tensor, respectively.
R	Number of components in the Parafac model.
i, j, k, r	Indices.
x_{ijk}	Tensor element at position ijk , with $x_{ijk} \in \underline{X}$.
$\hat{x}_{ijk}$	Approximation of the element x_{ijk} .
ε_{ijk}	Error when approximating the element x_{ijk} .
$\mathbf{A}, \mathbf{B}, \mathbf{C}$	Loading matrices of the first, second and third mode of the Parafac model, respectively.
a_{ir}	Element of loading matrix $\mathbf{A}$ at position ir , with $a_{ir} \in \mathbf{A}$.
b_{jr}	Element of loading matrix $\mathbf{B}$ at position jr , with $b_{jr} \in \mathbf{B}$.
c_{kr}	Element of loading matrix $\mathbf{C}$ at position kr , with $c_{kr} \in \mathbf{C}$.
$\mathbf{X}_A, \mathbf{X}_B, \mathbf{X}_C$	Frontal, horizontal and vertical slices matrices from $\underline{X}$, respectively.
$\mathbf{E}_A, \mathbf{E}_B, \mathbf{E}_C$	Error matrices when approximating matrices $\mathbf{X}_A, \mathbf{X}_B$ and $\mathbf{X}_C$, respectively.
Ψ	Component correlation matrix.
$\mathbf{I}_J$	The $J \times J$ identity matrix.
$\circ$	Hadamard product.
$\odot$	Khatri–Rao product.
$\mathbf{X}^\top$	The transpose of matrix $\mathbf{X}$.
$\mathbf{X}^+$	The Moore–Penrose generalized inverse of matrix $\mathbf{X}$.
$\ \mathbf{X}\ $	The Frobenius norm of matrix $\mathbf{X}$.
$\mathbf{B} \leftarrow \text{vs}(\mathbf{X}, R, \text{To1})$	The DisjointPCA algorithm with tolerance To1 is applied to the data matrix $\mathbf{X}$, and the disjoint orthogonal loading matrix $\mathbf{B}$ with R components is obtained as a result using the Vichi-Saporta (vc) algorithm [22].

2. The Parafac Model and the Disjoint Approach

2.1. The Parafac Model

This model for an $I \times J \times K$ three-way table, $\underline{X}$ namely, has elements defined as

$$x_{ijk} = \hat{x}_{ijk} + \varepsilon_{ijk}, \quad i = 1, \dots, I, j = 1, \dots, J, k = 1, \dots, K, \quad (1)$$

where $\hat{x}_{ijk}$ is the approximation of the element $x_{ijk} \in \underline{X}$ and ε_{ijk} is the error term that contains all the variability not explained by the model. In (1), note that $\hat{x}_{ijk} = \sum_{r=1}^R a_{ir} b_{jr} c_{kr}$, where a_{ir} , b_{jr} , and c_{kr} are the elements of the $I \times R$, $J \times R$, and $K \times R$ matrices, $\mathbf{A} = (a_{ir})$, $\mathbf{B} = (b_{jr})$, and $\mathbf{C} = (c_{kr})$ namely, respectively, which are called loading matrices. The integer R is such that $R < I$, $R < J$ and $R < K$, with R representing the number of components required for each mode.

The Parafac model can be represented by matrix equations based on all modes stated as $\mathbf{X}_A = \mathbf{A}(\mathbf{C} \odot \mathbf{B})^\top + \mathbf{E}_A$, $\mathbf{X}_B = \mathbf{B}(\mathbf{C} \odot \mathbf{A})^\top + \mathbf{E}_B$, and $\mathbf{X}_C = \mathbf{C}(\mathbf{B} \odot \mathbf{A})^\top + \mathbf{E}_C$, where $\odot$ is the Khatri–Rao product. Furthermore, the matrices $\mathbf{X}_A$, $\mathbf{X}_B$, and $\mathbf{X}_C$ are the frontal, horizontal and vertical slices matrices from the tensor $\underline{X}$, respectively, whereas $\mathbf{E}_A$, $\mathbf{E}_B$ and $\mathbf{E}_C$ are the corresponding error matrices [27].

The three loading matrices stated above may be estimated using some algorithms, with the ParafacALS algorithm being the most popular [28]. This algorithm calculates the three loading matrices by fixing two of them and finding the third matrix. This calculation is performed iteratively either until (i) the fit of the model do not differ significantly in

two consecutive iterations (for example, $|\text{Fit}_\ell - \text{Fit}_{\ell-1}| < \text{Tol}$, with the tolerance Tol often being 10^{-6} , a value that the package *ThreeWay* uses by default), or (ii) if a maximum number of iterations of the ALS algorithm, *ALSMaxIter* namely, is attained (for example, *ALSMaxIter* = 10,000). The aforementioned two conditions, (i) and (ii), are called the ParafacALS stopping criteria.

2.2. The ParafacALS Algorithm

This approach is summarized in Algorithm 1 with its outputs being the model fit, loading matrices A , B , C , and component correlation $R \times R$ matrix Ψ . In Algorithm 1, the symbol $\circ$ is the Hadamard product [29], $\odot$ is the Khatri–Rao product, and $^+$ is the Moore–Penrose generalized inverse of a matrix. For initialization of matrices B and C , and to compute matrices X_A , X_B , and X_C , see details in [9].

Algorithm 1: ParafacALS approach

```

1 begin
2   Input:  $\underline{X}$ ,  $R$ ;
3   Initialization:  $B$ ,  $C$ ;
4   From  $\underline{X}$ , compute  $X_A$ ,  $X_B$  and  $X_C$ ;
5   repeat
6      $A \leftarrow X_A[(C \odot B)^T]^+$ ;
7      $B \leftarrow X_B[(C \odot A)^T]^+$ ;
8      $C \leftarrow X_C[(B \odot A)^T]^+$ ;
9   until the ParafacALS stopping criteria are reached;
10  Calculate  $\Psi = (A^T A) \circ (B^T B) \circ (C^T C)$ ;
11  Output:  $A$ ,  $B$ ,  $C$ ,  $\Psi$ , and Fit calculated as  $\text{Fit} = (\sum_{i=1}^I \sum_{j=1}^J \sum_{k=1}^K \hat{x}_{ijk}^2) / (\sum_{i=1}^I \sum_{j=1}^J \sum_{k=1}^K x_{ijk}^2) \times 100\%$ .
12 end
```

2.3. A Disjoint Approach for the Parafac Model

Next, the disjoint approach for the Parafac model is considered. The definition of a disjoint orthogonal matrix is as follows:

Definition of disjoint orthogonal matrix.

Let $X = (x_{ij})$ be an $I \times J$ matrix, where $I > J$. Then, X is disjoint if and only if, for any i , there exists a unique j such that $x_{ij} \neq 0$, and for any j , there exists i such that $x_{ij} \neq 0$. If X also satisfies $X^T X = I_J$, where I_J is the $J \times J$ identity matrix, then X is a disjoint orthogonal matrix.

If a loading matrix of the Parafac model satisfies the definition of a disjoint orthogonal matrix, its interpretation is simplified. Suppose, for example, that the entities in the second mode of a data tensor are the following courses ($J = 8$): grammar, math, physics, psychology, literature, history, chemistry and sports. Suppose also that the loading matrix B is the one observed in Table 2 with $R = 3$ components. Note that component COMP1 groups the courses of math, physics and chemistry. We can then affirm that component COMP1 is the “science component”. Component COMP2 represents the courses of grammar, psychology, literature and history. We can call component COMP2 as the “humanity component”. Component COMP3 only has the sports course. If there were other courses like gymnastics, crossfit and yoga, surely component COMP3 would catch them. Then, component COMP3 would group courses related to physical activities. By imposing the constraint on the Parafac model that one of the loading matrices is disjoint orthogonal, a clear interpretation of the components can be obtained. The algorithm proposed in this work allows us to compute disjoint orthogonal loading matrices like those that have been

observed in the previous illustrative example. See Section 5 for other examples of disjoint orthogonal matrices.

Table 2. Loading matrix B of the example of courses.

Course	COMP1	COMP2	COMP3
Grammar	0	0.42683051	0
Math	0.58009486	0	0
Physics	0.56675657	0	0
Psychology	0	0.52692660	0
Literature	0	0.51315878	0
History	0	0.52614840	0
Chemistry	0.58504440	0	0
Sports	0	0	1

Below, we present the mathematical model of an optimization problem that must be solved when one of the three loading matrices A , B or C is required to be disjoint orthogonal. The number of decision variables of this problem is $(I + J + K)R$. Suppose that B is required to be a disjoint orthogonal matrix. Thus, the model is described as

$$\min_{A,B,C} \|X_A - \underbrace{A(C \odot B)^T}_{\tilde{X}_A}\|^2, \quad (2)$$

$$\text{subject to: } B^T B = I_R, \quad \text{with } B \text{ being a disjoint matrix.} \quad (3)$$

In (2), note that $\|\cdot\|$ is the Frobenius norm. The full constraint stated in (3) is needed for the matrix B to be disjoint orthogonal. In the previous mathematical model, the objective function that is defined in (2) must be minimized, but in practice, the fit is calculated according to step 11 of Algorithm 1. The optimization problem can be solved with the DisjointParafacALS algorithm that allows for a dimensional reduction of R components in its three modes. The focus of this work is on the case where one of the three loading matrices is constrained to be a disjoint orthogonal matrix.

In a two-way table, a disjoint orthogonal loading matrix B can be obtained by the following formulation:

$$\min_{A,B} \|X - AB^T\|^2 \quad (4)$$

$$\text{subject to: } B^T B = I_R, \quad \text{with } B \text{ being a disjoint matrix.} \quad (5)$$

where X is the $I \times J$ data matrix with I individuals and J variables, A is the $I \times R$ score matrix, and B is the $J \times R$ loading matrix. Observe that $R < J$ is the required number of components in the mode of variables and the full constraint (5) is needed for the loading matrix B to be disjoint orthogonal.

In this paper, we use a greedy algorithm, known as the DisjointPCA algorithm, that was proposed in [22], in order to find a solution to the minimization problem defined in (4). The DisjointPCA algorithm plays a fundamental role in the operation of the DisjointParafacALS algorithm.

The notation $B \leftarrow \text{vs}(X, R, \text{To1})$ means that the DisjointPCA algorithm is applied to the data matrix X with a tolerance To1 , and the disjoint orthogonal loading matrix B with R components is obtained as a result using the Vichi-Saporta (vc) algorithm [22]. Here, as mentioned, To1 is a tolerance parameter that represents the maximum distance allowed in the fit of the model for two consecutive iterations of the DisjointPCA algorithm. The above notation is used to explain how the DisjointParafacALS algorithm works. For more details about the DisjointPCA algorithm, see [23,25]. The DisjointPCA algorithm is also used by the DisjointTuckerALS algorithm in [26] to compute disjoint orthogonal components in the Tucker3 model.

3. The DisjointParafacALS Algorithm

3.1. Definitions

This novel algorithm allows us to compute the disjoint orthogonal components in the Parafac model. The algorithm has been designed to compute a single disjoint orthogonal loading matrix. The operation of the DisjointParafacALS algorithm is explained below. Before executing the algorithm, we recommend carrying out the corresponding preprocessing of the data. The DisjointParafacALS algorithm has two stages and its input parameters are:

- $\underline{X}$: Three-way table of data;
- R : Number of components in modes A, B, and C;
- $ALSM_{\text{Iter}}$: Maximum number of iterations of the ALS algorithm; and
- $To1$: Maximum distance allowed in the fit of the model for two consecutive iterations.

3.2. Stages of the Algorithm

[Stage 1] Computation of a disjoint orthogonal loading matrix with the DisjointPCA algorithm: The first stage is computing the disjoint loading matrix. In order to obtain R disjoint orthogonal components in the loading matrices A , B , or C , the DisjointPCA algorithm is applied to the matrices $X_A^\top$, $X_B^\top$, or $X_C^\top$, respectively. If A , B , and C are required to be disjoint orthogonal, then we have that: $A \leftarrow \text{vs}(X_A^\top, R, To1)$; $B \leftarrow \text{vs}(X_B^\top, R, To1)$; and $C \leftarrow \text{vs}(X_C^\top, R, To1)$, respectively.

[Stage 2] Computation of non-disjoint orthogonal loading matrices: The second stage is computing the non-disjoint loading matrices. For instance, if it is required that the matrix B has disjoint orthogonal components, the ALS algorithm must be applied to compute loading matrices A and C (B matrix is fixed) as happens in the ParafacALS algorithm, initializing A or C ; see Figure 1. The other two cases can be developed from the illustration based on B in this figure. The DisjointParafacALS algorithm finishes by outputting the matrices A , B , and C , and the fit of the model.

3.3. Algorithm and Flowchart

We summarize our proposal in Algorithm 2 to explain how the DisjointParafacALS algorithm works when performing a component analysis for three-way tables with the Parafac model. Computing the disjoint orthogonal components in Step 6 of Algorithm 2 simultaneously in two or three loading matrices is an approach that the analyst must choose. However, we do not recommend this approach due to the significant loss of fit that we have observed in the different computational experiments carried out. We suggest computing a single disjoint orthogonal loading matrix.

Note that the DisjointParafacALS algorithm can be utilized regardless of whether the degeneracy problem is present or not. This algorithm can be applied directly to a three-way table in order to interpret the loading matrices, with the corresponding loss of fit in the model; see Figure 2.

Figure 1. Flowchart of the DisjointParafacALS algorithm to compute a single disjoint orthogonal matrix (B).

Algorithm 2: Procedure for using the DisjointParafacALS algorithm

- Step 1** Collect the data in an $I \times J \times K$ three-way table $\underline{X}$, where I is the number of individuals, J is the number of variables, and K is the number of situations or times.
 - Step 2** Preprocess $\underline{X}$ according to the analyst's criterion.
 - Step 3** Fit $\underline{X}$ with an unconstrained Parafac model based on R components.
 - Step 4** Compute the fit of the model with no restrictions by using, for example, Algorithm 1. If degeneracy is suspected when analyzing the total number of iterations and the component correlation matrix Ψ , continue to **Step 5**; otherwise, go to **Step 8**.
 - Step 5** Impose orthogonality (or non-negativity) constraints to at least one of the loading matrices and compute loading matrices A , B , and C (apply the scaling technique if necessary to maintain the fit and improve the interpretation).
 - Step 6** Enforce the disjoint and orthogonality constraints on a single loading matrix and compute the loading matrices A , B , and C by using the DisjointParafacALS algorithm (that is, compute the disjoint orthogonal components for one loading matrix).
 - Step 7** Compare the results obtained in **Step 5** and **Step 6**, and go to **Step 10**.
 - Step 8** Calculate the loading matrices A , B , and C without constraints by using, for example, Algorithm 1 (apply the scaling technique if necessary).
 - Step 9** Do **Step 6** and compare the results with those obtained in **Step 8**.
 - Step 10** Analyze the results and conclusions.
-

Figure 2. Flowchart for using the DisjointParafacALS algorithm.

4. Computational Aspects

4.1. Characteristics of Hardware and Software

Table 3 shows the details of the software and hardware used in the implementation of the algorithms and in the computational experiments carried out.

Table 3. Hardware and software used in computational experiments.

Element	Characterist
Operating System	Windows 10 for 64 bits
RAM	8 Gigabytes
Processor	Intel Core i7-4510U 2-2.60 GHZ
Development tool—IDE—	Microsoft Visual Studio Express
Programming language	C#.NET
Statistical software	R
Data entry and presentation of results	Microsoft Excel
Communication between C#.NET and Excel	COM+
Communication between C#.NET and R	R.NET
Random number generation	runif function of the stats package
Singular value decomposition (SVD)	irlba function of the irlba package
Moore–Penrose generalized inverse of a matrix	ginv function of the MASS package

We must mention that the DisjointParafacALS algorithm requires more processing time (runtime) than the ParafacALS algorithm, even if the degeneracy problem is present. This is explained by the fact that calculating a disjoint orthogonal loading matrix consumes significant computational resources, which leads to longer processing time. The Disjoint-PCA, ParafacALS, and DisjointParafacALS algorithms that are presented in this paper were implemented mainly in C#.NET. For the particular case of the generation of random numbers, the SVD decomposition and the calculation of the generalized inverse of a matrix, the R statistical software was used.

4.2. Experimental Settings for Simulated Data

We designed and implemented a simulation algorithm to randomly build a three-way table with a disjoint latent structure of R components and its three modes, according to the Parafac model. Then, the DisjointParafacALS algorithm should be able to detect such a structure.

The loading matrices A , B and C used in the numerical experiments related to the simulated data were generated as in [26]. In order to complete the creation of the random three-way table $\underline{X}$, the matrix equation was defined as

$$\mathbf{X}_A = A(C \odot B)^\top. \quad (6)$$

The $I \times JK$ matrix $\mathbf{X}_A$ defined in (6) had the frontal slices of $\underline{X}$. The algorithm that built the $I \times J \times K$ three-way random table $\underline{X}$ had to implement an application φ stated as

$$\begin{aligned} \varphi : \mathbb{N}^4 \times \mathbb{N}^R \times \mathbb{N}^R \times \mathbb{N}^R &\rightarrow T_{I \times J \times K} \\ \left(\{I, J, K, R\}, \{\alpha_r\}_{r=1}^R, \{\beta_r\}_{r=1}^R, \{\gamma_r\}_{r=1}^R \right) &\mapsto \varphi \left(\{I, J, K, R\}, \{\alpha_r\}_{r=1}^R, \{\beta_r\}_{r=1}^R, \{\gamma_r\}_{r=1}^R \right). \end{aligned} \quad (7)$$

This was subject to: $R < I$, $R < J$, and $R < K$. Note that $T_{I \times J \times K}$ is the set of all three-way tables with real entries. Additionally, it must satisfy that

$$I = \sum_{r=1}^R \alpha_r, \quad J = \sum_{r=1}^R \beta_r, \quad K = \sum_{r=1}^R \gamma_r, \quad (8)$$

where α_r is the number of original individuals in the r th latent individual; β_r is the number of original variables in the r th latent variable; and γ_r is the number of original locations in the r th latent location. The output φ defined in (7) is the $I \times J \times K$ three-way table $\underline{X}$ that has a disjoint latent structure to be detected by the DisjointParafacALS algorithm.

4.3. Experimental Application with Simulated Data

We conducted computational experiments to evaluate the performance and benefits of the DisjointParafacALS algorithm. The first experiment corresponded to data simulated by an algorithm that was exclusively designed to generate three-way data with a disjoint structure according to the Parafac model. The other experiments are presented in the following section and related to (i) real data of evaluation of TV programs [30]; (ii) real data of perception of parents' behavior by their children and by the parents themselves [31]; and (iii) real data from the study of tongue movements when speakers pronounce vowels in English [32].

Here, we show how the DisjointParafacALS algorithm works by using the application φ to generate the $15 \times 20 \times 10$ three-way random table $\underline{X}$. Specifically, consider the instance given by $\varphi(\{15, 20, 10, 3\}, \{5, 6, 4\}, \{7, 8, 5\}, \{2, 5, 3\})$, according to the definition of φ given in (7), with $R = 3$. Furthermore, constraints stated in (8) were satisfied. The other parameter setting was given by $\text{ALSMaxIter} = 10,000$ and $\text{To1} = 10^{-6}$. Table 4 shows the component correlation matrix Ψ from where we can conclude, based on the output, that the degeneracy problem was not present in the data.

Table 4. Component correlation matrix Ψ for the simulated data.

	COMP1	COMP2	COMP3
COMP1	1	0.019363	0.024297
COMP2	0.019363	1	0.010313
COMP2	0.024297	0.010313	1

The DisjointParafacALS algorithm was executed in three different situations. There are four execution scenarios. In the first scenario, the ParafacALS algorithm gave a fit of 99.31%. In the second scenario, in which only the loading matrix *A* was disjoint orthogonal, a fit of 95.70% was obtained. In the third scenario, in which only the loading matrix *B* was disjoint orthogonal, a fit of 93.71% was obtained. In the last and fourth scenario, in which only the loading matrix *C* was disjoint orthogonal, a fit of 95.76% was obtained. These computational experiments are summarized in Table 5.

Table 6 reports the loading matrix *A* for the second scenario. Observe that the DisjointParafacALS algorithm was able to detect the disjoint structure in the first mode.

Table 7 presents the loading matrix *B* for the third scenario. Note that indeed the DisjointParafacALS algorithm was able to assess the disjoint structure in the second mode.

Table 8 shows the loading matrix *C* for the fourth scenario. Notice that the DisjointParafacALS algorithm was able to identify the disjoint structure in the third mode.

As observed in Tables 6–8, the disjoint structure facilitated the interpretation of the loading matrices, but with some loss of fit. In a fifth scenario, the DisjointParafacALS algorithm was executed with the loading matrices *A* and *B* set to be disjoint orthogonal as a constraint (illustration of a computational experiment with two disjoint orthogonal matrices). A fit of 87.41% was obtained, that is, with two disjoint orthogonal matrices, we obtain a high loss of fit. In the rows of Tables 6–8 are the original entities, and in the columns are the latent entities.

We recommend running the DisjointParafacALS algorithm with a single disjoint orthogonal matrix as shown in Table 5, and where appropriate, running the DisjointParafacALS algorithm with two disjoint orthogonal matrices.

Table 5. Disjoint orthogonal components, its fit (in %) and runtime (in minutes) for the indicated matrix with simulated data.

Disjoint Orthogonal Components	Fit (in %)	Runtime (in Minutes)
None (ParafacALS)	99.31	0.0007
<i>A</i> (DisjointParafacALS)	95.70	0.1736
<i>B</i> (DisjointParafacALS)	93.71	0.1862
<i>C</i> (DisjointParafacALS)	95.76	0.1325

Table 6. Loading matrix *A* with the DisjointParafacALS algorithm for the simulated data.

Subject (Individual)	COMP1	COMP2	COMP3
<i>S</i> ₁	−0.39987415	0	0
<i>S</i> ₂	−0.38886439	0	0
<i>S</i> ₃	−0.46369775	0	0
<i>S</i> ₄	−0.47750129	0	0
<i>S</i> ₅	−0.49584480	0	0
<i>S</i> ₆	0	0.45327357	0
<i>S</i> ₇	0	0.40688744	0
<i>S</i> ₈	0	0.37939130	0
<i>S</i> ₉	0	0.34345457	0
<i>S</i> ₁₀	0	0.41482141	0
<i>S</i> ₁₁	0	0.44159947	0
<i>S</i> ₁₂	0	0	−0.54088342
<i>S</i> ₁₃	0	0	−0.41894439
<i>S</i> ₁₄	0	0	−0.53459196
<i>S</i> ₁₅	0	0	−0.49612716

Table 7. Loading matrix **B** with the DisjointParafacALS algorithm for the simulated data.

Variable	COMP1	COMP2	COMP3
V_1	−0.40287714	0	0
V_2	−0.38117182	0	0
V_3	−0.38769074	0	0
V_4	−0.34328899	0	0
V_5	−0.31499905	0	0
V_6	−0.34929550	0	0
V_7	−0.45057171	0	0
V_8	0	−0.39457824	0
V_9	0	−0.33906833	0
V_{10}	0	−0.37809945	0
V_{11}	0	−0.30556330	0
V_{12}	0	−0.32902385	0
V_{13}	0	−0.32439850	0
V_{14}	0	−0.33306471	0
V_{15}	0	−0.41059636	0
V_{16}	0	0	−0.46109260
V_{17}	0	0	−0.43821079
V_{18}	0	0	−0.41637465
V_{19}	0	0	−0.38723790
V_{20}	0	0	−0.52157826

Table 8. Loading matrix **C** with the DisjointParafacALS algorithm for the simulated data.

Time	COMP1	COMP2	COMP3
T_1	−0.72171067	0	0
T_2	−0.69219484	0	0
T_3	0	−0.43505908	0
T_4	0	−0.39485340	0
T_5	0	−0.49233526	0
T_6	0	−0.42701150	0
T_7	0	−0.47966818	0
T_8	0	0	−0.59150295
T_9	0	0	−0.53266172
T_{10}	0	0	−0.60530633

5. Applications with Real-World Data

5.1. Applying the DisjointParafacALS Algorithm to TV Data

We considered the well-known “TV data” set, where 30 university students classify 15 American programs using 16 bipolar scales. As noted in [15], the first mode corresponded to students, the second mode corresponded to bipolar scales and the third mode corresponded to TV programs. The full data set can be downloaded from the R package named *ThreeWay*.

A three component dimensional reduction with the Parafac model was chosen as in [30]. As part of the preprocessing, the three-way table was centered first with respect to scales (second mode) and then centered with respect to TV programs (third mode). Thus, normalization was applied with respect to the bipolar scales.

In the reported computational experiments, the following values were used: $ALSM_{\text{MaxIter}} = 10,000$ and $To1 = 10^{-6}$. When executing the ParafacALS algorithm with these data, the degeneracy problem was presented. After more than 7000 iterations, we obtained a fit of 47.93% and the component correlation matrix stated in Table 9. In this table, the correlation between the components one and three was approximately -1 , a result that suggests the presence of degeneracy. Furthermore, a high number of iterations (7231 iterations) was required to reach the stopping criterion of the ParafacALS algorithm. To solve the degeneracy problem, in [10], it is recommended to impose orthogonality constraints on at least one of the loading matrices. By imposing the orthogonality constraint on the loading matrix **B**, a fit of 47.27% was obtained in 239 iterations. The component correlation matrix Ψ obtained for **B** is reported in Table 10.

Similarly to [15], in order to aid the interpretation, in this last execution, we set the columns of the loading matrices B and C to be unit vectors. Such a constraint was compensated in matrix A . Then, the DisjointParafacALS algorithm was executed in two different scenarios. In the first of them, only the loading matrix B had disjoint orthogonal components, but in the second scenario, only the loading matrix C had disjoint orthogonal components. The fit obtained and the processing time in all executions is summarized in Table 11.

Table 9. Component correlation matrix Ψ for TV data.

	COMP1	COMP2	COMP3
COMP1	1	0.05652381	−0.97495543
COMP2	0.05652381	1	−0.11506567
COMP3	−0.97495543	−0.11506567	1

Table 10. Component correlation matrix Ψ with the orthogonal matrix B for TV data.

	COMP1	COMP2	COMP3
COMP1	1	0	0
COMP2	0	1	0
COMP3	0	0	1

Table 11. Comparison of fit and runtime for TV data.

Scenario	Fit (in %)	Runtime (in Minutes)
ParafacALS - Degeneracy	47.93	0.4258
ParafacALS with Orthogonal B	47.27	0.0182
DisjointParafacALS with Disjoint Orthogonal B	41.12	0.5664
DisjointParafacLAS with Disjoint Orthogonal C	41.21	0.5456

Note that having disjoint orthogonal components affected the fit, but the interpretation increased, as we will illustrate later. As stated in [30], the three components could be interpreted as “humor”, “sensitivity” and “violence”. It should be noted that a negative value in loading matrix B was interpreted as belonging to the left side of the bipolar scale. Table 12 shows the orthogonal loading matrix B that was obtained using the ParafacALS algorithm. Table 13 reports the disjoint orthogonal matrix B that was obtained with the DisjointParafacALS algorithm. Loadings with absolute values greater than or equal to 0.25 that we observe in Tables 12 and 13 facilitate comparison. Except for minor differences, the interpretation was the same.

Table 14 presents the loading matrix C that was obtained using the ParafacALS algorithm. Table 15 reports the disjoint orthogonal matrix C that was reached with the DisjointParafacALS algorithm. Loadings with absolute values greater than or equal to 0.25 reported in Tables 14 and 15 facilitate comparison. Except for minor differences, the interpretation was the same.

Table 12. Orthogonal loading matrix *B* with the ParafacALS algorithm for TV data.

Bipolar Scale	Humor	Sensitivity	Violence
(1) Thrilling-Boring	−0.09	0.29	−0.12
(2) Intelligent-Idiotic	0.30	0.17	0.12
(3) Erotic-Not Erotic	−0.25	−0.06	−0.27
(4) Sensitive-Insensitive	0.03	−0.43	0.12
(5) Interesting-Uninteresting	0.00	0.37	0.26
(6) Fast-Slow	−0.08	0.19	−0.29
(7) Intell. Stimulating-Intell. Dull	0.27	0.20	0.25
(8) Violent-Peaceful	0.08	0.04	−0.68
(9) Caring-Callous	0.04	−0.47	0.02
(10) Satirical-Not Satirical	−0.46	0.09	0.18
(11) Informative-Uninformative	0.31	0.20	0.19
(12) Touching-Leave Me Cold	−0.07	−0.31	0.15
(13) Deep-Shallow	0.21	−0.14	0.16
(14) Tasteful-Crude	0.19	−0.31	0.02
(15) Real-Fantasy	0.38	0.06	−0.06
(16) Funny-Not Funny	−0.46	0.06	0.30

Table 13. Disjoint orthogonal matrix *B* with the DisjointParafacALS algorithm for TV data.

Bipolar Scale	Humor	Sensitivity	Violence
(1) Thrilling-Boring	0	0.40	0
(2) Intelligent-Idiotic	0.39	0	0
(3) Erotic-Not Erotic	−0.35	0	0
(4) Sensitive-Insensitive	0	−0.50	0
(5) Interesting-Uninteresting	0	0.37	0
(6) Fast-Slow	0	0	−0.52
(7) Intell. Stimulating-Intell. Dull	0.39	0	0
(8) Violent-Peaceful	0	0	−0.47
(9) Caring-Callous	0	−0.52	0
(10) Satirical-Not Satirical	−0.39	0	0
(11) Informative-Uninformative	0.40	0	0
(12) Touching-Leave Me Cold	0	−0.41	0
(13) Deep-Shallow	0	0	0.51
(14) Tasteful-Crude	0	0	0.50
(15) Real-Fantasy	0.36	0	0
(16) Funny-Not Funny	−0.38	0	0

5.2. Applying the DisjointParafacALS Algorithm to Kojima Data

A Parafac model was applied in [31]. The three-way table *X* was a $153 \times 18 \times 4$ tensor with 153 Japanese girls in the first mode, 18 behavior scales in the second mode, and four judgements or conditions in the third mode. The full data set can be downloaded from <http://three-mode.leidenuniv.nl> (accessed on 9 July 2021).

With these data, we studied the perception of parents' behavior by their children and by the parents themselves. To do so, the reactions of parents and children were received in parallel versions of the same questionnaire. The four conditions of the third mode were: FF (the father evaluates his own behavior), MM (the mother evaluates her own behavior), DF (the daughter evaluates the father's behavior) and DM (the daughter evaluates the mother's behavior).

Table 14. Loading matrix *C* with the ParafacALS algorithm for TV data.

TV Program	Humor	Sensitivity	Violence
(1) Mash	0.14	0.03	−0.11
(2) Charlie’s angels	0.21	0.13	0.54
(3) All in the family	0.20	0.04	−0.18
(4) 60 min	−0.35	−0.19	−0.18
(5) The tonight show	0.16	−0.23	−0.19
(6) Let’s make a deal	0.19	0.00	0.25
(7) The Waltons	−0.13	0.55	−0.23
(8) Saturday night live	0.35	−0.31	0.11
(9) News	−0.39	−0.27	0.00
(10) Kojak	0.09	−0.04	0.37
(11) Mork and Mindy	0.38	0.12	−0.10
(12) Jacques Cousteau	−0.40	−0.10	−0.29
(13) Football	−0.07	−0.32	0.39
(14) Little house on the prairie	−0.06	0.54	−0.18
(15) Wild kingdom	−0.31	0.04	−0.21

Table 15. Disjoint orthogonal matrix *C* with the DisjointParafacALS algorithm for TV data.

TV Program	Humor	Sensitivity	Violence
(1) Mash	0.17	0	0
(2) Charlie’s angels	0	0	0.53
(3) All in the family	0.28	0	0
(4) 60 min	−0.51	0	0
(5) The tonight show	0	−0.24	0
(6) Let’s make a deal	0	0	0.36
(7) The Waltons	0	0.60	0
(8) Saturday night live	0	−0.44	0
(9) News	−0.63	0	0
(10) Kojak	0	0	0.25
(11) Mork and Mindy	0.49	0	0
(12) Jacques Cousteau	0	0	−0.58
(13) Football	0	−0.31	0
(14) Little house on the prairie	0	0.55	0
(15) Wild kingdom	0	0	−0.44

The 18 scales were grouped into four types: Parental support or acceptance (PSA), behavioral and psychological control (BPC), rejection (R), and lax discipline (LD). As in [31], $R = 3$ components were chosen. Furthermore, as part of the data preprocessing, a centering was performed in the first mode and then a normalization in the second mode. In the reported computational experiments, the following values were used: $ALSM_{\text{MaxIter}} = 10,000$ and $To1 = 10^{-6}$. When executing the ParafacALS algorithm with these data, the degeneracy problem was once again presented. After more than 5 000 iterations, we reached a fit of 38.82% and the component correlation matrix Ψ is reported in in Table 16.

Table 16. Component correlation matrix Ψ with $R = 3$ components for Kojima data.

	COMP1	COMP2	COMP3
COMP1	1	−0.20159703	−0.93725207
COMP2	−0.20159703	1	0.06099119
COMP3	−0.93725207	0.06099119	1

Table 16 tells us that the degeneracy problem could be present again. Components 1 and 3 had a high negative correlation (value close to -1). In [31], the orthogonality condition of matrix A was imposed. We proceed to assume that matrix B must be disjoint orthogonal to improve the interpretation in the second mode and the DisjointParafacALS algorithm was used. We obtained that the component correlation matrix Ψ was the 3×3 identity matrix. The DisjointParafacALS algorithm was executed in two different scenarios. In the first of these, only the loading matrix B had disjoint orthogonal components with $R = 3$ components, and in the second scenario, only the loading matrix B had disjoint orthogonal components with $R = 4$ components. Additionally, the ParafacALS algorithm was executed with $R = 4$ components and the presence of the degeneracy problem was also observed. Table 17 reports that components 2 and 3 had a high negative correlation. The DisjointParafacALS algorithm was used to solve the degeneracy problem with $R = 4$ components. Then, we obtained that the component correlation matrix Ψ was the 4×4 identity matrix. The fit obtained and the processing time in all executions are summarized in Table 18.

Table 17. Component correlation matrix Ψ with $R = 4$ components for Kojima data.

	COMP1	COMP2	COMP3	COMP4
COMP1	1	0.09060317	−0.17620287	0.02597586
COMP2	0.09060317	1	−0.93490587	0.06082605
COMP3	−0.17620287	−0.93490587	1	−0.14096143
COMP4	0.02597586	0.06082605	−0.14096143	1

Table 18. Comparison of fit and runtime for Kojima data.

Scenario	Fit (in %)	Runtime (in Min)
ParafacALS ($R = 3$) - Degeneracy	38.82	0.0753
ParafacALS ($R = 3$) with orthogonal A	38.01	0.0084
DisjointParafacALS ($R = 3$) with disjoint orthogonal B	32.28	0.2396
ParafacALS ($R = 4$) - Degeneracy	45.95	0.0492
DisjointParafacALS ($R = 4$) with disjoint orthogonal B	36.69	0.4473

We proceed to the analysis of the loading matrix for the scales. In Table 19, we see the three components calculated in [31]. Table 20 shows the three disjoint components calculated with the DisjointParafacALS algorithm. In Table 19, it is observed that the component 1 mainly groups the “parental support or acceptance” scales with high positive loadings and the “rejection” scales with high negative loadings. The component 2 mainly groups both, the “behavioral and psychological control” and “rejection” scales with high positive loadings, the second one with lower values. In the component 3, we observe some “behavioral and psychological control” scales with high positive loadings. The “lax discipline” scales are not related with any particular component.

When analyzing the three components, note that component 1 represents the “parental support or acceptance” scales. The component 2 groups “behavioral and psychological control” and “rejection”. The component 3 is represents “lax discipline”. Table 21 reports what happens if four disjoint components are computed with the DisjointParafacALS algorithm. The components 1, 2, 3 and 4 group “parental support or acceptance”, “behavioral and psychological control”, “rejection” and “lax discipline” scales, respectively.

Table 19. Matrix *B* obtained by the orthogonal matrix *A* condition for Kojima data.

Scale	Scale Type	COMP1	COMP2	COMP3
(1) Acceptance	PSA	0.64	0.27	0.24
(2) Child centredness	PSA	0.54	0.32	0.27
(3) Possessiveness	BPC	0.06	0.47	0.38
(4) Rejection	R	−0.53	0.36	0.24
(5) Control	BPC	0.02	0.32	0.56
(6) Enforcement	BPC	−0.18	0.34	0.49
(7) Positive involvement	PSA	0.54	0.4	0.3
(8) Intrusiveness	BPC	0.16	0.37	0.49
(9) Control through guilt	R	−0.34	0.36	0.35
(10) Hostile control	BPC	−0.17	0.34	0.61
(11) Inconsistent discipline	LD	−0.3	0.43	0.14
(12) Nonenforcement	LD	0.28	−0.22	0.01
(13) Acceptance individuation	PSA	0.64	0.29	0.17
(14) Lax discipline	LD	0.37	0	0.04
(15) Instilling persistent anxiety	BPC	−0.11	0.41	0.49
(16) Hostile detachment	R	−0.58	0.24	0.16
(17) Withdrawal of relations	R	−0.45	0.35	0.23
(18) Extreme autonomy	LD	0.32	−0.2	0.02

Table 20. Disjoint orthogonal matrix *B* using $R = 3$ components and obtained with the Disjoint-ParafacALS algorithm for Kojima data.

Scale	Scale Type	COMP1	COMP2	COMP3
(1) Acceptance	PSA	0.49	0	0
(2) Child centredness	PSA	0.46	0	0
(3) Possessiveness	BPC	0	0.26	0
(4) Rejection	R	0	0.35	0
(5) Control	BPC	0	0.29	0
(6) Enforcement	BPC	0	0.33	0
(7) Positive involvement	PSA	0.47	0	0
(8) Intrusiveness	BPC	0.33	0	0
(9) Control through guilt	R	0	0.34	0
(10) Hostile control	BPC	0	0.36	0
(11) Inconsistent discipline	LD	0	0.26	0
(12) Nonenforcement	LD	0	0	0.57
(13) Acceptance individuation	PSA	0.46	0	0
(14) Lax discipline	LD	0	0	0.59
(15) Instilling persistent anxiety	BPC	0	0.34	0
(16) Hostile detachment	R	0	0.28	0
(17) Withdrawal of relations	R	0	0.33	0
(18) Extreme autonomy	LD	0	0	0.57

Table 21. Disjoint orthogonal matrix B using $R = 4$ components and obtained with the Disjoint-ParafacALS algorithm for Kojima data.

Scale	Scale Type	COMP1	COMP2	COMP3	COMP4
(1) Acceptance	PSA	0.52	0	0	0
(2) Child centredness	PSA	0.49	0	0	0
(3) Possessiveness	BPC	0	0.34	0	0
(4) Rejection	R	0	0	0.55	0
(5) Control	BPC	0	0.39	0	0
(6) Enforcement	BPC	0	0.37	0	0
(7) Positive involvement	PSA	0.50	0	0	0
(8) Intrusiveness	BPC	0	0.35	0	0
(9) Control through guilt	R	0	0.35	0	0
(10) Hostile control	BPC	0	0.43	0	0
(11) Inconsistent discipline	LD	0	0	0.45	0
(12) Nonenforcement	LD	0	0	0	0.57
(13) Acceptance individuation	PSA	0.50	0	0	0
(14) Lax discipline	LD	0	0	0	0.59
(15) Instilling persistent anxiety	BPC	0	0.41	0	0
(16) Hostile detachment	R	0	0	0.50	0
(17) Withdrawal of relations	R	0	0	0.49	0
(18) Extreme autonomy	LD	0	0	0	0.57

5.3. Applying the DisjointParafacALS Algorithm to Tongue Data

In [32], a study was carried out with the Parafac model about the position of certain points of the tongue while some speakers pronounced different vowels in English. These data were stored in the $10 \times 13 \times 5$ three-way table $\underline{X}$. The first mode corresponded to 10 vowels in English, the second mode corresponded to 13 predefined points of the tongue and the third mode corresponded to five speakers. To generate the data, x-rays were taken on five people while they pronounced the 10 vowels. The full data set is in [32].

Two components were chosen ($R = 2$) and a centering was carried out in the first mode. In the reported computational experiments, the following values were used: $ALSM_{\text{MaxIter}} = 10,000$ and $Tol = 10^{-6}$. Table 22 presents the fit obtained and the execution time in four scenarios. In the first one, no disjoint orthogonal components were computed. From the component correlation matrix observed in Table 23, the presence of the degeneracy problem is not suspected now. In the other three scenarios, disjoint orthogonal components were calculated for the loading matrices A , B and C , respectively. In addition, in the last three scenarios, the component correlation matrix Ψ is the 2×2 identity matrix.

From Table 24, observe that vowels 1, 2, 3 and 4 are better represented in the first component with positive loadings. Furthermore, vowels 8 and 9 were also better represented in the first component but with negative loadings. In addition, vowels 5 and 6 were better represented in the second component with negative loadings and vowel 10 was better represented in the second component with positive loading. When interpreting the disjoint components of Table 25, we conclude the same as in [32]. Vowel 7 had loadings less than -1 in both components in Table 24, but the DisjointParafacALS algorithm placed vowel 7 in the first component with negative loading.

Table 26 shows the components calculated in [32]. The first component was mainly related to large movements of the points of the tongue located in the front of the mouth, while the second component was mainly associated with larger movements of the points of the tongue located in the back of the mouth. In the analysis with disjoint components (see Table 27), note that the tongue points 11 to 16, which were the closest to the mouth, were represented in the first component with positive loadings. Tongue points 4 to 7, which were the points that are further from the mouth, were in the first component with negative loadings. Furthermore, the tongue points 8, 9 and 10 located in the back of the mouth were represented in the second component with negative loadings.

As a reference, keep in mind that the tongue point 4 was located near the epiglottis (in the throat behind the tongue) and the other tongue points were located upwards in the direction of the mouth ending at point 16 which was located near the lips.

Table 22. Comparison of fit and runtime for tongue data.

Disjoint Orthogonal Components	Fit (in %)	Runtime (in Min)
None (ParafacALS)	92.62	0.0016
A (DisjointParafacALS)	86.62	0.0913
B (DisjointParafacALS)	85.23	0.1432
C (DisjointParafacALS)	85.31	0.0765

Table 23. Component correlation matrix Ψ with $R = 2$ components for tongue data.

	COMP1	COMP2
COMP1	1	0.06890887
COMP2	0.06890887	1

Table 24. Loading matrix A with the ParafacALS algorithm for tongue data.

Vowel	COMP1	COMP2
1	1.52	0.69
2	0.97	0.33
3	1.04	0.42
4	0.77	−0.17
5	0.33	−0.53
6	−0.31	−2.04
7	−1.07	−1.24
8	−1.38	0.26
9	−1.05	0.58
10	−0.83	1.69

Table 25. Disjoint orthogonal matrix A with the DisjointParafacALS algorithm for tongue data.

Vowel	COMP1	COMP2
1	0.55	0
2	0.33	0
3	0.37	0
4	0.19	0
5	0	−0.12
6	0	−0.91
7	−0.51	0
8	−0.35	0
9	−0.20	0
10	0	0.40

Table 26. Loading matrix *B* with the ParafacALS algorithm for tongue data.

Tongue Point	COMP1	COMP2
4	−0.60	−1.19
5	−0.88	−1.09
6	−1.05	−0.95
7	−1.16	−0.59
8	−1.14	−0.01
9	−0.95	0.46
10	−0.55	0.93
11	0.14	1.24
12	0.87	1.42
13	1.26	1.37
14	1.41	1.22
15	1.32	0.93
16	0.91	0.49

Table 27. Disjoint orthogonal matrix *B* with the DisjointParafacALS algorithm for tongue data.

Tongue Point	COMP1	COMP2
4	−0.25	0
5	−0.30	0
6	−0.31	0
7	−0.29	0
8	0	−0.66
9	0	−0.61
10	0	−0.43
11	0.18	0
12	0.34	0
13	0.40	0
14	0.41	0
15	0.36	0
16	0.23	0

In Table 28, the speakers 1, 2 and 5 have negative loadings and are better represented in the first component. Furthermore, speaker 4 also has negative loading and is better represented in the second component, just as we see in Table 29 where we have the disjoint components. Speaker 3 has similar values in the two components of Table 28, but the DisjointParafacALS algorithm places speaker 3 in the second component with negative loading.

Table 28. Loading matrix *C* with the ParafacALS algorithm for tongue data.

Speaker	COMP1	COMP2
1	−0.43	−0.27
2	−0.70	−0.36
3	−0.37	−0.37
4	−0.34	−0.49
5	−0.47	−0.31

Table 29. Disjoint orthogonal matrix *C* with the DisjointParafacALS algorithm for tongue data.

Speaker	COMP1	COMP2
1	−0.46	0
2	−0.72	0
3	0	−0.66
4	0	−0.75
5	−0.52	0

6. Conclusions, Other Potential Applications, and Future Work

One of the most popular models for component analysis in three-way tables is the Parafac model. However, a big problem with the existing techniques in the Parafac model is that interpretable loading matrices are not always obtained.

In the present study, a novel heuristic algorithm for computing disjoint orthogonal components in a three-way table with the Parafac model was proposed. This algorithm mixes the ParafacALS and disjoint PCA algorithms. The computational experiments reported that the main benefit of the algorithm is the easier interpretation of the loading matrices. In addition, the algorithm permitted us to analyze a three-way table when the degeneracy problem exists. Such experiments indicated that the algorithm identifies and catches disjoint structures in a three-way table with the Parafac model. These experiments based on simulated and real data sets enabled us to demonstrate the good performance and potential applications of the new algorithm. This algorithm may be a helpful multivariate tool for practitioners, applied statisticians, and data scientists.

A limitation of our algorithm is that it does not guarantee an optimal solution. When additional constraints to those present in the original problem are absent, the feasible solutions state the global optimum. However, if the constraints of the DisjointParafacALS algorithm are incorporated, the feasible solutions are compressed, reaching a solution close to the global optimum. Therefore, the fit for the solution given by the algorithm is less than the fit of the ParafacALS algorithm. Nevertheless, the built-in constraints permit us to add zeros in the positions of the variables with low contribution in a component of the loading matrix, allowing the interpretation of the components more clearly. The new algorithm takes longer than the ParafacALS algorithm, even when the degeneracy problem is present in the data, so there is a trade-off between interpretation and speed.

In order to motivate practitioners, the new algorithm can be applied to:

- (i) Marketing and sales, for example in the selection of an appropriate commercial spoken for a given brand [8,21].
- (ii) The Parafac model also has applications in chemistry (chromatography, fluorescence spectroscopy, food industry, and second-order calibration) [33–35].
- (iii) There are also applications in environmental sciences related to water [36].
- (iv) The Parafac model is also used in psychometry [11,31].

Some open problems that arose from this study and that authors are analyzing are the following. We think that the disjoint method may be employed together with other existing methods to improve its performance, as in the multidimensional scaling of three-way data. Also, bootstrapping for the loading matrices of the Parafac model may be used. In addition, there exists a wide field of usages in statistical learning. Computing disjoint orthogonal components in higher order tensors is also a challenge.

Author Contributions: Data curation, X.C., J.A.R.-F., C.M.-B.; investigation, C.M.-B., M.P.G.-V.; formal analysis and methodology, C.M.-B., J.A.R.-F., X.C., V.L., A.M.-C., M.P.G.-V.; writing—original draft, C.M.-B., J.A.R.-F., X.C., V.L., A.M.-C., M.P.G.-V.; writing—review and editing, V.L., C.M.-B., X.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially supported by FONDECYT, project grant number 1200525 (V.L.), from the National Agency for Research and Development (ANID) of the Chilean government under the Ministry of Science, Technology, Knowledge, and Innovation.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data analyzed are available from the authors upon request.

Acknowledgments: The authors would also like to thank the editor and reviewers for their constructive comments which led to improving the presentation of the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hached, M.; Jbilou, K.; Koukouvinos, C.; Mitrouli, M. A multidimensional principal component analysis via the C-product Golub–Kahan–SVD for classification and face recognition. *Mathematics* **2021**, *9*, 1249. [\[CrossRef\]](#)
2. Martin-Barreiro, C.; Ramirez-Figueroa, J.A.; Cabezas, X.; Leiva, V.; Galindo-Villardón, M.P. Disjoint and functional principal component analysis for infected cases and deaths due to COVID-19 in South American countries with sensor-related data. *Sensors* **2021**, *21*, 4094. [\[CrossRef\]](#)
3. Hitchcock, F.L. The expression of a tensor or a polyadic as a sum of products. *J. Math. Phys.* **1927**, *6*, 164–189. [\[CrossRef\]](#)
4. Hitchcock, F.L. Multiple invariants and generalized rank of a p-way matrix or tensor. *J. Math. Phys.* **1927**, *7*, 39–79. [\[CrossRef\]](#)
5. Harshman, R.A. Foundations of the Parafac procedure: Models and conditions for an explanatory multimodal factor analysis. *UCLA Work. Pap. Phon.* **1970**, *16*, 1–84.
6. Carroll, J.D.; Chang, J.J. Analysis of individual differences in multidimensional scaling via an N-way generalization of “Eckart–Young” decomposition. *Psychometrika* **1970**, *35*, 283–319. [\[CrossRef\]](#)
7. Bro, R.; Kiers, H.A.L. A new efficient method for determining the number of components in Parafac models. *J. Chemom.* **2003**, *17*, 274–286. [\[CrossRef\]](#)
8. Harshman, R.A.; Desarbo, W.S. An Application of Parafac to a Small Sample Problem, Demonstrating Preprocessing, Orthogonality Constraints, and Split-Half Diagnostic Techniques. In *Research Methods for Multimode Data Analysis*; Law, H.G., Snyder, C.W., Jr., Hattie, J.A., McDonald, R.P., Eds.; Praeger: New York, NY, USA, 1984; pp. 602–642.
9. Leardi, R. *Multi-Way Analysis with Applications in the Chemical Sciences*, Age Smilde; Bro, R., Geladi, P., Eds.; Wiley: Chichester, UK, 2004.
10. Harshman, R.; Lundy, M.E. Data preprocessing and the extended Parafac model. In *Research Methods for Multimode Data Analysis*; Law, H.G., Snyder, C.W., Jr., Hattie, J.A., McDonald, R.P., Eds.; Praeger: New York, NY, USA, 1984; pp. 216–284.
11. Kroonenberg, P. *Applied Multiway Data Analysis*; Wiley: New York, NY, USA, 2008.
12. Smilde, A.; Geladi, P.; Bro, R. *Multi-Way Analysis: Applications in the Chemical Sciences*; Wiley: New York, NY, USA, 2005.
13. Amigo, J.M.; Marini, F. Multiway methods. In *Data Handling in Science and Technology*; Elsevier: Amsterdam, The Netherlands, 2011; Volume 28, pp. 265–313.
14. Favier, G.; de Almeida, A.L. Overview of constrained Parafac models. *EURASIP J. Adv. Signal Process.* **2014**, *2014*, 142. [\[CrossRef\]](#)
15. Giordani, P.; Kiers, H.; Ferraro, M.D. Three-way component analysis using the R package threeWay. *J. Stat. Softw. Artic.* **2014**, *57*, 1–23.
16. Papalexakis, E.; Faloutsos, C.; Sidiropoulos, N.D. ParCube: Sparse parallelizable tensor decompositions. In *Machine Learning and Knowledge Discovery in Databases*; Springer: Berlin, Germany, 2012; pp. 521–536.
17. Kaya, O.; Ucar, B. Scalable sparse tensor decompositions in distributed memory systems. In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, Austin, TX, USA, 15–20 November 2015; pp. 1–11.
18. Li, J.; Ma, Y.; Wu, X.; Li, A.; Barker, K. PASTA: A parallel sparse tensor algorithm benchmark suite. *CCF Trans. High Perform. Comput.* **2019**, *1*, 111–130. [\[CrossRef\]](#)
19. Kiers, H.A.; Giordani, P. Candecomp/Parafac with zero constraints at arbitrary positions in a loading matrix. *Chemom. Intell. Lab. Syst.* **2020**, *207*, 104145. [\[CrossRef\]](#)
20. Wilderjans, T.F.; Ceulemans, E. Clusterwise Parafac to identify heterogeneity in three-way data. *Chemom. Intell. Lab. Syst.* **2013**, *129*, 87–97. [\[CrossRef\]](#)
21. Cariou, V.; Wilderjans, T.F. Consumer segmentation in multi-attribute product evaluation by means of non-negatively constrained CLV3W. *Food Qual. Prefer.* **2018**, *67*, 18–26. [\[CrossRef\]](#)
22. Vichi, M.; Saporta, G. Clustering and disjoint principal component analysis. *Comput. Stat. Data Anal.* **2009**, *53*, 3194–3208. [\[CrossRef\]](#)
23. Macedo, E.; Freitas, A. The alternating least-squares algorithm for CDPCA. In *EURO Mini-Conference on Optimization in the Natural Sciences*; Plakhov, A., Tchemisova, T., Freitas, A., Eds.; Springer: Cham, Switzerland, 2015; pp. 173–191.
24. Ferrara, C.; Martella, F.; Vichi, M. Dimensions of well-being and their statistical measurements. In *Topics in Theoretical and Applied Statistics*; Springer: Cham, Switzerland, 2016; pp. 85–99.
25. Ramirez-Figueroa, J.A.; Martin-Barreiro, C.; Nieto-Librero, A.B.; Leiva, V.; Galindo-Villardón, M.P. A new principal component analysis by particle swarm optimization with an environmental application for data science. *Stoch. Environ. Res. Risk Assess.* **2021**. [\[CrossRef\]](#)

26. Martin-Barreiro, C.; Ramirez-Figueroa, J.A.; Nieto-Librero, A.B.; Leiva, V.; Martin-Casado, A.; Galindo-Villardón, M.P. A new algorithm for computing disjoint orthogonal components in the three-Way Tucker model. *Mathematics* **2021**, *9*, 203. [[CrossRef](#)]
27. Kolda, T.G.; Bader, B.W. Tensor decompositions and applications. *SIAM Rev.* **2009**, *51*, 455–500. [[CrossRef](#)]
28. Tomasi, G.; Bro, R. A comparison of algorithms for fitting the Parafac model. *Comput. Stat. Data Anal.* **2006**, *50*, 1700–1734. [[CrossRef](#)]
29. Caro-Lopera, F.J.; Leiva, V.; Balakrishnan, N. Connection between the Hadamard and matrix products with an application to matrix-variate Birnbaum–Saunders distributions. *J. Multivar. Anal.* **2012**, *104*, 126–139. [[CrossRef](#)]
30. Lundy, M.E.; Harshman, R.; Kruskal, J. A two stage procedure incorporating good features of both trilinear and quadrilinear models. In *Multiway Data Analysis*; Elsevier: Amsterdam, The Netherlands, 1989; pp. 123–130.
31. Kroonenberg, P.M.; Harshman, R.A.; Murakami, T. Analysing three-way profile data using the Parafac and Tucker3 models illustrated with views on parenting. *Appl. Multivar. Res.* **2009**, *13*, 5–41. [[CrossRef](#)]
32. Harshman, R.; Ladefoged, P.; Goldstein, L. Factor analysis of tongue shapes. *J. Acoust. Soc. Am.* **1977**, *62*, 693–707. [[CrossRef](#)] [[PubMed](#)]
33. Andersen, C.M.; Bro, R. Practical aspects of Parafac modeling of fluorescence excitation-emission data. *J. Chemom.* **2003**, *17*, 200–215. [[CrossRef](#)]
34. Bro, R.; Heimdal, H. Enzymatic browning of vegetables. Calibration and analysis of variance by multiway methods. *Chemom. Intell. Lab. Syst.* **1996**, *34*, 85–102. [[CrossRef](#)]
35. Bro, R. Exploratory study of sugar production using fluorescence spectroscopy and multi-way analysis. *Chemom. Intell. Lab. Syst.* **1999**, *46*, 133–147. [[CrossRef](#)]
36. Leardi, R.; Armanino, C.; Lanteri, S.; Alberotanza, L. Three-mode principal component analysis of monitoring data from Venice lagoon. *J. Chemom.* **2000**, *14*, 187–195. [[CrossRef](#)]