Stat Comput
DOI 10.1007/s11222-016-9677-7

# Diagonal latent block model for binary data

**Charlotte Laclau**[1] · **Mohamed Nadif**[1]

Received: 7 December 2015 / Accepted: 11 June 2016


**Abstract** This paper addresses the problem of co-clustering binary data in the latent block model framework with diagonal constraints for resulting data partitions. We consider the Bernoulli generative mixture model and present three new methods differing in the assumptions made about the degree of homogeneity of diagonal blocks. The proposed models are parsimonious and allow to take into account the structure of a data matrix when reorganizing it into homogeneous diagonal blocks. We derive algorithms for each of the presented models based on the classification expectation-maximization algorithm which maximizes the complete data likelihood. We show that our contribution can outperform other state-of-the-art (co)-clustering methods on synthetic sparse and non-sparse data. We also prove the efficiency of our approach in the context of document clustering, by using real-world benchmark data sets.

**Keywords** Co-clustering · Latent block model · Binary data · Document clustering

## 1 Introduction

Co-clustering, also known as biclustering or block-clustering, involves simultaneous clustering of a set of observations and a set of features in a data matrix. By creating permutations of rows and columns, co-clustering algorithms aim to reorganize the initial data matrix into homogeneous blocks. These blocks, also called co-clusters, can therefore be seen as subsets of the data matrix characterized by a set of observations and a set of features whose elements are similar (Govaert 1983; Vichi 2001; Madeira and Oliveira 2004; Van Mechelen et al 2004; Bock 2003). Co-clustering algorithms offer several advantages over simple clustering algorithms: for instance, they reduce the initial matrix into a simpler form with the same basic structure and require far less computation when compared with applying a clustering algorithm separately on both modes of a data set. Applications of co-clustering include but not limited to recommendation systems (George 2005; Hofmann and Puzicha 1999), gene expression analysis (Cheng and Church 2000) and text mining (Dhillon 2001; Dhillon et al 2003). As a result, these methods are of an increasing interest to the data mining community. According to the recent survey on co-clustering (Govaert and Nadif 2013), we can distinguish between metric based and probabilistic based co-clustering methods. The former ones consist in defining a clustering criterion and then, in finding an algorithm optimizing this criterion (see for instance; Govaert (1995), Cho et al (2004); Banerjee et al (2007)). These methods are usually solved using heuristic techniques because of their high optimization complexity. Probabilistic approaches (Si and Jin 2003; Wyse and Friel 2012; Keribin et al 2015) use the framework of generative mixture models. A particular case of them is the latent block model (LBM) described in Govaert and Nadif (2003). The LBM relies on the intuitive idea that a population is composed of several blocks generated by different parametrized probability density functions (pdf). This model became quite popular for the following reasons: (1) imposing various constraints on the parameters that define the model allows to handle a wide variety of different data structures; (2) they are strongly linked to metric based methods while not suffering from the same computational issues; (3) using an appropriate probability distribution allows to deal with continuous as

✉ Charlotte Laclau
charlotte.laclau@parisdescartes.fr

[1] LIPADE, Paris Descartes University, 45, rue des Saints Pères, 75270 Paris, France

Published online: 29 June 2016

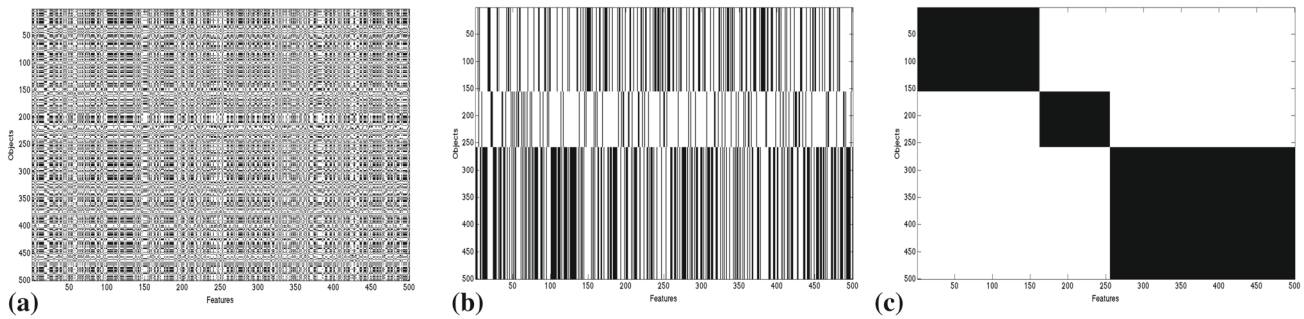



**Fig. 1** **a** Original binary data, **b** data reorganized according to rows, and **c** data reorganized according to rows and columns

well as co-occurrence and binary data sets (see for instance; Govaert and Nadif 2013).

One of the constraints that can be incorporated into a co-clustering method is to seek a block diagonal structure, i.e., the number of clusters of observations is equal to the number of clusters of features. An illustration of this idea is given in Fig. 1 where (a) represents an original binary matrix, (b) represents the same matrix after a proper permutation of rows whilst (c) adds a permutation of columns resulting in a clear block diagonal structure. Imposing the diagonal structure on the resulting co-clustering can be motivated based on two different arguments. First, if one considers that all row clusters are described by non-overlapping subsets of features then these latter are forced to be orthogonal among themselves. In this case, each row cluster would be spanned by a set of vectors that are as different as possible from any other subset of features in other clusters thus maximizing the inter-cluster distance. This idea is also linked to the learning on Stiefel manifolds and, for instance, imposing orthogonality constraints on both row and column partitions in Bi-Orthogonal NMF model (Ding et al 2006). On the other hand, diagonal structure is assumed to be sparsity inducing as it penalizes $\ell_0$ norm of each row. While this may be too restrictive on data sets with small number of features, for high-dimensional data it presents clear benefits and can be seen as a matrix regularization constraint. This idea is also widely studied in generalized blockmodeling (Batagelj et al 1998; Doreian et al 2005) and seriation methods (Garcia and Proth 1986; Marcotorchino 1987) and has found its application in, for example, group technology, botany and social network analysis. When the data set is typically represented by a sparse high-dimensional $document \times term$ matrix, these methods have proven again to be efficient when dealing with the problem of clustering or co-clustering. Their objective is to group documents based on words within them and to group words based on documents in which they appear. As we will see in this paper, seeking a diagonal structure on $document \times term$ matrices improves clustering results. Among the co-clustering methods, the processing of binary data is probably the less investigated one, although we can cite, for instance, the work of Girolami (2001) and Kabán and Bingham (2008). In this paper, we introduce a diagonal latent block model for co-clustering binary data. We use a parsimonious Bernoulli distribution as a parametrized pdf where the block parameter is split into a center and a dispersion parameters. The idea of this splitting scheme was first introduced in Govaert and Nadif (2003). While this model shows good results for highly dimensional data, we will show that it is less efficient for highly sparse data. In order to overcome the issues of the original method and to ensure the diagonal structure, we set the center parameter for diagonal elements to 1 and consider three different scenarios for the dispersion parameter. These scenarios explicitly cover the following cases: (1) first model puts no constraints on the degree of homogeneity of blocks; (2) second model assumes that object blocks share the same degree of homogeneity; (3) third model imposes the same degree of homogeneity for all blocks. In the following sections, we introduce an extensive comparison of the proposed models and show for what types of data they are most suitable for. Furthermore, we show that co-clustering algorithms in general give superior results when compared to clustering methods when applied on high-dimensional data. To this end, our results agree well with a common belief that in a high-dimensional feature space instances can be more accurately represented only based on a small subset of relevant features. This idea also lies at the foundation of such approaches as dimensionality reduction, feature selection and sparse coding.

To this end, our work is related to the following papers: Li (2005), Dhillon (2001)and Labiod and Nadif (2011). Li (2005), the author proposed a block diagonal algorithm applied to binary data. This algorithm alternates the clustering of observations and features minimizing the error between the original data matrix and the reconstructed matrix based on the cluster structure. Dhillon (2001) a spectral algorithm has been proposed; it consists in building a bipartite graph from the $document \times term$ matrix which is partitioned to minimize the cut objective function. Another attempt to use graph related criteria for block diagonal clustering was presented in Labiod and Nadif (2011). The proposed spec-

tral co-clustering algorithm maximizes a generalization of the modularity that is further casted as a trace maximization problem. However, all these methods are metric-based and, thus, they suffer from the restrictions described above.

The remainder of this paper is organized as follows. Section 2 provides the needed background on the LBM and defines the parsimonious Bernoulli model. In Sect. 3, we present the diagonal latent block model for binary data and derive three algorithms. Section 4 is devoted to numerical experiments on synthetic data sets to assess and compare algorithms on binary data. We compare our model with state-of-the-art (co)-clustering algorithms on real *document* × *term* data sets showing the appropriateness of our contribution in Sect. 5. Finally we summarize the study and give possible research perspectives in the last section.

## 2 Preliminary knowledge

In this section, we first introduce notations that are used in this paper, then we proceed by presenting the latent block model and an expectation-maximization (EM) type algorithm used to estimate its parameters.

### 2.1 Notations

The notations used in this paper are the following:

- Data is denoted by a $n$ by $d$ matrix $\mathbf{x} = \{x_{ij}, i \in I = \{1, \ldots, n\}; j \in J = \{1, \ldots, d\}\}$.
- A partition of $I$ into $g$ clusters is represented by the classification matrix $\mathbf{z} = (z_{ik}, i = 1, \ldots, n, k = 1, \ldots, g)$ where $z_{ik} = 1$ if element $i$ belongs to cluster $k$ and $z_{ik} = 0$ otherwise. In a similar way we define the partition of $J$ into $m$ clusters by $\mathbf{w} = (w_{j\ell}, j = 1, \ldots, d, \ell = 1, \ldots, m)$.
- Sums and products related to rows, columns, row's cluster and column's cluster are subscripted by the letters $i$, $j, k$ and $\ell$ without indicating the limits of variation which will be implicit. So, the sums $\sum_i, \sum_j, \sum_k$ and $\sum_\ell$ stand for $\sum_{i=1}^n, \sum_{j=1}^d, \sum_{k=1}^g$, and $\sum_{\ell=1}^m$, respectively.

We also use intermediate forms of the original data matrix $\mathbf{x}$. These matrices are defined in Table 1.

**Table 1** Reduced matrices, sizes and definitions of $\mathbf{x}^{\mathbf{z}}$, $\mathbf{x}^{\mathbf{w}}$ and $\mathbf{x}^{\mathbf{zw}}$

| Matrix | Size | Definition |
|---|---|---|
| $\mathbf{x}^{\mathbf{z}} = (x^{\mathbf{z}}_{kj})$ | $(g \times d)$ | $x^{\mathbf{z}}_{kj} = \sum_i z_{ik} x_{ij}$ |
| $\mathbf{x}^{\mathbf{w}} = (x^{\mathbf{w}}_{i\ell})$ | $(n \times m)$ | $x^{\mathbf{w}}_{i\ell} = \sum_j w_{j\ell} x_{ij}$ |
| $\mathbf{x}^{\mathbf{zw}} = (x^{\mathbf{zw}}_{k\ell})$ | $(g \times m)$ | $x^{\mathbf{zw}}_{k\ell} = \sum_{i,j} z_{ik} w_{j\ell} x_{ij}$ |

### 2.2 Definition of the latent block model

To embed the co-clustering into a probabilistic framework, in Govaert and Nadif (2003) the authors proposed the latent block model, called LBM. Given a data matrix $\mathbf{x} \in \mathbb{R}^{n \times d}$, the latent block model assumes that the univariate random variables $x_{ij}$ are conditionally independent knowing $\mathbf{z}$ and $\mathbf{w}$ with parametrised pdf $f(x_{ij}; \alpha_{k\ell})$ if the row $i$ belongs to the cluster $k$ and the column $j$ belongs to the cluster $\ell$. The conditional pdf of $\mathbf{x}$ knowing $\mathbf{z}$ and $\mathbf{w}$ can be expressed as

$$\prod_{i,j} f(x_{ij}; \alpha_{z_i w_j}) = \prod_{i,j,k,\ell} \{f(x_{ij}; \alpha_{k\ell})\}^{z_{ik} w_{j\ell}}. \tag{1}$$

In this case, the two sets $I$ and $J$ are assumed to be random samples so that the row and column labels become latent variables. This model is based on the following assumptions:

- Conditional independence defined before;
- Independent latent variables: the partitions $\mathbf{z}_1, \ldots, \mathbf{z}_n$, $\mathbf{w}_1, \ldots, \mathbf{w}_d$ are considered as latent variables and assumed to be independent:

$$p(\mathbf{z}, \mathbf{w}) = p(\mathbf{z}) p(\mathbf{w}), \tag{2}$$

$$p(\mathbf{z}) = \prod_i p(z_i) \quad \text{and} \quad p(\mathbf{w}) = \prod_j p(w_j), \tag{3}$$

  where $p(\mathbf{z})$ and $p(\mathbf{w})$ are the distributions of rows' and columns' labels, respectively.
- For all $i$, the distribution of $p(z_i)$ is the multinomial distribution $\mathcal{M}(\pi_1, \ldots, \pi_g)$ and does not depend on $i$. Similarly, for all $j$, the distribution of $p(w_j)$ is the multinomial distribution $\mathcal{M}(\rho_1, \ldots, \rho_m)$ and does not depend on $j$.

The parameter of the latent block model is given by $\boldsymbol{\theta} = (\boldsymbol{\pi}, \boldsymbol{\rho}, \boldsymbol{\alpha})$, where $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_g)$ and $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_m)$ represent the mixing proportions and $\alpha_{k\ell}$ is the parameter of the distribution for the block $(k, \ell)$. Denoting by $\mathcal{Z}$ and $\mathcal{W}$ the sets of possible labels $\mathbf{z}$ for $I$ and $\mathbf{w}$ for $J$, the pdf of $\mathbf{x}$ can be written

$$\begin{aligned} f(\mathbf{x}; \boldsymbol{\theta}) &= \sum_{(z,w) \in \mathcal{Z} \times \mathcal{W}} p(\mathbf{z}, \mathbf{w}) f(\mathbf{x} | \mathbf{z}, \mathbf{w}; \boldsymbol{\theta}) \\ &= \sum_{(z,w) \in \mathcal{Z} \times \mathcal{W}} \prod_i \pi_{z_i} \prod_j \rho_{w_j} \prod_{i,j} f\left(x_{ij}; \alpha_{z_i w_j}\right) \\ &= \sum_{(z,w) \in \mathcal{Z} \times \mathcal{W}} \prod_{i,k} \pi_k^{z_{ik}} \prod_{j\ell} \rho_\ell^{w_{j\ell}} \\ &\quad \times \prod_{i,j,k,\ell} \left\{f(x_{ij}; \alpha_{k\ell})\right\}^{z_{ik} w_{j\ell}}. \end{aligned} \tag{4}$$

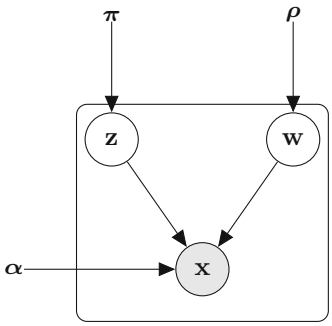



**Fig. 2** Latent block model as a graphical model

This model can be represented by a graphical model depicted in Fig. 2. For the latent block model, the complete data are defined by the vector $(\mathbf{x}, \mathbf{z}, \mathbf{w})$ and its log-likelihood can be written as follows

$$\begin{aligned} \mathrm{L_C}(\mathbf{z}, \mathbf{w}, \boldsymbol{\theta}) &= \mathrm{L}(\boldsymbol{\theta}; \mathbf{x}, \mathbf{z}, \mathbf{w}) \\ &= \log\{p(\mathbf{z}; \boldsymbol{\theta}) p(\mathbf{w}; \boldsymbol{\theta}) f(\mathbf{x}|\mathbf{z}, \mathbf{w}; \boldsymbol{\alpha})\} \\ &= \log p(\mathbf{z}; \boldsymbol{\theta}) + \log p(\mathbf{w}; \boldsymbol{\theta}) + \log f(\mathbf{x}|\mathbf{z}, \mathbf{w}; \boldsymbol{\alpha}) \\ &= \log \prod_{i,k} \pi_k^{z_{ik}} + \log \prod_{j,\ell} \rho_\ell^{w_{j\ell}} + \log f(\mathbf{x}|\mathbf{z}, \mathbf{w}; \boldsymbol{\alpha}). \end{aligned}$$

Finally, the complete-data log-likelihood becomes

$$\begin{aligned} \mathrm{L_C}(\mathbf{z}, \mathbf{w}, \boldsymbol{\theta}) = &\sum_{i,k} z_{ik} \log \pi_k + \sum_{j,\ell} w_{j\ell} \log \rho_\ell \\ &+ \sum_{i,j,k,\ell} z_{ik} w_{j\ell} \log f(x_{ij}; \alpha_{k\ell}). \end{aligned} \tag{5}$$

Depending on the type of data, different probability distributions were considered to deal with binary (Govaert and Nadif 2007, 2008), contingency (Govaert and Nadif 2010) and continuous (Govaert and Nadif 2013) data.

### 2.3 Parsimonious Bernoulli models

In order to deal with binary data, we assume that values of $x_{ij}$ are distributed according to a Bernoulli distribution $\mathcal{B}(\alpha_{k\ell})$ ($\alpha_{k\ell} \in \mathbb{R}$ and $0 < \alpha_{k\ell} < 1$) and the pdf is defined by

$$f(x_{ij}; \alpha_{k\ell}) = (\alpha_{k\ell})^{x_{ij}} (1 - \alpha_{k\ell})^{(1-x_{ij})}.$$

From this formulation, a parsimonious model can be defined by splitting the block parameter $\alpha_{k\ell}$ into a "center" parameter and a "dispersion" parameter and by imposing constraints on the latter. More precisely, each parameter $\alpha_{k\ell}$ is replaced by $a_{k\ell} \in \{0, 1\}$ and $\varepsilon_{k\ell} \in [0, \frac{1}{2}]$ with

$$\begin{cases} a_{k\ell} = 1 & \text{and} \quad \varepsilon_{k\ell} = 1 - \alpha_{k\ell} \text{ if } \alpha_{k\ell} \in [\frac{1}{2}, 1], \\ a_{k\ell} = 0 & \text{and} \quad \varepsilon_{k\ell} = \alpha_{k\ell} \text{ if } \alpha_{k\ell} \in [0, \frac{1}{2}[. \end{cases}$$

Thus, the Bernoulli pdf can be written as

$$f(x_{ij}; (a_{k\ell}, \varepsilon_{k\ell})) = (\varepsilon_{k\ell})^{|x_{ij} - a_{k\ell}|} (1 - \varepsilon_{k\ell})^{1 - |x_{ij} - a_{k\ell}|},$$

where

- $a_{k\ell} \in \{0, 1\}$ is the most frequent binary value of the block, representing the center of the block and
- $\varepsilon_{k\ell} \in [0, 1/2]$ is the probability of any particular variable having a value different from that of the center of the block representing the dispersion, or degree of heterogeneity, of the block. Therefore, a high value of this parameter means a high degree of heterogeneity for a given block.

From this formulation of the pdf, we can impose several constraints on the parameters $a_{k\ell}$ and $\varepsilon_{k\ell}$ and therefore define parsimonious models.

### 2.4 Classification maximum likelihood approach

Considering the *Classification Maximum Likelihood* (CML) approach (Symons 1981), the optimization of the complete-data log-likelihood defined in Eq. (5) can be achieved through an approximation of the *Latent Block* EM algorithm (Govaert and Nadif 2005) referred as the *Latent Block* Classification EM (LBCEM) (Govaert and Nadif 2008). In the sequel, we focus on LBCEM that relies on the complete data $(\mathbf{x}, \mathbf{z}, \mathbf{w})$. It maximizes the complete data log-likelihood by alternating two conditional optimizations of $\mathrm{L_C}(\mathbf{z}, \boldsymbol{\theta}|\mathbf{w})$ for the partition of instances and $\mathrm{L_C}(\mathbf{w}, \boldsymbol{\theta}|\mathbf{z})$ for the partition of columns. These maximizations rely on the following three principal steps:

1. Maximization of $\mathrm{L_C}(\mathbf{z}, \mathbf{w}, \boldsymbol{\theta})$ w.r.t. $\mathbf{z}$ for fixed $\boldsymbol{\theta}$ and $\mathbf{w}$.
2. Maximization of $\mathrm{L_C}(\mathbf{z}, \mathbf{w}, \boldsymbol{\theta})$ w.r.t. $\mathbf{w}$ for fixed $\boldsymbol{\theta}$ and $\mathbf{z}$.
3. Maximization of $\mathrm{L_C}(\mathbf{z}, \mathbf{w}, \boldsymbol{\theta})$ w.r.t. $\boldsymbol{\theta}$ for fixed $\mathbf{z}$ and $\mathbf{w}$.

The first two steps correspond to the Estimation and Classification Steps of the CEM algorithm (Celeux and Govaert 1992). They allow us to construct a pair of partition $(\mathbf{z}, \mathbf{w})$ before the Maximization Step. In Algorithm 1 we describe the different steps of LBCEM.

## 3 Diagonal Bernoulli LBM

In this section, we propose to derive a diagonal version of the latent block model based on a parsimonious Bernoulli distribution. The Diagonal Bernoulli Latent Block Model (DBLBM) aims at dealing with data that is assumed to have a strong latent diagonal structure composed of homogeneous

**Algorithm 1** LBCEM

**input:** $\mathbf{x}$, $g$, $m$
**initialization:** $\mathbf{z}$, $\mathbf{w}$
**Compute:** $\hat{\pi}_k = \frac{\sum_i z_{ik}}{n}$, $\hat{\rho}_\ell = \frac{\sum_j w_{j\ell}}{d}$, and
$\hat{\alpha}_{k\ell} = \arg\max_{k,\ell} \sum_{i,j} z_{ik} w_{j\ell} \log f(x_{ij}, \alpha_{k\ell})$
**repeat**
  **repeat**
    **step 1.** $z_i = \arg\max_k (\sum_{j,\ell} w_{j\ell} \log f(x_{ij}, \alpha_{k\ell}) + \log \pi_k)$
    **step 2.** Compute $\hat{\pi}_k = \frac{\sum_i z_{ik}}{n}$, and
    $\hat{\alpha}_{k\ell} = \arg\max_{k,\ell} \sum_{i,j} z_{ik} w_{j\ell} \log f(x_{ij}, \alpha_{k\ell})$
  **until** convergence
  **repeat**
    **step 3.** $w_j = \arg\max_\ell (\sum_{i,k} z_{ik} \log f(x_{ij}, \alpha_{k\ell} + \log \rho_\ell)$
    **step 4.** Compute $\hat{\rho}_\ell = \frac{\sum_j w_{j\ell}}{d}$ and
    $\hat{\alpha}_{k\ell} = \arg\max_{k,\ell} \sum_{i,j} z_{ik} w_{j\ell} \log f(x_{ij}, \alpha_{k\ell})$
  **until** convergence
**until** convergence
**return** $\mathbf{z}$, $\mathbf{w}$, $\alpha_{k\ell}$, $\pi_k$ and $\rho_\ell$

blocks. This type of data occurs in many real-world applications like $document \times term$ partitioning, network analysis or any other application that implies a symmetry between the set of objects and the set of features.

### 3.1 Definition of the model

The Diagonal Bernoulli LBM aims to reorganize the initial data matrix in a way that the resulting partitions are composed of diagonal blocks of 1's. To achieve this, we set $a_{kk} = 1$ $\forall k = 1, \ldots, g$ and $a_{k\ell} = 0$ $\forall \ell \neq k$. Thus, the Bernoulli pdf takes the two following forms

$$f(x_{ij}; \varepsilon_{kk} | a_{kk} = 1) = (\varepsilon_{kk})^{|x_{ij}-1|}(1-\varepsilon_{kk})^{1-|x_{ij}-1|},$$

and

$$f(x_{ij}; \varepsilon_{k\ell} | a_{k\ell} = 0) = (\varepsilon_{k\ell})^{x_{ij}}(1-\varepsilon_{k\ell})^{1-x_{ij}}.$$

Therefore, $\boldsymbol{\theta} = (\boldsymbol{\pi}, \boldsymbol{\rho}, \boldsymbol{\alpha})$ denotes the parameters of the model with $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_g)$, $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_g)$ and $\boldsymbol{\epsilon} = (\varepsilon_{11}, \ldots, \varepsilon_{k\ell})$. In this case, we have $\varepsilon_{k\ell} \in [0, \frac{1}{2}]$ that denotes the degree of heterogeneity of the block $(k, \ell)$. The probability density associated with this parsimonious Bernoulli distribution is given by:

$$\sum_{(\mathbf{z},\mathbf{w}) \in \mathcal{Z} \times \mathcal{W}} \prod_{i,k} \pi_k^{z_{ik}} \prod_{j,\ell} \rho_\ell^{w_{j\ell}} \prod_{i,j,k} \left(f(x_{ij}; \varepsilon_{kk} | a_{kk} = 1)\right)^{z_{ik} w_{jk}} \times \prod_{i,j,k,\ell \neq k} \left(f(x_{ij}; \varepsilon_{k\ell} | a_{k\ell} = 0)\right)^{z_{ik} w_{j\ell}}.$$

We note that unlike the Bernoulli LBM presented in the previous section, we do not have to estimate the $a_{k\ell}$ parameters but we set them beforehand to a predefined value.

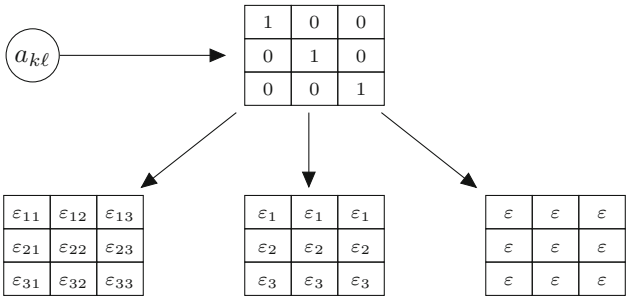



**Fig. 3** Values of the $a_{k\ell}$ and $\varepsilon_{k\ell}$ depending on hypothesis on the degree of homogeneity for the three models where $g = 3$

In what follows, we propose three new diagonal parsimonious models by imposing certain constraints on $\varepsilon_{k\ell}$. The first model assumes for each block $(k, \ell)$ a different degree of homogeneity. For the second model the assumption used is that the diagonal blocks have different degree of homogeneity w.r.t. rows. Finally, the third model hypothesizes that all blocks share the same global degree of dispersion and that the row's and column's clusters sizes are equal. A summary of parametrization schemes of these three models is given in Fig. 3 for a matrix with 3 blocks. In the sequel, models are referred to as model $\mathcal{M}1$, model $\mathcal{M}2$ and model $\mathcal{M}3$.

### 3.2 Model $\mathcal{M}1$

When we consider different degrees of homogeneity for each block, the complete-data log-likelihood takes the following form:

$$\begin{aligned} \mathrm{L_C}(\mathbf{z}, \mathbf{w}, \boldsymbol{\theta}) = &\sum_{i,k} z_{ik} \log \pi_k + \sum_{j,\ell} w_{j\ell} \log \rho_\ell \\ &+ \sum_{i,j,k} z_{ik} w_{jk} \log f(x_{ij}; (a_{kk} = 1, \varepsilon_{kk})) \\ &+ \sum_{i,j,k,\ell \neq k} z_{ik} w_{j\ell} \log f(x_{ij}; (a_{k\ell} = 0, \varepsilon_{k\ell})). \end{aligned}$$

As we consider the Bernoulli distribution, we obtain:

$$\begin{aligned} \mathrm{L_C}(\mathbf{z}, \mathbf{w}, \boldsymbol{\theta}) = &\sum_k z_{\cdot k} \log \pi_k + \sum_\ell w_{\cdot \ell} \log \rho_\ell \\ &+ \sum_{i,j,k} z_{ik} w_{jk} \left( |x_{ij} - 1| \log \frac{\varepsilon_{kk}}{(1-\varepsilon_{kk})} + \log(1-\varepsilon_{kk}) \right) \\ &+ \sum_{i,j,k,\ell \neq k} z_{ik} w_{j\ell} \left( x_{ij} \log \frac{\varepsilon_{k\ell}}{1-\varepsilon_{k\ell}} + \log(1-\varepsilon_{k\ell}) \right), \end{aligned} \tag{6}$$

where $z_{\cdot k} = \sum_i z_{ik}$ and $w_{\cdot \ell} = \sum_j w_{j\ell}$.

Similar to LBCEM (see Algorithm 1), we optimize this criterion by alternating the maximization of two complete

log-likelihoods: $L_C(\mathbf{z}, \boldsymbol{\theta}|\mathbf{w})$ and $L_C(\mathbf{w}, \boldsymbol{\theta}|\mathbf{z})$. The maximization procedures consist of the following three steps. 1. Maximization of $L_C(\mathbf{z}, \mathbf{w}, \boldsymbol{\theta})$ w.r.t. $\mathbf{z}$ for fixed $\boldsymbol{\theta}$ and $\mathbf{w}$. As $L_C(\mathbf{z}, \mathbf{w}, \boldsymbol{\theta})$ can be written as $\sum_{i,k} z_{ik} A_{ik} + \sum_{\ell} w_{.\ell} \rho_{\ell}$ where

$$A_{ik} = \log \pi_k + \log \frac{\varepsilon_{kk}}{(1-\varepsilon_{kk})} |x_{ik}^{\mathbf{w}} - w_{.k}| + w_{.k} \log(1-\varepsilon_{kk}) + \sum_{\ell \neq k} x_{i\ell}^{\mathbf{w}} \log \frac{\varepsilon_{k\ell}}{(1-\varepsilon_{k\ell})} + w_{.\ell} \log(1-\varepsilon_{k\ell}), \quad (7)$$

we deduce $z_i = \arg\max_k A_{ik}$. 2. Maximization of $L_C(\mathbf{z}, \mathbf{w}, \boldsymbol{\theta})$ w.r.t. $\mathbf{w}$ for fixed $\boldsymbol{\theta}$ and $\mathbf{z}$. In the same manner, as $L_C(\mathbf{z}, \mathbf{w}, \boldsymbol{\theta})$ can be written as $\sum_{j\ell} w_{j\ell} B_{j\ell} + \sum_k z_{.k} \pi_k$ where

$$B_{j\ell} = \log \rho_{\ell} + \sum_{k \neq \ell} |x_{kj}^{\mathbf{z}} - z_{.k}| \log \frac{\varepsilon_{kk}}{(1-\varepsilon_{kk})} + z_{.k} \log(1-\varepsilon_{kk}) + \sum_{k \neq \ell} x_{kj}^{\mathbf{z}} \log \frac{\varepsilon_{k\ell}}{(1-\varepsilon_{k\ell})} + z_{.k} \log(1-\varepsilon_{k\ell}), \quad (8)$$

we deduce $w_j = \arg\max_{\ell} B_{j\ell}$.
3. Maximization of $L_C(\mathbf{z}, \mathbf{w}, \boldsymbol{\theta})$ w.r.t. $\boldsymbol{\theta}$ for fixed $\mathbf{z}$ and $\mathbf{w}$: $L_C(\mathbf{z}, \mathbf{w}, \boldsymbol{\theta})$ takes the following form

$$\sum_k |x_{kk}^{\mathbf{zw}} - z_{.k} w_{.k}| \log \frac{\varepsilon_{kk}}{(1-\varepsilon_{kk})} + z_{.k} w_{.k} \log(1-\varepsilon_{kk}) + \sum_{k,\ell \neq k} x_{k\ell}^{\mathbf{zw}} \log \frac{\varepsilon_{k\ell}}{(1-\varepsilon_{k\ell})} + z_{.k} w_{.\ell} \log(1-\varepsilon_{k\ell}) + \sum_k z_{.k} \log \pi_k + \sum_{\ell} w_{.\ell} \log \rho_{\ell},$$

where $x_{kk}^{\mathbf{zw}} = \sum_{i,j} z_{ik} w_{jk} x_{ij}$ and $x_{k\ell}^{\mathbf{zw}} = \sum_{i,j} z_{ik} w_{j\ell} x_{ij}$. The optimization of $L_C(\mathbf{z}, \mathbf{w}, \boldsymbol{\theta})$ is subject to the following constraints:

$$\sum_k \pi_k = 1, \pi_k \in [0,1] \quad \text{and} \quad \sum_{\ell} \rho_{\ell} = 1, \rho_{\ell} \in [0,1]. \quad (9)$$

Therefore, to maximize this criterion with respect to the necessary conditions for optimality we use the method of Lagrange multiplier and obtain:

$$\mathcal{L}_C = L_C(\mathbf{z}, \mathbf{w}, \boldsymbol{\theta}) - \lambda_1 \left( \sum_k \pi_k - 1 \right) - \lambda_2 \left( \sum_{\ell} \rho_{\ell} - 1 \right), \quad (10)$$

where $\lambda_1$ and $\lambda_2$ are the Lagrange multipliers. Then, by setting the derivatives of $\mathcal{L}_C$ with respect to each parameter and the Lagrange multipliers to zero, we obtain $\forall k, \ell$:

$$\hat{\varepsilon}_{kk} = \frac{|x_{kk}^{\mathbf{zw}} - z_{.k} w_{.k}|}{z_{.k} w_{.k}} \quad \text{and} \quad \hat{\varepsilon}_{k\ell} = \frac{x_{k\ell}^{\mathbf{zw}}}{z_{.k} w_{.\ell}}$$
$$\hat{\pi}_k = \frac{z_{.k}}{n} \quad \hat{\rho}_{\ell} = \frac{w_{.\ell}}{d} \quad (11)$$

From an algorithmic point of view, we choose to use the same strategy as for LBCEM. The key idea behind this approach is to fix the partition of columns while optimizing $L_C(\mathbf{z}, \boldsymbol{\theta}|\mathbf{w})$ on one hand (corresponds to steps 1 and 3 described above) and then, to fix the partition of rows while optimizing $L_C(\mathbf{w}, \boldsymbol{\theta}|\mathbf{z})$ on the other hand (corresponds to steps 2 and 3). As a result, we work on intermediate matrices of reduced size when compared with the initial one and therefore, this approach can handle data sets of higher dimension and also, converge faster. This strategy is summarized in Algorithm 2 which is referred to as [M1] in the sequel.

**Algorithm 2** [M1]

**input:** $\mathbf{x}$, $g$
**initialization:** $\mathbf{z}$, $\mathbf{w}$,
**compute:** $\hat{\pi}_k = \frac{z_{.k}}{n}$, $\hat{\rho}_{\ell} = \frac{w_{.\ell}}{d}$, $\hat{\varepsilon}_{kk}$ and $\hat{\varepsilon}_{k\ell}$ given in (11)
**repeat**
  $x_{i\ell}^{\mathbf{w}} = \sum_j w_{j\ell} x_{ij}$
  **repeat**
    **step 1.** $z_i = \arg\max_k A_{ik}$ given in eq (7)
    **step 2.** Compute $\hat{\pi}_k = \frac{z_{.k}}{n}$, $\hat{\varepsilon}_{kk}$ and $\hat{\varepsilon}_{k\ell}$ given in (11)
  **until** convergence
  $x_{kj}^{\mathbf{z}} = \sum_i z_{ik} x_{ij}$
  **repeat**
    **step 3.** $w_j = \arg\max_{\ell} B_{j\ell}$ given in eq (8)
    **step 4.** Compute $\hat{\rho}_{\ell} = \frac{w_{.\ell}}{d}$, $\hat{\varepsilon}_{kk}$ and $\hat{\varepsilon}_{k\ell}$ given in (11)
  **until** convergence
**until** convergence
**return** $\mathbf{z}$, $\mathbf{w}$, $\varepsilon_{k\ell}$, $\pi_k$ and $\rho_{\ell}$

### 3.3 Model $\mathcal{M}2$

In this model, we impose that the $\varepsilon_{k\ell}$'s of the *kth* cluster are equal for $\ell = 1, \ldots, m$. In this case, the complete data log-likelihood $L_C(\mathbf{z}, \mathbf{w}, \boldsymbol{\theta})$ or $L_C$ is given by

$$L_C = \sum_k z_{.k} \log \pi_k + \sum_{\ell} w_{.\ell} \log \rho_{\ell} + \sum_{i,j,k} z_{ik} w_{jk} \left( |x_{ij} - 1| \log \frac{\varepsilon_k}{(1-\varepsilon_k)} + \log(1-\varepsilon_k) \right) + \sum_{i,j,k,\ell \neq k} z_{ik} w_{j\ell} \left( x_{ij} \log \frac{\varepsilon_k}{(1-\varepsilon_k)} + \log(1-\varepsilon_k) \right).$$

Therefore, the three steps previously developed for [M1] become:
1. Maximization of $L_C(\mathbf{z}, \mathbf{w}, \boldsymbol{\theta})$ w.r.t. $\mathbf{z}$ for fixed $\boldsymbol{\theta}$ and $\mathbf{w}$. As $L_C(\mathbf{z}, \mathbf{w}, \boldsymbol{\theta})$ can be written as $\sum_{i,k} z_{ik} A_{ik} + \sum_{\ell} w_{.\ell} \rho_{\ell}$ where

$$A_{ik} = \log \pi_k + |x_{ik}^{\mathbf{w}} - w_{.k}| \log \frac{\varepsilon_k}{(1-\varepsilon_k)} + w_{.k} \log(1-\varepsilon_k) + \sum_{\ell \neq k} x_{i\ell}^{\mathbf{w}} \log \frac{\varepsilon_k}{(1-\varepsilon_k)} + w_{.\ell} \log(1-\varepsilon_k)$$
$$= \log \pi_k + \log \frac{\varepsilon_k}{(1-\varepsilon_k)} \left( |x_{ik}^{\mathbf{w}} - w_{.k}| + \sum_{\ell \neq k} x_{i\ell}^{\mathbf{w}} \right) + d \log(1-\varepsilon_k), \quad (12)$$

where $x_{ik}^{\mathbf{w}} = \sum_j w_{jk} x_{ij}$. This leads to $z_i = \arg\max_k A_{ik}$.
2. Maximization of $\mathrm{L_C}(\mathbf{z}, \mathbf{w}, \boldsymbol{\theta})$ w.r.t. $\mathbf{w}$ for fixed $\boldsymbol{\theta}$ and $\mathbf{z}$. As $\mathrm{L_C}(\mathbf{z}, \mathbf{w}, \boldsymbol{\theta})$ can be written as $\sum_{j,\ell} w_{j\ell} B_{j\ell} + \sum_k z_{.k} \pi_k$ where

$$B_{j\ell} = \log \rho_\ell + \sum_{k \neq \ell} |x_{kj}^{\mathbf{z}} - z_{.k}| \log \frac{\varepsilon_k}{(1-\varepsilon_k)} + z_{.k} \log(1-\varepsilon_k) + \sum_{k \neq \ell} x_{kj}^{\mathbf{z}} \log \frac{\varepsilon_k}{(1-\varepsilon_k)} + z_{.k} \log(1-\varepsilon_k). \quad (13)$$

This leads to $w_j = \arg\max_\ell B_{j\ell}$.
3. Maximization of $\mathrm{L_C}(\mathbf{z}, \mathbf{w}, \boldsymbol{\theta})$ w.r.t. $\boldsymbol{\theta}$ for fixed $\mathbf{z}$ and $\mathbf{w}$: $\mathrm{L_C}(\mathbf{z}, \mathbf{w}, \boldsymbol{\theta})$ takes he following form

$$\sum_k |x_{kk}^{\mathbf{zw}} - z_{.k} w_{.k}| \log \frac{\varepsilon_k}{(1-\varepsilon_k)} + z_{.k} w_{.k} \log(1-\varepsilon_k) + \sum_{k,\ell \neq k} x_{k\ell}^{\mathbf{zw}} \log \frac{\varepsilon_k}{(1-\varepsilon_k)} + z_{.k} w_{.\ell} \log(1-\varepsilon_k) + \sum_k z_{.k} \log \pi_k + \sum_\ell w_{.\ell} \log \rho_\ell$$

and we obtain $\hat{\pi}_k = \frac{z_{.k}}{n}$, $\hat{\rho}_\ell = \frac{w_{.\ell}}{d}$ and

$$\hat{\varepsilon}_k = \frac{|x_{kk}^{\mathbf{zw}} - z_{.k} w_{.k}| + \sum_{\ell \neq k} x_{i\ell}^{\mathbf{w}}}{z_{.k} \times d}. \quad (14)$$

Concerning the optimization process, we use the same strategy as for the model $\varepsilon_{k\ell}$ and obtain the Algorithm 3 which is referred to as [M2] in the following sections.

### 3.4 Model $\mathcal{M}3$

This last model is the simplest one. We consider that all the $\varepsilon_{k\ell}$ are equal for $\ell = 1, \ldots, m$ and $k = 1, \ldots, g$. The complete data log likelihood becomes

$$\mathrm{L_C}(\mathbf{z}, \mathbf{w}, \boldsymbol{\theta}) = \sum_k z_{.k} \log \pi_k + \sum_\ell w_{.\ell} \log \rho_\ell + \log \frac{\varepsilon}{(1-\varepsilon)} \sum_{i,k,j} z_{ik} w_{jk} |x_{ij} - 1| + \log(1-\varepsilon) \sum_k w_{.k} z_{.k}$$
$$+ \log \frac{\varepsilon}{(1-\varepsilon)} \sum_{i,j,k,\ell \neq k} z_{ik} w_{j\ell} x_{ij} + \log(1-\varepsilon) \sum_{k,\ell \neq k} z_{.k} w_{.\ell}.$$

which is equivalent to

$$\mathrm{L_C}(\mathbf{z}, \mathbf{w}, \boldsymbol{\theta}) = \sum_k z_{.k} \log \pi_k + \sum_\ell w_{.\ell} \log \rho_l + \log \frac{\varepsilon}{(1-\varepsilon)} \sum_{i,k,j} z_{ik} w_{jk} |x_{ij} - 1| + \log \frac{\varepsilon}{(1-\varepsilon)} \sum_{i,j,k,\ell \neq k} z_{ik} w_{j\ell} x_{ij} + (n \times d) \log(1-\varepsilon).$$

As for [M1] and [M2], the algorithm resulting from this model is based on the three steps defined as follows:
1. $A_{ik}$ becomes

$$A_{ik} = |x_{ik}^{\mathbf{w}} - w_{.k}| + \sum_{\ell \neq k} x_{i\ell}^{\mathbf{w}} + \log \pi_k, \quad (15)$$

and $z_i = \arg\min_k A_{ik}$.
2. $B_{j\ell}$ becomes

$$B_{j\ell} = \sum_{k \neq \ell} |x_{kj}^{\mathbf{z}} - z_{.k}| + \sum_{k \neq \ell} x_{kj}^{\mathbf{z}} + \log \rho_\ell, \quad (16)$$

and $w_j = \arg\min_\ell B_{j\ell}$.
3. And $\mathrm{L_C}(\mathbf{z}, \mathbf{w}, \boldsymbol{\theta})$ can be written as

$$\log \frac{\varepsilon}{(1-\varepsilon)} \left[ \sum_k |x_{kk}^{\mathbf{zw}} - z_{.k} w_{.k}| + \sum_{k,\ell \neq k} x_{k\ell}^{\mathbf{zw}} \right] + nd \log(1-\varepsilon) + \sum_k z_{.k} \log \pi_k + \sum_\ell w_{.\ell} \log \rho_\ell,$$

so we obtain $\hat{\pi}_k = \frac{z_{.k}}{n}$, $\hat{\rho}_\ell = \frac{w_{.\ell}}{d} \quad \forall k, \ell$.

**Algorithm 3** [M2]

**input:** $\mathbf{x}$, $g$
**initialization:** $\mathbf{z}$, $\mathbf{w}$,
**compute:** $\hat{\pi}_k = \frac{z_{.k}}{n}$, $\hat{\rho}_\ell = \frac{w_{.\ell}}{d}$ and $\hat{\varepsilon}_k = \frac{|x_{kk}^{\mathbf{zw}} - z_{.k} w_{.k}| + \sum_{\ell \neq k} x_{i\ell}^{\mathbf{w}}}{z_{.k} \times d}$
**repeat**
  $x_{i\ell}^{\mathbf{w}} = \sum_j w_{j\ell} x_{ij}$
  **repeat**
    **step 1.** $z_i = \arg\max_k A_{ik}$ given in eq (12)
    **step 2.** Compute $\hat{\pi}_k = \frac{z_{.k}}{n}$ and $\hat{\varepsilon}_k$ given in eq (14).
  **until** convergence
  $x_{kj}^{\mathbf{z}} = \sum_i z_{ik} x_{ij}$
  **repeat**
    **step 3.** $w_j = \arg\max_\ell B_{j\ell}$ given in eq (13)
    **step 4.** Compute $\hat{\rho}_\ell = \frac{w_{.\ell}}{d}$ and $\hat{\varepsilon}_k$ given in eq (14).
  **until** convergence
**until** convergence
**return** $\mathbf{z}$, $\mathbf{w}$, $\varepsilon_k$, $\pi_k$ and $\rho_\ell$

We notice that $\varepsilon$ is not involved in the optimization but it can be estimated from optimal partitions $(\mathbf{z}^*, \mathbf{w}^*)$ by $\hat{\varepsilon} = \frac{\sum_k |x_{kk}^{\mathbf{zw}} - z_{.k} w_{.k}| + \sum_{k,\ell \neq k} x_{k\ell}^{\mathbf{zw}}}{n \times d}$ and $(1 - \hat{\varepsilon})$ represents the global degree of homogeneity.

Once again, we choose to work on intermediates matrices like for the two previous models. Finally, the corresponding algorithm is presented in Algorithm 4 and is referred to as [M3].

**Algorithm 4** [M3]

**input:** $\mathbf{x}$, $g$
**initialization:** $\mathbf{z}$, $\mathbf{w}$
**compute:** $\hat{\pi}_k = \frac{z_{.k}}{n}$, $\hat{\rho}_\ell = \frac{w_{.\ell}}{d}$
**repeat**
  $x_{i\ell}^{\mathbf{w}} = \sum_j w_{j\ell} x_{ij}$
  **repeat**
    **step 1.** $z_i = \arg\max_k A_{ik}$ given in eq (15)
    **step 2.** Compute $\hat{\pi}_k = \frac{z_{.k}}{n}$
  **until** convergence
  $x_{kj}^{\mathbf{z}} = \sum_i z_{ik} x_{ij}$
  **repeat**
    **step 3.** $w_j = \arg\max_\ell B_{j\ell}$ given in eq (16)
    **step 4.** Compute $\hat{\rho}_\ell = \frac{w_{.\ell}}{d}$
  **until** convergence
**until** convergence
**return** $\mathbf{z}$, $\mathbf{w}$, $\varepsilon$, $\pi_k$, $\rho_\ell$

Finally, if we also assume equal proportions of the cluster in rows and columns which is $\pi_k = \frac{1}{g} \forall k$ and $\rho_\ell = \frac{1}{g} \forall \ell$, the proposed criterion can be expressed as:

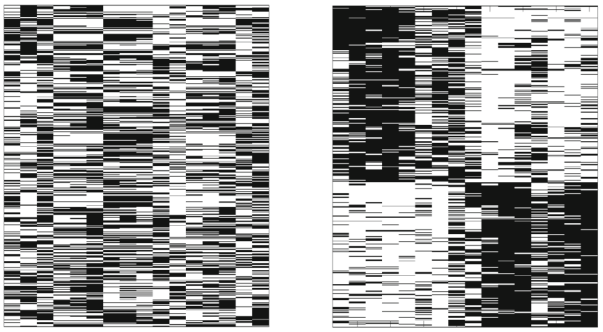
$$\log \frac{\varepsilon}{(1-\varepsilon)} W(\mathbf{z}, \mathbf{w}) + D,$$

where

$$W(\mathbf{z}, \mathbf{w}) = \sum_{i,j,k} z_{ik} w_{jk} |x_{ij} - 1| + \sum_{i,j,k,\ell \neq k} z_{ik} w_{j\ell} x_{ij} \quad (17)$$

and $D = (n \times d) \log(1 - \varepsilon)$. Since $\log \frac{\varepsilon}{(1-\varepsilon)} \leq 0$ and $D$ does not depend on $(\mathbf{z}, \mathbf{w})$, maximizing $\mathrm{L_C}(\mathbf{z}, \mathbf{w}, \boldsymbol{\theta})$ is equivalent to minimizing $W(\mathbf{z}, \mathbf{w})$. The final criterion has a very clear interpretation. On one hand, it minimizes the number of values different from 1 on the diagonal, and on the other hand it minimizes the number of 1's outside the diagonal. What's more, it is directly related to the criterion of CROBIN (Govaert 1983) algorithm and, more precisely, the constrained version proposed in Garcia and Proth (1986) used in a group technology application. It is also strongly related with the techniques of block seriation (Marcotorchino 1987). The algorithm resulting from this last model consists of two simple following steps:

1. $A_{ik} = |x_{ik}^{\mathbf{w}} - w_{.k}| + \sum_{\ell \neq k} x_{i\ell}^{\mathbf{w}}$, and $z_i = \arg\min_k A_{ik}$;
2. $B_{j\ell} = \sum_{k \neq \ell} |x_{kj}^{\mathbf{z}} - z_{.k}| + \sum_{k \neq \ell} x_{kj}^{\mathbf{z}}$ and $w_j = \arg\min_k B_{jk}$.



**(a)** Original data **(b)** Ordered data

**Fig. 4** Visualisation of Vote data set: original data and data reorganized according to the partitions of rows and columns. *Black* cell for "yea", *white* for "nay"

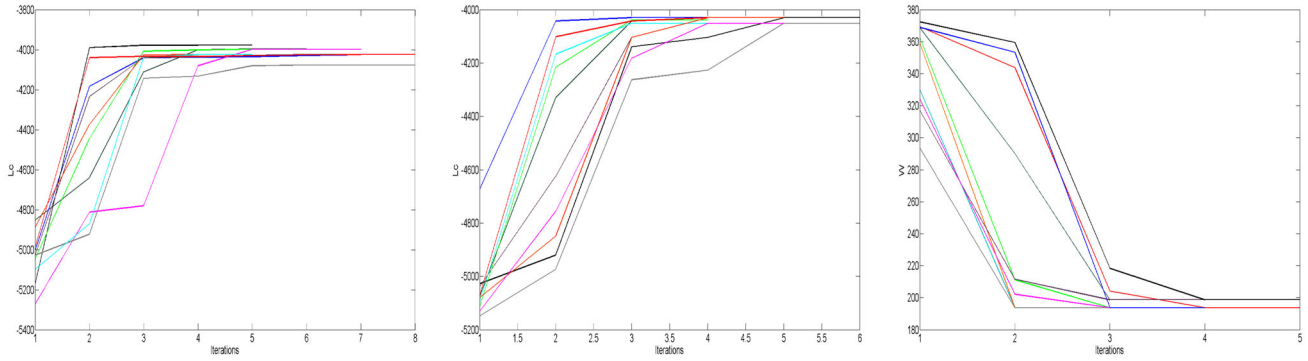
To summarize, we defined three parsimonious models and derived three algorithms [M1], [M3] and [M3] that are evaluated in Sects. 4 and 5 on both dense and sparse binary data sets.

### 3.5 Illustration on U.S. congressional voting data

In order to give a comprehensive illustration of how the proposed models work, we apply them to the United States Congressional Voting Records data.[1] This data set contains votes of each of the 435 members of the U.S. House of Representatives on 16 political issues. Each element is coded by 1 for "yea", 0 for "nay". Missing values (5.4 %) represent an absence or an abstention. For convenience, we convert missing values into "nay", i.e., 0 as in Lee and Huang (2014), Wyse and Friel (2012).

We apply the original LBM for Bernoulli data (referred as LBCEM) and all proposed algorithms by setting $g = 2$ because members are separated into two groups: the Republicans (168) and the Democrats (267). Also, in our models, the data are supposed to have a strong diagonal structure that leads to setting $m = g$. Figure 4 shows the original data set and the same data set where rows and columns are reorganized according to partitions obtained with [M3] resulting in a clear diagonal structure. Given that the obtained partitions with the two other models are quite similar to the one presented in Fig. 4, we choose not to show them. From this figure, we can observe a structure with clear high density regions on the diagonal when the initial matrix is reorganized with respect to the obtained cluster assignments. Figure 5 is the criterion optimized by each algorithm as a function of the number of iterations. We can see that all three algorithms

[1] https://archive.ics.uci.edu/ml/datasets.html.




**(a)** **(b)** **(c)**

**Fig. 5** Criterion of each algorithm obtained when starting with 10 random initializations: **a** [M1], **b** [M2] and **c** [M3] as functions of the number of iterations. As regards [M3], we represent the criterion $W$ that requires to be minimized

**Table 2** Vote data set: confusion matrices obtained with the four algorithms

| True label | Cluster | |
|---|---|---|
| | *R.* | *D.* |
| LBCEM | | |
| R. | **154** | 14 |
| D. | 42 | **225** |
| [M1] | | |
| R. | **154** | 14 |
| D. | 43 | **224** |
| [M2] | | |
| R. | **147** | 21 |
| D. | 33 | **234** |
| [M3] | | |
| R. | **154** | 14 |
| D. | 42 | **225** |

converge. Indeed, [M1] and [M2] maximize $L_C$ (see Eq. 6 and 12) and both converge in less than 6 iterations. In its turn, [M3] minimizes $W$ defined in Eq. 17 and converges in 4 iterations at most.

Concerning the column clusters, we obtain the exact same partition with all three models. Block (1,1) associates 8 issues: handicapped-infants, adoption-budget-resolution, anti-satellite-test-ban, aid-to-nicaraguan-contras, mx-missile, syn fuels-corporation-cutback, duty-free-export, export-administration-act-south-africa where most of the Democrats voted "yea" for these propositions, while the majority of Republicans voted 'nay" (block (2,1)). Inversely, the block (2,2) associates the following issues: water-project-cost-sharing, religious-groups-in-schools, el-salvador-aid, education-spending, crime, superfund-right-to-sue, physician-fee-freeze, immigration with the majority of Republicans who voted "yea" (block (2,2)) while the Democrats voted 'nay" for these propositions (block (1,2)). For each model, we report the confusion matrix (Table 2) resulting from the partition of rows.

**Table 3** Value of $\boldsymbol{\epsilon}$'s obtained with [M1] (on the left), [M2] (on the right) and [M3] (bottom line)

| | |
|---|---|
| $\boldsymbol{\epsilon} = (\varepsilon_{k\ell}) = \begin{pmatrix} 0.2838 & 0.2035 \\ 0.2273 & 0.1697 \end{pmatrix}$ | $\boldsymbol{\epsilon} = (\varepsilon_k) = \begin{pmatrix} 0.1795 & 0.2694 \end{pmatrix}$ |
| | $\boldsymbol{\epsilon} = (\varepsilon) = \begin{pmatrix} 0.2276 \end{pmatrix}$ |

Finally, we report the estimators of $\varepsilon_{k\ell}$, $\varepsilon_k$ and $\varepsilon$ reached with [M1], [M2] and [M3], respectively in Table 3. From this table, we can see that two models agree on the fact that two parties votes are distributed with a different level of heterogeneity: the values 0.2838 and 0.1697 in $\boldsymbol{\epsilon}$ matrix stand for Democrats' and Republicans' "yea" votes for their respective issues in $\mathcal{M}1$ model; the same behaviour can be observed in vector $\boldsymbol{\epsilon}$ for $\mathcal{M}2$ model, where the obtained values 0.2694 and 0.1795 summarize the heterogeneity of votes for Democrats and Republicans for both "yea" and "nay". The values obtained by both models indicate that in general Democrats vote in a more diverse way as their degree of heterogeneity is higher. Lastly, the $\boldsymbol{\epsilon}$ value obtained with [M3] shows in some way the average of the heterogeneity levels of both parties.

To conclude, we show that on the Vote data set, all proposed models are able to find a coherent partition with respect to the observed labels and perform equally well compared to the original LBCEM. From the values presented in Table 3, we also observe that each of the proposed models reveals different information about the data set: (1) $\mathcal{M}1$ shows a detailed heterogeneity of votes within two parties for both blocks of propositions; (2) $\mathcal{M}2$ presents the average heterogeneity of votes withing each party without specifying the corresponding issues; (3) $\mathcal{M}3$ presents the mean heterogeneity of votes for both parties. This important property of the introduced methods gives us a hint on how their results should be interpreted. Next, we propose to compare these

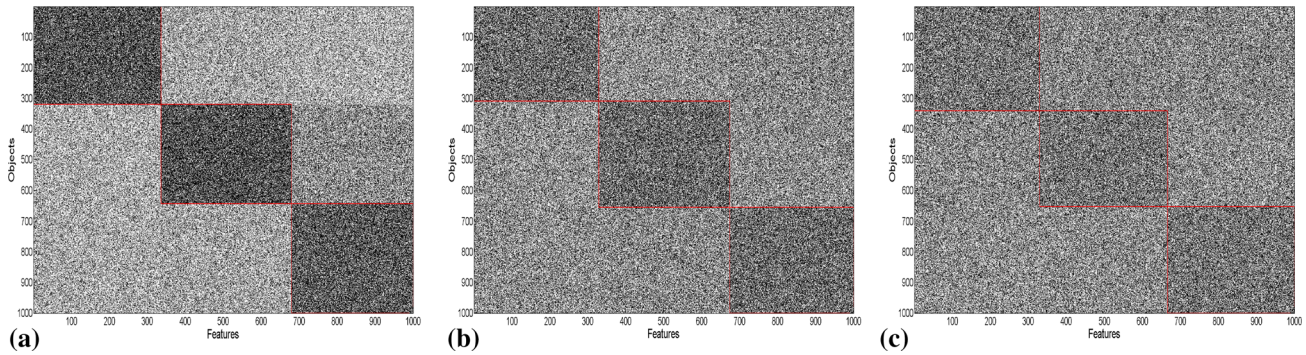



**Fig. 6** Visualisation of a simulated data set of size $1000 \times 1000$ with co-clusters of equal proportion **a** well separated, **b** moderately separated and **c** ill-separated

**Table 4** Value of the parameters used for the simulations

| Data | Size | Sparsity | $\alpha$ | Size | Sparsity | $\alpha$ |
|---|---|---|---|---|---|---|
| $+$ | $1000 \times 500$ | [50.1; 51.2] | $\boldsymbol{\alpha} = \begin{pmatrix} 0.57 & 0.43 & 0.47 \\ 0.43 & 0.58 & 0.45 \\ 0.44 & 0.46 & 0.58 \end{pmatrix}$ | $1000 \times 1000$ | [50.4; 50.6] | $\boldsymbol{\alpha} = \begin{pmatrix} 0.57 & 0.46 & 0.47 \\ 0.46 & 0.55 & 0.46 \\ 0.47 & 0.46 & 0.56 \end{pmatrix}$ |
| $++$ | – | [50.3; 50.8] | $\boldsymbol{\alpha} = \begin{pmatrix} 0.57 & 0.46 & 0.47 \\ 0.46 & 0.55 & 0.46 \\ 0.47 & 0.46 & 0.55 \end{pmatrix}$ | – | [50.8; 51.1] | $\boldsymbol{\alpha} = \begin{pmatrix} 0.54 & 0.46 & 0.47 \\ 0.46 & 0.54 & 0.46 \\ 0.47 & 0.48 & 0.53 \end{pmatrix}$ |
| $+++$ | – | [50.2; 50.9] | $\boldsymbol{\alpha} = \begin{pmatrix} 0.54 & 0.48 & 0.47 \\ 0.48 & 0.54 & 0.45 \\ 0.47 & 0.46 & 0.54 \end{pmatrix}$ | – | [50.3; 50.5] | $\boldsymbol{\alpha} = \begin{pmatrix} 0.53 & 0.48 & 0.475 \\ 0.48 & 0.54 & 0.47 \\ 0.47 & 0.48 & 0.53 \end{pmatrix}$ |

models on synthetic binary data sets with various characteristics in order to highlight the advantages and drawbacks of each one.

## 4 Synthetic data sets

In this section, we evaluate the three models on simulated binary data sets using three different criteria for 12 configuration of data depending on the size, the degree of overlapping and the co-cluster proportions. We also show what model is the most suitable for dense and for highly sparse data sets by performing extensive empirical experiments. We focus on the results in terms of rows clustering. All experiments are conducted on the same machine (OS: windows 7 Pro. 64-bits, Memory: 16 GiB, Processor: Intel®Core™i7-3770 CPU @ 3.40 GHz).

### 4.1 Experimental scheme

In order to study the three models described in Sect. 5, we simulate binary data sets using the latent block model proposed in Govaert and Nadif (2003). In our experiments we selected 12 types of data arising from $3 \times 3$ component Bernoulli mixture models corresponding to two data dimensions ($n \times d = 1000 \times 500$ and $n \times d = 1000 \times 1000$) and three cases of separation (well separated [$+$], moderately separated [$++$] and ill-separated [$+++$]) (see Fig. 6). The degree of separation can be measured by the true error rate, that is defined as the expectation of misclassification probability $\mathbb{E}(\delta(\mathbf{z}, d(\mathbf{x})))$, where $\mathbf{z}$ and $\mathbf{x}$ are random variables associated to the model, $d$ is the optimal Bayes rule, $d(\mathbf{x}) = \arg\max_{\mathbf{x}} P(\mathbf{z}|\mathbf{x})$, associated to this model and $\delta$ is the error rate defined previously. Since $\delta$ is difficult to compute theoretically, we used Monte Carlo simulations and approximated the error rate by comparing the partition obtained by initializing the algorithm with the true classes and stopped after the first iteration. In our experiments the parameters were selected so as to obtain true error rates in $[0.01, 0.05]$ for the well-separated [$+$], in $[0.09, 0.15]$ for the moderately-separated [$++$] and in $[0.19, 0.25]$ for the ill-separated [$+++$] classes. Values of $\boldsymbol{\alpha}$ used for all simulations are reported in Table 4. Finally, we consider both equal and unequal proportions ($\pi = \rho = (0.2, 0.3, 0.5)$) of co-clusters.

For each data structure, we generate 100 samples. For each sample, we run the algorithms 200 times starting from random initialisations and retain the best result according to the corresponding criterion.

### 4.2 Performance evaluation

In order to assess and to compare the performance of the proposed algorithms we use three commonly adopted metrics

including accuracy, Normalize Mutual Information (Strehl and Ghosh 2003) and Adjusted Rand Index (Hubert and Arabie 1985). Clustering accuracy (noted Acc) is one of the most widely used evaluation criteria and is defined as:

$$\text{Acc} = \frac{1}{n} max \left[ \sum_{\mathcal{C}_k, \mathcal{L}_\ell} T(\mathcal{C}_k, \mathcal{L}_\ell) \right],$$

where $\mathcal{C}_k$ is the *kth* cluster in the final results, and $\mathcal{L}_\ell$ is the true $\ell th$ class. $T(\mathcal{C}_k, \mathcal{L}_\ell)$ is the proportion of objects that were correctly recovered by the clustering algorithm, i.e., $T(\mathcal{C}_k, \mathcal{L}_\ell) = \mathcal{C}_k \cap \mathcal{L}_\ell$. Accuracy computes the maximum sum of $T(\mathcal{C}_k, \mathcal{L}_\ell)$ for all pairs of clusters and classes.

The second measure used is the Normalized Mutual Information (NMI). It is calculated as follows:

$$\text{NMI} = \frac{\sum_{k,\ell} \frac{n_{k\ell}}{n} \log \frac{n_{k\ell}}{n_k \hat{n}_\ell}}{\sqrt{\left(\sum_k \frac{n_k}{n} \log \frac{n_k}{n}\right) \left(\sum_\ell \frac{\hat{n}_\ell}{n} \log \frac{\hat{n}_\ell}{n}\right)}},$$

where $n_k$ denotes the number of data contained in the cluster $\mathcal{C}_k (1 \leq k \leq K)$, $\hat{n}_\ell$ is the number of data belonging to the class $\mathcal{L}_\ell$ $(1 \leq \ell \leq K)$, and $n_{k\ell}$ denotes the number of data that are in the intersection between the cluster $\mathcal{C}_k$ and the class $\mathcal{L}_\ell$.

The last evaluation criterion *Adjusted Rand Index* (noted ARI) measures the similarity between two clustering partitions. From a mathematical standpoint, the Rand index is related to the accuracy. The adjusted form of the Rand Index is defined as:

$$\text{ARI} = \frac{\sum_{k,\ell} \binom{n_{k\ell}}{2} - \left[\sum_k \binom{n_k}{2} \sum_\ell \binom{\hat{n}_\ell}{2}\right] \Big/ \binom{n}{2}}{\frac{1}{2}\left[\sum_k \binom{n_k}{2} + \sum_\ell \binom{\hat{n}_\ell}{2}\right] - \left[\sum_k \binom{n_k}{2} \sum_\ell \binom{\hat{n}_\ell}{2}\right] \Big/ \binom{n}{2}}.$$

The values of Acc and NMI are between 0 and 1 and the ARI ranges from $-1$ to 1. For both metrics, a value equal to 1 indicates a perfect agreement between the true and the estimated partition. In the sequel, we express these metrics in percentage.

Finally, in clustering, the proportion of misclassified instances can be directly defined as $e(\mathbf{z}, \hat{\mathbf{z}}) = 1 - \text{Acc}$. This definition can be extended to the comparison of co-clusterings results as follows. Denoted by CCE, the co-clustering error (Patrikainen and Meila 2006; Govaert and Nadif 2008), which takes into account the relationship between the partition of instances and the partition of variables we have:

$$\text{CCE}((\mathbf{z}, \mathbf{w}), (\hat{\mathbf{z}}, \hat{\mathbf{w}})) = e(\mathbf{z}, \hat{\mathbf{z}}) + e(\mathbf{w}, \hat{\mathbf{w}}) - e(\mathbf{z}, \hat{\mathbf{z}}) \times e(\mathbf{w}, \hat{\mathbf{w}}),$$

where $\hat{\mathbf{z}}$ and $\hat{\mathbf{w}}$ are the partitions of instances and variables estimated by the algorithm while $\mathbf{z}$ and $\mathbf{w}$ are the true partitions. However, since the true labels of variables are not available for benchmark data sets we only used this criteria on synthetic data sets.

### 4.3 Results

Table 5 presents the mean and the standard deviation of the accuracy, the NMI and the ARI for each data structure. The degree of sparsity (i.e., the proportion of 0's in the matrix) of these data sets is between 50 and 52 %. We compare all three proposed algorithms and the original LBCEM for binary data. We also use the parametric Student's t-test to check for significant difference of performance between tested methods. A $p$ value smaller than 0.05 suggests that the null hypothesis is false, i.e., there are statistically significant differences of performance between the compared algorithms. We compare algorithms in pairs and a boldface type indicates that the algorithm significantly outperforms all others. Several comments can be made based on these results.

- All three algorithms give comparable results in terms of performance on the well separated (+) and moderately separated (++) data. The classic LBCEM also performs equally well.
- On (+++) data sets with both equal and unequal proportions of the clusters, we note that [M2] is more effective than [M1] for row clustering, and the difference with [M3] is all the more important; as [M3] assumes an equal degree of heterogeneity, it is not able to propose a good partition when the structure of data sets is more complex. However, when considering row and column clustering performances measured by CCE criterion, we observe that [M1] surpasses [M2] in all situations. This can be easily explained by the fact that the hypothesis behind [M2] allows it to distinguish only the degree of heterogeneity between rows contrary to [M1]. In its turn, [M1] is allowed to assign levels of heterogeneity to blocks that makes it more flexible in terms of overall co-clustering performance. To summarize, for [M1] correctly revealing the true co-clustering partition of data takes priority over accurate clustering of rows only, while [M2] is particularly aimed at finding homogeneous row clusters and the expression of $A_{ik}$ (12) highlights this fact.

Regarding the computational cost, Table 6 reports means and standard deviations of the time required by our algorithms in order to converge:

- One can note that [M3] requires less time to converge than [M1] and [M2]. The difference becomes even more

**Table 5** Means of Acc, NMI and ARI (±standard deviations) computed on 200 simulated samples for equal and unequal proportions of the co-clusters

| Proportion | Size | Degree of overlap | Performance | Algorithms | | | |
|---|---|---|---|---|---|---|---|
| | | | | LBCEM | [M1] | [M2] | [M3] |
| Equal prop. | (1000, 500) | + | Acc | 98.04 ± 0.45 | 98.07 ± 0.40 | 97.86 ± 0.50 | 97.79 ± 0.48 |
| | | | NMI | 90.13 ± 1.79 | 90.60 ± 1.56 | 90.10 ± 1.92 | 89.84 ± 1.83 |
| | | | ARI | 94.22 ± 1.30 | 94.30 ± 1.15 | 93.70 ± 1.45 | 93.50 ± 1.40 |
| | | | CCE | 1.76 ± 0.29 | 1.47 ± 0.14 | 2.39 ± 0.50 | 2.34 ± 0.48 |
| | – | ++ | Acc | 90.12 ± 1.09 | 90.91 ± 1.07 | 91.49 ± 0.95 | 91.22 ± 1.06 |
| | | | NMI | 66.75 ± 2.85 | 67.17 ± 2.94 | 68.83 ± 2.48 | 68.05 ± 2.85 |
| | | | ARI | 74.12 ± 2.75 | 74.69 ± 2.78 | 76.19 ± 2.43 | 75.48 ± 2.73 |
| | | | CCE | 13.33 ± 1.18 | 11.46 ± 0.79 | 12.08 ± 1.42 | 12.58 ± 2.01 |
| | – | +++ | Acc | 77.06 ± 10.90 | 77.13 ± 8.11 | 77.98 ± 7.38 | 73.76 ± 7.74 |
| | | | NMI | 43.28 ± 9.94 | **43.51** ± 8.70 | 41.71 ± 11.43 | 36.52 ± 10.39 |
| | | | ARI | 46.77 ± 14.20 | **47.39** ± 11.58 | 46.79 ± 13.20 | 41.34 ± 11.65 |
| | | | CCE | 29.31 ± 12.56 | **24.12** ± 7.96 | 26.61 ± 8.12 | 26.48 ± 8.36 |
| Unequal prop. | (1000, 500) | + | Acc | 97.29 ± 0.39 | **98.37** ± 0.33 | 97.64 ± 0.48 | 97.42 ± 0.54 |
| | | | NMI | 90.08 ± 1.79 | **91.48** ± 1.56 | 89.05 ± 1.95 | 87.86 ± 2.23 |
| | | | ARI | 94.18 ± 1.13 | **95.42** ± 0.90 | 93.30 ± 1.37 | 92.62 ± 1.50 |
| | | | CCE | 1.72 ± 0.44 | **1.11** ± 0.15 | 2.58 ± 0.59 | 2.87 ± 0.60 |
| | – | ++ | Acc | 89.09 ± 1.33 | 89.72 ± 1.92 | **91.08** ± 1.03 | 86.56 ± 3.35 |
| | | | NMI | 62.55 ± 3.18 | 62.88 ± 3.92 | **66.13** ± 2.72 | 56.49 ± 6.61 |
| | | | ARI | 73.31 ± 2.96 | 73.73 ± 3.53 | **76.41** ± 2.41 | 66.09 ± 6.90 |
| | | | CCE | 15.40 ± 5.89 | **11.64** ± 0.74 | 13.57 ± 1.72 | 16.52 ± 4.09 |
| | – | +++ | Acc | 78.57 ± 4.18 | 80.21 ± 4.18 | **83.67** ± 1.89 | 76.48 ± 3.83 |
| | | | NMI | 49.77 ± 2.56 | 50.13 ± 2.92 | **51.54** ± 3.07 | 40.69 ± 5.67 |
| | | | ARI | 59.31 ± 4.11 | 60.89 ± 4.46 | **64.38** ± 2.73 | 52.49 ± 6.24 |
| | | | CCE | 31.02 ± 11.78 | 26.54 ± 8.25 | 28.31 ± 9.91 | 27.35 ± 9.64 |
| Equal prop. | (1000, 1000) | + | Acc | 96.05 ± 0.45 | 96.07 ± 0.58 | 96.56 ± 0.51 | 96.28 ± 0.57 |
| | | | NMI | 82.60 ± 1.74 | 82.95 ± 2.06 | **84.70** ± 1.87 | 83.91 ± 2.06 |
| | | | ARI | 88.07 ± 1.24 | 88.58 ± 1.60 | 89.95 ± 1.42 | 89.14 ± 1.57 |
| | | | CCE | 2.37 ± 0.42 | 1.93 ± 0.24 | 2.84 ± 0.51 | 2.83 ± 0.55 |
| | – | ++ | Acc | 91.74 ± 1.23 | 91.98 ± 1.35 | 93.02 ± 1.13 | 92.97 ± 1.00 |
| | | | NMI | 70.46 ± 3.42 | 70.40 ± 3.67 | 73.17 ± 3.19 | 73.13 ± 2.95 |
| | | | ARI | 77.46 ± 3.26 | 77.49 ± 3.51 | 80.20 ± 2.99 | 80.10 ± 2.65 |
| | | | CCE | 12.56 ± 2.31 | 7.64 ± 1.81 | 7.87 ± 2.02 | 7.35 ± 2.55 |
| | – | +++ | Acc | 70.57 ± 13.09 | 74.44 ± 13.25 | **80.24** ± 12.11 | 75.05 ± 14.72 |
| | | | NMI | 31.86 ± 18.20 | 36.37 ± 18.97 | **44.14** ± 17.03 | 36.69 ± 20.85 |
| | | | ARI | 35.95 ± 20.75 | 40.30 ± 25.77 | **49.01** ± 19.87 | 41.22 ± 23.80 |
| | | | CCE | 32.03 ± 11.23 | **21.40** ± 6.73 | 27.02 ± 9.92 | 23.75 ± 6.32 |
| Unequal prop | (1000, 1000) | + | Acc | 95.01 ± 1.02 | 95.55 ± 0.90 | 96.07 ± 0.81 | 94.61 ± 0.87 |
| | | | NMI | 78.21 ± 3.54 | 80.34 ± 3.05 | **82.32** ± 2.84 | 77.98 ± 2.78 |
| | | | ARI | 86.25 ± 2.87 | 87.93 ± 2.27 | **89.51** ± 2.02 | 85.369 ± 2.07 |
| | | | CCE | 3.75 ± 0.66 | **2.22** ± 0.39 | 3.26 ± 0.70 | 3.70 ± 0.55 |
| | – | ++ | Acc | 91.11 ± 1.00 | 93.35 ± 1.08 | 94.01 ± 0.84 | 87.68 ± 4.34 |
| | | | NMI | 71.45 ± 2.95 | 73.11 ± 2.89 | **75.08** ± 2.46 | 59.40 ± 9.43 |
| | | | ARI | 80.99 ± 2.12 | 82.66 ± 2.48 | **84.34** ± 2.22 | 68.46 ± 9.57 |
| | | | CCE | 14.12 ± 6.31 | **7.11** ± 3.57 | 9.12 ± 5.12 | 12.47 ± 5.21 |

**Table 5** continued

| Proportion | Size | Degree of overlap | Performance | Algorithms | | | |
|---|---|---|---|---|---|---|---|
| | | | | LBCEM | [M1] | [M2] | [M3] |
| | – | +++ | Acc | 77.35 ± 5.84 | 81.62 ± 6.42 | **84.35** ± 0.72 | 67.53 ± 5.50 |
| | | | NMI | 42.66 ± 6.02 | 49.35 ± 5.02 | **52.14** ± 1.71 | 24.20 ± 5.56 |
| | | | ARI | 44.54 ± 8.96 | 61.65 ± 7.59 | **64.41** ± 1.66 | 28.59 ± 7.75 |
| | | | CCE | 28.21 ± 9.63 | **22.65** ± 8.56 | 25.25 ± 9.81 | 24.37 ± 9.32 |

The degree of sparsity for all datasets is between 50 and 52 %. A boldface type indicates a significantly better result regarding a Student Test ($p$ value lower than 0.05)

**Table 6** Means (±standard deviations) of the time (in milliseconds) required for the convergence on each data structure

| Size | Degree of overlap | Equal prop. | | | | Unequal prop. | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | LBCEM | [M1] | [M2] | [M3] | LBCEM | [M1] | [M2] | [M3] |
| (1000, 500) | + | 17.9 ± 5.7 | 19.2 ± 6.5 | 20.8 ± 7.9 | **12.5** ± 3.0 | 17.6 ± 5.3 | 19.6 ± 5.8 | 20.5 ± 7.2 | **14.4** ± 4.00 |
| | ++ | 76.0 ± 31.5 | 83.6 ± 38.6 | 68.4 ± 22.5 | **45.7** ± 13.1 | 70.4 ± 31.3 | 70.4 ± 31.5 | 64.1 ± 24.5 | **56.0** ± 15.8 |
| | +++ | 94.3 ± 31.9 | 136.4 ± 34.7 | 105.6 ± 24.2 | **50.9** ± 12.0 | 67.3 ± 14.4 | 84.1 ± 29.1 | 77.3 ± 25.6 | **44.8** ± 12.6 |
| (1000, 1000) | + | 96.8 ± 24.4 | 281.7 ± 125.7 | 104.1 ± 29.4 | **68.8** ± 28.9 | 127.3 ± 42.1 | 214.8 ± 135.4 | 132.2 ± 55.1 | **112.4** ± 33.7 |
| | ++ | 244.6 ± 75.6 | 274.1 ± 64.8 | 225.9 ± 73.8 | **102.9** ± 28.8 | 182.3 ± 79.1 | 221.6 ± 113.2 | **155.7** ± 66.1 | 181.3 ± 26.7 |
| | +++ | 228.4 ± 46.6 | 290.5 ± 49.3 | 316.7 ± 58.3 | **164.4** ± 47.8 | 185.2 ± 42.7 | 304.8 ± 62.9 | 218.9 ± 62.5 | **131.1** ± 57.3 |

A boldface type indicates a significantly better result regarding a Student Test ($p$ value lower than 0.05)

remarkable on (+++) data set where [M3] is up to 2 times faster than [M2] and 3 times faster than [M1].
– These observations are valid regardless to the size although differences of performances are more marked on 1000 × 1000 than 1000 × 500 data sets.

As a summary, we can say that on (+) and (++) data sets with equal proportions, all algorithms give comparable clustering results and the speed of convergence observed for [M3] makes it particularly attractive. However, the fall of its accuracy observed on data sets with unequal proportions suggests that [M1] and [M2] are better candidates when the clusters are unbalanced. For more complex data sets (high degree of overlapping and unequal proportions), [M2] appears to be a good trade-off between the number of parameters to estimate, larger with [M1], and the clustering performance, lower with [M3].

### 4.4 Model selection

In the previous sections we described and evaluated models $\mathcal{M}1$, $\mathcal{M}2$ and $\mathcal{M}3$ that seek a diagonal block structure in binary data. These models are based on different assumptions about the heterogeneity of blocks and therefore involve different number of parameters. We demonstrate that the size of a data set as well as the proportions and the degree of separability of the co-clusters can influence the performance of each model. Hereafter, we propose a strategy aiming to help the user in selection of the most suitable model based on the data set characteristics.

We choose to study this question from a model selection point of view, i.e., identify the best model with respect to a chosen penalized criteria. However, in the context of co-clustering, commonly used penalized criteria such as AIC or BIC are not directly available. In order to address this issue, different model selection techniques have been adapted to latent block models, including AIC3 (Van Dijk et al 2009), ICL (Lomet 2012; Keribin et al 2015) and an approximation of BIC (Keribin et al 2015). In what follows, we select ICL over BIC for two reasons: (1) we are maximizing the complete log-likelihood of the data which is directly involved in ICL calculation; (2) the results presented in Keribin et al (2015) demonstrate that ICL outperforms BIC on small data sets and performs equally well on large ones. We also exclude AIC3 because it is mainly dedicated to the problem of prediction and lacks a clear probabilistic interpretation. The approximation of the Integrated Completed Likelihood (ICL) (Biernacki et al 2000) adapted for latent block models can be expressed as:

$$ICL(g,m) \simeq L_C(\mathbf{z},\mathbf{w},\boldsymbol{\theta}) - \frac{g-1}{2}\log(n) - \frac{m-1}{2}\log(d) - \frac{K_{\mathcal{M}}}{2}\log(nd),$$

where the second and the third terms are penalties related to the estimation of the proportions of blocks and $K_M$ depends

**Table 7** The frequency of the models selected by the ICL criterion on 50 data sets of size $1000 \times 500$ with well separated clusters, with equal proportions and different degree of sparsity

| Model | Sparsity | | | |
|---|---|---|---|---|
| | ∼50 % | ∼60 % | ∼70 % | ∼80 % |
| $\mathcal{M}1$ | **47** | **26** | 0 | 0 |
| $\mathcal{M}2$ | 3 | 24 | 7 | 0 |
| $\mathcal{M}3$ | 0 | 0 | **43** | **50** |

**Table 8** Intermediate matrices obtained at iteration 1

$$\mathbf{x}^{\mathbf{zw}} = (x_{k\ell}^{\mathbf{zw}}) = \begin{pmatrix} 2250 & 499 & 616 \\ 569 & 1875 & 620 \\ 590 & 486 & 2465 \end{pmatrix}$$

$$\widetilde{\mathbf{x}}^{\mathbf{zw}} = \left(\frac{x_{k\ell}^{\mathbf{zw}}}{z_{.k}w_{.\ell}}\right) = \begin{pmatrix} 0.0397 & 0.0106 & 0.0140 \\ 0.0098 & 0.0386 & 0.0102 \\ 0.0100 & 0.0099 & 0.0402 \end{pmatrix}$$

on the number of probability function parameters of the model. In the case of the proposed models $\mathcal{M}1$, $\mathcal{M}2$, $\mathcal{M}3$, we have $K_{\mathcal{M}1} = gm$, $K_{\mathcal{M}2} = g$ and $K_{\mathcal{M}3} = 1$. Finally, the fact that we are specifically seeking a structure of diagonal blocks naturally lead us to take $g = m$.

In the results subsection presented above, we simulated the data sets with a degree of sparsity between 50 and 52 % and came to the conclusion that $\mathcal{M}1$ and $\mathcal{M}2$ are more appropriate in this specific case. However, many real-world data sets are known to have about 98 % of zeros (see Table 10). To cover this scenario, we propose to study the impact of sparsity on model selection by simulating data sets of size $1000 \times 500$ with well-separated co-clusters and a degree of sparsity varying between 50 and 80 %. In Table 7, we report the frequency of the models selection obtained using ICL criterion. One can observe that on data sets with degree of sparsity close to 50 %, $\mathcal{M}1$ is preferred over $\mathcal{M}2$ and $\mathcal{M}3$. When the degree of sparsity increases and achieves 60 %, the frequency of picking $\mathcal{M}1$ is approximately equivalent to the frequency of $\mathcal{M}2$ while $\mathcal{M}3$ still remains unpicked. Finally, when the sparsity is larger than 70 %, $\mathcal{M}3$ is selected almost all the time.

The fact that sparsity strongly impacts the model selection can be explained with the following simple example: we simulate a $1000 \times 500$ data sets with 98 % of zeros and well separated co-clusters. We initialise [M1] and [M2] with the true classes for the rows and columns partitions. We report in Table 8 the intermediate matriced obtained after the first iteration.

Although we can notice that the diagonal blocks contain more 1's than the blocks outside the diagonal, their number is still negligible when compared with the number of 0's. Therefore, on such sparse data set, the hypothesis that $a_{kk}$'s should be equal to one is not verified. The violation of this hypothesis leads to very high values of $\varepsilon$ on the diagonal ($\geq \frac{1}{2}$) and results in the inability of both [M1] and [M2] to achieve the clustering tasks. However, [M3] is not sensitive to this issue from the fact that, by assuming all $\varepsilon$'s equals, these latter are not anymore involved in the estimations and maximisation process of the algorithm.

To conclude, we can say that based on the presented results, model selection based on a penalized information criteria like ICL appears to be a good strategy when assessing the quality of co-clustering models on a given data set. It also confirms the fact that sparsity of the data set has an important influence on the choice of an appropriate model and justifies our decision to keep $\mathcal{M}3$ when dealing with highly sparse data.

### 4.5 Sparse binary data

Hereafter, we compare the behaviour of [M3] and the original LBCEM on very sparse data in terms of clustering performance. In order to evaluate the performance of both algorithms on ∼90 % sparse data and ∼98 % data, we simulate data set of size $1000 \times 500$ with both equal and unequal proportions of blocks and choose $g = 3$ and the results are reported in Table 9.

From these experiments we can see that on very sparse data LBCEM and [M3] give comparable results in terms of performance (see Table 9). Table 9 also shows the percentage of success over 100 trials of each algorithm with random initialisation. While [M3] almost always succeeds in finding the block structure, LBCEM is not able to achieve this goal almost half of the times on 90 % sparse data and 90 % of the times on 98 % sparse data. For the latter, [M3] always gives better results in terms of performance than LBCEM and this difference is all the more important when the proportions are unequal.

Finally, Fig. 7 presents the accuracy, NMI and ARI as functions of the sparsity degree. It can be easily seen that [M3] is much more robust to increasing sparsity than LBCEM. Next we confirm this observation on real data sets with high sparsity.

## 5 Document-term partitioning

In this section, we study the effectiveness of our algorithm for some well-known text data sets with different sizes and balances (the balance coefficient is defined as the ratio of the number of instances in the smallest class to the number of instance in the largest class). Regarding the terminology used before, each instance now is represented by a document from a corpus while each feature is a term appearing at least once in the corpus. We compare the clustering performance

**Table 9** Means of Acc, NMI and ARI (±standard deviations) computed on 200 simulated samples

| Size | Performance | Sparsity | | | |
|---|---|---|---|---|---|
| | | ∼90 % | | ∼98 % | |
| | | LBCEM | [M3] | LBCEM | [M3] |
| (1000, 500) Eq. prop. | Acc | 98.05 ± 0.63 | 98.00 ± 0.41 | 89.71 ± 6.34 | **94.82 ± 1.05** |
| | NMI | 90.76 ± 2.58 | 90.51 ± 1.71 | 68.89 ± 7.46 | **78.57 ± 3.04** |
| | ARI | 94.23 ± 1.84 | 94.12 ± 1.18 | 73.91 ± 11.62 | **85.10 ± 2.89** |
| | Success (%) | 58.9 | 100 | 8.2 | 100 |
| (1000, 500) Uneq. prop | Acc | 97.24 ± 0.52 | 97.17 ± 0.45 | 83.64 ± 8.12 | **92.95 ± 0.80** |
| | NMI | 86.53 ± 2.03 | 86.90 ± 1.88 | 65.71 ± 5.54 | **72.30 ± 2.70** |
| | ARI | 92.26 ± 1.49 | 91.93 ± 1.31 | 70.59 ± 9.52 | **81.79 ± 2.16** |
| | Success (%) | 60.1 | 100 | 11.2 | 100 |

Success indicates the percentage of times the algorithms achieved to return a partition with the requested number of co-clusters
Numbers in bold stand for the best obtained result

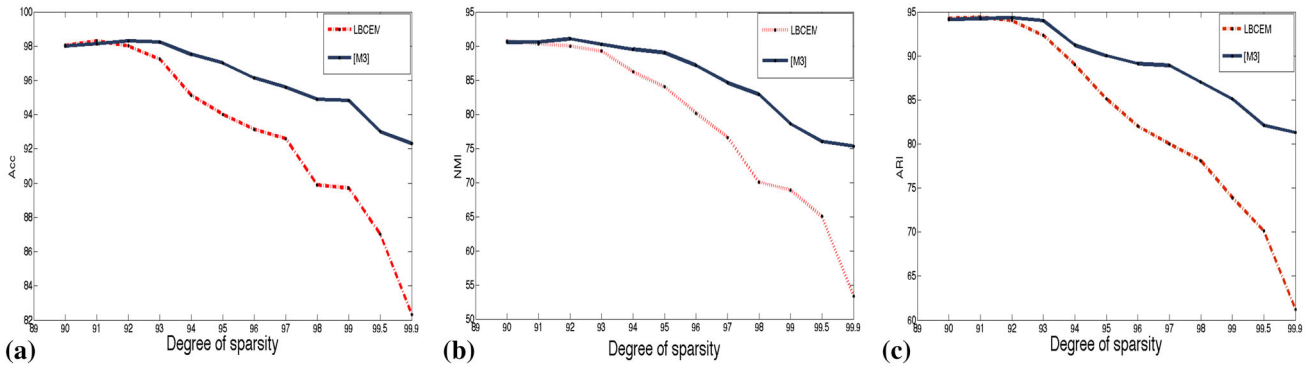



**Fig. 7** Evolution of the mean of the accuracy (**a**), the NMI (**b**) and the ARI (**c**) for [M3] and LBCEM with the degree of sparsity of the data set. We consider data sets of size 1000 × 500 with equal proportion

of our algorithm with state-of-the-art (co)-clustering algorithms commonly used in the context of *document* × *term* clustering.

### 5.1 Data sets

Hereafter, we give a detailed description of chosen data sets.

- **Classic4** consists of 4 different document collections: MED, CISI, CRAN and CACM. We also use a subset of this data set (CISI, CRAN and MED only) referred to as Classic3, in the sequel. Terms which appear in less than 3 documents, or in more than 95 % of the documents were removed. Moreover, Porter's stemming was applied as pre-processing step.
- **CSTR** contains abstracts of technical reports published in the department of Computer Science at a research university. These abstracts were divided into four research areas: Natural Language Processing (NLP), Robotics/ Vision, Systems and Theory.
- **Reviews** and **Sports** are two *document* × *term* matrices from the software *CLUTO*.[2] They were derived from the San Jose Mercury newspaper articles. Reviews contains documents on such topics as food, movies, music, radio and restaurants; Sports contains articles about baseball, basketball, bicycling, boxing, football, golfing and hockey.
- **TDT2** consists of data collected from 6 sources, including 2 newswires (APW, NYT), 2 radio programs (VOA, PRI) and 2 television programs (CNN, ABC). We use a subset of this data set where documents that appear in two or more categories were removed and only the largest 30 categories were kept.
- **RCV1** is a corpus of news-wire stories made available by Reuters, Ltd. We use a subset of the RCV1 corpus including categories C15, ECAT, GCAT and MCAT.
- **20 Newsgroups** is a set of Usenet articles organized into 20 topics. We also use two subsets of NG20: (1) NG5 that includes the following topics: comp.os.ms-

[2] http://www.cs.umn.edu/~cluto.

**Table 10** Description of the data sets in terms of size ($n \times p$), number of clusters ($K$), sparsity (%0) and degree of balance of the clusters

| Data set | $n \times p$ | K | %0 | Balance |
|---|---|---|---|---|
| CSTR | 475 × 1000 | 4 | 96.60 | 0.399 |
| NG2 | 500 × 2000 | 2 | 97.19 | 1 |
| NG5 | 500 × 2000 | 5 | 97.19 | 1 |
| Classic3 | 3891 × 4303 | 3 | 98.95 | 0.710 |
| Classic4 | 7095 × 5896 | 4 | 99.41 | 0.323 |
| Reviews | 4069 × 18,483 | 5 | 98.99 | 0.099 |
| Sports | 8580 × 14,870 | 7 | 99.04 | 0.036 |
| TDT2 | 9394 × 36,771 | 30 | 99.65 | 0.028 |
| RCV1 | 9625 × 29,992 | 4 | 99.75 | 0.697 |
| NG20 | 19,949 × 43,586 | 20 | 99.82 | 0.991 |

windows, comp.windows.x, rec.motorcycles, sci.crypt and sci.space; and (2) NG2 containing two topics rec.motorcycles and sci.crypt, sci.space.

The characteristics of all data sets used are reported in Table 10. Originally each value of these data sets shows the number of occurrences of a term in a document. The original data were converted into binary by setting each value greater than 0 to 1 and 0 otherwise, i.e., we only consider the absence or presence of a term in a document.

### 5.2 Results

We compare our method to state-of-the-art (co)-clustering methods including LBCEM for binary data (Govaert and Nadif 2003), ITCC (Dhillon et al 2003), Block (Li 2005) and SpCo (Dhillon 2001). ITCC is a divisive algorithm that directly minimizes an objective function based on the mutual information. We choose K-means and the simple Bernoulli mixture model (CEM) (Celeux and Govaert 1992) as baselines. LBCEM and CEM belongs to the same family of approach than our model. Block, ITCC and SpCo are co-clustering algorithms. We note that SpCo specifically seeks diagonal blocks.

For ITCC, we use the Matlab Toolbox for Biclustering Analysis (MTBA) (Gupta et al 2013). For SpCo algorithm we use the implementation proposed by Assaf Gottlieb.[3]. We use the Matlab Statistics and Machine Learning Toolbox for K-means and implement other (co)-clustering algorithms using Matlab.

We set the number of clusters to the true number of classes for all data sets. We run all algorithms 100 times with random initializations and report the best result, i.e., the one that minimizes (or maximizes depending on the algorithm) the corresponding criterion over all trials in Table 11. Figure 9 shows the computational costs for each algorithm. From these experiments, we can observe the following statements:

- [M3] vs. LBCEM and CEM: on each data set (except Classic3 where [M3] ≈ ITCC), [M3] significantly outperforms the original LBCEM and CEM in terms of accuracy, NMI and ARI. For instance, this gap is obvious on data sets from the 20 News Group corpus: CEM is unable to propose a partition on NG2 and NG20, and LBCEM's ARI are never above 20 %. Furthermore, [M3] is faster than both CEM and LBCEM: up to 6 and 3.5 times faster respectively than these latter on Sports and RCV1. Figure 8 is an illustration of the results of LBCEM and [M3].One can see that while [M3] achieves to find a clear diagonal structure on CSTR, the original LBCEM fails.
- [M3] vs SpCo: [M3] is always better in terms of clustering performance than SpCo. We can also observe that on RCV1 and Classic4, [M3] performs well while SpCo is unable to propose a partition. We have deliberately chosen not to report times obtained with SpCo. Although the implementation we used is faithful to the method described in Dhillon (2001), we believe it is not the optimal one.
- [M3] vs Block and ITCC: we observe that ITCC outperforms Block on almost all data sets (except in the case of CSTR and RCV1). ITCC is mostly effective when the clusters are well separated, except for NG20 in terms of NMI and ARI. However, on all other data sets [M3] outperforms ITCC. In terms of speed, [M3] and Block require comparable times on Classic4 and CSTR, but on larger data sets [M3] is faster than Block. For the same reason as stated for SpCo, times for ITCC are not reported.

To conclude, we can make two remarks: (1) [M3] outperforms all other state-of-the-art (co)-clustering algorithms on almost all data sets, especially on the most unbalanced one. Indeed, its ARI for Sports, Reviews and TDT2 are significantly higher than others; (2) it is important to underline that we are processing the binary version of the $document \times term$ matrices and not the original co-occurrence table that is supposed to provide more information.

## 6 Conclusion

In this article, we proposed to study the problem of diagonal co-clustering using the framework of latent block model. By imposing constraints on a parsimonious Bernoulli latent block model that only depends the degree of heterogeneity in blocks and the proportions of rows and columns, respec-

[3] http://adios.tau.ac.il/.

**Table 11** Accuracy, normalized mutual information and adjusted rand index obtained on binary data sets

| Data set | Metric | Algorithms | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | Kmeans | CEM | LBCEM | Block | ITCC | SpCo | [M3] |
| **CSTR** | Acc | 85.05 | 86.32 | 83.79 | 81.05 | 69.94 | 79.79 | **90.11** |
| | NMI | 64.74 | 66.95 | 69.91 | 63.31 | 70.00 | 66.67 | **77.92** |
| | ARI | 68.14 | 71.60 | 71.73 | 60.15 | 65.79 | 70.20 | **81.55** |
| **NG2** | Acc | – | – | 55.80 | 72.01 | 90.80 | 90.20 | **95.20** |
| | NMI | – | – | 1.87 | 20.56 | 56.57 | 55.35 | **72.79** |
| | ARI | – | – | 1.22 | 24.51 | 66.52 | 64.57 | **81.68** |
| **NG5** | Acc | 33.07 | 33.67 | 38.28 | 53.21 | 54.31 | 60.32 | **80.36** |
| | NMI | 16.61 | 11.66 | 17.50 | 34.15 | 36.74 | 50.75 | **55.40** |
| | ARI | 4.08 | 7.82 | 8.79 | 29.96 | 31.45 | 37.31 | **56.82** |
| **Classic3** | Acc | 90.47 | **99.20** | 64.02 | 92.24 | 98.46 | 70.60 | 98.12 |
| | NMI | 73.81 | **95.47** | 43.87 | 74.71 | 92.32 | 59.64 | 90.77 |
| | ARI | 73.34 | **97.58** | 39.87 | 79.29 | 95.42 | 40.20 | 94.40 |
| **Classic4** | Acc | 74.88 | 77.80 | 56.11 | 58.34 | 64.87 | – | **85.47** |
| | NMI | 51.46 | 64.23 | 30.12 | 39.23 | 50.96 | – | **66.09** |
| | ARI | 43.28 | 57.77 | 21.71 | 29.76 | 42.56 | – | **62.29** |
| **Reviews** | Acc | 53.58 | 54.56 | 40.53 | 53.94 | 57.97 | 57.97 | **65.59** |
| | NMI | 42.76 | 41.81 | 21.25 | 42.23 | 45.71 | 45.71 | **53.44** |
| | ARI | 29.81 | 25.64 | 11.75 | 29.17 | 34.51 | 34.51 | **48.98** |
| **Sports** | Acc | 41.81 | 52.16 | 35.85 | 50.16 | 44.73 | 55.73 | **78.08** |
| | NMI | 33.21 | 48.05 | 26.35 | 45.29 | 49.16 | 47.18 | **63.43** |
| | ARI | 13.67 | 29.33 | 15.30 | 24.62 | 18.67 | 34.45 | **68.15** |
| **TDT2** | Acc | 47.60 | 18.19 | 39.73 | 46.55 | 56.32 | 35.31 | **74.75** |
| | NMI | 62.02 | 12.81 | 59.09 | 60.91 | 71.25 | 51.78 | **76.66** |
| | ARI | 32.36 | 8.21 | 32.64 | 30.71 | 61.24 | 29.45 | **67.83** |
| **RCV1** | Acc | 54.36 | 58.03 | 39.82 | 53.92 | 65.58 | – | **74.18** |
| | NMI | 32.92 | 36.47 | 15.21 | 32.67 | 41.35 | – | **51.13** |
| | ARI | 24.08 | 26.62 | 11.49 | 24.01 | 38.56 | – | **51.00** |
| **NG20** | Acc | 22.46 | 7.21 | 23.57 | 20.05 | 43.01 | 24.51 | **55.71** |
| | NMI | 26.03 | 2.34 | 35.17 | 22.71 | **55.18** | 29.35 | 52.57 |
| | ARI | 5.37 | 1.31 | 13.75 | 4.56 | **41.75** | 5.71 | 39.20 |

(–) denotes that the algorithm cannot propose a partition with a required number of co-clusters
Numbers in bold stand for the best obtained result

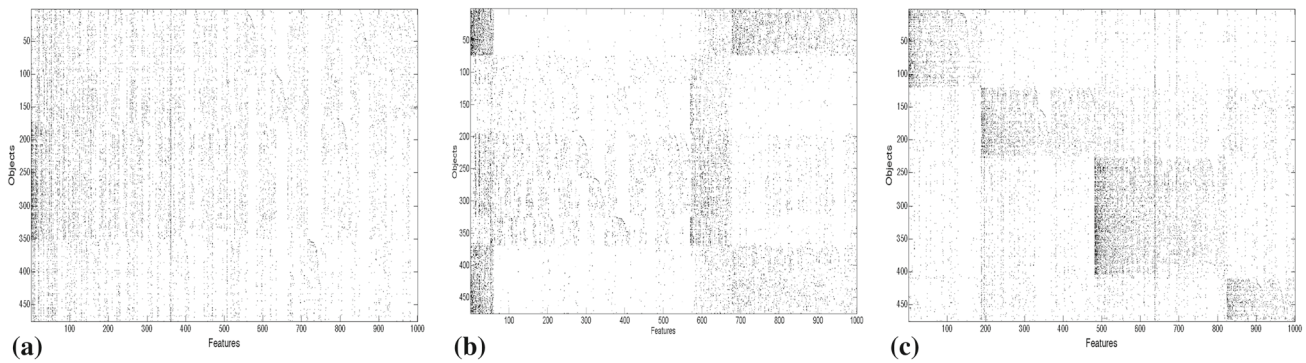



**Fig. 8** **a** Original CSTR data, **b** CSTR reorganized according to partitions obtained with LBCEM, and **c** CSTR according to partitions obtained with [M3]

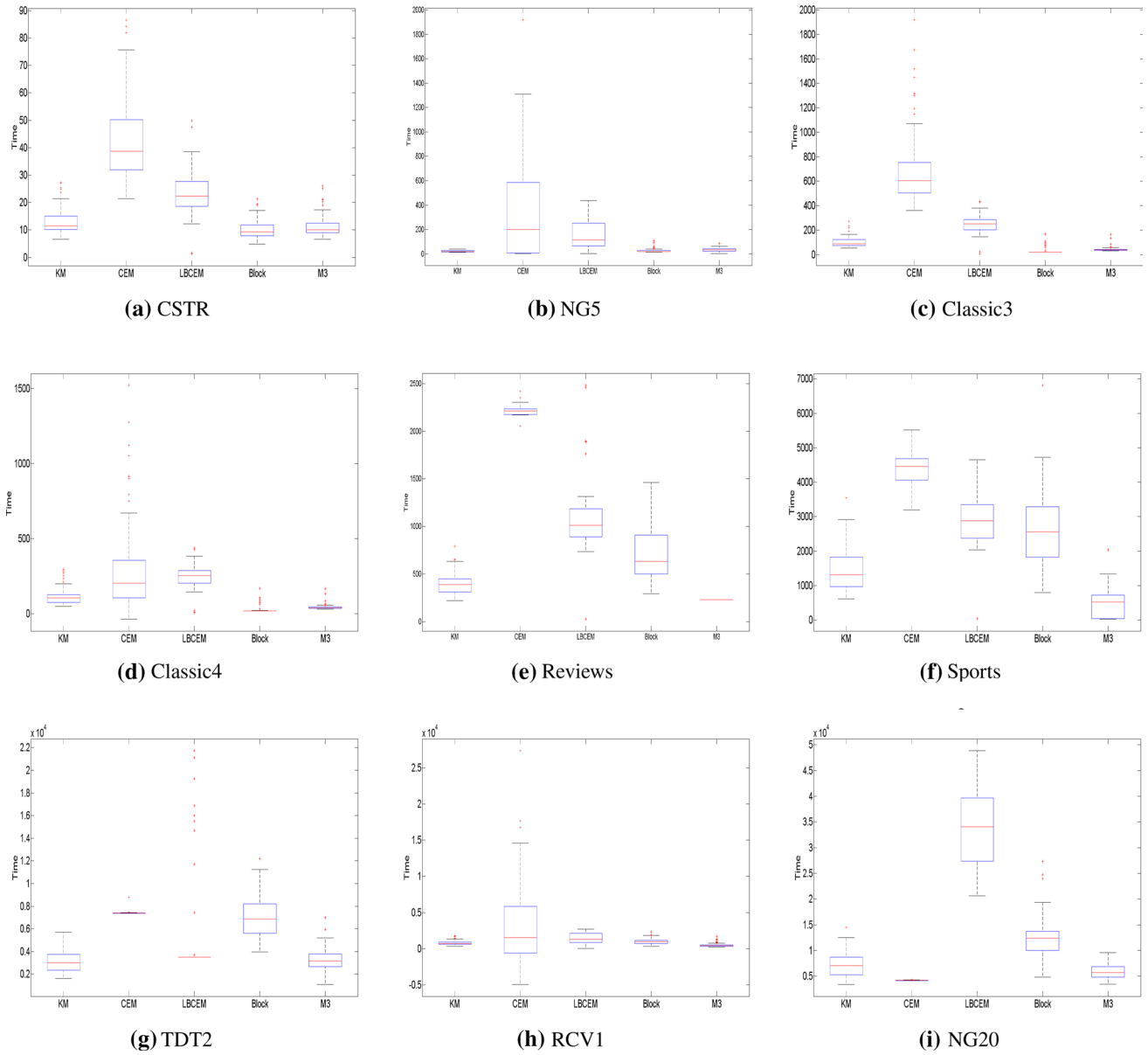



**Fig. 9** Time (in milliseconds) required for the convergence on each data set

tively we derived three models. We set our approach in the classifications maximum likelihood context and carried out the optimisation Classification EM algorithm yielding the development of three algorithms: [M1], [M2] and [M3]. We evaluated them on synthetic data in order to highlight the strength and weakness of each one on different data structure and came to the conclusion that on data sets with a medium degree of sparsity, $\mathcal{M}1$ and $\mathcal{M}2$ are the most accurate while on very sparse data the last model, $\mathcal{M}3$, remains the only valid one. In order to strengthen these observations, we also proposed to use the ICL criterion for model selection, that confirmed the interest of $\mathcal{M}3$ for sparse data. Compared with other methods, we also demonstrated that our proposed $\mathcal{M}3$ is more effective for $document \times term$ partitioning and competitive in terms of speed. Therefore, we can argue that seeking a diagonal structure is of a considerable interest in the context of document clustering, where the data sets are highly sparse.

Further research perspectives are many. The most important one is to use variational EM algorithm for parameters optimization. This algorithm is known to be more accurate as its goal is to find soft partitions of rows and columns. In real-world applications, the knowledge of the number of co-clusters is mostly required. Another initiative will be to investigate an efficient way to assess this parameter. In particular, we proposed to study the problem of model selection by

using ICL, an information penalized criteria; one could, for instance, extend this strategy for determining an appropriate number of co-clusters.

**Acknowledgments** This work has been funded by AAP Sorbonne Paris Cité.

## References


Banerjee, A., Dhillon, I., Ghosh, J., Merugu, S., Modha, D.S.: A generalized maximum entropy approach to bregman co-clustering and matrix approximation. J. Mach. Learn. Res. **8**, 1919–1986 (2007)

Batagelj, V., Ferligoj, A., Doreian, P.: Fitting Pre-specified Blockmodels. Springer, Tokyo (1998)

Biernacki, C., Celeux, G., Govaert, G.: Assessing a mixture model for clustering with the integrated completed likelihood. IEEE Trans. Pattern Anal. Mach. Intell. **22**(7), 719–725 (2000)

Bock, H.: Convexity based clustering criteria: theory, algorithm and applications in statistics. Stat. Methods Appl. **12**, 293–318 (2003)

Celeux, G., Govaert, G.: A classification EM algorithm for clustering and two stochastic versions. Comput. Stat. Data Anal. **14**(3), 315–332 (1992)

Cheng, Y., Church, G.M.: Biclustering of expression data. In: ISMB, pp. 93–103 (2000)

Cho, H., Dhillon, I., Guan, Y., Sra, S.: Minimum sum-squared residue co-clustering of gene expression data. In: SIAM-SDM, pp. 114–125 (2004)

Dhillon, I.: Co-clustering documents and words using bipartite spectral graph partitioning. In: KDD '01: SIGKDD, ACM, pp. 269–274 (2001)

Dhillon, I., Mallela, S., Kumar, R.: A divisive information theoretic feature clustering algorithm for text classification. J. Mach. Learn. Res. **3**, 1265–1287 (2003)

Ding, C., Li, T., Peng, W., Park, H.: Orthogonal nonnegative matrix t-factorizations for clustering. In: SIGKDD, ACM, pp. 126–135 (2006)

Doreian, P., Batagelj, V., Ferligoj, A.: Generalized Blockmodeling. Cambridge University Press, New York (2005)

Garcia, H., Proth, J.M.: A new cross-decomposition algorithm: The GPM comparison with the bond energy method. Control Cybern. **15**, 155–165 (1986)

George, T.: A scalable collaborative filtering framework based on co-clustering. In: ICDM, pp. 625–628 (2005)

Girolami, M.: The topographic organization and visualization of binary data using multivariate-Bernoulli latent variable models. IEEE Trans. Neural Netw. **12**(6), 1367–1374 (2001)

Govaert, G.: Classification croisée. Thèse d'état, Université Paris 6, France (1983)

Govaert, G.: Simultaneous clustering of rows and columns. Control Cybern. **24**(4), 437–458 (1995)

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

Govaert, G., Nadif, M.: An EM algorithm for the block mixture model. IEEE Trans. Pattern Anal. Mach. Intell. **27**(4), 643–647 (2005)

Govaert, G., Nadif, M.: Block bernoulli parsimonious clustering models. Selected Contributions in Data Analysis and Classification, Studies in Classification, Data Analysis, and Knowledge Organization, pp. 203–212. Springer, Berlin (2007)

Govaert, G., Nadif, M.: Block clustering with Bernoulli mixture models: comparison of different approaches. Comput. Stat. Data Anal. **52**(6), 3233–3245 (2008)

Govaert, G., Nadif, M.: Latent block model for contingency table. Commun. Stat. Theory Methods **39**(3), 416–425 (2010)

Govaert, G., Nadif, M.: Co-Clustering. Wiley (2013)

Gupta, J., Singh, S., Verma, NK.: MTBA: Matlab toolbox for biclustering analysis. IEEE, pp. 94–97 (2013)

Hofmann, T., Puzicha, J.: Latent class models for collaborative filtering. IJCAI, pp. 688–693. Morgan Kaufmann Publishers Inc., San Francisco (1999)

Hubert, L., Arabie, P.: Comparing partitions. J. Classif. **2**, 193–218 (2013)

Kabán, A., Bingham, E.: Factorisation and denoising of 0–1 data: a variational approach. Neurocomputing **71**(10), 2291–2308 (2008)

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

Labiod, L., Nadif, M.: Co-clustering for binary and categorical data with maximum modularity. In: ICDM, pp. 1140–1145 (2011)

Lee, S., Huang, J.: A biclustering algorithm for binary matrices based on penalized Bernoulli likelihood. Stat. Comput. **24**(3), 429–441 (2014)

Li, T.: A general model for clustering binary data. In: ACM SIGKDD, ACM, pp. 188–197 (2005)

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

Madeira, S.C., Oliveira, A.L.: Biclustering algorithms for biological data analysis: a survey. IEEE ACM Trans. Comput. Biol. Bioinform. **1**(1), 24–45 (2004)

Marcotorchino, F.: Block seriation problems: a unified approach. Appl. Stoch. Models Data Anal. **3**, 73–91 (1987)

Patrikainen, A., Meila, M.: Comparing subspace clusterings. IEEE Trans. Knowl. Data Eng. **18**(7), 902–916 (2006)

Si, L., Jin, R.: Flexible mixture model for collaborative filtering. In: ICML, pp. 704–711 (2003)

Strehl, A., Ghosh, J.: Cluster ensembles—a knowledge reuse framework for combining multiple partitions. J. Mach. Learn. Res. **3**, 583–617 (2003)

Symons, M.J.: Clustering criteria and multivariate normal mixture. Biometrics **37**, 35–43 (1981)

Van Dijk, B., Van Rosmalen, J., Paap, R.: A bayesian approach to two-mode clustering (2009)

Van Mechelen, I., Bock, H.H., De Boeck, P.: Two-mode clustering methods: a structured overview. Stat. Methods Med. Res. **13**(5), 363–394 (2004)

Vichi, M.: Double k-means clustering for simultaneous classification of objects and variables. In: Borra et al., (eds). Springer, Heidelberg (2001)

Wyse, J., Friel, N.: Block clustering with collapsed latent block models. Stat. Comput. **22**(2), 415–428 (2012)