Self-Organized Co-Clustering for textual data synthesis

Margot Selosse, Julien Jacques, Christophe Biernacki

Self-Organized Co-Clustering for textual data synthesis

Margot Selosse
Université Lumière Lyon 2, laboratoire ERIC,
Julien Jacques
Université Lumière Lyon 2, laboratoire ERIC,
and
Christophe Biernacki
CNRS, Inria, Université de Lille

April 30, 2019

Abstract

Recently, different studies have demonstrated the interest of co-clustering, which simultaneously produces row-clusters of observations and column-clusters of features. The present work introduces a novel co-clustering model for parsimoniously summarizing textual data in document-term format. In addition to highlighting homogeneous co-clusters - as other existing algorithms do - we also distinguish noisy co-clusters from significant ones, which is particularly useful for sparse document-term matrices. Furthermore, our model proposes a structure among the significant co-clusters and thus provides better interpretability for the user. The approach proposed competes with state-of-the-art methods for document and term clustering, and offers user-friendly results. The model relies on the Poisson distribution, and a constrained version of the Latent Block Model, which is a probabilistic approach for co-clustering. A Stochastic Expectation-Maximization algorithm is proposed to perform the model's inference as well as a model selection criterion to choose the number of co-clusters.

Keywords: Latent Block Model, document-term matrix

1 Introduction

This work presents the Self-Organised Co-Clustering model (SOCC). It aims at providing a tool for synthesizing large document-term matrices, whose rows correspond to documents and columns correspond to terms. The clustering approach, which consists in forming homogeneous groups of observations (here documents), is a useful unsupervised technique for this task and has proved its efficiency in several domains. However, most of the existing clustering procedures exclusively focus on partitioning along one dimension of a data matrix, by performing only row-clustering. In high-dimensional and sparse contexts, they sometimes are less adapted or difficult to interpret. When considering such data sets, co-clustering, which groups observations and features simultaneously, turns out to be more efficient. It exploits the dualism between rows and columns and the data set is summarized in blocks (the crossing of a row-cluster and a column-cluster). In the present work, we focus on co-clustering of document-term matrices which are sparse and high dimensional. In this context, our work helps finding similar documents and their interplay with word clusters.

Most of the time, we distinguish two kinds of co-clustering approaches. Matrix factorization based methods, e.g. Ding et al. (2006); Wang et al. (2011), consist in factorizing the $N \times J$ data matrix $\boldsymbol{x}$ into three matrices $\boldsymbol{a}$ (of size $N \times G$), $\boldsymbol{b}$ (size $G \times H$), $\boldsymbol{c}$ (size $H \times J$), with the property that all three matrices are non-negative. More specifically, the approximation of $\boldsymbol{x}$ by $\boldsymbol{x} \approx \boldsymbol{abc}$ is achieved by minimizing the error function $\min_{(\boldsymbol{a},\boldsymbol{b},\boldsymbol{c})} ||\boldsymbol{x} - \boldsymbol{abc}||_F$, with the constraints $(\boldsymbol{a} \geq 0, \boldsymbol{b} \geq 0, \boldsymbol{c} \geq 0)$, and $||.||_F$ being a norm to choose. The matrices $\boldsymbol{a}$ and $\boldsymbol{c}$ define the row and column cluster memberships respectively. Each value of the matrix $\boldsymbol{a}$ (respectively $\boldsymbol{c}$) corresponds to the degree in which a row (resp. a column) belongs to a row-cluster (resp. a column-cluster). The matrix $\boldsymbol{b}$ represents the *block* matrix: an element b_{gh} of $\boldsymbol{b}$ is a scalar that summarizes the observations belonging to row-cluster g and column-cluster h. Therefore, these methods require to choose the metric $||.||_F$ that best fits the structure of underlying latent blocks based on available data, which can be difficult. Furthermore, to the best of our knowledge, they do not propose a way to select the correct number of blocks.

Probabilistic approaches, for example the Latent Block Model Govaert and Nadif (2010), proceed differently. They usually assume that data were generated from a mixture of probability distributions whose each associated component corresponds to a block. Then, the parameters of the related distributions and the posterior probabilities of the blocks given the data are to be estimated. This approach models the elements of a block with a parametric distribution, which gives more information than a simple scalar as in the previous methods. In addition, each block is interpretable thanks to its distribution's parameters. Moreover, criterion as the ICL Biernacki et al. (2000) can be used for model selection purpose, including the choice for the number of blocks.

However, when dealing with high-dimensional sparse data, several blocks may be mainly sparse (composed of zeros) and cause inference issues. In addition, highlighting homogeneous blocks is not always sufficient to obtain easy-to-interpret results. Indeed, despite being homogeneous, these sparse blocks are not relevant from an interpretation perspective, and we need a new step to select the pertinent blocks. In other words, it is left to the user to choose the most useful co-clusters so as to determine which term clusters (column-clusters) are more specific to which document clusters (row-clusters). This task is not straightforward even with a reasonable number of row and column clusters. Therefore, it is necessary to work on a co-clustering technique that offers ready-to-use results.

We can address this problem by imposing a pattern on the co-clustering structure. Such an approach directly produces the most meaningful co-clusters, and significantly simplifies the results and their analysis. In the present work, we propose a co-clustering approach based on the Latent Block Model Govaert and Nadif (2014), in which we impose a structure whose column-clusters (clusters of terms) are separated into three parts. In the first part, each cluster of terms is specific to one cluster of documents. In the second part, each cluster of terms is specific to two clusters of documents. The third part contains only one column-cluster and gathers terms that are common to all clusters of documents. The main motivation of this paper is to provide a tool with high understandability: having three sections offers explicable results, with a reasonable number of co-clusters. Choosing to restrain our model to pairwise interactions between clusters was essentially motivated

by a mimicking of the classical ANOVA modeling which is usually limited to the two-way analysis. Namely we have adopted the related arguments motivating this current restriction: first, pairwise interactions offer a larger interpretability than higher interactions do; second, interactions between more than three factors are expected to be infrequent. Figure 1 illustrates the structure proposed. On the left, we present a usual co-clustering with the Poisson Latent Block Model. On the right, we show a co-clustering with the SOCC structure: we added thin separations between the three parts of column-clusters, and the noisy blocks are the lighter ones.

Other works have introduced a structure in their related co-clustering. In Laclau and Nadif (2016) and Ailem et al. (2017), the authors propose block diagonal co-clustering techniques. Firstly, it consists in constraining the co-clustering such that the number of row-clusters is equal to the number of column-clusters. Secondly, the blocks out of the diagonal are considered to be noisy, and share the same parameter. Actually, these models are particular cases of the model we propose: they restrain the structure to only the first part of column-clusters we mentioned above. While these methods proved their efficiency in the case of document-term matrices, they assume that a cluster of terms is specific to only one cluster of documents. However, a group of terms could be specific to several groups of documents. Let us assume for instance that documents are research papers, with a cluster related to computer science, and an other one related to mathematics. Each cluster has its own specific terms, but many terms (for instance those related to probability distributions) will appear in both communities.

The rest of this paper is organized as follows. Section 2 presents the Latent Block Model and its application to counting data with the Poisson distribution. Section 3 describes the novel method referred to as "Self-Organized Co-Clustering" (SOCC). In Section 4, we assess the efficiency of our solution in three ways. Firstly, we use simulated data, to evaluate the partition estimation. Secondly, we use real textual data sets to compare the approach proposed with state-of-the-art methods, regarding both document clustering and term clustering. Thirdly, we describe a use case of the SOCC model on a real data set. The last section concludes the paper and discusses hints for possible future research.

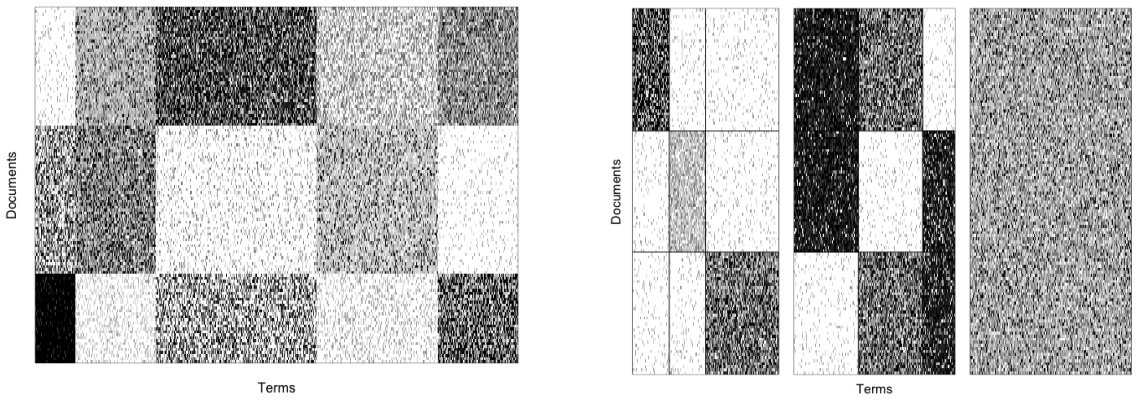

Figure 1: On the left, the usual Poisson Latent Block Model: we see that some blocks are not easily classifiable into noisy or significant blocks. On the right, the SOCC approach: we easily distinguish the noisy blocks (lighter ones) and the significant ones.

2 Background and notation

2.1 The Latent Block Model

The Latent Block Model (LBM) is a widely used model for performing co-clustering Govaert and Nadif (2014). It assumes that knowing the rows and columns partitions, the elements of a block are independent and identically distributed. In this section, the hypotheses for the LBM are defined, and the mathematical details are given.

Let us consider the data matrix $\boldsymbol{x} = (x_{ij})_{i,j}$ with $1 \leq i \leq N$ and $1 \leq j \leq J$. It is assumed that G row-clusters and H column-clusters exist, and that they correspond to a partition $\boldsymbol{v} = (\boldsymbol{v}_i)_i$ of the rows and a partition $\boldsymbol{w} = (\boldsymbol{w}_j)_j$ of the columns. We have $\boldsymbol{v}_i = (v_{ig})_g$ with v_{ig} equal to 1 if row i belongs to cluster g ($1 \leq g \leq G$), and 0 otherwise. Similarly, we have $\boldsymbol{w}_j = (w_{jh})_h$ with w_{jh} equal to 1 when column j belongs to cluster h ($1 \leq h \leq H$), and 0 otherwise. Thereafter, we no longer specify the ranges of i, j, g and h.

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

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

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

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

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

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

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

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

2.2 The Poisson Latent Block Model (PLBM)

Counting data, such as those present in document-term matrix, can be modeled by the Poisson distribution. For a block (g, h) a Poisson distribution with a specific parameterization is considered: $\mathcal{P}(n_{i.} n_{.j} \delta_{gh})$, where $n_{i.} = \sum_j x_{ij}$ and $n_{.j} = \sum_i x_{ij}$. The values $n_{i.}$ and $n_{.j}$ are independent of the co-clustering and are computed from the document term matrix beforehand. Consequently, the LBM parameter α_{gh} correspond to δ_{gh}, and is referred to as "the effect of block (g, h)" Govaert and Nadif (2010). The probability density function is given by:

$$f(x_{ij}; \delta_{gh}) = \frac{1}{x_{ij}!} e^{-n_{i.} n_{.j} \delta_{gh}} (n_{i.} n_{.j} \delta_{gh})^{x_{ij}}. \tag{2}$$

2.3 Inference

The EM-algorithm Dempster et al. (1977) is a well-known method for performing parameter estimation with latent variables. It iterates two steps. The first one, referred to as "E-step", computes the expected complete log-likelihood conditionally to the observed data.

The second one, referred to as "M-step" consists in maximizing the expected complete log-likelihood over the parameters $\boldsymbol{\theta}$. Given equations (1) and (2), the complete log-likelihood is written:

$$\begin{aligned} L_c(\boldsymbol{\theta}; \boldsymbol{x}, \boldsymbol{v}, \boldsymbol{w}) &= \sum_{ig} v_{ig} \log \gamma_g + \sum_{jh} w_{jh} \log \rho_h \\ &+ \sum_{ijgh} v_{ig} w_{jh} \big(x_{ij} \log(n_{i.} n_{.j} \delta_{gh}) - n_{i.} n_{.j} \delta_{gh} - x_{ij}! \big). \end{aligned} \tag{3}$$

Then, the E-step will require to compute the probability $p(v_{ig} w_{jh} = 1 | \boldsymbol{x})$, which is not computationally tractable since the row and column partitions are not independent conditionally to $\boldsymbol{x}$. In such a situation, several alternatives to the EM algorithm exist, as the variational EM algorithm, the SEM-Gibbs algorithm or other algorithm linked to a Bayesian inference Keribin et al. (2013). In this work, we use the SEM-Gibbs version for its simplicity of implementation, its low sensitivity to initialization and its good performance. Instead of computing the probability $p(v_{ig} w_{jh} = 1 | \boldsymbol{x})$, we sample $(\boldsymbol{v}, \boldsymbol{w})$ through a Gibbs sampler. It requires to compute the probabilities $p(v_{ig} = 1 | \boldsymbol{x}, \boldsymbol{w}; \boldsymbol{\theta})$ and $p(w_{jh} = 1 | \boldsymbol{x}, \boldsymbol{v}; \boldsymbol{\theta})$ which are tractable. Algorithm 1 presents the SEM-Gibbs algorithm for the PLBM inference.

3 Self-Organized Co-Clustering

3.1 An easy-to-read structure

In the latter Section, all the δ_{gh} are unrelated, and consequently, each block should be interpreted separately from each other. In the model we propose, this independence is not true anymore: a structure is forced among the blocks so that the result is easier to read. Thus, for a given block (g, h), the corresponding block effect δ_{gh} will either be specific to column-cluster h with $\delta_{gh} = \delta_h$, or non-specific, with $\delta_{gh} = \delta$. In the case of non-specific block effect $\delta_{gh} = \delta$, the block (g, h) is considered as a noisy block. We refer to it as a "non-meaningful" block, and it shares the same δ with all the other non-meaningful blocks. In the case of $\delta_{gh} = \delta_h$, the block (g, h) is "meaningful", and shares the same δ_h with all the meaningful blocks of the same column-cluster h. In this case, the terms of the h^{th} column-cluster are thought to be specific to the documents of one or several row-clusters.

Input: $\boldsymbol{x}$, G, H

Initialization: $\boldsymbol{v}$, $\boldsymbol{w}$, $\gamma_g = \frac{1}{N}\sum_i v_{ig}$, $\rho_h = \frac{1}{J}\sum_j w_{jh}$

for *i in 1:nbSEM* **do**

Step 1: Sample $\boldsymbol{v}$ such that:

$p(v_{ig} = 1|\boldsymbol{x}, \boldsymbol{w}; \boldsymbol{\theta}) \propto \gamma_g \times \prod_{jh} f(x_{ij}; \delta_{gh})^{w_{jh}}$

Step 2: $\gamma_g = \frac{1}{N}\sum_i v_{ig}$,

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

with $n_{g.} = \sum_{ij} v_{ig}x_{ij}$ and $n_{.h} = \sum_{ij} w_{jh}x_{ij}$.

Step 3: Sample $\boldsymbol{w}$ such that:

$p(w_{jh} = 1|\boldsymbol{x}, \boldsymbol{v}; \boldsymbol{\theta}) \propto \rho_h \times \prod_{ig} f(x_{ij}; \delta_{gh})^{v_{ig}}$

Step 4: $\rho_h = \frac{1}{J}\sum_j w_{jh}$ and δ_{gh} as in Step 2.

end

Algorithm 1: Poisson SEM-Gibbs algorithm

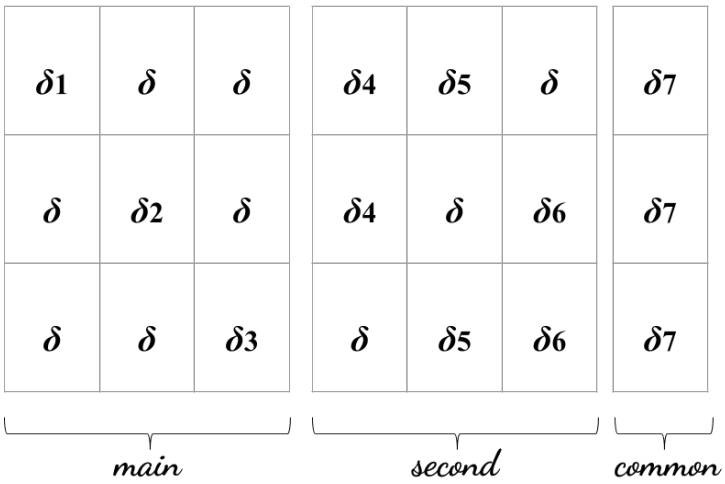

Figure 2: Co-clustering structure of the Self-Organized Co-Clustering model, with block effect parameters.

To organize these meaningful and not-meaningful blocks, several rules are given. First of all, after choosing the number of row-clusters G, the co-clustering necessarily has $H =$

$G + \binom{G}{2} + 1$ column-clusters. Moreover, the column-clusters are divided into three sections called *main*, *second* and *common*. We detail here the purpose of these sections.

The *main* section concerns the first G column-clusters, for $h \in \{1, ..., G\}$. In each column-cluster h of this section, only one block is meaningful, parameterized by δ_h. All the other blocks are non-meaningful and parameterized by δ. Consequently, for each cluster of documents (row-cluster), the meaningful block indicates the terms that are specific to these documents.

The *second* section concerns the following $\binom{G}{2}$ column-clusters ($h \in \{G+1, ..., G+\binom{G}{2}\}$). In each column-cluster h of this section, two blocks are meaningful. Consequently, each column-cluster contains terms that are specific to two clusters of documents (row-clusters). Finally, the *common* section is made of only one column-cluster and gathers the terms that are common to all documents. This structure, as well as the corresponding block effect, are illustrated by Figure 2.

3.2 The SOCC model and its inference

From Section 3.1, knowing column-cluster h we can write: $g \in \mathcal{C}_h \cup \overline{\mathcal{C}}_h$, such that $\mathcal{C}_h$ are the meaningful blocks of column-cluster h and $\overline{\mathcal{C}}_h$ are the non-meaningful blocks of column h. In this case, the probability of the SOCC model is written as:

$$p(\boldsymbol{x}; \boldsymbol{\theta}) = \sum_{(\boldsymbol{v},\boldsymbol{w}) \in V \times W} \prod_{ig} \gamma_g^{v_{ig}} \prod_{jh} \rho_h^{w_{jh}} \prod_{ijh} \prod_{g \in \mathcal{C}_h} f(x_{ij}; \delta_h)^{v_{ig} w_{jh}} \prod_{g \in \overline{\mathcal{C}}_h} f(x_{ij}; \delta)^{v_{ig} w_{jh}}. \tag{4}$$

The complete log-likelihood is given by:

$$\begin{aligned} &L_c(\boldsymbol{\theta}; \boldsymbol{x}, \boldsymbol{v}, \boldsymbol{w}) = \\ &\sum_{ig} v_{ig} \log \gamma_g + \sum_{jh} w_{jh} \log \rho_h + \sum_{ijh} \left(\sum_{g \in \mathcal{C}_h} v_{ig} w_{jh} \left[x_{ij} \log(n_{i.} n_{.j} \delta_h) - n_{i.} n_{.j} \delta_h - log(x_{ij}!) \right] + \right. \\ &\left. \sum_{g \in \overline{\mathcal{C}}_h} v_{ig} w_{jh} \left[x_{ij} \log(n_{i.} n_{.j} \delta) - n_{i.} n_{.j} \delta - log(x_{ij}!) \right] \right). \end{aligned} \tag{5}$$

As in Section 2.2, the SEM-Gibbs algorithm is chosen to estimate the partitions $(\boldsymbol{v}, \boldsymbol{w})$ and parameters $\boldsymbol{\theta} = (\boldsymbol{\gamma}, \boldsymbol{\rho}, \boldsymbol{\delta})$ with $\boldsymbol{\delta} = (\delta, \delta_1, \ldots, \delta_H)$. In contrast with the Poisson LBM,

Input: $\boldsymbol{x}$, G, H

Initialization: $\boldsymbol{v}$, $\boldsymbol{w}$, $\gamma_g = \frac{1}{N}\sum_i v_{ig}$, $\rho_h = \frac{1}{J}\sum_j w_{jh}$

for *i in 1:nbSEM* **do**

Step 1: Sample $\boldsymbol{v}$ such that:

$p(v_{ig} = 1|\boldsymbol{x}, \boldsymbol{w}; \boldsymbol{\theta}) \propto \gamma_g \times \prod_{jh} f(x_{ij}; \delta_{gh})^{w_{jh}}$

Step 2: $\gamma_g = \frac{1}{N}\sum_i v_{ig}$,

$$\delta = \frac{\sum_{ijhg \in \overline{\mathcal{C}}_h} v_{ig} w_{jh} x_{ij}}{\sum_{ijhg \in \overline{\mathcal{C}}_h} v_{ig} w_{jh} n_{i.} n_{.j}},$$

$$\delta_h = \frac{\sum_{ijg \in \mathcal{C}_h} v_{ig} w_{jh} x_{ij}}{\sum_{ijg \in \mathcal{C}_h} v_{ig} w_{jh} n_{i.} n_{.j}}.$$

Step 3: Sample $\boldsymbol{w}$ such that:

$p(w_{jh} = 1|\boldsymbol{x}, \boldsymbol{v}; \boldsymbol{\theta}) \propto \rho_h \times \prod_{ig} f(x_{ij}; \delta_{gh})^{v_{ig}}$

Step 4: $\rho_h = \frac{1}{J}\sum_j w_{jh}$, δ and δ_h as in Step 2.

end

Algorithm 2: Self-Organized Co-clustering

the Poisson distribution $f(x_{ij}; \delta_{gh})$ of block (g, h) will depend on the meaningfulness of block (g, h). For all $h \in \mathcal{H}$ if $g \in \mathcal{C}_h$, then $f(x_{ij}; \delta_{gh}) = f(x_{ij}; \delta_h)$, while if $g \in \overline{\mathcal{C}}_h$, then $f(x_{ij}; \delta_{gh}) = f(x_{ij}; \delta)$, where f is the Poisson p.d.f. given by Equation (2).

The SEM-Gibbs algorithm proposed for the Self-Organized Co-Clustering inference is summarized in Algorithm 2. It iterates the partitions sampling and the maximization of the parameters (step 1 to 4) during a given number of iterations (nbSEM). The final parameter estimation, denoted now by $\hat{\boldsymbol{\theta}}$, is obtained by averaging the model parameters over the sample distribution (after a burn-in period). Lastly, the final partitions $\hat{\boldsymbol{v}}$ and $\hat{\boldsymbol{w}}$ are estimated with $\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}$, through an other Gibbs sampler.

Choice for the number of iterations For the SEM-Gibbs algorithm, two numbers have to be chosen: the total number of SEM-Gibbs iterations (nbSEM) and the number

of iterations for the burn-in period. These numbers are graphically chosen by visualizing the values of the model's parameters along with the SEM-Gibbs iterations. The parameters must have reached their stationary state after the burn-in period, and the remaining number of iterations until the end must be sufficient to compute their respective mean. Less subjective ways exist to assess the distribution's stationarity. In Gelman and Rubin (1992), the authors propose a general approach to monitor the convergence of MCMC outputs in which parallel chains are run with starting values that are spread relatively to the posterior distribution. Convergence is confirmed when the output from all chains is indistinguishable. Although this method is relevant here, we did not use it to avoid increasing the overall execution time of the algorithm.

3.3 Model selection

The definition of a model selection criterion has two purposes. First, in the context of unsupervised methods, choosing the number of row-clusters G is an issue. One of the great advantages of the SOCC model is that the number of column-clusters H is directly fixed by the number of row-clusters G. Indeed, as explained before, $H = G + \binom{G}{2} + 1$. However, the choice for the number of row-clusters G is still a problem. Second, as described in Algorithm 2, the SEM-Gibbs algorithm starts with a random initialization of partitions $(\boldsymbol{v}, \boldsymbol{w})$. However, this initialization has an impact on the convergence of the algorithm and on the resulting estimations. It is therefore recommended to execute several times the algorithm with different initializations and to have a criterion to choose the best solution.

The most classical criteria, such as BIC Schwarz (1978), rely on penalizing the maximum log-likelihood value $L(\hat{\boldsymbol{\theta}}; \boldsymbol{x})$. However, due to the dependency structure on the row and column partitions conditionally to $\boldsymbol{x}$, the log-likelihood is not tractable.

Alternatively, an approximation of the ICL information criterion Biernacki et al. (2000), referred to as "ICL-BIC", can be used to overcome this problem. The key point is that this latter vanishes since ICL relies on the complete latent block information $(\boldsymbol{v}, \boldsymbol{w})$, instead of integrating on it as it is the case in BIC. In particular, Keribin et al. (2013) detailed how to express ICL-BIC for the general case of categorical data. It is possible to straightforwardly

transpose the ICL-BIC expression given by these authors by following step by step their piece of work, with no new technical material. The resulting ICL-BIC is expressed by:

$$\text{ICL-BIC}(G) = \log p(\hat{\boldsymbol{\theta}}; \boldsymbol{x}, \hat{\boldsymbol{v}}, \hat{\boldsymbol{w}}) - \frac{1}{2}(G-1)\log N - \frac{1}{2}(H-1)\log J - \frac{1}{2}GH\log(NJ). \quad (6)$$

The number G of row-clusters maximizing this criterion has to be retained.

4 Numerical Experiments

In this section, we assess the quality of the SOCC model in two ways. First, we simulated a data set according to the SOCC model process generation. On this data set, we executed Algorithm 2 and verified that the partitions were well estimated. Secondly, we used real textual data sets whose documents are known to belong to classes and compared the row-clustering (resp. column-clustering) quality with other clustering and co-clustering methods. In both cases, we experimentally noticed that when the algorithm reaches its stationary state, it reaches it very fast (around 25 iterations). Furthermore, this stationary state is extremely stable: the parameters do not significantly change in this period. In the few cases the algorithm did not reach its stationary state after 25 iterations, the algorithm would systematically leads to empty the clusters (so an invalidate solution). Therefore, the total number of iterations was set to 50, and the burn-in period's number of iterations was set to 35, in both cases.

4.1 Simulated Data set

4.1.1 Blocks estimation

A data set with $N = 120$, $J = 1200$, $G = 3$ and $H = 7$ was simulated. The parameters were chosen arbitrarily: the row mixing proportions $\boldsymbol{\gamma}$ are equal to $(.33, .33, .33)$ and the column mixing proportions $\boldsymbol{\rho}$ are equal to $(.08, .08, .17, .17, .17, .08, .25)$. The block effects are given in Table 1. The SOCC model was performed on 100 simulations, and the Adjusted Rand Index, referred to as "ARI" Hubert and Arabie (1985) between the right partitions and the estimated ones were computed. The ARI for row-clusters was always equal to 1. Regarding

Table 1: Simulated parameters $\delta_{gh} \times 10^{-7}$. For each cell x_{ij} the Poisson parameter is equal to $n_{i.}n_{.j}\delta_{gh}$, with row margins $n_{i.}$ averagely equal to 2455, and columns margins $n_{.j}$ averagely equal to 249.

Cluster	1	2	3	4	5	6	7
1	8.6	2.9	2.9	49.8	47.8	2.9	34.0
2	2.9	9.0	2.9	49.8	2.9	52.9	34.0
3	2.9	2.9	9.4	2.9	47.8	52.9	34.0

Table 2: Number of row and column-clusters (G, H) selected by ICL-BIC on the 100 simulated data sets, the right one being $(3, 7)$.

(G, H)	(2,6)	**(3,7)**	(4,11)	(5,16)
# chosen	25	**75**	0	0

the column-clusters, the mean ARI was equal to .99. It shows that the inference algorithm for SOCC works appropriately.

4.1.2 Selection for G

For each of the 100 simulations, the co-clustering was performed for $G = \{2, 3, 4, 5\}$ and the ICL criterion was computed. Table 2 presents how many times each G was selected: the right number was selected in 75% of the cases. For the remaining 25% executions, $G = 2$ was selected.

4.2 Real Data sets Experiments

In this section, real labeled data sets are used to assess the quality of the method proposed. We describe here the data sets that were used, the methods the SOCC was compared to, and the results.

4.2.1 Data sets

Seven data sets were retained for this Section. The **classic3** data set (dimension 3 891 × 5 236) and the **classic4** data set[1] (dimension 7 094 × 5 896) consist respectively of 3 different document collections (CISI, CRANFIELD, and MEDLINE) and 4 different document collections (CACM, CISI, CRANFIELD, and MEDLINE). **Pubmed5** (12 648 × 8 863), **Pubmed4** (11 131 × 8 257) and **Pubmed3** (9 582 × 7 454) were built from the collection Pubmed10 Chen et al. (2009), with approximately 15 500 medical abstracts from the Medline database, partitioned across 10 different diseases and published between 2000 and 2008. Pubmed3 contains the three largest classes, while Pubmed4 (resp. Pubmed5) contains the four (resp. five) largest classes. The classes, ranked from the largest to the smallest, include documents about Otitis, Migraine, Age-related Macular Degeneration, Kidney Calculi and Hay Fever. **Pubmed4min** (2 121 × 3 660) was also extracted from the Pubmed10 data set. However, only the four smallest classes were extracted. The documents are about Jaundice, Raynaud Disease, Chickenpox and Gout. The **sports** (8 580 × 14 870) and **yahoo** (2 340 × 10 431) data sets were obtained from the cluto toolkit Karypis (2002). yahoo contains 6 different document categories where each document corresponds to a web page listed in the subject hierarchy of *Yahoo!*. The sports data set contains documents about 7 different sports including baseball, basketball, bicycling, boxing, football, golfing and hockey.

4.2.2 Baselines

Seven clustering, co-clustering and topic-modeling methods were selected as baselines to compare our results. Two of them are based on the Latent Block Model. The Poisson Latent Block Model (PLBM,Govaert and Nadif (2010)), as detailed in Section 2, is a co-clustering algorithm that uses the direct application of the Latent Block Model. The Sparse Poisson Latent Block Model Ailem et al. (2017), referred to as "SPLBM", is a constrained version of the Poisson Latent Block Model, which was also developed for co-clustering document-term matrices. This model, already described in the introduction, is a particular

[1] http://www.dataminingresearch.com/index.php/2010/09/classic3-classic4-datasets/

case of our model restraining the co-clustering structure to only the *main* section. Both models were implemented in `C++` from the pseudo-code of their respective papers. The Information Theory Co-Clustering method, referred to as "ITCC" Dhillon et al. (2003), is a co-clustering technique that uses information theory and the mutual information to discover the blocks. We used the `C++` implementation provided by their authors. The Orthogonal Non-negative Matrix Tri-Factorization method, referred to as "ONMTF" Ding et al. (2006), is a co-clustering algorithm based on matrix factorization. We implemented the provided pseudo-code in `R`. The Non-negative Matrix Factorization NMF Paatero and Tapper (1994) is a clustering algorithm based on matrix factorization. The R Package `NMF` Gaujoux and Seoighe (2010) was used for the experiments. The Spherical Kmeans clustering method ("Skmeans") is the implementation of the kmeans algorithm, but embedding the Cosine similarity (and not the Euclidean distance). The R Package `skmeans` Hornik et al. (2012) was used for the experiments. Latent Dirichlet allocation (LDA) Blei et al. (2003) is a generative statistical model for topic modeling. The R package `textmineR` implementation was used to perform it on the data sets. To assess the row-clusters quality, all of these seven methods were used. To assess the column-clusters quality, we obviously selected the four co-clustering methods only.

4.2.3 Assessing the quality of row-clusters

To assess the document clustering quality, the ARI between the known partitions and the estimated ones were computed. For each data set, each method was executed 30 times. Figure 3 plots the ARIs boxplots for all data sets and methods. We see that the SOCC approach is clearly the best model for data sets classic3, pubmed4min and sports. On the other data sets, it obtains satisfying results, and ranks as the second-best method in terms of ARI, just after Skmeans. This latter clustering method yields better results on data sets pubmed3, pubmed4, and pubmed5 but it presents one of the worst performances for classic4, pubmed4min and sports. Therefore, even if it obtains good results on some data sets, the inconstancy on the other ones makes it an unreliable method. For this reason, SOCC seems to be the best method from a document clustering point of view. The reason

of this success probably relies on the model's parsimony.

4.2.4 Assessing the quality of column-clusters

In most studies, co-clustering algorithms evaluation is based on the resulting row-clusters only. This is due to the lack of public data sets providing the true partitions for both observations and features. In document clustering for example, popular benchmarks provide the true documents labels, while the term clusters remain unknown. To overcome this problem and improve over currently used evaluation methods, we propose the following strategy. For a given column-cluster, we extract the ten most frequents terms. We compute the average Jaccard similarity between these terms on the basis of the whole basis of documents: this value is considered as a proximity measure between terms of the column-cluster. We average this proximity measure over all the column-clusters. In terms of interpretation, this criterion based on Jaccard similarities is going to assess how a co-clustering gathers terms that often occur in the same document. We report the scores obtained by the methods on the data sets in Table 3. From these results, we observe that on the classic4, pubmed3, pubmed4, pubmed4min, pubmed5, sports and yahoo data sets, all algorithms perform equally well but the SOCC model brings the highest averaged score. Regarding the classic3 data set, ONMTF yields a better result (.89), but is closely followed by the SOCC model (.88).

4.2.5 pubmed4min use case

In this section, we show on the Pubmed4min data set that the SOCC's results are easy-to-interpret. Regarding the *main* section, when we seek the 10 most frequent terms of the first column-cluster, we get "varicella", "vaccin", "ag", "children", "year", "immun", "zoster", "hospit", "chickenpox", "adult". These terms are closely related to chickenpox (or varicella), so we can easily guess that the first row-cluster's documents are those about chickenpox. When we seek the 10 most frequent terms of the second column-cluster, we get "jaundic", "obstruct", "liver", "bile", "biliari", "hepat", "duct", "rat", "stent", "bilirubin". Again, we can easily say that the second row-cluster's documents are about jaundice.

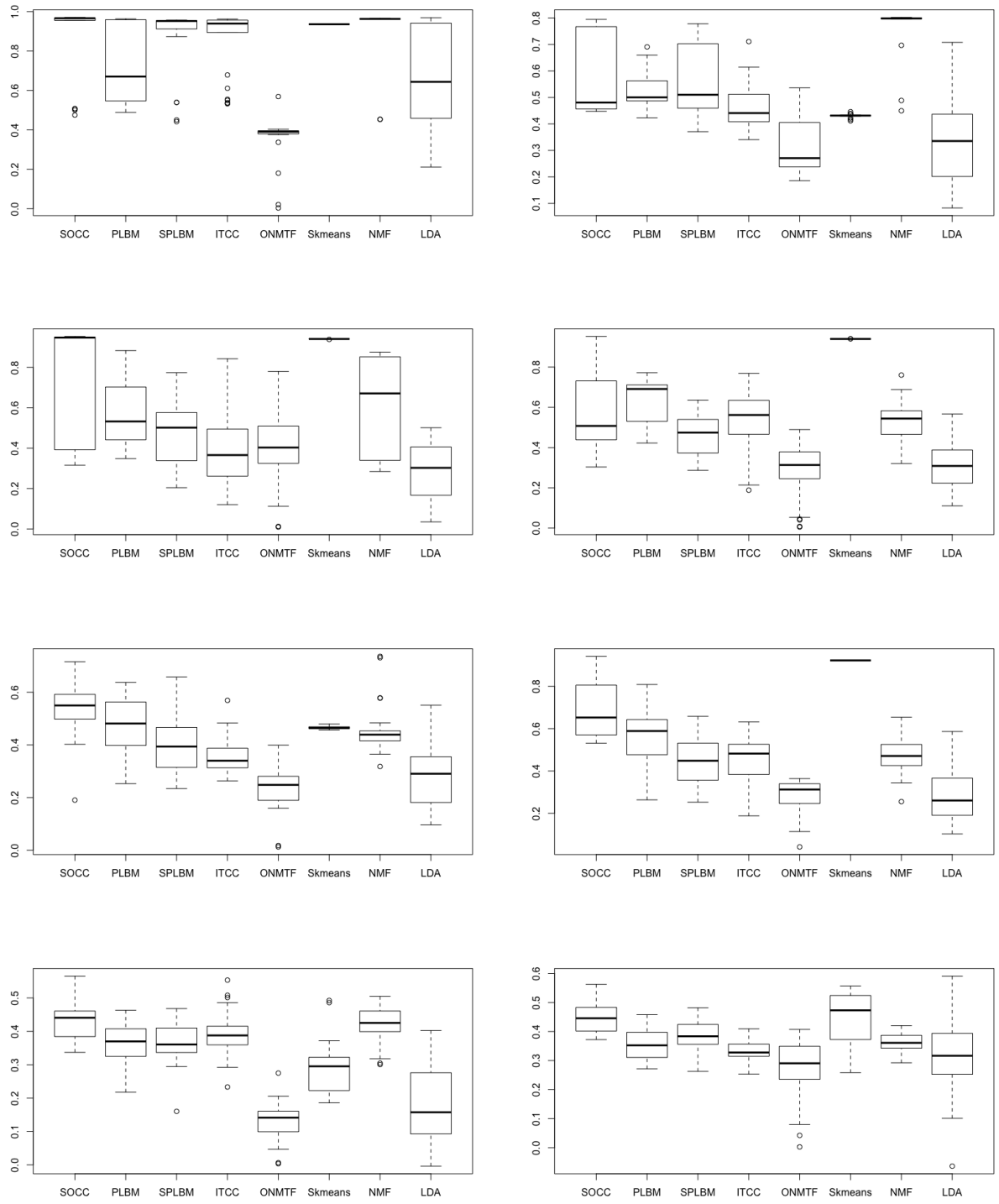

Figure 3: ARIs for document clustering. From left to right and top to bottom: classic3, classic4, pubmed3, pubmed4, pubmed4min, pubmed5, sports, yahoo.

Table 3: Average similarity measure between the 10 top terms of each column-cluster.

Data set	SOCC	PLBM	SPLBM	ITCC	ONTMF
Classic3	.88 (.07)	.86 (.08)	.86 (.08)	.86 (.08)	**.89** (.07)
Classic4	**.91** (.06)	.88 (.07)	.88 (.07)	.87 (.07)	.87 (.07)
Pubmed3	**.85** (.13)	.77 (.13)	.79 (.12)	.76 (.13)	.80 (.08)
Pubmed4	**.88** (.12)	.80 (.15)	.80 (.13)	.80 (.14)	.81 (.09)
Pubmed4min	**.87** (.11)	.79 (.13)	.81 (.09)	.80 (.13)	.84 (.08)
Pubmed5	**.90** (.12)	.78 (.13)	.81 (.13)	.83 (.13)	.85 (.08)
Sports	**.88** (.11)	.79 (.11)	.79 (.11)	.77 (.11)	.78 (.10)
YahooKB1	**.85** (.20)	.67 (.31)	.70 (.33)	.69 (.31)	.69 (.31)

Now, regarding the *second* section, if we look at column-cluster 5, which corresponds to the terms specific to row-clusters 1 and 2, we get: "rate", "complic", "neg", "mortal", "morbid", "infant", "neonat", "bacteri", "safe", "inva". These terms are mostly related to children, which seems coherent since jaundice and chickenpox are very common for toddlers and newborns. Furthermore, jaundice can occur as a complication of chickenpox, which justifies the presence of "complic" in the list.

5 Conclusion and Future Work

In this paper, we propose the SOCC model, a novel approach for parsimoniously co-cluster textual data sets. It offers an easy-to-read result, and quickly shows which terms are specific to one group of documents, which terms are specific to two groups of documents and which terms are common to all documents. The resulting algorithm is not only more accurate than other state-of-the-art methods but also able to detect the number of co-clusters, with the ICL criterion.

In future works, we could define other structures, for example with clusters of terms specific to 3 or more groups of documents. The first concern here is the increasing number of column-clusters (it would require at least $\binom{3}{G}$ more column-clusters). Also, it would be

Figure 4: Co-clustering of pubmed4min data set with SOCC method. From left to right: the *main*, the *second* and the *common* sections. The graphic was produced using the `Python` function *spy*() with argument *markersize* set to 1.3.

interesting to investigate a more developed model selection: we could allow the structure not to have all $G + \binom{G}{2} + 1$ column clusters. For example, on Figure 4, we see the pubmed4min SOCC co-clustering with $G = 4$. We know that the *second* part comprises $\binom{4}{2} = 6$ column-clusters. We easily notice five of them, but the sixth one is very small: is this column-cluster necessary? We could use the ICL criterion to get rid of the irrelevant column-clusters. However, relaxing the strict structure assumption brings other issues: testing all solutions could truly increase the overall execution time.

6 Supplementary Materials

SOCC_1.0.tar.gz: R-package containing the implementation of the SOCC method as well as the SPLBM and PLBM methods.

example.R: R code with examples on the SOCC package.

References

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

Biernacki, C., G. Celeux, and G. Govaert (2000, July). Assessing a mixture model for clus-

tering with the integrated completed likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence. 22*(7), 719–725.

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

Chen, Y., L. Wang, M. Dong, and J. Hua (2009, November). Exemplar-based visualization of large document corpus (infovis2009-1115). *IEEE Transactions on Visualization and Computer Graphics 15*(6), 1161–1168.

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

Dhillon, I. S., S. Mallela, and D. S. Modha (2003). Information-theoretic co-clustering. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '03, New York, NY, USA, pp. 89–98. ACM.

Ding, C., T. Li, W. Peng, and H. Park (2006). Orthogonal nonnegative matrix t-factorizations for clustering. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, New York, NY, USA, pp. 126–135. ACM.

Gaujoux, R. and C. Seoighe (2010). A flexible r package for nonnegative matrix factorization. *BMC Bioinformatics 11*(1), 367.

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

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

Govaert, G. and M. Nadif (2014). *Co-Clustering*. Computing Engineering series. ISTE-Wiley.

Hornik, K., I. Feinerer, M. Kober, and C. Buchta (2012). Spherical k-means clustering. *Journal of Statistical Software 50*(10), 1–22.

Hubert, L. and P. Arabie (1985, Dec). Comparing partitions. *Journal of Classification 2*(1), 193–218.

Karypis, G. (2002). CLUTO a clustering toolkit. Technical Report 02-017, Dept. of Computer Science, University of Minnesota.

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

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

Paatero, P. and U. Tapper (1994, 06). Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics 5*, 111–126.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics 6*, 461–464.

Wang, H., F. Nie, H. Huang, and C. Ding (2011, Dec). Nonnegative matrix tri-factorization based high-order co-clustering and its fast implementation. In *2011 IEEE 11th International Conference on Data Mining*, pp. 774–783.