HAL open science

Co-clustering of evolving count matrices with the dynamic latent block model: application to pharmacovigilances

Giulia Marchello, Audrey Fresse, Marco Corneli, Charles Bouveyron

To cite this version:

Giulia Marchello, Audrey Fresse, Marco Corneli, Charles Bouveyron. Co-clustering of evolving count matrices with the dynamic latent block model: application to pharmacovigilances. Statistics and Computing, 2022, 32 (41), 10.1007/s11222-022-10098-y . hal-03146769v2

HAL Id: hal-03146769

https://hal.science/hal-03146769v2

Submitted on 12 May 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Co-clustering of evolving count matrices with the dynamic latent block model: application to pharmacovigilances

Giulia Marchello[1], Audrey Fresse[2], Marco Corneli[3] & Charles Bouveyron[1]

[1] *Université Côte d'Azur, Inria, CNRS, Laboratoire J.A.Dieudonné, Maasai team, Nice, France.*

[2] *Université Côte d'Azur, Department of Clinical Pharmacology, Pasteur Hospital, Nice, France.*

[3] *Université Côte d'Azur, Inria, Maison de la Modélisation des Simulations et des Interactions (MSI), Maasai team, Nice, France.*

Abstract. The simultaneous clustering of observations and features of datasets (known as co-clustering) has recently emerged as a central topic in machine learning applications. However, most models focus on continuous data in stationary scenarios, where cluster assignments do not evolve over time. We propose in this paper the dynamic latent block model (dLBM), which extends the classical binary latent block model, making amenable such analysis to dynamic cases where data are counts. Our approach operates on temporal count matrices allowing to detect abrupt changes in the way existing clusters interact with each other. The time breaks detection is performed through clustering of time instants, that allows for better model parsimony. The time dependent counting data are modeled via non-homogeneous Poisson processes (HHPPs), conditionally to the latent variables. In order to handle the model inference, we rely on a SEM-Gibbs algorithm and the ICL criterion is used for model selection. Numerical experiments on simulated data highlight the main features of the proposed approach and show the interest of dLBM with respect to related works. An application to adverse drug reaction in pharmacovigilance is also proposed, where dLBM was able to recognize clusters in a meaningful way that identified safety events that were consistent with retrospective knowledge. Hence, our aim is to propose this dynamic co-clustering method as a tool for automatic safety signal detection, to support medical authorities.

Keywords. Co-clustering, pharmacovigilance, latent block model, dynamic count matrices, SEM-Gibbs algorithm.

1 Introduction

Nowadays, there is an increasing need to group observations and features of high-dimensional matrices simultaneously. Machine learning models that allow to perform this kind of operations are called co-clustering methods and are used in a wide range of applications (e.g. natural language processing, recommending systems, biomedical data, etc.). However, although the study in this field has been greatly expanded by many notable methods introduced in the last few decades, the development of dynamic co-clustering methods still remains almost an unexplored territory. Also, count data modeling is another application area that is still underdeveloped but whose importance is gradually increasing in a number of domain. Therefore, this paper attempts to fill this gap by allowing temporal counting matrices to be managed, where count data are modeled by non-homogeneous Poisson processes. Our goal is to develop a method capable of identifying deep changes that may occur in the evolution of the data structure or in the way in which existing groups interact. For example, e-commerce systems record in continuous time all purchases made by customers but, to avoid information overload and to manage these unstructured data, it is important to identify methods capable of summarizing large amounts of data in a dynamic context. In fact, it is of interest for those companies to cluster both customers and products to better understand users preferences.

Another particularly topical application area is pharmacovigilance, whose main activity concerns the detection of safety signals about drugs. However, the method currently used, i.e. manual expert detection of safety signals, despite being unavoidable, has the disadvantage of being incomplete due to its workload and to require a significant amount of data before being able to detect a critical event. This is why, developing automatic method of safety signal detection is currently a major issue in pharmacovigilance to allow health professionals to enlarge the application of pharmacovigilance to all numerical documents and reports that are generated by health facilities. Figure 1 shows the data structure of a large-scale dataset made of the notifications of adverse drug reactions (ADRs) gathered between 2010 to 2020 by the Regional Center of Pharmacovigilance (RCPV) of Nice (France). Each upper panel shows the interactions between a few molecules, on the x-axis, and a few adverse drug reactions (ADRs), on the y-axis, for three selected periods (highlighted in dark blue in the lower panel). The histograms below shows the distribution of ADR notifications arrived to the RCPV during the whole time period. In

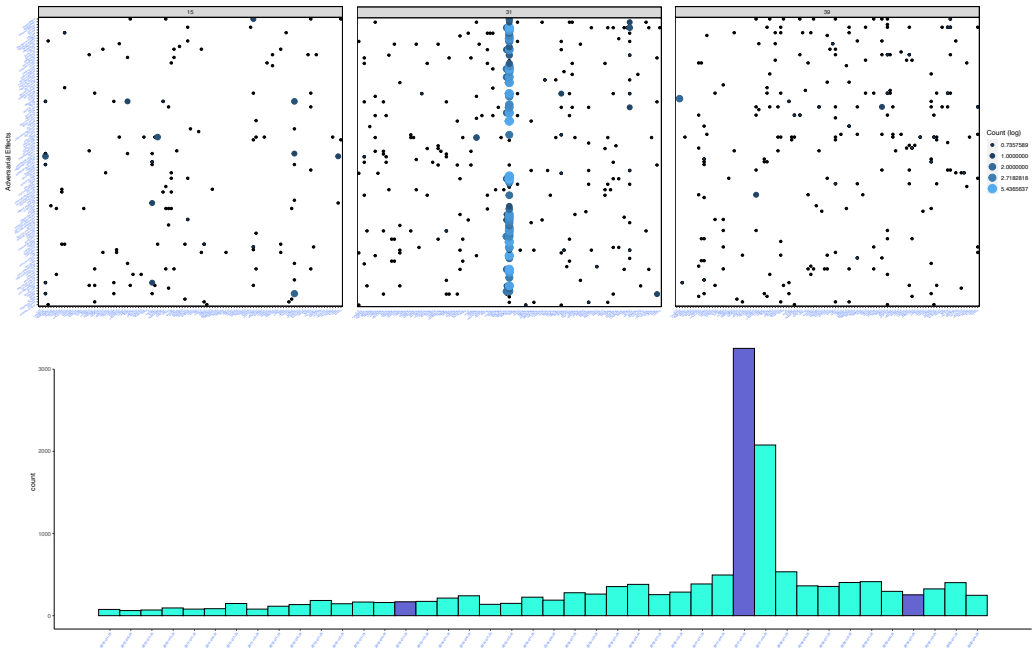

Figure 1: Notifications of adverse drug reactions (ADRs) reported Regional Center of Pharmacovigilance (RCPV) of Nice (France) in 3 different time periods, highlighted in dark blue.

such a context, co-clustering may play an important role in summarizing the information carried out by pharmacovigilance data and identifying patterns of interest. It would be indeed of interest to cluster both the drugs and the adverse reactions along the temporal dimension to assist medical experts in the retrospective detection of safety signals. For this reason, we investigate in this work the use of model-based co-clustering as a retrospective tool for automatic safety signal detection. Since the ADR notifications can be viewed as count data observed along the time, we introduce a generative co-clustering model, named the dynamic latent block model (dLBM), to extend the classical binary latent block model to the case of dynamic count data, allowing in turn the detection of temporal breaks in the signals. This work aims to demonstrate the potential of dLBM as a routine tool in retrospective pharmacovigilance.

1.1 Related works

A co-clustering problem can be seen as the simultaneous clustering of rows (individuals) and columns (variables) of a data matrix. To avoid any doubt, it may be useful to point out here the difference in the concepts between co-clustering and bi-clustering. Bi-clustering methods allow to have overlapping clusters and to leave some of the observations unclustered, while in co-clustering each entry belongs to one and only one cluster (it is not possible to leave some entries unassigned). In the last decades, different co-clustering techniques have been developed for a wide range of applications. Due to the spreading of e-commerce companies (e.g. Amazon, Ebay) and the exponential growth of social networks (e.g. Facebook, Linkedin, Twitter), application of co-clustering techniques to recommending systems and collaborative filtering have emerged (George and Merugu, 2005; Deodhar and Ghosh, 2010; Xu et al., 2012). Other applications include gene expression analysis (Cheng et al., 2008; Hanisch et al., 2002), text mining and topic modeling (Bergé et al., 2019; Dhillon et al., 2003a; Wang et al., 2009).

Co-clustering methods can be distinguished into metric based approches such as: non-negative matrix tri-factorization (NMTF) (Labiod and Nadif, 2011; Ding et al., 2006), spectral co-clustering (Dhillon, 2001), information theory (Dhillon et al., 2003b) and model-based co-clustering approaches (Bouveyron et al., 2019).

Among those approaches, model-based co-clustering is widely appreciated for its sound statistical foundations and its flexibility in terms of sparsity and data types. The much popular latent block model (LBM) (Govaert and Nadif, 2003) was introduced for the co-clustering of binary data matrices, based on the assumption that rows and columns are grouped in hidden clusters and that observations within a block (intersection of a row cluster and a column cluster) are independent and identically distributed. Whereas the original formulation of the model dealt with binary data, the model has been extended in the last two decades to counting data (Govaert and Nadif, 2010), continuous data (Lomet, 2012), categorical data (Keribin et al., 2015), ordinal data (Jacques and Biernacki, 2018; Corneli et al., 2020), functional data (Bouveyron et al., 2018) and textual data (Bergé et al., 2019). Several inference procedures have been also proposed for LBM, including likelihood based methods (Govaert and Nadif, 2008), variational inference (Keribin et al., 2012), Bayesian inference (Keribin et al., 2012; Wyse and Friel, 2012) and greedy search approaches (Wyse and Friel, 2012; Wyse et al., 2017). Recently,

Boutalbi et al. (2020) proposed the tensor latent block model (TLBM) for co-clustering, whose aim is to simultaneously cluster rows and columns of a 3D matrix, where covariates represent the third dimension. TLBM was also implemented for different types of datasets: continuous data (Gaussian TLBM), binary data (Bernoulli TLBM) and contingency tables (Poisson TLBM).

Among the dynamic extensions of the co-clustering approaches, we can cite the evolutionary spectral co-clustering (ESCC, Green et al., 2011). In that paper, two different ways of taking into account the historical relationship between the instances and features are proposed, one respect to current (RTC) and the other respect to historical (RTH). The first one gives most importance to the present, considering only the previous time-step, while the second one takes into account all the previous time-steps. For the continuous time modeling, an original approach was proposed by Corneli et al. (2018) for the dynamic stochastic block model, in a slightly different context: the clustering of dynamic networks. The authors considered the interactions between nodes in a dynamic network, where the goal is to cluster both nodes and time intervals into groups. The number of interactions between two nodes is modeled via a non-homogeneous Poisson process (NHPP) whose instantaneous intensity functions are assumed to be constant on each time cluster. A greedy maximization of the exact-ICL (Biernacki et al., 2000; Côme and Latouche, 2015) was used to cluster the nodes, the time intervals and to select the optimal number of clusters. Concerning the temporal stochastic block model, another closely related paper is the one of Matias et al. (2018) where the interactions are modeled through a non-homogeneous Poisson process. Here, the nodes are assumed to belong to clusters whose composition do not change over time.

Concerning co-clustering applications to pharmacovigilance, a seminal article was proposed by Robert et al. (2015). In that paper, the authors introduced the multiple latent block model (MLBM) by extending the latent block model (Govaert and Nadif, 2008) through the construction of one row partition and two columns partitions that respectively rely on two binary matrices containing the relation between the individual and the medical product and the relation between the individual and the ADR.

1.2 Contributions of this work

This work introduces a generative co-clustering model, named the dynamic latent block model (dLBM), allowing in turn the detection of temporal breaks in the signals, as a retrospective tool for Pharmacovigilance. We consider ADR count data matrices evolving over a time period $[0, T]$, whose number of rows and columns are fixed. We assume that the number of interactions occurring in time between rows and columns follows a non-homogeneous Poisson process (NHPP). To handle the dynamic framework, we led a segmentation over the continuous time such that we obtain as many static matrices as the number of the identified time intervals. As inference procedure, we use the SEM-Gibbs algorithm while the ICL criterion is adopted for selecting the optimal number of clusters. Thus, dLBM provides a meaningful summary of massive datasets, possibly with a large number of missing data. To demonstrate the interest in pharmacovigilance, we run a large-scale retrospective experiment on an 10-year ADRs dataset from Regional Center of Pharmacovigilance (RCPV), located in the University Hospital of Nice (France). The interest of this application lies not only in summarizing the massive amount of drug adverse reactions (ADRs) data but also in identifying possible unexpected phenomena, such as atypical side effects of certain types of drugs. Indeed, dLBM was not only able to identify clusters that are coherent with retrospective knowledge, such as the Lévothyrox® and Mirena® crises, but also to detect an under-notification of bleeding ADRs during the Lévothyrox® crisis, the health professionals were unaware of.

1.3 Organization of the paper

This paper is organized as follows. Section 2 first recalls the original latent block model, before introducing the generative model of dLBM. In Section 3, the inference procedure is detailed and a model selection criterion is discussed. Section 4 presents various experiments on simulated data to test and evaluate the model performances. In Section 5, an application on a real ADRs dataset is presented to illustrate the potential of dLBM in pharmacovigilance. Section 6 provides some concluding remarks. Appendix A, in supplementary materials, presents various experiments on simulated data to assess the model performances.

2 The dynamic latent block model

In this section, we introduce the dynamic latent block model (dLBM). The main goal of this model is the simultaneous clustering of rows and columns of high-dimensional sparse matrices in a dynamic time framework. The data we consider are organized such that the rows (drugs in pharmacovigilance application) are indexed by $i = 1, \dots N$ and the columns (adversarial effects) by $j = 1, \dots, P$. Moreover, we consider a fixed time period $[0, T]$ during which the total number of rows, N, and columns, P, is fixed. We indicate as $\mathcal{X}(t)$ the $N \times P$ matrix that contains the number of interactions occurring between the individual i and the item j at time $t \in [0, T]$. The first part of this section reviews the latent block model, while, in the second part, the proposed dynamic extension is introduced.

2.1 The latent block model

Let us first neglect the time dimension and recall the original latent block model (Govaert and Nadif, 2010), assuming that the time period is restricted to a single time point. The data structure is therefore a $N \times P$ random matrix $X = \{X_{ij}\}_{i \in 1,\dots,N, j \in 1,\dots,P}$. Rows and columns of X are assumed to be clustered respectively into K and L groups, such that the data belonging to the same block are independent and identically distributed. More formally, the latent structure of rows and columns of X is identified by the following latent variables:

- $Z := \{z_{ik}\}_{i \in 1,\dots,N, k \in 1,\dots,K}$ represents the clustering of rows into K groups: $\mathcal{A}_1, ..., \mathcal{A}_K$. The row i belongs to cluster $\mathcal{A}_k$ iff $z_{ik} = 1$;
- $W := \{w_{j\ell}\}_{j \in 1,\dots,P, \ell \in 1,\dots,L}$ represents the clustering of columns into L groups: $\mathcal{B}_1, ..., \mathcal{B}_L$. The column j belongs to cluster B_ℓ iff $w_{j\ell} = 1$.

Moreover, Z and W are assumed to be independent and distributed according to multinomial distributions:

$$p(Z|\gamma) = \prod_{k=1}^{K} \gamma_k^{|\mathcal{A}_k|}, \qquad p(W|\rho) = \prod_{\ell=1}^{L} \rho_\ell^{|\mathcal{B}_\ell|},$$

where $\gamma_k = \mathbb{P}\{z_{ik} = 1\}$, $\rho_\ell = \mathbb{P}\{w_{j\ell} = 1\}$, $\sum_{k=1}^{K} \gamma_k = 1$, $\sum_{\ell=1}^{L} \rho_\ell = 1$, and $|\mathcal{A}_k|$ and $|\mathcal{B}_\ell|$ respectively represent the number of rows in cluster $\mathcal{A}_k$ and the number of columns in cluster $\mathcal{B}_\ell$.

The LBM model further assumes the entries X_{ij} are independent, conditionally to Z and W, and their distribution $\varphi(.,\alpha)$ belongs to the same parametric family, where the parameter α only depends on the given block:

$$X_{ij} \mid z_{ik}w_{j\ell} = 1 \sim \varphi(X_{ij}, \alpha_{k\ell}). \tag{1}$$

With these assumptions the complete data likelihood can be written as:

$$\begin{aligned} p(X, Z, W; \theta) &= p(Z; \theta)p(W; \theta)p(X|Z, W; \theta) = \\ &= \prod_{k=1}^{K} \gamma_k^{|\mathcal{A}_k|} \prod_{\ell=1}^{L} \rho_\ell^{|\mathcal{B}_\ell|} \prod_{i,k} \prod_{j,\ell} \varphi(X_{ij}; \alpha_{k\ell})^{z_{ik}w_{j\ell}}. \end{aligned} \tag{2}$$

As mentioned in Section 1.1 various versions of LBM have been proposed along the decades. The first one was proposed for the binary case, where $\varphi(X_{ij}, \alpha_{k\ell})$ is a Bernoulli distribution with $p(X_{ij} = 1|Z, W; \theta) = \alpha_{kl}$. Since this article focuses on count data, the most suitable extension of LBM for our purpose is the Poisson LBM. In fact, this model is based on the assumption that, conditionally on Z and W, the entries X_{ij} follow a Poisson distribution $\mathcal{P}(\lambda_{k\ell})$:

$$\varphi(X_{ij}; \lambda_{k\ell}) = \left(\frac{\lambda_{k\ell}^{X_{ij}}}{X_{ij}!} \exp(-\lambda_{k\ell}) \right). \tag{3}$$

2.2 Modeling the dynamic framework

Let us now introduce the time dimension such that $\mathcal{X}_{ij}$ is time dependent. Thus, $\mathcal{X}_{ij}(t)$, $t \in [0, T]$, represents the cumulative number of interactions at time t between i and j.
A possible approach for the dynamic modeling relies on non-homogeneous Poisson processes (NHPPs), thus assuming that $\{\mathcal{X}_{ij}(.)\}_{i,j}$ are independent point processes, with instantaneous intensity functions $\lambda_{ij}(t)$:

$$\mathcal{X}_{ij}(t) \sim \mathcal{P}\left(\int_0^t \lambda_{ij}(u)du \right), \tag{4}$$

where $\mathcal{P}(\lambda)$ denotes the Poisson probability mass function of parameter λ. With the notation adopted so far, we thus assume the existence of $N \times P$ independent Poisson processes.
In order to cluster both the rows and the columns, we further assume that

the intensity function $\lambda_{ij}(t)$ only depends on the respective clusters of row i and column j:

$$\mathcal{X}_{ij}(t) \mid z_{ik}, w_{j\ell} = 1 \sim \mathcal{P}\left(\int_0^t \lambda_{k\ell}(u)du\right).$$

For further use, let us introduce the parameter $\lambda := (\lambda_{k\ell}(t))_{k\leqslant K;\ell\leqslant L}$. Given the above assumptions, the conditional distribution of the number of interactions between i and j, over the time interval $[s,t]$, where $0 \leqslant s \leqslant t \leqslant T$, is:

$$p(\mathcal{X}_{ij}(t)-\mathcal{X}_{ij}(s) \mid z_{ik}w_{j\ell} = 1, \lambda) = \frac{(\int_s^t \lambda_{k\ell}(v)dv)^{\mathcal{X}_{ij}(t)-\mathcal{X}_{ij}(s)}}{(\mathcal{X}_{ij}(t)-\mathcal{X}_{ij}(s))!} \exp\left(-\int_s^t \lambda_{k\ell}(v)dv\right). \tag{5}$$

2.2.1 A discrete time version

In order to ease the understanding of the dynamic model and to make the inference tractable and computationally efficient, we also operate a clustering over the time dimension. Let us first introduce a discretization of the considered time interval $[0,T]$. Thus, without loss of generality the following partition of $[0,T]$ is introduced:

$$0 = t_0 < t_1 < \cdots < t_U = T, \tag{6}$$

where the U intervals, $I_u = [t_{u-1}, t_u[$, will also be clustered. The number of interactions between i and j on the time interval I_u can be therefore summarized by X_{iju}:

$$X_{iju} := \mathcal{X}_{ij}(t_u) - \mathcal{X}_{ij}(t_{u-1}), \qquad \forall(i,j,u),$$

where $\mathcal{X}_{ij}(t_u)$ represents the cumulative number of interactions at time t_u between i and j. Hence, we introduce the tensor $X := \{X_{iju}\}_{iju}$ with dimension $N \times P \times U$ that contains the number of interactions between any observation and feature pair at any given time interval. We can also see X as a time series (along the third dimension) of incidence matrices.

Since our goal is to perform clustering over the time dimension as well, each time interval $I_1, \ldots, I_U$ is also assumed to be assigned to a hidden time cluster $\mathcal{D}_1, \ldots, \mathcal{D}_c$. To model the membership to time clusters, a new latent variable S has to be introduced, such that $s_u = c$ if the time interval I_u belongs to the time cluster $\mathcal{D}_c$. As it is shown in Figure 2, it is worth noticing

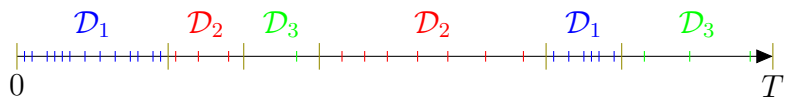

Figure 2: Time clusters.

that a specific time cluster can occur more than once in the temporal line when a similar interactivity pattern is repeated in time. Furthermore, as for Z and W, we assume that S follows a multinomial distribution:

$$p(S \mid \delta) = \prod_{c=1}^{C} \delta_c^{|\mathcal{D}_c|}, \tag{7}$$

where $\delta_c = \mathbb{P}\{s_{uc} = 1\}$; $\sum_{c=1}^{C} \delta_c = 1$ and $\mid \mathcal{D}_c \mid$ represents the number of time intervals in the cluster $\mathcal{D}_c$.

Once these additional assumptions have been made, we can rewrite Eq. (4) considering that the intensity functions are stepwise constant on each time cluster $\mathcal{D}_c$. Thus:

$$X_{iju} | z_{ik} w_{j\ell} s_{uc} = 1 \sim \mathcal{P}(\lambda_{k\ell c} \Delta_u), \tag{8}$$

where Δ_u indicates the length of the interval I_u. Henceforth, in order to simplify the exposition, we assume that Δ_u is constant, $\Delta_u = \Delta$. We can finally set $\Delta = 1$ without loss of generality. A graphical representation of dLBM can be seen in Figure 3.

From Eqs. (3)-(8), it holds that:

$$p(X_{iju} \mid z_{ik} w_{j\ell} s_{uc} = 1, \lambda_{k\ell c}) = \left(\frac{(\lambda_{k\ell c})^{X_{iju}}}{X_{iju}!} \exp\left(-\lambda_{k\ell c}\right) \right). \tag{9}$$

Therefore, we can introduce the $K \times L \times C$ tensor λ, whose elements are denoted by $\lambda_{k\ell c}$.

It is now possible to write the complete data likelihood of the model:

$$p(X, Z, W, S | \gamma, \rho, \delta, \lambda) = p(Z|\gamma) p(W|\rho) p(S|\delta) p(X|Z, W, S, \lambda), \tag{10}$$

where $p(Z|\gamma)$, $p(W|\rho)$ and $p(S|\delta)$ were defined in the previous section. The conditional distribution of X, given Z, W, and S, can be easily obtained

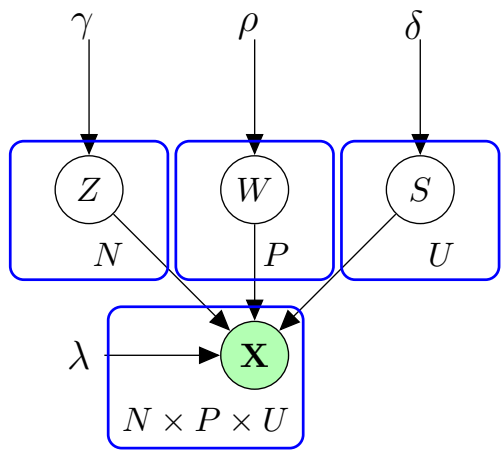

Figure 3: Graphical representation of dLBM.

from Eq. (9) by independence:

$$p(X|Z,W,S,\lambda) = \prod_{k,\ell,c} \left(\frac{(\lambda_{k\ell c})^{R_{k\ell c}}}{P_{k\ell c}} \exp\left(- \mid \mathcal{A}_k \mid\mid \mathcal{B}_\ell \mid\mid \mathcal{D}_c \mid \lambda_{k\ell c}\right) \right), \tag{11}$$

where $R_{k\ell c} = \sum_{i=1}^{N}\sum_{j=1}^{P}\sum_{u=1}^{U} z_{ik} w_{j\ell} s_{uc} X_{iju}$ and $P_{k\ell c} = \prod_{i=1}^{N}\prod_{j=1}^{P}\prod_{u=1}^{U} (z_{ik} w_{j\ell} s_{uc} X_{iju})!$. Denoting by θ the set of all model parameters, $\theta = (\gamma, \rho, \delta, \lambda)$, the log-likelihood can be finally written as:

$$\ell(\theta; X) = \sum_{Z}\sum_{W}\sum_{S} \log p(X, Z, W, S \mid \theta). \tag{12}$$

2.3 Link with related models

At this point, dLBM can be related with the following models:

- If we do not take into account the time dependency, assuming that the time period is restricted to a single time point t_0, dLBM coincides with the Poisson LBM.
- dLBM reduces to dSBM (Corneli et al., 2016) if the row individuals are the same as the column. In that case, in fact, Z would be equal to W and, then consequently, $K = L$. Therefore, in this way, we pass from an incidence matrix to an adjacency matrix.

- If $C = U$, dLBM corresponds to TensorLBM (Boutalbi et al., 2020), where the third dimension is considered but the slices are not clustered. In fact, when the contingency table case is analyzed, the authors consider a Poisson TensorLBM, where for each slice a, the entries are distributed according to a $Poisson(\lambda_{ij}^a)$.

3 Inference algorithm and model selection

3.1 Model inference

For the co-clustering model based approach outlined in the previous section, as well as for standard mixture models, a direct maximization of the log-likelihood with respect to the model parameters is not feasible. A widely used approach to overcome this problem is the expectation-maximization (EM) algorithm (Dempster et al., 1977). It consists in alternating two steps, E step and M step, in order to maximize a lower bound of the log-likelihood. The EM-algorithm exploits the following decomposition of the log-likelihood:

$$\ell(\theta; X) = \log p(X \mid \theta) = \mathcal{L}(q(Z, W, S); \theta) + KL(q(Z, W, S) \mid\mid p(Z, W, S \mid X, \theta)), \tag{13}$$

with:

$$\mathcal{L}(q(Z, W, S); \theta) = \sum_{Z,W,S} q(Z, W, S) \log \frac{p(X, Z, W, S \mid \theta)}{q(Z, W, S)}, \tag{14}$$

where $\mathcal{L}(\cdot; \theta)$ represents a lower bound of the log-likelihood while $KL(q||p)$ is the Kullback-Leibler divergence between q and p. During the E-Step, the algorithm maximizes the lower bound $\mathcal{L}(q(Z, W, S); \theta)$ with respect to $q(.)$ for a given value of θ. From Eq.(13), we can asses that maximizing $\mathcal{L}(q; \theta)$ is equivalent to minimizing $KL(q||p(.|X, \theta))$ and since the Kullback-Leibler divergence cannot be negative, we conclude that the lower bound is maximized when $KL(q||p(.|X, \theta)) = 0$. This leads us to the conclusion that $\mathcal{L}(q(Z, W, S); \theta)$ is optimized when $q^*(Z, W, S) = p(Z, W, S \mid X, \theta)$. The M-Step usually consists in maximizing $\mathcal{L}(q(Z, W, S); \theta)$ over θ, keeping $q = q^*$ fixed, in order to obtain an updated version of the parameter, θ.

However, in the context of the present work the joint posterior distribution $p(Z, W, S \mid X, \theta)$ is not computationally tractable as well. To go through this limitation, we propose to approximate it through a Gibbs sampler within

the E-step. Such an approach was proposed by Keribin et al. (2010) and exploited, for instance, by Bouveyron et al. (2018) for the functional latent block model (funLBM). The resulting stochastic alternative of the EM algorithm, called SEM-Gibbs algorithm, starts with some initial values of the parameter $\theta^{(0)}$, the column clusters $W^{(0)}$ and the time clusters $S^{(0)}$. In this way, at the hth iteration the algorithm alternates the following SE step and M step:

SE step: a partition for Z, W and S is drawn according to $q^*(Z, W, S) = p(Z, W, S \mid X, \theta)$, which is approximated by making use of a Gibbs sampler, using the current values of the parameter set θ. In this way, the unknown labels are simulated from their posterior distribution, given the observed data and the parameter set. It consists in executing a small number of iterations of the following three steps:

1) Generate the row partition $z_i^{(h+1)} = (z_{i1}^{(h+1)}, \ldots, z_{iK}^{(h+1)})|X, W^{(h)}, S^{(h)}$ according to $z_i^{(h+1)} \sim \mathscr{M}(1, \widetilde{z}_{i1}, \ldots, \widetilde{z}_{iK})$, for all $1 \leq i \leq N$ and $1 \leq k \leq K$, where:

$$\widetilde{z}_{ik} = p(z_{ik} = 1 \mid X, W^{(h)}, S^{(h)}; \theta^{(h)}) = \frac{\gamma_k^{(h)} f_k(X_i \mid W^{(h)}, S^{(h)}; \theta^{(h)})}{\sum\limits_{k'} \gamma_{k'}^{(h)} f_{k'}(X_i | W^{(h)}, S^{(h)}; \theta^{(h)})},$$

where $X_i = (X_{iju})_{ju}$ and $f_k(X_i \mid W^{(h)}, S^{(h)}; \theta^{(h)}) = \prod_{j\ell} \prod_{uc} p(X_{iju}; \theta_{k\ell c}^{(h)})^{w_{j\ell}^{(h)} s_{uc}^{(h)}}$.

2) Generate the column partition $w_j^{(h+1)} = (w_{j1}^{(h+1)}, \ldots, w_{jL}^{(h+1)})|X, Z^{(h+1)}, S^{(h)}$ according to $w_j^{(h+1)} \sim \mathscr{M}(1, \widetilde{w}_{j1}, \ldots, \widetilde{w}_{jL})$, for all $1 \leq j \leq P$ and $1 \leq \ell \leq L$, where:

$$\widetilde{w}_{j\ell} = p(w_{j\ell} = 1 | \ X, Z^{(h+1)}, S^{(h)}; \theta^{(h)}) = \frac{\rho_\ell^{(h)} f_\ell(X_j | Z^{(h+1)}, S^{(h)}; \theta^{(h)})}{\sum\limits_{\ell'} \rho_{\ell'} f_{\ell'}(X_j | Z^{(h+1)}, S^{(h)}; \theta^{(h)})},$$

where $X_j = (X_{iju})_{iu}$ and $f_\ell(X_j | Z^{(h+1)}, S^{(h)}; \theta^{(h)}) = \prod_{ik} \prod_{uc} p(X; \theta_{klc}^{(h)})^{z_{ik}^{(h+1)} s_{uc}^{(h)}}$.

3) Generate the time cluster partition $s_u^{(h+1)} = (s_{u1}^{(h+1)}, \ldots, s_{uC}^{(h+1)})|X, Z^{(h+1)}, W^{(h+1)}$ according to $s_u^{(h+1)} \sim \mathscr{M}(1, \widetilde{s}_{u1}, \ldots, \widetilde{s}_{uC})$, for all $1 \leq u \leq U$ and $1 \leq c \leq C$, where:

$$\widetilde{s}_{uc} = p(s_{uc} = 1 | X, Z^{(h+1)}, W^{(h+1)}; \theta^{(h)}) = \frac{\delta_c^{(h)} f_c(X_u | Z^{(h+1)}, W^{(h+1)}; \theta^{(h)})}{\sum\limits_{c'} \delta_{c'} f_{c'}(X_u | Z^{(h+1)}, W^{(h+1)}; \theta^{(h)})},$$

where $X_u = (X_{iju})_{ij}$ and $f_c(X_u|Z^{(h+1)}, W^{(h+1)}; \theta^{(h)}) = \prod_{ik}\prod_{j\ell} p(X_{iju}; \theta^{(h)}_{klc})^{z^{(h+1)}_{ik} w^{(h+1)}_{j\ell}}$.

M step: in this step, $\mathcal{L}(q^*(Z,W,S);\theta^{(h)})$ is maximized with respect to θ, where:

$$\begin{aligned}\mathcal{L}(q^*(Z,W,S);\theta^{(h)}) &\simeq \sum_{Z,W,S} p(Z,W,S \mid X,\theta^{(h)}) \log \frac{p(X,Z,W,S \mid \theta)}{p(Z,W,S \mid \theta^{(h)})} \\ &\simeq E[\log(p(X, Z^{(h+1)}, W^{(h+1)}, S^{(h+1)} \mid \theta) \mid \theta^{(h)}] + \xi,\end{aligned}$$

where ξ is a constant term related to θ. This conditional expectation of the complete data log-likelihood can be written in a developed form as follows:

$$\begin{aligned}E[\log(p(X, Z^{(h+1)}, W^{(h+1)}, S^{(h+1)} \mid \theta) \mid \theta^{(h)})] = \sum_{i,k} z^{(h+1)}_{ik} \log \gamma_k + \sum_{j,\ell} w^{(h+1)}_{j\ell} \log \rho_\ell + \\ + \sum_{u,c} s^{(h+1)}_{u,c} \log \delta_c + \sum_{i,k}\sum_{j,\ell}\sum_{u,c} z^{(h+1)}_{ik} w^{(h+1)}_{j\ell} s^{(h+1)}_{uc} \log(p(X_{iju} \mid \theta_{klc})).\end{aligned} \tag{15}$$

The last term of the previous equation can be further developed as:

$$\begin{aligned}&\sum_{i,k}\sum_{j,\ell}\sum_{u,c} z^{(h+1)}_{ik} w^{(h+1)}_{j\ell} s^{(h+1)}_{uc} \log(p(X_{iju} \mid \theta_{klc})) = \\ &\sum_{i,k}\sum_{j,\ell}\sum_{u,c} z^{(h+1)}_{ik} w^{(h+1)}_{j\ell} s^{(h+1)}_{uc} \log\left[\frac{(\lambda^{(h)}_{klc})^{X_{iju}}}{X_{iju}!}\exp(-\lambda^{(h)}_{klc})\right] = \\ &\sum_{i,k}\sum_{j,\ell}\sum_{u,c} z^{(h+1)}_{ik} w^{(h+1)}_{j\ell} s^{(h+1)}_{uc} \left[X_{iju}\log(\lambda^{(h)}_{klc}) - \log(X_{iju}!) - \lambda^{(h)}_{klc}\right].\end{aligned} \tag{16}$$

Thanks to the previous equation, the parameter set $\theta^{(h+1)}$ can be estimated. The mixture proportions are updated as follows (proof in Appendix B):

$$\gamma^{(h+1)}_k = \frac{1}{N}\sum_i z^{(h+1)}_{ik}, \quad \rho^{(h+1)}_\ell = \frac{1}{P}\sum_j w^{(h+1)}_{j\ell}, \quad \delta^{(h+1)}_c = \frac{1}{U}\sum_u s^{(h+1)}_{uc}.$$

Moreover, the ML estimator of $\lambda_{k\ell c}$ is defined as follows (proof in Appendix C):

$$\lambda^{(h+1)}_{k\ell c} = \frac{R^{(h+1)}_{k\ell c}}{|\mathcal{A}_k||\mathcal{B}_\ell||\mathcal{D}_c|}, \qquad \forall(k,\ell,c).$$

Since assessing the convergence in stochastic inference algorithms is a challenging task, the algorithm runs for a certain number of iterations of the two steps and, to assess that it has been reached, we check both that the log-likelihood reached a plateau and that the values assumed by the model parameter values are stabilized during the iterations of the algorithm. We can obtain the final estimation of the parameter set $\widehat{\theta}$ by computing the mean from the sampled distribution, after the burn-in period.Finally, the optimal values for Z, W and S are estimated by the mode of their sampled distributions.

3.2 Model selection

Up to now, we have assumed that the number of row clusters (K), column clusters (L) and time clusters (C) was known. However, for real data sets, this assumption is of course unrealistic. For this reason, our purpose in this section is to define a model selection criterion that can automatically identify the optimal number of clusters that are appropriate for the data at hand. The model selection approach is considered. We propose to rely on ICL (Integrated Completed Likelihood, Biernacki et al. (2000)) to approximate the complete-data integrated log-likelihood. Hence, we derived the formulation of the ICL criterion for the model proposed above:

$$\begin{aligned} ICL(K,L,C) = \log p(X,\hat{Z},\hat{W},\hat{S};\hat{\theta}) - \frac{K-1}{2}\log N + \\ -\frac{L-1}{2}\log P - \frac{C-1}{2}\log U - \frac{KLC}{2}\log(NPU) \end{aligned} \tag{17}$$

The triplet $(\hat{K},\hat{L},\hat{C})$ that leads to the highest value for the ICL is considered as the most meaningful for those data. Alternative approaches may rely on greedy searches (Côme and Latouche, 2015; Keribin et al., 2017) which can be seen as less computationally expensive, but which operate sequentially and, therefore, cannot be parallelized.

4 Numerical experiments

The main purpose of this section is to highlight the most important features of dLBM over simulated datasets. We aim at demonstrating the validity of the inference algorithm and model selection criterion, presented in the previous

sections. Regarding the initialization of the algorithm, in all the reported experiments the partitions of rows, columns and slices were initialized with a k-means applied to the rows, columns or slices of the proper unfolding of tensor X. The first experiment consists in applying dLBM to an easy scenario to explain its main outputs. Then, the second example shows a model selection application on 25 simulated datesets. In the third experiment, we compare the performances of dLBM with some state-of-the-art methods in three simulated scenarios. In the fourth experiment, we compare the performances of dLBM with the same state-of-the-art methods of the previous experiment, on a simulation setup that differs from our model assumptions.

4.1 Introductory example

As a first example, we simulate a dataset with $K = 3$ groups of rows, $L = 2$ groups of columns and $C = 2$ groups of time clusters, with a level of sparsity $\tau = 0.97$. Table 1 shows the main features of this dataset.

N	P	U	τ	K	L	C	γ	ρ	δ
200	200	150	0.97	3	2	2	(0.15, 0.35, 0.55)	(0.2, 0.8)	(0.6, 0.4)

Table 1: Parameter values for the first simulated dataset.

We fit dLBM to simulated dataset with the actual values for K, L and C to show the ability of the model to fully recover the model parameters. Figure 4 shows the evolution of the complete data log-likelihood. As we can see the convergence is reached in less than 10 iterations. Figure 5 shows the evolution of the estimated mixture parameters $\hat{\gamma}$, $\hat{\rho}$ and $\hat{\delta}$ along the iterations of the SEM-Gibbs algorithm. Comparing the values reached by each line with the actual values of the model parameters showed in Table 1, we can observe that dLBM fully recovers the original values in few iterations. Moreover, Figure 6 shows a bar plot of the number of interactions between rows and columns of the array X for each time period, where the two different time clusters are identified by different colors. We can easily deduce that dLBM selects the two time clusters in a meaningful way in terms of level of counted interactions in each time cluster.
Figure 7 shows the value of the estimated intensity parameter $\hat{\lambda}$ for each cluster of rows and columns where different colors represent different time clusters. For instance, the algorithm detects that, in the first cluster of rows

and the first cluster of columns (Block(1,1)), there is an high intensity of interactions in both of the time clusters. Figure 8 and Figure 9, display the data structure before running dLBM and the reorganized incidence matrices one for each time cluster. To this end, rows and columns of the incidence matrix are permuted, thanks to the estimates $\hat{Z}$ and $\hat{W}$, in such a way that nearby rows (columns) belong to the same cluster of rows (columns). The blocks are also delimited by black dashed lines.
Finally, to evaluate the performance of the model in identifying the correct rows, columns and times partitions, we use the adjusted Rand index (ARI) Rand (1971) for all of the three variables. The adjusted Rand index, from a mathematical point of view, is closely related to the accuracy measure, however it is a commonly used method for evaluating clustering problems since it can be applied for measuring the similarity between two partitions even with different number of clusters and it is invariant to label switching. The closer the index is to 1, the more two label vectors are similar to each other. We compared the original matrices Z, W and S, with the estimates $\hat{Z}$, $\hat{W}$ and $\hat{S}$ given by the output of the dLBM. The model obtained an ARI index of 1 for rows, columns and times partitions. Thus, we can conclude that our algorithm perfectly identifies the composition of the original clusters.

4.2 Model selection experiment

In the previous experiment, we assumed to know the value of K, L and C. In this section we aim at validating the ICL criterion for model selection. To do that, 25 independent datasets are generated with the setup indicated in Table 2. dLBM is applied on those simulated datasets for values of K, L and C ranging from 1 to 6. The results are sorted according to the ICL values. Table 3 shows the percentage of selections by ICL criterion on 25 simulated datasets. The highlighted cell corresponds to the actual value of K, L and C. ICL succeeds 64% of the time to identify the correct model. It is worth to notice that, when ICL does not select the right combination of K, L and C, the wrongly selected models are close to the simulated one. In particular, 28% of the selections only differ from the actual model by one cluster, on one of the three cluster dimensions.

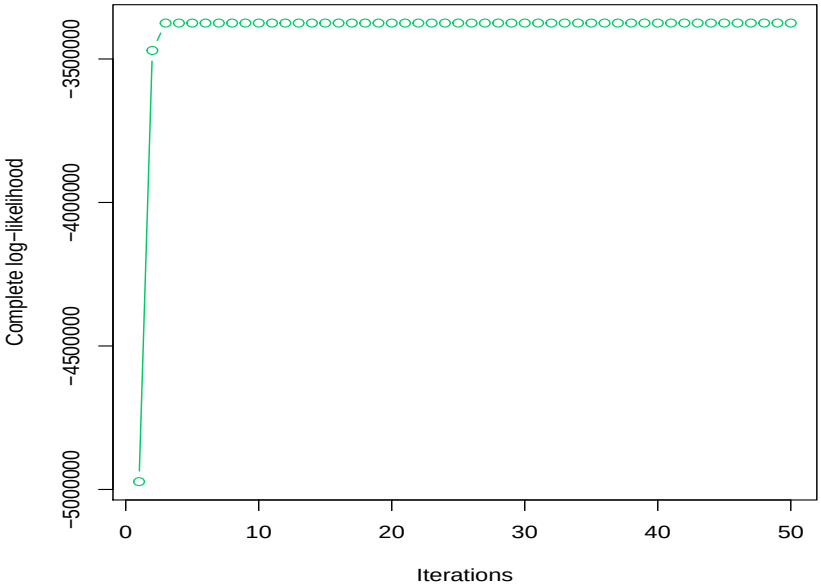

Figure 4: Complete data log-likelihood over the iterations of the dLBM algorithm.

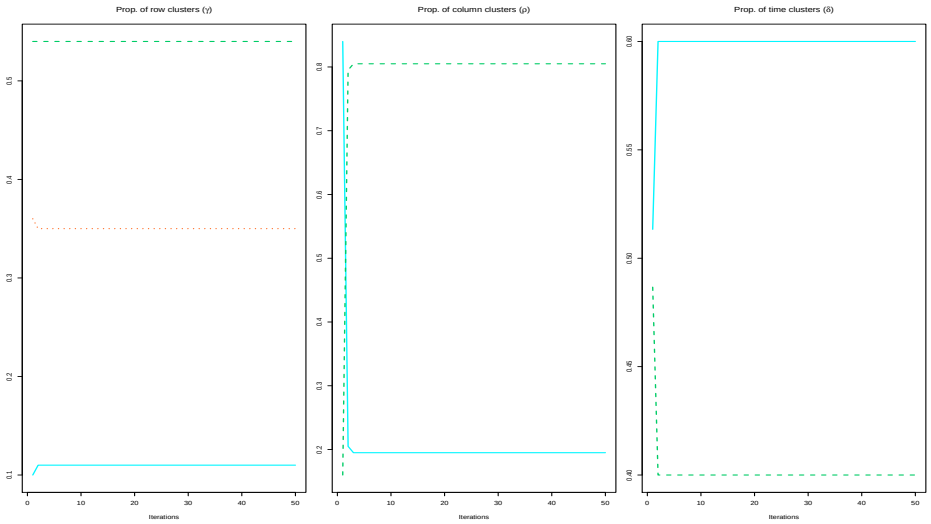

Figure 5: Estimates of the mixture parameters of the first simulated dataset.

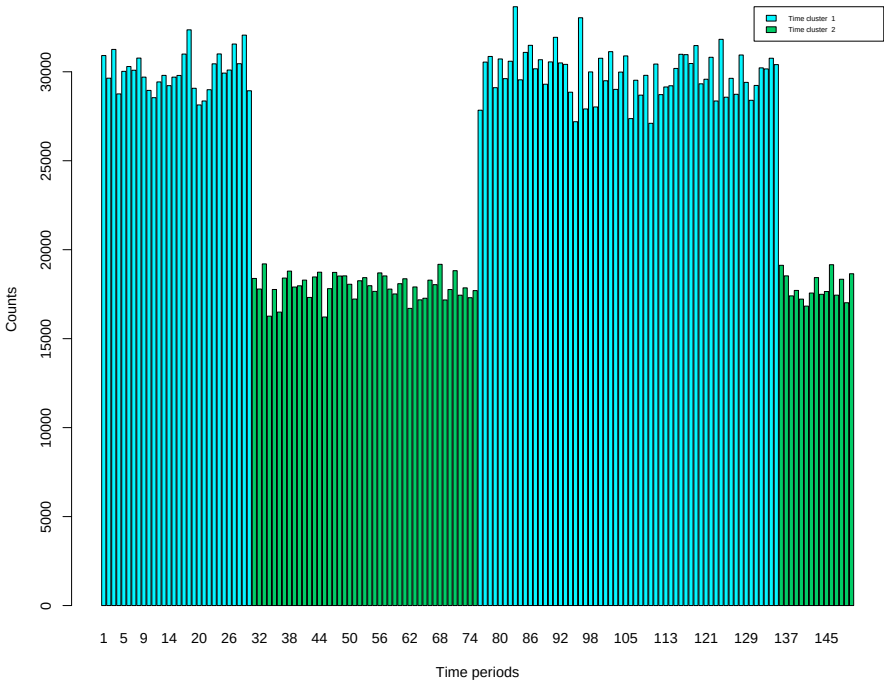

Figure 6: Time periods representation of the first simulated dataset, where different colors correspond to different time clusters.

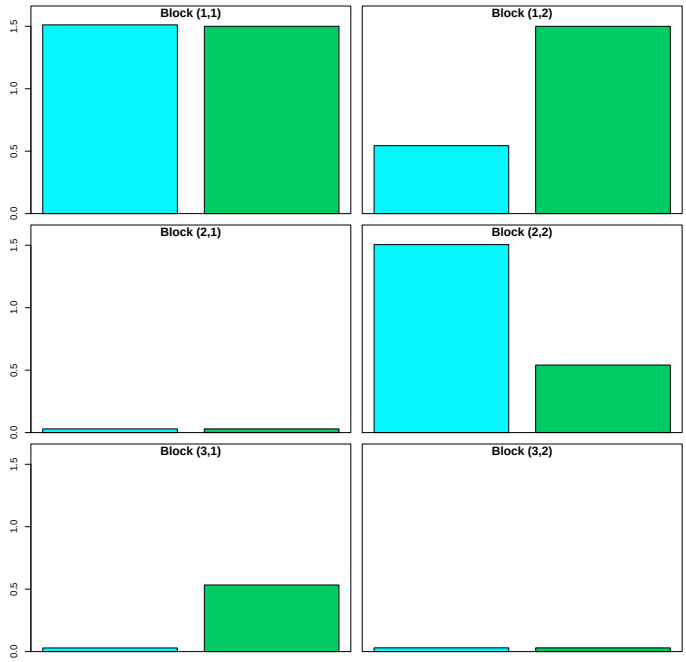

Figure 7: Estimated λ values for each cluster of rows and columns.

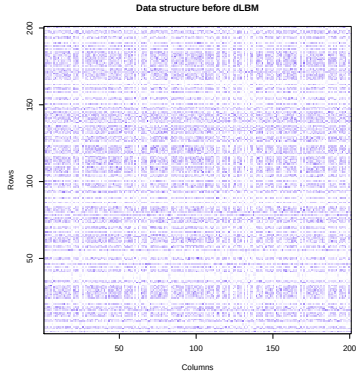

Figure 8: Unordered data structure before running dLBM.

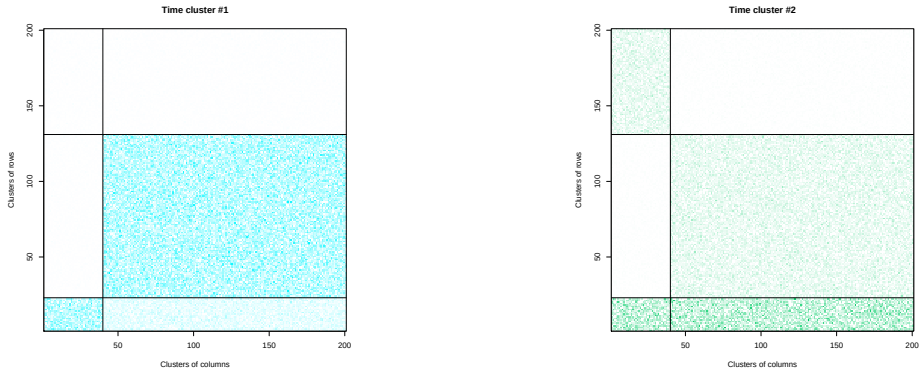

Figure 9: Reorganized incidence matrices, one for each time cluster, according to the estimates $\hat{Z}$ and $\hat{W}$. Nearby rows (columns) belong to the same cluster of rows (columns). The blocks are also delimited by black dashed lines.

N	P	U	τ	K	L	C	γ	ρ	δ
200	200	200	0.97	4	3	3	(0.2,0.4,0.1,0.3)	(0.4,0.3,0.3)	(0.25,0.3,0.45)

Table 2: Parameter values for the second simulated dataset.

C = 3

K/L	1	2	3	4	5	6
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	64	8	0	0
5	0	0	12	0	4	0
6	0	0	0	0	0	0

C = 4

K/L	1	2	3	4	5	6
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	8	0	0	0
5	0	0	0	4	0	0
6	0	0	0	0	0	0

Table 3: Model selection. Percentage of selections by ICL criterion on 25 simulated datasets. The highlighted cell corresponds to the actual value of K, L and C.

4.3 Benchmark study

The goal of this third experiment is to compare dLBM with some state-of-the-art methods in terms of recovering the data structure. dLBM is compared with TensorLBM (Boutalbi et al., 2020) where, in absence of the original code, we set the number of time clusters of dLBM equal to the number of time intervals, $C = U$, and with the Poisson LBM by making use of the `bikm1` package (Robert et al., 2020). Since LBM supports only two dimensions, we shrink the third dimension summing up alternatively on rows, columns and slices, obtaining respectively the Row_LBM, Col_LBM and Slice_LBM methods.

We chose to evaluate the results with the ARI index by comparing the resulting cluster partitions with the simulated ones. To make this comparison more complete, we defined four simulation scenarios ("Easy", "Medium", "Hard" and "Row_LBM"), detailed in Table 4. In particular, in the "Easy", "Medium" and "Hard" scenarios, the data are simulated according to the dLBM model using different parameters to gradually increase the difficulty. In order to also add a fair comparison with competitors, the "Row_LBM" scenario generates data from the Poisson LBM model (Robert et al., 2020) using the R package bikm1 of the authors. As detailed in Table 4, the simulation parameters for this additional simulation scenario are the ones used in the default example of the R package `bikm1`.

Table 5 displays the results of this comparison, in terms of average ARI values, reported with standard deviations. The dash indicates that no value is reported because the calculation is not allowed by the model.

Scenario	N	P	U	τ	K	L	C	γ	ρ	δ	λ
A - Easy	250	250	100	0.97	3	2	2	0.15,0.35,0.55	0.2,0.8	$0.6, 0.4$	Λ_A
B - Medium											Λ_B
C - Hard					4	3	3	0.2,0.4,0.1,0.3	$0.4, 0.3, 0.3$	$0.25, 0.3, 0.45$	Λ_C
D - Row_LBM	100	150	50	1	1	3	2	1	$0.33, 0.33, 0.33$	$0.5, 0.5$	Λ_D

Table 4: Parameter values for the four simulation senarios (see Appendix C for details about Λ_A, Λ_B, Λ_C and Λ_D).

Scenario A - **Easy**				Scenario B - **Medium**			
	ARI_Rows	ARI_Cols	ARI_Slices		ARI_Rows	ARI_Cols	ARI_Slices
dLBM	1 ± 0	1 ± 0	1 ± 0	dLBM	0.89 ± 0.17	1 ± 0	1 ± 0
TensorLBM	0.8 ± 0.3	1 ± 0	-	TensorLBM	0.74 ± 0.18	1 ± 0	-
Row_LBM	-	0.12 ± 0.21	0.96 ± 0.2	Row_LBM	-	0.13 ± 0.21	0.12 ± 0.23
Col_LBM	0.1 ± 0.21	-	0.92 ± 0.2	Col_LBM	0.09 ± 0.21	-	0.15 ± 0.23
Slice_LBM	0.09 ± 0.2	0.13 ± 0.22	-	Slice_LBM	0.1 ± 0.2	0.14 ± 0.21	-
Scenario C - **Hard**				Scenario D - **Row_LBM**			
	ARI_Rows	ARI_Cols	ARI_Slices		ARI_Rows	ARI_Cols	ARI_Slices
dLBM	0.79 ± 0.19	0.68 ± 0.22	0.63 ± 0.18	dLBM	-	0.96 ± 0.13	1 ± 0
TensorLBM	0.64 ± 0.21	0.71 ± 0.19	-	TensLBM	-	0.94 ± 0.14	0.85 ± 0.19
Row_LBM	-	0.09 ± 0.14	0.09 ± 0.15	Row_LBM	-	1 ± 0	1 ± 0
Col_LBM	0.12 ± 0.14	-	0.12 ± 0.15	Col_LBM	-	-	0.94 ± 0.11
Slice_LBM	0.2 ± 0.18	0.25 ± 0.22	-	Slice_LBM	-	1 ± 0	-

Table 5: Co-clustering results for dLBM, TensorLBM, and LBM applied respectively by summing up the rows (Row_LBM), the columns (Col_LBM) and the slices (Slice_LBM) on 25 simulated data according to the four scenarios. Average ARI values are reported with standard deviations.

In the "Easy" situation, dLBM works perfectly. Also TensorLBM provides excellent results, even though calculated only on rows and columns. Row_LBM and Col_LBM only give good results on one dimension, while Slice_LBM produces extremely low results.
In "Medium" and "Hard" situations, dLBM continues to obtain excellent results, although not perfect, due to the increasing complexity of the proposed situations. The three other LBM models perform poorly, while TensorLBM obtains rather high ARI values. Specifically, in the "Hard" situation the ARI value of TensorLBM on the columns is slightly higher than that of dLBM, even if the one calculated on the rows partitions is lower.
Finally, regarding the "Row_LBM" scenario, we can first observe that the results of Row_LBM are perfect, as expected, but dLBM demonstrates also to have very good performances in this less favorable simulation, conversely

to the other competitors.

4.4 Robustness to model assumptions

The goal of this fourth experiment on simulated data is to test the performance of dLBM and compare it to its competitors when data are not simulated according to the model assumptions. Specifically, we decided to simulate the data from a negative binomial distribution. The negative binomial distribution is a discrete probability distribution that models the number of successes in a series of iid Bernoulli trials before a given number of failures, r. The probability mass function of the negative binomial is given by:

$$f(k, r, p) = \binom{k+r-1}{k} \cdot (1-p)^r \cdot p^k = \frac{\Gamma(k+r)}{k!\Gamma(r)}(1-p)^r p^k, \tag{18}$$

where k is the number of successes and p is the probability of success. When modeling counts data the negative binomial distribution is often a valid alternative to the Poisson one, because it allows the mean and the variance to be different: Mean: $\lambda = \frac{pr}{1-p}$; Variance: $= \frac{pr}{(1-p)^2} = \lambda + \frac{1}{r}\lambda^2$.

A particular property of the negative binomial distribution is that it converges to the Poisson distribution, with expected value μ, when $r \to \infty$.

To accurately evaluate the performance of dLBM in comparison with its competitors we simulate with the negative binomial distribution 25 datasets for each value of r equal to 0.05, 0.1, 0.5, 0.8, 1, 5 and 10, while keeping the values of λ unchanged to the ones of Scenario A in Section 4.3.

Figures 10, 11 and 12 report the results of the experiment, depicting line plots of the values taken by the ARI index in the partitions of rows, columns, and slices, respectively. The x-axis depicts the values of r based on which the data were simulated, and each color represents a different model. The models considered are the same as those described in Section 4.3: Row LBM, ColLBM, SliceLBM, and TensorLBM. The ARI values shown in these graphs refer to the average values reached by the index during the 25 simulations performed for each r value. Looking at these results we can observe that for values of r very close to zero, all models have difficulties in finding the correct cluster partition. This is due to the fact that when r is small (e.g. $r = 0.05, 0.1$), a negative binomial distribution is more spread than a Poisson distribution with the same mean and therefore the different clusters are extremely difficult to distinguish. For slightly higher values of r (e.g. $r = 0.5, 0.8$) dLBM

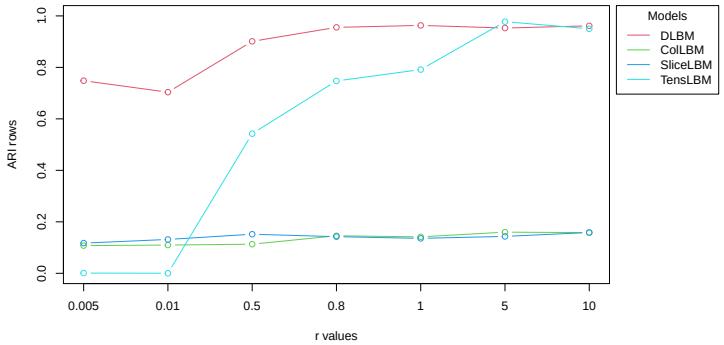

Figure 10: Evolution of the row ARI according to r for the different methods.

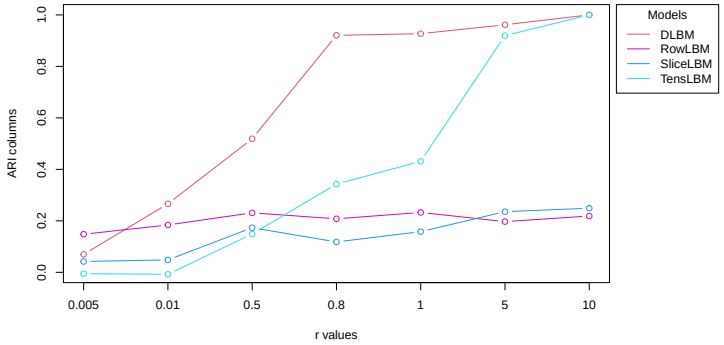

Figure 11: Evolution of the column ARI according to r for the different methods.

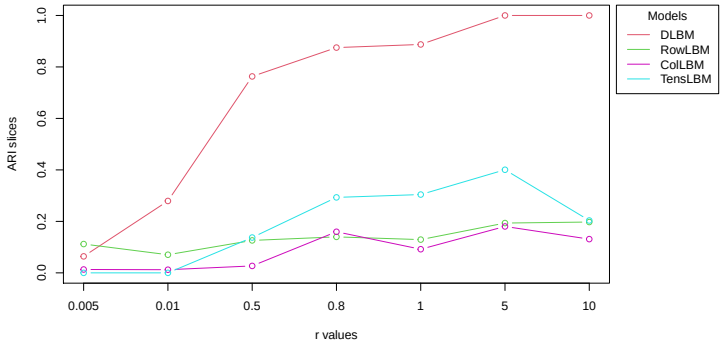

Figure 12: Evolution of the slice ARI according to r for the different methods.

outperforms its competitors, achieving excellent ARI values. Whereas,for sufficiently large values of r (e.g. $r = 5, 10$), the block distributions start to be enough different and dLBM managed to find the correct cluster partitions, with an ARI very close to 1 in all the dimensions. In this case, in fact, the results tend to be very similar to those of Scenario A in Section 4.3. It may be of interest to notice that also TensorLBM managed to have high values of ARI, for row and column partitions, although it needs a higher value of r to perform well. However, looking at the partition of slices we can see that the performance is poor, due to the fact that the assumption TensorLBM relies on is that the number of slice clusters is equal to the number of slices themselves, i.e. $C = U$.

5 Analysis of the adverse drug reaction dataset

This section focuses now on the application of dLBM to a large-scale pharmacovigilance dataset, with the aim of illustrating the potential of the tool for such studies.

5.1 Protocol and data

This section considers a large dataset consisting of ADR data collected by the Regional Center of Pharmacovigilance (RCPV), located in the University Hospital of Nice (France). The center covers an area of over 2.3 million inhabitants and receives notifications about ADRs from different channels: a website[1] form that everyone can freely fill and send, phone calls, emails, medical visits at the hospital units, etc. A time horizon of 10 years is considered, from January 1st, 2010 to September 30th, 2020, the unity measure for time intervals is a month ($\Delta_u = \Delta = 1$ month). The overall dataset is made of by 44,269 declarations, for which the market name of the drug, the notified ADR, the channel used for the declaration and its origin, as well as an identification number and the reception date are reported. To prevent the same medicine from being considered more than once if reported under slightly different names, we decided to use the international nonproprietary name (INN) of the drug (to simplify the comprehension, the INNs would be referred as drugs for the rest of the study). Moreover, we only considered molecules and ADRs that were notified more than 10 times over the 10 years. The

[1] `https://signalement.social-sante.gouv.fr`

resulting dataset contains 542 drugs, 586 ADRs and 129 months with 13,363 non-zero entries.
Looking at Figure 1, it can easily be noticed that, during the year 2017, there is an extremely uncommon behavior in the progress of notifications to the pharmacovigilance center. In fact, in that year an unexpected rise of reports for ADRs happened concerning two specific drugs: Mirena® and Lévothyrox®. Mirena® has been available in Europe since 1995. This birth control product contains a hormone called levonorgestrel. In 2017, concerns regarding ADRs associated with the use of levonorgestrel releasing intra-uterine device (IUD) started to grown with a media coverage peak occurred in May 2017, which resulted in a massive wave of ADRs reports from patients to French RCPVs (Langlade et al., 2019).
Also, Lévothyrox® has been marketed in France for about 40 years as a treatment for hypothyroidism and, in 2017, a new formula was introduced on the market. The Lévothyrox® case had an extremely high media coverage in France: the RCPVs received 18,241 reports of Lévothyrox® ADRs in 2017 only. Lévothyrox® spontaneous reports represent almost the 90% of all the spontaneous notifications that the Nice center received from patients in 2017. This phenomenon has been fully described in a recent article of Viard et al. (2019).
From Figure 1, one can understand the difficulty to work with such data which contain signals of very different amplitude. Indeed, behind those very visible effects, many ADR signals need to be detected for obvious public health reasons. In particular, those data also contain ADR reports regarding Médiator®, which is here far less visible than Lévothyrox® and Mirena, but also led to many avoidable serious cardiovascular diseases. This is why, we expect dLBM to be a useful tool to reveal such hidden signals.

5.2 Summary of the results

Remembering that our aim is to find an underling latent structure in our dataset by applying co-clustering on the three dimensions of the array X with dimension $542 \times 586 \times 129$, we have run dLBM for different values of K, L and C. Specifically, we tested rows (here drugs), columns (here ADRs) and times groups ranging from 1 to 12. The ICL criterion identified the optimal values for the triplet $(\hat{K}, \hat{L}, \hat{C})$ as: $\hat{K} = 7$, $\hat{L} = 10$, $\hat{C} = 6$, with a running time of about 6 hours. The process has been parallelized on 8 cores on a MacBook Pro, 2020, with a processor of 2,3 GHz Quad-Core Intel Core i7

and 16 GB of RAM.
Figure 13 shows the frequency of the declarations received by the RCPV from 2010 to 2020, sorted by month, where the colors represent the identified time clusters. Figure 14 displays the estimated intensity function representation. In particular, this figure is very helpful for giving an overview of the relationships between drug clusters and ADR clusters and how they evolve over time. The colors refer to different time clusters and the brighter the color, the stronger the relation (i.e. the expected number of notifications in the time unit) between drug cluster and ADR cluster. Finally, Figure 15 shows more specifically the evolution of the relationship between drug clusters and ADR clusters over time. In fact, each panel represents a cluster of drugs and within them each line identifies a cluster of ADRs and its intensity changes over time.

Time clusters Starting from the analysis of the time clusters, one can easily notice on Figure 13 that the segmentation proposed by the algorithm confirms our knowledge about the previous mentioned health scandals while revealing a time structure more complex than expected. In fact, while cluster 1 and cluster 2 include various time intervals, cluster 3 clearly refers to the health crisis due to the Mirena® scandal while cluster 4 relates to the peak period in the Lévothyox® crisis. Time clusters 5 and 6 refer to the final stage of the Lévothyrox® crisis, when generics were introduced to the market. It is worth noticing that without the dLBM application it would have been impossible to detect the presence of other health scandal just before the one of Lévothyrox®. In fact, looking at Figure 13, one can see that the increase of declarations during the Mirena® health crisis are completely masked by the Lévothyrox® ones.

Drug clusters The clusters of drugs identified by the algorithm are also coherent with retrospective knowledge and adequately represent the variety of drugs present in the dataset. In particular, cluster 1, cluster 6 and cluster 7 are very specific, with one element only: they correspond respectively to lévothyroxine (Lévothyrox® and generics), benfluorex (Médiator®) and lévonorgestrel (Mirena®). It is worth noticing that Médiator® [2] was involved (like Lévothyrox® and Mirena®) in an important health scandal in 2009-

[2] `https://www.ansm.sante.fr/Dossiers/Mediator-R/Mediator-R-et-accompagnement-des-personnes/(offset)/0`

2010. Moreover, cluster 2 contains the five most frequently reported drugs and cluster 5 contains other common drugs, while cluster 4 is very large and heterogeneous, with drugs that are rarely reported and finally cluster 3 contains drugs that cause bleeding.

ADR clusters Concerning the clusters of ADRs, cluster 3 (e.g. coma, confusion, hepatic cytolysis, etc) and cluster 8 (e.g. agitation, agranulocytosis, arthralgia, etc.) contain the most frequently notified ADRs. Cluster 1 contains recurring ADRs (e.g. sweats, transient ischemic accident, lactic acidosis, etc.) but less than the other two previously mentioned. Cluster 2 (e.g. anemia, hemorrhagic stroke) and cluster 4 (e.g. hemorrhagic schock, deglobulisation, etc.) respectively include the most and the less frequent bleeding related ADRs. Cluster 7 is composed of ADRs clearly related to Lévothyrox® and Mirena® (e.g hair loss, cramps, insomnia, etc.). In cluster 10 there are general ADRs, although it contains some ADRs specifically related to Lévothyrox® and Médiator® (e.g. respectively abnomral TSH, valvular disease, etc.). Finally, cluster 5, 6 and 9 contain more general and nonspecific ADRs.

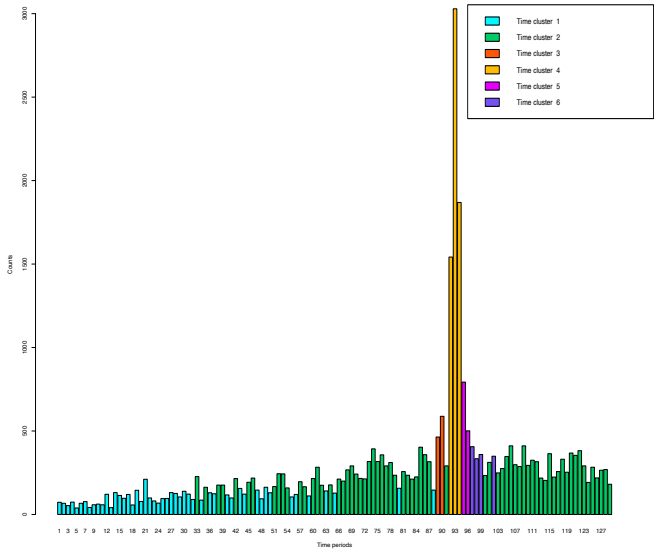

Figure 13: Number of declarations received by the pharmacovigilance center from 2010 to 2020, sorted by month, where the colors represent the time clusters.

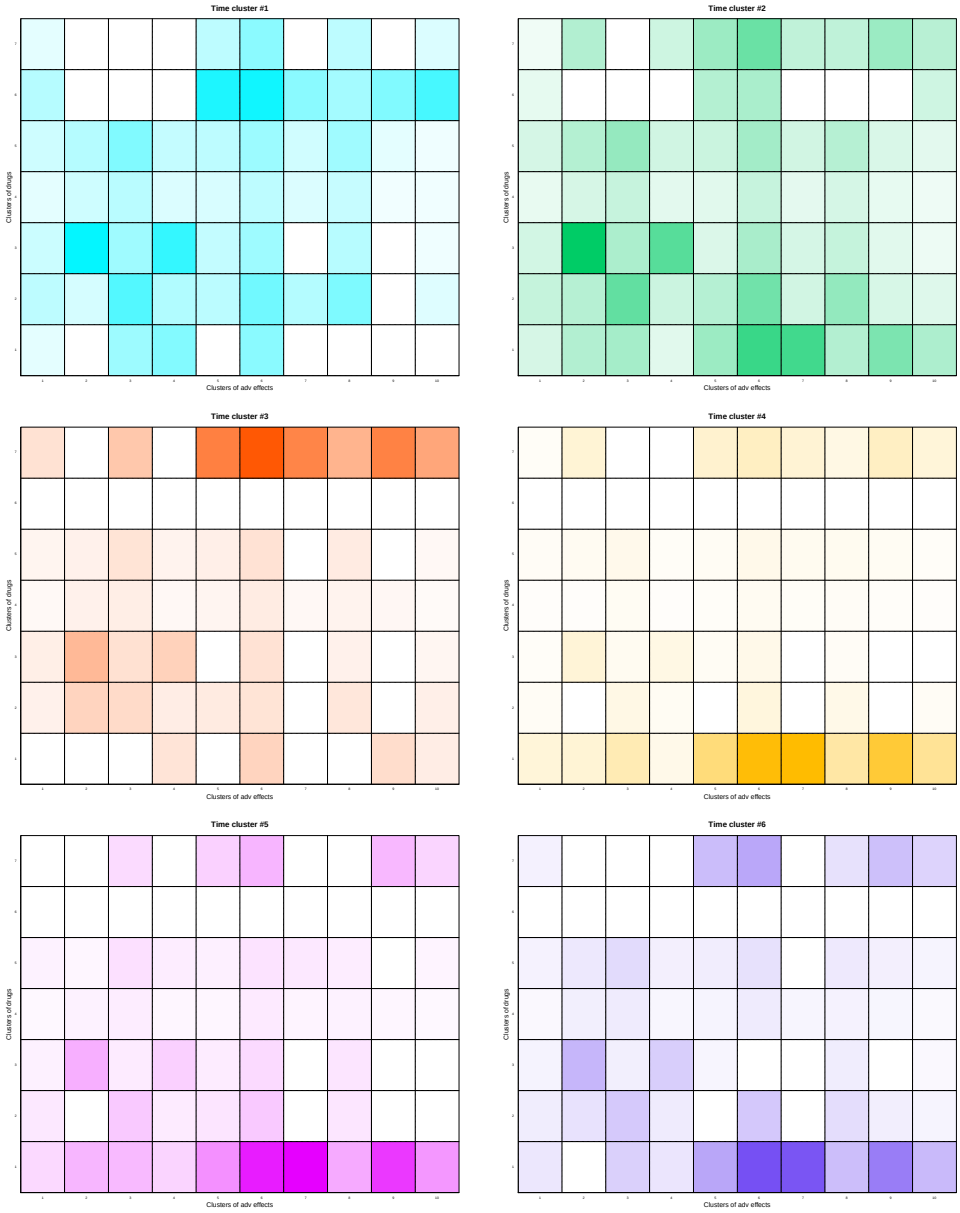

Figure 14: Evolution of the relation between each drug cluster and the all ADR clusters over time. Each color corresponds to a different cluster of adverse drug reaction.

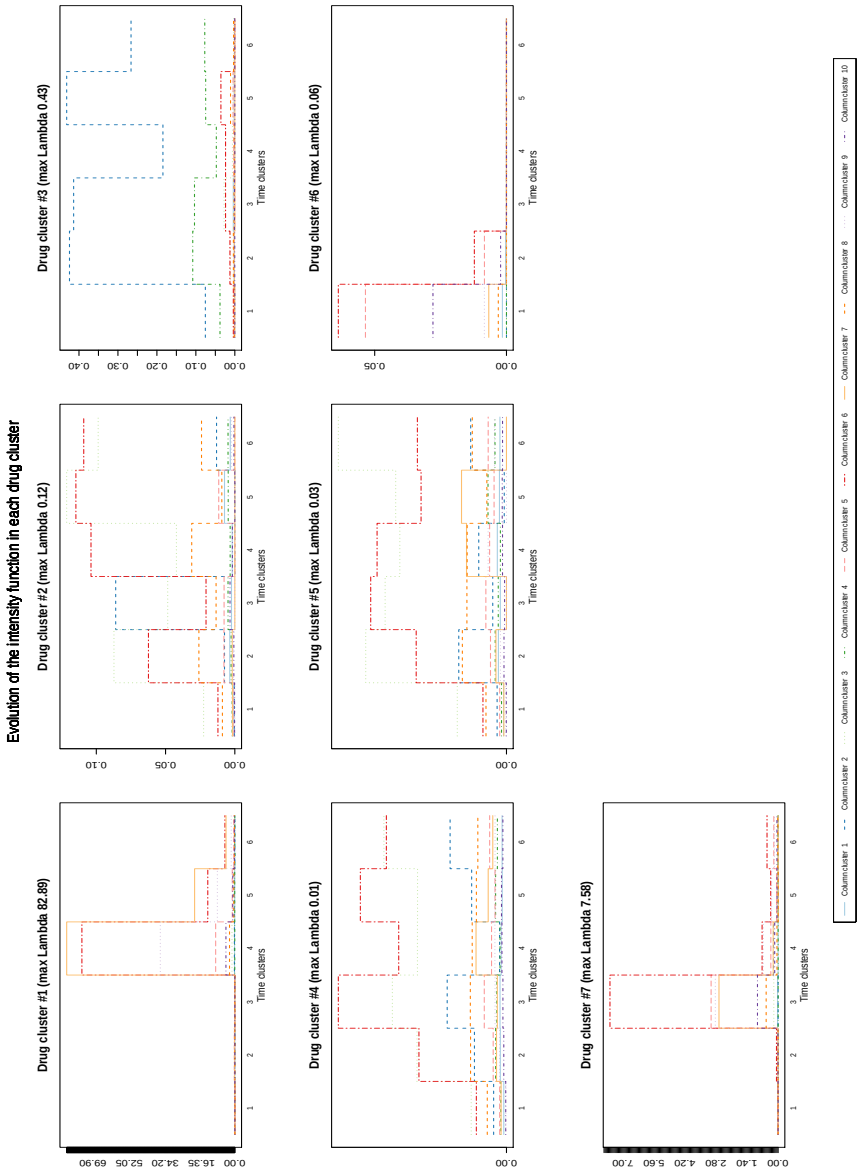

Figure 15: Evolution of the relation between each drug cluster and the all ADR clusters over time. Each color corresponds to a different cluster of adverse drug reaction.

5.3 Detailed results

Time clusters Figure 14 is a graphical representation of the estimated intensity functions. It gives a clear idea about the relationships between clusters of molecules and clusters of ADRs, with respect to time clusters. In particular:

- Time cluster 1: here one can notice the presence of all the drug clusters, with different levels of intensity. The peculiarity of this cluster lies in the strong presence of cluster 6 of drugs which gradually disappears in subsequent temporal clusters. In fact, it contains the drug benfluorex (Médiator®) which in 2010 was involved in a major health scandal and it has strong interactions with clusters 5, 6 and 10 of ADRs. It is worth noticing that dLBM managed to highlight this peculiarity that cannot be detected by simply looking at Figure 1. In the first time cluster it can also be noticed a strong relation between the cluster 3 of drugs (drugs that causes bleeding) and clusters 2 and 4 of ADRs which is coherent.

- Time cluster 2: the presence of Médiator® decreases while the interactions between the drugs that cause bleeding (cluster 3) and clusters 2 and 4 of ADRs is still strong. That cluster represent the actual profile of notifications received by RCPV. Similarly, the second cluster of drugs (the most frequently used) appears to have ADRs in almost all clusters, especially the third. We also notice the presence of Lévothyrox® (cluster 1) with ADRs especially in clusters 6 and 7.

- Time cluster 3: this cluster includes two months only: 05-06/2017. This period refers to the Mirena® scandal (cluster 7) with ADRs in clusters 5, 6, 7 and 9. They are not very specific but it may suggest a hormonal cause. In this matrix all the drug clusters are present (even though with a lower intensity with respect to Mirena®), with the exception of Médiator® which is not reappearing subsequently.

- Time cluster 4: it refers to the Lévothyrox® peak going from August to October 2017. Unlike previous clusters where most drugs clusters were present, this cluster only recognizes 2 drugs: Lévothyrox® (mainly) and Mirena® (weakly). Consequently, the interactions that stand out are those of Lévothyrox® with ADRs in clusters 6, 7 and 9. They are not very specific but coherent with the statements received by RCPV.

- Time cluster 5: it refers to 11-12/2017 and it is characterized by a reduction of Lévothyrox® declarations. Compared to the previous cluster, we note a reappearance of other drug clusters even if the intensities remain low compared to time cluster 1 and 2. The drugs/ADRs combinations remain those of the previous time cluster.

- Time cluster 6: it refers to the 1st semester 2018, which corresponds to end of Lévothyrox® crisis. Globally it is similar to the two previous time clusters with some small variations on the intensities of the drug/ADRs pairs.

Drug clusters For a more in-depth analysis regarding the evolution of drug clusters over time and their interactions with the clusters of ADRs, we can refer to Figure 15. The following remarks derive:

- Drug cluster 1: this cluster refers to Lévothyrox® and its generics. There are almost no declared effects during the first three time clusters, from the fourth time cluster we observe a peak of declarations which corresponds to the start of the Lévothyrox® crisis, especially for ADRs in cluster 6, 7 and 9 and to a lesser extent on 5. These four clusters recognized by dLBM, include all of the ADRs described during Lévothyrox® crisis, namely hormonal ADRs (weight gain), general ADRs (fatigue, cramps) and neuro-psychic ADRs (anxiety, irritability, sleep disturbances). Time cluster 5 marks a decrease in Lévothyrox® reports in terms of numbers but the ADRs profile remains similar to the previous time cluster. It should be noticed that generics of Lévothyrox® began to be available from mid-October 2017, which could explain this decrease in the number of reports: patients started to have therapeutic alternatives. Finally, time cluster 6 represents the end of the Lévothyrox® crisis with a clear decrease in the number of reports.

- Drug cluster 2: this cluster includes drugs that are very frequently prescribed, the ADRs profile is globally constant over time with a predominance of clusters 3, 6 and 8, with variations in terms of proportions according to the time clusters. Cluster 3 of ADRs corresponds to frequent and generally serious ADRs. Cluster 8 also includes ADRs that are generally serious but a little less often reported than cluster 3. Cluster 6 corresponds to general ADRs that can be found with many other drugs (especially Lévothyrox®).

- Drug cluster 3: this cluster includes coagulation drugs whose main ADRs are bleeding, hence the predominance of cluster 2 and 4. The application of dLBM led us to identify, from the temporal point of view, 3 interesting events: the increase in ADRs between time clusters 1 and 2, a significant decrease in the number of declared ADRs in time cluster 4 (Lévothyrox® crisis) and a marked regression of cluster 2 in time cluster 6 (late Lévothyrox®) but without cluster 4 being affected.

- Drug cluster 4: this cluster includes a fairly large set of drugs declared relatively frequently and commonly prescribed, but disparate in terms of their therapeutic uses or their ADRs profile. However, we observe a predominance of ADRs of cluster 3 and 6 (general ADRs). At the temporal level, we observe an overall decrease in ADRs in time cluster 4 (Lévothyrox® crisis).

- Drug cluster 5: as for drug cluster 4, this cluster includes many heterogeneous drugs. The ADRs profile is similar to that of drug cluster 4, which is coherent with the fact that these two clusters are similar.

- Drug cluster 6: this is the Médiator® cluster, the ADRs are concentrated in time clusters 1 and 2 with a decrease in the number of ADRs in the second one. This is coherent with the history of the drug (Médiator® scandal happened in 2009 with withdrawal of the market). Regarding the profile of ADRs, we can notice a predominance of clusters 6, 5, and 10.

- Drug cluster 7: this is the Mirena® cluster, ADRs declarations predominate in time cluster 3 (Mirena® crisis) then drop drastically. At the level of ADRs profile, cluster 6 predominates (general ADRs), then come cluster 5, 7 and 9.

5.4 Discussion

In this application to pharmacovigilance, dLBM proved to be a very useful tool for identifying phenomena that would have been difficult to detect otherwise, even by an expert eye. In fact, dLBM revealed that in addition to Lévothyrox® health crisis, which was the one with the widest media coverage, two other major events have occurred. The first one concerning Médiator®, which took place in 2009-2010, and the second one concerning Mirena®, which took place

in the first half of 2017. In addition, dLBM was also able to put in light some unexpected variations of notifications such as an under-notification of bleeding related ADRs during Lévothyrox® crisis. Bleeding related ADRs were expected to be constant over time because of the follow-up made the RCPV to monitor ADRs of direct oral anticoagulants (DOAs), a recent class of anticoagulant. However, the Lévothyrox® crisis has caused such an overload of work that the DOAs follow-up have been temporarily interrupted. Another thing that dLBM has highlighted is the existence of 3 different phases during the Lévothyrox® crisis corresponding to the reporting peak, the marketing period of generics and the end of the crisis, respectively. Those phases were not noticed by the RCPV staff during the Lévothyrox® crisis. In general, we can conclude that dLBM could be extremly useful as a routine tool for signal detection, since it might help health professionals to identify structural changes or patterns of interest and, perhaps, prevent some of the consequences a health crisis can lead to.

6 Conclusion

This work is born out of the need to analyze and summarize observations and features of a dynamic count matrix in a simultaneous way. We have proposed a dynamic co-clustering technique, with the purpose of simultaneously performing clustering of rows, columns and slices (time dimension). We have proposed to consider a dynamic framework because it is of great interest to look for structural changes in the way clusters interact with each other along the time. To this end, we have introduced a generative model, named dynamic latent block model (dLBM). The dynamic time modeling relies on non-homogeneous Poisson processes, with a latent partition of time intervals. Inference is done using a SEM-Gibbs algorithm and the ICL criterion is used for model selection. The performance of dLBM was evaluated through applications to several simulated data scenarios and compared with that of competing methods. Then, dLBM was fit to a large-scale data set supplied by the Regional Center of Pharmacovigilance of Nice (France). In this context, dLBM provided meaningful segmentations of drugs, adverse drug reactions and time. Its potential use by medical authorities for identifying meaningful pharmacovigilance patterns looks very promising. As a further work, it would be of interest to develop a online inference algorithm for dLBM, in order to be used as a real-time tool working on the flow of ADR declarations. This

however will require to be able to also do the model selection in an online mode, which will be probably the most difficult question.

Acknowledgements

This work has been supported by the French government, through the 3IA Côte d'Azur, Investment in the Future, project managed by the National Research Agency (ANR) with the reference number ANR-19-P3IA-0002.

A Estimation of the mixture proportions

The proof about how to obtain the updated mixture proportions is only shown for the estimation of parameter $\gamma_k^{(h+1)}$ because for the estimation of the other parameters, ρ and δ, the procedure is similar:

$$p(Z|\gamma) = \mathcal{L}(\gamma; Z) = N! \prod_{i=1}^{N} \prod_{k=1}^{K} \frac{\gamma_k}{z_{ik}!};$$

$$\ell(\gamma_k; z_{ik}^{(h+1)}) = \log \mathcal{L}(\gamma_k, z_{ik}^{(h+1)}) = \log \left(N! \prod_{i=1}^{N} \prod_{k=1}^{K} \frac{\gamma_k}{z_{ik}^{(h+1)}!} \right) =$$

$$= \log N! + \sum_{i=1}^{N} \sum_{k=1}^{K} z_{ik}^{(h+1)} \log \gamma_k - \sum_{i=1}^{N} \sum_{k=1}^{K} \log z_{ik}^{(h+1)}!$$

For a constrained maximization of this quantity we employ the Lagrange Multipliers, taking into account the constraint $\sum_{k=1}^{K} \gamma_k = 1$.

$$\mathcal{L}(\gamma_k; \lambda) = \ell(\gamma_k; z_{ik}^{(h+1)}) + \lambda(1 - \sum_{k=1}^{K} \gamma_k)$$

$$\frac{\partial \mathcal{L}(\gamma_k; \lambda)}{\partial \gamma_k} = \frac{\partial \ell(\gamma_k; z_{ik}^{(h+1)})}{\partial \gamma_k} + \frac{\partial \lambda(1 - \sum_k \gamma_k)}{\partial \gamma_k} = 0$$

$$\frac{\partial \sum_{i=1}^{N} \sum_{k=1}^{K} z_{ik}^{(h+1)} \log \gamma_k}{\partial \gamma_k} - \lambda \frac{\partial \sum_{k=1}^{K} \gamma_k}{\partial \gamma_k} = 0$$

$$\frac{\sum_{i=1}^{N} z_{ik}^{(h+1)}}{\gamma_k} - \lambda = 0$$

$$\sum_{i=1}^{N} z_{ik}^{(h+1)} = \lambda \gamma_k \Rightarrow \frac{\sum_{i=1}^{N} z_{ik}^{(h+1)}}{\lambda} = \gamma_k$$

Since λ is equal to N:
$\sum_{k=1}^{K} \sum_{i=1}^{N} \frac{z_{ik}^{(h+1)}}{\lambda} = \sum_{k=1}^{K} \gamma_k \Rightarrow \frac{1}{\lambda} \sum_{k=1}^{K} \sum_{i=1}^{N} z_{ik}^{(h+1)} = 1;$
we can conclude that the estimation of $\gamma_k^{(h+1)}$is the following:

$$\gamma_k^{(h+1)} = \frac{1}{N} \sum_{i=1}^{N} z_{ik}^{(h+1)}$$

B Maximum likelihood estimator of $\lambda_{k\ell c}$

The maximum likelihood estimator of $\lambda_{k\ell c}$ is obtained through the following process:

$$\log L(\lambda|X,Z,W,S) = \sum_{k=1}^{K}\sum_{\ell=1}^{L}\sum_{c=1}^{C}(R_{k\ell c}\log\lambda_{k\ell c} - |\mathcal{A}_k||\mathcal{B}_\ell||\mathcal{D}_c|\lambda_{k\ell c} + c)$$

where c is a constant that includes all the terms that does not depend on λ.

$$\frac{\partial \log \mathcal{L}(\lambda|X,Z,W,S)}{\partial\lambda} = \frac{R_{k\ell c}}{\lambda_{k\ell c}} - |\mathcal{A}_k||\mathcal{B}_\ell||\mathcal{D}_c| = 0 \Rightarrow \widehat{\lambda}_{k\ell c} = \frac{R_{k\ell c}}{|\mathcal{A}_k||\mathcal{B}_\ell||\mathcal{D}_c|}$$

C Intensity functions in the three scenarios

From Table 4, the scenarios "Easy" and "Medium" may look the same. However, the main difference between the two scenarios is the value assumed by the intensity function λ. The values of this parameter in the three different scenarios are:

- Scenario A - Easy: $\lambda = \Lambda_A$
  $\Lambda_A[,,1] = \begin{bmatrix} 50 & 18 \\ 1 & 1 \\ 1 & 50 \end{bmatrix}$; $\Lambda_A[,,2] = \begin{bmatrix} 50 & 50 \\ 18 & 1 \\ 1 & 18 \end{bmatrix}$
- Scenario B - Medium: $\lambda = \Lambda_B$
  $\Lambda_B[,,1] = \begin{bmatrix} 1 & 1 \\ 1 & 7 \\ 7 & 20 \end{bmatrix}$; $\Lambda_B[,,2] = \begin{bmatrix} 20 & 20 \\ 7 & 1 \\ 1 & 7 \end{bmatrix}$
- Scenario C - Hard: $\lambda = \Lambda_C$
  $\Lambda_C[,,1] = \begin{bmatrix} 70 & 12 & 1 \\ 35 & 1 & 35 \\ 1 & 70 & 12 \\ 12 & 35 & 70 \end{bmatrix}$; $\Lambda_C[,,2] = \begin{bmatrix} 35 & 70 & 12 \\ 70 & 70 & 70 \\ 12 & 1 & 35 \\ 1 & 70 & 1 \end{bmatrix}$; $\Lambda_C[,,3] = \begin{bmatrix} 12 & 70 & 35 \\ 35 & 12 & 70 \\ 70 & 35 & 12 \\ 12 & 1 & 35 \end{bmatrix}$
- Scenario D - Row_LBM: $\lambda = \Lambda_D$
  $\Lambda_D[,,1] = \begin{bmatrix} 1 & 6 & 4 \end{bmatrix}$; $\Lambda_D[,,2] = \begin{bmatrix} 1 & 7 & 1 \end{bmatrix}$

D Data structure representation

Fig. 16 shows a representation of the interactivity patterns between all the drugs and adversarial effects at any given time interval. Each panel represents a time interval and the size and the color of the points depend on the number of declarations received.

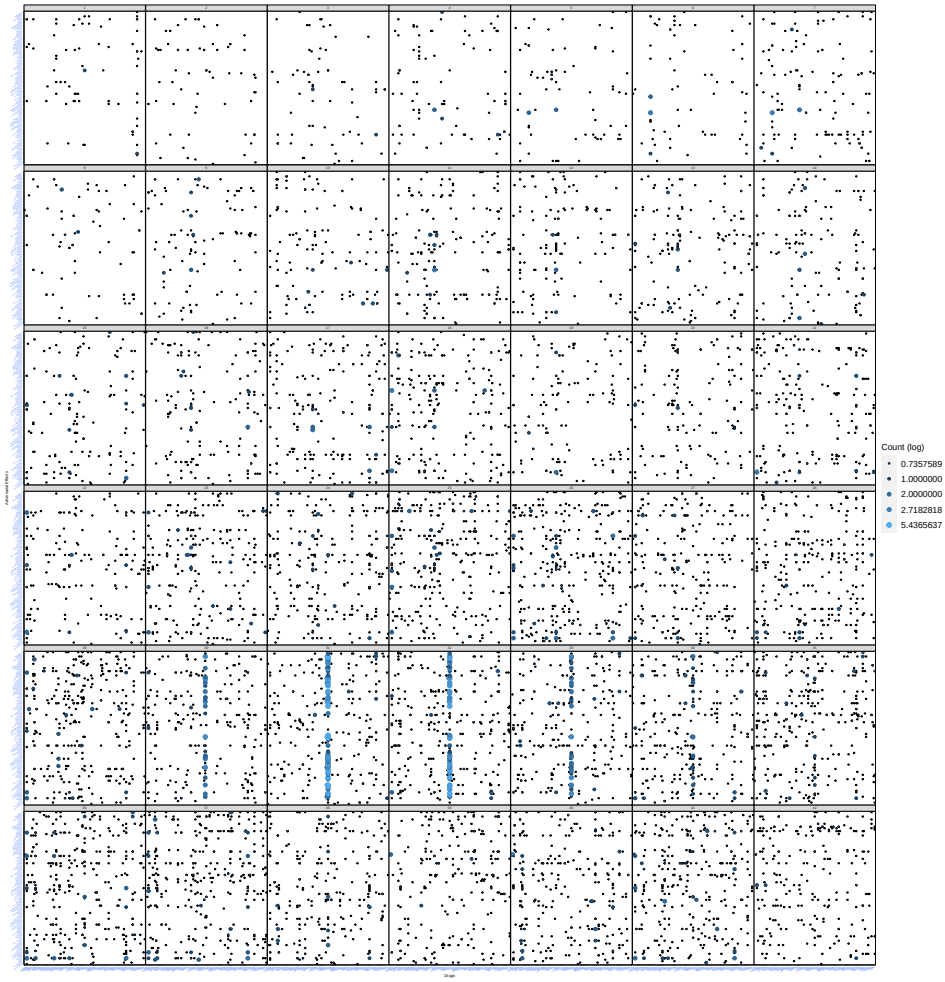

Figure 16: Representation of the interactivity patterns between drugs and adversarial effects at any given time interval. A small sample of the whole data set is considered.

Bibliography

Bergé L. R., Bouveyron C., Corneli M., Latouche P. (2019) The latent topic block model for the co-clustering of textual interaction data. Computational Statistics & Data Analysis 137:247–270

Biernacki C., Celeux G., Govaert G. (2000) Assessing a mixture model for clustering with the integrated completed likelihood. IEEE transactions on pattern analysis and machine intelligence 22(7):719–725

Boutalbi R., Labiod L., Nadif M. (2020) Tensor latent block model for co-clustering. International Journal of Data Science and Analytics pp. 1–15

Bouveyron C., Bozzi L., Jacques J., Jollois F.-X. (2018) 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

Bouveyron C., Celeux G., Murphy T. B., Raftery A. E. (2019) Model-based clustering and classification for data science: with applications in R, vol 50. Cambridge University Press

Cheng K.-O., Law N.-F., Siu W.-C., Liew A. W.-C. (2008) Identification of coherent patterns in gene expression data using an efficient biclustering algorithm and parallel coordinate visualization. BMC bioinformatics 9(1):210

Côme E., Latouche P. (2015) Model selection and clustering in stochastic block models based on the exact integrated complete data likelihood. Statistical Modelling 15(6):564–589

Corneli M., Latouche P., Rossi F. (2016) Block modelling in dynamic networks with non-homogeneous poisson processes and exact icl. Social Network Analysis and Mining 6(1):55

Corneli M., Bouveyron C., Latouche P., Rossi F. (2018) The dynamic stochastic topic block model for dynamic networks with textual edges. Statistics and Computing DOI 10.1007/s11222-018-9832-4, URL `https://hal.archives-ouvertes.fr/hal-01621757`

Corneli M., Bouveyron C., Latouche P. (2020) Co-clustering of ordinal data via latent continuous random variables and not missing at random entries. Journal of Computational and Graphical Statistics pp. 1–15

Dempster A. P., Laird N. M., Rubin D. B. (1977) Maximum likelihood from incomplete data via the em algorithm. Journal of the Royal Statistical Society: Series B (Methodological) 39(1):1–22

Deodhar M., Ghosh J. (2010) Scoal: A framework for simultaneous co-clustering and learning from complex data. ACM Transactions on Knowledge Discovery from Data (TKDD) 4(3):1–31

Dhillon I. S. (2001) Co-clustering documents and words using bipartite spectral graph partitioning. In: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 269–274

Dhillon I. S., Mallela S., Kumar R. (2003a) A divisive information-theoretic feature clustering algorithm for text classification. Journal of machine learning research 3(Mar):1265–1287

Dhillon I. S., Mallela S., Modha D. S. (2003b) Information-theoretic co-clustering. In: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 89–98

Ding C., Li T., Peng W., Park H. (2006) Orthogonal nonnegative matrix t-factorizations for clustering. In: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 126–135

George T., Merugu S. (2005) A scalable collaborative filtering framework based on co-clustering. In: Fifth IEEE International Conference on Data Mining (ICDM'05), pp. 4 pp.–

Govaert G., Nadif M. (2003) Clustering with block mixture models. Pattern Recognition 36(2):463–473

Govaert G., Nadif M. (2008) Block clustering with bernoulli mixture models: Comparison of different approaches. Computational Statistics & Data Analysis 52(6):3233–3245

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

Green N., Rege M., Liu X., Bailey R. (2011) Evolutionary spectral co-clustering. In: The 2011 international joint conference on neural networks, IEEE, pp. 1074–1081

Hanisch D., Zien A., Zimmer R., Lengauer T. (2002) Co-clustering of biological networks and gene expression data. Bioinformatics 18(suppl_1):S145–S154

Jacques J., Biernacki C. (2018) Model-based co-clustering for ordinal data. Computational Statistics & Data Analysis 123:101–115

Keribin C., Govaert G., Celeux G. (2010) Estimation d'un modèle à blocs latents par l'algorithme SEM.

Keribin C., Brault V., Celeux G., Govaert G., et al. (2012) Model selection for the binary latent block model. In: Proceedings of COMPSTAT, vol 2012

Keribin C., Brault V., Celeux G., Govaert G. (2015) Estimation and selection for the latent block model on categorical data. Statistics and Computing 25(6):1201–1216

Keribin C., Celeux G., Robert V. (2017) The Latent Block Model: a useful model for high dimensional data. In: ISI 2017 - 61st world statistics congress, Marrakech, Morocco, pp. 1–6, URL `https://hal.inria.fr/hal-01658589`

Labiod L., Nadif M. (2011) Co-clustering under nonnegative matrix tri-factorization. In: International Conference on Neural Information Processing, Springer, pp. 709–717

Langlade C., Gouverneur A., Bosco-Lévy P., Gouraud A., Pérault-Pochat M.-C., Béné J., Miremont-Salamé G., Pariente A., of Pharmacovigilance Centres F. N. (2019) Adverse events reported for mirena levonorgestrel-releasing intrauterine device in france and impact of media coverage. British Journal of Clinical Pharmacology 85(9):2126–2133

Lomet A. (2012) Sélection de modèle pour la classification croisée de données continues. PhD thesis, Compiègne

Matias C., Rebafka T., Villers F. (2018) A semiparametric extension of the stochastic block model for longitudinal networks. Biometrika 105(3):665–680

Rand W. M. (1971) Objective criteria for the evaluation of clustering methods. Journal of the American Statistical Association 66(336):846–850, URL http://www.jstor.org/stable/2284239

Robert V., Celeux G., Keribin C. (2015) Un modèle statistique pour la pharmacovigilance. In: 47èmes Journées de Statistique de la SFdS, Lille, France, URL https://hal.inria.fr/hal-01255701

Robert V., Vasseur Y., Brault V. (2020) Comparing high-dimensional partitions with the co-clustering adjusted rand index. Journal of Classification pp. 1–29

Viard D., Parassol-Girard N., Romani S., Van Obberghen E., Rocher F., Berriri S., Drici M.-D. (2019) Spontaneous adverse event notifications by patients subsequent to the marketing of a new formulation of levothyrox® amidst a drug media crisis: atypical profile as compared with other drugs. Fundamental & clinical pharmacology 33(4):463–470

Wang P., Domeniconi C., Laskey K. B. (2009) Latent dirichlet bayesian co-clustering. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, pp. 522–537

Wyse J., Friel N. (2012) Block clustering with collapsed latent block models. Statistics and Computing 22(2):415–428

Wyse J., Friel N., Latouche P. (2017) Inferring structure in bipartite networks using the latent blockmodel and exact ICL. Network Science 5(1):45–69

Xu B., Bu J., Chen C., Cai D. (2012) An exploration of improving collaborative recommender systems via user-item subgroups. In: Proceedings of the 21st international conference on World Wide Web, pp. 21–30