Co-clustering of ordinal data via latent continuous random variables and not missing at random entries

Marco Corneli, Charles Bouveyron, Pierre Latouche

Co-Clustering of ordinal data via latent continuous random variables and a classification EM algorithm

Marco Corneli
Department of Statistics (LJAD), University Côte d'Azur, Nice, France
and
Charles Bouveyron
Department of Statistics (LJAD), University Côte d'Azur, Nice, France
Epione, INRIA Sophia-Antipolis, Valbonne, France
and
Pierre Latouche
Department of Statistics (MAP5), University Paris Descartes, Paris, France

January 11, 2019

Abstract

This paper is about the co-clustering of ordinal data. Such data are very common on e-commerce platforms where customers rank the products/services they bought. More in details, we focus on arrays of ordinal (possibly missing) data involving two disjoint sets of individuals/objects corresponding to the rows/columns of the arrays. Typically, an observed entry (i, j) in the array is an ordinal score assigned by the individual/row i to the object/column j. A generative model for arrays of ordinal data is introduced along with an inference algorithm for parameters estimation. The model relies on latent continuous random variables and the fitting allows to simultaneously co-cluster the rows and columns of an array. The estimation of the model parameters is performed via a classification expectation maximization (C-EM) algorithm. A model selection criterion is formally obtained to select the number of row and column clusters. In order to show that our approach reaches and often outperforms the state of the art, we carry out numerical experiments on synthetic data. Finally, applications on real datasets highlight the model capacity to deal with very sparse arrays.

Keywords: categorical data, model based clustering, ICL.

1 Introduction

Data clustering plays a central role in all scientific and industrial fields where there is a need of data analysis. The goal of data clustering is to group similar data together, to provide a synthetic view of a dataset. Thus, *clusters* are homogeneous groups of data that can be interpreted in a common way. The nature of the data requires different types of clustering techniques to be applied. This paper focuses on peculiar categorical data in which categories are ordered: the *ordinal* data (Agresti, 2010). Such data is very common in marketing researches (see for instance Dillon et al., 1990), where people are asked to evaluate products and/or services on an ordinal scale. As an example, the last section of this paper focuses on a related dataset, where Amazom customers review fine foods. In more details, the dataset consists of an ordinal data matrix whose entry (i, j) is the note attributed to the j-th products by the i-th customer.

In the last decades, several clustering algorithms for ordinal data have been introduced in the literature (D'Elia and Piccolo, 2005; Podani, 2006; Gouget, 2006; Jollois and Nadif, 2009; Giordan and Diana, 2011; Fernández et al., 2016; Ranalli and Rocci, 2016; Biernacki and Jacques, 2016). Moreover, a recent work of McParland and Gormley (2016) adopted a model-based clustering approach for mixed data (including ordinal data) based on latent Gaussian random variables. In the example of the Amazon dataset, the clustering algorithms listed above can group the *rows* of the ordinal data matrix such that two customers share the same cluster if they tend to rate products similarly.

However, when the number of *columns* of the matrix is high, uncovering relevant row clusters is a particularly hard problem. Moreover, the number of variables makes the interpretation of the row clusters challenging. A solution to this issue is provided by co-clustering, which aims at simultaneously clustering the rows and the columns of a data matrix, thus providing a partition for rows and another one for columns. Several model-based co-clustering methods rely on the latent block model (LBM, Govaert and Nadif, 2008) who deals with matrices of binary data. Other extensions of LBM can tackle counting (Govaert and Nadif, 2010), real (Lomet, 2012), categorical (Keribin et al., 2015), functional (Bouveyron et al., 2018) and ordinal data (Jacques and Biernacki, 2018). Up to our knowledge, the

model described in Jacques and Biernacki (2018) is the only one specifically designed for the co-clustering of ordinal data. In that paper, the authors rely on a generative model for ordinal data, called the binary ordinal search (BOS, Biernacki and Jacques, 2016) model. This statistical model is both parsimonious, since each co-cluster is summarized by only two parameters, and easily interpretable. However, the inference procedure to learn the model parameters is based on a stochastic version of the EM algorithm (see both Dempster et al., 1977; Celeux and Govaert, 1991) which suffers scalability issues. Furthermore, the co-clustering approach proposed by Jacques and Biernacki (2018) only takes into account missing *at random* data. In other words, the frequency of missing data is assumed to be constant on average on each data co-cluster. As a matter of fact, in real applications, this assumption can be restrictive. For instance, a group of users could systematically review and note one subset of products more than another one, thus discriminating the two subsets, otherwise (possibly) indistinguishable. To overcome these issues, this work proposes an extension of the LBM to perform ordinal data co-clustering. Our approach takes advantage of the binary formulation of LBM to manage missing data possibly *not* missing at random. Moreover, the latent Gaussian modeling introduced in Gormley and Murphy (2010) is adapted to the co-clustering framework. We show in turn that the posterior distribution of the latent random variables is fully tractable. A classification EM algorithm (Celeux and Govaert, 1991) is then used to estimate the model parameters and simultaneously cluster the rows as well the columns of an ordinal data matrix. A model selection criterion is formally obtained to simultaneously select the number of row and column clusters.

This paper is organised as follows. Section 2 describes the co-clustering generative model that we introduce. The inference of the model parameters is detailed in Section 3. This section also focuses on further issues such as the algorithm initialization and the selection of the number of co-clusters. Section 4 presents some numerical experiments on synthetic data to assess the proposed methodology and shows that it performs favourably, compared to the state of the art. Section 5 presents a real data application on the Amazon fine foods dataset.

2 The model

The present section describes a statistical model to generate arrays of *ordinal* data. In order to properly take into account data sparsity (see Section 2.2), the model extends the binary LBM of the original paper of Govaert and Nadif (2008). Thus, the first part of this section is devoted to a description of the binary LBM, while the second part focuses on the latent framework adopted to model ordinal data.

2.1 Latent block model

Consider an $M \times P$ *incidence* matrix A such that A_{ij} is equal to 1 if one interaction between i and j is observed (e.g. the i-th user assigns a score to the j-th product), 0 otherwise. Rows are assumed to be clustered into Q row clusters and columns into L column clusters. An hidden vector R, of length M, is such that $R_i = q$ if the i-th row of A is in the q-th row cluster. Moreover, the i-th row of A is assumed to be assigned to its row cluster according to a multinomial distribution

$$\mathcal{M}(1, \rho := \{\rho_1, \ldots, \rho_Q\}),$$

where $\rho_q > 0$, for all q, and $\sum_{q=1}^{Q} \rho_q = 1$. Thence, being the cluster of the i-th row recorded into R_i, it holds that

$$\mathbb{P}(R_i = q) = \rho_q, \qquad \forall i.$$

Similarly, an hidden vector C, of length P, is such that $C_j = l$ iff the j-th column of A is in the l-th column cluster. The j-th column of A is assigned to its column cluster according to a multinomial distribution

$$\mathcal{M}(1, \delta := \{\delta_1, \ldots, \delta_L\}),$$

where $\delta_l > 0$, for all l and $\sum_{l=1}^{L} \delta_l = 1$, so that

$$\mathbb{P}(C_j = l) = \delta_l, \qquad \forall j.$$

The two vectors R and C are further assumed to be independent. In the following, when no confusion arises, the equivalent 0-1 notation will be employed. In that case, R will denote

a binary $M \times Q$ matrix and if the i-th row of A is in the q-th row cluster, $R_{iq} = 1$ and $R_{iv} = 0$, for all $v \neq q$. Similarly, C will denote a binary $P \times L$ matrix.

Conditionally on R and C, the entries of A are all independent and such that

$$A_{ij}|\{R, C\} \sim \mathcal{B}(\pi_{R_i C_j}), \qquad \forall i \leq M, j \leq P$$

where $\mathcal{B}(p)$ denotes a Bernoulli distribution of parameter p, $\pi_{ql} \in [0, 1]$, for all q, l and $\pi := \{\pi_{ql}\}_{q,l}$. According to this model, the complete data likelihood is

$$p(A, R, C|\pi, \rho, \delta) = \left(\prod_{i=1}^{M} \prod_{j=1}^{P} \pi_{R_i C_j}^{A_{ij}} (1 - \pi_{R_i C_j})^{1 - A_{ij}} \right) \left(\prod_{i=1}^{M} \rho_{R_i} \right) \left(\prod_{j=1}^{P} \delta_{C_j} \right). \tag{1}$$

2.2 Modeling ordinal data

Let us now consider an $M \times P$ matrix Y, whose *ordinal* entry Y_{ij}, conditionally on $A_{ij} = 1$, is a random variable taking values in $\{1, \ldots, K\}$, for some $K \in \mathbb{N}^*$, not depending on the pair (i, j). For the pairs (i, j) such that $A_{ij} = 0$, we assume that $Y_{ij} = 0$. Note that the matrix Y contains both *observed* and *missing* data. The observed data are the values Y_{ij} corresponding to $A_{ij} = 1$ and in real applications these values could be scores that users assign to some products. However, one user could (and generally will) rate only a subset of products. Thus, the unrated products are seen as missing values and coded as 0 in Y. Now, the sparsity of A is modelled by LBM. For instance, users densely ranking a single class of products are more likely to be clustered together (via π) when fitting the model to the data. Hence, the link between Y and A has an important consequence: the missing data in Y are *not* missing at random (see both Little and Rubin, 2014; Jacques and Biernacki, 2018). Before going further, two assumptions should be made.

Assumption 1. *Henceforth, we assume that an ordinal scale is consistently defined. For instance, in the example of customers evaluating products,* 1 *always means "very poor" and* K *always means "excellent". The assumption is necessary, otherwise the results obtained when fitting the model to the data would be completely misleading. The analyst should therefore take this into account when designing the data collection.*

Assumption 2. *The number of ordered levels K is assumed to be the same for all $Y_{ij}|A_{ij} = 1$. If it was not the case, a scale conversion pre-processing algorithm (see for instance Gilula et al., 2018) should be employed to normalize the number of levels.*

The model that we assume to be generating Y relies on hidden Gaussian random variables Z_{ij} such that

$$Z_{ij}|\{A_{ij} = 1, R, C\} \sim \mathcal{N}(\mu_{R_iC_j}, \sigma^2_{R_iC_j}). \tag{2}$$

Henceforth, $\mu := \{\mu_{ql}\}_{q,l}$ and $\sigma^2 := \{\sigma^2_{ql}\}_{q,l}$ will denote the sets of the Gaussian parameters. Similarly to Y,

$$Z_{ij}|\{A_{ij} = 0, R, C\} = 0 \qquad \text{a.s.} \tag{3}$$

and all the random variables Z_{ij} are collected into an hidden $M \times P$ matrix denoted by Z. Assume that $K - 1$ unknown real numbers (*thresholds*) $\gamma := (\gamma_1, \dots, \gamma_{K-1})$ are such that

$$-\infty =: \gamma_0 < \gamma_1 < \cdots < \gamma_{K-1} < \gamma_K := \infty.$$

Then, conditionally on the event $\{A_{ij} = 1\}$

$$Y_{ij} := \sum_{k=1}^{K} k \mathbf{1}_{]\gamma_{k-1}, \gamma_k[}(Z_{ij}), \tag{4}$$

where $\mathbf{1}_\Omega(\cdot)$ is the indicator function over the set $\Omega \subset \mathbb{R}$. Finally, conditionally on $\{A_{ij} = 1\}$ as well as R and C, we assume that the pairs (Y_{ij}, Z_{ij}) are mutually independent. Hence, the joint density of (Y, Z) can be written as

$$p(Y, Z|A, R, C, \mu, \sigma^2) = \prod_{i=1}^{M} \prod_{j=1}^{P} \left(\phi(Z_{ij}; \mu_{R_iC_j}, \sigma^2_{R_iC_j}) \mathbf{1}_{]\gamma_{Y_{ij}-1}, \gamma_{Y_{ij}}[}(Z_{ij}) \right)^{A_{ij}}, \tag{5}$$

where $\phi(\cdot; \mu_{ql}, \sigma^2_{ql})$ is the probability density function of a Gaussian distribution $\mathcal{N}(\mu_{ql}, \sigma^2_{ql})$ and we used that, conditionally on $\{A_{ij} = 0\}$, the pairs (Y_{ij}, Z_{ij}) are equal to $(0, 0)$ a.s. Eqs. 1-5 can be combined to obtain the complete data likelihood

$$p(Y, Z, A, R, C|\theta) = p(Y, Z|A, R, C, \mu, \sigma^2) p(A, R, C|\pi, \rho, \delta), \tag{6}$$

where $\theta := \{\mu, \sigma^2, \pi, \rho, \delta\}$ denotes the set of the model parameters.

3 Inference

In the first part of this section, the numbers Q of row clusters and L of column clusters are assumed to be given. A model selection criterion will be detailed later. Now, we aim at estimating the model parameters θ as well as the most likely posterior values of R and C. Let us start with a remark.

Remark 1 (**Thresholds**). *It is immediate to see that either γ or (μ, σ^2) need to be fixed in order for the model parameters to be identifiable and, from a generative point of view, it seems reasonable to fix γ, as it can be seen in Eq.* (4)*. However, notice that once γ is fixed to some value (for instance by randomly selecting $K-1$ Gaussian quantiles and sorting them) and the model fitted to the data, the estimated (parameters of the) random variables in Z lie in a space which is in general not related with the range of the ordinal entries in Y. Thus, in order to have easily interpretable results, γ is fixed as*

$$\gamma = (1.5, 2.5, \ldots, (K - 0.5)).$$

In order to illustrate the estimation strategy in detail, we need the following proposition.

Proposition 1. *Conditionally on the event $\{A_{ij} = 1, R_{iq} = 1, C_{jl} = 1\}$, the random variable Y_{ij} has probability mass function*

$$\mathbb{P}(Y_{ij} = k | A_{ij} = 1, R_{iq} = 1, C_{jl} = 1) = \eta_k^{(q,l)} \mathbf{1}_{\{1,\ldots,K\}}(k), \tag{7}$$

where

$$\eta_k^{(q,l)} := \Phi\left(\frac{\gamma_k - \mu_{ql}}{\sigma_{ql}}\right) - \Phi\left(\frac{\gamma_{k-1} - \mu_{ql}}{\sigma_{ql}}\right). \tag{8}$$

Proof. By the definition of marginal probability density function, it follows that

$$\begin{aligned} p(Y_{ij} | A_{ij} = 1, R, C, \theta) &= \int_{\mathbb{R}} p(Y_{ij}, z | A_{ij} = 1, R, C, \theta) dz \\ &= \int_{\mathbb{R}} \phi(z; \mu_{R_i C_j}, \sigma^2_{R_i C_j}) \mathbf{1}_{]\gamma_{Y_{ij}-1}, \gamma_{Y_{ij}}[}(z) dz \\ &= \Phi\left(\frac{\gamma_{Y_{ij}} - \mu_{R_i C_j}}{\sigma_{R_i C_j}}\right) - \Phi\left(\frac{\gamma_{Y_{ij}-1} - \mu_{R_i C_j}}{\sigma_{R_i C_j}}\right), \end{aligned}$$

where $\Phi(\cdot)$ denotes the cumulative density function of the Gaussian distribution $\mathcal{N}(0,1)$. If we denote

$$\eta_k^{(q,l)} := \Phi\left(\frac{\gamma_k - \mu_{ql}}{\sigma_{ql}}\right) - \Phi\left(\frac{\gamma_{k-1} - \mu_{ql}}{\sigma_{ql}}\right),$$

it is immediate to verify that $\sum_{k=1}^{K} \eta_k^{(q,l)} = 1$. □

Propostion 1 has two important consequences. First, the posterior density function of Z_{ij} can be obtained by

$$p(Z_{ij}|Y_{ij}, A_{ij} = 1, R, C, \theta) = \frac{\phi(Z_{ij}; \mu_{R_i C_j}, \sigma^2_{R_i C_j})}{\eta_{Y_{ij}}^{(R_i,C_j)}} \mathbf{1}_{]\gamma_{Y_{ij}-1}, \gamma_{Y_{ij}}[}(Z_{ij}). \tag{9}$$

The above probability density function defines a truncated Gaussian distribution and it is fully tractable. Second, due to independence arguments, the marginal likelihood $p(Y|A, R, C, \theta)$ can be computed as

$$p(Y|A, R, C, \theta) = \prod_{i=1}^{M} \prod_{j=1}^{P} \left(\eta_{Y_{ij}}^{(R_i,C_j)}\right)^{A_{ij}}, \tag{10}$$

where we used again that $Y_{ij} = 0$ when $A_{ij} = 0$, a.s.

In the light of these results, it is possible to design an estimation strategy, called C-EM, consisting of the following two steps:

1. **C step.** The model parameters being fixed to a local optimum, $\log p(Y, A, R, C|\theta)$ is maximized with respect to R and C in a greedy fashion. This classification step replaces the Expectation step in the EM algorithm (Dempster et al., 1977). See also Celeux and Govaert (1991) for a description of Classification EM algorithms.

2. **M step.** R and C being fixed, the likelihood

$$\log p(Y, A, R, C|\theta) = \log p(Y|A, R, C, \mu, \sigma^2) + \log p(A, R, C|\pi, \rho, \delta) \tag{11}$$

is maximised with respect to the model parameters θ. As we will see in the next sections, the maximization with respect to (π, ρ, δ) is straightforward. On the contrary, the first term on the right hand side of the above equation (detailed in Eq. 10) cannot be directly maximized with respect to μ and σ^2 and no close formulas can be derived for these Gaussian parameters. Therefore, we will rely on $Q \times L$ independent EM algorithms to maximize this term with respect to (μ, σ^2).

The above two steps are alternatively repeated until convergence of $\log p(Y, A, R, C|\theta)$. Each step is detailed in the following sections.

3.1 C step

Let us focus on the log-likelihood in Eq. (11) and assume that the model parameters are fixed to the (local) optima $\hat{\theta}$, obtained in the M step (see Section 3.2). The goal of the C step is to maximize the left hand side of Eq. (11) with respect to R and C . No closed formula exists for such combinatorial maximization problem and testing all possible combinations $(Q^M L^P)$ would be computational prohibitive. Thus, we rely on a *greedy* search strategy *not* looking for all the possible solutions. Greedy strategies are quite popular in network and bipartite network analysis (see for instance Côme and Latouche, 2015; Wyse et al., 2017). The basic idea is to swap each row (column) of A to the row (column) cluster leading to the highest increase of the log-likelihood $\log p(Y, A, R, C|\hat{\theta})$. Of course, if no swap increases the log-likelihood, the row (column) is not moved.

Assume that the i-th row of A formerly belonging to the cluster q' is now moved to the cluster q''. Furthermore, let us denote by R^* the row label vector R after that the swap occurred. Thus

$$\begin{aligned}\Delta^{i:q'\to q''} &:= \log p(Y, A, R^*, C|\hat{\theta}) - \log p(Y, A, R, C|\hat{\theta}) \\ &= \log \frac{p(Y|A, R^*, C, \hat{\eta})}{p(Y|A, R, C, \hat{\eta})} + \log \frac{p(A|R^*, C, \hat{\pi})}{p(A|R, C, \hat{\pi})} + \log \frac{p(R^*|\hat{\rho})}{p(R|\hat{\rho})}\end{aligned}$$

is the increase (possibly null or negative) of the log-likelihood in Eq. 11 when moving row i from cluster q' to cluster q''. Moreover, we used $\eta := \{\eta_k^{(q,l)}\}_{q,l,k}$, where $\eta_k^{(q,l)}$ is defined in (8). By using Eqs. 1 and 10, we can further obtain

$$\begin{aligned}\Delta^{i:q'\to q''} = &\sum_{j=1}^{P} A_{ij} \log \frac{\hat{\eta}_{Y_{ij}}^{(q'',C_j)}}{\hat{\eta}_{Y_{ij}}^{(q',C_j)}} \\ &+ \sum_{j=1}^{P} \left(A_{ij} \log \frac{\hat{\pi}_{q''C_j}}{\hat{\pi}_{q'C_j}} + (1 - A_{ij}) \log \frac{1 - \hat{\pi}_{q''C_j}}{1 - \hat{\pi}_{q'C_j}} \right) \\ &+ \log \frac{\hat{\rho}_{q''}}{\hat{\rho}_{q'}}.\end{aligned} \tag{12}$$

This quantity can be computed in $\mathcal{O}(P)$ and can be used to rank the possible swaps of the i-th row of A to all row clusters. An equivalent formula can be obtained to assess the contribution of a column swap into a column cluster.

It is important to notice that, since Q and L are fixed, one row (column) alone in its current row (column) cluster is not allowed to move. In case one row (column) remains alone in its group, another criterion (that will be introduced in Section 3.4) will decide whether that group is suppressed or not.

3.2 M step

The label vectors R and C are fixed throughout this section. Notice that the maximization of the right hand side of Eq. (11) with respect to (π, ρ, δ) only involves the second term. Moreover, this maximization is straightforward. Taking the logarithm in Eq. 1, differentiating with respect to π, ρ and δ and setting the derivatives equal to zero leads to the following stationary points

$$\hat{\pi}_{ql} := \frac{\sum_{i=1}^{M}\sum_{j=1}^{P} R_{iq}C_{jl}A_{ij}}{\sum_{i=1}^{M}\sum_{j=1}^{P} R_{iq}C_{jl}}, \tag{13}$$

$$\hat{\rho}_{q} := \frac{\sum_{i=1}^{M} R_{iq}}{M}, \tag{14}$$

$$\hat{\delta}_{l} := \frac{\sum_{j=1}^{P} C_{jl}}{P}, \tag{15}$$

for all q, l.

As anticipated in Section 3.1, the maximization of the first term on the right hand side of Eq. (11) is more challenging. Since no close formula for such optimization does exist, let us consider the following inequality

$$\log p(Y|A, R, C, \mu, \sigma^2) \geq \mathbb{E}_Z \left[\log \frac{p(Y, Z|A, R, C, \mu, \sigma^2)}{p(Z|Y, A, R, C, \mu_0, \sigma_0^2)} \right], \tag{16}$$

where the expectation is taken with respect to Z following the posterior probability density function $p(\cdot|Y, A, R, C, \mu_0, \sigma_0^2)$. The inequality comes from a standard variational decomposition (see for instance Ch.10, Bishop, 2006), it holds for all (μ, σ^2) and it turns into an equality when (μ, σ^2) is equal to (μ_0, σ_0^2). Since the posterior distribution of Z is known and tractable, the EM algorithm can be used to provide numerical estimates of $\hat{\mu}$ and $\hat{\sigma}^2$.

M-Expectation. We now focus on the right hand side of the inequality in Eq. (16). By taking the logarithm of Eq. 5, it holds that

$$\begin{aligned}\mathbb{E}_Z\left[\log p(Y,Z|A,R,C,\mu,\sigma^2)\right] &= -\frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{P} A_{ij}\left(\log\sigma^2_{R_iC_j} + \mathbb{E}_{Z_{ij}}\left[\frac{(Z_{ij}-\mu_{R_iC_j})^2}{\sigma^2_{R_iC_j}}\right]\right) + c \\ &= -\frac{1}{2}\sum_{q=1}^{Q}\sum_{l=1}^{L}\sum_{i=1}^{M}\sum_{j=1}^{P} A_{ij}R_{iq}C_{jl}\left(\frac{m^{(2)}_{ij}+\mu^2_{ql}-2\mu_{ql}m^{(1)}_{ij}}{\sigma^2_{ql}} + \log\sigma^2_{ql}\right) + c,\end{aligned} \tag{17}$$

where c regroups the constant terms not depending on (μ,σ^2). We assumed that $Z_{ij} \in]\gamma_{Y_{ij}-1},\gamma_{Y_{ij}}[$ a.s. for all i,j such that $A_{ij}=1$ and

$$m^{(1)}_{ij} := \mathbb{E}_{Z_{ij}}[Z_{ij}] = \mu_{R_iC_j} - \sigma_{R_iC_j}\frac{\phi(\beta_{ij})-\phi(\alpha_{ij})}{\Phi(\beta_{ij})-\Phi(\alpha_{ij})}, \tag{18}$$

$$\begin{aligned}m^{(2)}_{ij} := \mathbb{E}_{Z_{ij}}[(Z_{ij})^2] &= (\mu_{R_iC_j})^2 - 2\sigma_{R_iC_j}\mu_{R_iC_j}\left(\frac{\phi(\beta_{ij})-\phi(\alpha_{ij})}{\Phi(\beta_{ij})-\Phi(\alpha_{ij})}\right) \\ &\quad - \sigma^2_{R_iC_j}\left[\left(\frac{\beta_{ij}\phi(\beta_{ij})-\alpha_{ij}\phi(\alpha_{ij})}{\Phi(\beta_{ij})-\Phi(\alpha_{ij})}\right)-1\right],\end{aligned} \tag{19}$$

with

$$\alpha_{ij} = \frac{\gamma_{Y_{ij}-1}-\mu_{R_iC_j}}{\sigma_{R_iC_j}}, \qquad \beta_{ij} = \frac{\gamma_{Y_{ij}}-\mu_{R_iC_j}}{\sigma_{R_iC_j}},$$

and $\mathbb{E}_{Z_{ij}}$ is the expectation taken with respect to Z_{ij} following the probability density function in Eq. 9. Notice that both $m^{(1)}_{ij}$ and $m^{(2)}_{ij}$ only depend on the pair (i,j) via R_i, C_j and Y_{ij}. Thus, $m^{(1)}_{ij}$ and $m^{(2)}_{ij}$ are the same for all pairs (i,j) in clusters (q,l), respectively, associated with the score k.

M-Maximization. Once the expectation in Eq. 17 is computed, the right hand side of the equality can be maximized with respect to (μ,σ^2). The maximization of μ_{ql} can be performed independently of σ^2_{ql}, but the opposite is not true. Thus, we first differentiate the right hand side of Eq. 19 with respect to μ_{ql} and set the derivative equal to zero to obtain the following stationary point

$$\hat{\mu}_{ql} := \frac{\sum_{i=1}^{M}\sum_{j=1}^{P} A_{ij}R_{iq}C_{jl}m^{(1)}_{ij}}{\sum_{i=1}^{M}\sum_{j=1}^{P} A_{ij}R_{iq}C_{jl}}, \tag{20}$$

for all q,l. In order to compute the optimal σ^2_{ql}, $\hat{\mu}$ is plugged into Eq. 17 in place of μ and then differentiating with respect to σ^2_{ql} and setting the partial derivative to zero leads to

$$\hat{\sigma}^2_{ql} := \frac{\sum_{i=1}^{M}\sum_{j=1}^{P} A_{ij}R_{iq}C_{jl}m^{(2)}_{ij}}{\sum_{i=1}^{M}\sum_{j=1}^{P} A_{ij}R_{iq}C_{jl}} - \hat{\mu}^2_{ql}, \tag{21}$$

for all q, l.

The two steps of the EM algorithm described so far are part of the M step. Therefore they are called M-Expectation (Eqs. 18-19) and M-Maximization (Eqs. 20-21). They are alternatively applied up to convergence. We stress that all the equations involving M-Expectation and M-Maximization factorized over q and l. Thus, $Q \times L$ independent EM algorithms are used, one for each pair (q, l) of clusters and this task can be done in parallel.

3.3 Initialization

Assuming that Q and L are momentarily fixed, the C-EM algorithm described in the previous sections needs some initial values of R and C to be provided. Then, a first M step can be implemented, followed by a greedy search C step and so on. In this paper two different initialization strategies are considered:

1. **Multiple random initializations.** Both the initial R and C are independently sampled from multinomial distributions with uniform parameters. Since the C-EM algorithm is not guaranteed to converge toward a global optimum, the algorithm is provided in the applications with several independent initializations. The estimates $\hat{R}$ and $\hat{C}$ leading to the highest log-likelihood are finally retained. This initialization strategy is assessed in Section 4.4.

2. **K-means initialization.** Two k-means algorithms are independently run on the rows and the columns of the matrix Y. The C-EM algorithm is then initialised with the estimates $\hat{R}$ and $\hat{C}$ provided by the two k-means.

Notice that missing values could be present in Y. This is not a problem when adopting random initializations, but it can be one when using k-means. Indeed, when the proportion of missing data (i.e. zeros) in Y is very large, the k-means algorithm will provide very poor initial estimates of R and C. In a similar scenario, it is preferable to opt for multiple random initializations.

The pseudocode in Algorithm 1 summarizes the estimation routine detailed so far.

Algorithm 1 Pseudocode

function ESTIM(Y,A,Q,L, type)
  $(R, C) \leftarrow$ INIT(Y, type) ▷ type is "multiple random" or "k-means"
  while $\log p(Y, A, R, C|\theta)$ increases **do**
    $\theta \leftarrow$ M step ▷ Including $Q \times L$ M-EM algorithms
    $(R, C) \leftarrow$ C step
  end while
  return $(\hat{R}, \hat{C}, \hat{\theta})$
end function

3.4 Model selection

So far, the numbers Q and L of row and column clusters were assumed to be known. Of course, in real applications, this assumption is too restrictive. Thus, we now detail a model selection criterion we propose to select the numbers of row and column clusters.

In clustering contexts, the integrated classification likelihood (ICL, Biernacki et al., 2003) criterion is often used to approximate a complete data integrated log-likelihood and to select the number of components. In our case

$$ICL(Q, L) \approx \log p(Y, A, R, C) = \int_{\mathcal{D}_\theta} \log p(Y, A, R, C|\theta)\nu(\theta)d\theta,$$

where, in a Bayesian framework, the model parameters θ are seen as random variables and the integral is taken over the support $\mathcal{D}_\theta$ of any prior probability density function $\nu(\cdot)$. The following proposition details the functional form of the ICL for our model.

Proposition 2. *An $ICL(Q, L)$ criterion for the generative model described in Section 2 is*

$$ICL(Q, L) = \log p(Y, A, \hat{R}, \hat{C}|\hat{\theta}) - QL \log D - \frac{QL}{2} \log(MP) - \frac{Q-1}{2} \log M - \frac{L-1}{2} \log P, \tag{22}$$

where $\hat{\theta}, \hat{R}, \hat{C}$ are the stationary points obtained after convergence of the algorithm described in Sections 3.2-3.1.

Proof. By definition of conditional probability density function, it follows that

$$ICL(Q, L) \approx \log p(Y|A, R, C) + \log p(A, R, C).$$

The last term on the right hand side is the complete data integrated log-likelihood of a binary LBM. This log-likelihood can be approximated as follows

$$\log p(A, R, C) \approx \max_{\pi,\rho,\delta} \log p(A, R, C|\pi, \rho, \delta) - \frac{QL}{2} \log(MP) - \frac{Q-1}{2} \log M - \frac{L-1}{2} \log P, \tag{23}$$

see for instance Keribin et al. (2012). In order to approximate $\log p(Y|A, R, C)$ we propose a BIC-like approximation (Schwarz et al., 1978)

$$\log p(Y|A, R, C) \approx \max_{\mu,\sigma^2} \log p(Y|A, R, C, \mu, \sigma^2) - QL \log D, \tag{24}$$

where D is the number of ordinal (i.e. not zero) entries in Y and $2QL$ accounts for the number of parameters in μ and σ^2. Combining Eqs. 23-24, the proposition is proven. □

When fitting to data the model presented in Section 2, the ICL criterion in Eq. (22) is computed for several values of Q and L. The pair $(\hat{Q}, \hat{L})$ leading to the highest value of $ICL(Q, L)$ is finally retained. An exhaustive strategy would consist into fixing some sufficiently high values Q_{max} and L_{max} and computing $ICL(Q, L)$ for all (Q, L) in the grid

$$\{1, \ldots, Q_{max}\} \times \{1, \ldots, L_{max}\}.$$

Notice that, as long as either $\hat{Q}$ or $\hat{L}$ lie on the boundary of the grid, Q_{max} and/or L_{max} can be increased to obtain a solution which is interior. In this sense, the grid search described so far is exhaustive.

However, when dealing with massive datasets, this strategy could be computationally prohibitive since the number of elements in the grid is $Q \times L$. Alternatively, a greedy search algorithm can be employed to select Q and L. The ICL criterion is still used, but it is not computed for all values of Q and L in the grid. We propose here a greedy search algorithm inspired by the one introduced in Keribin et al. (2017). Initially, both Q and L are set to 1. In a second step, Q is increased by one and the value of $ICL(2, 1)$ is computed via the C-EM algorithm and recorded. Then, $ICL(2, 1)$ is compared with $ICL(1, 2)$, obtained by setting $Q = 1$ and $L = 2$. Thus, if

$$ICL(2, 1) > ICL(1, 2)$$

Q is definitely set to 2 and L to 1. The opposite otherwise. This routine is recursively applied in such a way that, if Q^* and L^* are the *current* values of Q and L, then $ICL(Q^* + 1, L^*)$ is compared with $ICL(Q^*, L^* + 1)$ and Q^* and L^* are updated accordingly. The algorithm stops when no further increase in the ICL criterion is possible. A pseudo-code illustrating the greedy search detailed so far is reported in Appendix A.1.

4 Experiments on synthetic data

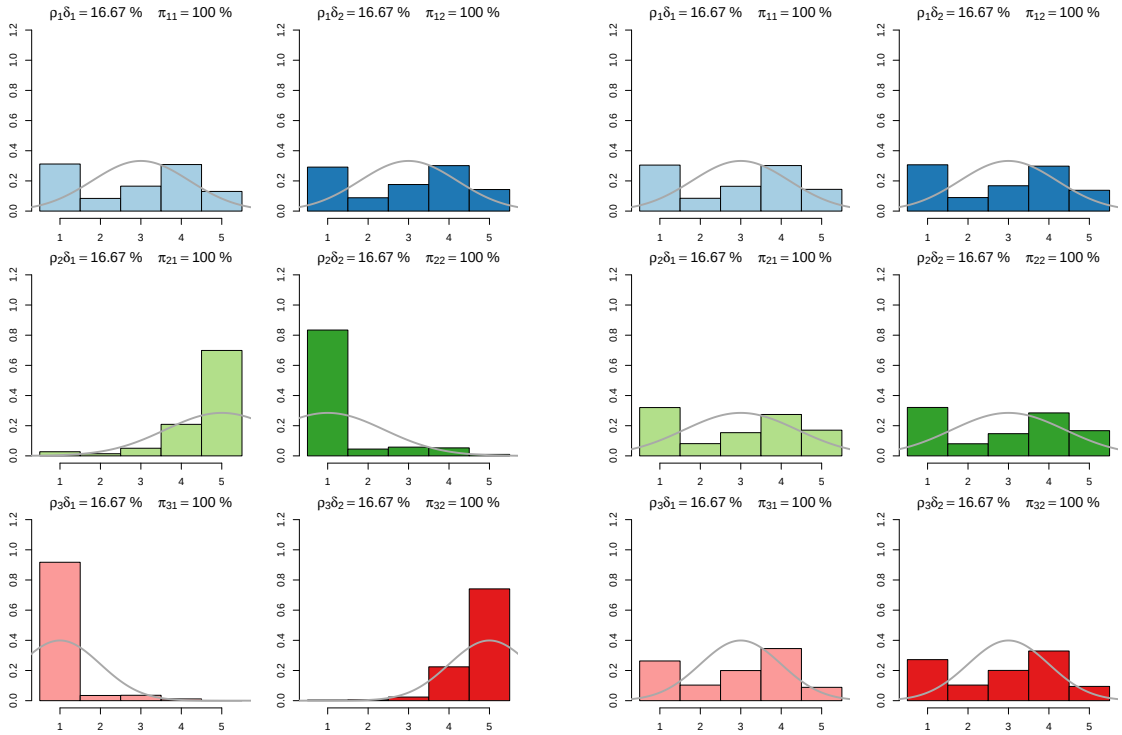
The estimation procedure detailed in Section 3 is now tested on simulated data. Two main scenarios are considered: the former adopting the generative model described in Section 2 and the latter adopting the generative model described in Jacques and Biernacki (2018). Henceforth, these two generative models will be referred to as OLBM (Ordinal Latent Block Model, our proposal) Co-Clustering and BOS (Binary Ordinal Search) Co-Clustering. For each simulated scenario, both OLBM-CC and BOS-CC are fitted to the data to provide estimates of R and C, and the results are compared. The parameters of BOS-CC are estimated via the R package **ordinalClust**[1]. A third competitor approach is considered in this section, namely LBM for continuous data (cLBM) as described in Bhatia et al. (2017). This model is fitted to the data via the R package **blockcluster**[2]. Unfortunately, this package does not support missing data in Y. Therefore, comparisons will not always be possible.

4.1 Data simulated according to OLBM-CC

For this first experiment, the data are simulated according to the generative model described in Section 2. We consider incidence matrices with $M = 150$ rows, grouped into $Q = 3$ row clusters and $P = 100$ columns, grouped into $L = 2$ column clusters. Rows and columns are randomly assigned to their clusters in uniform proportions. The observed ordinal entries in Y take values in $\{1, \dots, K\}$ and $K = 5$. Recalling that $\Phi(\cdot)$ denotes the standard Gaussian cumulative distribution function, the thresholds $\gamma_1, \dots, \gamma_K$ are set as follows. We sample

[1] `https://cran.r-project.org/web/packages/ordinalClust/index.html`

[2] `https://cran.r-project.org/web/packages/blockcluster/index.html`



(a) $\zeta = 2$. (b) $\zeta = 0$.

Figure 1: Histograms of the ordinal entries of Y organized by block pairs. On the left hand side figure, a high value of ζ induces asymmetries in both the histograms and the underlying Gaussian distributions. See in particular the block pairs $(2,1)$,$(2,2)$ and $(3,1)$,$(3,2)$. On the right hand side figure, $\zeta = 0$ and the two columns are indistinguishable.

$U_1, \ldots, U_{K-1}$ independent random variables, uniformly distributed in $[0,1]$. Without loss of of generality, let us assumed that they are sorted. A last variable $U_K = 1$ is introduced. Then:

$$\gamma_i := \frac{K+1}{2} + \Phi^{-1}(U_i),$$

for all $i \in \{1, \ldots, 5\}$ and $\gamma_0 = -\infty$. Then, the Gaussian parameters μ and σ^2 are set to

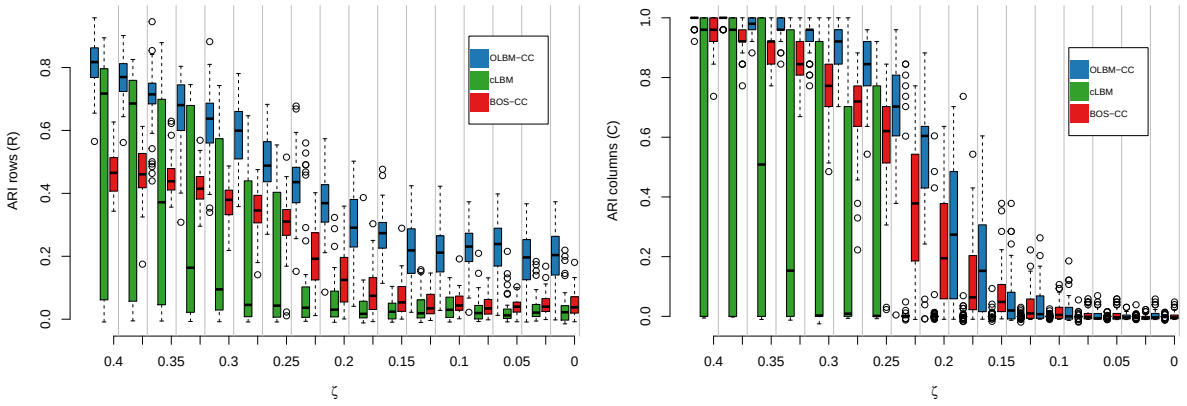
$$\mu = \begin{pmatrix} 0 & 0 \\ \zeta & -\zeta \\ -\zeta & \zeta \end{pmatrix}, \qquad \sigma^2 = \begin{pmatrix} 1.2 & 1.2 \\ 1.4 & 1.4 \\ 1.0 & 1.0 \end{pmatrix},$$

where $\zeta \geq 0$ is a real parameter controlling how the block distributions differ: as long as ζ is far enough from 0, we expect that the estimation algorithms would correctly estimate

the row and the column clusters. When ζ approaches to 0, the column clusters become indistinguishable and the row groups are only separated via the matrix σ^2 (see Figure 1).

The last parameter to set is π, defining the probability of observing an ordinal entry in Y. Three different setups are considered.

No missing data. In this framework, $\pi_{ql} = 1$, for all q, l and only the Gaussian parameters (μ, σ^2) induce a block structure. For each value of ζ, fifty datasets Y are independently simulated according to the setup described so far and the three approaches (OLBM-CC, cLBM, BOS-CC) are fitted to each dataset. At first, the selection of Q and L is not consid-



(a) Adjusted Rand Indexes (rows).

(b) Adjusted Rand Indexes (cols).

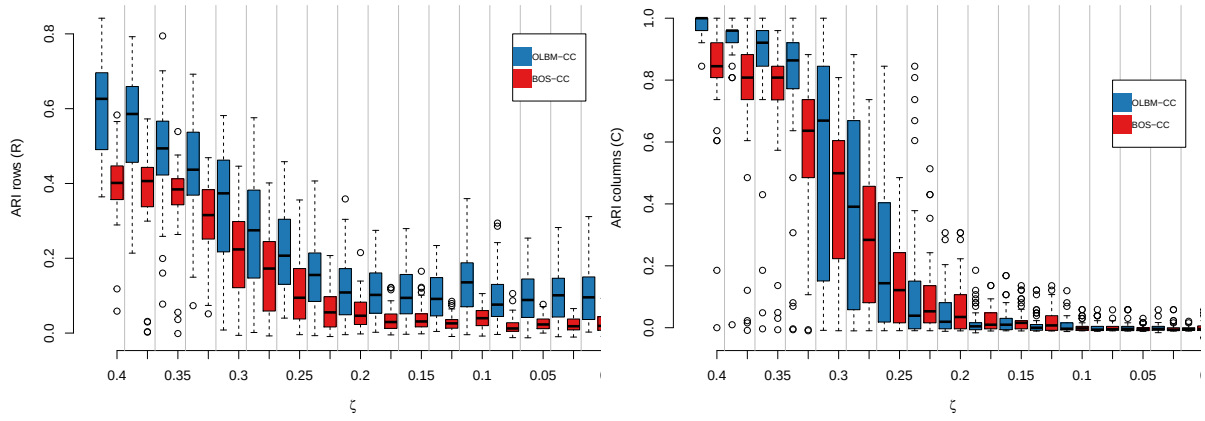
Figure 2: Results on the dataset simulated according to OLBM-CC - Not missing data.

ered and the three estimation algorithms are provided with the *actual* values of $Q = 3$ and $L = 2$. OLBM-CC is initialized through a k-means initialization (see Section 3.3) whereas the standard initialization in the **blockcluster** and **ordinalClust** packages is adopted for cLBM and BOS-CC, respectively. For each simulated data matrix Y, the estimates provided by the three methods are assessed via the adjusted Rand index (ARI, Rand, 1971). This metric compares the estimated label vectors $\hat{R}$ and $\hat{C}$ with their actual counterparts R and C. The ARI takes real values in $[0, 1]$, where 0 means that the obtained clustering is poor (as good as a random assignment to each class) and 1 means perfect recovery, up to label switching.

The results of the experiment are reported in Figure 2, where boxplots of the ARIs

can be observed in two sub-figures. The one on the left hand side reports the ARIs for the row label estimates ($\hat{R}$). Blue bars refer to OLBM-CC, green bars to cLBM and red bars to BOS-CC. Not surprisingly, as ζ decreases, the performance of the three methods deteriorates. However, the structure of σ^2 still slightly discriminates the three row clusters, thus allowing OLBM-CC to reach a median ARI around 0.3, even when ζ is null. In these simulations, OLBM-CC clearly outperforms its competitors. The right hand side of Figure 2 confirms the intuition raised by Figure 1: when ζ approaches to zero, the column clusters become indistinguishable. However, also in this case, OBLM-CC outperforms its competitors.

Missing at random data. In this framework, the probability of a missing data in Y is independent of the cluster assignments. In other words, it is the same for each entry Y_{ij}. We set $\pi_{ql} = 0.7$, for all q, l (30% of missing data in Y, on average) and repeat the experiment. As previously mentioned, the blockcluster package does not support missing data. Thus, in the reminder of this section, OLBM-CC will only be compared with BOS-CC.



(a) Adjusted Rand Indexes (rows).

(b) Adjusted Rand Indexes (cols).

Figure 3: Results on the dataset simulated according to OLBM-CC - Missing at random data.

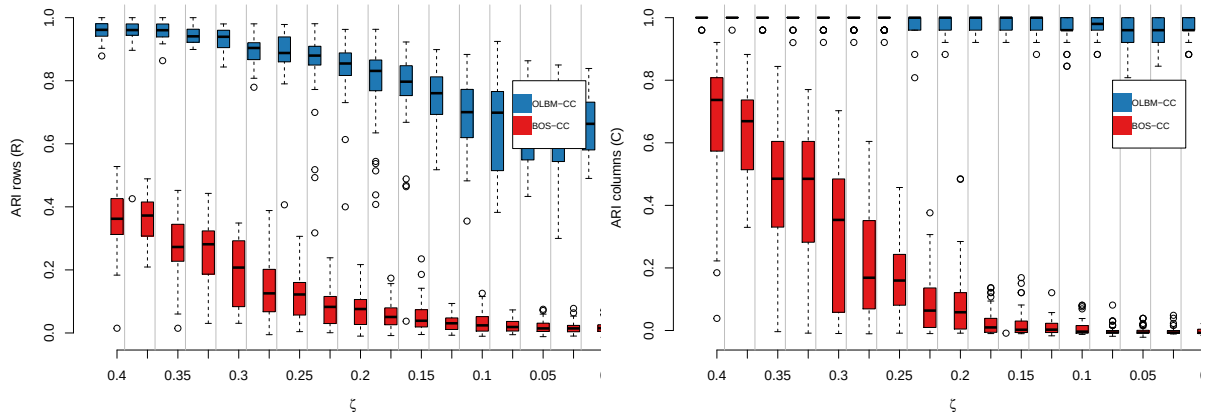
The results can be observed in Figure 3, where boxplots of the ARIs can be observed in two sub-figures. The one on the left hand side reports the ARIs for the row labels estimates. The ARIs for the column labels estimates are reported in Figure 3b. These

results are coherent with the ones in Figure 2. Since missing at random data does not bring any information, this experiment reduces to the previous one with *less* ordinal data. This explains the slightly worse performance of OLBM-CC and BOS-CC in Figure 3 with respect to Figure 2.

Not missing at random data. In the reminder of this section, the assumption of missing at random data is relaxed. An alternative setup can be obtained by adopting the following connectivity matrix

$$\pi = \begin{pmatrix} 0.5 & 0.7 \\ 0.7 & 0.5 \\ 0.7 & 0.5 \end{pmatrix},$$

meaning that the probability of a missing data in Y is no longer independent on the cluster assignments. It is lower (30%) in cluster pairs $(1,2),(2,1)$ and $(3,1)$ and higher (50%) for the remaining block pairs. Other settings being unchanged, the experiment is repeated (50 simulated data matrices Y for each value of ζ and Q, L known). Results can be seen in Figure 4. As expected, with respect to the missing at random setup, the estimates obtained



(a) Adjusted Rand Indexes (rows). (b) Adjusted Rand Indexes (cols).

Figure 4: Results on the dataset simulated according to OLBM-CC - Not missing at random data.

via OLBM-CC (blue bars) are more accurate (higher ARIs) for both rows and columns. In particular, the average column ARI is around 1 even when $\zeta = 0$. Indeed, the π matrix itself

induces an additional block structure that discriminates the two column clusters although $\zeta = 0$. Such a framework, in which the block structure is accentuated by the connectivity patterns, is very common in real data (see also Section 5). In contrast with OLBM-CC, BOS-CC has very similar performances with or without missing at random data. Since it cannot deal with block dependent missing data, the additional information carried by π cannot be exploited by the model.

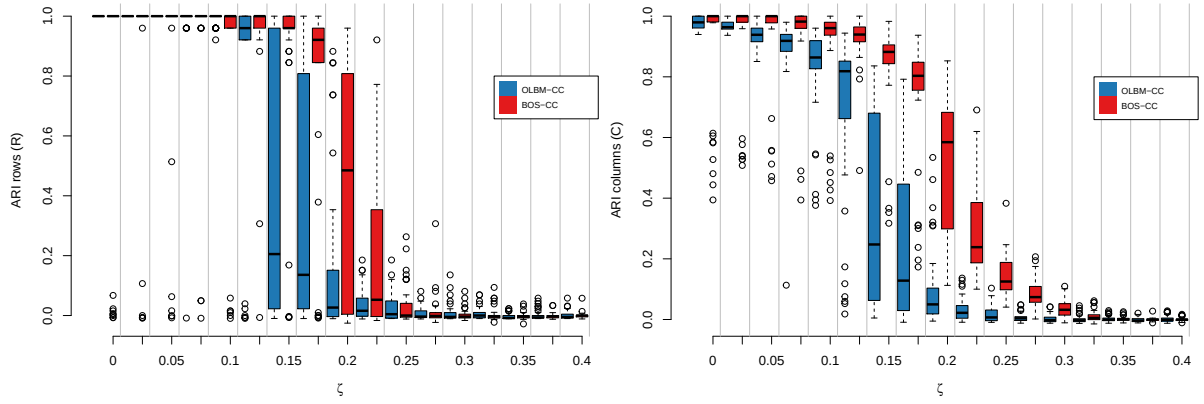
4.2 Data simulated according to BOS-CC

In order to present fair results regarding methods other than ours, the datasets are now generated according to the BOS-CC model. The reader is referred to Jacques and Biernacki (2018) for a full description of the model. The simulated incidence matrices have now $M = 100$ rows and $P = 150$ columns. The rows are clustered in $Q = 2$ groups and the columns in $L = 3$ groups. The model parameters are

$$\mu = \begin{pmatrix} 2 & 3 & 1 \\ 2 & 1 & 3 \end{pmatrix}, \qquad \varrho = \begin{pmatrix} (0.4-\zeta) & (0.4-\zeta) & (0.4-\zeta) \\ (0.4-\zeta) & (0.4-\zeta) & (0.4-\zeta) \end{pmatrix},$$

where, as long as $\varrho_{ql} \neq 0$, μ_{ql} can be seen as the *mode* of the ordinal entries associated with the pair (q, l). The parameter ϱ_{ql} measures the dispersion around the mode, which is minimal when $\varrho_{ql} = 1$ and maximal when $\varrho_{ql} = 0$ (in this case the ordinal entries of the pair (q, l) are uniformly distributed in $\{1, \ldots, K\}$). As in the previous section, ζ is a real parameter which controls how different the block distributions are. Here, however, $\zeta \in [0, 0.4]$ and the contrast is maximum when $\zeta = 0$ whereas the row and column clusters are indistinguishable when $\zeta = 0.4$. Missing values are injected into Y in two different ways.

Missing at random data. For some values of ζ in $[0, 4]$ fifty data matrices Y are independently sampled according to the setup detailed so far. Then, 30% of the entries of each matrix Y is randomly selected and replaced by missing values. The true values of Q and L are assumed to be known and both OLBM-CC and BOS-CC are fitted to each dataset. Results can be seen in Figure 5. Not surprisingly, BOS-CC globally provides



(a) Adjusted Rand Indexes (rows). (b) Adjusted Rand Indexes (cols).

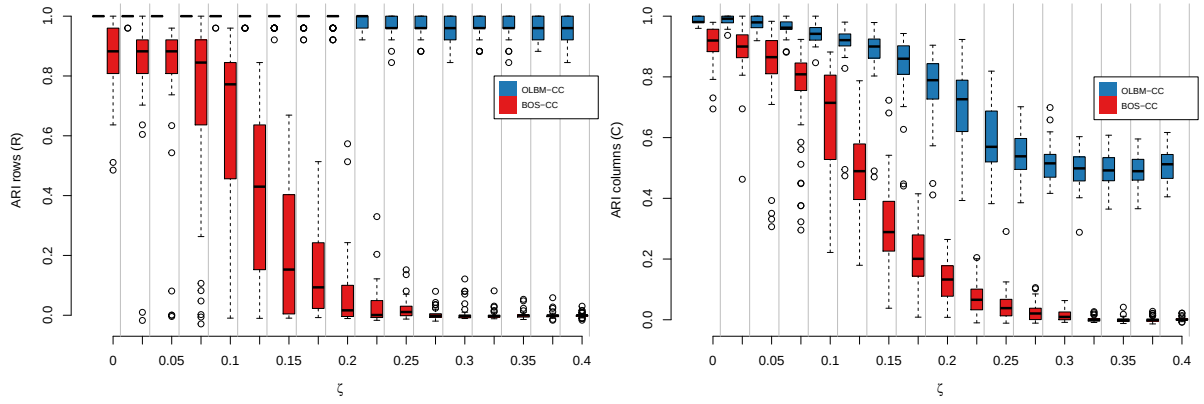
Figure 5: Results on the dataset simulated according to BOS-CC - Missing at random data.

better estimates than OLBM-CC in this scenario. Notice that, when ζ is small (< 0.05), BOS-CC produces outlier ARIs both for rows and column estimates. The same does not happen to OLBM-CC. However, let us recall that the estimation procedures adopted for the two models are very different. For instance, increasing the burn-in step and/or the number of EM iterations in the stochastic EM for BOS-CC could reduce the variance of BOS-CC results.

Not missing at random data. In this framework, once an ordinal data matrix Y is sampled, missing data are no longer uniformly injected into Y. As in the previous section, a connectivity matrix π can be introduced to model block pair missing data

$$\pi = \begin{pmatrix} 0.7 & 0.5 & 0.5 \\ 0.5 & 0.7 & 0.7 \end{pmatrix}.$$

As it can be seen, 50% of the ordinal entries corresponding to the block pairs $(2,1)$, $(2,2)$ and $(3,2)$ are randomly replaced by 0 according to π. Instead, only 30% of the ordinal entries corresponding to the remaining block pairs are replaced by 0. Other settings being unchanged, the experiment is repeated and results can be seen in Figure 6. As in the previous section, OLBM-CC exploits the information carried by π to produce better estimates of R and C, whereas BOS-CC performs slightly worse than in the missing at random framework. Notice that, when ζ approaches to zero, the only parameter allowing



(a) Adjusted Rand Indexes (rows).

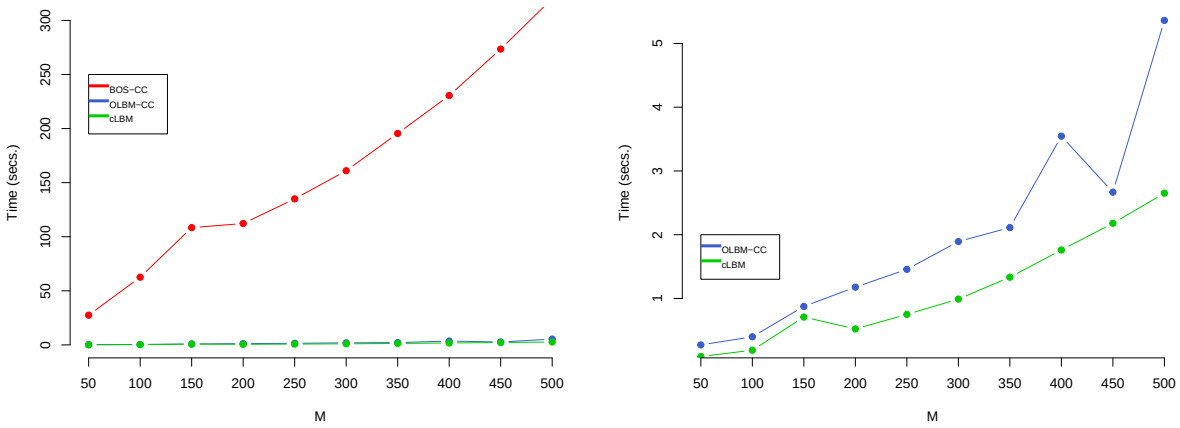
(b) Adjusted Rand Indexes (cols).

Figure 6: Results on the dataset simulated according to BOS-CC - Not missing at random data.

to discriminate clusters is π. However, while the row clusters are perfectly discriminated, the second and the third column clusters are indistinguishable when looking at π. This explains why the median columns ARI for OLBM-CC is around 0.6 on the right hand side of Figure 6.

4.3 Scalability

When dealing with massive datasets, the question of the scalability of the algorithm is of course of great interest. A deep understanding of the computational complexity of the C-EM algorithm (see Sections 3.2 and 3.1) is outside the scope of this paper. Nonetheless, this section aims at providing some insights about the scalability of the OLBM-CC estimation algorithm. In particular, we aim at assessing how the algorithm behaves when either the number of rows/columns of Y or the number of ordinal levels K increase. At first, M is set equal to P and both vary between 50 and 500. Data are missing at random, in such a way that the (mean) number of ordinal entries in Y is equal to $10 \times M$. Then, the data are simulated according to the OLBM-CC generative model and all the remaining parameters $(Q, L, K, \mu, \sigma^2, \gamma)$ are as in Section 4.1. For each simulated matrix Y, the three estimation algorithms considered so far (OLBM-CC, cLBM, BOS-CC) are provided with



(a) Running times.

(b) Running times (without BOS-CC).

Figure 7: Figure 7a reports the running times of the three competitor algorithms versus the number of rows of Y, with $(M = P)$. Figure 7b zooms on the running times of OLBM-CC and cLBM.

the true values of Q and L and fitted to the data. Their running times for each value of M (equal to P) are recorded and reported in Figure 7a. Since the stochastic EM algorithm for BOS-CC is much slower than its competitors, Figure 7b only focuses on the running times of OLBM-CC and cLBM, showing that the estimation algorithm of cLBM is slightly faster.

Figure 8 highlights another feature of the C-EM estimation algorithm: its scalability with respect to number of ordinal levels K. The previous experiment was repeated with $M = P = 100$, but now K ranges is $\{3, \ldots, 13\}$. For each value of K, 10 OLBM-CC estimation algorithms are independently run. In Figure 8, the average running times are plotted versus the number of ordinal levels K. As it can be seen, the computing time of OLBM-CC does not seem to be dependent on the number of ordinal levels.

4.4 Initialization

In the previous experiments, for each simulated matrix Y, the OLBM-CC estimation algorithm was provided with a single k-means initialization. This section aims at comparing

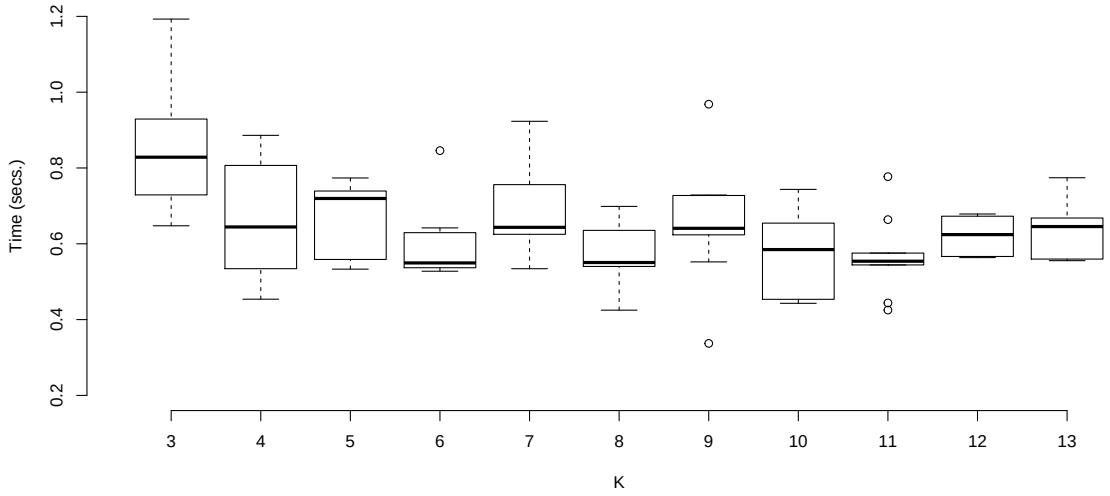
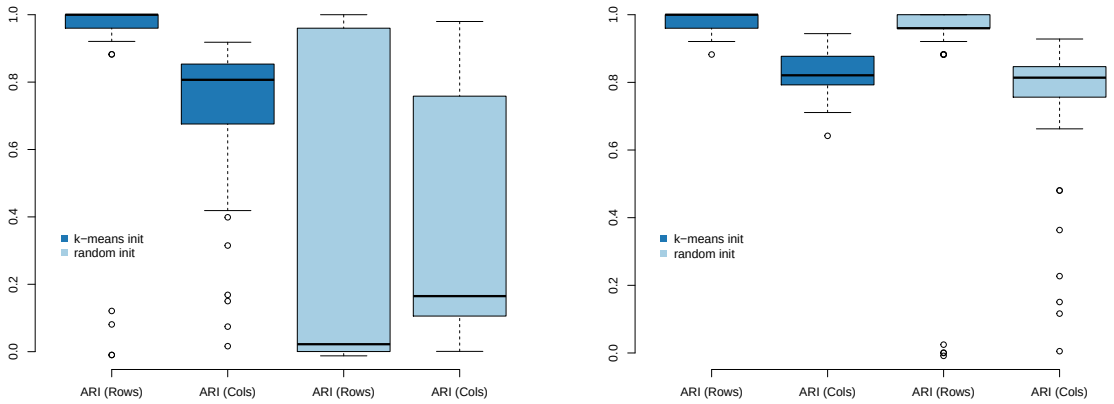


Figure 8: The number K of ordinal levels (on the horizontal axis) varies between 3 and 13. The running times of the C-EM estimation algorithm for OLBM-CC are computed and box-plotted (10 runs for each K).

k-means and random initializations (see Section 3.3). Here, the data is simulated according to the BOS-CC generative model described in Section 4.2, with $\zeta = 0.125$ and missing at random values. As in can be seen in Figure 5, our approach (blue bars) works quite well when $\zeta = 0.125$. However, some outliers can be observed both in row and column ARIs. Thus, 50 matrices Y are independently sampled according to the setup described in Section 4.2 and the OLMB-CC estimation algorithm is run on each matrix, provided with two different initializations: a k-means initialization and a purely random one. Not surprisingly, as in can be seen in Figure 9a, both the row and the column label estimates are more accurate when a k-means initialization is provided. The experiment is now repeated but the multiple random initializations detailed in Section 3.3 is adopted in place of a single k-means. More in details, 10 independent random initializations are used for each dataset. The results can be seen in Figure 9b. Two remarks can be made: first, 10 random initializations are enough to sensibly reduce the gap between the multiple random initializations and k-means initializations; second, the outliers ARIs in Figure 5 are no more present, due



(a) One initialization provided.

(b) Ten initializations provided.

Figure 9: Fifty data matrices Y are simulated according to the BOS-CC generative model, with $\zeta = 0.125$. The OLBM-CC estimation algorithm is run on each dataset provided with one k- means initialization and one random initialization. Boxplots of the resulting ARIs are plotted in Figure 9a. Figure 9b reports the results of the same experiment but the number of provided initializations of each type is 10.

to the increased number of k-means initializations. We stress that Figures 9a and 9b refer to the very same simulated dataset.

4.5 Model selection

So far, the numbers Q of row clusters and L of column clusters were assumed to be known. However, in real applications, the pair (Q, L) needs to be estimated from the data. This can be done via the ICL criterion in Eq. (22). In order to assess the criterion, the BOS-CC generative model described in Section 4.2 is employed to simulate 50 data matrices Y, with missing at random data, in two different scenarios. The former (easier scenario) is obtained by setting $\zeta = 0$, the latter (harder scenario) is adopted by setting $\zeta = 0.125$. The OLBM-CC estimation algorithm is run on each Y for different values of $(Q, L) \in \{1, \ldots, 6\}^2$, thus leading to 36 models to test for each simulated dataset. For each value of (Q, L), the algorithm is initialised via a k-means (one initialization). The results of the easier scenario can be observed in Table 1. In bold, the number of times the true values of Q and L are

Q/L	1	2	3	4	5	6
1	0	0	0	0	0	0
2	0	0	**49**	1	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0

Table 1: *Easier* setup, 50 simulated datasets. In bold, the number of times the actual values of Q and L are recovered by ICL.

Q/L	1	2	3	4	5	6
1	0	0	0	0	0	0
2	0	3	**40**	6	0	0
3	0	1	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0

Table 2: *Harder* setup, 50 simulated datasets. In bold, the number of times the actual values of Q and L are recovered by ICL.

correctly estimated by the ICL criterion. The criterion succeeds 49 times over 50 and only fails once, by selecting $L = 4$. In the harder scenario (Table 2), the number of times ICL correctly estimates Q and L is lower and it is not surprising. As it can be seen in Eq. 22, the estimated complete data log-likelihood plays a central role in the computation of ICL. Thus, a less accurate estimate of R and/or C leads to a lower value of the log-likelihood and hence of the ICL.

5 Amazon fine foods

This section focuses on a real dataset consisting of reviews of fine foods from Amazon. The dataset can be freely downloaded at `https://snap.stanford.edu/data/web-FineFoods.html`. A time horizon of 10 years is considered, up to October 2012. The number of reviews reported is 568,464 and in the original dataset, each row corresponds to one review. Some additional information is reported for each review: the user/product numerical identifiers, a summary of the review and a rating attributed to the product by the user. The rating is expressed via an integer number spanning from 1 (very bad) to 5 (very good). To focus on the most meaningful part of the data, we only considered the users reviewing more than

20 times and the products being reviewed more than 50 times.

By doing that, an ordinal data matrix Y with $M = 1,644$ rows and $P = 1,733$ columns was obtained by neglecting all the information but the ratings. The entry Y_{ij} was either an ordinal entry (a score) or a missing value. The number of observed ordinal entries in Y (the scores) is $32,836$, corresponding to 98.85% of missing data. The score frequencies are reported are reported in Table 3.

Scores	1	2	3	4	5
Frequencies	1849	2126	4174	7912	16775

Table 3: The score frequencies in Y for the Amazon fine food data.

Due to the dimensions of Y, adopting a grid search to select Q and L via the ICL criterion (see Section 3.4) would be very long. Thus, we opted for the greedy search scheme described in Section 3.4. Moreover, as pointed out in Section 3.3, a k-means initialization is useless when the rows/columns of Y contain a majority of missing values and here it is the case. Therefore, for each value of (Q, L), the C-EM algorithm was initialized with multiple (25 times) random initializations. The highest ICL criterion (for each pair) was finally retained. The co-clustering of Y provided by our method can be observed in Figures 10 and 11.

Figure 10 reports the reorganised incidence matrix A. Darker regions correspond to lower portions of missing data in the corresponding co-clusters. As in can be seen, the ICL criterion selected $Q = L$ row clusters and $L = 6$ column clusters. The way the scores are assigned on each co-cluster can be assessed by looking at Figure 11. The score frequencies of each co-cluster are plotted as histograms and, in grey, one can see the estimated underlying Gaussian distributions. On the top of each histogram, the estimated parameters $\hat{\rho}, \hat{\delta}$ and $\hat{\pi}$ are reported (without hat, to keep the plot uncluttered). We present hereafter some (not exhaustive) remarks about the results.

1. As it can be seen by looking at both Figures 10 and 11, the users in row cluster $q = 4$ have a peculiar behaviour, both in terms of missing values and score assignments. They assign most of the scores to the products in column cluster $l = 5$ and they

do not review goods in clusters $l = 2$ and $l = 3$. Notice also that the co-cluster $(q = 4, l = 5)$ is the only one *not* containing missing values ($\hat{\pi}_{45} = 1$). The products in column cluster $l = 5$ are all herbal teas of the same brand "alvita" and the most common rating is 5 (over 60% of scores). When taking a look to the texts associated with the scores on the Amazon website, we noticed that most texts are similar to each other. One might think that users in cluster $q = 4$ are paid to review.

2. Still on column cluster $l = 5$. As it can be seen in Figure 11, products in this cluster are only rated by users in row clusters $q = 2$, $q = 4$ and $q = 6$. The score distributions are very different from one row cluster to another. Users in row cluster $q = 6$ only rate 2 and the underlying Gaussian distribution is peaked in 2. Users in cluster $q = 2$ all note 1, but in that case the mode of the underlying Gaussian distribution is slightly shifted toward 0 with a higher variance. It reflects the fact that 1 is the worst note that one user can assign (some users would even rate worse if they could).

3. Products in column cluster $l = 1$ are a mix of food and beverages, including (e.g.) cat food, teas, coffees and chips. Together with products in cluster $l = 4$ they are scored

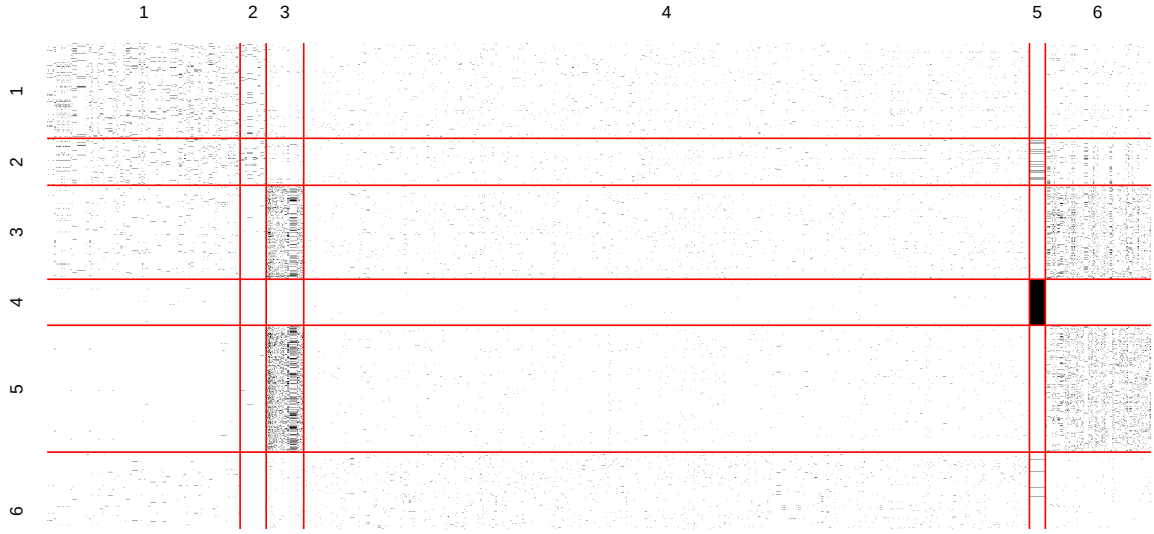


Figure 10: The incidence matrix A of the Amazon fine food notes reorganised according to the estimates $\hat{R}$ and $\hat{C}$, provided by the C-EM algorithm. The ICL criterion selected $Q = 6$ row clusters and $L = 6$ column clusters.

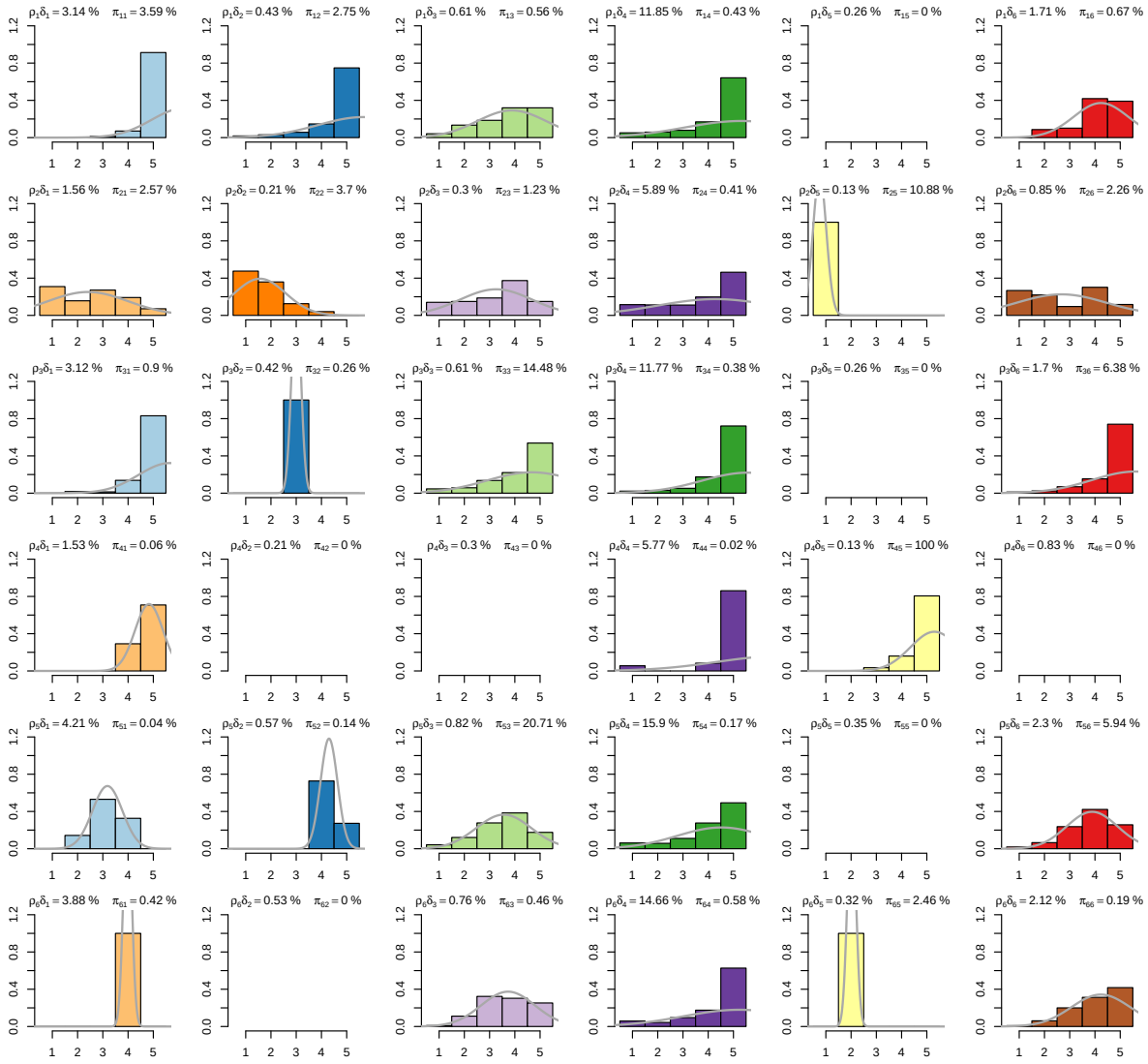


Figure 11: Histograms of the Amazon fine foods ratings. One histogram corresponds to a co-cluster. In grey, one can see the estimated underlying Gaussian distributions. On the top of each histogram, the corresponding *estimated* ρ_q, δ_l and π_{ql} are reported.

by users of all row clusters. However, users seem to be more satisfied by the products $l = 4$ than by those $l = 1$. It can clearly be seen in Figure 11 for row clusters $q = 2$, $q = 5$ and $q = 6$. Indeed, the products $l = 4$ are the best rated in the dataset: the mode is 5 in all row clusters.

4. In Figure 11 we see that the note distributions in column cluster $l = 2$ have very

different shapes from one row cluster to another: very positive ratings for $q = 1$, negative ratings for $q = 2$, neutral ratings for $q = 3$ and positive but (on average) not excellent ratings for $q = 5$. The column cluster $q = 2$ is certainly the one exhibiting the highest variety of preferences. Notice also that, the amount of missing data is higher in cluster pairs $(q = 3, l = 2)$ and $(q = 5, l = 2)$ than in cluster pairs $(q = 1, l = 2)$ and $(q = 2, l = 2)$.

5. In Figure 10, row clusters $q = 3$ and $q = 5$ look very similar (except for column cluster $l = 1$, where users $q = 3$ note more frequently than those $q = 5$). However, when looking at Figure 11, we see that the way they note is quite different. Users $q = 5$ are more demanding than users $q = 3$ on goods $l = 1$ and less demanding on goods $l = 3$. More important, on column cluster $l = 3$, the one where they both are more active ($\hat{\pi}_{33} = 14.48\%$ and $\hat{\pi}_{53} = 20.71\%$), users $q = 3$ are more enthusiastic than users $q = 5$.

6. Users in row cluster $q = 2$ are definitely the more demanding ones: the means of their underlying Gaussian distributions are constantly the leftmost ones. For instance, products in column cluster $l = 6$ are globally well rated except for users in $q = 2$.

Similar analyses can be done for the remaining blocks. This experiment demonstrated that OLBM-CC can be fitted to large and very sparse datasets to provide a synthetic and comprehensive view.

6 Conclusion and perspectives

A new method for the co-clustering of ordinal data has been introduced in this paper. This method relies on the binary LBM to manage data sparsity and adopts latent Gaussian random variables to generate ordinal entries in a data matrix. In our view, the reduced computational burden of the estimation procedure, the modeling of missing data and the easy interpretation of the latent distributions are the main advantages of the outlined approach.

Hereafter, we suggest two topics that could be taken into account for future researches. First, it could be useful to assess the advantages/disadvantages of using *not* Gaussian latent random variables to model ordinal data. Indeed, other distributions could be employed to capture some features in the data, e.g. asymmetric frequencies. Second, Section 3.4 describes a greedy search algorithm to select the number of row/column clusters when an exhaustive grid search is computationally prohibitive. Alternative greedy schemes could certainly be implemented, for instance based on the genetic algorithms described in Scrucca (2016).

A Appendix

A.1 Greedy model selection

Algorithm 2 Pseudocode

function MODSEL(Y)
  Initialization: $Q^* = 1$, $L^* = 1$ and GoOn = TRUE
  while GoOn **do**
    Calculate $T_1 = ICL(Q^* + 1, L^*)$ ▷ Call to function ESTIM($\cdot$)
    Calculate $T_2 = ICL(Q^*, L^* + 1)$
    if $T_1 < T_2$ **then**
      $Q^* = Q^* + 1$
    else if $T_2 \leq T_1$ **then**
      $L^* = L^* + 1$
    else
      GoOn = FALSE
    end if
  end while
  return (Q^*, L^*)
end function

References

Agresti, A. (2010). *Analysis of ordinal categorical data*, volume 656. John Wiley & Sons.

Bhatia, P. S., Iovleff, S., Govaert, G., et al. (2017). blockcluster: An r package for model-based co-clustering. *Journal of Statistical Software*, 76(i09).

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(3):561–575.

Biernacki, C. and Jacques, J. (2016). Model-based clustering of multivariate ordinal data relying on a stochastic binary search algorithm. *Statistics and Computing*, 26(5):929–943.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg.

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.

Celeux, G. and Govaert, G. (1991). A classification em algorithm for clustering and two stochastic versions. *Computational Statistics Quaterly*, 2(1):73–82.

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

D'Elia, A. and Piccolo, D. (2005). A mixture model for preferences data analysis. *Computational Statistics & Data Analysis*, 49(3):917–934.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38.

Dillon, W., Firtle, N. C., and Madden, T. C. (1990). Marketing research in a marketing environment. Technical report, IRWIN,.

Fernández, D., Arnold, R., and Pledger, S. (2016). Mixture-based clustering for the ordered stereotype model. *Computational Statistics & Data Analysis*, 93:46–75.

Gilula, Z., McCulloch, R., Ritov, Y., and Urminsky, O. (2018). A study into mechanisms of attitudinal scale conversion: A stochastic ordering approach.

Giordan, M. and Diana, G. (2011). A clustering method for categorical ordinal data. *Communications in Statistics?Theory and Methods*, 40(7):1315–1334.

Gormley, I. C. and Murphy, T. B. (2010). A mixture of experts latent position cluster model for social network data. *Statistical methodology*, 7(3):385–405.

Gouget, C. (2006). *Utilisation des modèles de mélange pour la classification automatique de données ordinales*. PhD thesis, Compiègne.

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

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

Jollois, F.-X. and Nadif, M. (2009). Classification de données ordinales: modèles et algorithmes. In *41èmes Journées de Statistique, SFdS, Bordeaux*.

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

Keribin, C., Celeux, G., and Valérie, R. (2017). The latent block model: a useful model for high dimensional data. In *ISI 2017-61st world statistics congress*, pages 1–6.

Little, R. J. and Rubin, D. B. (2014). *Statistical analysis with missing data*, volume 333. John Wiley & Sons.

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

McParland, D. and Gormley, I. C. (2016). Model based clustering for mixed data: clustmd. *Advances in Data Analysis and Classification*, 10(2):155–169.

Podani, J. (2006). Braun-blanquet's legacy and data analysis in vegetation science. *Journal of Vegetation Science*, 17(1):113–117.

Ranalli, M. and Rocci, R. (2016). Mixture models for ordinal data: a pairwise likelihood approach. *Statistics and Computing*, 26(1-2):529–547.

Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850.

Schwarz, G. et al. (1978). Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464.

Scrucca, L. (2016). Genetic algorithms for subset selection in model-based clustering. In *Unsupervised Learning Algorithms*, pages 55–70. Springer.

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.