Co-clustering for binary and functional data

Yosra Ben Slimen, Julien Jacques, Sylvain Allio

Co-clustering for binary and functional data

Yosra Ben Slimen[1,2], Julien Jacques[1] and Sylvain Allio[2]

[1]Université de Lyon, Lyon 2, ERIC EA3083, Lyon, France

[2]Orange Labs, Belfort, France

yosra.benslimen@gmail.com, Julien.Jacques@univ-lyon2.fr, sylvain.allio@orange.com

Key Words: Co-clustering, Functional data, Mixed data, EM algorithm, Latent block model, ICL-BIC criterion, Mobile network.

ABSTRACT: Due to the diversity of mobile network technologies, the volume of data that has to be observed by mobile operators in a daily basis has become enormous. This huge volume has become an obstacle to mobile networks management. This paper aims to provide a simplified representation of these data for an easier analysis. A model-based co-clustering algorithm for mixed data, functional and binary, is therefore proposed. Co-clustering aims to identify block patterns in a dataset from a simultaneous clustering of rows and columns. The proposed approach relies on the latent block model, and three algorithms are compared for its inference: stochastic EM within Gibbs sampling, classification EM and variational EM. The proposed model is the first co-clustering algorithm for mixed data that deals with functional and binary features. The model has proven its efficiency on simulated data and on real data extracted from live 4G mobile networks.

1 Introduction

The mobile telecommunication industry has and is still undergoing interesting changes as a result of the introduction of new technologies and services. The operators and manufacturers

of mobile equipments are undertaking huge efforts to adapt the cellular networks to the new technologies, while maintaining a top quality of services. Consequently, the operation of the radio network is becoming increasingly complex in an environment where the fault management and network optimization are still a manual process. They are accomplished by experts that are dedicated to daily analyze multiple sources of information that describe the network performances. Key Performance Indicators (KPIs) and alarms [1] are the most used source of data due to their accessibility and because they give high level information about the network.

An alarm is a message generated to alert the experts in case of problems in the network. It can be defined as a binary variable y, which is equal to 1 if a problem has occurred, 0 otherwise. KPIs are aggregated measurements, defined by mathematical formulas and derived from different counters. They are computed periodically from the network with different temporal granularities (weekly, daily, hourly or less). Figure 1 illustrates a sample of 30 KPIs and 10 alarms for 20 daily observations.

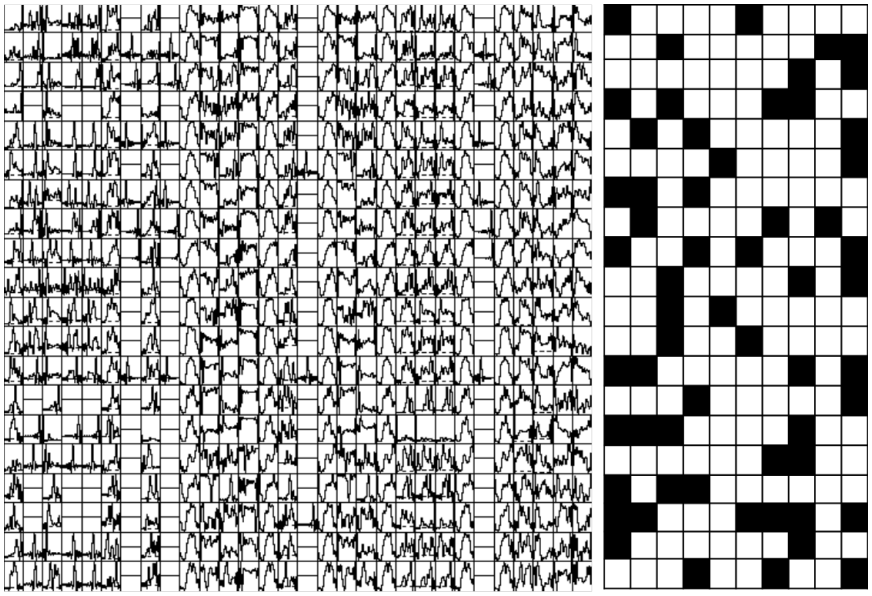

Figure 1: An example of functional dataset composed of 20 observations, 30 KPIs and 10 alarms

Commonly, KPIs are treated as vectors of multivariate data because each KPI is given

as a set of measurements along a time interval. However, the measurements within each KPI may be highly correlated which represents a challenge for the multivariate approaches. Another alternative is to consider the KPIs as time series. In these latter, it is assumed that the data are observed at regular intervals of time. They are therefore not appropriate for the irregularly spaced KPIs as the path is no longer a constant. In this paper, we propose to consider the KPIs as functional data [2] which is the observation $x = x(t)$ of a stochastic process $X = \{X(t), t \in T\}$, where T is a time interval. Many advantages arise from this choice [3]. First, the generating models can be described by continuous smooth dynamics which may allow for accurate estimates of the parameters that have to be used in the analysis phase. Second, Functional Data Analysis (FDA) methods allow the data noise reduction and also the treatment of missing data through curve smoothing. Third, by saying that a curve is smooth, we usually mean that it is differentiable to a certain degree, implying that a number of derivatives can be derived or estimated from the data. Such derivative information may reveal patterns in a (functional) dataset that address important research questions. Finally, FDA is applicable to data with regular or even irregular time sampling schedules.

Let $\mathbf{X} = (\mathbf{x}, \mathbf{y})$ be the dataset under study, composed of N observations (rows) of F functional features (KPIs) and B binary features (alarms). The dataset structure is illustrated in Figure 2. The functional part of the dataset is $\mathbf{x} = \{x_{if}\}_{if}$ with $1 \leq i \leq N, 1 \leq f \leq F$ and $x_{if} = x_{if}(t)$ for $t \in [0, T]$. All the functional data are measured on the same time interval T but they can be measured with different time slots and even with different number of time slots which is an advantage when dealing with functional data. The binary part is $\mathbf{y} = \{y_{ib}\}_{ib}$ with $1 \leq i \leq N, 1 \leq b \leq B$ and $y_{ib} \in \{0, 1\}$.

For only one radio access technology (GSM, UMTS, LTE,...) and one constructor (Huawei, Ericsson, ...), hundreds of KPIs and of alarms may be defined. Hence, as the number of technologies, services and constructors grows, the number of KPIs and of alarms become enormous and they may need to be observed over a large period of time (several weeks or months). As a consequence, it becomes very difficult for a human to analyze such a large amount of information. To solve this problem, automatic networks have lately been intro-

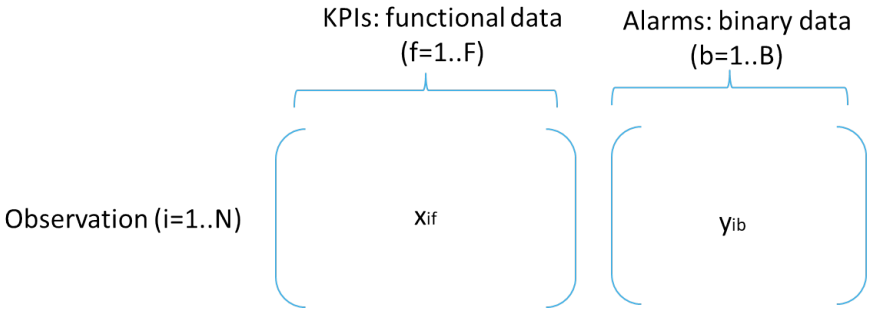

Figure 2: Dataset structure

duced [4] that aim to automate some network procedures. Although these networks can help engineers with their daily work, the huge amount of data is still a challenge. On one hand, observing all the KPIs and all the alarms makes their treatment more greedy in terms of time and memory. On the other hand, ignoring some KPIs or some alarms risks to decrease the network performance. Therefore, it is of primary importance for mobile operators to provide to their engineers some decision support tools. Our work aims to provide a simplified representation of the data that have to be analyzed in a daily basis. To do so, we propose a methodology that identifies groups of network cells having similar behaviors as well as groups of similar KPIs and alarms.

A straightforward solution consists in applying a clustering algorithm on the overall data set. However, this solution would not always work as expected, and this for several reasons: first the feature space contains different types of attributes i.e. functional and binary data; second, KPIs show different properties than alarms and they can be affected by certain problems such as noise, which might affect in a different form the other feature space (alarms). Hence, each feature space may fit a specific model. Third, even though the spaces do not fall into the above-mentioned cases, their cardinality may vary drastically having the consequence that larger feature spaces have more influence on the clustering process than smaller ones. To cope with this limitation of traditional clustering approaches, a co-clustering seems more adequate. Co-clustering aims to define partitions of the observations and of the variables where the blocks (or co-clusters) are obtained by crossing both partitions. Thus, the large data matrix can be summarized by a reduced number of blocks of data. Figure 3

illustrates the differences between both approaches.

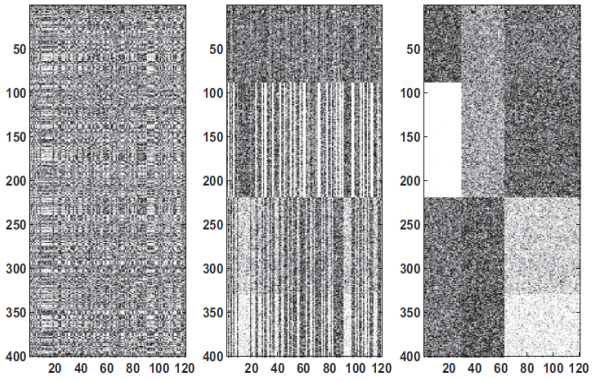

Figure 3: Clustering versus co-clustering: Initial data (left), clustering result (middle) and co-clustering result (right).

Another straightforward solution is to perform a co-clustering for each type of data i.e a co-clustering of functional data and a co-clustering of binary data and then combine the two partitions of instance according to each co-clustering.Although this solution can decompose complex problems into simpler problems, it has one inconvenient since it is sub-optimal and it induces an independence between binary and functional features. Contrarily to co-clustering of mixed data that allows to model some sort of dependence between the two types of features since it assumes that the functional and binary variables are independent conditionally to the block. Although this assumption imposes a conditional independence, it is still less strong than a total independence condition. This assumption (latent class assumption) is classic when we work with categorical and mixed data as explained in [5].

Several approaches of co-clustering have been introduced in literature. The reader can refer to a recent survey related to this topic proposed in [6]. A first category is the matrix reconstruction based family in which the problem is formulated as a matrix decomposition using dissimilarity metrics and a set of constraints such as in [7, 8, 9, 10]. The challenge in this family is in the choice of appropriate distance metrics. Besides, it is useful for quantitative data. In the case of other types of data, one needs to reformulate the whole process. A second category defines co-clustering as a combination method [11, 12] and it

is widely used in text mining and web mining applications. One drawback of this family is its complexity due to the big number of constraints. A third category is the model-based family that uses probabilistic models in order to define the blocks [13, 14, 15]. The advantage of this family is its flexibility compared to the two previous ones. Besides, it has successfully proven its efficiency in many applications such as recommendation systems [16] or text mining [17]. It has also been used for several types of data such as continuous [18], binary [19], contingency [20], categorical [21], [22] or functional data [23, 24, 25]. Although the popularity of probabilistic co-clustering approaches, the case of features of mixed types is much less explored, the only work we found is [26], which proposes a Latent Block Model (LBM) for continuous and binary features.

The purpose of this paper is to propose a co-clustering technique based on an extension of the LBM, quoted as Multiple LBM (MLBM), which is able to take into account two types of features. This model will identify clusters of observations, clusters of functional features (KPIs) and clusters of binary features (alarms). Crossing all these clusters will lead to define blocks of homogeneous data. The blocks related to binary features (alarms) are described by a Bernoulli distribution with block-specific parameters. For the functional features (KPIs), a multivariate Gaussian distribution on the Functional Principal Components Analysis (FPCA, [2]) scores is considered.

The paper is organized as follows. Section 2 explains the specificity of the functional features and how to treat them with a functional data analysis. The proposed MLBM for mixed data is described in Section 3. Model inference is proposed in Section 4, with the comparison of three algorithms: Stochastic EM within Gibbs sampling (SEM-Gibbs), Classification EM (CEM) and Variational EM (VEM). A model selection criterion for choosing the number of row- and column-clusters is proposed in Section 5. The behavior of the model is studied on simulated data in Section 6. Section 7 presents an application of the proposed approach on real data of mobile networks extracted within Orange France. Finally, conclusion and future works are presented in Section 8.

2 Preliminaries: Functional data analysis

The part $\mathbf{x}$ of the dataset $\mathbf{X}$ is composed of functional data $x_{if}(t)$. The main source of difficulty when dealing with functional data consists in the fact that these latter belong to an infinite-dimensional space, whereas in practice, data are generally observed at discrete time points and with some noise. Thus, in order to reflect the functional nature of data, smoothing techniques of the discrete observations into a finite basis of functions is considered.

2.1 Smoothing

Each observed curve x_{if} $(1 \leq i \leq N,\ 1 \leq f \leq F)$ is assumed to be decomposed as a linear combination of basis functions $\{\phi_{fr}\}_{r=1,\dots,M_f}$:

$$x_{if}(t) = \sum_{r=1}^{M_f} a_{ifr}\phi_{fr}(t), \quad t \in [0,T], \tag{1}$$

where $\{a_{ifr}\}_{r=1,\dots,M_f}$ are the basis expansion coefficients for the curve $x_{if}(t)$.

Many basis of functions exist in literature such as trigonometric functions, B-splines or wavelets (see [2] for a detailed study). The choice of the basis as well as the number M_f of basis functions are quite subjective [22]. For example, if the sample paths of $\mathbf{X}$ are smooth and periodic, Fourier basis could be a good choice. If the data are noisy, B-spline basis could be more appropriate due to the optimal properties of cubic B-spline functions.

Regarding the choice of the number M_f of basis functions, the decision is often related to the bias-variance trade-off. A high number of basis functions will yield to a curve that is more faithful to the observed data (low bias) but that is often less smooth (high variance). However, using a small number of basis functions will produce a curve that places less importance on interpolating the discrete points (high bias) but more importance on smoothness (low variance).

Once the basis is chosen, the estimation of (Eq 1) is generally done by smoothing [2]. In this work, after a visual observation of the KPIs under study, smoothing with a B-spline

basis is chosen. Besides, the same basis $\{\phi_r\}_{r=1,\dots,M}$ is used for all the functional features.

2.2 Functional Principal Component Analysis

From the set of functional data, it is interesting to have optimal representation of curves into a functional space of reduced dimension. The main tool to answer this request is the Functional Principal Component Analysis (FPCA, [2]). It consists in computing the principal components C^s and principal eigen-functions f^s of the Karhunen-Loeve expansion:

$$x(t) = \mu(t) + \sum_{s \geq 1} C^s f^s(t), \qquad t \in [0, T]. \tag{2}$$

where $C^s = \int_{t=0}^{T} (x(t) - \mu(t)) f^s(t) dt$ are independent for a sample of independent trajectories and they are uncorrelated across s with $E(C^s) = 0$ and $var(C^s) = \lambda_s$ where λ_s are the eigenvalues. $\mu(t) = E(x(t))$ is the mean function. The convergence of the sum in Eq (2) holds uniformly in the sense that $sup_t E[x(t) - \mu(t) - \sum_{s=1}^{S} C^s f^s(t)]^2 \to 0$ as $S \to \inf$. Expansion (2) facilitates dimension reduction as the first S terms, for large enough S, provides a good approximation to the infinite sum and therefore for x. Hence, the information contained in x is essentially contained in the S-dimensional vector $C = (C^1, \dots, C^S)$ and one works with the approximated process $X(t) = \mu(t) + \sum_{s=1}^{S} C^s f^s(t)$.

In theory, the number of principal components are infinite. However, in practice, due to the fact that the curves are observed at discrete time points and that they are approximated on a finite basis of functions, the maximum number of components one can compute is equal to the number M of basis functions used for approximation. Moreover, in order to reduce the dimensionality of the problem, only the first $m \leq M$ principal components are considered. In this work, in order to project all the data onto the same space, FPCA is applied on the whole dataset of curves, without distinction between curves from different observations or curves from different features.

The number of principal components m is chosen so that it gives the best trade-off between bias and variance. In traditional PCA, several procedures are routinely applied such

as the scree plot or the fraction of variance explained by the first few principal components. These procedures can be directly extended to the functional setting. In [27], a model is designed in order to handle sparse and irregular longitudinal data. The authors develop a version of FPCA in which the functional principal components scores are conditional expectations. In order to choose the best number of PCs, they use less subjective criteria such as pseudo-versions of Bayesian Information Criterion (BIC) and of Akaike Information Criterion (AIC). In this paper, m is fixed empirically so that the principal components express 90% of the total variance. Besides, we can assume without loss of generality, that $\mu(t) = 0$ (in practice the mean curve is subtracted).

Consequently, each curve $x_{if}(t)$ is finally summarized by its m principal components $\mathbf{c}_{if} = (c_{if}^s)_{1\leq s\leq m}$.
Let $\mathbf{c} = (\mathbf{c}_{if}^s)_{1\leq i\leq N, 1\leq f\leq F, 1\leq s\leq m}$ denotes the set of the principal components of all the curves. The dataset under study is thus quoted by $\mathbf{X} = (\mathbf{c}, \mathbf{y})$.

3 Multiple LBM

The objective of our co-clustering methodology is to divide the data into G row-clusters, H column-clusters of functional features and L column-clusters of the binary features. Let's introduce $\mathbf{z} = (z_{ig})_{i=1\ldots N, g=1\ldots G}$ with $z_{ig} = 1$ if row i belongs to cluster g, 0 otherwise. Let $\mathcal{Z}$ be the set of possible values for $\mathbf{z}$ and $\mathbf{z}_i = (z_{i1}, \ldots, z_{iG})$. Similarly, let's define $\mathbf{v} = (v_{fh})_{f=1\ldots F, h=1\ldots H}$ (resp. $\mathbf{w} = (w_{bl})_{b=1\ldots B, l=1\ldots L}$) as the dummy variables for the column-cluster of the functional part (resp. binary part) of the data, and $\mathcal{V}$ and $\mathcal{W}$ the sets of possible values for $\mathbf{v}$ and $\mathbf{w}$, with $\mathbf{v}_f = (v_{f1}, \ldots, v_{fH})$ and $\mathbf{w}_b = (w_{b1}, \ldots, w_{bL})$.

- **Assumption A1** The latent variables $\mathbf{z}$, $\mathbf{v}$ and $\mathbf{w}$ are independent. Therefore, $\forall(\mathbf{z}, \mathbf{v}, \mathbf{w}) \in \mathcal{Z} \times \mathcal{V} \times \mathcal{W}, p(\mathbf{z}, \mathbf{v}, \mathbf{w}) = p(\mathbf{z})p(\mathbf{v})p(\mathbf{w})$.

- **Assumption A2** Conditionally on the row and column partitions $(\mathbf{z}, \mathbf{v}, \mathbf{w})$, the functional $\mathbf{c}$ and binary $\mathbf{y}$ variables are independent.

- **Assumption A3** Into a block gh of functional data, the $\mathbf{c}_{if}$ are independent and identically distributed (i.i.d.) according to a m-variate Gaussian distribution, of parameter (μ_{gh}, Σ_{gh}) specific to the block. Similary, the binary data y_{ib} of block $g\ell$ are i.i.d. according to a Bernoulli distribution of block specific parameter $\alpha_{g\ell}$.

Assumption A1 is already present in the usual LBM in order to reduce the number of possible partitions. The local independence that are presented by assumptions A2 and A3 are commonly used when dealing with latent variables [5].

Under assumptions A1 to A3, the Multiple Latent Block Model (MLBM) is given by:

$$\begin{aligned} \mathrm{p}(\mathbf{c}, \mathbf{y}; \theta) &= \sum_{\mathbf{z}\in\mathcal{Z}, \mathbf{v}\in\mathcal{V}, \mathbf{w}\in\mathcal{W}} p(\mathbf{z})p(\mathbf{v})p(\mathbf{w})p(\mathbf{c}, \mathbf{y}|\mathbf{z}, \mathbf{v}, \mathbf{w}) \\ &= \sum_{\mathbf{z},\mathbf{v},\mathbf{w}} \Big(\prod_{ig} \pi_g^{z_{ig}} \prod_{fh} \rho_h^{v_{fh}} \prod_{b\ell} \tau_\ell^{w_{b\ell}} \prod_{ig}\prod_{fh} \varphi_f(c_{if}; \mu_{gh}, \Sigma_{gh})^{z_{ig}v_{fh}} \prod_{b\ell} \varphi_b(y_{ib}; \alpha_{g\ell})^{z_{ig}w_{b\ell}}\Big) \quad (3) \end{aligned}$$

where $\pi_g = p(z_{ig} = 1)$, $\rho_h = p(v_{fh} = 1)$ and $\tau_\ell = p(w_{b\ell} = 1)$ are the mixing proportion, φ_f is the m-variate Gaussian density and φ_b is the Bernoulli probability.
The parameter $\theta = (\pi_g, \rho_h, \tau_\ell, \mu_{gh}, \Sigma_{gh}, \alpha_{g\ell})_{g,h,\ell}$ of the MLBM are to be estimated, and we propose to use maximum likelihood inference.

4 Parameter estimation

The objective of this section is to estimate θ by maximizing the observed log-likelihood $l(\theta; \mathbf{x}) = \ln \mathrm{p}(\mathbf{c}, \mathbf{y}; \theta)$. The usual way used to maximize the log-likelihood in the presence of missing observations (here $\mathbf{z}, \mathbf{v}, \mathbf{w}$), is the EM algorithm [28]. However, in a co-clustering context, involving a double missing structure on the rows and on the columns, an EM algorithm is computationally intractable. Several alternatives exist [13]. In this work, we propose to compare three of them: Stochastic EM within Gibbs sampling (SEM-Gibbs), Classification EM (CEM) and Variational EM (VEM).

4.1 SEM-Gibbs

The first solution is a stochastic version of the EM algorithm in which the missing data simulation is performed without the need of computing the whole missing data distribution thanks to a Gibbs sampler [29]. Starting from initial values of the partitions $\mathbf{z}^{(0)}$, $\mathbf{v}^{(0)}$, $\mathbf{w}^{(0)}$ and of the parameters $\theta^{(0)}$, the algorithm performs a number q^{max} of iterations. At each iteration q, the SE step consists in computing the probabilities $\widetilde{z}_{ig}^{(q+1)} = p(z_{ig} = 1|\mathbf{c}, \mathbf{y}, \mathbf{v}^{(q)}, \mathbf{w}^{(q)}; \theta^{(q)})$ (for every $1 \leq i \leq N$ and $1 \leq g \leq G$) and simulating the row partition according to $\widetilde{z}_{ig}^{(q+1)}$. Then the column partition for the functional part are simulated according to $\widetilde{v}_{fh}^{(q+1)} = p(v_{fh} = 1|\mathbf{c}, \mathbf{z}^{(q+1)}, \mathbf{w}^{(q)}; \theta^{(q)})$ and those for the binary part are simulated according to $\widetilde{w}_{bl}^{(q+1)} = p(w_{bl} = 1|\mathbf{y}, \mathbf{z}^{(q+1)}, \mathbf{v}^{(q+1)}; \theta^{(q)})$. The M step consists in maximizing the following complete log-likelihood according to θ:

$$L_c(\mathbf{c}, \mathbf{z}, \mathbf{v}, \mathbf{w}; \theta) = \sum_{ig} z_{ig} \log \pi_g + \sum_{fh} v_{fh} \log \rho_h + \sum_{b\ell} w_{b\ell} \log \tau_\ell + \sum_{igfh} z_{ig} v_{fh} \log \varphi_f(c_{if}; \mu_{gh}, \Sigma_{gh}) + \sum_{igb\ell} z_{ig} w_{b\ell} \log \varphi_b(y_{ib}; \alpha_{g\ell})$$

The SEM-Gibbs is detailed in Algorithm 1. The details of the computation at the SE and M steps are as follows:

SE-step

$$\widetilde{z}_{ig}^{(q)} = \frac{\pi_g^{(q)} \prod_{fh} \varphi_f(c_{if}; \mu_{gh}^{(q)}, \Sigma_{gh}^{(q)})^{v_{fh}^{(q)}} \prod_{b\ell} (\alpha_{g\ell}^{y_{ib}} (1 - \alpha_{g\ell})^{1-y_{ib}})^{w_{b\ell}^{(q)}}}{\sum_{g'} \pi_{g'} \prod_{fh} \varphi_f(c_{if}; \mu_{g'h}^{(q)}, \Sigma_{g'h}^{(q)})^{v_{fh}^{(q)}} \prod_{b\ell} (\alpha_{g'\ell}^{y_{ib}} (1 - \alpha_{g'\ell})^{1-y_{ib}})^{w_{b\ell}^{(q)}}}$$

$$\widetilde{v}_{fh}^{(q)} = \frac{\rho_h \prod_{ig} \varphi_f(c_{if}; \mu_{gh}^{(q)}, \Sigma_{gh}^{(q)})^{z_{ig}^{(q+1)}}}{\sum_{h'} \rho_{h'} \prod_{ig} \varphi_f(c_{if}; \mu_{gh'}^{(q)}, \Sigma_{gh'}^{(q)})^{z_{ig}^{(q+1)}}}$$

$$\widetilde{w}_{b\ell}^{(q)} = \frac{\tau_\ell \prod_{ig} (\alpha_{g\ell}^{y_{ib}} (1-\alpha_{g\ell})^{1-y_{ib}})^{z_{ig}^{(q+1)}}}{\sum_{\ell'} \tau_{\ell'} \prod_{ig} (\alpha_{g\ell'}^{y_{ib}} (1-\alpha_{g\ell'})^{1-y_{ib}})^{z_{ig}^{(q+1)}}}$$

M-step

- $\pi_g^{(q+1)} = \frac{1}{N} \sum_i z_{ig}^{(q+1)}$, $\rho_h^{(q+1)} = \frac{1}{F} \sum_f v_{fh}^{(q+1)}$, $\tau_l^{(q+1)} = \frac{1}{B} \sum_b w_{bl}^{(q+1)}$,
- $\mu_{gh}^{(q+1)} = \frac{1}{n_{gh}^{(q+1)}} \sum_i \sum_f c_{if} z_{ig}^{(q+1)} v_{fh}^{(q+1)}$ where $n_{gh}^{(q+1)} = \sum_{if} z_{ig}^{(q+1)} v_{fh}^{(q+1)}$,
- $\Sigma_{gh}^{(q+1)} = \frac{1}{n_{gh}^{(q+1)}} \sum_{if} ((c_{if} - \mu_{gh}^{(q+1)})^t (c_{if} - \mu_{gh}^{(q+1)})) z_{ig}^{(q+1)} v_{fh}^{(q+1)}$,
- $\alpha_{gl}^{(q+1)} = \frac{\sum_i \sum_b z_{ig}^{(q+1)} w_{b\ell}^{(q+1)} y_{ib}}{\sum_i \sum_b z_{ig}^{(q+1)} w_{b\ell}^{(q+1)}}$.

Note that the goal of the the iterations of SEM-Gibbs algorithm is to achieve a stationary distribution. Hence, the burn-in period and the maximum number of iterations q^{max} are empirically fixed by testing different values and by verifying that the parameters distributions are visually stationary in the iterations after the burn-in. One alternative is to use the variance reduction index of Gelman that is popular in Bayesian statistics.

Algorithm 1 SEM-Gibbs algorithm for the MLBM

Require: $\mathbf{c}, \mathbf{y}, G, H, L$

1. Randomly initialize the partitions $\mathbf{z}^{(0)}$, $\mathbf{v}^{(0)}$ and $\mathbf{w}^{(0)}$
2. Compute the parameters $\theta^{(0)}$
3. For $q = 1, \ldots, q^{max}$
   (a) SE step
   - Compute $\widetilde{z}_{ig}^{(q+1)}$ given $\mathbf{c}, \mathbf{y}, \mathbf{v}^{(q)}$ and $\mathbf{w}^{(q)}$
   - Simulate $\mathbf{z}_i^{(q+1)} \sim \mathcal{M}(\widetilde{z}_{i1}^{(q+1)}, \ldots, \widetilde{z}_{iG}^{(q+1)})$
   - Update the parameter $\pi^{(q+.5)}, \mu^{(q+.5)}, \Sigma^{(q+.5)}, \alpha^{(q+.5)}$
   - Compute $\widetilde{v}_{fh}^{(q+1)}$ given $\mathbf{c}, \mathbf{y}, \mathbf{w}^{(q)}$ and $\mathbf{z}^{(q+1)}$
   - Simulate $\mathbf{v}_f^{(q+1)} \sim \mathcal{M}(\widetilde{v}_{f1}^{(q+1)}, \ldots, \widetilde{v}_{fH}^{(q+1)})$
   - Compute $\widetilde{w}_{b\ell}^{(q+1)}$ given $\mathbf{c}, \mathbf{y}, \mathbf{v}^{(q+1)}$ and $\mathbf{z}^{(q+1)}$
   - Simulate $\mathbf{w}_b^{(q+1)} \sim \mathcal{M}(\widetilde{w}_{b1}^{(q+1)}, \ldots, \widetilde{w}_{bL}^{(q+1)})$

   (b) M step
   - Update $\pi^{(q+1)}, \rho^{(q+1)}, \tau^{(q+1)}, \mu^{(q+1)}, \Sigma^{(q+1)}$, $\alpha^{(q+1)}$
4. Compute the estimator $\hat{\theta}$ and the final partitions
   - $\hat{\theta}$ is the mean of the parameters after a burn-in period
   - Generate a new chain of simulation of $(\mathbf{z}^{(q)}, \mathbf{v}^{(q)}, \mathbf{w}^{(q)})$ (as in the SE step) using $\hat{\theta}$
   - Compute the final partition $(\hat{\mathbf{z}}, \hat{\mathbf{v}}, \hat{\mathbf{w}})$ using the marginal modes of the new chain.

4.2 CEM

Similarly to SEM-Gibbs, the Classification EM is an iterative algorithm. At each iteration, we compute the probabilities $\widetilde{z}_{ig}^{(q+1)}$, $\widetilde{v}_{fh}^{(q+1)}$ and $\widetilde{w}_{b\ell}^{(q+1)}$, and, rather than simulating the partition as in SEM algorithm, we compute the partition by maximum a posteriori. Besides, the CE- and M-steps are iterated till convergence. This latter is evaluated by computing the difference between the complete log-likelihood at two consecutive steps. The final parameters and the final partitions correspond to the results of the final iteration.

4.3 VEM

VEM uses a variational approximation that approximates the conditional distributions of the latent variables in a factorizable form. More precisely, we approach $p(\mathbf{z}, \mathbf{v}, \mathbf{w}|\mathbf{X}, \theta)$ by the product of adjustable distributions $q(\mathbf{z}|\mathbf{X}, \theta)$, $q(\mathbf{v}|\mathbf{X}, \theta)$ and $q(\mathbf{w}|\mathbf{X}, \theta)$. Thus, we minor the complete likelihood by the following criterion F_c:

$$\begin{aligned} F_c(\widetilde{z}, \widetilde{v}, \widetilde{w}, \theta) = &\textstyle\sum_{ig} \widetilde{z}_{ig} \log \pi_g + \sum_{fh} \widetilde{v}_{fh} \log \rho_h + \sum_{bl} \widetilde{w}_{bl} \log \tau_l + \sum_{igfh} \widetilde{z}_{ig} \widetilde{v}_{fh} \log \varphi_f(c_{if}; \mu_{gh}, \Sigma_{gh}) \\ &+ \sum_{ibgb} \widetilde{z}_{ig} \widetilde{w}_{bl} \log \varphi_b(y_{ib}; \alpha_{gl}) - \sum_{ig} \widetilde{z}_{ig} \log \widetilde{z}_{ig} - \sum_{fh} \widetilde{v}_{fh} \log \widetilde{v}_{fh} - \sum_{bl} \widetilde{w}_{bl} \log \widetilde{w}_{bl} \end{aligned}$$

The maximization of this criterion is expected to lead to a maximum of the complete likelihood.

In practice, this algorithm consists in computing the probabilities $\widetilde{z}_{ig}^{(q+1)}$, $\widetilde{v}_{fh}^{(q+1)}$ and $\widetilde{w}_{b\ell}^{(q+1)}$. It does not simulate the partitions as the two previous algorithms. It updates θ in the M step using all the observations pondered by there probabilities: the update formula of the M step are the same as SEM-Gibbs algorithm but by replacing $z_{ig}^{(q+1)}$, $v_{fh}^{(q+1)}$ and $w_{b\ell}^{(q+1)}$ by $\widetilde{z}_{ig}^{(q+1)}$, $\widetilde{v}_{fh}^{(q+1)}$ and $\widetilde{w}_{b\ell}^{(q+1)}$. The algorithm is iterated till convergence of F_c.

5 Choice of the number of clusters

In order to choose the number of clusters (G, H, L), a model selection criterion must be involved. One of the most classical ones is the Bayesian Information Criterion (BIC, [30]) that relies on penalizing the maximum log-likelihood value $l(\hat{\theta}; x)$. However, due to the dependency structure of the observed data, this value is not available. Therefore, an approximation of the Integrated Completed likelihood (ICL,[31]) called ICL-BIC is employed. By adapting the formulation of the ICL-BIC criterion developed in [21] in the general case of categorical data, we obtain:

$$\text{ICL-BIC}(G, H, L) = L_c - \frac{G-1}{2} \log(N) - \frac{H-1}{2} \log(F) - \frac{L-1}{2} \log(B) - \frac{\nu_F}{2} \log(NF) - \frac{GL}{2} \log(NB)$$

with $\nu_F = GH(m + \frac{m(m+1)}{2})$ is the number of continuous parameters for the functional part of the MLBM. The number of clusters maximizing this criterion has to be retained.

6 Experiments with simulated data

The aim of this section is to evaluate, through simulation studies, the robustness of the co-clustering model and to determine its strengths and limitations. The first experiment consists in producing a set of simulated data in order to verify the quality of the parameter estimations and of the final partitions. The second experiment checks if the proposed ICL-BIC criterion can detect the right number of row- and column-clusters.

To the best of our knowledge, there does not exist any other model that addresses the co-clustering of functional and binary data. For this reason, no comparison to competitors is possible. However, we could compare our work to other algorithms by considering the KPIs as continuous data instead of functional data. One of those algorithms could be the co-clustering of continuous and binary data proposed in [26] or the co-clustering algorithm of continuous data proposed in [18]. By considering the KPIs as continuous data, it is obvious that the functional information hidden in the KPIs will be lost. Moreover, the missing values and the strong correlation of the measurements within each KPI will pose a challenge for the parameters estimations which will impact the performance of the different experiments as proved in [32].

6.1 Experimental setup

Let's assume that the data is composed of $G = 4$ row-clusters, $H = 2$ functional column-clusters and $L = 2$ binary column-clusters. Data are simulated according to the MLBM (Eq 3). The parameters configuration of each block are described in Table 1. The mean curves of the functional blocks are composed of 96 values i.e. $\mu_1 = (\mu_1^d)$ where $1 \leq d \leq 96$ and $\mu_1^1 = \mu_1^2 = \ldots = \mu_1^{96}$ (similarly to μ_2, μ_3 and σ). By considering the configuration of Table 1, it is obvious that a traditional clustering approach will be unable to dissociate the block

structure whereas it could be possible with a co-clustering.

Table 1: Configuration of the simulated data

	h_1	h_2	l_1	l_2
g_1	μ_1, σ	μ_3, σ	α_1	α_2
g_2	μ_2, σ	μ_1, σ	α_1	α_2
g_3	μ_1, σ	μ_3, σ	α_2	α_1
g_4	μ_2, σ	μ_3, σ	α_2	α_1

Two levels of difficulty are considered. The first level, denoted by $Level+$, is set such that the co-clustering task is easy since the blocks are very well dissociated. The second level, denoted by $Level++$, provides a more challenging situation since the blocks are more difficult to dissociate. The parameters of each level are as follows:

- $Level+$: $\mu_1^d = 100, \mu_2^d = 20, \mu_3^d = -50, \sigma^d = 10, \alpha_1 = 0.2, \alpha_2 = 0.8$
- $Level++$: $\mu_1^d = 100, \mu_2^d = 20, \mu_3^d = -50, \sigma^d = 40, \alpha_1 = 0.4, \alpha_2 = 0.6$

In order to study the impact of the size of the data on the co-clustering results, two sizes are considered: datasets of size $(N, F, B) = (500, 100, 100)$, denoted by $DS1$, and datasets of size $(N, F, B) = (1000, 300, 300)$ denoted by $DS2$. For each scenario, a number s of datasets are generated. Then, Functional data analysis and MLBM are performed for each dataset.

6.2 Validation of the parameter estimation

In this first experiment, the purpose is to verify that the co-clustering algorithm is capable to determine the real blocks if it has the information about the true number of clusters i.e. $(G, H, L) = (4, 2, 2)$. B-spline basis is applied for the smoothing step where the number of basis functions $M = 15$. The number of principal components are chosen so that they cover at least 90% of the total variance. A comparison between SEM-Gibbs, CEM and VEM algorithms is held for the parameters estimation. The maximum number of iterations is fixed to 100 for the SEM-Gibbs algorithm and the burn-in period is fixed to 50. The performance of the co-clustering is judged over $s = 50$ simulations using the following metrics:

- ARI_r (respectively ARI_f and ARI_b): the adjusted rand index of the true row-clustering (respectively functional column-clustering and binary column-clustering) and its estimation,

- *Iterations*: the minimum and maximum number of iterations before the convergence over all the simulations. This metric is for CEM and VEM algorithms since they have convergence criteria. It is not considered for the SEM-Gibbs algorithm since the number of iterations if fixed to 100.

Table 2 presents the results over the 50 simulations where the level of difficulty is *Level+*. The results of $Level++$ are presented in Table 3. Each ARI metric is composed of two values: the mean and the standard deviation in parenthesis.

Table 2: Simulation results on data with *Level+*

	SEM-Gibbs		CEM		VEM	
Sizes	DS1	DS2	DS1	DS2	DS1	DS2
ARI_r	0.91 (0.17)	0.89 (0.03)	0.95 (0.11)	1 (0)	0.84 (0.18)	0.91 (0.16)
ARI_f	1 (0)	1 (0)	1 (0)	1 (0)	0.98 (0.14)	1 (0)
ARI_b	1 (0)	1 (0)	1 (0)	1 (0)	0.96 (0.19)	0.96 (0.20)
Iterations	100	100	14-30	13-16	15-100	17-100

For the *Level+*, as shown in Table 2, the co-clustering model is able to discern the different blocks. In terms of $ARIs$, when the dataset is bigger in size, the performances are, most of the time, better since the mean ARIs are closer to 1 with small standard deviations. Besides, CEM algorithm has the best performance followed by SEM-Gibbs and then VEM. In terms of rapidity, CEM algorithm is the best since it requires a less number of iterations before its convergence. SEM-Gibbs and VEM are the slowest since their computation times took approximately 4 times more than CEM algorithm.

For the $Level++$, as shown in Table 3, the co-clustering model has a more difficulty in discerning the different blocks. However, the performances are still good especially for the bigger dataset. CEM algorithm is less efficient at determining the clusters compared to

Table 3: Simulation Results on data with $Level++$

	SEM-Gibbs		CEM		VEM	
Sizes	DS1	DS2	DS1	DS2	DS1	DS2
ARI_r	0.74 (0.14)	0.89 (0.03)	0.78 (0.22)	0.85 (0.03)	0.94 (0)	0.89 (0.17)
ARI_f	0.96 (0.2)	1 (0)	1 (0)	1 (0)	0.98 (0.14)	1 (0)
ARI_b	0.99 (0.01)	1 (0)	0.79 (0.44)	0.97 (0.03)	1 (0)	0.95 (0.22)
Iterations	100	100	23-32	15-38	177-500	32-500

the two other algorithms. Besides, 14 simulations among the 50s led to a spurious solution with CEM algorithm. This problem did not occur neither with SEM-Gibbs nor with VEM algorithm. Moreover, these latter are better in terms of performance. However, VEM needs a bigger number of iterations before converging compared to the number of iterations of SEM-Gibbs. In terms of computation time, VEM algorithm needed the double of time needed by SEM-Gibbs algorithm. This latter took by itself the double of time needed by CEM algorithm.

Actually, the model is implemented with R programming language over a machine with 4 cores, 16 Go of RAM and 1.9 GHz processor. The execution time for all the algorithms depend to three attributes: (1) the complexity of the problem i.e. tests with $Level++$ take more time than tests with $Level+$; (2) the size of the data set i.e. $DS2$ needs more time than $DS1$ since it is bigger; (3) the algorithm used for the parameter estimation. The minimum execution time was about half an hour for CEM algorithm for $DS1$ and $Level+$ and up to 11 hours for VEM algorithm for $DS2$ and $Level++$. An implementation with C language may reduce the execution times.

Since in SEM-Gibbs algorithm, no convergence criterion is defined, it should be interesting to visually verify that after a burn-in period, the parameters are stable. Since SEM-Gibbs has more difficulty with the smaller dataset $DS1$ and with $Level++$, this scenario is chosen. Two examples are presented in Figure 4 to illustrate the estimation of π and of τ for this scenario. The parameters convergence are illustrated for all the simulations where each curve represents the values of the corresponding parameter over the 100 iterations and per

simulation. The colors of the curves do not have a specific meaning. They are used for the clarity of the figures. The visual verification showed that SEM-Gibbs algorithm succeed to achieve its stationary distribution. The different results are good enough to prove the efficiency of the proposed model at determining the right blocks.

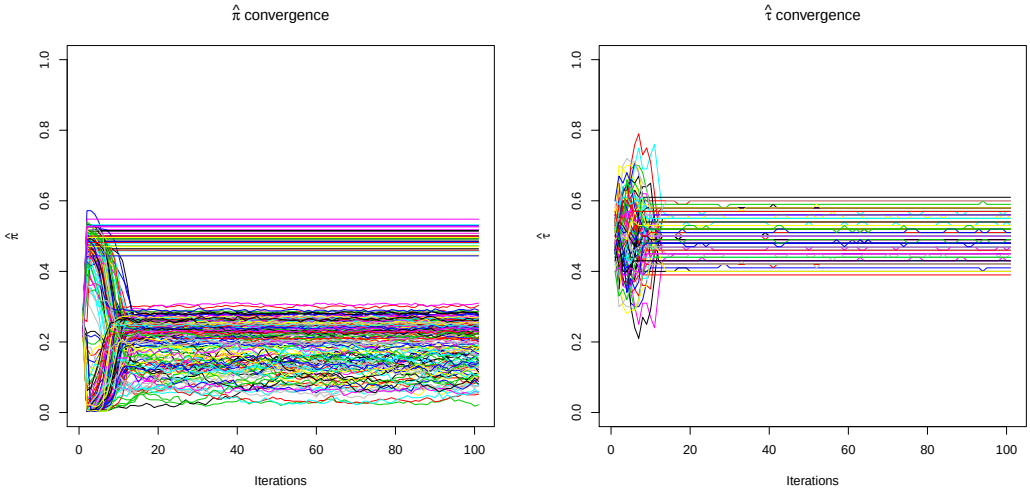

Figure 4: Convergence test of π and τ for the SEM-Gibbs algorithm

6.3 Choice of (G, H, L)

The aim of this section is to verify that the ICL-BIC criterion can detect the right numbers of row- and column-clusters. Therefore, $s = 20$ simulations are generated over datasets of size $(500, 100, 100)$ where the two level of difficulty are considered. Given $3 \leq G \leq 4$, $2 \leq H \leq 3$, $2 \leq L \leq 3$, we compute the number of selections of each combination (G, H, L). Since by the 40^{th} iteration, SEM-Gibbs algorithm has already reached its stationary state with $DS1$, the number of iterations is set to $q^{max} = 40$ with a burn-in period equal to 30 in order to gain in terms of execution time. The results are presented in Table 4.

When the level of difficulty is high, the ICL-BIC criterion doubts between $(3, 3, 2)$ and $(4, 3, 2)$. Regarding the *Level+*, the most selected combination is $(4, 3, 2)$. Although the number H of the functional column-clusters is over-estimated, the number of row- and binary column-clusters, G and L, are most of the time correct.

Table 4: Results of the number of co-clusters retained by ICL-BIC criterion over 20 simulations for *Level+* and 20 simulations for *Level++*

(G,H,L)	*Level+*	*Level++*
(3,2,2)	0	0
(3,2,3)	0	0
(3,3,2)	5	8
(3,3,3)	2	0
(4,2,2)	0	0
(4,2,3)	0	0
(4,3,2)	11	9
(4,3,3)	2	3

This experiment shows that when the model knows the real number of clusters, the co-clustering is efficient. However, although the ICL-BIC criterion has proved its efficiency in other co-clustering models, it seems to over-estimate H in this experiment.

6.4 Comparison of mixed co-clustering w.r.t two separate co-clusterings

Although in the previous configuration, achieving two independent co-clusterings, one for functional data and one for binary data can solve the problem, in other scenarios, it will be more difficult. For instance, a more complicated configuration is detailed in Table 5. Twenty data sets of *Level+* and of size $\{500, 300, 300\}$ are simulated with a comparison between mixed co-clustering approach w.r.t two separate co-clustering. Experimental results are detailed in Table 6. Each ARI metric is composed of two values: the mean and the standard deviation in parenthesis over the 20 simulations. The results illustrate that the binary co-clustering succeed to determine the binary column-clusters with a high ARI_b but it is unable to determine the row-clusters with a small ARI_r. As for the functional co-clustering, its low ARI_r and ARI_f prove that it is unable to discover both the row-clusters and the functional column-clusters as well. Only the mixed co-clustering succeed to discern the blocks with higher performance. This experiment illustrates that a functional co-clustering and a binary co-clustering are not able to discover the block structure in the data contrarily to the mixed co-clustering which proves the advantage of this approach

Table 5: Co-clusters configuration

	h_1	h_2	l_1	l_2
g_1	μ_1, σ	μ_3, σ	α_1	α_2
g_2	μ_3, σ	μ_1, σ	α_1	α_2
g_3	μ_1, σ	μ_3, σ	α_2	α_1
g_4	μ_3, σ	μ_1, σ	α_2	α_1

Table 6: Comparison between a mixed co-clustering and two separate co-clustering over 20 simulations

Performance	Mixed co-clustering	Functional co-clustering	Binary co-clustering
ARI_r	0.85 (0.03)	0.45 (0.0006)	0.49 (0.0005)
ARI_f	0.85 (0.03)	0.002 (6.03e-05)	-
ARI_b	0.85 (0.03)	-	1 (0)

7 Real data experiments

This section presents an application of the proposed co-clustering algorithm to mobile network monitoring. In 2017, the number of mobile subscribers worldwide hit 5 billion [33]. This is achieved thanks to the evolution of the mobile industry and especially to the introduction of the 4^{th} generation of mobile networks (Long Term Evolution, LTE). If the fact that two-third of the global population is connected to mobile services implies more profits to operators, it also means that operators face significant operational changes in terms of works and costs. Actually, the different mobile technologies will have to run for a long period of time in parallel and the network infrastructure as whole will be rather complex and heterogeneous. We talk about heterogeneous networks (Figure 5).

As mentioned before, the operation of the mobile network is achieved by observing KPIs and alarms information by human operators. The network management with manual efforts is time-consuming, expensive, error prone and requires a high degree of expertise. Besides, the fact that KPIs and alarms depend to each technology and to each constructor makes it impossible for a human operator to interpret them. Even with self-organizing networks

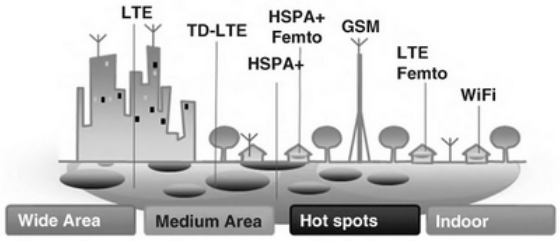

Figure 5: Example of a Heterogeneous Network

(SON, [4]) that aim to automate the network functions, the huge number of these network indicators are still an obstacle. One reason is that the automated functions still need the intervention of a human operator in order to verify their output before they directly change the network parameters.

As a consequence, it is essential to reduce the dimension of KPIs and of alarms while maintaining their information. The co-clustering of KPIs, alarms and cells will study the relationship between the behavior of different daily-captured KPIs and alarms for a specific geographical area. The block structure will help to create new network measures (macro-KPIs, and macro-alarms) specific to each cluster of cells having a similar behavior. In this way, the information induced by KPIs and by alarms can be summarized which offers a simple representation for engineers as well as self-organizing networks.

7.1 Description of the real data

The data are collected using an internal tool of Orange France. The observed cells are located in Paris, France. Paris is decomposed of 20 districts. The data are extracted from the 5^{th}, 6^{th} and 7^{th}, districts that are known as a touristic area. They are also extracted from the 14^{th} and 15^{th} districts that are known as a residential area. We focus our study on Long Term Evolution (LTE) sites and more specifically, to two frequency bands namely "LTE2600" and "LTE1800". In wireless telephony, a cell [1] is the geographical area covered by a cellular telephone transmitter. The transmitter facility itself is called the cell site. The

cell provided by a cell site can be from one mile to twenty miles in diameter, depending on terrain and transmission power. The data are extracted from 209 cells related to 44 sites. LTE KPIs are mainly classified into five classes. In this study, we focus on KPIs belonging to "Accessibility" family. It regroups measurements related to the network's ability to meet with the users demands for accessing to the different services. 70 KPIs are extracted with a granularity of 1 hour (therefore, each daily KPI contains 24 values). 47 alarms are addressed and they are all related to traffic information in the network. The extraction covers 35 days: from November 13 to December 17, 2017.

7.2 The model configuration

The overall size of the dataset is 7210 rows (= 206 cells $\times$ 35 days) with 117 features (70 KPIs and 47 alarms). Each KPI is described by 24 discrete values. Since the KPIs have different range of values, a normalization of the data seems inevitable. We started by using the min-max normalization technique on KPIs. Moreover, the dataset contains 10% of missing values. These latter are easily treated by applying the smoothing step. The number of basis functions $M = 15$ has been chosen empirically. A FPCA is then applied so that 90% of the total variance is retained. Given $2 \leq G \leq 5, 2 \leq H \leq 4, 2 \leq L \leq 3$, the ICL-BIC criterion is maximized with $(G, H, L) = (3, 3, 3)$ suggesting to dissociate the dataset in 3 row-clusters, 3 KPI column-clusters and 3 alarm column-clusters.

7.3 Experimental results

We present here the estimation results for the MLBM with $(G, H, L) = (3, 3, 3)$. The parameters estimation is achieved with a SEM-Gibbs inference. We chose SEM-Gibbs algorithm since it proved its efficiency in the simulation study: it did not generate empty clusters contrarily to CEM and it was more rapid than VEM with almost the same performance. Following the results on the simulated data of the previous section, we fixed the number of iterations of the SEM-Gibbs algorithm to 60 with a burn-in period equal to 40. Figure 6 illustrates the estimated mixing proportions per cluster i.e. $\hat{\pi}_g$, $\hat{\rho}_h$ and $\hat{\tau}_l$. Figure 7 illustrates

the estimated mean curve $\hat{\mu}_{gh}$ for the multivariate Gaussian distribution per block. It can be noticed that the proposed approach is able to resume the information since the initial dataset has a $(7210, 70, 47)$ structure, whereas the co-clustering result has a $(3, 3, 3)$ structure.

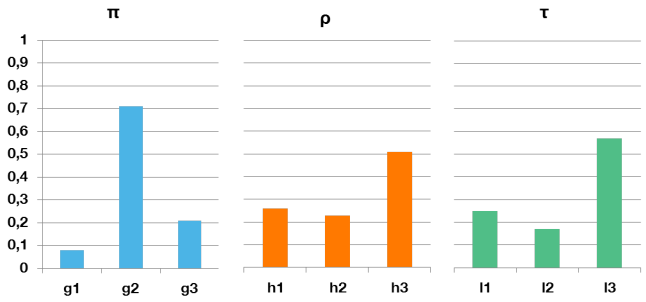

Figure 6: The estimated mixing proportions of the row-clusters, KPI-clusters and alarm-clusters

We start by digging in the data of each row-cluster in terms of geographical area and of days of observation. Figure 8 illustrates the distribution of the seven days of the week in each row-cluster. The equiprobability of the days is illustrated with the red line in each sub-figure.

The first row-cluster contains 5 sites related to the touristic area of the study zone. They also contains 6 sites related to the residential districts. These latter are mainly commerce attractions in this row-cluster. The days of the week are equiprobable in this first row-cluster with a bigger activity in Friday and Monday. The second row-cluster mainly contains sites related to the residential area (29 sites) with only 5 sites of the touristic area. These sites have a bigger activity in the week-ends compared to the work-days. The third row-cluster contains 19 sites of the residential area and 4 sites of the touristic area. The sites in this third row-cluster have stronger activity in the work-days compared to the week-ends. This classification of sites can be explained by a different behavior regarding the traffic volume and the accessibility demands in each cluster. The cells are grouped depending to their location and to their behavior in work-days and week-ends. This result may help the mobile operators for a better planning of the topological structure of their networks and for a better

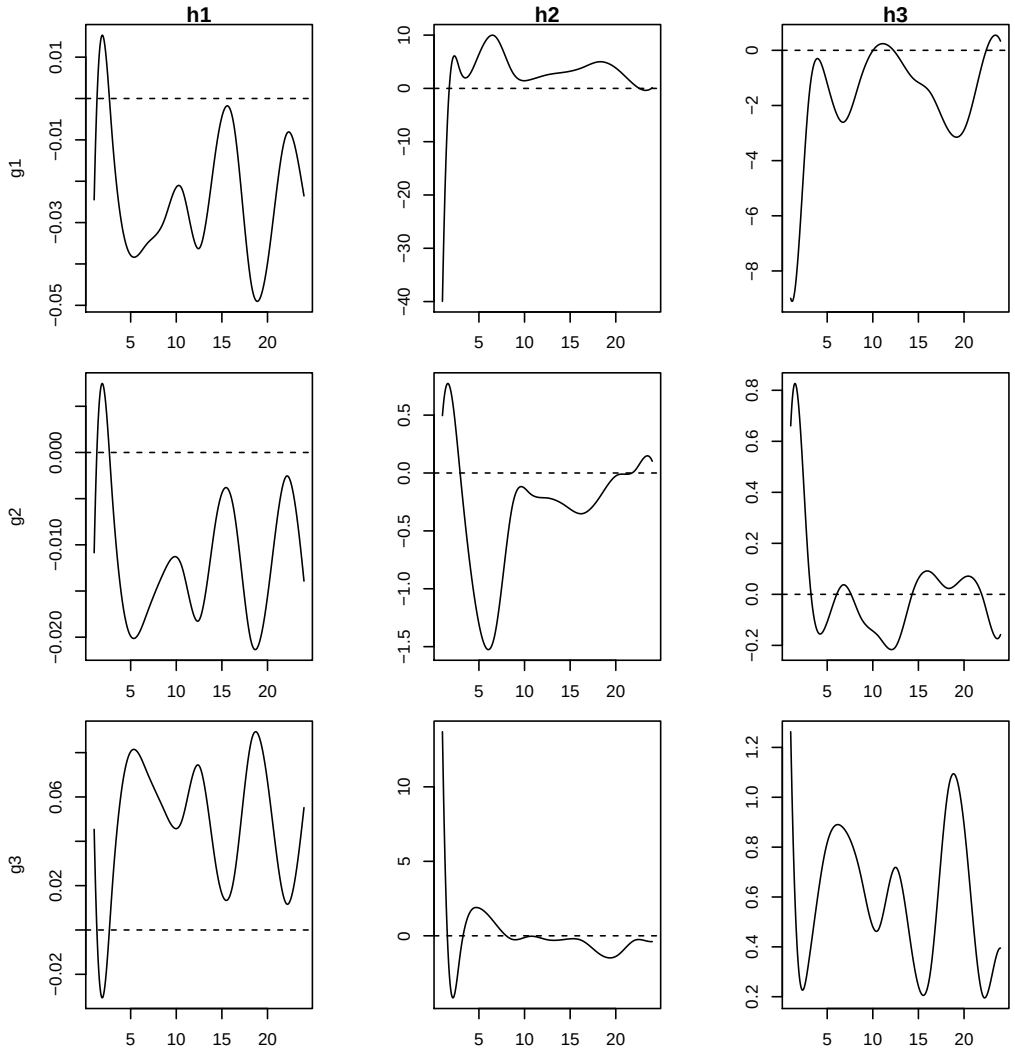

Figure 7: The estimated mean curve of each functional block

optimization. The network topology [1] is the arrangement of the various elements of a network in a geographical area. Since the environment in which the network is inserted is not static and changes might occur, optimization aims to constantly monitor the network parameters and its environment accordingly. It aims to guarantee that the network performs as efficiently as possible. As a result, the cells in a same row-cluster may have the same network parameters.

For the column-clusters related to KPIs, the first cluster contains KPIs related to the physical up-link control channel. The second cluster contains KPIs related to Radio resource

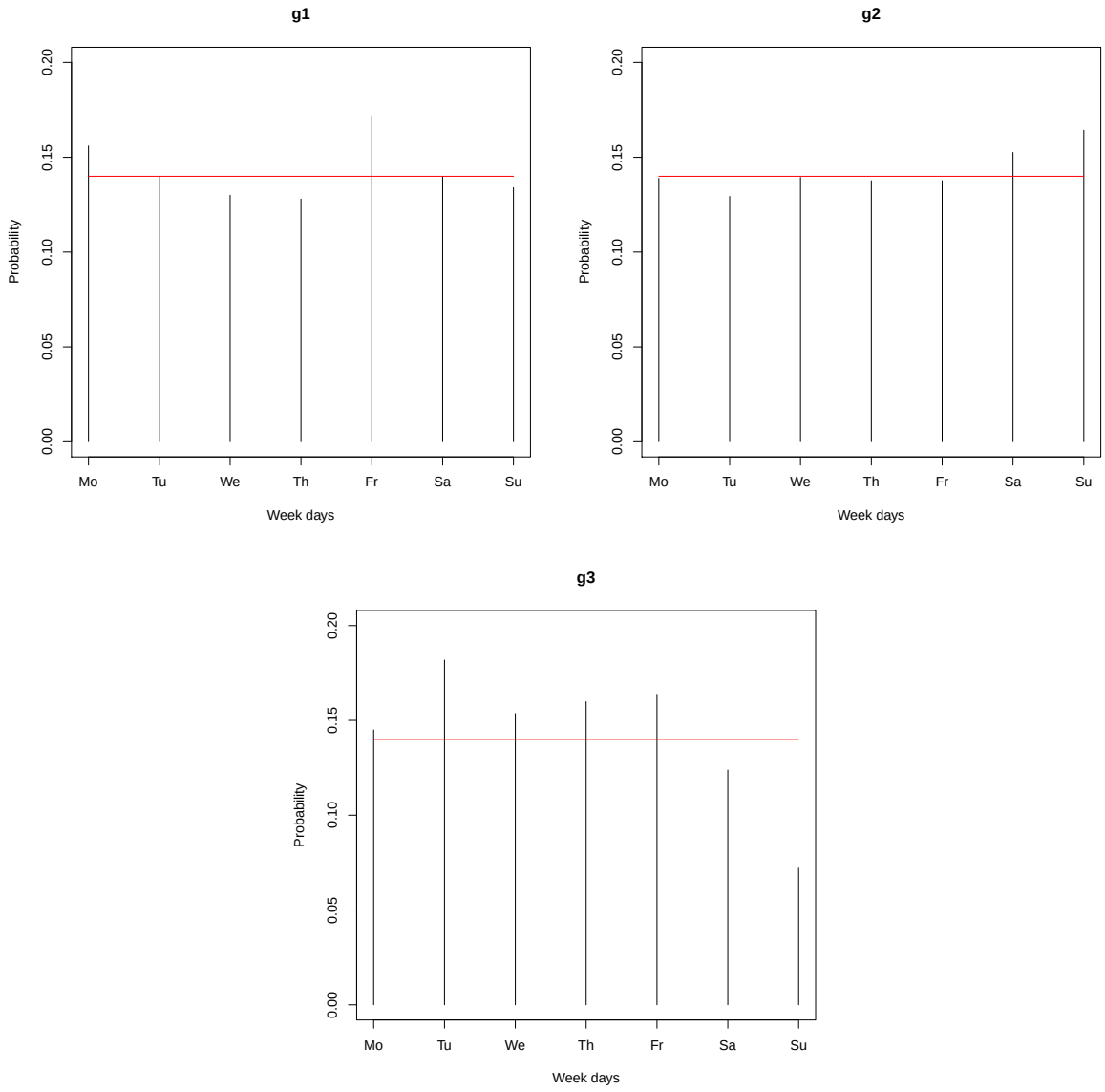

Figure 8: Days distribution in each row-cluster

control (RRC) re-establishment. The third cluster contains KPIs related to call setup success rates, to S1 interface and to random access channels. KPIs related to the modification of the EPC radio access bearer (ERAB) belong to the first and second clusters. However, KPIs related to RRC setup and establishment are grouped in the second and third clusters. This classification is interesting and logical since it groups the KPIs according to their sub-domains.

Regarding the column-clusters related to the alarms, it appears that these latter have been dissociated in terms of the probability to have an alert. If the probability p to have a problem when observing an alarm belongs to $[0.79, 0.99]$, the alarm belongs to the first alarm-cluster. However, if $p \in [0, 0.28]$, the alarm belongs to the second alarm-cluster. Finally, if $p \in [0.41, 0.72]$, the alarm belongs to the third alarm-cluster. This allows to create new network measurements that resume the information hidden in the alarms with reduced dimensions. These new measurements will be able to help network engineers and self-organizing networks in the management of mobile networks especially in the configuration and optimization processes.

8 Conclusion

Co-clustering of mixed data is a new research field that started to appeal to researchers recently. While functional data analysis is widely used in many real applications, the co-clustering of mixed data where some features have functional characteristics has never been proposed. In this paper, a model-based co-clustering for functional and binary data is introduced.

In the presented Multiple Latent Block Model, each block of curves is identified by the multivariate Gaussian distribution of the principal components of the curves. Each block of binary features is identified by a Bernoulli distribution. The model parameters can be estimated using a SEM-Gibbs, a CEM or a VEM algorithm. Through a simulation study, these algorithms have proven their efficiency in terms of parameters estimation, especially SEM-Gibbs that had better performances. The application of the model on mobile network monitoring has shown promising results. The co-clustering of cells, KPIs and alarms can help network engineers to create new network measurements and to better plan the network parameters. Since the network behaviour is complex and it exhibits behavioural dynamics on multiple seasonal timescales, different models related to different geographical areas should be trained separately. Besides, the training should be programmed periodically with updated

training data.

As future work, many improvements are possible. First, a deeper study can be held on the choice of the model by defining a new criterion to select the number of clusters. Rather than using the ICL-BIC criterion, it would be interesting to estimate the number of clusters together with the other parameters of the model. This can be achieved by using a fully Bayesian framework considering G, H and L as random variables as proposed in [34] for Non-negative matrix factorization. Second, in the proposed model, the FPCA is applied on the whole set of functional data, and consequently for all the curves of all the functional blocks. This is possible in our application since all the curves are approximated into the same basis of functions. In [35], the authors introduce a clustering algorithm for multivariate functional data in which each functional feature can be approximated into a different basis. Inspired by their work, the proposed work can be improved by considering different basis of functions. Third, a more sophisticated model could consider FPCA per blocks, as it is done in the clustering context in [36] for functional data or in [37] for a mixture of probabilistic PCA models. Fourth, a definition of a criteria could be investigated for the choice of the basis of functions and the number of coefficients in the context of unsupervised learning as discussed in [32]. Fifth, the likelihood of the model relies on a mixture of densities (for Gaussian distributions) and probabilities (for Bernoulli distributions). Hence an open problem arises regarding if these two kinds of likelihood are commensurable which is a common question when dealing with probabilistic models for mixed data. Finally, implementing an R package could be an interesting option for the prospective users of the model.

References

[1] A. Mishra, *Fundamentals of Cellular Network Planning and Optimisation: 2G/2.5G/3G... Evolution to 4G.* John Wiley & Sons, 2004.

[2] J. O. Ramsay and B. W. Silverman, *Functional data analysis*, 2nd ed., ser. Springer Series in Statistics. New York: Springer, 2005.

[3] J. O. Ramsay and C. J. Dalzell, "Some tools for functional data analysis," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 53, no. 3, pp. 539–572, 1991. [Online]. Available: http://www.jstor.org/stable/2345586

[4] 3GPP, "Telecommunication management; Self-Organizing Networks (SON); Concepts and requirements," 3rd Generation Partnership Project (3GPP), TS 32.500, Jul. 2008. [Online]. Available: http://www.3gpp.org/ftp/Specs/html-info/32500.htm

[5] B. Everitt, *An introduction to latent variable models / B.S. Everitt.* Chapman and Hall London ; New York, 1984.

[6] G. Chao and J. Sun, S.and Bi, "A survey on multi-view clustering," 12 2017.

[7] I. S. Dhillon, "Co-clustering documents and words using bipartite spectral graph partitioning," in *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '01. New York, NY, USA: ACM, 2001, pp. 269–274. [Online]. Available: http://doi.acm.org/10.1145/502512.502550

[8] B. Long, Z. Zhang, and P. S. Yu, "Co-clustering by block value decomposition," in *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, ser. KDD '05, 2005, pp. 635–640.

[9] A. Banerjee, I. Dhillon, J. Ghosh, S. Merugu, and D. S. Modha, "A generalized maximum entropy approach to bregman co-clustering and matrix approximation," *J. Mach. Learn. Res.*, vol. 8, pp. 1919–1986, Dec. 2007.

[10] F. Pan, X. Zhang, and W. Wang, "A general framework for fast co-clustering on large datasets using matrix decomposition," in *2008 IEEE 24th International Conference on Data Engineering*, April 2008, pp. 1337–1339.

[11] D. Ienco, C. Robardet, R. G. Pensa, and R.Meo, "Parameter-less co-clustering for star-structured heterogeneous data," *Data Min. Knowl. Discov.*, vol. 26, no. 2, pp. 217–254, 2013. [Online]. Available: https://doi.org/10.1007/s10618-012-0248-z

[12] B. Gao, T. y. Liu, and W. y. Ma, "Star-structured high-order heterogeneous data co-clustering based on consistent information theory," in *Sixth International Conference on Data Mining (ICDM'06)*, Dec 2006, pp. 880–884.

[13] G. Govaert and M. Nadif, "Clustering with block mixture models," *Pattern Recognition*, vol. 36, no. 2, pp. 463 – 473, 2003, biometrics.

[14] H. Shan and A. Banerjee, "Bayesian co-clustering," in *2008 Eighth IEEE International Conference on Data Mining*, Dec 2008, pp. 530–539.

[15] J. Wyse and N. Friel, "Block clustering with collapsed latent block models," *Statistics and Computing*, vol. 22, no. 2, pp. 415–428, 2012.

[16] J. Bennett and S. Lanning, "The netflix prize," in *In KDD Cup and Workshop in conjunction with KDD*, 2007.

[17] I. Dhillon, "Co-clustering documents and words using bipartite spectral graph partitioning," in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, ser. KDD '01. New York, NY, USA: ACM, 2001, pp. 269–274.

[18] M. Nadif and G. Govaert, "Model-Based Co-clustering for Continuous Data," in *ICMLA 2010, The Ninth International Conference on Machine Learning and Applications*, Washington, United States, Dec. 2010, pp. 1–6.

[19] C. Keribin, V. Brault, G. Celeux, and G. Govaert, "Model selection for the binary latent block model," in *20th International Conference on Computational Statistics (COMPSTAT 2012)*, Limassol, Cyprus, Aug. 2012, pp. 379–390. [Online]. Available: https://hal.inria.fr/hal-00924210

[20] G. Govaert and M. Nadif, *Co-Clustering*. Wiley-ISTE, 2013.

[21] C. Keribin, V. Brault, G. Celeux, and G. Govaert, "Estimation and selection for the latent block model on categorical data," *Statistics and Computing*, vol. 25, no. 6, pp. 1201–1216, 2014.

[22] J. Jacques and C. Biernacki, "Model-Based Co-clustering for Ordinal Data," Jan. 2017, working paper or preprint. [Online]. Available: https://hal.inria.fr/hal-01448299

[23] Y. Ben Slimen, S. Allio, and J. Jacques, "Model-Based Co-clustering for Functional Data," Dec. 2016, working paper or preprint. [Online]. Available: https://hal.inria.fr/hal-01422756

[24] F. Chamroukhi and C. Biernacki, "Model-Based Co-Clustering of Multivariate Functional Data," in *ISI 2017 - 61st World Statistics Congress*, Marrakech, Morocco, Jul. 2017. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01653782

[25] C. Bouveyron, L. Bozzi, J. Jacques, and F.-X. Jollois, "The functional latent block model for the co-clustering of electricity consumption curves," *Journal of the Royal Statistical Society, Series C*, in press.

[26] A. Bouchareb, M. Boullé, and F. Rossi, "Co-clustering de données mixtes à base des modèles de mélange," in *Conférence Internationale Francophone sur l'Extraction et gestion des connaissances (EGC 2017)*, ser. Actes de la 17ème Conférence Internationale Francophone sur l'Extraction et gestion des connaissances (EGC'2017), F. Gandon and G. Bisson, Eds., vol. RNTI-E-33, Grenoble, France, Jan. 2017, pp. 141–152. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01469546

[27] F. Yao, H. G. Muller, and J. L. Wang, "Functional data analysis for sparse longitudinal data," *Journal of the American Statistical Association*, vol. 100, no. 470, pp. 577–590, 2005.

[28] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society*, vol. 39, no. 1, pp. 1–38, 1977.

[29] C. Keribin, G. Govaert, and G. Celeux, "Estimation d'un modèle à blocs latents par l'algorithme SEM," in *42èmes Journées de Statistique*, Marseille, France, France, 2010.

[30] G. Schwarz, "Estimating the dimension of a model," *Ann. Statist.*, vol. 6, pp. 461–464, 1978.

[31] C. Biernacki, G. Celeux, and G. Govaert, "Assessing a mixture model for clustering with the integrated completed likelihood," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 7, pp. 719–725, 2001.

[32] J. Jacques and C. Preda, "Functional data clustering: a survey," *Advances in Data Analysis and Classification*, vol. 8, no. 3, pp. 231–255, 2014.

[33] Ericsson, "Ericsson Mobility Report 2017," White paper, 2017. [Online]. Available: https://www.ericsson.com/assets/local/mobility-report/documents/2017/ericsson-mobility-report-june-2017.pdf

[34] H. Kim, D. Kim, and S. Y. Bang, "An efficient model order selection for pca mixture model," *Pattern Recogn. Lett.*, vol. 24, no. 9-10, pp. 1385–1393, Jun. 2003.

[35] J. Jacques and C. Preda, "Model-based clustering of multivariate functional data," *Computational Statistics and Data Analysis*, vol. 71, pp. 92–106, 2014.

[36] C. Bouveyron and J. Jacques, "Model-based clustering of time series in group-specific functional subspaces," *Advances in Data Analysis and Classification*, vol. 5, no. 4, pp. 281–300, 2011.

[37] M. E. Tipping and C. Bishop, "Mixtures of principal component analyzers," *Neural Computation*, vol. 11, no. 2, pp. 443–482, 1999.