Model-based co-clustering for mixed type data

Margot Selosse[1,3], Julien Jacques[1,3], Christophe Biernacki[2,3,4]
[1]Université de Lyon, [2]CNRS & [3]Inria & [4]Université de Lille

Abstract

Over decades, a lot of studies have shown the importance of clustering to emphasize groups of observations. More recently, due to the emergence of high-dimensional datasets with a huge number of features, co-clustering techniques have emerged and proposed several methods for simultaneously producing groups of observations and features. By synthesizing the dataset in blocks (the crossing of a row-cluster and a column-cluster), this technique can sometimes summarize better the data and its inherent structure. The Latent Block Model (LBM) is a well-known method for performing a co-clustering. However, recently, contexts with features of different types (here called mixed type datasets) are becoming more common. Unfortunately, the LBM is not directly applicable on this kind of dataset. The present work extends the usual LBM to the so-called Multiple Latent Block Model (MLBM) which is able to handle mixed type datasets. The inference is done through a Stochastic EM-algorithm embedding a Gibbs sampler and model selection criterion is defined to choose the number of row and column clusters. This method was successfully used on simulated and real datasets.

Keywords— co-clustering; mixed-type data; latent block model

1 Introduction

Clustering algorithms have become a widely used method due to their ability to provide new insights that may be difficult to perceive for human beings into unlabeled datasets. They consist in forming homogeneous groups of observations that are called clusters. Performing a clustering can therefore highlight an inherent structure of the data. However, the recent "big-data" phenomenon has largely increased the number of features, leading to the emergence of high-dimensional datasets. Clustering techniques are consequently not always sufficient to discern structures. Indeed, the analysis of a cluster relies on a representative of the cluster (mean, mode...). Yet, this latter is itself described by a high number of features, which makes more difficult the interpretation and degrades the synthesis of the dataset. From this consideration comes the need to also "summarize" the features, which can be done by gathering them into clusters, symmetrically to the classical clustering of

observations. Co-clustering methods appear as a good candidate for performing this task because it realizes a joint clustering of rows and columns. Thus, the initial large data matrix can be summarized by a reduced number of blocks, resulting from the crossing of row-clusters and column-clusters.

Among the most famous co-clustering techniques, the Non-negative Matrix Tri-Factorization [10] consists in factorizing the $N \times P$ data matrix $\boldsymbol{x}$ into three matrices $\boldsymbol{a}$ (of size $N \times G$), $\boldsymbol{b}$ (size $G \times H$), $\boldsymbol{c}$ (size $H \times P$), with the property that all three matrices have non-negative elements. More specifically, the approximation of $\boldsymbol{x}$ by $\boldsymbol{x} \approx \boldsymbol{abc}$ is achieved by minimizing the error function $\underset{(\boldsymbol{a},\boldsymbol{b},\boldsymbol{c})}{min} ||\boldsymbol{x} - \boldsymbol{abc}||_F$, with the constraints $(\boldsymbol{a} \geq 0, \boldsymbol{b} \geq 0, \boldsymbol{c} \geq 0)$, meaning that all elements of $\boldsymbol{a}$, $\boldsymbol{b}$ and $\boldsymbol{c}$ are greater than 0, and $||.||_F$ being a matrix norm to be chosen. The matrices $\boldsymbol{a}$ and $\boldsymbol{c}$ play the roles of row and columns cluster memberships respectively. Each value of the matrix $\boldsymbol{a}$ (respectively $\boldsymbol{c}$) corresponds to the degree in which a row (resp. a column) belongs to a row-cluster (resp. a column-cluster). The matrix $\boldsymbol{b}$ represents the *block* matrix: an element b_{gh} of $\boldsymbol{b}$ summarizes the observations belonging to row-cluster g and column-cluster h. Despite the non-negative property of the matrices, it is not always easy to interpret the resulting matrices. For example the matrices $\boldsymbol{a}$ and $\boldsymbol{c}$ are not always normalized which makes difficult the interpretation for the rows and columns memberships to the corresponding clusters. Furthermore, this technique relies on the choice for the distance which is not always obvious. An alternative probabilistic approach considers the Latent Block Model (LBM, [16]). With this model, the elements of a block are modeled by a parametric distribution, which gives more information than a simple scalar, as in the previous methods. Each block is therefore interpretable thanks to the parameters of the block-distribution. Moreover, model selection criterion as the ICL criterion [2] can be used for model selection purpose, including the choice of the number of co-clusters. This technique proved its efficiency for the co-clustering of numerous types of data: continuous [26], nominal [30], binary [20], ordinal [17], functional [31, 7]. That is why an extension of this model is used in the present work, although it is originally not able to take heterogeneous data into account.

Heterogeneous data concern the datasets that are formed with features of different types. For example, in medicine, a patient's file can be made of images (X-rays pictures), text (medical reports), continuous data (age, blood test results...), categorical data (social category, pregnancy, drug addiction...), or even functional data (pulse, blood pressure...). In a clustering context, several frameworks were developed to address this particularity. The latent class model [13] is one of the most used ones. It assumes that the variables are independent conditionally to the row-cluster membership, and consequently, the joint probability distribution function (p.d.f.) of features of different types is obtained by the product of the p.d.f. of each individual feature (see an implementation with the Mixtcomp software [3]). However, when the variables are inherently highly correlated in a row-cluster, this model is not truly adapted. To overcome this matter, the authors of [24] want to conserve standard marginal distributions but also try to loosen the conditional independence on the variables. For this purpose, they use copula, which allow to define on a side the dependence model and on another side the type of the marginal distributions. The proposed model relies on the main hypothesis that each cluster follows a Gaussian copula. However,

the authors note that the model complexity increases with the number of variables, which is sensitive in a big-data context. Another way of addressing the heterogeneous data is to see some variables as the manifestation of a latent vector. For example, in [25], the clustMD model considers continuous and categorical data (nominal and ordinal) and assumes that all the categorical variables are the incarnation of an underlying latent continuous variable. Then, it is assumed that all the continuous variables (observed and not observed) follow a multivariate Gaussian mixture model. Until now, these methods propose models for basic data as categorical (nominal or ordinal) and continuous data. In [8], the authors allow the introduction of more complex data as functional data or networks by projecting the dataset into a reproducing kernel Hilbert space.

However, these techniques were not developed in a co-clustering point of view. To the best of our knowledge, the only work in a co-clustering context is [6], which extends the LBM in the case of a dataset made of continuous and binary data. The present work goes further by proposing an extension that allow to take into account more type of data: categorical data (nominal, ordinal, binary), textual data, counting data and continuous data. Furthermore, the algorithm for inference is able to deal with missing values and proposes a way of imputing them. At last, the ICL criterion [2] is adapted to the proposed model in order to select the number of row-clusters and column-clusters.

The paper is organized as follows. Section 2 gives an overview of the LBM for a good understanding of this paper. Then, it proposes an extension to a new LBM version that allows heterogeneous datasets. Section 3 proposes an algorithm for model inference, based on a Stochastic Expectation Maximization [11] algorithm coupled with a Gibbs sampler. In Section 4, a description of the different types of data that can be taken into account with this method is given, and updates formulas for model inference are presented. Section 5 assesses the efficiency of the proposed method on simulated data while Section 6 shows how the method can perform on real datasets. Section 7 concludes this paper.

2 Multiple Latent Block Model

Here, the Latent Block Model is presented. Then its extension to the Multiple Latent Block Model, which is introduced in the present work, is detailed.

2.1 Latent Block Model

The LBM is a widely used model for performing co-clustering [16]. Basically, it assumes that all the elements of a block follow the same distribution. In this section, the hypotheses for the LBM are defined, and the mathematical details are given.

The LBM considers that all features can potentially be grouped together (restrictions will be imposed in the next section to defined the so-called Multiple LBM). Let consider the data matrix $\boldsymbol{x} = (x_{ij})_{i,j}$, where $i \in \{1, \dots, N\}$ is the row (observation) index and $j \in \{1, \dots, J\}$ is the column (feature) index. It is assumed that there exists G row-clusters and H column-clusters that correspond to a partition $\boldsymbol{v} = (v_{ig})_{i,g}$ of the rows and a partition $\boldsymbol{w} = (w_{jh})_{j,h}$ of the columns, with $1 \leq g \leq G$ and $1 \leq h \leq H$, where v_{ig} is equal

to 1 if row i belongs to cluster g, and 0 otherwise; and similarly w_{jh} is equal to 1 when column j belongs to cluster h, and 0 otherwise. In order to simplify the notations, the underlying range of variation will be omitted in the sums and products, which consequently will be written $\sum_i$, $\sum_j$, $\sum_g$ and $\sum_h$, $\prod_i$, $\prod_j$, $\prod_g$ and $\prod_h$.

The first LBM hypothesis is that the univariate random variables x_{ij} are assumed to be conditionally independent given the row and column partitions $\boldsymbol{v}$ and $\boldsymbol{w}$. Therefore, the conditional probability density function of $\boldsymbol{x}$ given $\boldsymbol{v}$ and $\boldsymbol{w}$ can be written:

$$p(\boldsymbol{x}|\boldsymbol{v},\boldsymbol{w};\boldsymbol{\alpha}) \;=\; \prod_{i,j,g,h} p(x_{ij};\alpha_{gh})^{v_{ig}w_{jh}},$$

where $\boldsymbol{\alpha} = (\alpha_{gh})_{g,h}$ is the distribution's parameters of block (g,h).

The second LBM hypothesis is that the latent variables $\boldsymbol{v}$ and $\boldsymbol{w}$ are independent so $p(\boldsymbol{v},\boldsymbol{w};\boldsymbol{\gamma},\boldsymbol{\rho}) \;=\; p(\boldsymbol{v};\boldsymbol{\gamma})p(\boldsymbol{w};\boldsymbol{\rho})$ with:

$$p(\boldsymbol{v};\boldsymbol{\gamma}) \;=\; \prod_{i,g}\gamma_g^{v_{ig}} \;\text{ and }\; p(\boldsymbol{w};\boldsymbol{\rho}) \;=\; \prod_{j,h}\rho_h^{w_{jh}},$$

where $\gamma_g = p(v_{ig}=1)$ and $\rho_h = p(w_{jh}=1)$. This implies that, for all i, the distribution of $\boldsymbol{v}_i$ is the multinomial distribution $\mathcal{M}(\gamma_1,\ldots,\gamma_G)$ and does not depend on i. In a similar way, for all j, the distribution of $\boldsymbol{w}_j$ is the multinomial distribution $\mathcal{M}(\rho_1,\ldots,\rho_H)$ and does not depend on j.

From these considerations, the LBM parameter is defined as $\boldsymbol{\theta} = (\boldsymbol{\gamma},\boldsymbol{\rho},\boldsymbol{\alpha})$, with $\boldsymbol{\gamma} = (\gamma_1,\ldots,\gamma_G)$ and $\boldsymbol{\rho} = (\rho_1,\ldots,\rho_H)$ the rows and columns mixing proportions. Therefore, if V and W are the sets of all possible labels $\boldsymbol{v}$ and $\boldsymbol{w}$, the probability density function of $\boldsymbol{x}$ can be written:

$$p(\boldsymbol{x};\boldsymbol{\theta}) \;=\; \sum_{(\boldsymbol{v},\boldsymbol{w})\in V\times W} \prod_{i,g}\gamma_g^{v_{ig}} \prod_{j,h}\rho_h^{w_{jh}} \prod_{i,j,g,h} p(x_{ij};\alpha_{gh})^{v_{ig}w_{jh}}. \tag{1}$$

2.2 Extension to Multiple Latent Block Model

Now, let consider that the matrix $\boldsymbol{x}$ is made of D different sets of features. In this context, it has N rows and $J = \sum_{d=1}^{D} J_d$ columns, J_d being the number of features of the d-th set:

$$\boldsymbol{x} = (\boldsymbol{x}^1,...,\boldsymbol{x}^D), \text{ with } \boldsymbol{x}^d = (x_{ij}^d)_{i=1,...,N;\; j=1,...,J_d}.$$

Here, the idea of "sets" of features is introduced to define the features we potentially want to group together in a column-cluster, and those we do not want to be together. Thus, features of a same set can be grouped together in an *intra-set* column-cluster whereas features of different sets can not. The motives for separating features into different sets are twofold: a technical one and a semantic one. First, two features of different types (*e.g.* a categorical feature and a continuous one) can not be modeled with a similar probability distribution, and consequently should technically be separated. Indeed, since it will be assumed afterwards that all the features into a column-cluster have the same p.d.f., such assumption is not be possible for features of different types. The present work is motivated by this first reason. Secondly, the user could consider, for practical reasons, that some features necessarily have to be separated because it does not make sense to gather them in

a same column cluster. This case is not explored in the present work, but the reader can for instance refer to [29] for a detailed example. Besides, the sets of elements regarding $d \in \{1, \ldots, D\}$ will not be entirely set forth: for example $(\boldsymbol{x}^1, \ldots, \boldsymbol{x}^D)$ will be annotated $(\boldsymbol{x}^d)_d$.

Being in a co-clustering context, it is assumed that there exist G row-clusters and $H = H_1 + \ldots + H_D$ column-clusters inherent to the matrix $\boldsymbol{x}$. Moreover, the sums and the products relating to sets the of features will be subscripted by the letter d. Again, he underlying range of variation will be omitted in the sums and products, which consequently will be written $\sum_d$ and $\prod_d$.

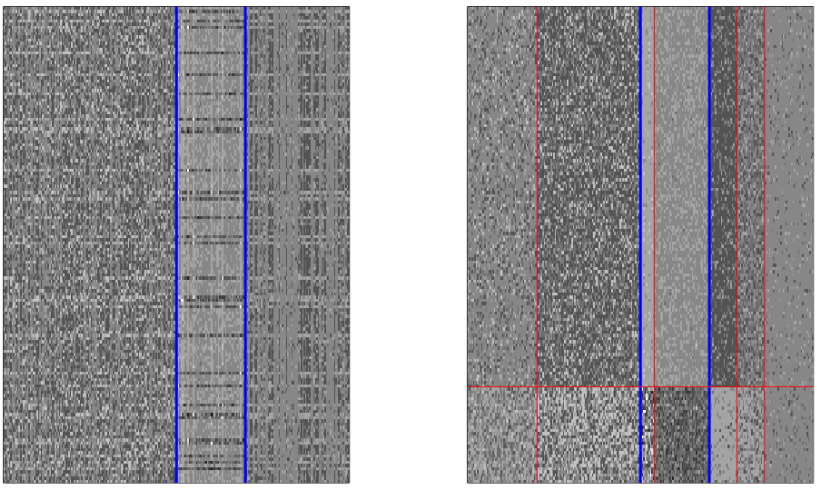
Finally, a dataset can contain missing data. To deal with this aspect, the d^{th} matrix $\boldsymbol{x}^d$ is said to be made with two sets $\check{\boldsymbol{x}}^d$ and $\hat{\boldsymbol{x}}^d$. $\check{\boldsymbol{x}}^d$ is the observed data, and $\hat{\boldsymbol{x}}^d$ is the missing data. An element of $\boldsymbol{x}^d$ will be annotated $\check{x}^d_{ij}$ whether x^d_{ij} is observed, and $\hat{x}^d_{ij}$ otherwise. To model missing values, three main processes exist in data analysis [21]. The Missing Completely At Random (MCAR) process assumes that the missing data mechanism is unrelated to the values of any variables: for example, in a survey, participants accidentally skipped questions. The Missing At Random (MAR) process supposes that a missing value has nothing to do with the concerned variable, but it does have to do with the values of some other variable. As an example, males are less likely to fill in a depression survey but this has nothing to do with their level of depression, after accounting for maleness. The last process is called Missing Not At Random (MNAR) and occurs when the missing value is directly influenced by the variable itself. As an example, a drug addict would not answer to a question about drugs precisely because of their addiction. In the present work, it is assumed that the whole missing process is MAR, because this is the most frequent situation we encountered in practice.

In the case of heterogeneous data, the LBM as it was described can not be used. Indeed, it relies on the hypothesis that the block's elements are the realizations of a random variable that follows a distribution with parameter $\boldsymbol{\alpha}$. If the elements of the blocks are not of the same type, it is not possible to consider that they were sampled from the same distribution. The Multiple Latent Block Model (MLBM) introduced in [27] and generalized in this paper, offers a way of addressing this issue. In this model, $D \geq 1$, and the columns of the matrix $\boldsymbol{x}$ are reordered such that $\boldsymbol{x}$ is composed by D matrices put side by side, each matrix containing features of homogeneous type as described above. The co-clustering is performed so that features of different types can not be part of a same column-cluster. Consequently, it is possible to define a distribution on each block because it is made of variables of the same type. Figure 1 illustrates the intuition behind this model.

Let $\boldsymbol{w}^d$ denote the column partitions of the d-th matrix ($1 \leq d \leq D$), $\boldsymbol{\rho}^d = (\rho^d_1, \ldots, \rho^d_{H_d})$ the corresponding mixing proportions, and let introduce the notations $\boldsymbol{w} = (\boldsymbol{w}^d)_d$ and $\boldsymbol{\rho} = (\boldsymbol{\rho}^d)_d$.

The MLBM relies on the following hypothesis, which states that the D matrices data are independent conditionally on the row and column partitions, and that the d^{th} matrix does not depend on the column partitions different from $\boldsymbol{w}^d$:

$$p(\boldsymbol{x}|\boldsymbol{v}, \boldsymbol{w}) = p(\boldsymbol{x}^1|\boldsymbol{v}, \boldsymbol{w}^1) \times \ldots \times p(\boldsymbol{x}^D|\boldsymbol{v}, \boldsymbol{w}^D).$$



(a) (b)

Figure 1: (a) is the matrix $\boldsymbol{x}$. The blue lines represents the separation of the features that are not of same type. (b) is the matrix after having performed a co-clustering. The red lines represents the co-clusters limits.

The other assumptions of the MLBM are similar to those of the LBM. First, the univariate random variables x_{ij}^d are supposed to be independent conditionally on partitions $\boldsymbol{v}$ and $\boldsymbol{w}^d$. Thus, the conditional probability function of $\boldsymbol{x}$ given $\boldsymbol{v}$ and $(\boldsymbol{w}^d)_d$ is expressed as:

$$p(\boldsymbol{x}|\boldsymbol{v},\boldsymbol{w};\boldsymbol{\alpha}) = \prod_{i,j,g,h,d} p(x_{ij}^d;\alpha_{gh}^d)^{v_{ig}w_{jh}^d},$$

where $\boldsymbol{\alpha} = (\boldsymbol{\alpha}^d)_d$ with $\boldsymbol{\alpha}^d = (\alpha_{gh}^d)_{g,h}$ is the distribution's parameters of block (g,h) of matrix $\boldsymbol{x}^d$.

Second, the latent variables $\boldsymbol{v}, \boldsymbol{w}^1, \ldots, \boldsymbol{w}^D$ are assumed to be independent, so: $p(\boldsymbol{v},\boldsymbol{w};\boldsymbol{\gamma},\boldsymbol{\rho}) = p(\boldsymbol{v};\boldsymbol{\gamma})\prod_d p(\boldsymbol{w}^d;\boldsymbol{\rho}^d)$, where:

$$p(\boldsymbol{v};\boldsymbol{\gamma}) = \prod_{i,g}\gamma_g^{v_{ig}} \text{ and } p(\boldsymbol{w}^d;\boldsymbol{\rho}^d) = \prod_{j,h}{\rho_h^d}^{w_{jh}^d}.$$

The MLBM parameter is thus defined as $\boldsymbol{\theta} = (\boldsymbol{\gamma},\boldsymbol{\rho},\boldsymbol{\alpha})$. Moreover, if V and $(W^d)_d$ are the sets of all possible labels $\boldsymbol{v}$ and $(\boldsymbol{w}^d)_d$, the probability density function $p(\boldsymbol{x};\boldsymbol{\theta})$ can be written:

$$p(\boldsymbol{x};\boldsymbol{\theta}) = \sum_{(\boldsymbol{v},(\boldsymbol{w}^d)_d)\in V\times(W^d)_d} \prod_{i,g}\gamma_g^{v_{ig}} \prod_d\prod_{j,h}{\rho_h^d}^{w_{jh}^d} \prod_{i,j,g,h} p(x_{ij}^d;\alpha_{gh}^d)^{v_{ig}w_{jh}^d}. \quad (2)$$

Let remark that until now the type of $p(x_{ij}^d;\alpha_{gh}^d)$ has not be defined. It will be in Section 4, depending on the type of x_{ij}^d (nominal, ordinal, continuous, ...).

3 Model Inference

The MLBM inference aims at estimating $\boldsymbol{\theta}$ that maximizes the observed log-likelihood:

$$l(\boldsymbol{\theta}; \check{\boldsymbol{x}}) = \sum_{\hat{\boldsymbol{x}}} \log p(\boldsymbol{x}; \boldsymbol{\theta}). \tag{3}$$

The EM-algorithm [12] is a well known method for performing this task with latent variables. Regarding the co-clustering case, it is not computationally tractable, though. Indeed, this method needs to compute the expectation of the complete data log-likelihood. Yet, in the case with $D = 1$, this expression contains the probability $p(v_{ig} = 1, w_{jh} = 1|\boldsymbol{x}; \boldsymbol{\theta})$, which needs to consider all the possible values for $\boldsymbol{v}_{i'}$ and $\boldsymbol{w}_{j'}$ with $i' \neq i$ and $j' \neq j$. The E-step would require to calculate $G^N \times H^J$ terms: as an example, if $G = 2$, $H = 2$, $N = 20$ and $J = 20$, each E step of the EM algorithm would need to compute $2^{20} \times 2^{20} \approx 10^{12}$ terms. There exist different alternatives to the EM algorithm as the variational EM algorithm, the SEM-Gibbs algorithm or other algorithm linked to a Bayesian inference [16]. The SEM-Gibbs version is used in the present work since while it is know to be less sensitive to initialization, it is also simple to implement. Furthermore, it handles easily missing values $\hat{\boldsymbol{x}}$ in $\boldsymbol{x}$, which is an important advantage, particularly with real datasets.

3.1 SEM-Gibbs algorithm

The SEM-Gibbs begins with an initialization of partitions, parameters and missing values $\boldsymbol{v}^{(0)}, \boldsymbol{w}^{(0)}, \boldsymbol{\theta}^{(0)}, \hat{\boldsymbol{x}}^{(0)}$. This initialization process is described further. The following five steps describe the q-th iteration, with $q \in (1, ..., nbSEM)$. The choice for $nbSEM$ will be described afterwards as well.

(a) Sampling row partitions. Generate the row partitions with

$$p(v_{ig}^{(q)} = 1 \mid \boldsymbol{x}, \boldsymbol{w}^{(q-1)}; \boldsymbol{\theta}^{(q-1)}) \propto \gamma_g^{(q-1)} \times \prod_d t_g^d(\boldsymbol{x}_{i.}^d | \boldsymbol{w}^{d^{(q-1)}}; \boldsymbol{\alpha}^{d^{(q-1)}}), \tag{4}$$

where $t_g^d(\boldsymbol{x}_{i.}^d | \boldsymbol{w}^{d^{(q-1)}}; \boldsymbol{\alpha}^{d^{(q-1)}}) = \prod_{j,h} f(x_{ij}^d; {\alpha_{gh}}^{d^{(q-1)}})^{w_{jh}^{d\ (q-1)}}$ with $\boldsymbol{x}_{i.}^d = (x_{ij}^d)_j$.
Let note that this probability depends on the type of the data of the d-th matrix through the p.d.f $f(x_{ij}^d; {\alpha_{gh}}^{d^{(q-1)}})$, whose exact expression will be given in Section 4.

(b) First M-step This first M-step consists in updating the co-clusters parameters $\boldsymbol{\theta}^{(q)}$ to maximize the completed log-likelihood (3). The row mixing proportions are consequently updated by:

$$\gamma_g^{(q)} = \tfrac{1}{N} \sum_i v_{ig}^{(q)},$$

and the parameter $\boldsymbol{\alpha}^{d^{(q)}}$ is updated as well, yet the computations depend on the type of matrix $\boldsymbol{x}$ features. Section 4 describes how to update $\boldsymbol{\alpha}^{d^{(q)}}$ according to the type of the variables.

(c) Sampling column partitions. For all $d \in \{1, \ldots, D\}$ generate the column partitions for the d-th matrix $\boldsymbol{x}^d$ with

$$p(w_{jh}^{d\ (q)} = 1 \mid \boldsymbol{x}^d, \boldsymbol{v}^{(q)}; \boldsymbol{\theta}^{(q)}) \propto \rho_h^{d(q)} \times s_h^d(\boldsymbol{x}_{.j}^d | \boldsymbol{v}^{(q)}; \boldsymbol{\alpha}^{d^{(q-1)}}), \tag{5}$$

where $s_h^d(\boldsymbol{x}_{.j}^d | \boldsymbol{v}^{(q)}; \boldsymbol{\alpha}^{d^{(q)}}) = \prod_{i,g} f(x_{ij}^d; {\alpha_{gh}}^{d^{(q-1)}})^{v_{ig}^{(q)}}$ with $\boldsymbol{x}_{.j}^d = (x_{ij}^d)_i$.
Here, note that s_h^d obviously depends on the type of the d-th matrix (see Section 4).

(d) Second M-step In this second M-step, the column mixing proportions are updated by:

$$\rho_h^{d(q)} = \tfrac{1}{J_d} \sum_j {w_{jh}^d}^{(q)},$$

and the parameter $\boldsymbol{\alpha}^{d^{(q)}}$ is also updated depending on the type of the d-th matrix (see Section 4).

(e) Missing values imputation. Generate the missing data $\hat{x}_{ij}^{d^{(q)}}$ according to:

$$p(\hat{x}_{ij}^{d^{(q)}} | \check{\boldsymbol{x}}, \boldsymbol{v}^{(q)}, \boldsymbol{w}^{d^{(q)}}; \boldsymbol{\theta}^{(q)}) = \prod_{g,h} f(\hat{x}_{ij}^{d^{(q)}}; \boldsymbol{\alpha}^{d^{(q)}})^{v_{ig}^{(q)} {w_{jh}^d}^{(q)}}.$$

The SEM-Gibbs algorithm is iterated for a given number of iterations. The first part of these iterations are called the burn-in period, meaning that the parameters $\boldsymbol{\theta}$ are not simulated yet according to its stationary distribution. Consequently, only the iterations that occurred after this burn-in period are taken into account and are referred as sampling distribution hereafter. While the final estimations of discrete parameters is the mode of the sampling distribution, the final estimations of the continuous parameters are the median of the sample distribution. It leads to a final estimation of $\boldsymbol{\theta}$ called $\hat{\boldsymbol{\theta}}$. Then, a sample of $(\hat{\boldsymbol{x}}, \boldsymbol{v}, \boldsymbol{w})$ is simulated by iterating steps (a), (c) and (e) of the SEM-Gibbs algorithm with $\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}$. The final partitions $(\hat{\boldsymbol{v}}, \hat{\boldsymbol{w}})$ and the missing observations $\hat{\boldsymbol{x}}$ are estimated by the mode of their marginal sampled distribution.

Choice for the number of iterations The SEM-algorithm can be slow at reaching its stationary state. After having arbitrarily chosen the total number of iterations, the stability of the algorithm has to be checked. To accomplish that, the evolution of the parameters through the iterations can be simply graphically analyzed. If the parameters are "stable" between the burn-in period and the last iteration then the number of iterations was well chosen. Less subjective ways exist for evaluating if the stationary distribution has been achieved. The authors of [14] propose a general approach to monitoring convergence of MCMC output in which parallel chains are run with starting values that are spread relatively to the posterior distribution. Convergence is confirmed when the output from all chains is indistinguishable. This method is not used in this paper but could have been. Indeed, in Section 5, we show that we can obtain satisfying results without this technique.

Initialization The algorithm starts with an initialization of the partitions. Then the mixing proportions and the block parameters are estimated regarding these partitions. In the case of $D = 1$, this initialization can be made randomly [9], but with $D > 1$, it often leads to empty the clusters. In this work, a specific initialization strategy was worked out to tackle this issue. It relies on a first random initialization. However, for the first I iterations (such that I is lower or equal to the burn-in number of iterations), whenever a row-cluster gets empty, a percentage of the row partitions is sampled from the Multinomial distribution $\mathcal{M}(1/G, \ldots, 1/G)$. Similarly when a column-cluster gets empty on the d^{th} matrix, a percentage of the column partitions is sampled from the Multinomial distribution $\mathcal{M}(1/H_d, \ldots, 1/H_d)$.

3.2 Model Selection

To select the number of blocks $(G, H_1, \ldots, H_D)$, a model selection criterion must be involved. The most classical ones, like BIC [28], rely on penalizing the maximum log-likelihood value $l(\hat{\boldsymbol{\theta}}; \check{\boldsymbol{x}})$. However, due to the dependency structure of the observed data $\check{\boldsymbol{x}}$, this value is not available.

Alternatively, an approximation of the ICL information criterion [2], called here ICL-BIC, can be invoked to overcome the previous problem due to the dependency structure in the missing variables $(\check{\boldsymbol{x}}, \boldsymbol{v}, \boldsymbol{w})$. The key point is that this latter vanishes since ICL relies on the completed latent block information $(\boldsymbol{v}, \boldsymbol{w})$, instead of integrating on it as it is the case in BIC. In particular, [19] detailed how to express ICL-BIC for the general case of categorical data. It is possible to straightforwardly transpose the ICL-BIC expression given by these authors by following step by step their piece of work, with no new technical material. The resulting MLBM-specific ICL-BIC is expressed by:

$$\text{ICL-BIC}(G, H_1, \ldots, H_D) =$$
$$\log p(\check{\boldsymbol{x}}, \hat{\boldsymbol{v}}, \hat{\boldsymbol{w}}; \hat{\boldsymbol{\theta}}) - \tfrac{1}{2}(G-1)\log N - \sum_d \tfrac{1}{2}(H_d - 1)\log J_d - \sum_d \tfrac{1}{2}\nu_d \log(N \times J_d),$$

where ν_d is the number of parameters to estimate for the d-th matrix $\boldsymbol{x}^d$. It will depend on G, H_d and the type of the variables of $\boldsymbol{x}^d$. Table 1 in Section 4 gives ν_d for each type of distribution.

In theory, to find the best number of blocks $(G, H_1, \ldots, H_D)$, the co-clustering has to be executed for each possible value and the result with the highest ICL-BIC has to be retained. Let n_G be the number of candidate values for G, while n_{H_d} is the number of candidate values for H_d, $d \in \{1, \ldots, D\}$. Thus, the number of co-clustering to execute is $n_G \times n_{H_1} \times \ldots \times n_{H_D}$. As an example, if $D = 3$ and the user wants to try 10 values for G and for each H_d, then it would require to execute 10^4 co-clusterings. Depending on the dataset, it might take too much time to find the best solution. In practice, a good set $(G, H_1, \ldots, H_D)$ is searched through the following heuristic. Let (G_{min}) be the minimum of the candidate values for G. Then, $(H_{d_{min}})_d$ is the minimum of the candidate values for $(H_d)_d$. The algorithm starts with the set $(G_{min}, H_{1_{min}}, \ldots, H_{D_{min}})$. At iteration p, the current best set $(G, H_1, \ldots, H_D)$ is called $(G, H_1, \ldots, H_D)^{(p)}$ and is made of values: $(G^{(p)}, H_1^{(p)}, \ldots, H_D^{(p)})$. At p^{th} iteration, $(D+1)$ co-clusterings are realized with sets $(G^{(p)} + 1, H_1^{(p)}, \ldots, H_D^{(p)})$,

$(G^{(p)}, H^{(p)}+1, \ldots, H_D^{(p)})$,...,$(G^{(p)}, H_1^{(p)}, \ldots, H_D^{(p)}+1)$. Then, the ICL-BIC is computed for each result. If none of the ICL-BIC is better than for the set $(G, H_1, \ldots, H_D)^{(p)}$, the algorithm finishes and $(G, H_1, \ldots, H_D)^{(p)}$ is the set to use. In the other case, the set with the highest ICL-BIC is retained, and becomes $(G, H_1, \ldots, H_D)^{(p+1)}$. The algorithm afterwards reiterates the same steps.

4 Modeling of the different types of data

Representing the data as a mathematical object is challenging and requires to make a compromise. Often the user has to find a trade-off between information loss, interpretability and feasibility for their representation. The present work allows to work with the following type of data: categorical data (nominal, ordinal, binary), counting data, continuous data and textual data. If the probability distributions for nominal (multinomial), binary (Bernoulli), counting (Poisson) and continuous (Gaussian) data are commonly accepted, there exist several ways to model textual and ordinal data.

The most simple way to represent textual data is as a Document-Term counting matrix where a cell simply counts how many times a term appears in a document. The Poisson distribution is a good distribution for modeling this matrix because it models the occurrences of an event (in this case, the appearances of a word). In a bit more advanced way, the Document-Term TF-IDF matrix, counts the times a term appears, but penalizes the result if this same term appears in the other documents [18]. The resulting score is a continuous numeric which implies the usage of the Gaussian distribution. In the latter, the so-called "stop-words" terms are therefore discarded, and the attention of the user is not diverted by terms that do not really worth it. There exists also a lot of other Document-Term matrix types, and they have proven their efficiency in many applications [1, 20]. In this work, a simple Document-Term matrix representation is considered.

The ordinal data is also a sensitive type of data. It can seems very easy to model them as if they were nominal, but it spoils the order between the different levels, which is an intrinsic property of this type of data. In some applications, it can be interpreted as continuous [22] but in other cases it is not an option. For example, for clinical surveys, psychologists sometimes spend years to define ordinal scales on abstract concepts like pain, perception of control or anxiety [23, 32]; it is therefore delicate to project their results on other scales or in a continuous space. In the present work particularly, a recent distribution for ordinal data (BOS for Binary Ordinal Search model, [4]) is used. It has proven its efficiency for modeling and clustering ordinal data. One of the main advantage of the BOS model is its parsimony and the significance of its parameters.

This section describes the expression of the p.d.f. $f(x_{ij}^d; \boldsymbol{\alpha}^{d^{(q-1)}})$ and the update of $\boldsymbol{\alpha}^{d^{(q-1)}}$, involved in the SEM-Gibbs algorithm, depending on the type of the matrix $\boldsymbol{x}^d$. The superscripts (q) and (d) are omitted to simplify the expressions.

4.1 Modeling nominal data

A nominal variable is a variable that can take on one of a limited, fixed, number of possible values. Each of the possible values of a categorical variable is referred to as a level. For a block (g,h) of nominal data, we consider the Multinomial distribution $\mathcal{M}(1, \boldsymbol{\beta}_{gh})$, where $\boldsymbol{\beta}_{gh} = (\beta^r_{gh})_{r=1,\dots,m}$, and $\sum_r^m \beta^r_{gh} = 1$. Therefore, with this type of data, the MLBM block parameter α_{gh} is quoted as $\boldsymbol{\beta}_{gh}$, and the p.d.f. is given by:

$$f(x_{ij}; \boldsymbol{\beta}_{gh}) = \prod_r^m (\beta^r_{gh})^{I(x_{ij}=r)},$$

with $I(x_{ij} = r) = 1$ if $x_{ij} = r$, and 0 otherwise. The update of each β^r_{gh} is:

$$\beta^r_{gh} = \tfrac{1}{n_{gh}} \sum_{i,j} v_{ig} w_{jh} I(x_{ij} = 1),$$

with n_{gh} being the number of elements belonging to block (g, h).

In a first place, let note that if two nominal variables do not have the same number of levels m, then their distribution are not defined on the same support. Consequently, such variables should be separated in different matrices $\boldsymbol{x}^d$ of $\boldsymbol{x}$. In a second place, the co-clustering we propose is dependent on the order of the levels. For example two categorical features with $m = 3$ levels having respective parameters $\boldsymbol{\beta} = (0.1, 0.7, 0.2)$ and $\boldsymbol{\beta} = (0.7, 0.2, 0.1)$ won't be detected as two variables following the same distribution. Consequently they won't be grouped together in a similar column cluster, whereas a simple switch in the levels' order could change this and lead to group these variables together. Let remark that this problem is not specific to co-clustering and is also present in clustering [5]. While the user is aware that the results are conditional to the levels encoding, it is not an issue addressed in this work.

4.2 Modeling ordinal data

Ordinal data is a special case of nominal data, where the order between the levels has a meaning. In the present work, the BOS model [4] is chosen to model ordinal data. It is a probability distribution parametrized by a position parameter $\mu_{gh} \in \{1, \dots, m\}$ and a precision parameter $\pi_{gh} \in [0, 1]$. This distribution has interesting properties from an interpretation point of view: it rises from the uniform distribution when $\pi_{gh} = 0$ to a more peaked distribution around the mode μ_{gh} when π_{gh} grows, and reaches a Dirac distribution at the mode μ_{gh} when $\pi_{gh} = 1$. It is shown in [4] that the BOS distribution is a polynomial function of π_{gh} with degree $m-1$ whose coefficients depend on the position parameter μ_{gh}.

Therefore, with this type of data, the MLBM block parameter α_{gh} is quoted as (μ_{gh}, π_{gh}), and the p.d.f. is given by:

$$f(x_{ij}; \mu_{gh}, \pi_{gh}) = \sum_{r=0}^{m-1} C_r(\mu_{gh}, x_{ij}) \pi^r_{gh},$$

with $C_r(\mu_{gh}, x_{ij})$ a constant depending on μ_{gh} and x_{ij}.

Since BOS inference relies on an EM algorithm, the update of parameter (μ_{gh}, π_{gh}) is obtained through an EM-algorithm. For further details on this algorithm, see [4]. Similarly to nominal variables case, if two ordinal variables do not have the same number of levels, they have to be separated in different matrices $\boldsymbol{x}^d$ of $\boldsymbol{x}$.

4.3 Modeling continuous data

In the continuous case, the unidimensional Gaussian distribution $\mathcal{N}(\mu_{gh}, \sigma^2_{gh})$ is considered. Thus, the MLBM block parameter α_{gh} is here (μ_{gh}, σ_{gh}) and the p.d.f. is given by:

$$f(x_{ij}; \mu_{gh}, \sigma_{gh}) = \exp\{\tfrac{-1}{2\sigma^2_{gh}}(x_{ij} - \mu_{gh})^2\}/\sqrt{2\pi\sigma^2_{gh}}.$$

The update of parameters $(\mu_{gh}, \sigma^2_{gh})$ is:

$$\mu_{gh} = \frac{1}{n_{gh}} \sum_{i,j} v_{ig} w_{jh} x_{ij} \quad \text{and} \quad \sigma^2_{gh} = \frac{1}{n_{gh}} \sum_{i,j} v_{ig} w_{jh} (x_{ij} - \mu_{gh})^2.$$

4.4 Modeling counting data

Counting variables are modeled by the Poisson distribution. For a block (g,h) of counting data, a Poisson distribution with a specific parametrization is considered: $\mathcal{P}(n_{i.} n_{.j} \delta_{gh})$, where $n_{i.} = \sum_j x_{ij}$ and $n_{.j} = \sum_i x_{ij}$ are respectively the number of terms in document i and the number of observations of term j in all documents. The parameters $n_{i.}$ and $n_{.j}$ are independent of the co-clustering and are consequently preliminary estimated from the document term matrix. Consequently, the MLBM parameter α_{gh} are only the parameter δ_{gh}, which is the effect of the block (g, h) [15]. The p.d.f. is given by:

$$f(x_{ij}; \delta_{gh}) = \tfrac{1}{x_{ij}!} e^{-n_{i.} n_{.j} \delta_{gh}} (n_{i.} n_{.j} \delta_{gh})^{x_{ij}}.$$

The update of each parameter δ_{gh} is obtained by:

$$\delta_{gh} = \frac{1}{n_{g.} n_{.h}} \sum_{i,j} v_{ig} w_{jh} x_{ij},$$

with $n_{g.} = \sum_{i,j} v_{ig} x_{ij}$ and $n_{.h} = \sum_{i,j} w_{jh} x_{ij}$.
Finally, Table 1 summarizes the number of parameters ν for each type of data described above.

Table 1: Number of parameters (ν) of the distributions properties

Data type	Distribution	α_{gh}	ν
Nominal	Multinomial	$\boldsymbol{\beta}_{gh} = (\beta^r_{gh})_{r=1,\ldots,m}$	$(m-1)GH$
Ordinal	BOS	(μ_{gh}, π_{gh})	$2GH$
Continuous	Gaussian	(μ_{gh}, σ_{gh})	$2GH$
Count	Poisson	$(\mu_i, \nu_j, \delta_{gh})$	GH

5 Numerical experiments on artificial data

This section has two goals. The first one is to illustrate the efficiency of the proposed inference algorithm for parameter estimation. The second one is to evaluate the model selection strategy: the efficiency of the ICL-BIC criterion to select the true numbers of clusters and the ability of the heuristic search to sparsely explore the space of numbers of clusters.

5.1 Simulation settings

Two simulations settings are considered. While they both have the same parameters, the first one is built such that ($N = J_1 = J_2 = J_3 = J_4 = 100$), and the second one is built with ($N = J_1 = J_2 = J_3 = J_4 = 500$).

Parameters setup Both settings were simulated with four types of distribution: nominal (with $m = 5$ levels), continuous, ordinal (with $m = 3$ levels), and counting data. The number of blocks were fixed to $(G, H_1, H_2, H_3, H_4) = (3, 3, 3, 3, 3)$. Furthermore, the mixing row proportions were $\boldsymbol{\gamma} = (0.2, 0.3, 0.5)$ and the mixing column proportions were equal to: $\boldsymbol{\rho}_1 = (0.25, 0.3, 0.45)$, $\boldsymbol{\rho}_2 = (0.2, 0.35, 0.45)$, $\boldsymbol{\rho}_3 = (0.25, 0.35, 0.4)$, $\boldsymbol{\rho}_4 = (0.25, 0.35, 0.4)$. Table 2 details the parameters that were assigned to each block.

Table 2: Value of the blocks' parameters. For the Count data, parameters are not equal between the first and second simulation because it depends on the margins.

	Nominal $m = 5$ $\beta_1, \beta_2, \beta_3, \beta_4, \beta_5$		
	col-cluster 1	col-cluster 2	col-cluster 3
row-cluster 1	0.05,0.05,0.8,0.05,0.05	0.1,0.25,0.3,0.3,0.05	0.1,0.2,0.4,0.2,0.1
row-cluster 2	0.05,0.1,0.7,0.1,0.05	0.8,0.05,0.05,0.05,0.05	0.4,0.05,0.1,0.05,0.4
row-cluster 3	0.2,0.5,0.2,0.05,0.05	0.8,0.05,0.05,0.05,0.05	0.05,0.8,0.05,0.05,0.05

	Continuous μ, σ			Ordinal $m = 5$ μ, π		
	col-cluster 1	col-cluster 2	col-cluster 3	col-cluster 1	col-cluster 2	col-cluster 3
row-cluster 1	100,1	0.5,5	-90,5	3,0.4	1,0.2	3,0.7
row-cluster 2	10,4	-15,1	-95,1	2,0.1	3,0.5	2,0.8
row-cluster 3	-20,1	-30,3	500,4	2,0.5	1,0.8	2,0.2

	Count 100 $\delta \times 10^{-5}$			Count 500 $\delta \times 10^{-7}$		
	col-cluster 1	col-cluster 2	col-cluster 3	col-cluster 1	col-cluster 2	col-cluster 3
row-cluster 1	1.2	5.5	1.2	4.6	20.5	4.9
row-cluster 2	8.3	5.5	0.5	30.0	20.5	1.6
row-cluster 3	1.3	1.3	3.5	5.5	5.6	14.5

Experimental setup For both settings, 20 datasets are simulated, and the same process is run through. First, the co-clustering is performed on the 20 datasets with the true numbers of clusters $(G_1, H_1, \ldots, H_D)$ and the correctness of the parameters estimation is evaluated. Then, we assess the efficiency of the ICL-BIC criteria by using an exhaustive research among possible values for the number of clusters. In order to reduce the number of ICL-BIC to compute, we consider only the number of clusters obtained by adding or removing one to each of the element of the true $(G, H_1, \ldots, H_D)$. Therefore, for each simulation, $3^5 = 243$ co-clusterings are executed, because 3 values are tested for G,H_1,H_2,H_3 and H_4. Then, the set $(G, H_1, \ldots, H_D)$ with the best ICL-BIC value is retained. Afterward, the heuristic search from Section 3.2 is evaluated. In this case, the number of co-clusterings to

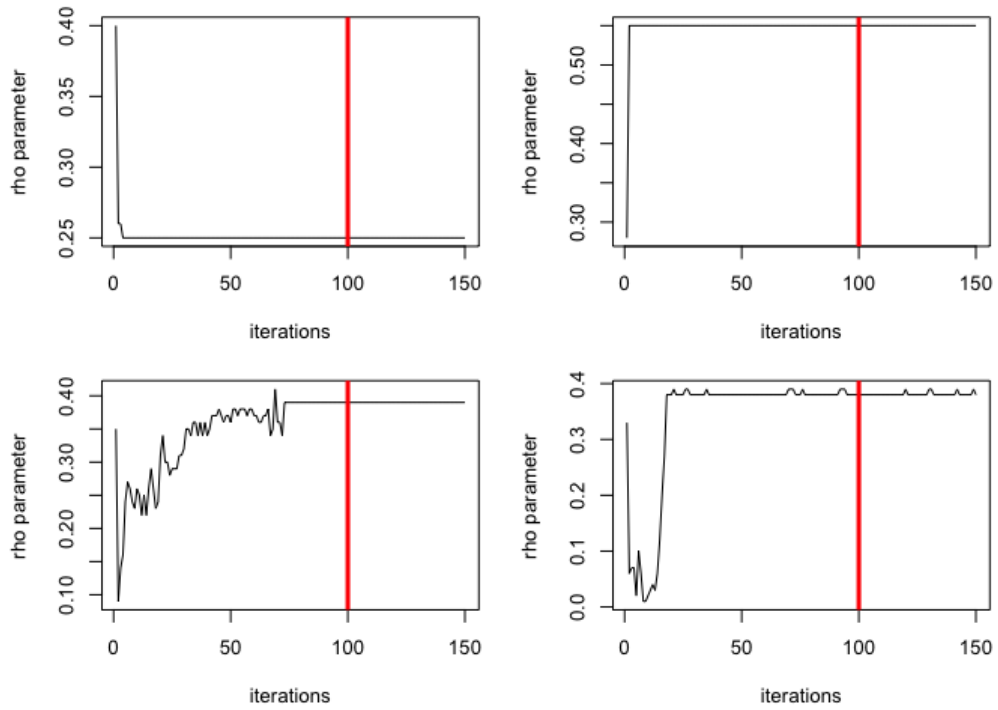


Figure 2: Evolution of parameters $\boldsymbol{\rho}$ through the SEM-Gibbs algorithm iterations. From left to right, and from top to bottom, the graph represents the evolution of the first element of each vector $\boldsymbol{\rho}_1$, $\boldsymbol{\rho}_2$, $\boldsymbol{\rho}_3$ and $\boldsymbol{\rho}_4$. The red vertical line indicates the burn-in period's end (*i.e* when the parameter is supposed to be stable).

be performed is not fixed because the algorithm stops once it can't find a better ICL-BIC value.

Choice for the number of iterations The number of iterations for the SEM-Gibbs algorithm was fixed to 150 and the burn-in period was considered to take 100 iterations. To check if this number of iterations is enough, the evolution of the parameters is graphically observed, as in Figure 2. Here, only the a few parameters are represented as an example, but it is useful to check for several parameters evolutions. We notice in this example that some of the parameters reached their stationary state since the beginning of the algorithm, and that other parameters need 50 to 100 iterations to get stable. Therefore, in order to ensure that all parameters have achieved their stationary distribution, a burn-in period of 100 iterations is considered (over a total number of iterations equal to 150). Numerical results also show that these choices are correct.

Choices for the initialization The number I corresponds to the number of iterations a certain percentage of the partitions are randomly sample when a cluster gets empty. Here, I is tuned to be equal to the number of iterations for burn-in, while the percentage value was fixed to 15.

5.2 Parameter estimation

The co-clustering was performed on 20 datasets, with the true numbers of clusters. The mean absolute error for the mixing proportions are in Table 3 and Table 4. The mean absolute error between the parameters values and their estimation are available in Table 5 and Table 6. All of these errors are extremely low, which means that the model parameters are correctly estimated.

Table 3: Mean absolute error for the mixing proportions with $N = J_d = 100$.

$\boldsymbol{\gamma}$	$\boldsymbol{\rho}_1$	$\boldsymbol{\rho}_2$	$\boldsymbol{\rho}_3$	$\boldsymbol{\rho}_3$
$(.09, .04, .06)$	$(.04, .04, .04)$	$(.03, .03, .03)$	$(.04, .02, .04)$	$(.05, .03, .05)$

Table 4: Mean absolute error for the mixing proportions with $N = J_d = 500$.

$\boldsymbol{\gamma}$	$\boldsymbol{\rho}_1$	$\boldsymbol{\rho}_2$	$\boldsymbol{\rho}_3$	$\boldsymbol{\rho}_3$
$(.05, .02, .04)$	$(.01, .02, .02)$	$(.01, .01, .01)$	$(.02, .02, .02)$	$(.02, .01, .01)$

Table 5: Value of the blocks' parameters mean absolute error on simulation with $N = J_d = 100$.

	Nominal $m = 5$ $\beta_1, \beta_2, \beta_3, \beta_4, \beta_5$		
	col-cluster 1	col-cluster 2	col-cluster 3
row-cluster 1	0.01 0.01 0.02 0.01 0.01	0.01 0.01 0.01 0.01 0.01	0.01 0.01 0.01 0.00 0.01
row-cluster 2	0.00 0.01 0.01 0.01 0.01	0.01 0.01 0.01 0.00 0.01	0.01 0.01 0.01 0.00 0.01
row-cluster 3	0.01 0.01 0.01 0.00 0.00	0.01 0.00 0.00 0.01 0.00	0.01 0.01 0.00 0.00 0.00

	Continuous μ, σ			Ordinal $m = 5$ μ, π			Integer $\delta \times 10^{-5}$		
	col-cluster 1	col-cluster 2	col-cluster 3	col-cluster 1	col-cluster 2	col-cluster 3	col-cluster 1	col-cluster 2	col-cluster 3
row-cluster 1	0.01,0.01	0.03,0.02	0.02,0.02	0.00,0.05	0.00,0.03	0.00,0.05	0.16	1.89	1.33
row-cluster 2	0.04,0.02	0.00,0.00	0.00,0.00	0.00,0.04	0.00,0.03	0.00,0.05	0.87	1.97	1.4
row-cluster 3	0.00,0.00	0.01,0.01	0.01,0.01	0.00,0.02	0.00,0.03	0.00,0.03	0.34	0.83	1.06

5.3 Model selection

In this section, the ICL-BIC criterion's efficiency is assessed to choose the right number of clusters in row and in column. Furthermore, the heuristic search described in 3.2 is evaluated. The complexity of the problem should be emphasized here. Usually, criteria like BIC or ICL are used to find the right number of clusters for the row partitions only. In the case of co-clustering, they are extended to find the right number of clusters for the row partitions and the columns partitions. In the present work, it is used to find $(D + 1)$ numbers of clusters (one for the rows, and one for each kind of features). Mathematically, the research space is much more extended which makes the problem more complex.

Table 6: Value of the blocks' parameters mean absolute error on simulation with $N = J_d = 500$.

	Nominal $m = 5$ $\beta_1, \beta_2, \beta_3, \beta_4, \beta_5$		
	col-cluster 1	col-cluster 2	col-cluster 3
row-cluster 1	0.00,0.00,0.00,0.00,0.00	0.00,0.00,0.00,0.00,0.00	0.00,0.00,0.00,0.00,0.00
row-cluster 2	0.00,0.00,0.00,0.00,0.00	0.00,0.00,0.00,0.00,0.00	0.00,0.00,0.00,0.00,0.00
row-cluster 3	0.00,0.00,0.00,0.00,0.00	0.00,0.00,0.00,0.00,0.00	0.00,0.00,0.00,0.00,0.00

	Continuous μ, σ			Ordinal $m = 5$ μ, π			Integer $\delta \times 10^{-7}$		
	col-cluster 1	col-cluster 2	col-cluster 3	col-cluster 1	col-cluster 2	col-cluster 3	col-cluster 1	col-cluster 2	col-cluster 3
row-cluster 1	0.2,0.03	0.3,0.09	0.1,0.08	0.00,0.04	0.00,0.03	0.00,0.01	0.1	0.2	0.1
row-cluster 2	0.01,0.02	0.1,0.1	0.1,0.06	0.00,0.02	0.00,0.03	0.00,0.04	0.5	0.1	0.1
row-cluster 3	0.00,0.00	0.01,0.01	0.01,0.01	0.00,0.03	0.00,0.02	0.00,0.03	0.3	0.2	0.3

Table 7: Exhaustive search results on 20 simulations results.

$N = J_d = 100$									
(G, H_1, H_2, H_3, H_4)	34333	34334	43333	**33333**	44333	34433	34443	44334	44433
number of occurrences	6	3	3	2	2	1	1	1	1

$N = J_d = 500$										
(G, H_1, H_2, H_3, H_4)	**33333**	33332	33323	43333	34342	32323	33343	34332	34322	44332
number of occurrences	6	3	2	2	2	1	1	1	1	1

Exhaustive Search Table 7 presents which sets (G, H_1, H_2, H_3, H_4) had the best ICL-BIC value in the exhaustive search. The number of occurrences indicates how many times the sets were chosen. It is here noticed that the right numbers of clusters $(G, H_1, H_2, H_3, H_4) = (3, 3, 3, 3, 3)$ has been chosen more times for the biggest dataset with $(N = J_d = 500)$. This result is coherent because the proposed ICL-BIC is based on asymptotic approximations.

Heuristic search Table 8 presents which sets (G, H_1, H_2, H_3, H_4) were chosen by the heuristic search. Once again, the algorithm works better for the biggest dataset with $(N = J_d = 500)$.

Search computation time The simulations were run with Linux 4.9.0-3-amd64 server, on a Debian 9 version. For the dataset with $N = J_d = 100$, the exhaustive search took

Table 8: Heuristic search results on 20 simulations results.

$N = J_d = 100$												
(G, H_1, H_2, H_3, H_4)	**33333**	22223	22233	22234	22243	22244	22334	23223	32223	33334	34334	37334
number of occurrences	5	3	3	1	1	1	1	1	1	1	1	1

$N = J_d = 500$					
(G, H_1, H_2, H_3, H_4)	**33333**	32233	22333	23234	32333
number of occurrences	10	5	3	1	1

23 minutes, while the heuristic search took at most 18 minutes. For the dataset with $N = J_d = 500$, the exhaustive search lasted 33 hours whereas the heuristic search took at most 2 hours. This means that in the case of a small dataset, it can be interesting to run an exhaustive search. Indeed, it does not take much more time than the heuristic search. However, an exhaustive search as it was realized in this simulation requires to know the neighborhood of the right set (G, H_1, H_2, H_3, H_4). For a bigger dataset, the heuristic search is recommended as it is very efficient and until 15 times faster than the exhaustive search. Furthermore, we can expect it to be even more than 15 times faster in case of bigger datasets than the ones used in these simulations.

5.4 Conclusion

As a conclusion for this simulation study, the SEM-Gibbs algorithm is efficient for estimating the model parameters and the partitions. For model selection, if we know that the ICL-BIC criterion leads to a consistent estimation of the number of blocks when the number of rows and column growth to infinity (see [19]), its behavior for finite sample size remains satisfying. Moreover, using the proposed heuristic search among the possible values for the number of clusters allows to drastically reduce the computing time without significantly decreasing the performance of the estimation.

6 Real data applications

In this section, two real datasets are considered. The first one regards the famous TED talks[1] and contains the transcripts and ratings of TED Talks uploaded to the official TED.com website until September 21st, 2017. It is a mixed dataset because the transcripts are textual data whereas the ratings are numbers. The second dataset is the result of a survey that Slovakian Statistic students gave to people around them. The responses were categorical with different numbers of level and some of them were ordinal.

6.1 Co-clustering of count and continuous data

The TED talks dataset TED is a non-profit organization which posts conferences on-line for free distribution. The conferences address a wide range of topics, including science, culture and innovation. The TED talks dataset[2] contains information about 2467 TED Talks. This work is focused on their transcript and their rating, which is given by the users. The rating system is special on this website. A list of fourteen words was defined (beautiful, inspiring, persuasive, fascinating, ok, longwinded, confusing, informative, courageous, ingenious, funny, obnoxious, unconvincing, jaw-dropping). When a user wants to rate a talk, he is asked to choose the three words that best describe the talk according to him.

[1] https://www.ted.com/talks

[2] https://www.kaggle.com/rounakbanik/ted-talks/data

Dataset pre-treatments First of all, a couple of TED talks were actually a musical performance. Their transcript were similar to "(Applause)(Music)(Applause)", which is information-less, so these talks were removed from the dataset. Then, the other transcripts were projected into a Document-Term matrix, each cell counting the occurrences of a term in a talk. By observing the matrix, we realized that some of the term would occur only once in the whole corpus. It appeared that most of them were onomatopoeia as "aargh" or "aaaaaaaargh". We decided to remove them: we assumed that these words would not bring valuable information, and that in the same time, it would reduce the dimension of the matrix. The ratings variables were used without changes. The resulting matrix is therefore of dimension $(2\ 464 \times (40\ 137 + 14))$, in other words, $N = 2\ 464$, $J_1 = 40\ 137$ $J_2 = 14$. The dataset is seen as two matrices of different types ($D = 2$). The first one is the Document-Term matrix of the transcripts whose occurrences are modeled by a Poisson distribution. The second matrix represents, for each talk, the number of users that voted for each of the words in the proposed adjectives list. Given the high number of votes, this number is modeled by a Normal distribution.

Co-clustering as a parsimonious clustering The main motivation on this dataset is to cluster the TED talks to distinguish the different kinds of talks, and to observe the ratings of each row-cluster. Using a classical clustering technique is not conceivable because of the high dimensionality of the dataset. The latent class model, for example, would define a distribution for each of the 40 151 variables and for each class, which is definitely over-parameterized not interpretable. With a co-clustering technique not only will the talks be clustered, but the variables will be clustered as well, which will result in a small number of interpretable blocks.

Co-clustering results After having searched for the highest ICL-BIC as explained in Section 3.2 with $(G_{min}, H_{1_{min}}, H_{2_{min}}) = (2, 2, 1)$, the best set (G, H_1, H_2) was found to be equal to $(8, 6, 2)$. Figure 3 gives a representation of the blocks' parameters. For the Document-Term matrix, the δ parameters are represented by the shades of gray. The lighter the block is, the lower its corresponding δ parameter is. When a block's δ parameter is high, it means that the column-cluster terms of this block are quite specific to the corresponding row-cluster. For the ratings matrix, the shades of gray represent the μ parameter of the resulting blocks. The darker the block is, the higher the μ parameter is.

First of all, we focus on the row-clusters of the document-term-matrix. By reading the titles of the same row-clusters, we notice that the co-clustering grouped talks with similar topics. As an example the third group seems to be about high technology and science with titles like "A robot that runs and swims like a salamander", "A mobile fridge for vaccines" or "The hunt for a supermassive black hole"; whereas the fourth group refers to politics, with talks called as "Why Brexit happened – and what to do next", "How ideas trump crises" , "Aid for Africa? No thanks.". By reading the ratings row-cluster parameters, we realize that the seventh row-cluster were marked about ten times more than the documents of the other row-clusters. It is interesting to observe that it corresponds to a row-cluster closely related to psychology and introspection. Table 9 gives an overview

of the Document-Term matrix row-clusters, by detailing some titles and the topic that was deduced from these latter. On another side, two row-clusters were more difficult to interpret. For example, the eighth row-cluster gathers talks as "Dare to educate Afghan girls", "Averting the climate crisis", "Fighting with nonviolence" or "What it's like to be a parent in a war zone". Even if the talks tend to be about education and parenting, the inherent topic is not obvious nor unique. The same issue was observed with the third row-cluster: with titles as "The magic of Fibonacci numbers", "A new equation for intelligence" or "New thinking on the climate crisis", it is hard to define a unique subject for this group.

It is not easy to explain the terms clusters because these column-clusters contain averagely about 6 000 variables. However, we extract here some of the 100 most frequent words for some notable blocks with high δ parameters to check if they are relevant to the row-clusters inherent topics of Table 9. First, from Figure 3a we notice the block $(6, 1)$ (corresponding to the 6^{th} row-cluster and 1^{st} column cluster). Among the most frequent words, we note "knowledge", "future", "company", "information", "community", "working", "imagine", which are relevant to the 6^{th} row-cluster topic about innovation and high-technology. Then, we notice the block $(4, 5)$. Some of the most frequent terms are "phenomenon", "coffee", "discovery", "organisms", "suffering", which, again, correspond to the 5^{th} row-cluster topic about medicine and health. Finally, the block $(5, 6)$ is investigated. It appears that its column-cluster terms are specific to the 5^{th} row-cluster about politics, and we find the words "india", "history", "technology", "program", "impact" among the most frequent ones.

We focus now on the column-clusters of the ratings for the TED talks. The adjectives were split into two groups. The first column cluster is composed of the following adjectives: "Inspiring", "Beautiful", "Courageous", "Persuasive", "Fascinating", "Informative", "Funny". In average, these adjective were more voted than those of the second column-cluster, and this for all the row clusters. The second column-cluster is composed of the adjectives "Ingenious", "Confusing", "Jaw-dropping" "Obnoxious", "Longwinded", "Unconvincing","OK".

From these observations, we can conclude that the co-clustering helped understanding and summarizing the dataset. First, it allowed to cluster the TED-talks documents and the resulting classes showed relevant results regarding the titles topics and corresponding term column-clusters. Furthermore, the rating matrix allows the user to interpret which kind of talks are preferred to the others. Overall, the co-clustering results gave an overview of a big dataset which could not be easily done by a human.

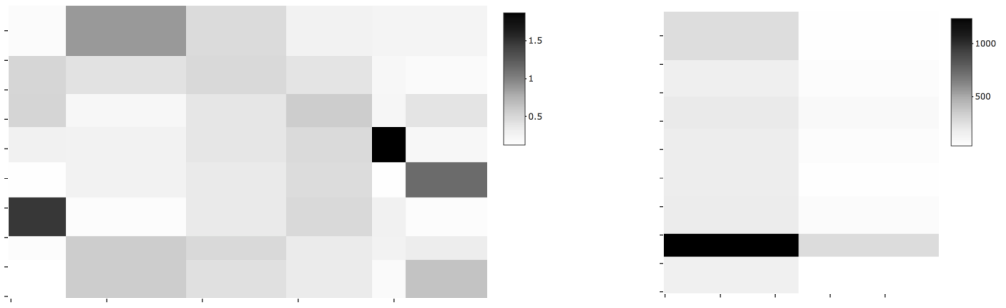
6.2 Co-clustering of ordinal and nominal data

Young People Responses to questionnaires In 2013, Slovakian students of a statistics class were asked to invite their friends to participate in a survey that concerned several aspects of their life [3]. The responses were defined on different scales; for example, a question as "I enjoy listening to music." could be answered from 1 ("Don't enjoy at all") to 5 ("Enjoy very much"). The questions regarding music preferences, movie preferences,

[3] https://www.kaggle.com/cardot/se-young-people-survey/data

Table 9: Row-cluster interpretation for the TED talks dataset.

Row-cluster number	Example titles	Interpreted topics
1	"My year of living biblically", "My journey from Marine to actor", "How I'm preparing to get Alzheimer's", "12 truths I learned from life and writing", "The year I was homeless"	Story-telling
2	"Art that craves your attention", "Building a museum of museums on the web", "How to engineer a viral music video", "A one-man orchestra of the imagination", "Moving sculpture"	Art, Culture
3	"The magic of Fibonacci numbers", "How behavioral science can lower your energy bill", "New thinking on the climate crisis", "A new equation for intelligence", "Winning the oil endgame"	Energy, Climate, Mathematics
4	"A map of the brain", "Your brain hallucinates your conscious reality", "Is anatomy destiny?", "Growing new organs", "A doctor's case for medical marijuana"	Medicine, Health
5	"Why Brexit happened – and what to do next", "How ideas trump crises", "Aid for Africa? No thanks.", "The surprising way groups like ISIS stay in power", "The attitudes that sparked Arab Spring"	Politics
6	"A robot that runs and swims like a salamander", "A mobile fridge for vaccines", "The hunt for a supermassive black hole", "Hands-on science with squishy circuits", "How we'll find life on other planets"	High technology, Science, Innovation
7	"Who are you, really? The puzzle of personality", "How to succeed? Get more sleep", "Your body language may shape who you are", "A kinder, gentler philosophy of success", "What really matters at the end of life"	Psychology, Introspection
8	"What it's like to be a parent in a war zone", "Teachers need real feedback", "Averting the climate crisis", "Dare to educate Afghan girls", "Fighting with nonviolence"	Education, Crisis



(a) (b)

Figure 3: Block representation of the Document-Term matrix (left) and of the ratings matrix (right). The shades of gray represent, for the Document-Term matrix, the δ parameter of each block, whereas they represent the μ parameter for the ratings matrix.

hobbies and interests, spending habits and phobias are seen as 5 levels ordinal data, not only because the answers are on a scale, but also because two answers can be compared. For example, questions concerning the music preferences could be : "I enjoy classical music." or "I enjoy rock music.", and both could be responded on a scale from 1 ("Don't enjoy at all") to 5 ("Enjoy very much"). In this case, the order in the responses is clear, and one can easily compare the two answers of a same user. However, in the case of personality traits, views on life and opinion, questions could be: "I have to be well prepared before public speaking."or "I always keep my promises.", still on a 5 levels scale from 1 "(Strongly disagree)" to 5 "(Strongly agree)". The order of the responses can not be compared, so considering them as ordinal would make their interpretation too arbitrary. That is why these questions where considered as categorical variables, with a number of levels equal to 5. Furthermore, demographic questions like "What is my gender?", with responses "Female" or "Male" are modeled as categorical variables with 2 levels. At last, 1 010 people answered this survey.

Thus, the resulting matrix is of dimension $(1\ 010 \times (80 + 5 + 54))$, so $N = 1\ 010$, $J_1 = 80$, $J_2 = 5$ and $J_3 = 54$. The dataset is seen as three matrices of different types. The first one regards the 80 questions answers considered as ordinal, with 5 levels. The second one contains the 5 questions with answers considered as nominal, with 2 levels. Finally, the third one represents the 54 questions with answers considered as nominal, with 5 levels.

At last, the dataset contained a few missing data (0.4%), which will be estimated through the SEM-Gibbs algorithm as explained in Section 3.

Co-clustering results The SEM-Gibbs algorithm repeated 150 iterations and the burn-in period was fixed to 100 iterations. These numbers were defined with the same technique as 6.1, by checking the evolution of several parameters through the SEM-Gibbs iterations. The best set (G, H_1, H_2, H_3) was found to be equal to $(3, 4, 2, 4)$. Figure 4 shows the resulting co-clustering, and Table 10 gives the estimated parameters of each block.

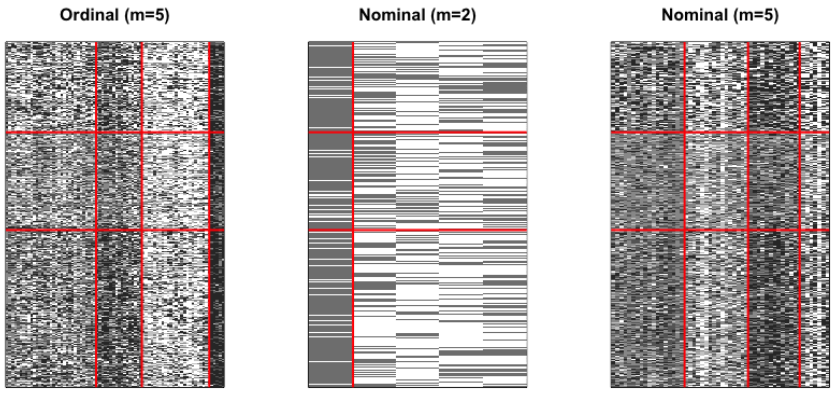


Figure 4: Co-clustering result on Young People Survey.

First of all, we notice that the first row cluster has the lowest position parameter μ on ordinal data's first column cluster. This means that people from this group globally enjoy less – or are less interested in, or are less afraid of – the topics of this column cluster's questions. Among others, some of these topics are classical music, branded clothing, psychology, politics or dangerous dogs. What's more, the parameters show that this row cluster is quite heterogeneous. Indeed, it has the lowest position parameters π on the two first column clusters of ordinal data, and they systematically has the highest β_1 and β_5 on categorical data with 5 levels. We observe now the second row-cluster. We notice that it has a β_3 parameter equals to 0.5 on the personality's questions first column-clusters, which is high. It means that people from this row-cluster are quite indecisive about the topics of these column-cluster's questions. Among others, the questions were "I am 100% happy with my life.", "I believe all my personality traits are positive.", "I have lot of friends.", "My moods change quickly.".

At last, we analyze the fourth column-cluster of the ordinal variables. We notice that it has the highest position parameter for all the row-clusters with $\mu = 5$. The questions of this column-cluster are: "I enjoy listening to music", "I enjoy watching movies", "I enjoy comedies", "I am interested in internet", "I am interested in socializing". This means that the interviewed persons globally agree on being very interested in these topics.

7 Conclusion

This article presented a model-based co-clustering model for datasets made of mixed type data. It relies on the latent block model and the inference is done through an SEM-Gibbs algorithm. The method has the great advantage to have an efficient criterion for selecting the number of row and column clusters. Furthermore, the parameters that are estimated on each block allow the user to easily interpret the partitions. Finally, the missing data is handled, which is often useful in the case of real datasets. The efficiency of the algorithm was proven on a simulated dataset and on real application and an R package implemented with C++ is available upon request to the authors. Moreover, if a user is interested in

Table 10: Resulting co-clustering parameters for dataset about students' survey.

	Ordinal ($m = 5$) μ, π				Nominal ($m = 2$) β_1, β_2	
	col-cluster 1	col-cluster 2	col-cluster 3	col-cluster 4	col-cluster 1	col-cluster 2
row-cluster1	1,0.08	5,0.16	1,0.41	5,0.69	0.1,0.9	0.63,0.37
row-cluster 2	3,0.25	3,0.21	1,0.31	5,0.52	0.11,0.89	0.62,0.38
row-cluster 3	3,0.14	5,0.28	1,0.24	5,0.74	0.1,0.9	0.7,0.3

	Nominal ($m = 5$) $\beta_1, \beta_2, \beta_3, \beta_4, \beta_5$			
	col-cluster 1	col-cluster 2	col-cluster 3	col-cluster 4
row-cluster1	0.10,0.13,0.32,0.2,0.25	0.3,0.15,0.25,0.12,0.18	0.10,0.10,0.19,0.16,0.45	0.39,0.10,0.15,0.09,0.27
row-cluster 2	0.02,0.14,0.5,0.28,0.06	0.13,0.29,0.38,0.16,0.04	0.02,0.13,0.34,0.33,0.18	0.23,0.23,0.26,0.16,0.12
row-cluster 3	0.03,0.11,0.36,0.35,0.15	0.16,0.25,0.31,0.18,0.10	0.03,0.08,0.20,0.31,0.38	0.24,0.16,0.19,0.18,0.23

clustering the observations, the proposed co-clustering algorithm gives a parsimonious way to do this, by grouping all the features in a small number of clusters.

We discuss now the limits of this model. Obviously, the main issue is that the variables of different types cannot be part of the same column-cluster since it assumes that the elements of a same block share the same distribution. It would be interesting to find an approach to overcome this problem. Furthermore, as it came up in Section 4.1, the way the data is encoded can have a strong impact on the co-clustering resulting partition. Although there exist ways to tackle the matter in some cases as detailed in [5], the user should be aware of it. Additionally, the influence of each kind of features on the resulting row partitions shall be investigated more deeply in a future work. Indeed, the expression of the probability of a row to belong to some row-cluster will be more impacted by certain types of data, even if the D matrices have the same number of features J_d. Also, the case where J_d are not equal should be studied. Actually, the user could be interested in giving the same importance to the D matrices, even if they do not have the same number of features.

References

[1] M. Ailem, F. Role, and M. Nadif. Sparse poisson latent block model for document clustering. *IEEE Transactions on Knowledge and Data Engineering*, 29(7):1563–1576, July 2017.

[2] C. Biernacki, G. Celeux, and G. Govaert. Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence.*, 22(7):719–725, July 2000.

[3] C. Biernacki, T. Deregnaucourt, and V. Kubicki. Model-based clustering with mixed/missing data using the new software MixtComp. In *CMStatistics 2015 (ERCIM 2015)*, London, United Kingdom, December 2015.

[4] C. Biernacki and J. Jacques. Model-Based Clustering of Multivariate Ordinal Data Relying on a Stochastic Binary Search Algorithm. *Statistics and Computing*, 26(5):929–943, 2016.

[5] C. Biernacki and A. Lourme. Unifying Data Units and Models in (Co-)Clustering. *Advances in Data Analysis and Classification*, 12(41), May 2018.

[6] A. Bouchareb, M. Boullé, and F. Rossi. Co-clustering de données mixtes à base des modèles de mélange. In *Actes de la 17ème Conférence Internationale Francophone sur l'Extraction et gestion des connaissances (EGC'2017)*, pages 141–152, Grenoble, France, 2017.

[7] C. Bouveyron, L. Bozzi, J. Jacques, and F. Jollois. The functional latent block model for the co-clustering of electricity consumption curves. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 67(4):897–915.

[8] C. Bouveyron, M. Fauvel, and S. Girard. Kernel discriminant analysis and clustering with parsimonious gaussian process models. *Statistics and Computing*, 25(6):1143–1162, 2015.

[9] V. Brault. *Estimation et selection de modele pour le modele des blocs latents.* PhD thesis, 2014. These de doctorat dirigee par Celeux, Gilles Mathematiques Paris 11 2014.

[10] N. Del Buono and G. Pio. Non-negative matrix tri-factorization for co-clustering: An analysis of the block matrix. *Information Sciences*, 301:13 – 26, 2015.

[11] G. Celeux, D. Chauveau, and J. Diebolt. Some stochastic versions of the em algorithm. *Journal of Statistical Computation and Simulation*, 55:287–314, 1996.

[12] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of he Royal Statistical Society, series B*, 39(1):1–38, 1977.

[13] B. S. Everitt. *Introduction to Latent Variable Models.* Chapman and Hall. 1984.

[14] A. Gelman and D.B. Rubin. Inference from iterative simulation using multiple sequences. *Statistical Science*, 7(4):457–472, 1992.

[15] G. Govaert and M. Nadif. Latent block model for contingency table. *Communications in Statistics - Theory and Methods*, 39(3):416–425, 2010.

[16] G. Govaert and M. Nadif. *Co-Clustering.* Computing Engineering series. ISTE-Wiley, 2014.

[17] J. Jacques and C. Biernacki. Model-based co-clustering for ordinal data. *Computational Statistics and Data Analysis*, 123:101–115, 2018.

[18] K. Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21, 1972.

[19] C. Keribin, V. Brault, G. Celeux, and G. Govaert. Estimation and Selection for the Latent Block Model on Categorical Data. Research Report RR-8264, INRIA, November 2013.

[20] C. Laclau and M. Nadif. Diagonal latent block model for binary data. *Statistics and Computing*, 27(5):1145–1163, Sep 2017.

[21] R. J. A. Little and D. B. Rubin. *Statistical Analysis with Missing Data*. John Wiley & Sons, Inc., New York, NY, USA, 1986.

[22] G.H. Lubke and B.O. Muthén. Applying multigroup confirmatory factor models for continuous outcomes to likert scale data complicates meaningful group comparisons. *Structural Equation Modeling: A Multidisciplinary Journal*, 11(4):514–534, 2004.

[23] E. E. MaloneBeach and S. H. Zarit. Dimensions of social support and social conflict as predictors of caregiver depression. *International Psychogeriatrics*, 7(1):25–38, 1995.

[24] M. Marbac, C. Biernacki, and V. Vandewalle. Model-based clustering of gaussian copulas for mixed data. *Communications in Statistics - Theory and Methods*, 46(23):11635–11656, 2017.

[25] D. McParland and I.C. Gormley. Model based clustering for mixed data: Clustmd. *Adv. Data Anal. Classif.*, 10(2):155–169, June 2016.

[26] M. Nadif and G. Govaert. Algorithms for model-based block gaussian clustering. In *DMIN'08, the 2008 International Conference on Data Mining*, Las Vegas, Nevada, USA, July 14-17 2008.

[27] V. Robert. *Classification croisee pour l'analyse de bases de donnees de grandes dimensions de pharmacovigilance*. PhD thesis, Université Paris-Sud, 2017.

[28] G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6:461–464, 1978.

[29] M. Selosse, J. Jacques, and C. Biernacki. Analyzing health quality survey using constrained co-clustering model for ordinal data and some dynamic implication. preprint <hal-01643910>, 2018.

[30] P. Singh Bhatia, S. Iovleff, and G. Govaert. blockcluster: An R package for model-based co-clustering. *Journal of Statistical Software*, 76(9):1–24, 2017.

[31] Y. Ben Slimen, S. Allio, and J. Jacques. Model-based co-clustering for functional data. *Neurocomputing*, 291:97 – 108, 2018.

[32] A. S. Zigmond and R. P. Snaith. The hospital anxiety and depression scale. *Acta Psychiatrica Scandinavica*, 67(6):361–370, 1983.