Model-Based Co-clustering for Functional Data

Yosra Ben Slimen, Sylvain Allio, Julien Jacques

Model-Based Co-clustering for Functional Data

Yosra Ben Slimen[a,b], Sylvain Allio[a], Julien Jacques[b]

[a] *Orange Labs, Belfort, France*
[b] *Université de Lyon, Université Lyon 2, ERIC EA3083, Lyon, France*

Abstract

In order to provide a simplified representation of key performance indicators for an easier analysis by mobile network maintainers, a model-based co-clustering algorithm for functional data is proposed. Co-clustering aims to identify block patterns in a data set from a simultaneous clustering of rows and columns. The algorithm relies on the latent block model in which each curve is identified by its functional principal components that are modeled by a multivariate Gaussian distribution whose parameters are block-specific. These latter are estimated by a stochastic EM algorithm embedding a Gibbs sampling. In order to select the numbers of row- and column-clusters, an ICL-BIC criterion is introduced. In addition to be the first co-clustering algorithm for functional data, the advantage of the proposed model is its ability to extract the hidden double structure induced by the data and its ability to deal with missing values. The model has proven its efficiency on simulated data and on a real data application that helps to optimize the topology of 4G mobile networks.

Keywords: co-clustering, functional data, SEM-Gibbs algorithm, latent block model, ICL-BIC criterion, mobile network, key performance indicators.

1. Introduction

With the introduction of new technologies and services in mobile networks, the complexity of these latter have increasingly grown creating an heterogeneous

Email addresses: `yosra.benslimen@orange.com` (Yosra Ben Slimen), `sylvain.allio@orange.com` (Sylvain Allio), `julien.jacques@univ-lyon2.fr` (Julien Jacques)

environment where different architectures (micro-, macro-, pico-, femto-cells) and different radio access technologies (GSM, UMTS, LTE ...) coexist. New difficulties have been resulted such as network management, optimization, troubleshooting and planning. Automated networks have also been created such as the self-organizing networks [1] that demand new techniques for self-healing and self-optimization. Therefore, mobile operators need to deal with these new challenges in order to provide a top quality of services without increasing costs [2].

The quality of services is measured by data that are generated from multiple sources including key Performance Indicators (KPI) which are measurements collected from network elements such as transceivers, cells, or sites [2]. They are defined by mathematical formulas derived from different counters and computed periodically from the network with different temporal granularities (weekly, daily, hourly or less). For instance, Figure 1 illustrates a sample of $p = 30$ KPIs for $n = 20$ daily observations. From a statistical perspective, these KPIs are considered as functional data [3] which is a type of data that has recently appealed to researchers since they were for longtime inaccessible for statistics. However, with the advance of modern technology, more and more data are being recorded continuously during a time interval (or intermittently at numerous discrete time points). They become very frequent, not only in the telecommunication field, but in numerous other domains like medicine, economics and chemometrics (see [3] for an overview). Functional data is the observation (sample path) of a stochastic process $X = \{X(t), t \in T\}$, where T can be for instance a time interval, or any other continuous subset.

The KPIs may be specific to each radio access technology (GSM, UMTS, LTE,...) and to each constructor (Huawei, Ericsson,...). Therefore, as the number of technologies, services, cell types, and constructors grows, the KPIs observed by the support team become enormous and they may need to be analyzed over a large period (several weeks or months). Hence, on one hand, observing all the KPIs makes their daily analysis by the engineers a difficult task and also their treatment by the self-organizing networks more greedy in

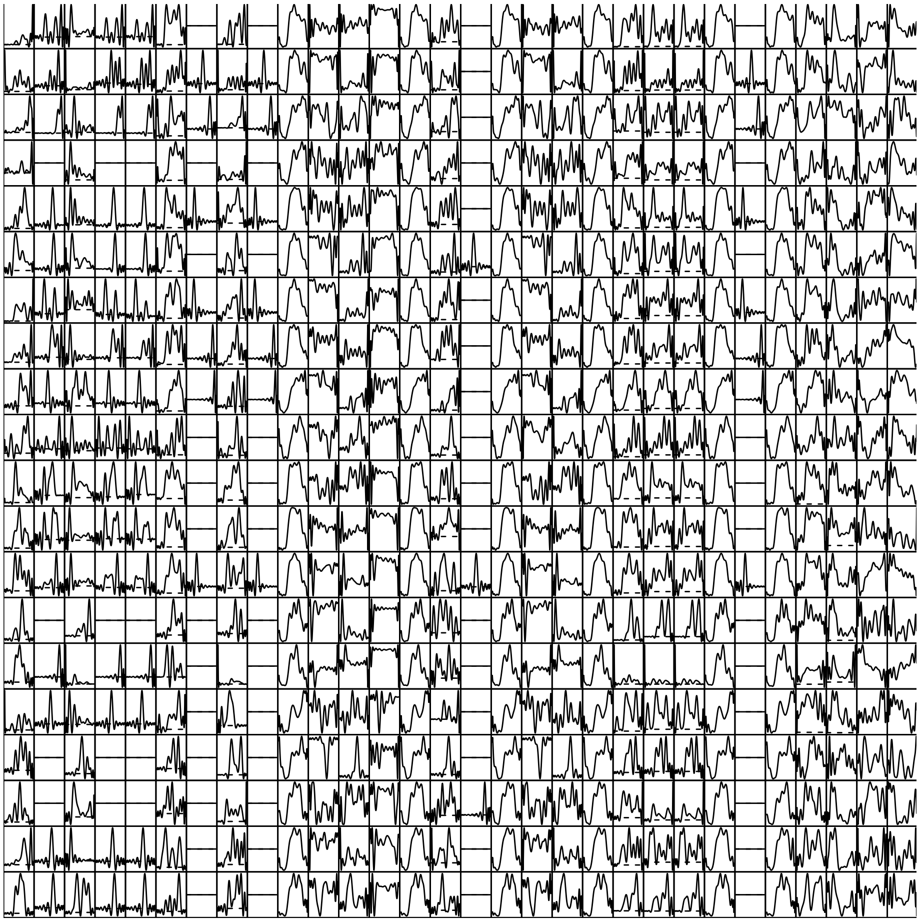

Figure 1: An example of functional data set composed of 20 observations and 30 KPIs

terms of time and memory. On the other hand, ignoring some KPIs risks to decrease the performance. Therefore, in order to help the network maintainers in their job, our work aims to provide a simplified representation of the daily evolutions of KPIs.

Since the number of days of observations and the number of KPIs are large, a co-clustering algorithm will be designed in order to cluster both of them. Thus, crossing days-clusters and KPIs-clusters will lead to define homogeneous blocks of data, containing daily KPI observations having the same behavior.

Let $\mathbf{x} = (x_{ij})_{i \in I, j \in J}$, where I is a set of n observations (rows, objects) and J is a set of p attributes (columns, features). The basic idea of co-clustering can be seen as making permutations of objects and variables in order to draw a correspondence structure on $I \times J$. For illustration, consider Figure 2 that represents a binary data set of $n = 10$ observations $I = \{A, B, \ldots, J\}$ and of $p = 7$ binary attributes $J = \{1, 2, \ldots, 7\}$. By permuting the rows and columns, the data set is re-organized into a set of 3×3 co-clusters, defining 9 blocks of homogeneous data.

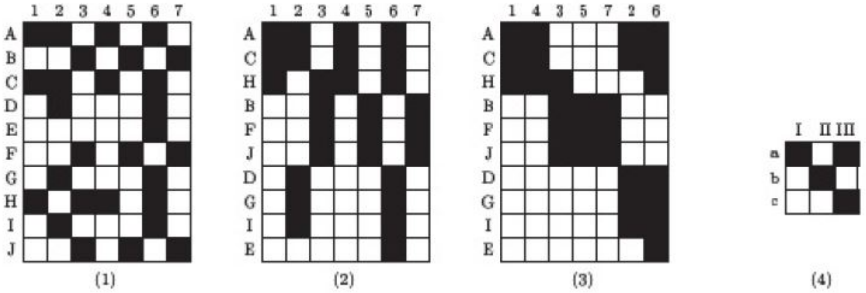

Figure 2: Binary data set (1), data re-organized by partition on I (2), by partitions on I and J simultaneously (3) and summary matrix (4)

Co-clustering has successfully proven its efficiency in many applications such as recommendation systems [4] or text mining [5]. According to [6], two families of the block co-clustering techniques can be distinguished, namely: (a) the matrix reconstruction based family in which the problem is formulated as a matrix approximation using dissimilarity metrics and a set of constraints (see [7] for an example) (b) the model-based family that uses probabilistic models in order to define the blocks [8, 9, 10]. Many types of data have been treated when dealing with co-clustering such as categorical data [11], ordinal data [12, 13], discrete [14] or continuous data [15]. However, there does not exist, to the best of our knowledge, co-clustering algorithm for functional data even though a lot of clustering algorithms have been proposed for this type of data (see [16] for a survey).

The purpose of this paper is to propose a co-clustering technique based on the Latent Block Model (LBM, [14]) that we adapt for functional data. The LBM assumes local independence, *i.e.* the $n \times p$ curves are assumed to be independent once the row and column partitions are fixed. Since the notion of probability distribution for functional data is not well defined [17], a Functional Principal Components Analysis (FPCA, [3]) is used in order to plug the functional data into a finite-dimensional space. This strategy is frequent in functional data clustering and has proven its efficiency [18, 19, 20, 21]. Once each curve is being identified by its principal components, the probability distribution of these latter can be modeled by a multivariate (Gaussian) distribution with block-specific parameters.

The paper is organized as follows. Section 2 introduces the notations, the transformation of the discrete observations of curves into functional data and functional principal components analysis. The definition of the latent block model for functional data is described in Section 3. Given the numbers of row-clusters and column-clusters, a Stochastic EM algorithm embedding a Gibbs sampling (SEM-Gibbs) is proposed in Section 4 for the estimation of the model parameters. Since the number of row- and column-clusters must be estimated in practice, a strategy based on an ICL-BIC criterion is proposed in Section 5 that may be used to determine these numbers. The behavior of the model is studied on simulated data in Section 6. Finally, Section 7 presents an application of the co-clustering model on real data of mobile networks extracted within Orange Labs, France.

2. From discrete data to functional principal components

The data under study are a sample of n observations. Each observation is described by a set of p curves (functional features). The statistical model underlying data, represented by multivariate curves, is a stochastic process with continuous time:

$$\mathbf{X} = \{\mathbf{X}(t)\}_{t\in[0,T]} \quad \text{with} \quad \mathbf{X}(t) = (X_1(t), \ldots, X_p(t))' \in \mathbb{R}^p, \quad p \geq 2.$$

One way to explore such data is to perform a clustering of multivariate curves [21]. In the present work, the goal is to go further by also clustering together the functional features having the same behavior.

As previously mentioned, the latent block model that we propose assumes a probability distribution on the functional principal components of the curves. This section explains how to transform the discrete observations into functional data and then, how to perform a FPCA.

2.1. Transformation of the observed discrete curves

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, a smoothing may be considered. Smoothing methods consider that the true curve belongs to a finite-dimensional space spanned by some basis of functions such as trigonometric functions, B-splines or wavelets (see [3] for a detailed study). Smoothing assumes that each observed curve x_{ij} $(1 \leq i \leq n,\ 1 \leq j \leq p)$ can be expressed as a linear combination of basis functions $\{\phi_{j\ell}\}_{\ell=1,\ldots,M_j}$:

$$x_{ij}(t) = \sum_{\ell=1}^{M_j} a_{ij\ell}\phi_{j\ell}(t), \quad t \in [0,T], \tag{1}$$

where $\{a_{ij\ell}\}_{\ell=1,\ldots,M_j}$ are the basis expansion coefficients. These coefficients can be estimated by least square smoothing for instance [3]. In this work, due to the nature of the KPIs under study, the same basis $\{\phi_\ell\}_{\ell=1,\ldots,M}$ is used for all the functional features. The choice of the basis as well as the number of basis functions strongly depends to the nature of data. Hence, they can be set empirically.

2.2. Principal components analysis for functional data

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 principal components analysis for functional data (FPCA, [3]). It consists in computing the principal components C^h and principal factors f^h of the Karhunen-Loeve expansion:

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

When curves are assumed to be decomposed into a finite basis of function (1), FPCA consists in a usual PCA of the basis expansion coefficients using a metric defined by the inner products between the basis functions. 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.

In this work, in order to project all the data onto the same FPCA space, functional principal components analysis is applied on the whole data set of curves $\mathbf{x}$, without distinction between curves from different observations or curves from different features. Moreover, in order to reduce the dimensionality of the problem, only the first $m \leq M$ principal components are considered. This number m is fixed empirically so that the principal components express a given part of the total variance.

3. Latent block model for functional data

This section describes the co-clustering technique for functional data based on the latent block model.

Let $\mathbf{x} = (x_{ij})_{1 \leq i \leq n, 1 \leq j \leq p}$ be the matrix of curves $(x_{ij} : x_{ij}(t),\ t \in [0, T])$ whose rows are observations and whose columns are the functional features. After functional principal components analysis, each curve x_{ij} is summarized by its principal components $\mathbf{c}_{ij} = (c_{ij}^h)_{1\leq h\leq m}$. Let $\mathbf{c} = (c_{ij}^h)_{1\leq i\leq n, 1\leq j\leq p, 1\leq h\leq m}$ denotes the set of all the principal components.

The objective of co-clustering is to divide the data into K_r row-clusters and K_c column-clusters. The clusters are mixed in varying proportions denoted by α_{k_r} for the row-mixing proportion of the row-cluster k_r and by β_{k_c} for the column-mixing proportion of the column-cluster k_c. Hence, the data are summed up in $K_r \times K_c$ blocks. Each block contains data belonging to the same Gaussian distribution $\mathcal{N}(\mu_{k_r k_c}, \Sigma_{k_r k_c})$.

Let $\mathbf{v}$ be the row-clustering matrix $\mathbf{v} = (v_{ik_r})_{i=1\ldots n, k_r=1\ldots Kr}$ with $v_{ik_r} = 1$ if row i belongs to the cluster k_r, 0 otherwise. Let $\mathrm{p}(v_{ik_r})$ be the probability of row i to belong to the cluster k_r, with the constraint that $\sum_{k_r=1}^{K_r} \mathrm{p}(v_{ik_r}) = 1$. In the same way for the column partitioning, let $\mathbf{w} = (w_{jk_r})_{j=1\ldots p, k_c=1\ldots K_c}$ be the column-clustering matrix and let $\mathrm{p}(w_{jk_c})$ be the probability of column j to belong to the cluster k_c. In the following, the straightforward ranges for i, j, h, k_r and k_c will be omitted for simplicity of notations.

The latent block model for functional data is defined by its density:

$$\mathrm{p}(\mathbf{x}; \theta) = \sum_{\mathbf{v}\in V} \sum_{\mathbf{w}\in W} \mathrm{p}(\mathbf{v}; \theta)\mathrm{p}(\mathbf{w}; \theta) f(\mathbf{c}|\mathbf{v}, \mathbf{w}; \theta) \tag{3}$$

where,

- V is the set of all possible partitions of the rows into K_r groups, W is the set of all possible partitions of the columns into K_c groups,
- $\mathrm{p}(\mathbf{v}; \theta) = \prod_{ik_r} \alpha_{k_r}^{v_{ik_r}}$; $\mathrm{p}(\mathbf{w}; \theta) = \prod_{jk_c} \beta_{k_c}^{w_{jk_c}}$,
- $f(\mathbf{c}|\mathbf{v}, \mathbf{w}; \theta) = \prod_{ijk_rk_c} \mathrm{p}(\mathbf{c}_{ij}; \mu_{k_rk_c}, \Sigma_{k_rk_c})^{v_{ik_r} w_{jk_c}}$ with:
  - $\mathrm{p}(\cdot; \mu_{k_rk_c}, \Sigma_{k_rk_c})$ is the m-variate Gaussian density with mean $\mu_{k_rk_c} = (\mu_{hk_rk_c})_{1\leq h\leq m}$ and covariance matrix $\Sigma_{k_rk_c}$,
- $\theta = (\alpha_{k_r}, \beta_{k_c}, \mu_{k_rk_c}, \Sigma_{k_rk_c})_{1\leq k_r\leq K_r, 1\leq k_c\leq K_c}$, is the whole set of the model parameters.

The parameter θ of the latent block model is to be estimated, and we propose to use maximum likelihood inference.

4. Inference via a SEM-Gibbs algorithm

The objective of this section is to determine the latent blocks of the model by estimating the parameter θ using maximum likelihood inference. The maximum likelihood estimation of θ results in an optimization of the observed log-likelihood $l(\theta; \mathbf{x}) = \ln \mathrm{p}(\mathbf{x}; \theta)$ where $\mathrm{p}(\mathbf{x}; \theta)$ is defined by (3). The usual way used to maximize the log-likelihood in the presence of missing observations is the EM algorithm [22]. However, since the LBM involves a double missing structure, $\mathbf{v}$ and $\mathbf{w}$, the maximum likelihood inference is computationally infeasible with an EM algorithm. For instance, with a data matrix of size 20×20 and with $K_r = K_c = 2$, computing (3) requires $K_r^n \times K_c^p \approx 10^{12}$ terms. For this work, we choose to use 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 [23]. A second advantage of SEM-Gibbs algorithm is that it is expected to be insensitive to its initial values. Starting from an initial value of the parameter $\theta^{(0)}$ and of the missing data $\mathbf{w}^{(0)}$, the q^{th} iteration of the partial SEM-Gibbs alternates the following SE-Gibbs and M steps.

SE-Gibbs step. Execute a small number (at least 1) of successive iterations of the two following steps:

1. Generate the row partition $v_{ik_r}^{(q+1)} | \mathbf{c}, \mathbf{w}^{(q)}$ for all $1 \leq i \leq n, 1 \leq k_r \leq K_r$ according to:

$$\mathrm{p}(v_{ik_r} = 1 | \mathbf{c}, \mathbf{w}^{(q)}; \theta^{(q)}) = \frac{\alpha_{k_r}^{(q)} f_{k_r}(\mathbf{c}_i | \mathbf{w}^{(q)}; \theta^{(q)})}{\sum_{k_r'} \alpha_{k_r'}^{(q)} f_{k_r'}(\mathbf{c}_i | \mathbf{w}^{(q)}; \theta^{(q)})}$$

where $\mathbf{c}_i = (c_{ij}^h)_{j,h}$ and $f_{k_r}(\mathbf{c}_i | \mathbf{w}^{(q)}; \theta^{(q)}) = \prod_{jk_c} \mathrm{p}(\mathbf{c}_{ij}; \mu_{k_r k_c}^{(q)}, \Sigma_{k_r k_c}^{(q)})^{w_{jk_c}^{(q)}}$.

2. Generate the column partition $w_{jk_c}^{(q+1)}|\mathbf{c},\mathbf{v}^{(q+1)}$ for all $1 \leq j \leq p, 1 \leq k_c \leq K_c$ according to:

$$\mathrm{p}(w_{jk_c} = 1|\mathbf{c},\mathbf{v}^{(q+1)};\theta^{(q)}) = \frac{\beta_{k_c}^{(q)} f_{k_c}(\mathbf{c}_j|\mathbf{v}^{(q+1)};\theta^{(q)})}{\sum_{k_c'} \beta_{k_c'}^{(q)} f_{k_c'}(\mathbf{c}_j|\mathbf{v}^{(q+1)};\theta^{(q)})}$$

where $\mathbf{c}_j = (c_{ij}^h)_{i,h}$ and $f_{k_c}(\mathbf{c}_j|\mathbf{v}^{(q+1)};\theta^{(q)}) = \prod_{ik_r} \mathrm{p}(\mathbf{c}_{ij};\mu_{k_rk_c}^{(q)},\Sigma_{k_rk_c}^{(q)})^{v_{ik_r}^{(q+1)}}$.

M step. Estimate $\theta^{(q+1)}$ given $\mathbf{v}^{(q+1)},\mathbf{w}^{(q+1)}$.

For the mixing proportions, the estimations are:

$$\alpha_{k_r}^{(q+1)} = \frac{1}{n}\sum_i v_{ik_r}^{(q+1)} \qquad \text{and} \qquad \beta_{k_c}^{(q+1)} = \frac{1}{p}\sum_j w_{jk_c}^{(q+1)},$$

and for the means and the covariances of each block:

$$\mu_{k_rk_c}^{(q+1)} = \frac{1}{n_{k_rk_c}^{(q+1)}}\sum_i\sum_j \mathbf{c}_{ij}^{v_{ik_r}^{(q+1)}w_{jk_c}^{(q+1)}} \qquad \text{and} \qquad \Sigma_{k_rk_c}{}^{(q+1)} = \frac{1}{n_{k_rk_c}^{(q+1)}} S_{k_rk_c}^{(q+1)}$$

where $n_{k_rk_c}^{(q+1)} = \sum_{ij} v_{ik_r}^{(q+1)} w_{jk_c}^{(q+1)}$ and ,
$S_{k_rk_c}^{(q+1)} = \sum_{ij}((\mathbf{c}_{ij} - \mu_{k_rk_c}^{(q+1)})^t(\mathbf{c}_{ij} - \mu_{k_rk_c}^{(q+1)}))^{v_{ik_r}^{(q+1)}w_{jk_c}^{(q+1)}}$.

Choosing the parameters estimation and the final partition. The SE-Gibbs and M steps are iterated for a given number of iterations. After a burn-in period, the final estimation $\hat{\theta}$ is defined by the mean of the sample distribution. For the final partitions, since $\mathbf{v}$ and $\mathbf{w}$ depend to each other, a new sampling of $\mathbf{v},\mathbf{w}|\hat{\theta}$ is simulated by successive SE-Gibbs steps with $\theta = \hat{\theta}$. Then, every v_{ik_r} (respectively w_{jk_c}) is obtained by computing the mode of the new sampling distribution.

5. Choice of the number of clusters

In the previous section, the numbers of clusters, K_r in rows and K_c in columns, are supposed to be known. However, in real applications, it may not be easy to precisely guess these numbers. Therefore, we propose to use the ICL-BIC criterion developed for continuous data in [24]:

$$\text{ICL-BIC}(K_r,K_c) = \log \mathrm{p}(\mathbf{x},\hat{\mathbf{v}},\hat{\mathbf{w}};\hat{\theta}) - \frac{K_r-1}{2}\log n - \frac{K_c-1}{2}\log p - \frac{\nu}{2}\log(np)$$

where:

- $\hat{\mathbf{v}}, \hat{\mathbf{w}}$ and $\hat{\theta}$ are the respective estimations of the row partition, the column partition and the model parameters obtained at the end of the estimation algorithm,
- The complete-log likelihood is given by:

$$\begin{aligned}\log \mathrm{p}(\mathbf{x}, \hat{\mathbf{v}}, \hat{\mathbf{w}}; \hat{\theta}) = &\sum_{ik_r} \hat{v}_{ik_r} \log \hat{\alpha}_{k_r} + \sum_{jk_c} \hat{w}_{jk_c} \log \hat{\beta}_{k_c} \\ &+ \sum_{ijk_rk_c} \hat{v}_{ik_r} \hat{w}_{jk_c} \log \mathrm{p}(\mathbf{c}_{ij}; \hat{\mu}_{k_rk_c}, \hat{\Sigma}_{k_rk_c}),\end{aligned}$$

- ν is the number of continuous parameters of the LBM: $\nu = K_r K_c \left(m + \frac{m(m+1)}{2} \right)$.

Strategy of exploration. In the co-clustering context, exploring all the possible combinations of values of $\{K_r, K_c\}$ (with $K_r \leq K_r^{max}$ and $K_c \leq K_c^{max}$) becomes rapidly computationally demanding. Inspired by the strategy developed in [25], a greedy search algorithm is proposed. It allows to only explore a relevant subspace of possible combinations of $\{K_r, K_c\}$ (see Algorithm 1). At each step, the algorithm consists in computing the ICL-BIC criterion of the models obtained with one additional cluster, either in row or in column. The solution with the best ICL-BIC criterion is retained and the previous step is repeated until the ICL-BIC criterion does no longer increase.

```
initialization: ${K_r, K_c} = {2, 2}$;
repeat
    $\{K_r^{old}, K_c^{old}\} = \{K_r, K_c\}$;
    if $\textit{ICL-BIC}(K_r + 1, K_c) \geq \textit{ICL-BIC}(K_r, K_c + 1)$ and $K_r < K_r^{max}$ then
        $K_r = K_r + 1$;
        else if $\textit{ICL-BIC}(K_r + 1, K_c) < \textit{ICL-BIC}(K_r, K_c + 1)$ and $K_c < K_c^{max}$ then
            $K_c = K_c + 1$;
        end
    end
until ($K_r \geq K_r^{max}$ and $K_c \geq K_c^{max}$) or $\textit{ICL-BIC}(K_r, K_c) \leq \textit{ICL-BIC}(K_r^{old}, K_c^{old})$;
```

Algorithm 1: Greedy search for exploring the numbers of clusters

6. Numerical experiments

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 to test the behavior of SEM-Gibbs algorithm in terms of convergence, in order to choose the best number of iterations. The second experiment verifies the quality of the parameter estimations and of the final partitions resulted by SEM-Gibbs algorithm. The third experiment checks if the greedy search algorithm can detect the right number of row- and column-clusters.

6.1. Experimental setup

The experimental setup is composed of four steps. Let's assume that the data set is divided into 9 blocks ($K_r = K_c = 3$), the first step consists in assigning, to each block, a mean curve with discrete values. The 9 mean curves are chosen among the daily evolution of the KPIs under study, presented in the next section. They are also chosen according to two scenarios so that two families of experiences are considered with different levels of difficulty. The first family ($Family1$) is set such that the co-clustering task is less difficult: only 2 couples among the 9 mean curves are close. The second family ($Family2$) provides a more challenging situation: the mean curves of the blocks are set such that 4 couples among them are close.

In the second step, each mean curve is smoothed using B-splines as follows: $\mu_{k_r k_c}(t) = \sum_{l=1}^{M} a_{k_r k_c l} \phi_l(t)$, where the number of basis functions is empirically set to $M = 10$. Figure 3 (respectively Figure 4) illustrates the $Family1$ (respectively $Family2$) and their corresponding smoothing.

The third step consists in simulating curves for each block. By using the basis expansion coefficients, each curve is simulated as follows : $x(t) = \sum_l a_l \phi_l(t)$ where $a_l \sim N(a_{k_r k_c l}, 10)$. These simulations are used to generate a number s of data sets of size $n \times p$, where $n, p \in \{50, 100, 500\}$.

In the fourth step, a FPCA is applied for each data set. The number of principal components m is chosen so that they cover at least 80% of the information. Figure 5 illustrates the mean of FPCA variance proportions after $s = 50$ simulations of data sets of size 100×100. For all the samples' sizes and the families of simulation, 80% of the information is summarized by the first 3 principal components. Consequently, for what follows, the number of principal components is set to $m = 3$.

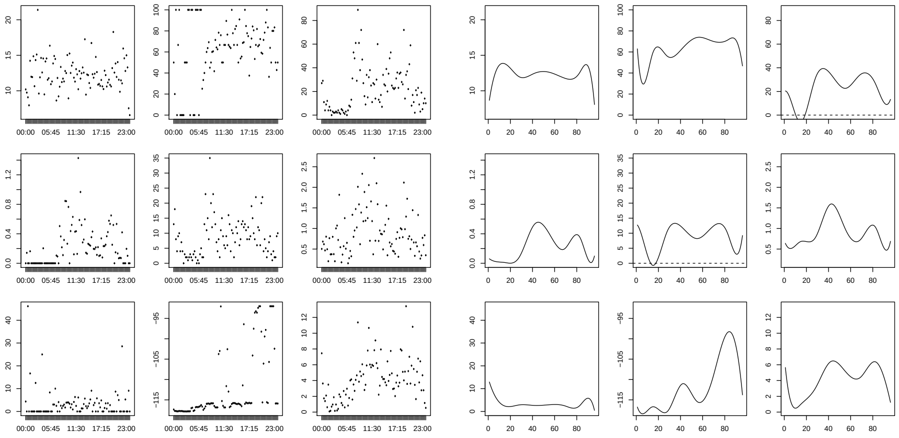

Figure 3: Discrete mean curves belonging to $Family1$ and their smoothing

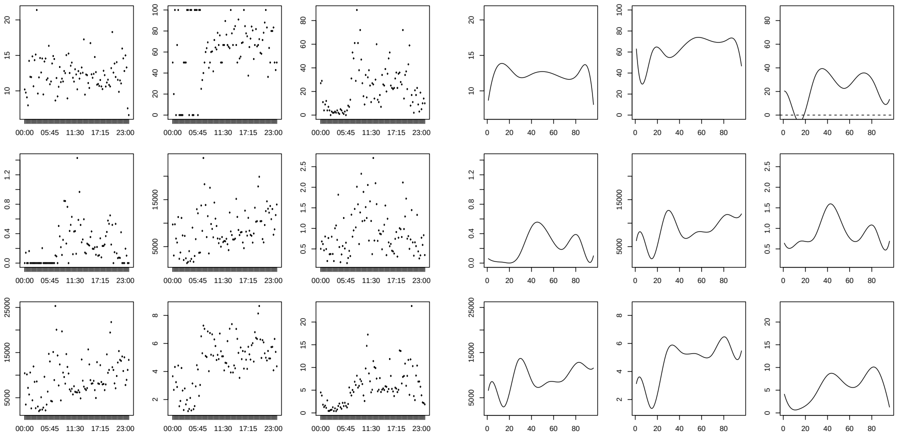

Figure 4: Discrete mean curves belonging to $Family2$ and their smoothing

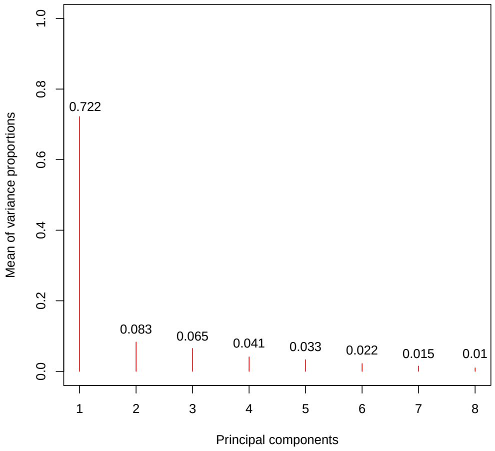

Figure 5: Mean of FPCA variance proportions after 50 simulations of data sets of size 100 × 100

6.2. Tuning of the number of iterations

This experiment aims to choose the length of the SEM-Gibbs chain. It allows to verify that after a burn-in period, the simulations are stable which means that the SEM-Gibbs chain has achieved its stationary distribution. Consequently, it allows to choose the number of iterations. The convergence is tested with q iterations of $\alpha_{k_r}^{(q)}$, $\beta_{k_c}^{(q)}$ and $\mu_{k_r k_c}^{(q)}$ where $q \in [0..100]$ over $s = 20$ simulations of data sets with different sizes (s simulations for each size). The number of blocks are set to $K_r = K_c = 3$ which are the right numbers of row- and column-clusters. Figure 6 illustrates the parameters convergence of all the simulations

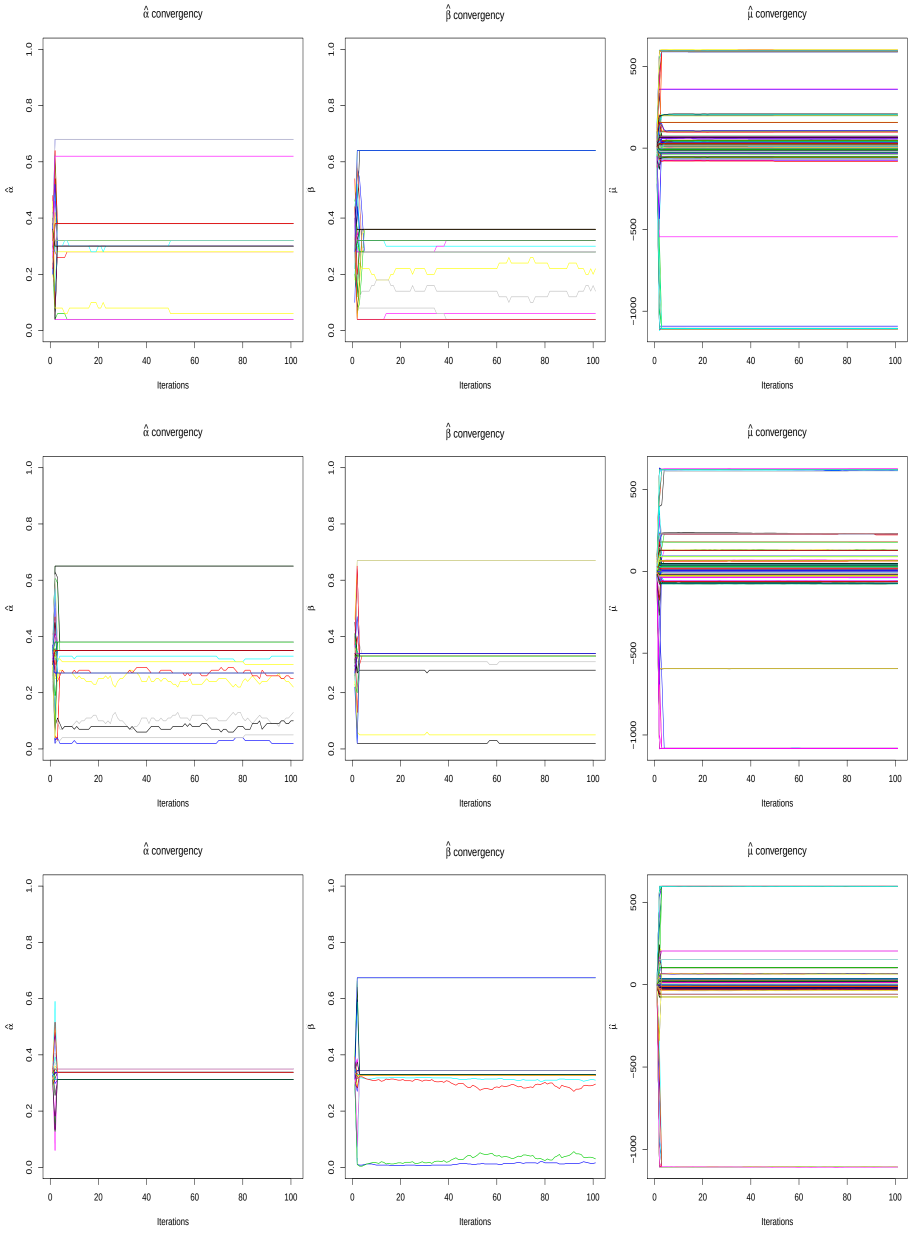

Figure 6: Parameters convergence after $s = 20$ simulations over data sets of sizes 50×50 (first line), 100×100 (second line), 500×500 (third line) with functional data belonging to $Family1$.

where each curve represents the values of the corresponding parameter over q iterations and per simulation. The data sets are of size 50×50, 100×100 and 500×500, where the functional data are derived from the $Family1$.

We notice that SEM-Gibbs algorithm has achieved its stationary distribution. Since by the 50^{th} iteration, the algorithm has already converged, for the rest of the experiment series, the number of iterations is set to $q = 50$ with a burn-in period equal to 20.

6.3. Validation of the SEM-Gibbs algorithm

In this second experiment, the purpose is to validate the parameters estimations resulted from SEM-Gibbs algorithm through $s = 50$ simulations of data sets with different sizes. Let $K_r = K_c = 3$ and the functional data are derived from both families. The comparison between the real parameters and their estimation is held in terms of the following metrics:

- ARI_r (respectively ARI_c): the adjusted rand index of the real row- (respectively column-) clustering and its estimation,
- $\Delta\alpha$: the distance between the real α and its estimation, where $\Delta\alpha = \frac{\sum_{k_r=1}^{K_r} |\alpha_{k_r} - \hat{\alpha}_{k_r}|}{K_r}$, and similarly for $\Delta\beta$,
- $\Delta\mu$: the distance between the real μ and its estimation, where $\Delta\mu = \frac{\sum_{k_r=1}^{K_r} \sum_{k_c=1}^{K_c} \sum_{l=1}^{h} |\mu_{k_r k_c}^{l} - \hat{\mu}_{k_r k_c}^{l}|}{K_r k_c l}$.

Table 1 presents the results over the 50 simulations. Each metric is composed of two values: a mean distance (or ARI) followed by a standard deviation in parenthesis.

Table 1: Validation of the parameters and the final partitions estimated by SEM-Gibbs algorithm over data sets of different sizes

	$Family1$			$Family2$		
Size of the data sets	50×50	100×100	500×500	50×50	100×100	500×500
ARI_r	0.93 *(0.18)*	0.97 *(0.11)*	0.98 *(0.09)*	0.95 *(0.15)*	0.96 *(0.14)*	1 *(0)*
ARI_c	0.92 *(0.19)*	0.92 *(0.19)*	0.95 *(0.15)*	0.9 *(0.2)*	0.99 *(0.08)*	0.97 *(0.11)*
$\Delta\alpha$	0.028 *(0.07)*	0.01 *(0.04)*	0.008 *(0.04)*	0.02 *(0.06)*	0.02 *(0.06)*	0 *(0)*
$\Delta\beta$	0.03 *(0.07)*	0.03 *(0.07)*	0.02 *(0.07)*	0.04 *(0.08)*	0.004 *(0.032)*	0.01 *(0.05)*
$\Delta\mu$	2.14 *(4.01)*	13.2 *(30.02)*	0.92 *(3.12)*	590.13 *(1248.48)*	33 *(229.52)*	103.13 *(412.38)*

As shown in Table 1, ARI_r and ARI_c are close to 1 which proves that the final partitions estimated by the model, $\hat{\mathbf{v}}$ and $\hat{\mathbf{w}}$, are very close to the real partitions $\mathbf{v}$ and $\mathbf{w}$. α and β are two parameters that belong to $[0, 1]$. The fact that the metrics $\Delta\alpha$ and $\Delta\beta$ are close to zero proves that these two parameters are successfully estimated. As for the parameter μ, the possible values depend to the data set. In our simulations, the range of values belongs to $[-120, 100]$ for $Family1$ and to $[-120, 14000]$ for $Family2$, and the data sets are not normalized. That is why, knowing the range of values, the results of $\Delta\mu$ are acceptable enough to conclude that the estimation of the different parameters is close to the reality which proves the efficiency of our model.

6.4. Choice of K_r and K_c

The aim of this section is to verify that the greedy search algorithm presented by Algorithm 1 with the ICL-BIC criterion can detect the right numbers of row- and column-clusters. Therefore, $s = 50$ simulations are generated over data sets of different sizes where the functional data are derived from $Family1$ or from $Family2$. Given $K_r^{max} = K_c^{max} = 4$, we compute the number of selections of each couple $\{K_r, K_c\}$. The results are compared to the exhaustive method in which all the possible combinations of the couples $\{K_r, K_c\}$ are tested as presented in Table 2.

Table 2: The results of the exhaustive method and the Greedy search algorithm over 50 simulations of data sets of different sizes

	$Family1$								$Family2$							
Size of the data sets	50×50				100×100				50×50				100×100			
Exhaustive method	K_r/k_c	2	3	4	K_r/k_c	2	3	4	K_r/k_c	2	3	4	K_r/k_c	2	3	4
	2	0	0	0	2	0	0	0	2	0	0	0	2	0	0	0
	3	0	36	8	3	0	40	5	3	0	42	5	3	0	34	11
	4	0	4	2	4	0	5	0	4	0	3	0	4	0	5	0
Greedy search	K_r/k_c	2	3	4	K_r/k_c	2	3	4	K_r/k_c	2	3	4	K_r/k_c	2	3	4
	2	0	0	0	2	0	1	0	2	0	4	1	2	0	0	0
	3	5	42	2	3	7	40	0	3	3	41	1	3	8	38	1
	4	0	1	0	4	0	2	0	4	0	0	0	4	3	0	0

We notice that the right number of clusters $\{3, 3\}$ is selected most of the time even when the blocks have close behaviors (i.e. the functional data belong to $Family2$). The results of both methods are equivalent but with the Greedy search algorithm, the execution time is faster, which proves the efficiency of the Greedy search algorithm using the ICL-BIC criterion for choosing the number of blocks.

7. Application to mobile networks monitoring

This section presents an application of the proposed co-clustering algorithm to mobile network monitoring. Due to the huge number of KPIs captured continuously from the network, analyzing all of them daily is impossible for engineers. As a result, they are forced to ignore most of the KPIs and to lose the information behind, which may affect the network performance and consequently the quality of services offered by the mobile operators. Even for the automated networks, the enormous amount of data treated by the algorithms may impose a huge computational capacities and a time consuming tasks which is not helpful especially in the case of real-time applications.

In this part, we are interested in the study of the relationship between the behavior of different daily-captured KPIs for a specific geographical area. KPIs with similar behaviors will be grouped in the same block. In this way, the information induced by KPIs 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 Labs, France. We focus our study on Long Term Evolution (LTE) sites located in Lyon as illustrated in Figure 7. LTE is a standard technology that brings the cellular communication to the fourth generation (4G) era [26]. In wireless telephony, a cell [2] 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 99 cells, for 170 KPIs and for 13 days: seven days among them correspond to an atypical week in the telecommunication field, distinguished by the end of summer holidays (from August, 25 to August 31, 2016) and the other six days correspond to a typical work days (from September, 16 to September, 21, 2016).

The KPIs are extracted with a granularity of 15 minutes (therefore, each daily KPI contains 96 values). The extracted KPIs belong to different families of indicators. The first family, labeled "Quality", refers to the network's ability to address its supported services regarding its characteristics. The second family, labeled "Accessibility", refers to the network's ability to meet with the users demands for accessing to the different services. The third family, labeled "Retainability", refers to the network's ability to maintain its users connections regardless the quality. In this extract, 85 KPIs belong to "Quality", 60 KPIs belong to "Accessibility" and 25 KPIs belong to "Retainability".

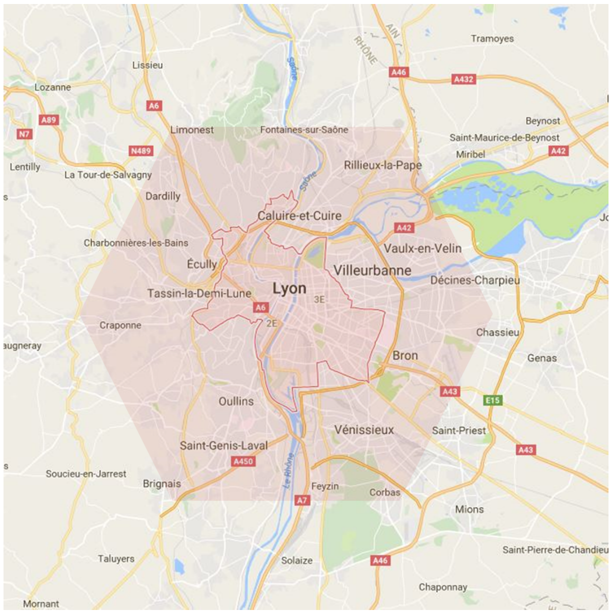

Figure 7: The geographical area used for the data extraction

7.2. Smoothing and FPCA

The overall size of the data set is 1287 rows (= 99 cells × 13 days), 170 columns and 96 discrete values per curve. The data set contains 7% of missing values. These latter are easily treated by applying the smoothing step, since it allows to gain the functional behavior of the daily KPIs which is an advantage when dealing with functional data. A FPCA is then applied, which helps to gain in term of dimensionality reduction. The number of basis functions $M = 20$, and the number of principal components $m = 8$ have been chosen empirically. After pre-processing, the data set is composed of 1287×170 curves, each curve is identified by 8 principal components.

7.3. Co-clustering

Given $\{k_r^{max}, k_c^{max}\} = \{4, 4\}$, the greedy search algorithm suggests to dissociate the data set in 4 row-clusters and 4 column-clusters, since they maximize the ICL-BIC criterion. In this first analysis, we are restricted to $\{4, 4\}$ for an easier interpretation, but since the Greedy search algorithm chose the maximum values, a deeper analysis should be tackled with $K_r^{max} > 4$ and $K_c^{max} > 4$.

By fixing $K_r = K_c = 4$, this experiment aims to apply the LBM for functional data on the data set. Let the number of iterations of the SEM-Gibbs algorithm be equal to 50 with a burn-in period equal to 20. Figures 8 and 9 illustrate the estimated parameters per block i.e. $\hat{\alpha}_{k_r}$, $\hat{\beta}_{k_c}$ and $\hat{\mu}_{k_r k_c}$ when $1 \leq k_r \leq 4$ and $1 \leq k_c \leq 4$. Let's notice that the third column-cluster only contains 2% of the KPIs.

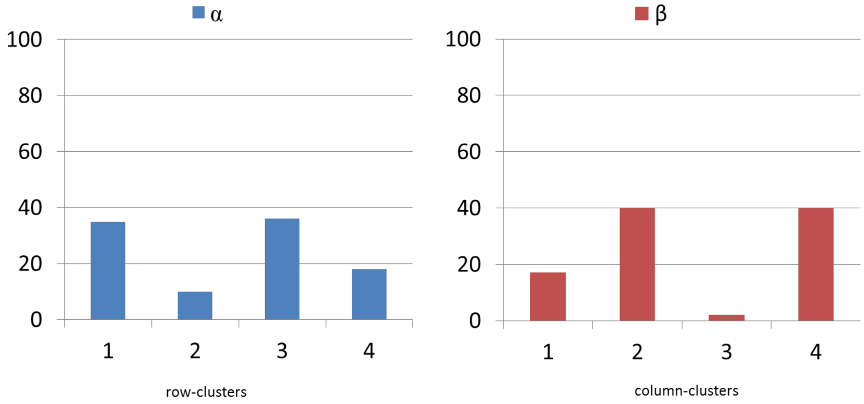

Figure 8: Estimated mixing proportions: $\hat{\alpha}$ (left) and $\hat{\beta}$ (right)

After digging in the data of each block, it can be noticed that the proposed approach has succeed to discern the row-clusters in terms of days. As illustrated in Figure 10, the first row-cluster mainly contains, Monday, Tuesday, Wednesday and Friday. The second row-cluster mainly contains Thursday. The third row-cluster mainly contains Saturday and Sunday. It also contains in less proportions the week days. However, when digging into the cells of each row-cluster as described in Table 3, we notice that the third row-cluster exclusively contains the cells of two sites ”Ile Roy” and ”Fontaine saône” that happens to be isolated sites situated next to the ”Saône” river of Lyon and to train railways. Having this information, we can conclude that these cells have atypical traffic due to their locations which makes the algorithm to consider their activities on work

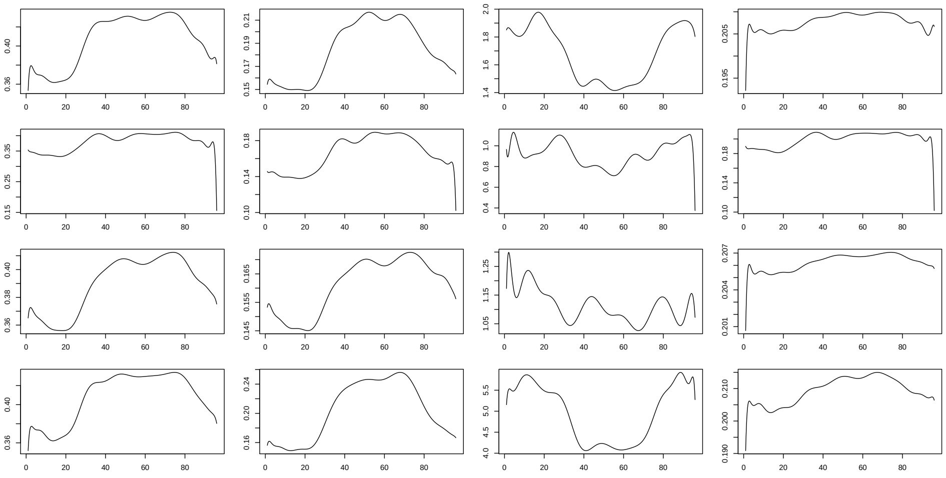

Figure 9: Estimated mean per block

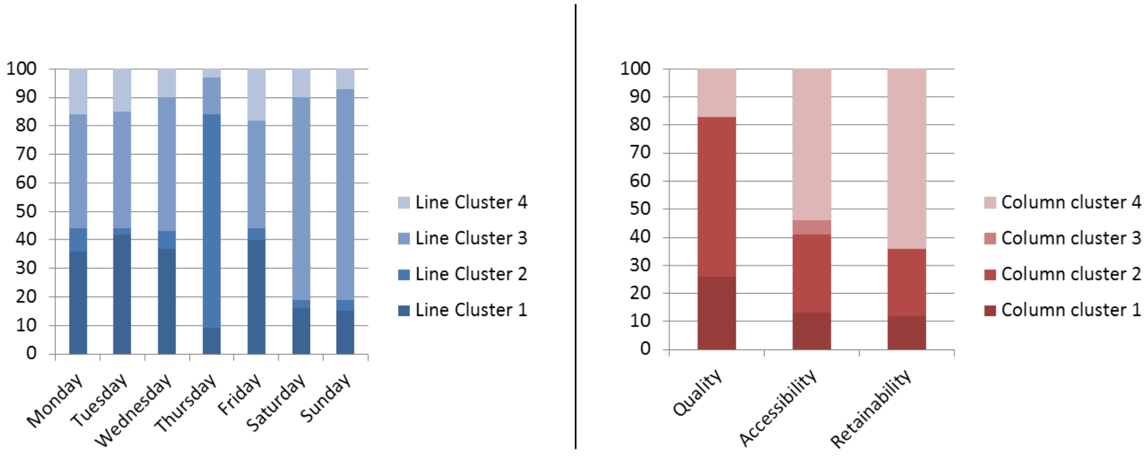

Figure 10: Row-clusters repartition in terms of days (left) and column-clusters repartition in terms of KPIs families (right). The y-axis are percentages.

days similar to the activity of the other cells on week-ends. The fourth row-cluster has a similar behavior as the first row-cluster but it differs with the sites as presented in Table 3.

Regarding the column-clusters, the algorithm has dissociated the KPIs according to their families. As illustrated in Figure 10, we can notice that the

Table 3: Row-clusters repartition in terms of sites

row-cluster	No.1	No.2	No.3	No.4
Brignais	x		x	
Ile Roy			x	
Fontaine Saône			x	
Dugusclin	x	x	x	
Fosses Ours	x	x	x	
Saint Gryphe	x	x	x	
The rest of the sites	x	x	x	x

KPIs of type "Quality" are mainly in the first and second column-clusters. The KPIs of type "Accessibility" and "Retainability" are mainly in the fourth column-cluster and the second column-cluster.

With this initial study on mobile networks, we notice that our model may help to dissociate the data in terms of cells and days which will help the mobile operators for a better planning of the topological structure of their networks. The network topology [2] is the arrangement of the various elements of a network in a geographical area and the presented work may help with its optimization which is a very known problematic in the telecommunication field. However, in order to have more interesting results for self-healing and self-optimization, a deeper study is considered (particularly, with bigger K_r^{max} and K_c^{max}).

8. Conclusion and Future works

While functional data analysis is widely used in many real applications, the co-clustering of functional data has never been proposed. In this paper, a model-based co-clustering for functional data is introduced. In the presented latent block model, each block of curves is identified by the multivariate Gaussian distribution of the principal components of the curves. The model parameters are estimated using a SEM-Gibbs algorithm and the number of row- and column-clusters can be chosen by using a greedy search algorithm based on an ICL-BIC criterion. In addition, the model can also be used for clustering of multivariate functional data [21], by simply setting the number of column clusters to p. Finally, through a simulation study, the proposed algorithm has proven its efficiency and the application of the model on mobile network monitoring has shown interesting results.

As future work, we will held a deeper study on mobile network troubleshooting, optimization and topology planning using the LBM for functional data. Inspired by the clustering model presented in [21], the proposed model can be improved by considering block-specific FPCA. Moreover, following the strategy of [27, 28], several parsimonious sub-models can be introduced by imposing constraints on the covariance matrix $\Sigma_{k_r k_c}$ such as: (1) full covariance matrix common to all blocks; (2) diagonal block-specific covariance matrices; (3) diagonal covariance matrix common to all blocks. The study of the possible parsimonious sub-models is an interesting alternative in order to reduce the execution time. Finally, we are currently working on an R package that may be an interesting option for the prospective users of the model.

References

[1] J. Ramiro, K. Hamied, Self-Organizing Networks (SON): Self-Planning, Self-Optimization and Self-Healing for GSM, UMTS and LTE, 1st Edition, Wiley Publishing, 2012.

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

[3] J. O. Ramsay, B. W. Silverman, Functional data analysis, 2nd Edition, Springer Series in Statistics, Springer, New York, 2005.

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

[5] I. Dhillon, Co-clustering documents and words using bipartite spectral graph partitioning, in: Proceedings of the seventh ACM SIGKDD inter-

national conference on Knowledge discovery and data mining, KDD ’01, ACM, New York, NY, USA, 2001, pp. 269–274.

[6] V. Brault, A. Lomet, Methods for co-clustering: a review, in: Journal de la Société Française de Statistique, 156 (3), 2015, pp. 27–51.

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

[8] G. Govaert, M. Nadif, Clustering with block mixture models, Pattern Recognition 36 (2) (2003) 463 – 473, biometrics.

[9] H. Shan, A. Banerjee, Bayesian co-clustering, in: 2008 Eighth IEEE International Conference on Data Mining, 2008, pp. 530–539.

[10] J. Wyse, N. Friel, Block clustering with collapsed latent block models, Statistics and Computing 22 (2) (2012) 415–428.

[11] C. Keribin, V. Brault, G. Celeux, G. Govaert, Estimation and selection for the latent block model on categorical data, Statistics and Computing 25 (6) (2014) 1201–1216.

[12] J. Jacques, C. Biernacki, Model-based co-clustering for ordinal data, in: 48èmes Journées de Statistique organisée par la Société Française de Statistique, 2016.

[13] E. Matechou, I. Liu, D. Fernandez, M. Farias, B. Gjelsvik, Biclustering models for two-mode ordinal data, Psychometrika 81 (3) (2016) 611–624.

[14] G. Govaert, M. Nadif, Latent Block Model for Contingency Table, Communications in Statistics-Theory and Methods 39 (3) (2010) 416 – 425.

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

[16] J. Jacques, C. Preda, Functional data clustering: a survey, Advances in Data Analysis and Classification 8 (3) (2014) 231–255.

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

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

[19] J.-M. Chiou, P.-L. Li, Functional clustering and identifying substructures of longitudinal data, Journal of the Royal Statistical Society. Series B. Statistical Methodology 69 (4) (2007) 679–699.

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

[21] J. Jacques, C. Preda, Model-based clustering of multivariate functional data, Computational Statistics and Data Analysis 71 (2014) 92–106.

[22] A. Dempster, N. Laird, D. Rubin, Maximum likelihood from incomplete data via the EM algorithm, Journal of the Royal Statistical Society 39 (1) (1977) 1–38.

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

[24] A. Lomet, Sélection de modèle pour la classification croisée de données continues, Ph.D. thesis, Université de Technologie de Compiègne, Compiègne, France (2012).

[25] V. Robert, G. Celeux, C. Keribin, Latent block model and model selection with application in pharmacovigilance, in: StatLearn 2016, 2016.

[26] M. Rinne, O. Tirkkonen, Lte, the radio technology path towards 4g, Computer Communications 33 (16) (2010) 1894 – 1906.

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

[28] G. Celeux, G. Govaert, Gaussian parsimonious clustering models, The Journal of the Pattern Recognition Society 28 (1995) 781–793.