Co-clustering de courbes fonctionnelles multivariées

Amandine Schmutz, Julien Jacques, Charles Bouveyron, Laurence Chèze, Pauline Martin

Co-clustering de courbes fonctionnelles multivariées

Amandine Schmutz [1,2,4], Julien Jacques [2], Charles Bouveyron [3], Laurence Chèze [4]& Pauline Martin [1]

[1] *Lim France, Chemin Fontaine de Fanny, Nontron, France*
CWD-Vetlab, Ecole Nationale Vétérinaire d'Alfort, Maisons-Alfort, F-94700, France
aschmutz@lim-group.com
[2] *Université de Lyon, Lyon 2, ERIC EA3083, Lyon, France et*
julien.jacques@univ-lyon2.fr
[3] *Université Côte d'Azur, LJAD-UMR 7351 & Epione - Inria Sophia Antipolis, Nice, France et charles.bouveyron@math.cnrs.fr*
[4] *Université de Lyon, Lyon 1, LBMC UMR T9406, Lyon, France et*
laurence.cheze@univ-lyon1.fr

Résumé. La croissance exponentielle des objets connectés présents maintenant dans tous les aspects de la vie quotidienne entraîne une collecte de données à haute fréquence pour un même individu. Ces objets facilitent aussi la collecte de plusieurs variables simultanément pour un même individu, entraînant des besoins croissants de méthodes pour résumer et interpréter ces données fonctionnelles multivariées. Ce travail propose une nouvelle méthode de co-clustering fonctionnelle de façon à faciliter la mise en évidence de groupes d'individus et de variables se ressemblant au sein de bases de données multivariées. Cette méthode s'appuie sur un modèle à blocs latents fonctionnels et l'inférence du modèle est faite à l'aide d'un algorithme SEM-Gibbs. L'efficacité de ce modèle sera testée sur un exemple de suivi de consommation électrique et de température au sein de maisons intelligentes connectées.

Mots-clés. Fouille de données, Statistique computationnelle, Etude de cas, Co-clustering fonctionnel

Abstract. The exponential growth of smart devices in all aspect of everyday life, leads to the collection of high frequency data for a same individual. Those devices also ease the collection of several variables simultaneously for an individual, which results in growing needs of methods to summarise and read such multivariate functional data. This work shows a new functional co-clustering method in order to help highlighting groups of individuals and variables that look alike. This method relies on a functional latent block model and model inference is done with a SEM-Gibbs algorithm. The model efficiency will be shown on a practical example of smart houses where the consumption of electricity and the temperature is monitored over time.

Keywords. Datamining, Computational statistics, Case study, Functional co-clustering

1 Introduction

L'apparition de capteurs de petite taille à un coût réduit a entraîné une croissance exponentielle du développement et de l'utilisation d'objets connectés par le grand public. Ces capteurs sont par exemple installés dans des maisons intelligentes pour suivre la température intérieure simultanément avec la température extérieure et la consommation électrique du foyer. Ces données peuvent être vues comme des données fonctionnelles multivariées : ce sont des entités quantitatives évoluant au cours du temps de façon simultanée pour un même individu statistique.

Afin d'analyser et comprendre un tel volume de données, il peut être intéressant d'identifier des clusters d'individus qui ont le même profil. Pour analyser ces clusters, on peut chercher à les caractériser par leur profil moyen, c'est à dire en examinant les courbes de températures et de consommation moyennes sur la période d'observation. Or cette période étant souvent longue, on cherchera plutôt à la découper sur un intervalle de temps plus facile à interpréter : typiquement la journée. Le nombre de journées d'observation étant lui aussi important, nous allons également chercher à créer des clusters de journées. Pour résumer, nous disposons d'un tableau de données où chaque case est constitué de plusieurs courbes. Typiquement les courbes de consommation et de températures pour un individu et une journée. Nous cherchons alors à regrouper simultanément les individus (lignes) et les journées (colonnes) en clusters homogènes. Ce type de partitionnement joint est appelé co-clustering (Goveart et al, 2013).

A ce jour seules 3 méthodes de co-clustering fonctionnel existent : Bouveyron et al. (2017), Slimen et al. (2018) et Chamroukhi et Biernacki (2017). Ces trois travaux proposent des méthodes de co-clustering fonctionnel univarié, où chaque case du tableau contient une unique courbe. Notre objectif est de proposer un nouveau modèle de co-clustering permettant de prendre en compte des courbes multivariées.

2 Présentation du modèle

2.1 Reconstruction des données

Soit $x = (x_{ij}^s(t))_{1\leq i\leq n, 1\leq j\leq p, 1\leq s\leq S}$ une matrice de données fonctionnelles composée de l'observation de S variables fonctionnelles $x_{ij}^s(t)$, $t \in [0, T]$, pour n individus (lignes) sur p périodes de temps (colonnes).

Dans la pratique, ces courbes ne sont jamais observées à chaque instant $t \in [0, T]$ mais en un nombre fini de points d'observation. La forme fonctionnelle des données est alors reconstruite en supposant que les fonctions peuvent être approchées à l'aide d'une combinaison linéaire de bases de fonctions :

$$x_{ij}^s(t) = \sum_{r=1}^{R_s} c_{ijr}^s \phi_{sr}(t) \tag{1}$$

où $\{\phi_{sr}\}_{1\leq r\leq R_s}$ est le système de bases de fonctions choisi pour approcher la s-ème variable fonctionnelle et c_{ijr}^s les coefficients dans la base de fonctions. Par simplicité, nous supposerons par la suite que les variables fonctionnelles sont exprimées dans une base de fonctions identique (Fourier ou spline) avec le même nombre de bases $R_s = R$. Mais le cas de bases différentes est tout à fait envisageable. On peut alors définir une matrice représentant les données fonctionnelles multivariées :

$$X(t) = \begin{pmatrix} x_{11}(t) & \dots & x_{1p}(t) \\ & \cdots & \\ x_{n1}(t) & \dots & x_{np}(t) \end{pmatrix},$$

où $x_{ij}(t) = (x_{ij}^s(t))_{1\leq s\leq S}$ est une donnée fonctionnelle multivariée.

2.2 Modèle à blocs latents

Nous cherchons à estimer une partition z des lignes en K clusters et une partition w des colonnes en L clusters de la matrice de données X, où $z = (z_i)_{1\leq i\leq n}$, $z_i = (z_{i1}, \dots, z_{iK})$ et $z_{ik} = 1$ si la ligne i appartient au cluster k et 0 sinon. De même $w = (w_j)_{1\leq j\leq p}$, $w_j = (w_{j1}, \dots, w_{jL})$. On note alors Z l'ensemble des partitions possibles des lignes en K groupes et W l'ensemble des partitions possibles des colonnes en L groupes.

Afin de proposer une modélisation probabiliste des données, une analyse en composantes principales fonctionnelle multivariée (MFPCA, Jacques et Preda (2014)) est réalisée par bloc. Chaque courbe multivariée x_{ij}, conditionnellement au fait qu'elle appartienne au bloc (k, l), peut être identifiée par ses scores (δ_{ij}^{kl}) issus de la MFPCA.

Nous considérons le modèle de co-clustering par bloc latent (Goveart et al, 2013), qui suppose que les deux variables aléatoires z et w sont indépendantes, et que, conditionnellement à z et w, les données au sein d'un bloc sont indépendantes et identiquement distribuées.

Le modèle à blocs latents pour données fonctionnelles multivariées est alors définit par :

$$p(\delta;\theta) = \sum_{z\in Z}\sum_{w\in W} p(z;\theta)p(w;\theta)p(\delta|z,w;\theta).$$

où $\theta = (\alpha_k, \beta_l, \gamma_{kl})_{1\leq k\leq K, 1\leq l\leq L}$ avec α_k et β_l les proportions de mélange ligne et colonne, appartenant à $[0, 1]$ et dont la somme vaut 1, et γ_{kl} les paramètres du bloc (k, l). Les scores δ sont supposés gaussiens, avec une paramétrisation parcimonieuse de la matrice de variance (Bouveryon et al., 2017) non abordée dans ce document.

2.3 Inférence du modèle

Dans le cas de notre modèle de co-clustering, l'algorithme EM n'étant pas utilisable pour des raisons calculatoires, un algorithme SEM Gibbs (Keribin et al., 2010) est utilisé pour

estimer les paramètres du modèle. Son but est d'estimer θ en maximisant la vraisemblance observée :

$$l(\theta;\delta) = \sum_{z\in Z}\sum_{w\in W}(\ln(\prod_{ik}\alpha_k^{z_{ik}}) + \ln(\prod_{jl}\beta_l^{w_{jl}}) + \ln(\prod_{ijkl}p(\delta_{ij},\gamma_{kl})^{z_{ik}w_{jl}})).$$

L'algorithme SEM-Gibbs alterne une étape SE où les partitions (z, w) sont simulées, et une étape M où les paramètres θ sont mis à jour de sorte à maximiser la vraisemblance complétée par les partitions simulées à l'étape précédente. Les partitions (z, w) sont simulées à l'aide d'un algorithme de Gibbs, ce qui permet de ne pas avoir à calculer leur distribution jointe. L'algorithme est itéré sur un certain nombre d'itérations, puis les estimateurs finaux des paramètres sont obtenus en moyennant leur valeurs sur l'ensemble des itérations hors période de chauffe.

2.4 Paramétrisation de l'algorithme

Comme énoncé précédemment, notre algorithme repose sur un algorithme SEM-Gibbs. Cet algorithme nécessite d'être initialisé avec des valeurs pour les partitions en lignes et en colonnes. Dans ce but, 2 stratégies d'initialisation sont proposées ici : *k-means* et aléatoire. Dans le cas de l'initialisation aléatoire, les partitions sont tirées aléatoirement en utilisant une distribution multinomiale avec des probabilités uniformes. La stratégie *k-means*, consiste à initialiser les partitions avec celles obtenues par la méthode *k-means* appliquée directement sur l'ensemble des données discrétisées. Le nombre d'itérations de l'algorithme SEM-Gibbs est fixé à 100.

3 Premiers résultats

Afin d'illustrer le fonctionnement de notre algorithme, une étude sur simulation est proposée.

Une matrice de données fonctionnelles bivariées avec 100 lignes et 100 colonnes est générée. Cette matrice contient 12 blocs ($K = 4$ clusters en ligne et $L = 3$ clusters en colonne). La première variable est simulée selon le processus suivant pour 30 instants de temps pris selon un intervalle régulier, $t = 0, 1/30, 2/30, ..., 1$:

$$x_{ij}(t)|z_{ik}w_{jl} = 1 \sim \mathcal{N}(m_{kl}(t), s^2),$$

où $s = 0.3$ et $m_{kl}(t)$ est une fonction moyenne prise parmi les fonctions suivantes (selon le schema $m_{11} = m_{21} = m_{33} = m_{42} = f_1$, $m_{12} = m_{22} = m_{31} = f_2$, $m_{13} = m_{32} = f_3$ et

$m_{23} = m_{41} = m_{43} = f_4$) :

$$\begin{aligned} f_1(t) &= sin(4\pi t) \\ f_2(t) &= 0.75 - 0.5\mathbb{1}_{t\in]0.7,0.9[} \\ f_3(t) &= h(t)/max(h(t)) \text{ où } h(t) = \mathcal{N}(0.2, \sqrt{0.02}); \\ f_4(t) &= sin(10\pi t) \end{aligned}$$

La seconde variable est créée selon le même processus que la première variable mais avec 4 fonctions moyennes différentes :

$$\begin{aligned} f_1(t) &= cos(4\pi t) \\ f_2(t) &= 0.75 - 0.5\mathbb{1}_{t\in]0.2,0.4[} \\ f_3(t) &= h(t)/max(h(t)) \text{ où } h(t) = \mathcal{N}(0.2, \sqrt{0.05}); \\ f_4(t) &= cos(10\pi t) \end{aligned}$$

Pour finir, on ajoute une certains proportion de bruit au sein de chaque bloc en tirant de façon aléatoire un pourcentage τ de courbes issu d'un autre bloc choisi aléatoirement.

Les données fonctionnelles sont lissées dans une base de Fourier à 15 fonctions, les données sont donc de dimension $100 \times 100 \times 15$. Les simulations sont répétées 20 fois, et la qualité des partitions estimées est évaluée avec l'Adjusted Rand Index (ARI).

Les résultats obtenus sont visibles en Figure 1. Comme attendu, les performances de l'algorithme décroissent avec l'augmentation du bruit, mais on peut tout de même noter que l'initialisation *k-means* donne de très bons résultats tant que le bruit ne dépasse pas 50% du volume des données. En effet, on observe que les résultats sont très bons avec l'initialisation *k-means* dans le cas des 4 premiers niveaux de bruit, et pour les 2 niveaux les plus faibles pour l'initialisation aléatoire.

4 Perspectives

Pour conclure, ces premiers résultats sont prometteurs quant au comportement de notre algorithme sur données simulées. Des simulations complémentaires de validation du modèle sont en cours. L'algorithme sera utilisé sur un exemple appliqué de données de consommation d'électricité et de températures dans le but de distinguer des profils de consommateurs différents, et de détecter des logements potentiellement mals isolés thermiquement.

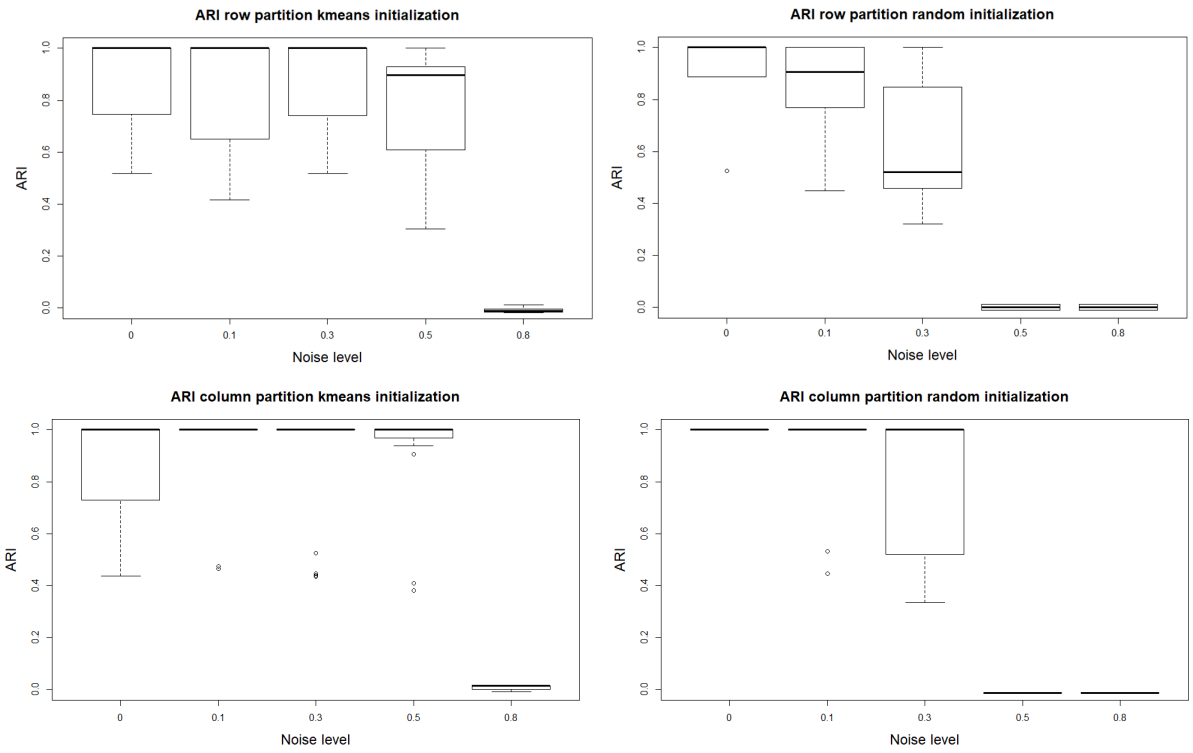

Figure 1: Resultats d'ARI pour des niveaux de bruit croissants en initialisation kmeans (gauche) et aléatoire (droite)

Bibliographie

Ben Slimen, Y., Allio, S., Jacques, J. (2018). Model-based Co-clustering for Functional Data. *Neurocomputing*, 291, pp. 97-108.
Bouveyron, C., Bozzi, L., Jacques, J., Jollois, F.X. (2017). The Functional Latent BLock Model for the Co-Clustering of Electricity Comsumption Curves. *Journal of the Royal Statistical Society: Series C Applied Statistics*.
Chamroukhi, F., Biernacki, C. (2017). Model-based Co-clustering of Multivariate Functional Data. In *ISI 2017 - 61 st World Statistics Congress*, Marrakech, Morocco.
Jacques, J., Preda, C. (2014), Model-based clustering of multivariate functional data, Computational Statistics and Data Analysis, 71, 92-106.
Govaert, G., Nadif, M. (2013). Co-clustering. Wiley-IEEE Press. 256 p.