A Survey on Model-Based Co-Clustering: High Dimension and Estimation Challenges

Christophe Biernacki, Julien Jacques, C. Keribin

A Survey on Model-Based Co-Clustering: High Dimension and Estimation Challenges

C. Biernacki[a], J. Jacques[b], C. Keribin[c]

[a] Inria, Université de Lille, CNRS, Laboratoire de mathématiques Painlevé, 59650 Villeneuve d'Ascq, France
christophe.biernacki@inria.fr
[b] Université de Lyon, Lyon 2 & ERIC EA3083, 5 Avenue Pierre Mendès France, 69500 Bron, France
julien.jacques@univ-lyon2.fr
[c] Université Paris-Saclay, CNRS, Inria, Laboratoire de mathématiques d'Orsay, 91405 Orsay, France
christine.keribin@universite-paris-saclay.fr

Abstract

Model-based co-clustering can be seen as a particularly valuable extension of model-based clustering for three main reasons: (1) while allowing parsimoniously a drastic reduction of both the number of lines/individuals and columns/variables of a data set, (2) it also allows interpretability of such a resulting reduced data set since initial individuals and features meaning is preserved in this latter; (3) moreover it benefits from the powerful mathematical statistics theory for both estimation and model selection. Hence, many authors produced new advances on this topic in the recent years, and this paper offers a general update of the related literature. In addition, it is the opportunity to pass two messages, supported by specific research materials: (1) co-clustering still requires some new and motivating researches for fixing some well-identified estimation issues, (2) co-clustering is probably one of the most promising clustering approach to be addressed in the (very) high dimension setting, which corresponds to the global trend on modern data sets.

Keywords – High dimension clustering; mixture models; EM-like algorithms; model selection; mixed data types.

Contents

1 Introduction

Statistics is the science of data summarization, aiming at providing a better data understanding of the data, leading to better decisions. Clustering is one of its prominent unsupervised learning principles [Jain et al., 1999] that combines extreme data reduction with meaning extraction. Clustering starts with a set of raw data, say $\mathbf{x}$ of size $n \times d$ (n lines or individuals/objects and d columns or variables/features) and summarizes them into a new data set of size $K \times d$, with $K \ll n$ (K corresponds to the so-called number of clusters). The resulting clusters are often presented as a partitioning of the individual data (a cluster is a collection of individuals), hence offering a limpid understanding of the resulting reduced data set. With the drastic increase of the number of individuals n that comes with many modern applications, it is then not surprising that clustering has become a central paradigm. In practice, many methods implement this general principle [Xu and jie Tian, 2015] but hints about the true complexities of the clustering problem at hand are rarely given. From both practical and theoretical viewpoints, the subtleties of dealing with potential heterogeneous dimensions, missing data, outliers, *etc.*, so well as subtleties of selecting the correct representations (the number K of clusters being one of the most crucial parameter) are challenging issues to address. For these precise reasons, model-based clustering (MBC) has become a reference approach, offering both the power of mathematical statistics machinery and the flexibility of mixture modelling by leveraging the assumption that a cluster is best represented by a specific probability distribution. Moreover, many model-free clustering methods can in fact be reinterpreted as model-based ones with specific, albeit often hidden, assumptions. As an example, the famous K-means algorithm corresponds to a very specific Gaussian mixture model, assuming that the covariance matrices of each cluster are identical and proportional to the identity matrix, and that the mixture proportions are identical so well.

One of the very specific feature of recent trends of data analytics is the growing number of co-variates d, to such a level that it now often exceeds the number of observations ($n < d$ or $n \ll d$). In that case, many clustering methods have emerged for dealing with this so-called high dimensional setting [Bouveyron and Brunet, 2014]. In reaction to this state of affairs, statisticians could be inclined to apply the same data reduction principle already used previously with success for a large number n of individuals, to the case of a large number d of variables. More precisely, the initial raw data set $\mathbf{x}$ of size $n \times d$ could be drastically reduced into a new data set of size $K \times L$, where now $L \ll d$ clusters summarize the d variables, symmetrically to the fact that $K \ll n$ clusters summarize the n individuals. Simultaneous clustering of both the individual and the variables is called *co-clustering*. For the purpose of illustrating this idea, Figure 1 displays in panel (1) a binary data set with $n = 10$ individuals and $d = 7$ binary variables [Govaert and Nadif, 2008]. In panel (2), it displays the clusterized version of this data set with three line clusters, following a suitable rows/individuals permutation. In panel (3), one more transformation stage has been applied to panel (2) by reorganizing also the columns/variables into three column clusters, leading to a so-called co-clusterized data set. Finally, panel (4) is the resulting co-clustering summary of the initial binary 10×7 data set into a smaller 3×3 binary data set (corresponding to 3×3 so-called *blocks*) where each binary value is directly deduced from panel (3).

Co-clustering is a specific bi-clustering model [Madeira and Oliveira, 2004], assuming that all the individuals belong to one and only one row cluster, and symmetrically all the variables belong

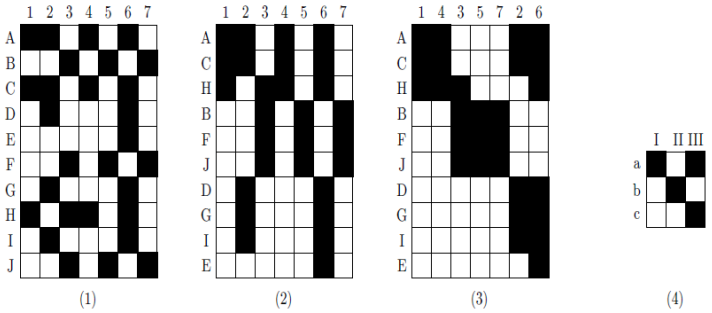

Figure 1: Illustration, in the binary case, of the co-clustering paradigm as an extension of the clustering one [Govaert and Nadif, 2008].

to one and only one column cluster. Bi-clustering algorithms, not making this assumption, aim to detect homogeneous blocks within the data matrix which do not cover the entire matrix and which may overlap. The price to pay for this greater freedom is greater algorithmic complexity. We refer to [Madeira and Oliveira, 2004] for a survey on bi-clustering algorithms. Let finally mention that the terms block clustering, two-mode clustering or two-way clustering are also sometimes used to design what we have defined as co-clustering.

Similarly to model-based clustering, and for similar motivations, there exists a model-based co-clustering approach known as Latent Block Model (LBM). In this survey paper, we focus on LBM with a special emphasis on its interest for addressing MBC in the high dimensional case. Surprisingly, LBM is relatively poorly used in this context, probably because this was historically not introduced for this specific purpose as it can be easily identified in the seminal papers [Good, 1965, Bock, 1979, Govaert, 1983, Dhillon et al., 2003]. This paper is then not an exhaustive review on co-clustering and variants that the reader can easily found in some recent papers [Brault and Lomet, 2015, Brault and Mariadassou, 2015]. It is neither an exhaustive review on LBM that the reader can found also in the book [Govaert and Nadif, 2013], even if we obviously describe more recent advances in LBM. However, this paper addresses some specificities in the LBM estimation process that are not really considered in literature but which drastically change in comparison to MBC behaviour. Finally we claim that LBM should be more used with high dimensional MBC but that further research works are absolutely required for addressing properly and specifically the estimation issues that are attached until now to LBM.

The outline of the paper is the following. Section 2 provides a general overview on the fundamentals of LBM (theoretical properties, methodological approaches, practical uses). Section 3 departs from this traditional LBM by describing some related recent extensions. The next two sections respectively correspond to specific focuses of this survey paper attached to the classical LBM: Section 5 illustrates the interest of using LBM from a high dimensional clustering perspective; Section 4 points out some often misunderstood but perilous issues when invoking most LBM estimation processes. Finally, Section 6 concludes this survey by highlighting a selection of key research directions on LBM that could be usefully led in the future.

Notation The data matrix is noted $\mathbf{x}$, with size $n \times d$. Each row of the matrix corresponds to an individual, and is noted $\mathbf{x}_i$. Each column of the matrix corresponds to a variable, and is noted $\mathbf{x}^j$. Each element of $\mathbf{x}_i$ or $\mathbf{x}^j$ is noted x_i^j and indicates the value of the j-th variable for the i-th individual. Along the paper, index i will refer to the individuals, j to the variables, k to the row clusters and ℓ to the column clusters. Their respective ranges are $\{1, \ldots, n\}$, $\{1, \ldots, d\}$, $\{1, \ldots, K\}$ and $\{1, \ldots, L\}$.

2 The fundamentals of model-based co-clustering

This section provides a general overview on the fundamentals of the Latent Block Model (LBM), an iconic model for model-based co-clustering (MBC). It is particularly dedicated for readers that are not familiar with these notions and allows us to set the notation. Note that LBM can be seen as extension of a specific mixture model-based clustering. Hence, we first recall MBC with mixtures, then define LBM and enlighten its specificity (theoretical properties, methodological approaches, practical uses).

2.1 Model-based clustering (MBC)

As a general reminder first, cluster analysis is one of the main data analysis method. It aims at partitioning a data set $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$, composed by n individuals and lying in a space $\mathcal{X}$ of dimension d, into K groups $G_1, \ldots, G_K$. This partition is denoted by $\mathbf{z} = (\mathbf{z}_1, \ldots, \mathbf{z}_n)$, lying in a space $\mathcal{Z}$, where $\mathbf{z}_i = (z_{i1}, \ldots, z_{iK})'$ is a vector of $\{0,1\}^K$ such that $z_{ik} = 1$ if individual $\mathbf{x}_i$ belongs to the k-th group G_k, and $z_{ik} = 0$ otherwise. Model-based clustering allows to reformulate cluster analysis as a well-posed estimation problem both for the partition $\mathbf{z}$ and for the number K of groups. It considers data $\mathbf{x}_1, \ldots, \mathbf{x}_n$ as n independent and identically distributed (i.i.d.) realizations of a mixture probability density function (p.d.f.) $f(\cdot; \boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k f(\cdot; \boldsymbol{\alpha}_k)$, where $f(\cdot; \boldsymbol{\alpha}_k)$ indicates the p.d.f. associated to the group k, parameterized by $\boldsymbol{\alpha}_k$, and π_k indicates the mixture proportion of this component ($\sum_{k=1}^{K} \pi_k = 1$, $\pi_k \geq 0$). The parameter $\boldsymbol{\theta} = (\pi_k, \boldsymbol{\alpha}_k)_k$ indicates the whole mixture parameters. From the whole data set $\mathbf{x}$ it is then possible to obtain a mixture parameter estimate $\hat{\boldsymbol{\theta}}$ to deduce a partition estimate $\hat{\mathbf{z}}$ from the conditional probability $p(\mathbf{z}|\mathbf{x}; \hat{\boldsymbol{\theta}})$. It is also possible to derive an estimate $\hat{K}$ from a model selection procedure. More details on mixture models, related estimation of $\boldsymbol{\theta}$, $\mathbf{z}$ and K are given for instance in [Biernacki, 2017] or [Bouveyron et al., 2019].

However, for parsimony reasons, it is often assumed that the variables are *conditionally independent* knowing the (latent) groups. In that case, data are supposed to arise independently from a mixture of K multivariate conditional p.d.f. expressed as the following product of d univariate p.d.f.:

$$f(\mathbf{x}_i|\mathbf{z}_i; \boldsymbol{\theta}) = \prod_{k=1}^{K} f(\mathbf{x}_i; \boldsymbol{\alpha}_k)^{z_{ik}} = \prod_{k=1}^{K} \prod_{j=1}^{d} f(x_i^j; \boldsymbol{\alpha}_k^j)^{z_{ik}}, \tag{1}$$

where $\boldsymbol{\alpha}_k = (\boldsymbol{\alpha}_k^1, \ldots, \boldsymbol{\alpha}_k^d)$. When variables are categorical with $f(\cdot; \boldsymbol{\alpha}_k^j)$ being a multinomial distribution, it corresponds to the so-called *latent class model* [Goodman, 1974]. When variables are continuous with $f(\cdot; \boldsymbol{\alpha}_k^j)$ being a univariate Gaussian distribution, the name of *di-*

agonal Gaussian model is given, indicating in this way that the covariance matrices related the multivariate Gaussian components of the mixture are diagonal [Banfield and Raftery, 1993, Celeux and Govaert, 1995].

2.2 Latent block model (LBM)

In addition to the row partition $\mathbf{z}$, we now consider the column partition $\mathbf{w}$ lying in a space $\mathcal{W}$, where $\mathbf{w}_j = (w_{j1}, \ldots, z_{jL})'$ is a vector of $\{0,1\}^L$ such that $w_{i\ell} = 1$ if variable $\mathbf{x}^j$ belongs to the ℓ-th group, and $w_{j\ell} = 0$ otherwise.

The basic idea of the latent block model is to extend the previous latent class principle of local (or conditional) independence. Each data point x_i^j is assumed to be independent once $\mathbf{z}_i$ and $\mathbf{w}_j$ are fixed:

$$f(\mathbf{x}|\mathbf{z}, \mathbf{w}; \boldsymbol{\theta}) = \prod_{k=1}^{K} \prod_{\ell=1}^{L} \prod_{i=1}^{n} \prod_{j=1}^{d} f(x_i^j; \boldsymbol{\alpha}_{k\ell})^{z_{ik} w_{j\ell}}. \tag{2}$$

The whole mixture parameter is now noted $\boldsymbol{\theta} = (\boldsymbol{\pi}, \boldsymbol{\rho}, \boldsymbol{\alpha})$, where $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_{k\ell})_{k,\ell}$ is the block specific model parameter, $\boldsymbol{\pi} = (\pi_k)_k$ (where $\pi_k > 0$ and $\sum_k \pi_k = 1$) and $\boldsymbol{\rho} = (\rho_\ell)_\ell$ (where $\rho_\ell > 0$ and $\sum_\ell \rho_\ell = 1$) are the vectors of probabilities π_k and ρ_ℓ that a row and a column belong to the k-th row component and to the ℓ-th column component, respectively. Assuming also independence between all $\mathbf{z}_i$ and $\mathbf{w}_j$, the latent block mixture model has final p.d.f.:

$$f(\mathbf{x}; \boldsymbol{\theta}) = \sum_{(\mathbf{z},\mathbf{w}) \in \mathcal{Z} \times \mathcal{W}} \prod_{i,k} \pi_k^{z_{ik}} \prod_{j,\ell} \rho_j^{w_{j\ell}} \prod_{i,j,k,\ell} f(x_i^j; \boldsymbol{\alpha}_{k\ell})^{z_{ik} w_{j\ell}}, \tag{3}$$

where $\mathcal{Z}$ (resp. $\mathcal{W}$) represents the set of all possible partitions of the rows (resp. the columns) of $\mathbf{x}$.

At this step, it is important to notice that the p.d.f. $f(\cdot; \boldsymbol{\alpha}_{k\ell})$ depends on the type of data for x_i^j:

- In the *binary* case: $x_i^j \in \{0,1\}$ and $f(\cdot; \boldsymbol{\alpha}_{k\ell})$ is the p.d.f of the Bernoulli distribution $\mathcal{B}(\boldsymbol{\alpha}_{k\ell})$ of parameter $\boldsymbol{\alpha}_{k\ell} = p(x_i^j = 1 | z_{ik} w_{j\ell} = 1)$, see [Govaert and Nadif, 2008];
- In the *categorical* case with r levels: $x_i^j = (x_i^{jh})_h \in \{0,1\}^r$, with $\sum_{h=1}^r x_i^{jh} = 1$ and $f(\cdot; \boldsymbol{\alpha}_{k\ell})$ is the p.d.f. of the multinomial distribution $\mathcal{M}(1, \boldsymbol{\alpha}_{k\ell})$ of parameter $\boldsymbol{\alpha}_{k\ell} = (\alpha_{kl}^1, \ldots, \alpha_{k\ell}^r)$ with $\alpha_{k\ell}^h = p(x_i^{jh} = 1 | z_{ik} w_{j\ell} = 1)$ for $h = 1, \ldots, r$, see [Keribin et al., 2015];
- In the *count data* case: $x_i^j \in \mathbb{N}$ and $f(\cdot; \boldsymbol{\alpha}_{k\ell})$ is the p.d.f. of the Poisson distribution $\mathcal{P}(\mu_i \nu_j \gamma_{k\ell})$, see [Govaert and Nadif, 2013]. The Poisson parameter is here split into μ_i and ν_j, the size effects of the row i and the column j respectively, and $\gamma_{k\ell}$ the effect of the block (k, ℓ). Unfortunately, this parameterization is not identifiable. It is therefore not possible to estimate simultaneously μ_i, ν_j and $\gamma_{k\ell}$ without imposing further constraints. The set of constraints $\sum_i \mu_i = \sum_j \nu_j = (\sum_k \pi_k \gamma_{kl})^{-1} = (\sum_\ell \rho_\ell \gamma_{kl})^{-1} = \sum_{ij} \mathbb{E}(x_i^j)$ for all k, ℓ are usually chosen, which ensures that $\mathbb{E}(\sum_j x_i^j) = \mu_i$ and $\mathbb{E}(\sum_i x_i^j) = \nu_j$. Hence, μ_i and ν_j are naturally estimated by the margins in rows and columns, and are then considered as fixed.

Table 1: Number of parameters of LBM and MBC. We have $\dim(\boldsymbol{\pi}) = K - 1$ in the case of free row proportions and $\dim(\boldsymbol{\pi}) = 0$ in the case of equal proportions. Symmetrically, we have $\dim(\boldsymbol{\rho}) = L - 1$ in the case of free column proportions and $\dim(\boldsymbol{\rho}) = 0$ in the case of equal proportions. (*) Takes into account correlation between the continuous components.

Model	Number of LBM parameters	Number of MBC parameters
Binary	$\dim(\boldsymbol{\pi}) + \dim(\boldsymbol{\rho}) + KL$	$\dim(\boldsymbol{\pi}) + Kd$
Categorical	$\dim(\boldsymbol{\pi}) + \dim(\boldsymbol{\rho}) + KL(m-1)$	$\dim(\boldsymbol{\pi}) + Kd(m-1)$
Contingency	$\dim(\boldsymbol{\pi}) + \dim(\boldsymbol{\rho}) + KL$	$\dim(\boldsymbol{\pi}) + Kd$
Continuous	$\dim(\boldsymbol{\pi}) + \dim(\boldsymbol{\rho}) + 2KL$	$\dim(\boldsymbol{\pi}) + 2Kd$
		(*) $\dim(\boldsymbol{\pi}) + Kd + Kd(d-1)/2$

- In the *continuous* case: $x_i^j \in \mathbb{R}$ and $f(\cdot; \boldsymbol{\alpha}_{k\ell})$ is generally taken to be the p.d.f. of the Gaussian distribution $\mathcal{N}(\mu_{k\ell}, \sigma_{k\ell}^2)$ of parameter $\boldsymbol{\alpha}_{k\ell} = (\mu_{k\ell}, \sigma_{k\ell}^2)$, denoting respectively the mean and the variance, see [Govaert and Nadif, 2013].

These data types for x_i^j are basic, and LBM has been extended to numerous other types. We refer the reader to section 3.1 for examples with more advanced data types such as ordinal, functional or textual data.

Such models can be very parsimonious[1] even in the High-Dimensional (HD) setting (when d is large, and even larger than n), provided that L is quite low, as it is shown in Table 1: the number of parameters with MBC involves a dependence in the number d of variables whereas in LBM this dependence is only in the number of column clusters $L \ll d$ and no longer on d.

Consequently, LBM could provide good candidates for performing HD clustering even if they are not exactly designed for this aim initially. In such a case, clustering of columns can just be seen as an instrumental strategy for obtaining HD parsimonious models. Indeed, the HD clustering purpose only concerns clustering of the n rows, and not that of the d columns. However, column clustering offered by co-clustering can provide a readability of the model to the practitioner. We shall develop this point of view in Section 5.

2.3 Model identifiability

Obviously, LBM parameters can only be identified up to a relabelling of the blocks, as in any mixture model. [Keribin et al., 2015] established the identifiability of the *binary* LBM for $n \geq 2L - 1$ and $d \geq 2K - 1$ and when the following two conditions are fullfilled:

- C_1: for all $1 \leq k \leq K$, $\pi_k > 0$ and all coordinates of vector $\boldsymbol{\alpha}\boldsymbol{\rho}$ are distinct,
- C_2: for all $1 \leq \ell \leq L$, $\rho_\ell > 0$ and all coordinates of vector $\boldsymbol{\pi}'\boldsymbol{\alpha}$ are distinct.

These conditions are not strongly restrictive since the set of vectors $\boldsymbol{\alpha}\boldsymbol{\rho}$ and $\boldsymbol{\pi}'\boldsymbol{\alpha}$ that do not fulfilled them is of Lebesgue measure 0. Therefore, this result asserts the *generic* identifiability of the binary LBM, which is a *practical* identifiability, explaining why it works in the applications [Carreira-Perpinán and Renals, 2000].

[1] Some more parsimonious versions are also defined (see [Govaert and Nadif, 2008]).

It follows from conditions C_1 (resp. C_2) that the probabilities $P(x_i^j = 1|z_{ik} = 1)$ (resp. $P(x_i^j = 1|w_{j\ell} = 1)$) to observe an event in a cell of a row of class k (resp. in a cell of a column of class ℓ) can be sorted in a strictly ascending order. Hence, contrary to what happens in the Gaussian mixture context, these conditions can be used to set a natural order on the row and column clusters.

Generic identifiability is easily extended to the categorical case, where the conditions are defined on the vectors $\boldsymbol{\alpha}^h\boldsymbol{\rho}$ and $\boldsymbol{\pi}'\boldsymbol{\alpha}^h$, with $\boldsymbol{\alpha}^h = (\alpha_{k\ell}^h)_{k\ell}$, and then more generally to family of distributions which are identifiable from probabilities defined on a finite number of sub-domains of their support, which is the case for example for Poisson and Gaussian data.

2.4 LBM estimation

Using Equation (3), the *observed* log-likelihood is defined as:

$$\ell(\boldsymbol{\theta}; \mathbf{x}) = \log f(\mathbf{x}; \boldsymbol{\theta}) = \log \left(\sum_{(\mathbf{z},\mathbf{w}) \in \mathcal{Z} \times \mathcal{W}} \prod_{i,k} \pi_k^{z_{ik}} \prod_{j,\ell} \rho_j^{w_{j\ell}} \prod_{i,j,k,\ell} f(x_i^j; \boldsymbol{\alpha}_{k\ell})^{z_{ik} w_{j\ell}} \right).$$

Contrarily to the case of simple mixture models, $f(\mathbf{x}; \boldsymbol{\theta})$ does not factorize due to the complex dependency induced by the block structure, and the calculation of the observed likelihood or its logarithm requires the sum of $K^n L^d$ terms, defined by all possible configurations of the unobserved labels $\mathbf{z}$ and $\mathbf{w}$; this is not numerically tractable in a reasonable time, even for a few observations and a few blocks. For example, it requires to compute around 10^{12} terms for a LBM with 2×2 blocks and 20×20 observations.

However, in presence of such latent data, a standard approach for maximum likelihood estimation is usually to perform Expectation Maximization (EM)-based algorithms, that do not use directly the log-likelihood. Their basic principle is to work with the log-likelihood of the *complete* data (represented as a vector $(\mathbf{x}, \mathbf{z}, \mathbf{w})$) and to introduce $Q(\boldsymbol{\theta}, \boldsymbol{\theta}') = \mathbb{E}(\ell_c(\boldsymbol{\theta}; \mathbf{x}, \mathbf{z}, \mathbf{w}); \mathbf{x}, \boldsymbol{\theta}')$, a surrogate function of $\boldsymbol{\theta}$, namely the expectation of the complete log-likelihood conditionally to the latent data $(\mathbf{z}, \mathbf{w})$ under a current parameter $\boldsymbol{\theta}'$. The EM algorithm lays on the fact that the parameter $\tilde{\boldsymbol{\theta}}$ maximizing Q in $\boldsymbol{\theta}$ increases the likelihood: $\ell(\tilde{\boldsymbol{\theta}}) \geq \ell(\boldsymbol{\theta}')$. Hence the maximum likelihood estimator results from the convergence of successive iterations of the two steps, written here at epoch q:

Expectation: computation of $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(q)})$ and *Maximization*: $\boldsymbol{\theta}^{(q+1)} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(q)})$.

Note If EM algorithms are known in the statistical community from the seminal work of [Dempster et al., 1977], they are seen as a special case of $Minimize - Majorize$ methods in the optimization community [De Leeuw and Michailidis, 1999]. Minimize-Majorize methods replace a function whose search for the maximum is delicate by a family of tangent functions minorizing it and easy to optimize. In fact, the log-likelihood can be decomposed as $\ell(\boldsymbol{\theta}; \mathbf{x}) = \ell(\boldsymbol{\theta}'; \mathbf{x}) + Q(\boldsymbol{\theta}, \boldsymbol{\theta}') - Q(\boldsymbol{\theta}', \boldsymbol{\theta}') + K(\boldsymbol{\theta}', \boldsymbol{\theta})$ where $K(\boldsymbol{\theta}', \boldsymbol{\theta}) \geq 0$ is the Kullback divergence between the conditional distribution of the labels under parameters $\boldsymbol{\theta}'$ and $\boldsymbol{\theta}$. Hence, function $\psi(\boldsymbol{\theta}, \boldsymbol{\theta}') = \ell(\boldsymbol{\theta}'; \mathbf{x}) + Q(\boldsymbol{\theta}, \boldsymbol{\theta}') - Q(\boldsymbol{\theta}', \boldsymbol{\theta}')$ minorizes the log-likelihood: for all $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$, $\psi(\boldsymbol{\theta}, \boldsymbol{\theta}') \leq \ell(\boldsymbol{\theta}'; \mathbf{x})$ and $\psi(\boldsymbol{\theta}', \boldsymbol{\theta}') = \ell(\boldsymbol{\theta}'; \mathbf{x})$. Each surface $\boldsymbol{\theta} \mapsto \psi(\boldsymbol{\theta}, \boldsymbol{\theta}')$ lies under the surface $\ell(\boldsymbol{\theta}; \mathbf{x})$ and is tangent

to it at the point $\boldsymbol{\theta} = \boldsymbol{\theta}'$. This is to say, $\ell(\boldsymbol{\theta}; \mathbf{x}) = \max_{\boldsymbol{\theta}'} \psi(\boldsymbol{\theta}, \boldsymbol{\theta}')$ where we recognize an ordinary block relaxation situation or alternate resolution of $\psi(\boldsymbol{\theta}, \boldsymbol{\theta}')$.

2.4.1 Parameter estimation for LBM

For LBM, the complete likelihood is written as:

$$\ell_c(\boldsymbol{\theta}; \mathbf{x}, \mathbf{z}, \mathbf{w}) = \sum_k (\sum_i z_{ik}) \log \pi_k + \sum_l (\sum_j w_{jl}) \log \rho_l + \sum_{i,j,k,l} z_{ik} w_{jl} \log f(x_i^j; \boldsymbol{\alpha}_{kl}),$$

and function $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(q)})$ involved at the q-th iteration of the standard E-Step is expressed by

$$\begin{aligned} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(q)}) &= \sum_{i,k} p(z_{ik} = 1 | \mathbf{x}; \boldsymbol{\theta}^{(q)}) \log \pi_k + \sum_{j,l} p(w_{jl} = 1 | \mathbf{x}; \boldsymbol{\theta}^{(q)}) \log \rho_l \\ &\quad + \sum_{i,j,k,l} p(z_{ik} w_{jl} = 1 | \mathbf{x}; \boldsymbol{\theta}^{(q)}) \log f(x_i^j; \boldsymbol{\alpha}_{kl}). \end{aligned} \tag{4}$$

Unfortunately again, difficulties arise in this E-step owing to the dependence structure in the model, and more precisely in the combinatorial difficulty for evaluating the terms $s_{ik}^{(q)} = p(z_{ik} = 1 | \mathbf{x}; \boldsymbol{\theta}^{(q)})$, $t_{j\ell}^{(q)} = p(w_{j\ell} = 1 | \mathbf{x}; \boldsymbol{\theta}^{(q)})$ and $p(z_{ik} w_{j\ell} = 1 | \mathbf{x}; \boldsymbol{\theta}^{(q)})$. Several solutions exist for skirting this difficulty (see [Govaert and Nadif, 2013] for more details), including:

- The so-called *variational approach* which constraints the problematic joint probability to satisfy the relation

  $$p(\mathbf{z}, \mathbf{w} | \mathbf{x}; \boldsymbol{\theta}) \approx p_z(\mathbf{z} | \mathbf{x}; \boldsymbol{\theta}) p_w(\mathbf{w} | \mathbf{x}; \boldsymbol{\theta}).$$

  Densities p_z and p_w are chosen to provide the closest approximation of $p(\mathbf{z}, \mathbf{w} | \mathbf{x}; \boldsymbol{\theta})$ while still being computable. Hence the algorithm maximizes a lower bound of the likelihood called *free energy* or *ELBO* (evidence lower bound):

  $$\ell(\boldsymbol{\theta}; \mathbf{x}) \geq \mathcal{F}(\boldsymbol{\theta}; \mathbf{x}) = \max_{p_z, p_w} \mathbb{E}_{p_z, p_w} \left(\ell_c(\boldsymbol{\theta}; \mathbf{x}, \mathbf{z}, \mathbf{w}) - \log(p_z(\mathbf{z}) p_z(\mathbf{w})) \right)$$

  and alternates computation of the free energy (E-Step) and maximization of $\mathcal{F}$ in $\boldsymbol{\theta}$ (M-step). This algorithm, either called BEM [Govaert and Nadif, 2008] or VEM [Keribin et al., 2012], is thus based on a numerical approximation and leads to the so-called variational estimator of the parameter.

- The so-called *SEM algorithm* [Celeux and Diebolt, 1986, Celeux et al., 1996] which replaces the E-step by a SE-step. In the S-step, random couples $(\mathbf{z}, \mathbf{w})$ are drawn according to the posterior distributions of the labels (conditionally to $\mathbf{x}$). As already seen, these distributions are not tractable for LBM. However, their outcomes can be easily sampled with a two-step Gibbs algorithm:

  $$\text{simulate} \quad \mathbf{z} | \mathbf{x}, \mathbf{w}; \boldsymbol{\theta} \quad \text{and then} \quad \mathbf{w} | \mathbf{x}, \mathbf{z}; \boldsymbol{\theta}.$$

  SEM using this Gibbs sampling SE-step is called SEM-Gibbs [Keribin et al., 2012]. In this case, there is no numerical approximation, the distributions used in the different steps of the

Gibbs sampler being exact. On the other hand, SEM-Gibbs does not increase the likelihood at each iteration, but generates an irreducible Markov chain with a unique stationary distribution which must be concentrated around the maximum likelihood parameter estimate [McLachlan and Krishnam, 1997]. Thus, [Keribin et al., 2012] advocated that SEM-Gibbs algorithm makes it possible to reasonably overcome the initialization problems encountered with any EM algorithm and recommended to initialize VEM by SEM-Gibbs. Note that the SEM-Gibbs algorithm could be subject to the label switching problem [Stephens, 2000] that, in addition could be exacerbated in the co-clustering context since its occurrence can be visible both in the row and in the column partitions.

The Bayesian approach has also been used for its beneficial regularization effect, in particular to prevent algorithms from degeneracy and void classes, see detailed discussion on this topic in Section 4. For example, defining a Dirichlet *a priori* distribution $\mathcal{D}(a, \ldots, a)$ on the mixing weights $\boldsymbol{\pi}$, leads to the following updating expression of $\boldsymbol{\pi}$ during the so called EM-VBayes algorithm (see details in [Keribin et al., 2015]):

$$\pi_k^{(q+1)} = \frac{a - 1 + \sum_i s_{ik}^{(q+1)}}{n + K(a-1)}$$

where the $s_{ik}^{(q+1)}$ are the current expressions of the conditional distributions of the labels $p(z_{ik} = 1|\mathbf{x}; \theta^{(q)})$. Choosing $a \neq 1$ prevents π_k to vanish while $a = 1$ corresponds to VEM. If EM-VBayes makes it possible to avoid class degeneracy, it is however sensitive to initialization like any variational Bayesian algorithm. A general Gibbs sampler defined with the posteriors of the parameters and latent variables can replace all the EM scheme. As it better explores the space, it can provide a sensible strategy for initialization, as experiments have shown [Brault et al., 2014].

These algorithms are governed by several parameters like the number of inside iterations of the E-Step, the tolerance which states the end of convergence of the criteria (VEM, EM-VBayes) and the maximum number of epochs. The estimator $\hat{\boldsymbol{\theta}}$ is then defined by the value $\boldsymbol{\theta}^{last}$ obtained at the last epoch (VEM, EM-VBayes), or by averaging a sequence of $\boldsymbol{\theta}^{(q)}$ values obtained after a burning period (SEM-Gibbs, Gibbs).

2.4.2 Estimating and evaluating the row and column clusters

The preceding algorithms also provide a mean to give an estimation $(\hat{\mathbf{z}}, \hat{\mathbf{w}})$ of the double partition. With VEM and EM-VBayes, it results from a Maximum A Posteriori (MAP) rule on the last value of the conditional distribution: for all i, $\widehat{z}_{ik} = 1$ for k maximizing $s_{ik}^{last} = p(z_{ik} = 1|\mathbf{x}; \boldsymbol{\theta}^{last})$ and symmetrically, for all j, $\widehat{w}_{j\ell} = 1$ for ℓ maximizing $t_{j\ell}^{last} = p(w_{j\ell} = 1|\mathbf{x}; \boldsymbol{\theta}^{last})$. For SEM-Gibbs and Gibbs, once $\hat{\boldsymbol{\theta}}$ is obtained, a new Gibbs algorithm should be used to simulate couples $(\mathbf{z}, \mathbf{w})|\mathbf{x}; \hat{\boldsymbol{\theta}}$. The final partitions $(\hat{\mathbf{z}}, \hat{\mathbf{w}})$ are then estimated using the mode of their marginal sampled distribution.

When only interested by the partitions, a Classification EM (CEM) algorithm can be used. In this case, the partition itself is seen as a parameter. The E step not only computes the conditional distribution of the labels, but also infers the partition $(\mathbf{z}^{(q)}, \mathbf{w}^{(q)})$ with a MAP rule. The parameters are then updated using $(\mathbf{z}^{(q)}, \mathbf{w}^{(q)})$ instead of their relaxed counterparts $(\mathbf{s}^{(q)}, \mathbf{t}^{(q)})$, with $\mathbf{s}^{(q)} = (s_{ik}^{(q)})_{ik}$ and $\mathbf{t}^{(q)} = (t_{j\ell}^{(q)})_{j\ell}$.

Concerning now the evaluation of the co-clustering partitions, more precisely measuring the agreement between two of them, special attention is required since it is not trivial. In particular, a new criterion has been developed, based on the Adjusted Rand Index [Rand, 1971], which is called the Co-clustering Adjusted Rand Index, called CARI [Robert et al., 2021]. This work also proposes extensions of other existing criteria such as the classification error rate or the normalized mutual information criterion.

2.5 Recent theoretical results on estimation

Theory's advances for LBM are closely linked to those of the *Stochastic Block Model* (SBM). SBM is a probabilistic model used to cluster the nodes of a (directed or undirected) graph. A graph can be represented by its adjacency matrix $\mathbf{x}$, where $x_i^j = 1$ whether an edge exists between the nodes i and j, 0 otherwise. Hence, clustering a graph, *i.e.* partitioning its nodes into classes sharing the same connection behaviours, is of great interest to describe the graph heterogeneity, as it sums up the network through groups with different behaviours. For example in community detection, one wants to find groups of people that are highly connected between them, and less connected to people from other groups, such as for internet web communities [Flake et al., 2002]. There are many applications, in various fields (such as ecology [Girvan and Newman, 2002] and transport [Etienne and Latifa, 2014] for example), see [Matias and Robin, 2014, Abbe, 2017] for recent reviews of this very active field. Graph clustering can be viewed as a co-clustering of its adjacency matrix, but with rows and columns representing the same entities. Hence, there is only one latent variable $\mathbf{z}$. Using probabilistic method and the paradigm we already presented for LBM, the expression of the likelihood for SBM in case of a directed graph is

$$f(\mathbf{x}, \theta) = \sum_{\mathbf{z} \in \mathcal{Z}} \prod_{i,k} \pi_k^{z_{ik}} \prod_{i,j,k,\ell} f(x_i^j; \alpha_{k\ell})^{z_{ik} z_{j\ell}} \tag{5}$$

where $f(.; \alpha_{k\ell})$ stands for the probability of a connection between a node of cluster k and a node of cluster ℓ. SBM encounters the same induced intricate dependence on the observations due to the structure of the blocks, despite the fact that it only uses one set of latent labels. Theoretical properties have then first be studied on SBM, then extended to LBM and its double missing structure.

While various estimation strategies were available, see Section 2.4, the consistency of maximum likelihood and variational estimators have been proved only recently. In fact, the theoretical study of the asymptotic properties of these estimators is a delicate problem where difficulties still arise from the complex dependence of the observations induced by the block structure. A first lead has been followed ten years ago with the study of the distribution of the labels conditionally to the observations, namely $p(\mathbf{z}, \mathbf{w}|\mathbf{x}; \boldsymbol{\theta})$. [Celisse et al., 2012] showed for the SBM that, under the true value of the parameter, $p(\mathbf{z}, \mathbf{w}|\mathbf{x}; \boldsymbol{\theta})$ tends to a Dirac of support on the true labels $(\mathbf{z}^\star, \mathbf{w}^\star)$. This convergence is also valid under the estimated value of the parameter if the estimator of the parameter of the conditional distribution converges at a rate of at least n^{-1} towards the true value, where n is the number of nodes of the graph (see their Proposition 3.8). This assumption is not trivial, and it was not established that such an estimator exists except in certain specific cases, see [Ambroise and Matias, 2012]. [Mariadassou and Matias, 2015] presented a unifying framework for SBM and LBM defined on observations coming from exponential

families, and showed the convergence of the conditional distribution for any parameter values in a neighborhood of the true value for observations satisfying a concentration property:

$$\hat{\boldsymbol{\theta}} \overset{n,d\to\infty}{\longrightarrow} \boldsymbol{\theta}^* \quad \Rightarrow \quad p(\hat{\mathbf{z}} = \mathbf{z}^*, \hat{\mathbf{w}} = \mathbf{w}^* | \mathbf{x}; \hat{\boldsymbol{\theta}}) \overset{n,d\to\infty}{\longrightarrow} 1, \tag{6}$$

where $\boldsymbol{\theta}^*$ and $\mathbf{w}^*$ respectively design the true $\boldsymbol{\theta}$ and $\mathbf{w}$. They however could not get rid of the assumed existence of a consistent estimator $\hat{\boldsymbol{\theta}}$.

Using another approach, [Bickel et al., 2013] proved consistency and asymptotic normality of the maximum likelihood and variational estimators of the SBM model. Breaking with the previous authors' view, they studied first the asymptotic behavior of the maximum likelihood estimator in the complete model (observations and labels) which is easier to handle. Then, using a Bernstein inequality for bounded observations, they proved that the complete and the observed likelihoods have a similar asymptotic behavior. This point is the delicate part of the proof, and the key for consistency and asymptotic normality that are finally deduced. Following the scheme of [Bickel et al., 2013], [Brault et al., 2020] extended these properties to LBM for observations coming from exponential families, dealing with the double asymptotic in rows and columns ($n, d \to \infty$, such that $(\log d)/n \to 0$ and $(\log n)/d \to 0$). Moreover, they pointed out the specific case of models presenting parameter symmetry (same complete likelihood under given parameter values, considering a set of labels, or some of their permutation) which was omitted by [Bickel et al., 2013].

2.6 LBM selection

One crucial point in clustering is the choice of the number of clusters. Similary, in co-clustering, the choice of the number K of row clusters and the number L of column clusters is an important question. Thanks to the model-based approach, this choice can be view as a model selection problem.

2.6.1 Model selection criteria

It is crucial to notice that model selection in co-clustering has to be performed with caution since some traditional criteria cannot be used straightforwardly. In particular, it is hazardous to use asymptotic criteria like BIC since asymptotic is now double with both quantities n and d. In addition, using non asymptotic evaluation of the likelihood has to be given up because of the combinatorial difficulty involved by the latent variables $\mathbf{z}$ and $\mathbf{w}$.

Avoiding both asymptotic problems and combinatorial difficulties is possible by using exact expression of the ICL criterion ([Biernacki et al., 2000], [Biernacki et al., 2011]). In the co-clustering context, ICL is written:

$$\text{ICL} = \ln f(\mathbf{x}, \hat{\mathbf{z}}, \hat{\mathbf{w}}) = \ln f(\mathbf{x}|\hat{\mathbf{z}}, \hat{\mathbf{w}}) + \ln p(\hat{\mathbf{z}}) + \ln p(\hat{\mathbf{w}}),$$

$\hat{\mathbf{z}}$ and $\hat{\mathbf{w}}$ being the MAP estimate of $\mathbf{z}$ and $\mathbf{w}$ respectively obtained from $\hat{\boldsymbol{\theta}}$ (see Section 2.4.2). [Lomet et al., 2012b] provide the corresponding closed-form expression of ICL for the Gaussian situation and [Keribin et al., 2015] similarly for the Bernoulli and multinomial cases. We refer

the reader to these references for detailed discussion about the Bayesian hyperparameter choice (ICL is natively defined within the Bayesian paradigm).

In addition, in the multinomial setting with r levels, [Keribin et al., 2015] use their non-asymptotic expression to derive the new following asymptotic one, called ICLbic:

$$\text{ICLbic} = \ell(\hat{\boldsymbol{\theta}}; \mathbf{x}, \hat{\mathbf{z}}, \hat{\mathbf{w}}) - \frac{K-1}{2}\ln(n) - \frac{L-1}{2}\ln(d) - \frac{KL(r-1)}{2}\ln(nd).$$

It is interesting to notice that, in comparison to the ICLbic formula in the simple mixture context, now both the row number n and the column number d are involved in the penalty. Using then the straightforward link $\text{ICL} = \ln f(\hat{\mathbf{z}}, \hat{\mathbf{w}}|\mathbf{x}; \hat{\boldsymbol{\theta}}) + \text{BIC}$ between ICLbic and ICL, they propose the following co-clustering specific asymptotic version of BIC:

$$\text{BIC} = \ell(\hat{\boldsymbol{\theta}}; \mathbf{x}) - \frac{K-1}{2}\ln(n) - \frac{L-1}{2}\ln(d) - \frac{KL(r-1)}{2}\ln(nd).$$

Again, it is interesting to observe the way that both n and d are present in the penalty. Nevertheless, the BIC calculus remains unattainable since it relies on the unavailable value of the log-likelihood $\ell(\hat{\boldsymbol{\theta}}; \mathbf{x})$. However, an approximate version

$$\text{BICvar} = \mathcal{F}(\hat{\boldsymbol{\theta}}; \mathbf{x}) - \frac{K-1}{2}\ln(n) - \frac{L-1}{2}\ln(d) - \frac{KL(r-1)}{2}\ln(nd)$$

can be defined by replacing the maximum likelihood by the maximum value of the surrogate free energy function used in variational inference. Simulations in [Keribin et al., 2015] show similar model choice efficiency for BICvar and ICL.

Finally, [Keribin et al., 2015] make the conjecture, corroborated with experiments, that BIC and ICL are asymptotically equivalent and thus have the same asymptotic behaviour. As a consequence, if BIC is consistent for LBM as for the simple mixture case, ICL criterion could also be expected to be consistent for selecting both K and L in co-clustering, for any true parameter setting. This would be drastically different from simple row clustering where ICL consistency is only true for sufficiently separated clusters [Baudry, 2015]. [Wang and Bickel, 2017] recently proved for SBM that if the penalty in BICvar is of order $n\log(n)$ instead of $\log(n)$, this criterion is consistent. Although this condition is only sufficient and obtained with large upperbounds, it questions the consistency of BIC.

2.6.2 Exploration of the space of possible values for (K, L)

The exploration of the set of all possible values for (K, L) is more tedious than in the simple clustering as two directions as to be considered. Indeed, if $1 \leq K \leq K^{\max}$ and $1 \leq L \leq L^{\max}$, the number $K^{\max}L^{\max}$ of possible models can be large. [Robert, 2017] consider a greedy search which consists in exploring only a relevant subspace of possible combinations of (K, L). At each step, the algorithm consists in computing the model selection criterion of the models obtained with one additional cluster, either in row or in column. The solution with the best criterion is retained and the previous step is repeated until the model selection criterion does no longer increase. Simulation studies show a good behaviour of this heuristic strategy.

Considering a Bayesian version of LBM, [Wyse and Friel, 2012] estimate simultaneously the partitions and the number of clusters through a Markov Chain Monte Carlo (MCMC) algorithm. [Wyse et al., 2017] replace the use ot the MCMC algorithm with a greedy search in the space of $(\mathbf{z}, \mathbf{w})$, optimizing directly the ICL criterion. This approach has the advantage to be more scalable than the MCMC approach in larger settings.

2.7 Some existing LBM packages

There exist several packages for performing co-clustering through the LBM model, mostly in R. This section summarizes which type of data these packages consider, which algorithms they use, which model choice criteria are available as well as which initialization strategies.

blockcluster The R package `blockcluster` is maybe the most complete [Singh Bhatia et al., 2017]. It allows to work with binary, categorical, count and continuous data, proposing different parsimonious model assuming that the proportions and the variances are equal or not between clusters. The three main algorithms (VEM, CEM and SEM-Gibbs) are implemented. Model choice is performed through the ICLbic criterion, and several initializations are possible (CEM, "small EM" or "random"). In addition, the package allows to perform semi-supervised co-clustering, given as an input the row or column partition.

blockmodels The R package `blockmodels` [Leger et al., 2020] is dedicated to the estimation of both LBM and SBM, for binary, count and continuous data. The VEM algorithm is considered, and model selection is performed through the ICLbic criterion. An heuristic exploration procedure for selecting the number of row and column clusters is proposed. Initialization is performed through the absolute eigenvalues spectral clustering method [Rohe et al., 2011]. The package allows to take into account covariates, by making the distribution within a block depends on these covariates.

mixedClust The R package `mixedClust` [Selosse et al., 2021] is dedicated to mixed-type data, *i.e.* when data of several type co-exists (binary, categorical, ordinal, count, continuous, functional; see description of co-clustering for functional data in Section 3). It implements the Multiple LBM model [Robert, 2017], in order to prevent variables of different types to be merged into a same cluster. More detail will be given in Section 3.1.3. Inference is performed through the SEM-Gibbs algorithm, and model selection with ICLbic. Initialization can be random, using K-means (on rows and columns), or random with a resampling technique which prevents empty clusters during a given burn-in phase. Let note that when only one type of data occurs, the considered model is then LBM, and the package allows thus to perform usual co-clustering for a large variety of type of data.

bikm1 The R package `bikm1` [Robert, 2021] implements the LBM and Multiple LBM for binary and count data (see description of co-clustering for Multiple LBM in Section 3). Inference is performed with the VEM algorithm, and model selection through ICL or BICvar. Initialization can be done randomly or using "small EM-VBayes". The package provides also the CARI index (see [Robert et al., 2021] and Section 2.4.2) to compare multi-partitions.

ordinalClust The R package `ordinalClust` [Selosse et al., 2020b] is dedicated to ordinal data and considers the LBM model with SEM-Gibbs algorithm. Model selection is performed through ICLbic criterion and initialization can be random, using K-means, or random with resampling (as for the `mixedClust` package). Let notice that this package also allows to perform row classification or row clustering, using the column clustering as a strategy to define parsimonious model.

funLBM The R package `funLBM` [Bouveyron et al., 2021] performs co-clustering of functional data matrix, *i.e.* when matrix entries are one or several curves. It considers the LBM model and the SEM-Gibbs algorithm. Model selection is performed through ICLbic criterion. Initialization can be "random", using K-means or using `funFEM` [Bouveyron et al., 2015], a clustering method for functional data.

greed The R package `greed` [Côme and Jouvin, 2021] is dedicated to count data. It has the originality to not use an EM-like algorithm to maximize the LBM likelihood, but considers a combination of greedy local search and a genetic algorithm to directly optimize the ICLbic criterion. Initialization of the algorithm is perfomed either using spectral clustering or K-means.

Sparsebm The Python package `Sparsebm` [Frisch et al., 2021b] is dedicated to binary data, and estimates the LBM model through a version of the VEM algorithm specific for sparse matrices. Model selection is performed using ICLbic, and only "random" initialization is proposed.

2.8 Some typical LBM use cases

This section presents some applications in which co-clustering has been used. This list of works is not intended to be exhaustive, but gives an overview of the different fields of application. Interested reader can refer to the introduction of [Govaert and Nadif, 2013] for additional examples.

Text mining Co-clustering has been widely used in text mining in order to simultaneously cluster a set of documents (individuals) and the terms they contain (variables). The seminal reference in the domain is [Dhillon, 2001]. In such applications, a simple clustering is uninterpretable because of the very large number of terms used in the set of documents. Clustering these terms into column clusters allows to summarize the information and to exhibit groups of terms which are similarly used in each cluster of documents. In this field of text mining, these works have been dethroned by the advent of topic models, among which the seminal Latent Dirichlet Allocation (LDA, [Blei et al., 2003]), who introduced more flexibility. Indeed, in LDA the notion of cluster of terms is replaced by topics, which is a kind of soft clustering of the terms (each term belonging to each topic with different probability). Nevertheless, some recent works in co-clustering for text mining have been proposed in order to ease the reading of the co-clustering results, by designing explicitly which are the clusters of terms specific to each cluster of documents [Laclau and Nadif, 2016, Ailem et al., 2017, Selosse et al., 2020c] (see Section 5.4 for more details).

A close related field, when the documents are web pages, is web mining. Earlier works in this application domain are for instance [Charrad et al., 2009, Xu et al., 2010].

Bioinformatics Co-clustering is widely used in bioinformatics, and more specifically in genetic in order to identify some particular biomarkers. Most of the time, co-clustering is used as a two-way clustering, since in addition to detect cluster of genes (rows of $\mathbf{x}$), another entity has to be clusterized. For instance, [Hasan et al., 2018] use a robust co-clustering method in order to simultaneously cluster genes and their regulatory doses of chemical compounds, for application in toxicogenomic studies and in drug design and development. In human cancer microarrays study, [Cho and Dhillon, 2008] clusterized genes simultaneously with conditions, whereas in single-celle genomic study [Zeng et al., 2020] clusterized genes simultaneously with cells. Another example in [Chen et al., 2019], where samples are simultaneously clusterized with genes.

Medicine and public health Co-clustering has also be used in different medical applications, and more generally in the health field. Once again, co-clustering is generally used in this domain for its ability of performing a two-way clustering. For instance, [George et al., 2021] clusters simultaneously chromatin accessibility patterns and their associated hematopoietic lineage structure. Another example in brain mapping, where we often have to deal with connectivity graphs (or matrices) between two regions of the brain. [Cheng and Liu, 2021] proposed a method based on co-clustering for separating theses two regions into functionally homogeneous brain subregions. Their method is based on the spectra co-clustering techniques from [Huang et al., 2020]. In medical imaging, most automatic methods perform image analysis, as for instance tumor segmentation, on mono-modal images. In [Lian et al., 2019], a co-clustering algorithm is used to concurrently segment 3D tumors in positron emission tomography-computed tomography images. Taking into account the two complementary imaging modalities can combine functional and anatomical information to improve segmentation performance. In public health, co-clustering has also been used to co-clusterize data matrices, where one dimension of the matrix corresponds to the spatial dimension of the data. In [Ullah et al., 2017], the second dimension is the temporal dimension, and their goal is then to extract spatio-temporal clusters according to the number of occurrences of a specific disease. In [Darikwa et al., 2019], the second dimension corresponds to cardiovascular conditions, and co-clustering allows them to assess joint spatial autocorrelations between mortality rates due to cardiovascular conditions.

Computer vision We can also find application of co-clustering in computer vision. In such field, models are commonly defined either with regard to low-level concepts such as pixels that are to be grouped, or with regard to high-level concepts such as semantic objects that are to be detected and tracked. In [Keuper et al., 2020], co-clustering is used to perform these two tasks simultaneously.

Recommender systems Finally, let's discuss the use of co-clustering in recommender systems. Recommender systems are powerful and popular tools for e-commerce which seek to predict preferences according to the user's choices in term of movies, music, books, research articles, *etc.* By performing simultaneous clustering of users and items, co-clustering can be used in order

to predict preferences of users. For instance, in [George and Merugu, 2005], simple prediction using the average preference of the co-clusters is considered. Nevertheless, the applications in the field remain quite confidential, and matrix decomposition methods are more generally used, without demonstrating any advantage over model-based co-clustering approach. Let finally cite a recent work, [Frisch et al., 2021a], who shows how co-clustering can be used in order to obtain a fair recommendation system.

3 Extending LBM

As often within the mixture modelling paradigm, the LBM offers a ground basis for proposing many appealing extensions. In this section we discuss such recent extensions, which concern the nature of the data, the relaxation of the LBM assumption, the use of LBM for graph clustering, the multiview co-clustering problem, and the extension of LBM to the tri-dimensional case.

3.1 Variable type diversity

In recent years, the LBM has been extended to the co-clustering of some other specific type of data: categorical ordinal, functional, and even mixed-type data. Figure 2 illustrates co-clustering results for these three type of data.

3.1.1 Ordinal data

Ordinal data is one particular type of categorical data, occurring when the categories are ordered. Such data are very frequent in practice, as for instance in marketing studies where people are asked through questionnaires to evaluate some products or services on an ordinal scale. However, contrary to nominal categorical data, ordinal data have received less attention from a statistical modelling point of view, and then, in face of such data, the practitioners often transform them into either quantitative data (associating an arbitrary number to each category, see [Kaufman and Rousseeuw, 1990] for instance) or into nominal data (ignoring the order information, see the Latent GOLD software [Vermunt and Magidson, 2005]) in order to "recycle" easily related distributions.

[Jacques and Biernacki, 2018] considered the LBM in which the p.d.f. $f(\cdot;\boldsymbol{\alpha}_{kl})$ corresponds to the Binary Ordinal Search (BOS) distribution [Biernacki and Jacques, 2015] of parameter $\boldsymbol{\alpha}_{kl} = (\mu_{kl}, \xi_{kl})$. The BOS distribution, proposed by modelling the data generative process, is parametrized by a position parameter $\mu_{kl} \in \{1, m\}$, which is the unique mode of the distribution if $\xi_{kl} > 0$, and a precision parameter $\xi_{kl} \in [0, 1]$. This unimodal distribution evolves continuously from the uniform distribution when $\xi_{kl} = 0$ to a Dirac distribution in μ_{kl} when $\xi_{kl} = 1$. The BOS distribution involves a marginalization over several latent variables, resulting from the modelled data generation process. Consequently, the inference of this LBM model relies on the SEM-Gibbs algorithm (described in Section 2.4) containing an additional stage in the SE step, in which these latent variables are simulated according to the simulated value of $\mathbf{z}$ and $\mathbf{w}$. This co-clustering model has shown his interest in applications in Psychology [Selosse et al., 2019b] and Marketing [Jacques and Biernacki, 2018].

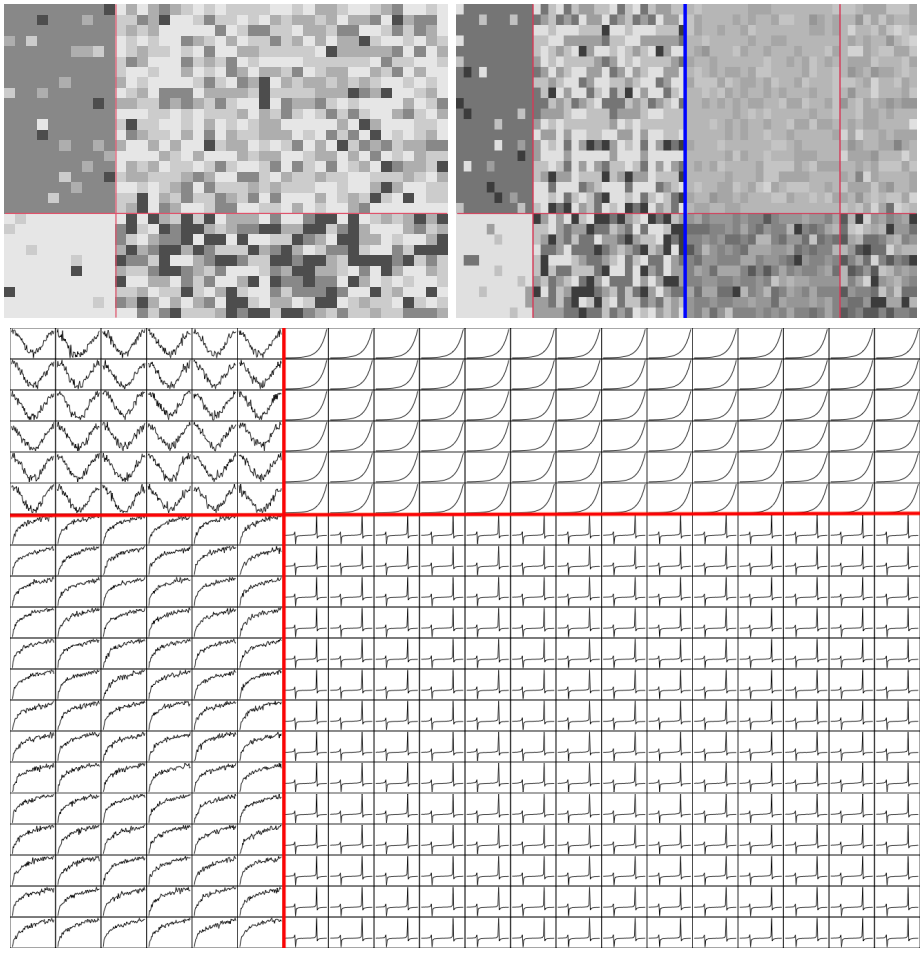

Figure 2: Illustration of co-clustering. *Top left:* ordinal data; *Top right:* mixed-type data; *Bottom:* functional data

An alternative model for ordinal data has been proposed in [Corneli et al., 2020]. It consists, as [McParland and Gormley, 2013] did in the clustering case, to assume that an ordinal variable is the discretization of a latent continuous variable. If this approach is interesting from a modelling point of view, the main difficulty with such a latent variable approach is to estimate the discretization thresholds. If [McParland and Gormley, 2013] consider them as model parameters to be estimated, [Corneli et al., 2020] fix them *a priori*, which is equivalent to code each ordinal category by an integer and to model them by a Gaussian distribution.

3.1.2 Functional data

Functional data occur when some quantity is observed over a continuum, often the time but not necessarily. Each element of the data matrix $\mathbf{x}$ that we want to co-cluster is then a function $x_i^j = x_i^j(t)$ with $t \in [0, T]$. The paradox when working with functional data, is that we never observed directly the infinite-dimensional functions $x_i^j(t)$, but only their values $x_i^j(t_s)$ a finite, but generally large, set of times. Consequently, a first step in functional data analysis is often to start by recovering the functional nature of the data, and this is generally done by assuming that the function $x_i^j(t)$ can be approximated into a finite basis of functions (as B-spline, Fourier...): $x_i^j(t) = \sum_{b=1}^{B} c_i^{jb} \psi_b(t)$. Such functional modelling allows a parsimonious modelling of regular curves, whereas their finite-dimensional vector of observations $x_i^j(t_s)$ is high-dimensional and highly correlated. But to take this advantage, one has to pay the price for the infinite-dimensional nature of the functions, and in particular in a model-based approach, since the notion of p.d.f. is not defined for such data. Nevertheless, [Delaigle and Hall, 2010] shown that the notion of p.d.f. can be approximated by the p.d.f. of the first Karhunen-Loeve expansion coefficients, which can itself be related to the p.d.f. of the basis expansion coefficients $c_i^j = (c_i^{jb})_b$.

[Bouveyron et al., 2018] considered the LBM in which the p.d.f. relate to the basis expansion coefficients c_i^j and is assumed to be a parsimonious multivariate Gaussian. Such a model is an extension of [Bouveyron and Jacques, 2011, Jacques and Preda, 2013] initially proposed in a clustering context. Let remark that contrary to the co-clustering of continuous data in which the p.d.f. is assumed to be a univariate Gaussian, the p.d.f. is in the functional case a multivariate Gaussian. This model has then be extended on the same principle to the case where x_i^j are multivariate functional data [Bouveyron et al., 2022]. After having approximated the curve into the basis of functions by least square smoothing, the inference of these functional LBM models inference is performed through a SEM-Gibbs algorithm (Section 2.4). These models have been applied to the co-clustering of electricity demand curves and to some Pollution and Climatology indicators.

Finally, [Goffinet et al., 2021] proposed an original Bayesian non-parametric LBM approach applied to multivariate time series. This methodology, called Functional Non-Parametric LBM (FunNPLBM), simultaneously creates a partition of observations and a partition of temporal variables, using latent multivariate Gaussian block distributions through a bi-dimensional Dirichlet Process as a prior for the block distributions parameters and for the block proportions. An interested consequence of this method is, since relying on the Bayesian non-parametric paradigm, to natively integrate model selection in the whole process.

3.1.3 Mixed-type data

Let now consider that the data matrix $\mathbf{x}$ is composed of S sets of features, each set corresponding to one type of data (continuous, binary, categorical nominal, ordinal, contingency or event functional). $\mathbf{x}$ has n rows and $d = \sum_{s=1}^{S} d_s$ columns, d_s being the number of features of the s-th type:

$$\mathbf{x} = (\mathbf{x}^1, ..., \mathbf{x}^S), \text{ with } \mathbf{x}^s = (x_i^{sj})_{i=1,...,n;\ j=1,...,d_s}.$$

The main idea proposed in [Selosse et al., 2020a] is to cluster the column inside each set of features of the same type, and not to group together features of different types. The space partitions for the column respecting this constraint is denoted by $\tilde{\mathcal{W}}$. Consequently, each set of features has to be clustered into L_s columns.

For this, [Selosse et al., 2020a] proposed to use the Multiple LBM, initially proposed in [Robert, 2017] in order not to group together variables linked to different information on the data (drug prescription variables and adverse effects variable). According to this model, each of the S sets of features is independent conditionally to $\mathbf{z}$ and $\mathbf{w}$. The p.d.f. is consequently an extension of (3), restricting the space of possible column partitions $\tilde{\mathcal{W}}$ and the number of column partitions depending on the feature type:

$$f(\mathbf{x};\boldsymbol{\theta}) = \sum_{(\mathbf{z},\mathbf{w})\in\mathcal{Z}\times\tilde{\mathcal{W}}} \prod_{i,k} \pi_k^{z_{ik}} \prod_{s=1}^{S} \prod_{j=1}^{d_s} \prod_{\ell=1}^{L_s} \rho_j^{w_{j\ell}^s} \prod_{i,k} \prod_{j=1}^{d_s} \prod_{\ell=1}^{L_s} f(x_i^{js};\boldsymbol{\alpha}_{k\ell})^{z_{ik}w_{j\ell}^s},$$

where $w_{j\ell}^s = 1$ is equal to 1 if column j of the s-th set of features belongs to cluster ℓ. Let remark that such a model greatly increases the number of models to evaluate in order to choose the number of co-clusters, since we have to choose K and $L_1, \ldots, L_T$. For this reason, [Selosse et al., 2020a] proposed a heuristic search into the space of possible number of clusters.

3.1.4 Textual interaction data

[Bergé et al., 2019] defined the Latent Topic Block Model (LTBM), to deal with textual interaction data involving two disjoint sets of individuals/objects. They take as example comments given by buyers on the products or services they bought. Hence, $\mathbf{x}$ is the binary matrix of interaction of customers on rows and products on columns. If $x_i^j = 1$, a set $W_{ij} = \{W_{ij}^d, d = 1, \ldots, D_{ij},\}$ of D_{ij} documents is associated with the connection between i and j: these documents are all the reviews made by i on the product j. The connections $\mathbf{x}$ are modelled with a classical binary LBM, see Section 2.2. Then, a document W_{ij}^d is composed of N_{ij}^d words. Each word $W_{ij}^{dn}, n = 1, \ldots, N_{ij}$ within a document follows a mixture distribution over a set of latent topics whose number K is unknown and must be estimated. Contrarily to a classic latent Dirichlet allocation (LDA, [Blei et al., 2003]), the specificity is here that the mixture proportions only depend on the row cluster k of the ith row of $\mathbf{x}$ and the column cluster ℓ of the jth column of $\mathbf{x}$. The authors introduced a new set of latent variables attached to the topic of a word in a document. They infer the model via a variational version of the EM algorithm and derive a ICLbic model selection criterion.

3.2 Relaxing partially parsimony/Flexibility increasing

Although co-clustering has advantages over other high dimensional techniques (especially in the number of free parameters), the model is fairly restrictive because, in the Gaussian case, all observations in a block are realizations of independent and identically distributed Gaussian random variables with mean $\mu_{k\ell}$ and variance $\sigma_{k\ell}^2$. Obviously, more flexibility can be obtained by fitting more column clusters and row clusters, but this is not always possible or advisable.

[Gallaugher et al., 2020] recently proposed a so-called parameter-wise co-clustering method by clustering columns according to both means and variances.

Similar to traditional co-clustering, their proposal preserves the concept of a partition in rows and columns. However, now there are two partitions in the columns; specifically, a partition with respect to means and a partition with respect to variances. The partition in columns by means is represented by $\mathbf{w}^{\mu} = (\mathbf{w}_1^{\mu}, \ldots, \mathbf{w}_d^{\mu})$, where

$$\mathbf{w}_j^{\mu} = (w_{j1}^{\mu}, \ldots, w_{jL^{\mu}}^{\mu}) \sim \mathcal{M}(1; \boldsymbol{\rho}^{\mu})$$

with $\boldsymbol{\rho}^{\mu} = (\rho_1^{\mu}, \ldots, \rho_{L^{\mu}}^{\mu})$ and the partition in columns by variances is denoted by $\mathbf{w}^{\Sigma} = (\mathbf{w}_1^{\Sigma}, \ldots, \mathbf{w}_d^{\Sigma})$, where

$$\mathbf{w}_j^{\Sigma} = (w_{j1}^{\Sigma}, \ldots, w_{jL^{\Sigma}}^{\Sigma}) \sim \mathcal{M}(1; \boldsymbol{\rho}^{\Sigma})$$

with $\boldsymbol{\rho}^{\Sigma} = (\rho_1^{\Sigma}, \ldots, \rho_{L^{\Sigma}}^{\Sigma})$. These two partitions in the columns is where the main novelty lies. Note that K, L^{μ} and L^{Σ} are the number of row clusters, column clusters by means, and column clusters by variances, respectively.

This co-clustering extension leads to the following observed log-likelihood

$$f(\mathbf{x}; \boldsymbol{\theta}) = \sum_{\mathbf{z} \in \mathcal{Z}} \sum_{\mathbf{w}^{\mu} \in \mathcal{W}^{\mu}} \sum_{\mathbf{w}^{\Sigma} \in \mathcal{W}^{\Sigma}} p(\mathbf{z}; \boldsymbol{\pi}) p(\mathbf{w}^{\mu}; \boldsymbol{\rho}^{\mu}) p(\mathbf{w}^{\Sigma}; \boldsymbol{\rho}^{\Sigma}) f(\mathbf{x} | \mathbf{z}, \mathbf{w}^{\mu}, \mathbf{w}^{\Sigma}; \boldsymbol{\mu}, \boldsymbol{\Sigma}),$$

where

$$p(\mathbf{z}; \boldsymbol{\pi}) = \prod_{i=1}^{n} \prod_{k=1}^{K} \pi_k^{z_{ik}}, \quad p(\mathbf{w}^{\mu}; \boldsymbol{\rho}^{\mu}) = \prod_{j=1}^{d} \prod_{l^{\mu}=1}^{L^{\mu}} (\rho_{l^{\mu}}^{\mu})^{w_{jl^{\mu}}^{\mu}}, \quad p(\mathbf{w}^{\Sigma}; \boldsymbol{\rho}^{\Sigma}) = \prod_{j=1}^{d} \prod_{l^{\Sigma}=1}^{L^{\Sigma}} (\rho_{l^{\Sigma}}^{\Sigma})^{w_{jl^{\Sigma}}^{\Sigma}}, \quad \text{and}$$

$$f(\mathbf{x} | \mathbf{z}, \mathbf{w}^{\mu}, \mathbf{w}^{\Sigma}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{i=1}^{n} \prod_{k=1}^{K} \prod_{j=1}^{d} \prod_{l^{\mu}=1}^{L^{\mu}} \prod_{l^{\Sigma}=1}^{L^{\Sigma}} \left[\frac{1}{\sqrt{2\pi}\sigma_{kl^{\Sigma}}} \exp\left\{ -\frac{1}{2\sigma_{kl^{\Sigma}}^2} (x_{ij} - \mu_{kl^{\mu}})^2 \right\} \right]^{z_{ik} w_{jl^{\mu}}^{\mu} w_{jl^{\Sigma}}^{\Sigma}}.$$

In terms of notation, $\boldsymbol{\mu} = (\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K)$, where $\boldsymbol{\mu}_k = (\mu_{k1}, \ldots, \mu_{kL^{\mu}})$. Note that $\mu_{kl^{\mu}}$ is the mean for row cluster k and column cluster by means l^{μ}. Likewise, $\boldsymbol{\Sigma} = (\boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_K)$, where $\boldsymbol{\Sigma}_k = (\sigma_{k1}^2, \ldots, \sigma_{kL^{\Sigma}}^2)$ and $\sigma_{kl^{\Sigma}}^2$ is the variance for row cluster k and column cluster by variances l^{Σ}. A SEM Gibbs algorithm is then used for maximizing the log-likelihood and also it is possible to compare this proposal to the standard co-clustering method though any model selection strategy.

The number of free parameters in the parameter-wise co-clustering model is

$$\begin{aligned} \#\text{Params}_{\text{new coclust}} &= K - 1 + L^{\mu} - 1 + L^{\Sigma} - 1 + KL^{\mu} + KL^{\Sigma} \\ &= K + (L^{\mu} + L^{\Sigma})(K + 1) - 3. \end{aligned}$$

There are a few comparisons with traditional co-clustering that are now discussed. First, similar to traditional co-clustering, the number of free parameters for the proposed parameter-wise method is independent of the dimension, meaning a high degree of parsimony is still maintained. Before mentioning the second point, note that the column clusters by means and column clusters by variances can be combined. For example, columns in column cluster 1 by means and column

cluster 1 by variances can be combined to form one column cluster. In general, columns in column cluster l^{μ} by means and column cluster l^{Σ} by variances can be combined to form one column cluster for any combination of l^{μ} and l^{Σ}, leading to a maximum of $L^{\mu}L^{\Sigma}$ column clusters. They can, however, be fewer than $L^{\mu}L^{\Sigma}$ combined column clusters because it is possible, for example, that no columns are clustered into column cluster 3 by means and column cluster 2 by variances. Now, assuming K is equal for both parameter-wise and traditional co-clustering, and $L^{\mu} = L^{\Sigma} = L$, then there are only an additional $L-1$ free parameters when using the parameter-wise model. Although there are these additional free parameters, there is the possibility of L^2 combined column clusters, allowing for a finer partition of the columns and increased flexibility.

There is also the possibility that the parameter-wise model has fewer free parameters than traditional co-clustering while still maintaining similar flexibility. For example, if traditional co-clustering is considered with $K = 4$ and $L = 5$, then the total number of free parameters is 47. In the parameter-wise case, if $K = 4$, $L^{\mu} = 3$, $L^{\Sigma} = 3$, then the total number of free parameters is 31. In this case, there is a possibility of a total of nine column clusters compared to five column clusters when using traditional co-clustering. Figure 3 illustrates this combination of column clusters by means and by variance.

3.3 Graph clustering and co-clustering

Recently, [Keribin, 2021] proposed an original application of LBM as a way to give another insight on the interpretation of the clustering of a directed graph. The Stochastic Block Model (SBM) for graph clustering has been introduced in Section 2.5: it aims to cluster the n nodes of a graph according to their connection profile. This model can be seen as a constraint LBM where $n = d$ and $\mathbf{z} = \mathbf{w}$. If, for any nodes i and j, edge (i, j) and edge (j, i) are the same, the graph is said to be undirected: its adjacency matrix $\mathbf{x}$ as well as its connection parameter matrix $\boldsymbol{\alpha}$ are symmetrical. If not, some edge $i \to j$ can exist while edge $j \to i$ does not exist: the graph is directed. In this case, $\mathbf{x}$ is non-symmetrical and a non-symmetrical $\boldsymbol{\alpha}$ is usually used for the inference. Then, the clustering of the graph builds a unique partition of the nodes, either the graph is undirected or directed. This choice is pertinent for undirected graphs, but [Keribin, 2021] discussed it for directed graphs because it implies that no difference is made between the clusters of source and target nodes.

Consider a graph where the nodes can have K different emitting profiles and L different receiving profiles. Clustering with SBM can lead to a model up to KL clusters by combining all the possibilities of source and target node profiles: $\boldsymbol{\alpha}$ can have up to $KL \times KL$ coefficients. A lot of coefficients would have the same value, but this could not be taken into account in the model which is in some manner over parameterized. LBM however, being more flexible with the two partitions in rows and columns, will be also more parsimonious as the size of matrix $\boldsymbol{\alpha}$ will be $K \times L$. Moreover, it can natively cluster nodes with the same emitting profile although they do not have the same receiving profile.

[Keribin, 2021] illustrated on real data sets that co-clustering can help to give a higher level of representation for the simple graph clustering. However, when the number of nodes is small, the two methods can give different results with less clear cross interpretation.With its propensity to give a greater number of small clusters, SBM clustering could be preferred for some applications where small clusters are of special interest. [Keribin, 2021] generally recommended to begin

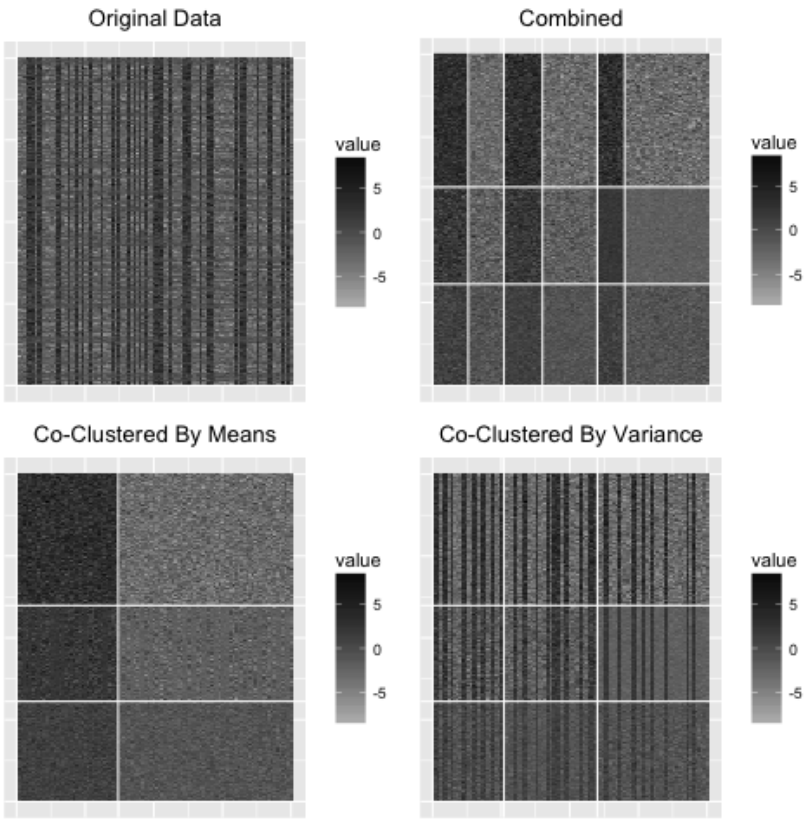

Figure 3: Illustration of the parameter-wise principle.

with (single) clustering, whether the graph is oriented or not; then, performing an additional co-clustering for oriented graphs can give a higher insight on the data set. If co-clustering gives much less blocks than the single node clustering, it brings a valuable information on the presence of (a lot of) identical row or column profiles, and reveals some specific constraints on the connection parameters between the SBM clusters. If LBM provides more blocks than SBM, both results should be questioned.

3.4 Multiview (co-)clustering

Multiview clustering Multiview data correspond to data sets viewed from different angles or in different modalities (for instance both audio and visual information), situation which is now frequent. The goal of multiview clustering is to cluster individuals into subgroups using such multiview data in order to arrive at a more effective and accurate grouping than what can be achieved by just using one view of data. Many multiview clustering methods exist, including

MBC extensions, as described in the recent survey [Chao et al., 2021].

Multiview co-clustering Multiview co-clustering is itself an extension of multiview clustering to the two-way clustering (or co-clustering), where both samples and variables are clustered. Such an approach is for instance considered in [Wang et al., 2018] through the matrix tri-factorization principle. Alternatively, [Tokuda et al., 2017] proposed a nonparametric Bayesian mixture model co-clustering approach, which is useful for analysis of high-dimensional data containing heterogeneous types of variables. With some similarities to the mixed data types described in Section 3.1.3, the authors gather different distribution families, such as Gaussian, Poisson, and multinomial distributions in each cluster block.

3.5 Multiway clustering

Multiway clustering has not to be confounded with multiview (co-)clustering that was discussed in the previous section. Multiway tensor arrays, which are a generalization of matrices, are the new data sets to be considered. They are frequently encountered in practice, for example in neuroimaging analysis where multi-dimensional images across multiple subjects or conditions naturally form a higher-way data array, or also in genomics where gene expression data collected at different times or locations can also be represented as a tensor. Multiway clustering is finally a natural extension of co-clustering which can be simply defined as a two-way clustering. This research direction is recent but very active, due to the need to analyze more numerous tensor data sets.

A seminal approach was proposed by [Robert et al., 2015, Robert, 2017] as part of a pharmacovigilance application. Pharmacovigilance deals with the spontaneous report of adversarial effects of drugs after their marketing authorization. A pharmacovigilance database contains the individual reports, each of them consisting of the list of prescribed drugs and observed effects for a given individual. Hence two binary matrices, one for the drugs, the other for the adversarial effects share the same rows, namely the individuals concerned by the reports. These authors introduced the multiple latent block model (MLBM) by extending the LBM through the construction of one row partition and two columns partitions, one for the drugs, the other for the effects.

[Selosse et al., 2019a] considered dynamic count data, meaning that occurrences of events are enumerated over several different time periods. It leads to a tri-way data sets, such a specific cubic tensor. The proposed approach develops a LBM tri-clustering algorithm relying on the Poisson distribution and on a variational EM algorithm. [Marchello et al., 2022] designed a dynamic latent block model (dLBM), which extends the classical binary LBM for analysis to dynamic cases where data are counts. They applied it to temporal contingency tables of drugs and adversarial effects in pharmacovigilance database, allowing to detect abrupt changes and providing a tool for automatic safety signal detection.

[Li, 2020] proposed another non-Gaussian multiway clustering relying on a tensor decomposition method. Non-Gaussian data are modeled with single-parameter exponential family

distributions and a regularized alternating (iteratively reweighted) least squares algorithm is proposed.

[Chi et al., 2020] developed a convex formulation of tensor co-clustering which allowed to obtain related estimators theoretical guarantees. In particular, they obtained a provable convex formulation of tensor co-clustering which reveals a surprising "blessing of dimensionality" phenomenon that does not exist in usual vector or matrix-variate cluster analysis. This interesting output has been already underlined for co-clustering in Section 5.2.

[Boutalbi et al., 2020] extended LBM to the case of a tri-dimensional tensor data by proposing the so-called Tensor LBM (TLBM) approach, while considering continuous, binary, and contingency tables data sets. They developed a variational EM algorithm and also developed a specific open-source Python package (TensorClus) [Boutalbi et al., 2022].

4 LBM and estimation issues

Mixture models are subject to numerous estimation difficulties (local maxima, empty cluster phenomenon, degeneracy) as attested by an abundant, and still productive, literature on this fundamental statistical topic. Consequently, it is natural to check in detail LBM estimation properties, and it will appear some specific issues that, as far as we known, are not yet completely identified and/or fixed in the relative literature.

4.1 MBC positioning

Multiple local maxima Estimation issues related to the MBC case are essentially implicated by the existence of possible multiple local maxima of the log-likelihood. This phenomenon has been identified a long time ago (see for instance [McLachlan and Peel, 2000, Sections 2.5] and is still an active domain of research (see for instance [Jin et al., 2016]). An obvious consequence of such local maxima in the log-likelihood function is that many EM algorithm runs (or of its variants) would be trapped by them, highlighting a very central importance to the choice of starting values of the estimation algorithm [Biernacki et al., 2003]. But beyond some "classical" local maxima, the mixture likelihood can suffer from so-called pathological maxima, specific to the mixture definition itself, namely empty clusters, degeneracy and spurious. We describe all of them now, since it will be relevant to then consider expected co-clustering estimation issues from these particular points of view.

Empty cluster solutions In the famous K-means algorithm [MacQueen, 1967], the early stopping of a run due to a partition including an empty cluster is well-known. Since the EM algorithm is a very close version of K-means, it is not surprising that EM (and many of its variants) may suffer from the same pathology (which corresponds to a component such that $\pi_k^{(q)} \simeq 0$ for some iteration step q). When the number of components is high, this event can be even particularly frequent, leading for instance to use this property for selecting the number of components by gradual elimination of empty components (see for instance [Malsiner-Walli et al., 2016] or [Forbes et al., 2019]).

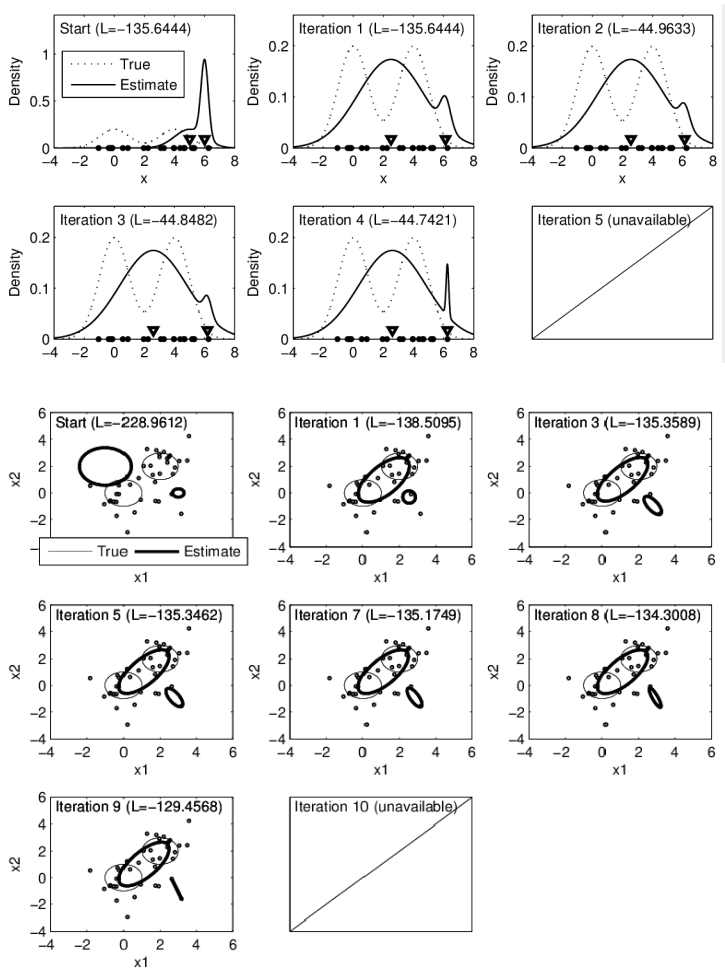

Figure 4: Degeneracy illustration of the EM algorithm in the univariate (top panel) and bivariate (bottom panel) Gaussian mixture cases.

Degenerate solutions In heteroscedastic Gaussian mixtures, the likelihood is known to be unbounded. For instance, in the univariate framework a so-called degenerated solution can be reached by setting the mean of one component equal to an observed data and letting tend its variance to zero [Day, 1969]. More generally, in the d-variate case, when for a given $k \in \{1, \ldots, K\}$ the corresponding means relies on the simplex of a sub-sample of size d and the corresponding generalized variance $|\Sigma_k| \to 0$, then $\ell(\boldsymbol{\theta}; \mathbf{x}) \to \infty$. However, such degenerated solutions rely on the border of the parameters space and thus are out of interest. Indeed, a root of the gradient of the likelihood is searched because it is known that one of them is consistent [Redner and Walker, 1984]. In practice, when the EM algorithm encounters a degenerated

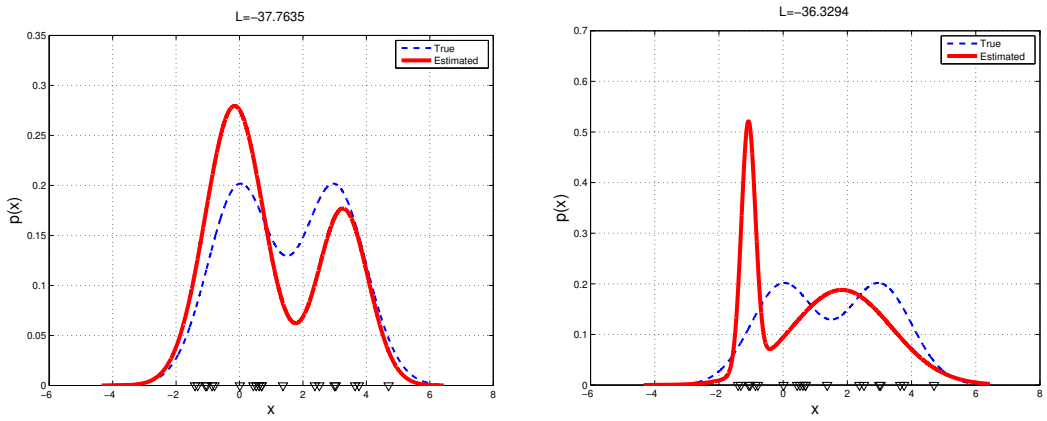

Figure 5: Spurious local maximizer illustration in the univariate Gaussian mixture case: convergence of EM towards either a "normal" solution (left panel) or a spurious one (right panel). Note that the likelihood is higher for the spurious case, even if not infinite at all (it is not a degenerate case).

solution, it acts like a trap as studied in the univariate case by [Biernacki and Chrétien, 2003] and in the multivariate case by [Ingrassia and Rocci, 2007], with in addition an exponential convergence degeneracy rate. Figure 4 illustrates a degeneracy situation both in the univariate and in the bivariate case, and we can observe their associated EM dynamic also.

It is also interesting to notice that degeneracy is not necessary limited to unbounded likelihood cases. Typically, in univariate Gaussian mixture models with binned data the likelihood stays bounded, however the degeneracy problem remains. Indeed, when all the non-empty intervals are small enough, the global maximum of the likelihood is located on the border of the parameters space [Biernacki, 2007]. In this case the EM algorithm can still be trapped by a degenerated solution.

Spurious local maximizers The problem of spurious local maximizers of the likelihood is introduced in [McLachlan and Peel, 2000, Sections 3.10] for the multivariate Gaussian case. Such spurious solutions typically have to be distinguished from the degenerate solutions just previously discussed. Indeed, they correspond to observing a relatively small (but nonzero) generalized variance (determinant of the covariance matrix) in at least one component and may lead to a quite large local maximum. But, since one or several covariance matrices are close to degeneracy, such local maxima can lead to very large, albeit *finite*, values of the likelihood. However, this latter may sometimes be larger than the local maximum corresponding to a "good" mixture estimate, although they do not correspond to some reality about the expected estimate parameter at all. Figure 5 illustrates such a specific spurious situation in a univariate Gausian mixture example. It is thus important to notice that an EM algorithm can again be trapped in such a solution, without possibility to discard it from the log-likelihood value point of view despite several restarts.

4.2 LBM: parameters vs latent variables

We have noticed in Table 1 that LBM is drastically more parsimonious than MBC. However, this parameter parsimony is not necessary a guaranty for limiting the number of local maxima of the likelihood associated to LBM. Indeed, an antagonist effect can be produced by the great increase of the space of latent variables in LBM. More precisely, MBC leads to a latent space of size K^n whereas of size $K^n L^d$ for LBM, thus corresponding to an additional order of magnitude.

Let us illustrate through a simple example the fact that increasing the number of latent variables may have a direct consequence for increasing the number of local maxima of the likelihood. Let notice that it corresponds to an example related to the MBC situation, since the log-likelihood is not computationally available in the LBM case. However, we expect that the impact of increasing the number of latent variables should be similar. Thus, we consider a sample of size $n = 1\ 000$ from a two-component univariate Gaussian mixture with proportions $\pi_1 = \pi_2 = 0.5$, means $\mu_1 = -0.8,\ \mu_2 = 0.8$ and variances $\sigma_1^2 = 1,\ \sigma_2^2 = 1.5$ (see an illustration of this mixture in Figure 6). All the parameters are supposed to be known, except the means μ_1 and μ_2. In addition, a given proportion of the latent variables is assumed to be known, and varies within the sequence $\{60\%, 30\%, 15\%, 0\%\}$. Figure 7 illustrates that a new local maximum gradually appears when the percent of latent variables increases, although the number of parameters is extremely low. And as a straightforward consequence, the EM solution (thus the detected local/global maximum) can highly depends on its starting position as it is already well-known.

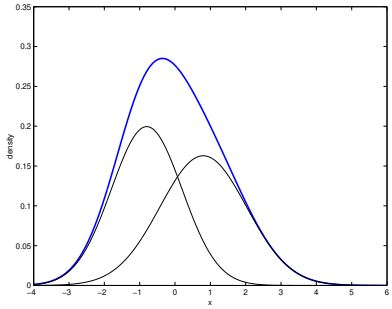

Figure 6: The simple univariate Gaussian mixture case used for illustrating local maxima existence due to latent variables existence.

We expect that this toy illustration could be transposed to the LBM case, meaning that: 1. the number of likelihood local maxima for LBM could be drastically higher than this one observed with MBC and 2. choosing the starting values is of primary importance for avoiding too many local traps in the likelihood local maxima. However, since the likelihood is numerically intractable, it is interesting to see whether this expected property on the likelihood is observed on the ELBO, which is a likelihood surrogate. This will be the objective of the numerical experiments given in Section 4.5.

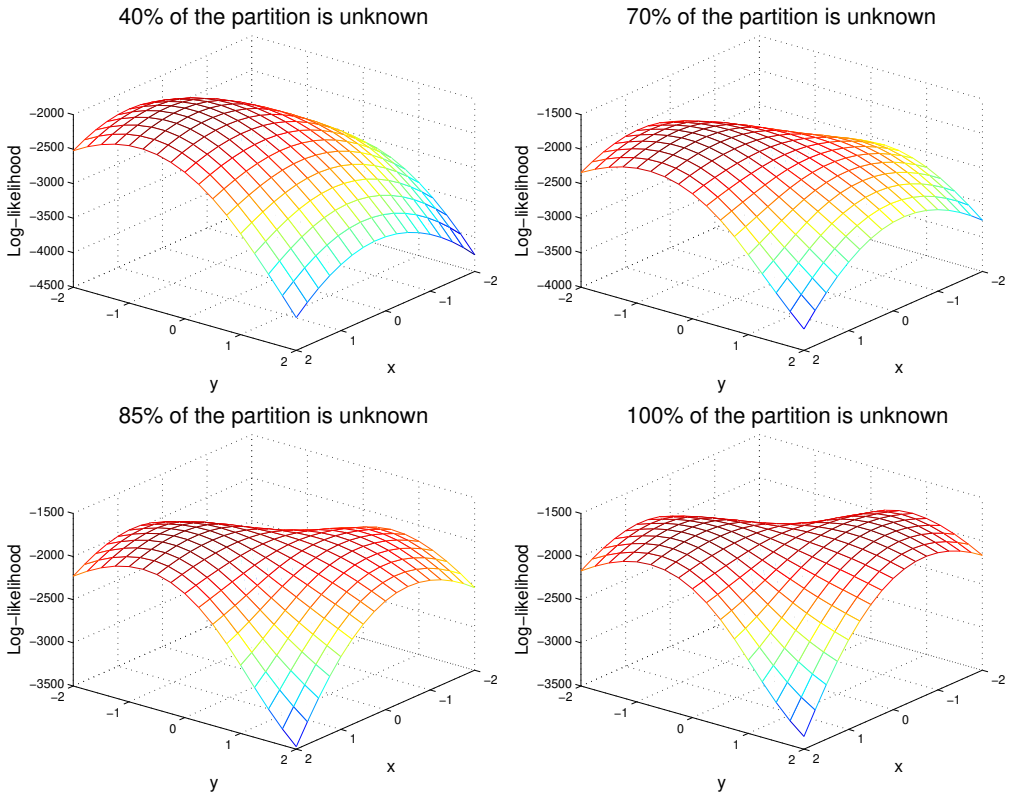

Figure 7: Effect of the number of latent variables (here the partition) on the number of log-likelihood local maxima, the number of mixture parameters being kept constant.

4.3 Empty blocks solutions in LBM

In co-clustering, initialization problems leading to algorithm failures due to empty block outputs during a VEM or a SEM estimation procedures are regularly reported. We can see for instance [Brault, 2014] which even devotes a specific section of his PhD thesis on this fact and also proposes a Bayesian variant of VEM (V-Bayes) for limiting this phenomenon. Many practitioners also complain about such frequent empty block fails when they use packages related to co-clustering (see for instance [Selosse et al., 2020a]). In fact, even if authors not systematically report such failures (it is often considered as side or negligible information within the experiment description), we give below some arguments suggesting that the empty block phenomenon could be even drastically more present than the empty cluster event in the clustering context.

The key idea to understand the event of empty clusters/blocks is to measure their frequency in the partition latent space. Indeed, any estimation algorithm is expected to be more attracted by such trap situations if these latter are more frequent in the latent space.

For the clustering situation, the set of empty clusters can be denoted by

$$\mathcal{Z}^0 = \{\mathbf{z} : \text{it exists at least one } k \text{ such that, for all } i,\ z_{ik} = 0\}.$$

Some classical combinatorial algebra lead to the frequency of empty cluster cases equal to

$$\#\mathcal{Z}^0 = K^n - S(n, K)K!,$$

where $S(n, K)$ corresponds to the Stirling number of the second kind. Symmetrically, for the co-clustering situation, the set of empty blocks is obtained by

$$(\mathcal{Z} \times \mathcal{W})^0 = \{\mathbf{z}, \mathbf{w} : \text{it exists at least one } (k, l) \text{ such that, for all } (i, j),\ z_{ik}w_{jl} = 0\},$$

leading then to the related frequency of empty block cases

$$\#(\mathcal{Z} \times \mathcal{W})^0 = K^n L^n - S(n, K)S(d, L)K!L!.$$

It can then be easily deduced, by using some properties of $S(.,.)$ that the ratio of the number of empty blocks over the number of empty clusters goes to infinity with the dimension, meaning that this phenomenon is much more present in co-clustering than in clustering:

$$\frac{\#(\mathcal{Z} \times \mathcal{W})^0}{\#\mathcal{Z}^0} \to \infty \text{ when } n \to \infty \text{ or } d \to \infty.$$

More precisely, the speed for going to infinity is extremely high as it can be illustrated in the special case $n = d$ and $K = L$. Indeed, we obtain

$$\frac{\#(\mathcal{Z} \times \mathcal{W})^0}{\#\mathcal{Z}^0} = K^n + S(n, K)K! > K^n.$$

This exponential speed can be observed even with very small sample sizes and small numbers of clusters/blocks within the following two examples:

$$\frac{\#(\mathcal{Z} \times \mathcal{W})^0}{\#\mathcal{Z}^0} = \begin{cases} 62 & \text{when } n = d = 5 \text{ and } K = L = 2, \\ 710\,768 & \text{when } n = d = 9 \text{ and } K = L = 4. \end{cases}$$

In conclusion, we highly suspect that obtaining non-empty blocks output during the estimation step in co-clustering is much more challenging than it is in clustering for non-empty clusters. This is one of the reasons why the initialization of the inference algorithms is often achieved by performing independent clustering of rows and columns (see Section 2.7).

4.4 Degenerate and spurious local maximizers solutions in LBM

Degenerate solutions To the best of our knowledge, no degeneracy problems are reported in co-clustering, whereas in clustering degeneracy situations are frequent in practice, and are even specifically studied through some works (see references in Section 4.1). In a similar way as the previous empty block failures case, we attempt here to explain this fact through a dedicated formalization.

As in clustering, degeneracy in co-clustering is simply defined by a situation where parameter estimates rely on the border of their parameter space. For instance, in the Gaussian case, it corresponds to a null variance. Such a situation happens when an estimation run is attracted by a block containing just a unique element. The proposed formalization is now restricted to this Gaussian case for simplification. And moreover, we consider only diagonal Gaussians for the clustering case, allowing a certain symmetry with the co-clustering Gaussian assumptions, which is quite required for comparing precisely both methods.

Following the same spirit as the previous empty clusters/blocks study, the key idea to understand the phenomenon of degeneracy is to measure the frequency of obtaining clusters or blocks with a single element in the whole partition latent space. Indeed, any estimation algorithm is expected to be more attracted by such trap situations if these latter are more frequent in the latent space.

For the clustering situation, the set of clusters where, say, $\mathbf{x}_1$ is the single element in a cluster (even if other clusters can be empty and also may contain a single element themselves) can be denoted by

$$\mathcal{Z}_1^1 = \left\{ \mathbf{z} : \text{it exists at least one } k \text{ such that } z_{1k} = 1 \text{ and } \sum_{i=2}^{n} z_{ik} = 0 \right\}.$$

Some basic calculus leads to the following *relative* frequency of partitions having $\mathbf{x}_1$ alone in a cluster:

$$\frac{\#\mathcal{Z}_1^1}{\#\mathcal{Z}} = \left(\frac{K-1}{K} \right)^{n-1}.$$

Symmetrically, for the co-clustering situation, the set of blocks where, say, x_1^1 (meaning the 1st individual and the 1st variable) is the single element in a block (not excluding that some other blocks can be empty or have also a single element themselves) can be written

$$(\mathcal{Z} \times \mathcal{W})_{1,1}^1 = \left\{ \mathbf{z}, \mathbf{w} : \text{it exists at least one } (k,l) \text{ such that } z_{1k} w_{1l} = 1 \text{ and } \sum_{i=2}^{n} \sum_{j=2}^{d} z_{ik} w_{jl} = 0 \right\}.$$

It leads then to the following related *relative* frequency of such single element cases

$$\frac{\#(\mathcal{Z} \times \mathcal{W})_{1,1}^1}{\#(\mathcal{Z} \times \mathcal{W})} = \left(\frac{K-1}{K} \right)^{n-1} \left(\frac{L-1}{L} \right)^{d-1}.$$

Consequently, we obtain the following relationship between both previous relative frequencies:

$$\frac{\#(\mathcal{Z} \times \mathcal{W})_{1,1}^1}{\#(\mathcal{Z} \times \mathcal{W})} = \left(\frac{L-1}{L} \right)^{d-1} \frac{\#\mathcal{Z}_1^1}{\#\mathcal{Z}},$$

meaning that degeneracy situations are expected to be drastically less present in co-clustering than in clustering. As a numerical illustration of this claim, with $L = 4$ and $d = 50$ (which corresponds to a quite simple co-clustering case), co-clustering degeneracy is expected to occur 7.5510^{-7} times less than classical (Gaussian diagonal) clustering. This result could explain the reason why the degeneracy problem is not visible in the co-clustering literature.

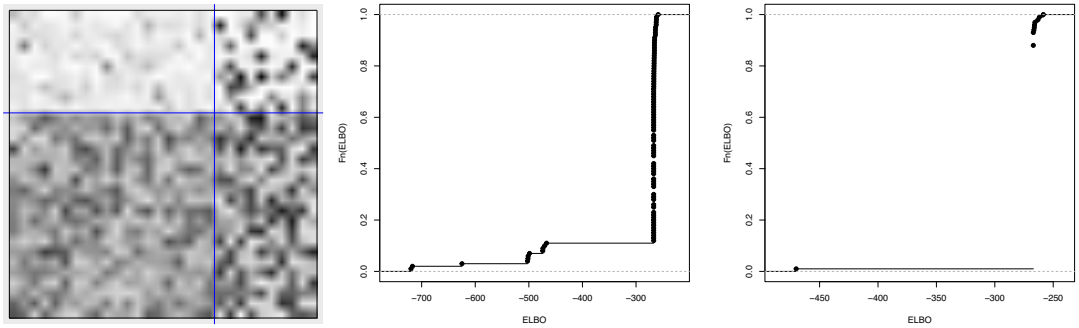

Figure 8: *Left*: Gaussian 2×2 LBM; e.c.d.f. of ELBO values obtained from $B = 100$ initializations with standard precision on *Center* and high precision on *Right*

Spurious local maximizers Similarly to previously discussed degenerate situations, spurious local maximizers never appear in the co-clustering literature, at least as far as we know. We think that it can be explained in the same manner as for degeneracy. More precisely, in the Gaussian case, spurious local maximizers may happen only when a block contains two individuals. And this event is expected to be highly less frequent in co-clustering than in clustering, by using similar combinatorial arguments than we used for degeneracy (we do not detail this calculus here, which is more tedious than for degeneracy).

4.5 Local maxima in LBM

In this section we present simulations to illustrate how difficulties arising in the estimation of mixture models can be also observed for LBM, such as (1) the existence of multiple local maxima (2) the propensity of the estimation algorithm to converge to empty cluster solutions, especially when a LBM of higher dimension is fitted; (3) the necessity to design smart initializations to avoid to be trapped in local solutions.

Gaussian LBM We first consider a Gaussian LBM with $(K, L) = (2, 2)$ clusters, mixing weights $\boldsymbol{\pi} = (1/3, 2/3)'$ in row and $\boldsymbol{\rho} = (2/3, 1/3)'$ in column and the following conditional distributions:

$$\left(\begin{array}{c|c} \mathcal{N}(\mu_{11} = 2, \sigma^2_{11} = 1) & \mathcal{N}(\mu_{12} = 2, \sigma^2_{12} = 2^2) \\ \hline \mathcal{N}(\mu_{21} = 4, \sigma^2_{21} = .5^2) & \mathcal{N}(\mu_{12} = 4, \sigma^2_{22} = 1) \end{array}\right).$$

An example of such a simulated LBM data set with $n = 30$ rows and $d = 30$ columns is displayed Figure 8-left. The estimation is performed with `blockcluster` running VEM (`algo=BEM`) from $B = 100$ random initializations and default parameters (`epsilonxem` $=10^{-4}$, `epsilonXEM` $=10^{-10}$, `nbiterationsXEM` $= 50$). The empirical cumulative distribution function (e.c.d.f.) of the resulting ELBO values is plotted on Figure 8-middle. It lets appear a large number (56) of different values. However, one does not have to jump so easily to the conclusion of numerous

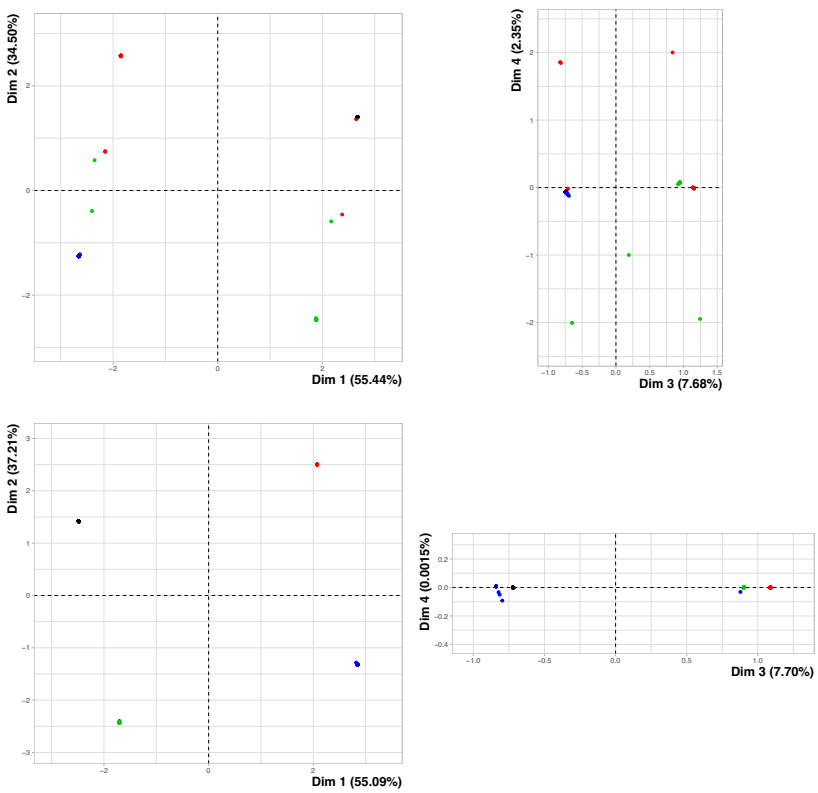

Figure 9: PCA of the parameters (row and column weights, means, variances) obtained after $B = 100$ initialisations. Top: convergence with standard precision; Bottom: convergence with high precision. Colors are set according to the cluster sizes: black ($\widehat{\rho} i_1 < \widehat{\pi}_2$ and $\widehat{\rho}_1 < \widehat{\rho}_2$), red ($\widehat{\pi}_1 > \widehat{\pi}_2$ and $\widehat{\rho}_1 < \widehat{\rho}_2$), blue ($\widehat{\pi}_1 < \widehat{\pi}_2$ and $\widehat{\rho}_1 > \widehat{\rho}_2$), green ($\widehat{\pi}_1 > \widehat{\pi}_2$ and $\widehat{\rho}_1 > \widehat{\rho}_2$).

local minima. In fact, using `blockcluster` with a higher precision (`epsilonxem` = `epsilonXEM` $=10^{-16}$, `nbiterationsXEM` = 10 000) leads to only 9 different values (see Figure 8-right):

ELBO	-470.1	-267	-266.5	-266.2	-266	-265.6	-263	-261.8	-258.4
count	1	92	1	1	1	1	1	1	1

The ELBO values are now clearly distinct, and Figure 9-bottom displays the parameters in the first two principal planes of a PCA of the parameters (row and column weights, means, variances) representing 99.98% of the total inertia. On the first principal plane we clearly observe the four permutations (two for row clusters, two for column clusters), while local maxima appear on the second principal plane. In fact, to be certain that these isolated values are really local maxima (and not that they denote a very flat ELBO), one would have need to access the number of iterations at the convergence, information not output by `blockcluster`. To compare with

the standard precision setting, Figure 9-top also displays projections of the parameters (row and column weights, means, variances) obtained with standard precision on the first two principal planes of PCA representing 99.98%. Here, these projections are more spread out than in the high precision setting, the algorithm often stopped before reaching the optimum in the standard precision setting. This reveals some potential slow convergence effect. However in both cases (standard and high precision), even if there could be large differences in ELBO values, all the 100 initializations lead to the same partition in row and column: clustering is easier than estimation and estimating with too much precision is not necessarily useful.

Using now a LBM with $(K, L) = (2, 3)$ to infer the data, without changing the generation scheme, leads to the degeneracy of a column cluster up to one variable. ELBO and parameter values have similar behaviors than those in the well specified case.

Binary LBM The previous simulations seem to present a certain stability, and this should be connected to the fact that `blockcluster`, even departing from random initializations, performs several small VEM steps before choosing the most promising one. Hence, it natively proposes a mean to deal with the initialization problem. To explore more deeply the existence of local maxima or degeneracy, we now use the package `bikm1` which allows to perform one direct run from a user initialization on both partitions. Moreover, it outputs the number of iterations used to produce the results, which is helpful to determine slow convergence pathology. However, `bikm1` only deals with binary and Poisson LBM, and we choose a binary LBM with $K = L = 3$. To assert the difficulty of the classification task, we use reference samples defined by [Lomet et al., 2012a], and begin with a sample $n = d = 50$ with a Bayes error of 5% (see Figure 10-left). Three hundred initializations of three types are launched: equally weighted random draws, weighted random draws, random draw of a sub-partition of rows (columns), and extension to all the rows (columns) by K-means. The generated partitions for started the algorithm are checked to be all different. The estimation with `bikm1` and high precision leads to only three (nearly two) different values:

ELBO	−1615.48	−1615.43	−1596.67
count	9	38	254

High ELBO values generate 36 co-clustering configurations, corresponding to all possible row and column cluster relabellings. In this case, the convergence is fast and only few iterations are necessary (less than 100 iterations), see Figure 10-middle-left. In another hand, low ELBO values correspond in fact to 30 different configurations, all with an empty row or column cluster; these configurations suffer from a very lazzy convergence (some thousands of iterations), as displayed on Figure 10-middle-right. Figure 10-bottom displays the two principal PCA planes gathering almost 100% of the explained variance. We see the 36 red points corresponding to the 36 relabeling situations; the isolated black points correspond to low ELBO values.

Simulations with $n = d = 100$ and $n = d = 200$ give similar results. However, a sample with a Bayes error of 20% and $n = d = 50$ drastically fails to recover the partition, see Figure 11, all low ELBO value cases being caused by empty clusters.

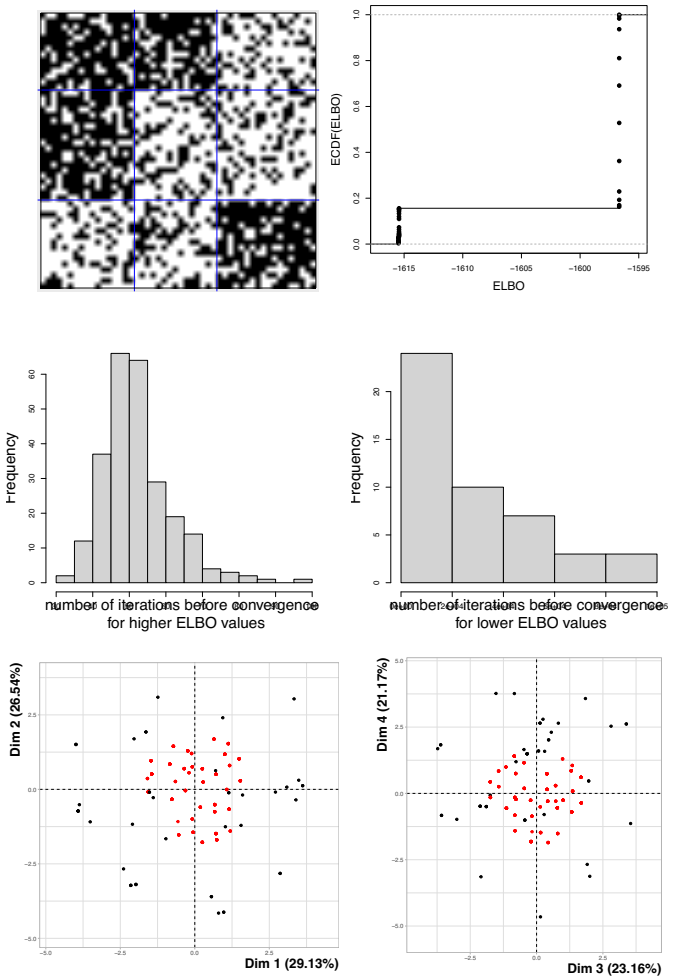

Figure 10: Binary 3×3 LBM with low Bayes classification error (5%) estimated from $B = 300$ random initializations.
Top: data set (*Left*); ECDF of the ELBO values (*Right*).
Middle: histograms of the number of iterations before convergence: cases with high ELBO values have fast convergence (*Left*) whereas cases with low ELBO values have lazzy convergence (*Right*).
Bottom: two first principal PCA planes of the estimated parameters: cases with high (low) ELBO values are colored in red (black).

Discussion We saw in these simulations that LBM can be easily trapped into local maxima, these being frequently due to empty cluster solutions. In case of non empty clusters, lower ELBO

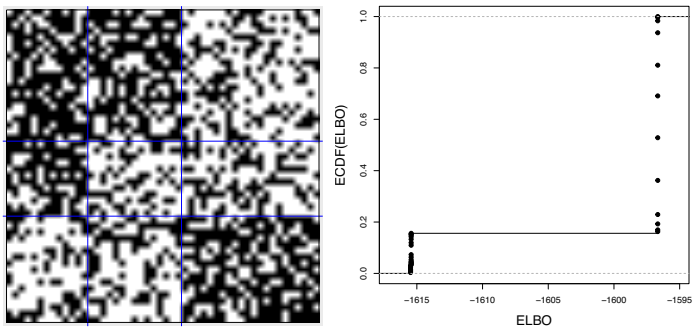

Figure 11: *Left*: Binary 3×3 LBM with Bayes error 20%; *Right*: e.c.d.f. of ELBO values obtained from $B = 300$ random initializations.

solutions can exist without jeopardizing the clustering which remains identical as the clustering of the higher ELBO values.

One would have thought that it could be easy to exhibit degenerate solutions or spurious local maximizers. However, we did not encounter such cases throughout our experiments. This could have two main reasons: first, the likelihood value is not available and the VEM algorithm optimizes a lower bound of the likelihood; hence it can be possible that this lower bound is more regular than the likelihood and less prone to local minima such as the one represented on Figure 7. Second, the block structure acts as a natural regularization, and could prevent to have simultaneously a row cluster with only one individual and a column cluster with only one variable, which is the case to get a degenerate solution in heteroscedastic Gaussian simple mixtures.

In fact, to deal with these difficulties, software have to propose smart initializations. We saw the regularization effect of the initialization of `blockcluster`. Next section describes different initialization strategies.

4.6 Initialization strategies

As discussed earlier in the section, the main problem in LBM estimation is due to traps into a solution with empty block. In order to avoid such situations, the most commonly used solution is to initialize the algorithm as well as possible. Initializations available in the main co-clustering packages are described in Section 2.7. There are of several types: 1. with two independent clusterings of the rows and the columns; 2. using several initializations with other algorithms (CEM, SEM, EM-VBayes) on few iterations; 3. by resampling in order to avoid to get empty blocks during the first iterations; 4. [Leger, 2016] proposes a smart and robust initialization, combining an absolute eigenvalues spectral clustering adapted for LBM and a reinitialization strategy: forward exploration of the space of models (K, L) by splitting already existing clusters as in `bikm1` [Robert, 2017] combined with a backward exploration by merging groups. Reinitialization is done while it improves the criterion. Hence, even for a required number of blocks, the

initialization always begins with a 1×1 LBM.

5 LBM and data dimensionality

Current data sets present an increasing number of variables, such as some hundreds for marketing studies, 10^2 to 10^4 variables for often less than a hundred observations in gene expression in microarray study or 10^4 to 10^5 voxels for only few tens of images in fRMI context (see section 2.8 for other instances). In these last two examples, the number of observations is even smaller than the number of variables. This is often known as the *high dimension* (HD) framework in the statistical community ($n \sim d$, $n < d$ or $n \ll d$); in the data science community, the extreme situation $n \ll d$ is referred to *fat* data, in opposition to *big* data ($n \gg d$ and n extremely large). This causes specific statistical and computational problems. We will see in this section their impact on MBC and develop our view that co-clustering can be a pertinent answer.

5.1 MBC positioning

Standard model-based clustering is known to be very efficient for low dimensional data sets. However, with data sets having a very large number of variables, MBC faces positive and negative effects of the dimension growth that have to be discussed. More precisely, effects are not the same from the estimation or clustering perspectives [Bouveyron and Brunet, 2014, Biernacki and Maugis, 2017].

5.1.1 HD density estimation: curse of dimensionality

MBC requires the estimation of a parameter whose size naturally increases with the dimension d of the observed space: in the Gaussian case, this rate of growth is d for the mean and d^2 for a full covariance matrix, see Table 1 Section 2.2. Hence, density estimation quality mechanically decreases as the number of variables d increases for the same number of observations n. This is clearly illustrated on Figure 12, which draws against d the estimated Kullback divergence of the estimate of a d-variate spherical Gaussian mixture defined by the two following components:

$$(M_1): \quad \mathbf{x}_i | z_i \sim \mathcal{N}_d(\boldsymbol{\mu}_{z_i}, \mathbf{I}_d), \quad \boldsymbol{\mu}_1 = \mathbf{0}_d, \quad \boldsymbol{\mu}_2 = \mathbf{1}_d, \quad \pi_1 = \pi_2 = \frac{1}{2},$$

where $\mathbf{0}_d$ and $\mathbf{1}_d$ denote a d-variate vector of 0 and 1 values, respectively. Moreover, dimension growth obviously leads to computational issues such as increasing needed resources (execution time, memory) or ill-conditioned systems. This latter can arise for example when computing an inverse covariance matrix in the E-step of the EM algorithm (in the general covariance matrix case).

5.1.2 HD clustering: blessing and curse

The objective of clustering is quite different from that of density estimation. Clustering could take advantage of some useful properties of these HD spaces, namely the fact that they are almost empty, and that most of the observations often lay in subspaces of low dimensions

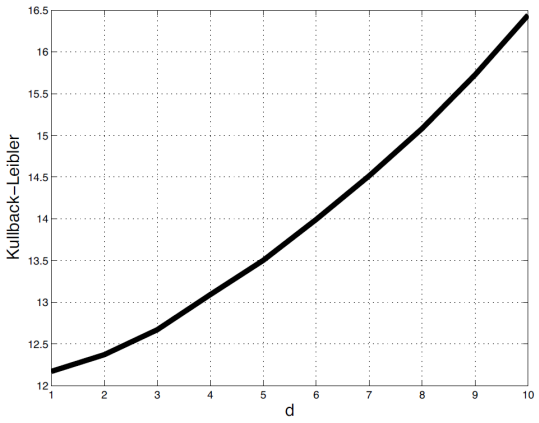

Figure 12: Kullback-Leibler divergence of a d-variate mixture density estimate when d increases.

[Bellman, 1957]. Even though the densities are very poorly estimated, clusters could be successfully retrieved if they are sufficiently separated. In fact, the blessing or curse of the dimension growth will depend on the discriminant quality of the added variables.

Blessing Let us consider again example (M_1) introduced in the previous section: the two components are more and more separated when d grows since $||\mu_1 - \mu_2|| = \sqrt{d}$. This is illustrated in Figure 13, where samples of the same size from (M_1) with four different dimensions ($d = 2, 20, 50, 200$) are projected on the main two factorial PCA axes. In this favorable case, each new variable is informative, and the theoretical classification error $err_{theo} = \Phi(-\sqrt{d}/2)$ decreases with d, where Φ is the cumulative distribution function (c.d.f.) of a univariate Gaussian random variable $\mathcal{N}(0, 1)$. The empirical error rate also decreases when MBC uses the spherical Gaussian model, until the number of variables is not too large. Figure 14-left depicts this behavior for $d \leq 20$, where the theoretical Bayes error is drawn in black, and the empirical error using MBC is in cyan. The error also decreases, although more slowly, when inferring with a more complex model than the generative (true) one, for example the following correlated Gaussian model denominated by (M_2):

$$(M_2): \quad \mathbf{x}_i | z_i \sim \mathcal{N}_d(\boldsymbol{\mu}_{z_i}, \boldsymbol{\Sigma}_d(c)), \quad \boldsymbol{\mu}_1 = \mathbf{0}_d, \quad \boldsymbol{\mu}_2 = \mathbf{1}_d, \quad \pi_1 = \pi_2 = \frac{1}{2}$$

with $(\boldsymbol{\Sigma}_d(c))_{jj'} = c < 1$ for all $1 \leq j \neq j' \leq d$ and $(\boldsymbol{\Sigma}_d(c))_{jj} = 1$. The empirical error is represented by the blue curve on Figure 14-left. In this case, there is *no model bias*, *estimation variance* is on control and the *classification error* ($\widehat{\mathbf{z}}$) decreases.

Consider now the case with a reasonable *estimation bias*, for example when the model estimation is performed with a spherical Gaussian model (M_1) instead of the (true) correlated Gaussian model (M_2): All the new variables are informative, although correlated. The theoretical Bayes error decreases when d increases up to the limit error $\Phi(-1/(2\sqrt{c}))$, which is non

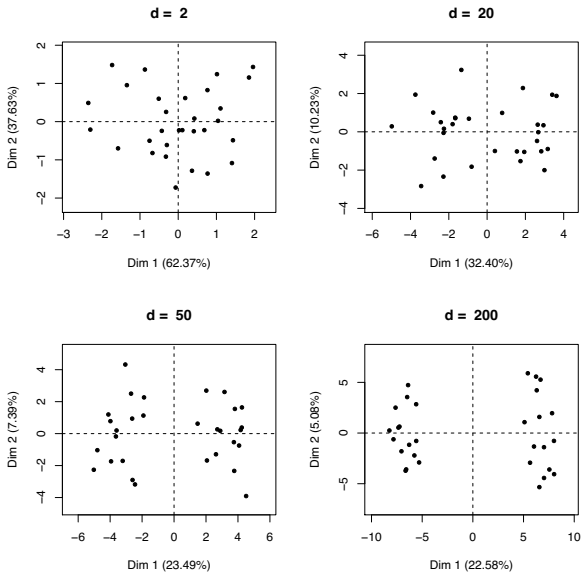

Figure 13: Principal Component Analysis (PCA) on the main two factorial axes of two Gaussian components more and more separated when the space dimension increases.

null when $c \neq 0$. Results on Figure 14-center show that the observed clustering error owns the same behavior when d increases, while the observed clustering error estimated with the true model is higher, and subject to divergence when d is to high: in this latter case, the curse of density estimation outweighs the blessing of having a greater number of informative variables and accepting some bias can help the clustering task in this HD setting.

Pitfalls However, when the added variables are less informative (*i.e.*, less discriminative), the blessing is lost. In Figure 14-right, we consider the following generating mixture model:

$$(M_3): \quad \mathbf{x}_i | z_i \sim \mathcal{N}_d(\boldsymbol{\mu}_{z_i}, \mathbf{I}_d), \quad \boldsymbol{\mu}_1 = \mathbf{0}_d, \quad \boldsymbol{\mu}_2 = \left(1, \frac{1}{2}, \ldots, \frac{1}{d}\right), \quad \pi_1 = \pi_2 = \frac{1}{2}.$$

The theoretical Bayes error still decreases when d grows, but more slowly and to a limit $\Phi(-\sqrt{\pi^4/90}/2)$ which is not null. Here, the empirical error rate does not decrease anymore: in that case, there is no model bias, but classification error increases as the separation is no longer improved enough by the dimension increase.

In any cases, blessing of dimensionality for clustering blows out when the number of variables is too large with regards to the number of observations, and model estimation fails. This is illustrated in following Section 5.3 on Figure 16-top left for $d > 200$ for the basic case and $d > 50$

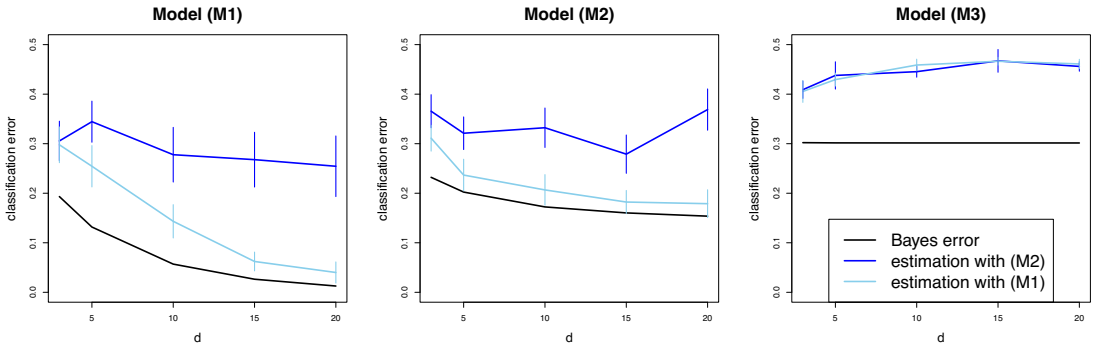

Figure 14: Classification error as a function of the dimension d, when $n = 30$ observations are generated from model (M_1) on the left, (M_2) on the center and $n = 100$ observations from model (M_3) on the right. In each case, the theoretical Bayes error is drawn in black, the estimation error with (M_1) in cyan and with (M_2) in blue. Errors are averaged on 50 replications and vertical bars represent 95% confidence intervals.

for the correlated one. This is due to the lack of regularization of the estimation process, not to the actual separation of the clusters.

Approaches for HD clustering In order to counterbalance this potential curse of dimensionality, several approaches have been proposed for MBC; see [Bouveyron and Brunet, 2014] for an extensive review. These authors split the earliest approaches into three families: dimension reduction methods (using PCA for example), regularization methods (tackling the numerical problem of the covariance matrix inversion by adding to it a positive matrix avoiding the degeneracy) and constrained and parsimonious models (introducing bias but limiting the variance, for instance by using a diagonal covariance matrix instead of a full one in the Gaussian case). However, they underline the risk to use dimension reduction methods without taking into consideration the clustering goal as it may lead to an irremediable loss of useful discriminating information. On the contrary, the regularization methods do not suffer from this drawback, but have a tuning parameter difficult to adjust in the unsupervised context. Finally, using constrained and parsimonious models in MBC represents an interesting trade-off between a bias modeling and a variance estimation.

In the latter context, some recent solutions adapt the idea of parsimonious modeling to exploit the empty space phenomenon of the HD setting: this is the family of the subspace clustering methods (*e.g.* mixture of factor analyzers and its extensions, mixture of parsimonious Gaussian mixture models). We refer to [Bouveyron and Brunet, 2014] for a detailed presentation of its taxonomy and the links between these methods. Note that the visualization of the resulting clusters could be difficult as they lie in specific and usually different subspaces. Moreover, the number of parameters could be large: about 1 500 to 2 000 for 4 clusters lying in subspaces of

dimension 3, with observations of $d = 100$ variables. Hence, if these methods suit for moderate dimension d, they lack of parsimony for higher dimension. Co-clustering, breaking the number of parameters from the dependence of the dimension d (*cf.* Table 1), is a convenient alternative.

More recent approaches propose to simultaneously cluster data and to reduce their dimensionality by selecting relevant variables regarding the clustering task. These have in view to assign different roles which is referred to *variable role modelling* in [Biernacki and Maugis, 2017]. In the case of Gaussian variables, [Maugis et al., 2009] proposed the SelvarClust algorithm, making it possible to classify the variables into three groups (informative, non-informative, and linearly dependent variables). [Sedki et al., 2014] proposed a lighter CPU-time consuming procedure based on the preceding one and [Marbac and Sedki, 2017] defined an alternative irrelevant variables approach. In a Bayesian framework, [Fop et al., 2017] defined two groups (informative, non-informative variables) for the analysis of latent classes. Refer to [Fop and Murphy, 2018] for a survey on variable selection in clustering. These methods take into account redundancy and variables utility, but related models are not suitable for too many variables due to performance limits (greater than few thousands). In other words, they stay limited to the cases $n \sim d$ or $n < d$, but the case $n \ll d$ is out of reach. We see now that LBM can act as a suitable modeling for addressing such a *very* HD case.

5.2 LBM and its blessing properties in HD clustering

Interpreting LBM as a MBC dimension reduction method PCA is certainly the most emblematic reduction dimension method for numerical data sets $\mathbf{x}$ (we consider in the following that $\mathbf{x}$ is centered in columns). It proceeds in two steps. First, it expresses each data unit $\mathbf{x}_i$ of the initial data set $\mathbf{x}$ in a new vector basis $(\mathbf{u}^1, \ldots, \mathbf{u}^d)$, ordered in the decreasing value of the preserved variance, with respective coordinates $(a_i^1, \ldots, a_i^d)$ and where each $\mathbf{u}^j$ is defined by a linear combination of the canonical vector basis $(\mathbf{e}^1, \ldots, \mathbf{e}^d)$, namely $\mathbf{u}^j = \sum_{j'=1}^d b_{j'} \mathbf{e}^{j'}$. Second, it selects just a reduced number of these new coordinates (say J), leading then to a new data set of smaller dimension ($J < d$). This PCA sequential procedure for the i-th data individual $\mathbf{x}_i$ can be expressed as follows:

$$\mathbf{x}_i = \sum_{j=1}^{d} x_i^j \mathbf{e}^j = \sum_{j=1}^{d} a_i^j \mathbf{u}^j \approx \sum_{j=1}^{J} a_i^j \mathbf{u}^j. \tag{7}$$

Consider now the (very) specific co-clustering case reduced to $K = 1$, thus meaning that just a variable clustering in L column clusters is performed. Let $\varepsilon_i^j \sim \mathcal{N}(0, \sigma^2_{\tilde{w}_j})$ in an i.i.d manner, where $\tilde{w}_j$ is the cluster index such that $\tilde{w}_j = \ell \Leftrightarrow w_{j\ell} = 1$. We can write, with $\mathbf{v}^\ell = \sum_{\{j:\tilde{w}_j=\ell\}} \mathbf{e}^j$ and $\mathbf{r}_i = \sum_{j=1}^d \varepsilon_i^j \mathbf{e}^j$:

$$\mathbf{x}_i = \sum_{j=1}^{d} x_i^j \mathbf{e}^j = \sum_{j=1}^{d} (\mu_{\tilde{w}_j} + \varepsilon_i^j) \mathbf{e}^j = \sum_{\ell=1}^{L} \mu_\ell \mathbf{v}^\ell + \mathbf{r}_i \approx \sum_{\ell=1}^{L} \mu_\ell \mathbf{v}^\ell. \tag{8}$$

This last approximation is justified by the fact that $\mathbb{E}[\mathbf{r}_i] = \mathbf{0}_d$.

By now comparing Equation (7) and Equation (8), we notice that LBM variable clustering performs a dimension reduction (the number of column clusters) in a novel vector basis which is itself a linear combination of the canonical vector basis, thus, exhibiting some strong similarities on these two points with PCA. Obviously criteria involved in both LBM and PCA are totally different, thus it is just the *functional expression* of both methods which is similar. We can also go further in this comparison. It is classical in PCA to force the linear coefficients $a_{j'}$ to be equal to values 0 or 1 for facilitating interpretation of new axes $\mathbf{u}^j$. In other words, we retrieve in this way a variable clustering process for PCA, which is in fact natively already involved in LBM.

Finally, coming back to the individual clustering target, PCA is *very popular* in the HD case. The process is to first reduce the dimension by PCA and then to apply a MBC on this reduced data set, what we could call a combined and sequential "PCA/MBC method". However, this procedure is not recommended since the dimension reduction step does not take into account the targeted classification task, as discussed for instance in [Bouveyron and Brunet, 2014]. On the contrary, LBM has the advantage to take into account the individual clustering *target simultaneously* within the dimension reduction process, while again including both PCA and MBC basic ingredients. Thus LBM could be interpreted somewhere as a "PCA/MBC-like" variant but expected to perform better by construction. We see below that such expected properties for LBM in the HD case are effectively obtained.

Properties of LBM in the HD case As pointed out by [Lomet et al., 2012a], co-clustering is subject to an unusual phenomenon in learning: for a given distribution on the entries of the table (*i.e.* a given number of blocks), the Bayes risk decreases as the table size grows. This is in contrast with most learning scenarios, where having more data usually leads to a better model estimation, but does not impact the Bayes classification risk. In a simple clustering point of view, adding new rows (individuals) while the number of columns (variables) is fixed leads to better estimate the underlying distribution as well as the classification risk; but this classification risk remains to have a lower limit due the intrinsic mixture situation. For co-clustering on the contrary, the decrease of the classification risk can be intuitively understood by considering that the table enlargement in one dimension results in more redundancy in the other dimension. In fact, under the true number of blocks, LBM recovers exactly the true labels of the rows (columns) when the number of columns (rows) tends to infinity.

In the binary co-clustering setting, this blessing phenomenon for the row clustering is studied in [Brault, 2014]. Noting $p(x_i^j = 1|z_i = k) = \tau_k = \sum_{l=1}^{L} \alpha_{kl}\rho_l$, then the distribution of $\sum_j x_i^j$ is a mixture of binomial distributions:

$$\sum_j x_i^j \mid z_i = k \sim \mathcal{B}(d, \tau_k).$$

In that case, [Brault, 2014] provides the following control of partition error $\mathbf{z}$ of this mixture, $\mathbf{z}^*$ denoting the true row partition:

$$p(\hat{\mathbf{z}} \neq \mathbf{z}^*) \leq 2n \exp\left\{-\frac{1}{8} d \left[\min_{k \neq k'} |\tau_k - \tau_{k'}|\right]\right\} + K(1 - \min_k \pi_k)^n.$$

It implies the important fact that row clustering is consistent in high-dimension provided some asymptotic constraints between n and d, for instance that $\ln(n) = o(d)$. This phenomenon is

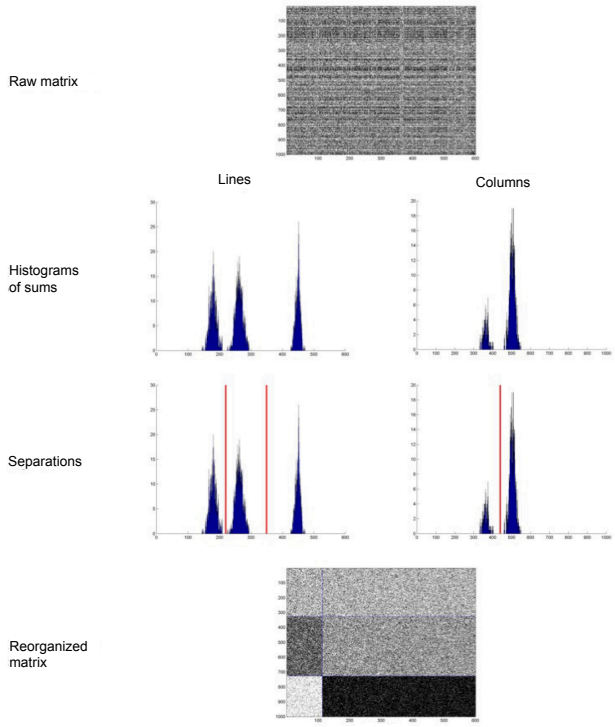

Figure 15: Illustration of the low row cluster overlap in the binary HD setting: The initial data matrix is at the top; Histogram of the rows sums (and columns sums) is displayed at the second line; The third line underlines that three row clusters (and two column clusters) are clearly present; The reorganized matrix (in row and columns) is available at the last line of the figure. This figure is an English version of the initial figure provided by [Brault, 2014].

illustrated on Figure 15, which shows the low row cluster overlap in a HD setting ($n = 1\ 000$ and $d = 500$): indeed, the histograms of the row (and columns) sums show that the clusters are well separated. These results can be related to the more general result on the consistency of the couple $(\hat{\mathbf{z}}, \hat{\mathbf{w}})$ we saw in Section 2.5 (see Equation (6)).

Notice that [Chi et al., 2020] recently established a non-asymptotic error bound for an estimator in the co-clustering of tensors, which is an extension of the co-clustering of matrices (see the discussion about so-called multi-way clustering in Section 3.5). This bound reveals also this surprising "blessing of dimensionality" phenomenon that does not exist in vector or matrix-variate cluster analysis.

LBM as a competitive candidate in HD clustering Hence LBM, defining a simultaneous partition of the rows and columns of a matrix, is a very parsimonious model (*cf.* Table 1) with interesting properties on label consistency. We advocate these properties make it a naturally

Table 2: Definition of the mean (in row) and variance (in column) of the generating models.

	$\mathbf{I}_d$	$\mathbf{\Sigma}_d(c)$
$\boldsymbol{\mu}_1 = \mathbf{0}_d$, $\boldsymbol{\mu}_2 = \mathbf{1}_d$	(M_1)	(M_2)
$\boldsymbol{\mu}_1 = \mathbf{0}_d$, $\boldsymbol{\mu}_2 = \left(1, 2^{-2}, \ldots, d^{-2}\right)$	(M_3)	(M_6)
(M_1) of size $n \times d/2$ duplicated twice	(M_4)	
$\boldsymbol{\mu}_1 \sim \mathcal{N}_d(\mathbf{0}, \mathbf{I}_d)$, $\boldsymbol{\mu}_2 \sim \mathcal{N}_d\left((\mathbf{1}_{\sqrt{d}}, \mathbf{0}_{d-\sqrt{d}}), \mathbf{I}_d\right)$	(M_5)	

regularized candidate for high-dimensional clustering, even if this is not its initial mission. It is not only parsimonious, but also robust to non informative variables and to redundancy. In fact, the clustering of columns can be seen as a strategy for a drastic control of the variance, but it generates bias, and it is interesting to study the effects on the classification error. The next section numerically illustrates this claim.

5.3 Numerical illustrations of LBM in HD clustering situations

This section illustrates in an empirical way the regularizing and beneficial effects of co-clustering strategy and its bias-variance trade-off behavior in scenarios involving HD fundamentals (correlated variables, irrelevant variables). These experiments show the ability of this approach to outperform simple mixture row clustering.

Six generating models are considered, with two balanced ($\pi_1 = \pi_2$) clusters in $\mathbb{R}^d$. Their means and variances are summarized in Table 2. In one hand, all the d variables are independent ($\mathbf{\Sigma} = \mathbf{I}_d$ for both components) and the models only differ by their means. In (M_1), each variable is informative for the clustering purpose ($\mu_1^j = 0$, $\mu_2^j = 1$ for $j = 1, \ldots, d$) although in (M_3) their discriminant power vanishes with d ($\mu_1^j = 0$, $\mu_2^j = 1/j^2$ for $j = 1, \ldots, d$). For (M_4), the $d/2$ first variables are the same as for (M_1), and the $d/2$ remaining ones are copy of the first ones. They are consequently strictly redundant variables. For (M_5), mean of each variable is itself drawn according to a Gaussian distribution ($\mu_1^j \sim \mathcal{N}(0, 1)$ and $\mu_2^j \sim \mathcal{N}(\mathbb{1}_{j \leq \sqrt{d}}, 1)$ for $j = 1, \ldots, d$), leading a different mean for each variable of each component. In another hand, correlation between the variables are considered with informative and separated means (M_2) or with informative means but with a discriminant power which decreases with d (M_6). Model (M_1) corresponds to a nominal LBM, although all other models break at least one of the LBM assumptions: conditional independence of the outcomes $(x_i^j | z_i = k, w_j = \ell)$ inside a given block for (M_2) and (M_6), no real block of same conditional distribution for the others.

Row clustering is then performed using four methods:

- clustering with a mixture of two spherical Gaussian distributions,
- clustering with a mixture of two full-covariance Gaussian distributions,
- co-clustering with a Gaussian LBM with $(K = 2 \times L = 1)$ blocks,
- co-clustering with a Gaussian LBM with $(K = 2 \times L = 2)$ blocks.

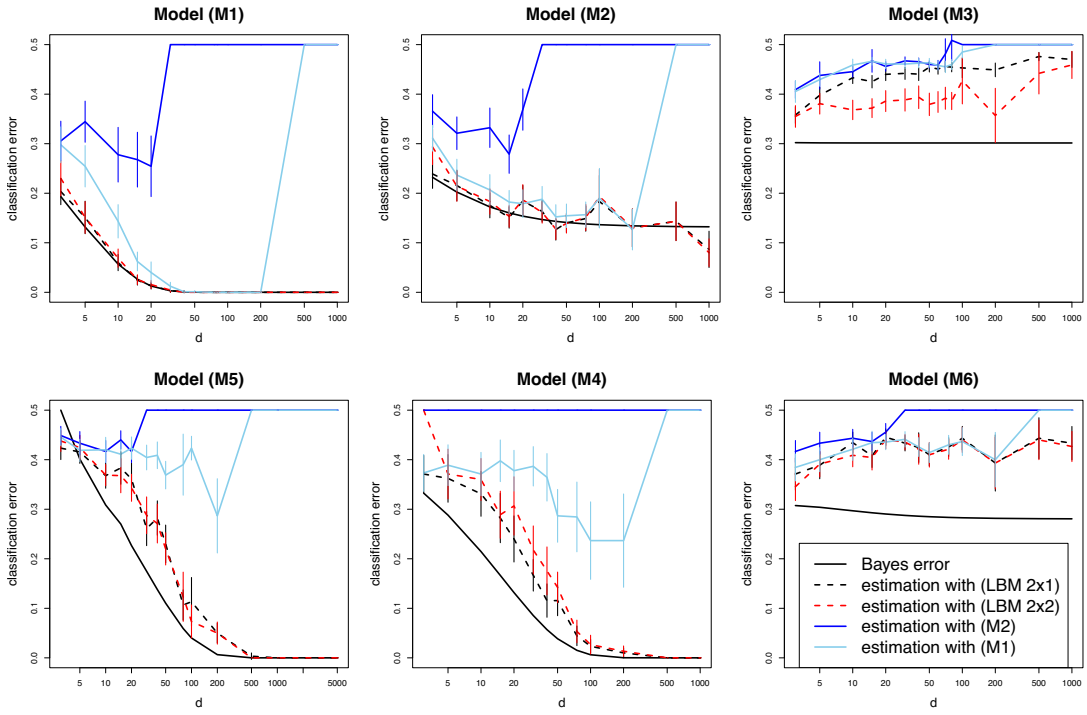

Figure 16: Influence of the number of variables d on the classification error for the six generative scenarios and the four clustering methods.
On top: (M_1) (*left*), (M_2) (*middle*), (M_3) (*right*). On bottom: (M_5) (*left*), (M_4) (*center*), (M_6) (*right*). Bayes classification error is drawn in black, empirical mean errors with (M_1) are represented in cyan, with (M_2) in dark blue, with co-clustering with a Gaussian LBM with $(K = 2 \times L = 1)$ blocks in dotted black, and with $(K = 2 \times L = 2)$ blocks in dotted red. Empirical error is averaged on 30 tables with $n = 30$ rows, vertical bars represent 95% confidence intervals.

The classification error of the six scenarios (M_1) to (M_6), averaged over 30 samples of size $n = 30$ is represented for these four methods in Figure 16, as well as the optimal Bayes error. The number d of variables evolves between 3 and 1 000. Notice that the three sub-figures on the first row represent the same generating models as those in Figure 14.

We easily note the beneficial regularization effect of the two co-clustering methods (in dotted lines) on the classification error. In fact, the estimated classification error tends to the Bayes error even if the true generative model is not used for the estimation, either because there is some correlation in the data $((M_2), (M_5))$ or because the means are not appropriate to LBM (M_5). In these cases, the bias introduced by co-clustering does not avoid its ability to recover the row clustering.

When the LBM assumption on the means is clearly violated ((M_3), (M_6)), classification error with co-clustering does not line up anymore on the Bayes risk, but slightly increases with the dimension d. However, co-clustering methods still perform better than simple mixture. In this case, defining more column classes is obviously better.

In all proposed scenarios, the clustering with simple mixture (M_1) (solid dark blue line) is never competitive even when this method is unbiased with regard to the generative model ((M_1), (M_4)).

These preliminary results illustrate co-clustering as a regularization tool for performing high-dimensional clustering. It offers an extremely parsimonious model, which, although generally biased, often provides excellent performance in classification which outperforms the performance of a simple mixture. It has for example the native property of grouping together exactly redundant variables (see results for model (M_6)), and can define clusters of non-informative variables. In the next section, a constrained version of LBM is presented, which imposes a cluster of non-informative variables.

5.4 Interpretability of LBM in high dimension

When the number of variables is large, for instance in textual analysis when working with document term matrix $\mathbf{x}$, the number L of variable clusters can be large, as well as the number of blocks, and therefore their analysis and interpretation could be difficult. It is the reason why [Selosse et al., 2020c] propose a structured version of the Poisson LBM, called Self Organized Co-Clustering (SOCC), which distinguishes noisy co-clusters from significant ones, and then reduce the number of significant blocks to be analyzed. Such a model is particularly useful when $\mathbf{x}$ is sparse, which is typically the case for textual data. In the LBM, each block parameter $\boldsymbol{\alpha}_{kl}$ is independent from each other, and should be interpreted separately. In the SOCC model, it is not true anymore: a structure is forced among the blocks so that the result is easier to read. Thus, for a given block (k, l), the corresponding block effect $\boldsymbol{\alpha}_{kl}$ will either be specific to column cluster l with $\boldsymbol{\alpha}_{kl} = \boldsymbol{\alpha}_l$, or non-specific, with $\boldsymbol{\alpha}_{kl} = \boldsymbol{\alpha}$. In this case of non-specific block effect, the block (k, l) is considered as a noisy or "non-meaningful" block, and it shares the same $\boldsymbol{\alpha}$ with all the other non-meaningful blocks. In the other case ($\boldsymbol{\alpha}_{kl} = \boldsymbol{\alpha}_l$), the block (k, l) is "meaningful", and shares the same $\boldsymbol{\alpha}_l$ with all the meaningful blocks of the same column cluster l.

To organize these meaningful and non-meaningful blocks, several rules are given. First of all, after choosing the number of row clusters K, the co-clustering necessarily has $L = K + \binom{K}{2} + 1$ column clusters. Moreover, the column clusters are divided into three sections called *main*, *second* and *common*. The *main* section concerns the first K column clusters, for $l \in \{1, ..., K\}$. In each column cluster l of this section, only one block is meaningful, parameterized by $\boldsymbol{\alpha}_l$. All the other blocks are non-meaningful and parameterized by $\boldsymbol{\alpha}$. If $\mathbf{x}$ is a document term matrix, for each cluster of documents (row cluster), the meaningful block indicates the terms that are specific to these documents. The *second* section concerns the following $\binom{K}{2}$ column clusters ($l \in \{K + 1, ..., K + \binom{K}{2}\}$). In each column cluster l of this section, two blocks are meaningful. Consequently, each column cluster contains terms that are specific to two clusters of documents (row clusters). Finally, the *common* section is made of only one column cluster and gathers the terms that are common to all documents.

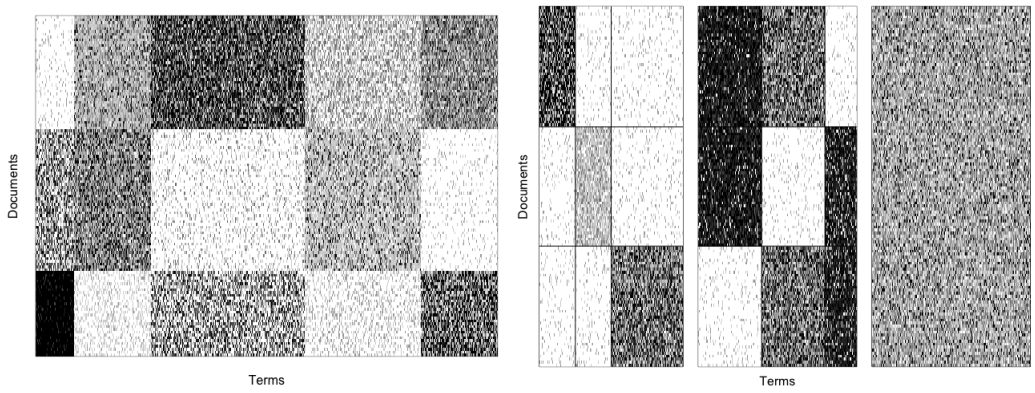

Figure 17: *Left*: the usual Poisson Latent Block Model: we see that some blocks are not easily classifiable into noisy or significant blocks. *Right*: the SOCC alternative approach: we easily distinguish the noisy blocks (lighter ones) and the significant ones (darker ones).

This structure is illustrated by Figure 17: we clearly see the meaningful blocks with $\boldsymbol{\alpha}_{kl} = \boldsymbol{\alpha}_l$ and non-meaningful blocks with $\boldsymbol{\alpha}_{kl} = \boldsymbol{\alpha}$.

SOCC estimation is performed in [Selosse et al., 2020c] through an SEM-Gibbs algorithm and model selection is obtained with the asymptotic ICL criterion. In addition to its interpretation properties, the parsimony of SOCC allows it to be competitive from a clustering point of view, as already discussed in the previous section for the more general LBM.

6 Conclusion and research avenues

LBM is fundamentally included in the mixture modeling paradigm and consequently inherits from its properties. On the first hand, LBM benefits from its high flexibility for approximating many kinds of complex distributions, allowing in particular to interchange some block distributions according to the data variables, to tune finely the convenient number of blocks, to be involved as a key ingredient for clustering applications, *etc.* A specific advantage of LMB however, among the wide mixture modeling family, is to reach a high flexibility degree despite a possibly extreme parsimony of parameters. This latter essential property leads to propose LBM as a very natural candidate when many variables are involved, typically in the so-called high dimension (HD) clustering context. It is the reason why we advocate in particular that using LBM in HD clustering should be more emphasized in the future. But, on the second hand, LBM is penalized by classical difficulties encountered by mixtures often related to the core estimation process itself, thus affecting both theorists and practitioners. Such complications in mixtures are essentially due to latent variables involvement which, antagonistically, are the basic ingredient for allowing flexibility of mixture modeling. In the LBM context, the number of latent variables being particularly high (one order of magnitude above classical mixtures), known problems in

mixture models are now exacerbated. Finally, all these advantages and difficulties conveyed by LBM encourage towards several research avenues where we list some of them below.

A first research avenue consists of exploiting the natural flexibility of LBM by proposing extensions specific to cases of interest, usually guided by some data specificity.

In this context, a particular emphasis on the missing data problem could be of interest, motivated by the fact that such events are mechanically more frequent in large data sets. Since both MBC and LBM reveal their interest when the data volume increases, it makes particularly sense to integrate the missing data problem in the modeling itself. The most considered missing data case is certainly the Missing At Random case (so-called MAR), where the missingness mechanism does not depend on the unobserved data values. There already exist some examples in LBM, as in [Selosse et al., 2019b, Selosse et al., 2020a, Frisch et al., 2022] and references herein. However, the Missing Not At Random case (so-called MNAR), where the missingness depends on the unobserved data values and possibly on the observed data values, is less studied both in MBC and in LBM even if some early works address this case in MBC with [Sportisse et al., 2021] and in LBM with [Corneli et al., 2020]. Noticing that the number of latent variables increases a lot in LBM in comparison to MBC (the unobserved partitioning in lines and also in columns), we argue that the MNAR situation is expected to be more frequent in LBM, and thus interest of this kind of modelling should increase in the future. Since it is well-known that MNAR proposals are not so simple to design, an interesting and simple family of MNAR proposals for LBM could be to rely on the work of [Sportisse et al., 2021], dedicated to MBC. This work considers that MNAR is simply obtained by conditioning the missing data pattern to the latent partition in line, leading to a very easy and flexible modeling family. A natural extension for LBM should then be to implement this conditioning to the double partition in lines and in columns simultaneously.

Another research direction could be to address spatio-temporal data [Cheam et al., 2017, Vandewalle et al., 2020, Bouveyron et al., 2022] in the LBM context, since such kind of data are increasingly frequent as well. Probably we could multiply such an LBM adaptation principle according data specificity which is encountered by the statistician.

Then, revealed along this paper, LBM abounds of open questions that are not really present, or at least not at this order of magnitude, in the more classical MBC case. It is crucial to properly address these issues in the near future for not limiting a promising larger use of LBM, for instance in the HD clustering task. We give here a list of some potential priorities in this direction.

Concerning first the estimation step, we have seen that designing new specific starting values strategies for estimation algorithms is necessary (see Section 4). Indeed, classical strategies used for the MBC case appear to usually fail in the LBM context which suffers from drastically more numerous empty blocks traps or also more numerous local maxima. Moreover, this phenomenon will increase with new extensions such as multiway co-clustering. Another kind of estimation difficulties relies on potential label switching both in line and column. Indeed, as noticed in Section 2.4, the SEM algorithm could be subject to the label switching problem both in the line and in the column partitions. A natural question to be addressed is to evaluate if the relative frequency of this phenomenon is higher or not in comparison to the standard single partitioning encountered in the MBC context.

Concerning now the model selection step, some questions remain also open. First of all, the model selection consistency should be fixed from a theoretical point of view, even if numerical experiments in different research papers suggest that it could hold. Again, this question is difficult due to the structure of the latent variable space where latent variables in line and column are interrelated. Then, still due to this block structure, a certain model multiplicity appears in LBM as a combination of the number of cluster line and the number of cluster column candidates. Some heuristic already exist for addressing this issue [Robert, 2021] but an idea could be to extend the idea of a direct estimation of the ICL value [Marbac and Sedki, 2017] in the MBC context, avoiding an intermediate cumbersome parameter estimation. Another idea could be to use a preliminary non-parametric Bayesian step [Goffinet et al., 2021] for filtering candidate LBM models then submitted to the ICL judgment.

Compliance with Ethical Statement

The authors declare that this manuscript is original, has not been published before and is not currently being considered for publication elsewhere. The authors confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome. The authors confirm that the manuscript has been read and approved by all named authors and that there are no other persons who satisfied the criteria for authorship but are not listed. The authors further confirm that the order of authors listed in the manuscript has been approved by all of them. The authors confirm that they have given due consideration to the protection of intellectual property associated with this work and that there are no impediments to publication, including the timing of publication, with respect to intellectual property. In so doing the authors confirm that they have followed the regulations of their institutions concerning intellectual property. The authors understand that the Corresponding Author is the sole contact for the Editorial process (including Editorial Manager and direct communications with the office). He/she is responsible for communicating with the other authors about progress, submissions of revisions and final approval of proofs. The authors confirm that they have provided a current, correct email address which is accessible by the Corresponding Author. Finally, the authors declare that no datasets were generated or analyzed during the current study, since it is a review article.

References

[Abbe, 2017] Abbe, E. (2017). Community detection and stochastic block models: recent developments. *The Journal of Machine Learning Research*, 18(1):6446–6531.

[Ailem et al., 2017] Ailem, M., Role, F., and Nadif, M. (2017). Sparse Poisson latent block model for document clustering. *IEEE Trans. Knowl. Data Eng.*, 29(7):1563–1576.

[Ambroise and Matias, 2012] Ambroise, C. and Matias, C. (2012). New consistent and asymptotically normal parameter estimates for random-graph mixture models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74(1):3–35.

[Banfield and Raftery, 1993] Banfield, J. D. and Raftery, A. E. (1993). Model-based Gaussian and non-Gaussian clustering. *Biometrics*, 49:803–821.

[Baudry, 2015] Baudry, J.-P. (2015). Estimation and model selection for model-based clustering with the conditional classification likelihood. *Electronic Journal of Statistics*, 9(1):1041 – 1077.

[Bellman, 1957] Bellman, R. (1957). *Dynamic Programming*. Princeton University Press, Princeton, NJ, USA, 1st edition.

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

[Bickel et al., 2013] Bickel, P., Choi, D., Chang, X., Zhang, H., et al. (2013). Asymptotic normality of maximum likelihood and its variational approximation for stochastic blockmodels. *The Annals of Statistics*, 41(4):1922–1943.

[Biernacki, 2007] Biernacki, C. (2007). Degeneracy in the maximum likelihood estimation of univariate gaussian mixtures for grouped data and behaviour of the em algorithm. *Scandinavian Journal of Statistics*, 34(3):569–586.

[Biernacki, 2017] Biernacki, C. (2017). Mixture models. In Droesbeke, J.-J., Saporta, G., and Thomas-Agnan, C., editors, *Choix de modèles et agrégation*. Technip.

[Biernacki et al., 2000] Biernacki, C., Celeux, G., and 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.

[Biernacki et al., 2003] Biernacki, C., Celeux, G., and Govaert, G. (2003). Choosing starting values for the EM algorithm for getting the highest likelihood in multivariate gaussian mixture models. *Computational Statistics & Data Analysis*, 41:561–575.

[Biernacki et al., 2011] Biernacki, C., Celeux, G., and Govaert, G. (2011). Exact and Monte Carlo calculations of integrated likelihoods for the latent class model. *Journal of Statistical Planning and Inference*, 140(11):2991–3002.

[Biernacki and Chrétien, 2003] Biernacki, C. and Chrétien, S. (2003). Degeneracy in the maximum likelihood estimation of univariate gaussian mixtures with em. *Statistics & Probability Letters*, 61:373–382.

[Biernacki and Jacques, 2015] Biernacki, C. and Jacques, J. (2015). Model-Based Clustering of Multivariate Ordinal Data Relying on a Stochastic Binary Search Algorithm. *Statistics and Computing*, 26(5):929–943.

[Biernacki and Maugis, 2017] Biernacki, C. and Maugis, C. (2017). High-dimensional clustering. In Droesbeke, J.-J., Saporta, G., and Thomas-Agnan, C., editors, *Choix de modèles et agrégation*. Technip.

[Blei et al., 2003] Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

[Bock, 1979] Bock, H. (1979). Simultaneous clustering of objects and variables. *Analyse des données et Informatique*, pages 187–203.

[Boutalbi et al., 2020] Boutalbi, R., Labiod, L., and Nadif, M. (2020). Tensor latent block model for co-clustering. *International Journal of Data Science and Analytics*, 10:161–175.

[Boutalbi et al., 2022] Boutalbi, R., Labiod, L., and Nadif, M. (2022). Tensorclus: A python library for tensor (co)-clustering. *Neurocomputing*, 468(C):464–468.

[Bouveyron et al., 2018] Bouveyron, C., Bozzi, L., Jacques, J., and 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 and Brunet, 2014] Bouveyron, C. and Brunet, C. (2014). Model-based clustering of high-dimensional data: A review. *Computational Statistics & Data Analysis*, 71:52–78.

[Bouveyron et al., 2019] Bouveyron, C., Celeux, G., Murphy, T. B., and Raftery, A. (2019). *Model-Based Clustering and Classification for Data Science*. Cambridge University Press.

[Bouveyron et al., 2015] Bouveyron, C., Côme, E., and Jacques, J. (2015). The discriminative functional mixture model for a comparative analysis of bike sharing systems. *The Annals of Applied Statistics*, 9(4):1726–1760.

[Bouveyron and Jacques, 2011] Bouveyron, C. and Jacques, J. (2011). Model-based clustering of time series in group-specific functional subspaces. *Advances in Data Analysis and Classification*, 5(4):281–300.

[Bouveyron et al., 2021] Bouveyron, C., Jacques, J., and Schmutz, A. (2021). *funLBM: Model-Based Co-Clustering of Functional Data*. R package version 2.2.

[Bouveyron et al., 2022] Bouveyron, C., Jacques, J., Schmutz, A., Simoes, F., and Bottini, S. (2022). Co-Clustering of Multivariate Functional Data for the Analysis of Air Pollution in the South of France. *Annals of Applied Statistics*, in press.

[Brault, 2014] Brault, V. (2014). *Estimation et sélection de modèle pour le modèle des blocs latents*. PhD thesis, Université Paris Sud.

[Brault et al., 2014] Brault, V., Celeux, G., and Keribin, C. (2014). Mise en œuvre de l'échantillonneur de Gibbs pour le modèle des blocs latents. In *46èmes Journées de Statistique de la SFdS*.

[Brault et al., 2020] Brault, V., Keribin, C., and Mariadassou, M. (2020). Consistency and asymptotic normality of latent block model estimators. *Electronic journal of statistics*, 14(1):1234–1268.

[Brault and Lomet, 2015] Brault, V. and Lomet, A. (2015). Revue des méthodes pour la classification jointe des lignes et des colonnes d'un tableau. *Journal de la Société Française de Statistique*, 156(3):27–51.

[Brault and Mariadassou, 2015] Brault, V. and Mariadassou, M. (2015). Co-clustering through latent bloc model: A review. *Journal de la Société Française de Statistique*, 156(3):120–139.

[Carreira-Perpinán and Renals, 2000] Carreira-Perpinán, M. A. and Renals, S. (2000). Practical identifiability of finite mixtures of multivariate bernoulli distributions. *Neural Computation*, 12(1):141–152.

[Celeux et al., 1996] Celeux, G., Chauveau, D., and Diebolt, J. (1996). Stochastic versions of the em algorithm: an experimental study in the mixture case. *Journal of Statistical Computation and Simulation*, 55(4):287–314.

[Celeux and Diebolt, 1986] Celeux, G. and Diebolt, J. (1986). L'algorithme sem: un algorithme d'apprentissage probabiliste pour la reconnaissance de mélange de densités. *Revue de statistique appliquée*, 34(2):35–52.

[Celeux and Govaert, 1995] Celeux, G. and Govaert, G. (1995). Gaussian parsimonious clustering models. *Pattern Recognition*, 28(5):781–793.

[Celisse et al., 2012] Celisse, A., Daudin, J.-J., and Pierre, L. (2012). Consistency of maximum-likelihood and variational estimators in the stochastic block model. *Electronic Journal of Statistics*, 6:1847–1899.

[Chao et al., 2021] Chao, G., Sun, S., and Bi, J. (2021). A survey on multiview clustering. *IEEE Transactions on Artificial Intelligence*, 2:146–168.

[Charrad et al., 2009] Charrad, M., Lechevallier, Y., Ahmed, M., and Saporta, G. (2009). Block clustering for web pages categorization. In *Intelligent Data Engineering and Automated Learning*, pages 260–267, Burgos. Springer.

[Cheam et al., 2017] Cheam, A. S. M., Marbac, M., and McNicholas, P. D. (2017). Model-based clustering for spatiotemporal data on air quality monitoring. *Environmetrics*, 28(3).

[Chen et al., 2019] Chen, X., Huang, J. Z., Wu, Q., and Yang, M. (2019). Subspace weighting co-clustering of gene expression data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 16(2):352–364.

[Cheng and Liu, 2021] Cheng, H. and Liu, J. (2021). Concurrent brain parcellation and connectivity estimation via co-clustering of resting state fmri data: A novel approach. *Human brain mapping*, 42(8):2477–2489.

[Chi et al., 2020] Chi, E. C., Gaines, B. R., Sun, W. W., Zhou, H., and Yang, J. (2020). Provable convex co-clustering of tensors. *The Journal of Machine Learning Research*, 21(1):1–58.

[Cho and Dhillon, 2008] Cho, H. and Dhillon, I. S. (2008). Coclustering of human cancer microarrays using minimum sum-squared residue coclustering. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 5(3):385–4004.

[Corneli et al., 2020] Corneli, M., Bouveyron, C., and 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*, 29(4):771–785.

[Côme and Jouvin, 2021] Côme, E. and Jouvin, N. (2021). *greed: Clustering and Model Selection with the Integrated Classification Likelihood*. R package version 0.5.1.

[Darikwa et al., 2019] Darikwa, T. B., Manda, S., and Lesaoana, M. (2019). Assessing joint spatial autocorrelations between mortality rates due to cardiovascular conditions in south africa. *Geospatial Health*, 14(2).

[Day, 1969] Day, N. E. (1969). Estimating the components of a mixture of normal distributions. *Biometrika*, 56:463–474.

[De Leeuw and Michailidis, 1999] De Leeuw, J. and Michailidis, G. (1999). Block relaxation algorithms in statistics. *Information systems and data analysis*, pages 308–325.

[Delaigle and Hall, 2010] Delaigle, A. and Hall, P. (2010). Defining probability density for a distribution of random functions. *The Annals of Statistics*, 38:1171–1193.

[Dempster et al., 1977] Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data (with discussion). *Journal of the Royal Statistical Society, Series B*, 39:1–38.

[Dhillon, 2001] 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*, KDD '01, page 269–274, New York, NY, USA. Association for Computing Machinery.

[Dhillon et al., 2003] Dhillon, I. S., Mallela, S., and Modha, D. S. (2003). Information-theoretic co-clustering. In *the ninth ACM SIGKDD international conference on Knowledge discovery and data mining , KDD '03*, pages 89–98.

[Etienne and Latifa, 2014] Etienne, C. and Latifa, O. (2014). Model-based count series clustering for bike sharing system usage mining: a case study with the vélib'system of paris. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 5(3):1–21.

[Flake et al., 2002] Flake, G. W., Lawrence, S., Giles, C. L., and Coetzee, F. M. (2002). Self-organization and identification of web communities. *Computer*, 35(3):66–70.

[Fop and Murphy, 2018] Fop, M. and Murphy, T. B. (2018). Variable selection methods for model-based clustering. *Statistics Surveys*, 12:18 – 65.

[Fop et al., 2017] Fop, M., Smart, K. M., and Murphy, T. B. (2017). Variable selection for latent class analysis with application to low back pain diagnosis. *The Annals of Applied Statistics*, pages 2080–2110.

[Forbes et al., 2019] Forbes, F., Arnaud, A., Lemasson, B., and Barbier, E. (2019). Component elimination strategies to fit mixtures of multiple scale distributions. In *RSSDS 2019 - Research School on Statistics and Data Science*, volume 1150 of *Communications in Computer and Information Science*, pages 81–95, Melbourne, Australia. Springer.

[Frisch et al., 2021a] Frisch, G., Leger, J.-B., and Grandvalet, Y. (2021a). Co-clustering for fair recommendation. In in Computer, C. and Science, I., editors, *Machine Learning and Principles and Practice of Knowledge Discovery in Databases. ECML PKDD 2021.*, volume 1524. Springer, Cham.

[Frisch et al., 2021b] Frisch, G., Leger, J.-B., and Grandvalet, Y. (2021b). SparseBM: A Python Module for Handling Sparse Graphs with Block Models. working paper or preprint.

[Frisch et al., 2022] Frisch, G., Léger, J.-B., and Grandvalet, Y. (2022). Learning from missing data with the latent block model. *Statistics and Computing*, 32(9).

[Gallaugher et al., 2020] Gallaugher, M. P. B., Biernacki, C., and McNicholas, P. D. (2020). Parameter-Wise Co-Clustering for High-Dimensional Data. working paper or preprint.

[George and Merugu, 2005] George, T. and Merugu, S. (2005). A scalable collaborative filtering framework based on co-clustering. In *Proceedings of the Fifth IEEE International Conference on Data Mining*, ICDM '05, page 625–628, USA. IEEE Computer Society.

[George et al., 2021] George, T. B., Strawn, N. K., and Leviyang, S. (2021). Tree-based co-clustering identifies chromatin accessibility patterns associated with hematopoietic lineage structure. *Frontiers in Genetics*, 12.

[Girvan and Newman, 2002] Girvan, M. and Newman, M. E. (2002). Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826.

[Goffinet et al., 2021] Goffinet, E., Lebbah, M., Azzag, H., Loïc, G., and Coutant, A. (2021). Non-parametric multivariate time series co-clustering model applied to driving-assistance systems validation. In Lemaire, V., Malinowski, S., Bagnall, A., Guyet, T., Tavenard, R., and Ifrim, G., editors, *Advanced Analytics and Learning on Temporal Data*, pages 71–87, Cham. Springer International Publishing.

[Good, 1965] Good, I. J. (1965). Categorization of classification. *Mathematics and Computer Science in Biology and Medicine*, pages 115–125. Her Majesty's Stationery Office, London.

[Goodman, 1974] Goodman, L. A. (1974). Exploratory latent structure models using both identifiable and unidentifiable models. *Biometrika*, 61:215–231.

[Govaert, 1983] Govaert, G. (1983). *Classification croisée*. PhD thesis, Thèse d'état, Université Paris 6.

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

[Govaert and Nadif, 2013] Govaert, G. and Nadif, M. (2013). *Co-Clustering*. Wiley.

[Hasan et al., 2018] Hasan, M. N., Rana, M. M., Begum, A. A., Rahman, M., and Mollah, M. N. H. (2018). Robust co-clustering to discover toxicogenomic biomarkers and their regulatory doses of chemical compounds using logistic probabilistic hidden variable model. *Frontiers in Genetics*, 9.

[Huang et al., 2020] Huang, S., Xu, Z., Tsang, I. W., and Kang, Z. (2020). Auto-weighted multi-view co-clustering with bipartite graphs. *Information Sciences*, 512:18–30.

[Ingrassia and Rocci, 2007] Ingrassia, S. and Rocci, R. (2007). Constrained monotone em algorithms for finite mixture of multivariate gaussians. *Computational Statistics & Data Analysis*, 51(11):5339–5351.

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

[Jacques and Preda, 2013] Jacques, J. and Preda, C. (2013). Funclust: a curves clustering method using functional random variable density approximation. *Neurocomputing*, 112:164–171.

[Jain et al., 1999] Jain, A. K., Murty, M. N., and Flynn, P. J. (1999). Data clustering: a review. *ACM Comput. Surv.*, 31(3):264–323.

[Jin et al., 2016] Jin, C., Zhang, Y., Balakrishnan, S., Wainwright, M., and Jordan, M. (2016). Local maxima in the likelihood of gaussian mixture models: Structural results and algorithmic consequences. In *Thirtieth Conference on Neural Information Processing Systems, NeurIPS 2016*.

[Kaufman and Rousseeuw, 1990] Kaufman, L. and Rousseeuw, P. J. (1990). *Finding groups in data: An introduction to cluster analysis*. Wiley.

[Keribin, 2021] Keribin, C. (2021). Cluster or co-cluster the nodes of oriented graphs? *Journal de la Société Française de Statistique*, 162(1):46–69.

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

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

[Keuper et al., 2020] Keuper, M., Tang, S., Andres, B., Brox, T., and Schiele, B. (2020). Motion segmentation amp; multiple object tracking by correlation co-clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(1):140–153.

[Laclau and Nadif, 2016] Laclau, C. and Nadif, M. (2016). Hard and fuzzy diagonal co-clustering for document-term partitioning. *Neurocomputing*, 193(C):133–147.

[Leger, 2016] Leger, J.-B. (2016). Blockmodels: A r-package for estimating in latent block model and stochastic block model, with various probability functions, with or without covariates.

[Leger et al., 2020] Leger, J.-B., Barbillon, P., and Chiquet, J. (2020). *blockmodels: Latent and Stochastic Block Model Estimation by a 'V-EM' Algorithm*. R package version 1.1.4.

[Li, 2020] Li, G. (2020). Generalized co-clustering analysis via regularized alternating least squares. *Computational Statistics & Data Analysis*, 150:106989.

[Lian et al., 2019] Lian, C., Ruan, S., Denoeux, T., Li, H., and Vera, P. (2019). Joint tumor segmentation in PET-CT images using co-clustering and fusion based on belief functions. *IEEE transactions on image processing*, 28(2):755–766.

[Lomet et al., 2012a] Lomet, A., Govaert, G., and Grandvalet, Y. (2012a). Design of artificial data tables for co-clustering analysis. Technical report, Université de Technologie de Compiègne, France.

[Lomet et al., 2012b] Lomet, A., Govaert, G., and Grandvalet, Y. (2012b). Model selection in block clustering by the integrated classification likelihood. In *20th International Conference on Computational Statistics (COMPSTAT 2012)*, pages 519–530, Lymassol, France.

[MacQueen, 1967] MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In LeCam, L. M. and Neyman, J., editors, *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, University of California Press.

[Madeira and Oliveira, 2004] Madeira, S. C. and Oliveira, A. L. (2004). Biclustering algorithms for biological data analysis : a survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, pages 24–45.

[Malsiner-Walli et al., 2016] Malsiner-Walli, G., Frühwirth-Schnatter, S., and Grün, B. (2016). Model-based clustering based on sparse finite gaussian mixtures. *Statistics and Computing*, 26:303–324.

[Marbac and Sedki, 2017] Marbac, M. and Sedki, M. (2017). Variable selection for model-based clustering using the integrated complete-data likelihood. *Statitics and Computing*, 27:1049–1063.

[Marchello et al., 2022] Marchello, G., Fresse, A., Corneli, M., and Bouveyron, C. (2022). Co-clustering of evolving count matrices with the dynamic latent block model: application to pharmacovigilance. *Statistics and Computing*, 32(3):1–22.

[Mariadassou and Matias, 2015] Mariadassou, M. and Matias, C. (2015). Convergence of the groups posterior distribution in latent or stochastic block models. *Bernoulli*, 21(1):537–573.

[Matias and Robin, 2014] Matias, C. and Robin, S. (2014). Modeling heterogeneity in random graphs through latent space models: a selective review. *ESAIM: Proceedings and Surveys*, 47:55–74.

[Maugis et al., 2009] Maugis, C., Celeux, G., and Martin-Magniette, M.-L. (2009). Variable selection in model-based clustering: A general variable role modeling. *Computational Statistics & Data Analysis*, 53(11):3872–3882.

[McLachlan and Peel, 2000] McLachlan, G. and Peel, D. (2000). *Finite Mixture Models*. Wiley, New-York.

[McLachlan and Krishnam, 1997] McLachlan, G. J. and Krishnam, T. (1997). *The EM algorithm and extensions*. Wiley, New York.

[McParland and Gormley, 2013] McParland, D. and Gormley, C. (2013). *Algorithms from and for Nature and Life: Studies in Classification, Data Analysis, and Knowledge Organization*, chapter Clustering Ordinal Data via Latent Variable Models, pages 127–135. Springer, Switzerland.

[Rand, 1971] Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of American Statistical Association*, 66:846–850.

[Redner and Walker, 1984] Redner, R. and Walker, H. (1984). Mixture densities, maximum likelihood and the EM algorithm. *SIAM Review*, 26(2):195–239.

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

[Robert, 2021] Robert, V. (2021). *bikm1: Co-Clustering Adjusted Rand Index and Bikm1 Procedure for Contingency and Binary Data-Sets*. R package version 1.1.0.

[Robert et al., 2015] Robert, V., Celeux, G., and Keribin, C. (2015). Un modèle statistique pour la pharmacovigilance. In *47èmes Journées de Statistique de la SFdS*.

[Robert et al., 2021] Robert, V., Vasseur, Y., and Brault, V. (2021). Comparing high-dimensional partitions with the co-clustering adjusted rand index. *Journal of Classification*, 38(1):158–186.

[Rohe et al., 2011] Rohe, K., Chatterjee, S., and Yu, B. (2011). Spectral clustering and the high-dimensional stochastic blockmodel. *The Annals of Statistics*, 39(4):1878–1915.

[Sedki et al., 2014] Sedki, M., Celeux, G., and Maugis-Rabusseau, C. (2014). SelvarMix: A R package for variable selection in model-based clustering and discriminant analysis with a regularization approach. Research report, Inria.

[Selosse et al., 2019a] Selosse, M., Gourru, A., Jacques, J., and Velcin, J. (2019a). Tri-clustering pour données de comptage. In *51èmes Journées de Statistique de la SFdS*.

[Selosse et al., 2020a] Selosse, M., Jacques, J., and Biernacki, C. (2020a). Model-based co-clustering for mixed type data. *Computational Statistics & Data Analysis*, 144:106866.

[Selosse et al., 2020b] Selosse, M., Jacques, J., and Biernacki, C. (2020b). *ordinalClust: Ordinal Data Clustering, Co-Clustering and Classification*. R package version 1.3.5.

[Selosse et al., 2020c] Selosse, M., Jacques, J., and Biernacki, C. (2020c). Textual data summarization using the self-organized co-clustering model. *Pattern Recognition*, 103:107315.

[Selosse et al., 2021] Selosse, M., Jacques, J., and Biernacki, C. (2021). *mixedClust: Co-Clustering of Mixed Type Data*. R package version 1.0.2.

[Selosse et al., 2019b] Selosse, M., Jacques, J., Biernacki, C., and Cousson-Gélie, F. (2019b). Analyzing health quality survey using constrained co-clustering model for ordinal data and some dynamic implication. *Journal of the Royal Statistical Society: Series C Applied Statistics*, 68(5):1327–1349.

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

[Sportisse et al., 2021] Sportisse, A., Marbac, M., Biernacki, C., Boyer, C., Celeux, G., Laporte, F., and Josse, J. (2021). Model-based clustering with missing not at random data.

[Stephens, 2000] Stephens, M. (2000). Dealing with label switching in mixture models. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 62(4):795–809.

[Tokuda et al., 2017] Tokuda, T., Yoshimoto, J., Shimizu, Y., Okada, G., Takamura, M., Okamoto, Y., Yamawaki, S., and Doya, K. (2017). Multiple co-clustering based on non-parametric mixture models with heterogeneous marginal distributions. *PLoS ONE*, 12.

[Ullah et al., 2017] Ullah, S., Daud, H., Dass, S. C., Khan, H. N., and Khalil, A. (2017). Detecting space-time disease clusters with arbitrary shapes and sizes using a co-clustering approach. *Geospatial Health*, 12(2).

[Vandewalle et al., 2020] Vandewalle, V., Preda, C., and Dabo-Niang, S. (2020). Clustering spatial functional data. In Mateu, J. and Giraldo, R., editors, *Geostatistical Functional Data Analysis : Theory and Methods*. John Wiley and Sons, Chichester, UK.

[Vermunt and Magidson, 2005] Vermunt, J. and Magidson, J. (2005). *Technical Guide for Latent GOLD 4.0: Basic and Advanced*. Statistical Innovations Inc., Belmont, Massachusetts.

[Wang et al., 2018] Wang, X., Yu, G., Domeniconi, C., Wang, J., Yu, Z., and Zhang, Z. (2018). Multiple co-clusterings. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 1308–1313.

[Wang and Bickel, 2017] Wang, Y. R. and Bickel, P. J. (2017). Likelihood-based model selection for stochastic block models. *The Annals of Statistics*, 45(2):500–528.

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

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

[Xu and jie Tian, 2015] Xu, D. and jie Tian, Y. (2015). A comprehensive survey of clustering algorithms. *Annals of Data Science*, 2:165–193.

[Xu et al., 2010] Xu, G., Zong, Y., Dolog, P., and Zhang, Y. (2010). Co-clustering analysis of weblogs using bipartite spectral projection approach. In *Knowledge-Based and Intelligent Information and Engineering Systems*, pages 398–407, Cardiff. Springer.

[Zeng et al., 2020] Zeng, P., Wangwu, J., and Lin, Z. (2020). Coupled co-clustering-based unsupervised transfer learning for the integrative analysis of single-cell genomic data. *Briefings in Bioinformatics*, 22(4). bbaa347.