Classification de données fonctionnelles multivariées par arbres binaires non-supervisées

Steven Golovkine [1] & Nicolas Klutchnikoff [2] & Valentin Patilea [3]

[1] *CREST, ENSAI, steven.golovkine@ensai.com*
[2] *IRMAR, Université Rennes 2, nicolas.klutchnikoff@univ-rennes2.fr*
[3] *CREST, ENSAI, valentin.patilea@ensai.fr*

Résumé. Nous proposons un algorithme de classification non-supervisée par modèle de mélange pour une classe générale de données fonctionnelles. Ces réalisations aléatoires peuvent être mesurées avec erreur à des points d'observations discrets, éventuellement aléatoires, dans leur domaine de définition. L'idée est de construire un ensemble d'arbres binaires par découpage récursif des observations. Le nombre de groupes est déterminé grâce aux données. Cet algorithme fournit des résultats facilement interprétables et de rapides prédictions sur de nouvelles données. Les résultats sur des données simulées montrent de bonnes performances dans différents cas complexes.

Mots-clés. Analyse en composantes principales fonctionnelles mutivariées, Classification non supervisée, Modèle de mélange

Abstract. We propose a model-based clustering algorithm for a general class of functional data. The random functional data realizations could be measured with error at discrete, and possibly random, points in the definition domain. The idea is to build a set of binary trees by recursive splitting of the observations. The number of groups are determined in a data-driven way. The algorithm provides easily interpretable results and fast predictions for online data sets. Results on simulated datasets reveal good performance in various complex settings.

Keywords. Gaussian mixtures, Model-based clustering, Multivariate functional principal components analysis

1 Introduction

Les capteurs sont de plus en plus présents dans notre vie quotidienne. Ceux-ci fournissent un grand nombre de données pouvant être modélisées comme données fonctionnelles. La quantité de données collectées de cette façon augmente rapidement, de même que leur coût d'étiquetage. Ainsi, il y a un intérêt croissant pour les méthodes qui visent à identifier des groupes homogènes au sein d'ensembles de données fonctionnelles.

Supposons un échantillon de N courbes provenant d'un même processus aléatoire, éventuellement mesurées à des instants différents et détériorées par un bruit aléatoire. Notre but est de définir une procédure, basée sur un échantillon de N courbes bruitées, permettant de constuire des groupes de courbes similaires.

2 Modèle

La structure de nos données, appellées *données fonctionnelles multivariées*, est similaire à celle présentée par Happ et Greven (2018). Les données consistent en des trajectoires indépendantes d'un processus stochastique $X = \left(X^{(1)}, \ldots, X^{(P)}\right)^\top, P \geq 1$. Pour chaque $1 \leq p \leq P$, soit $\mathcal{T}_p = [0,1]^{d_p}, d_p \geq 1$. Chaque coordonnée $X^{(p)} : \mathcal{T}_p \to \mathbb{R}$ est supposé appartenir à $\mathcal{L}^2(\mathcal{T}_p)$ muni du produit scalaire usuel, noté $\langle \cdot, \cdot \rangle_2$. Ainsi, X est un processus stochastique indexé par $\mathbf{t} = (t_1, \ldots, t_P)$ appartenant à $\mathcal{T} \coloneqq \mathcal{T}_1 \times \cdots \times \mathcal{T}_P$ et prenant ses valeurs dans $\mathcal{H} \coloneqq \mathcal{L}^2(\mathcal{T}_1) \times \cdots \times \mathcal{L}^2(\mathcal{T}_P)$. Considérons la fonction $\langle\!\langle \cdot, \cdot \rangle\!\rangle : \mathcal{H} \times \mathcal{H} \to \mathbb{R}$,

$$\langle\!\langle f, g \rangle\!\rangle \coloneqq \sum_{p=1}^{P} \langle f^{(p)}, g^{(p)} \rangle_2, \quad f, g \in \mathcal{H}.$$

Happ et Greven (2018) montrent que $\mathcal{H}$ est un espace de Hilbert pour le produit scalaire $\langle\!\langle \cdot, \cdot \rangle\!\rangle$. Soit K un entier positif, et soit Z une variable aléatoire prenant valeurs dans $\{1, \ldots, K\}$ tel que

$$\mathbb{P}(Z = k) = p_k \quad \text{avec} \quad p_k > 0 \quad \text{et} \quad \sum_{k=1}^{K} p_k = 1.$$

La variable Z représente l'appartenance à un groupe des réalisations du processus. Nous considérons que le processus stochastique X suit un *modèle de mélange fonctionnel à K composantes*, qui permet la représentation suivante :

$$X(\mathbf{t}) = \sum_{k=1}^{K} \mu_k(\mathbf{t}) \mathbf{1}_{\{Z=k\}} + \sum_{j \geq 1} \xi_j \phi_j(\mathbf{t}), \quad \mathbf{t} \in \mathcal{T}, \tag{1}$$

où

- $\mu_1, \ldots, \mu_K \in \mathcal{H}$ sont les courbes moyennes par groupe.
- $\{\phi_j\}_{j \geq 1}$ est une base orthonormale de $\mathcal{H}$.
- Les $\xi_j, j \geq 1$ sont des variables aléatoires de $\mathbb{R}$ et conditionnellement indépendantes sachant Z. Pour chaque $1 \leq k \leq K$, $\xi_j | Z = k \sim \mathcal{N}(0, \sigma_{kj}^2)$ pour tout $j \geq 1$.

Lemme 1 *Soit X défini par le modèle* (1) *pour une certaine base orthonormale $\{\phi_j\}_{j \geq 1}$. Soit $\{\psi_j\}_{j \geq 1}$ une autre base orthonormale de $\mathcal{H}$, considérons*

$$c_j = \langle\!\langle X - \mu, \psi_j \rangle\!\rangle, \quad j \geq 1 \quad \textit{avec} \quad \mu(\cdot) = \sum_{k=1}^{K} p_k \mu_k(\cdot).$$

Alors

$$c_j | Z = k \sim \mathcal{N}(m_{kj}, \tau_{kj}^2), \quad \textit{où} \quad m_{kj} = \langle\!\langle \mu_k - \mu, \psi_j \rangle\!\rangle \quad \textit{et} \quad \tau_{kj}^2 = \sum_{l \geq 1} \langle\!\langle \phi_l, \psi_j \rangle\!\rangle^2 \sigma_{kl}^2.$$

Remarque 1 *Le lemme 1 montre que, peu importe le choix de la base orthonormée $\{\psi_j\}_{j\geq 1}$, les groupes seront préservés après avoir projeté les observations dans cette base. Cependant, au regard de l'objectif, certaines bases seront plus adaptées que d'autres.*

En pratique, il n'est pas possible d'utiliser un nombre infini d'élément dans la base $\{\psi_j\}_{j\geq 1}$, et la représentation (1) doit donc être tronquée. On peut montrer, sans utiliser l'hypothèse de normalité des coefficients, que cette troncature peut être arbitrairement précise. Supposons donc que la représentation (1) est tronquée à J termes. La base construite par analyse en composantes principales fonctionnelles multivariées (MFPCA) est celle qui induit la meilleure approximation (en terme de variance expliquée) à J donné (cf. Happ et Greven (2018)). Ainsi, parmi les bases utilisables en pratique, la base MFPCA sera à privilégier dans l'optique de la classification non-supervisée.

2.1 Estimation des paramètres

En pratique, les réalisations de X sont généralement mesurées avec erreur à des points d'observations discrets, éventuellement aléatoires, dans leur domaine de définition. Pour chaque $1 \leq n \leq N$, et étant donné un vecteur d'entiers positifs $\mathbf{M}_n = (M_n^{(1)}, \ldots, M_n^{(P)})$, considérons $T_{n,\mathbf{m}} = (T_{n,m_1}^{(1)}, \ldots, T_{n,m_p}^{(P)}), 1 \leq m_p \leq M_n^{(p)}, 1 \leq p \leq P$ comme étant les points d'observations aléatoires pour la courbe X_n. L'entier $M_n^{(p)}$ représente le nombre de points d'échantillonnage pour la composante p de la courbe X_n. Ces instants sont obtenus comme étant des réalisations indépendantes d'une variable aléatoire T prenant ses valeurs dans $\mathcal{T}$. Les vecteurs $\mathbf{M}_1, \ldots, \mathbf{M}_N$ représentent un échantillon indépendant d'un vecteur aléatoire $\mathbf{M}$ de moyenne $\mu_{\mathbf{M}}$ qui croit avec N. Nous supposons que les réalisations de X, $\mathbf{M}$ et T sont mutuellement indépendantes. Ainsi, nous observons les paires $(Y_{n,\mathbf{m}}, T_{n,\mathbf{m}}) \in \mathbb{R}^P \times \mathcal{T}$, où $\mathbf{m} = (m_1, \ldots, m_P), 1 \leq m_p \leq M_n^{(p)}, 1 \leq p \leq P$ avec $Y_{n,\mathbf{m}}$ défini comme

$$Y_{n,\mathbf{m}} = X_n(T_{n,\mathbf{m}}) + \varepsilon_{n,\mathbf{m}}, \quad 1 \leq n \leq N, \tag{2}$$

et les $\varepsilon_{n,\mathbf{m}}$ sont des réalisations indépendantes de $\varepsilon \in \mathbb{R}^P$ centré et de variance finie.

Les estimateurs des fonctions moyenne et covariance d'une composante $X^{(p)}, 1 \leq p \leq P$ du processus X peuvent, par exemple, être estimés en utilisant Yao, Müller et Wang (2005). Concernant l'estimation des fonctions propres et des valeurs propres de la MPFCA, ainsi que les projections des observations sur la base de fonctions propres, nous utilisons Happ et Greven (2018).

3 fCUBT

Soit $\mathcal{S}_N = \{X_1, \ldots, X_N\}$ un ensemble de réalisations du processus défini en (1). Nous considérons le problème d'apprentissage d'une partition $\mathcal{U}$ tel que chaque élément U de $\mathcal{U}$ contienne des éléments de $\mathcal{S}_N$ similaires. Notre procédure suit les idées de l'algorithme

CUBT, proposé par Fraiman, Ghattas et Svarc (2013) que l'on adapte aux données fonctionelles. Dans la suite, nous décrivons l'algorithme de clustering fonctionnel par arbres binaires non-supervisées (fCUBT).

3.1 Construction de l'arbre maximal

Dans la suite, notons $\mathfrak{T}$, un arbre binaire complet représentant une partition imbriquée de $\mathcal{S}_N$, et $\mathfrak{D} \geq 1$ sa profondeur. Soit $\mathfrak{S}_{0,0}$ le noeud racine auquel on assigne l'ensemble $\mathcal{S}_N$. Chaque noeud $\mathfrak{S}_{\mathfrak{d},\mathfrak{j}} \subset \mathcal{S}_N$ est indexé par la paire $(\mathfrak{d}, \mathfrak{j})$ où $0 \leq \mathfrak{d} < \mathfrak{D}$ est l'indice de profondeur et $0 \leq \mathfrak{j} < 2^{\mathfrak{d}}$ est l'indice du noeud. Chaque noeud, non terminal, $\mathfrak{S}_{\mathfrak{d},\mathfrak{j}}$ a deux enfants, $\mathfrak{S}_{\mathfrak{d}+1,2\mathfrak{j}}$ et $\mathfrak{S}_{\mathfrak{d}+1,2\mathfrak{j}+1}$, tel que

$$\mathfrak{S}_{\mathfrak{d},\mathfrak{j}} = \mathfrak{S}_{\mathfrak{d}+1,2\mathfrak{j}} \cup \mathfrak{S}_{\mathfrak{d}+1,2\mathfrak{j}+1}.$$

Un arbre $\mathfrak{T}$ est ainsi défini par un découpage récursif des observations. À chaque étape, un noeud $\mathfrak{S}_{\mathfrak{d},\mathfrak{j}}$ est potentiellement découpé en deux s'il remplit certaines conditions. Une MFPCA avec $\mathtt{n_{comp}}$ composantes, $\mathtt{n_{comp}} \leq J$, est faite sur les éléments de $\mathfrak{S}_{\mathfrak{d},\mathfrak{j}}$. Il en résulte un ensemble de fonctions propres, associé à un ensemble de valeurs propres. Nous pouvons construire la matrice $C_{\mathfrak{d},\mathfrak{j}}$ dont les colonnes sont les projections des éléments de $\mathfrak{S}_{\mathfrak{d},\mathfrak{j}}$ sur l'ensemble des fonctions propres. Pour chaque $K = 1, \ldots, K_{max}$, nous ajustons un modèle de mélange gaussien (GMM) sur les colonnes de $C_{\mathfrak{d},\mathfrak{j}}$ par algorithme EM. Les modèles sont notés $\{\mathcal{M}_1, \ldots, \mathcal{M}_{K_{max}}\}$. Le nombre de groupes dans le noeud est estimé avec le BIC,

$$\widehat{K}_{\mathfrak{d},\mathfrak{j}} = \underset{K=1,\ldots,K_{max}}{\arg\max}\ \mathrm{BIC}(\mathcal{M}_K)$$

Si $\widehat{K}_{\mathfrak{d},\mathfrak{j}} > 1$, le noeud $\mathfrak{S}_{\mathfrak{d},\mathfrak{j}}$ est coupé en deux en utilisant le modèle $\mathcal{M}_2$. Sinon, ce noeud est considéré comme étant un noeud terminal, et la construction de l'arbre est stoppée pour celui-ci.

La procédure continue jusqu'à ce que l'un des critères d'arrêt soit satisfait : qu'il y ait moins de `minsize` observations dans le noeud ou alors que l'estimation $\widehat{K}_{\mathfrak{d},\mathfrak{j}}$ du nombre de clusters dans le noeud $\mathfrak{S}_{\mathfrak{d},\mathfrak{j}}$ soit égal à 1. Quand l'algorithme est terminé, un label est assigné à chaque feuille de l'arbre.

3.2 Étape de jointure

En théorie, l'arbre a le même nombre de feuilles que le processus X a de composantes. En pratique, c'est rarement le cas et le nombre de feuilles peut être bien supérieur au vrai nombre de groupes. Ainsi, une étape de jointure, dont l'idée est d'associer des noeuds qui n'ont pas le même ascendant, peut être nécessaire.

Soit $\mathcal{G} = (V, E)$ un graphe où $V = \{\mathfrak{S}_{\mathfrak{d},j}, 0 \leq j \leq 2^{\mathfrak{d}}, 0 \leq \mathfrak{d} < \mathfrak{D} | \mathfrak{S}_{\mathfrak{d},\mathfrak{j}} \text{ est une feuille}\}$ est l'ensemble des sommets et

$$E = \left\{(\mathfrak{S}_{\mathfrak{d},\mathfrak{j}}, \mathfrak{S}_{\mathfrak{d}',\mathfrak{j}'}) | \mathfrak{S}_{\mathfrak{d},\mathfrak{j}}, \mathfrak{S}_{\mathfrak{d}',\mathfrak{j}'} \in V, \mathfrak{S}_{\mathfrak{d},\mathfrak{j}} \neq \mathfrak{S}_{\mathfrak{d}',\mathfrak{j}'} \text{ et } \widehat{K}_{(\mathfrak{d},\mathfrak{j})\cup(\mathfrak{d}',\mathfrak{j}')} = 1\right\}$$

est l'ensemble des arêtes. $\widehat{K}_{(\mathfrak{d},\mathfrak{j})\cup(\mathfrak{d}',\mathfrak{j}')}$ est l'estimation du nombre de groupes dans $\mathfrak{S}_{\mathfrak{d},\mathfrak{j}}\cup\mathfrak{S}_{\mathfrak{d}',\mathfrak{j}'}$ en utilisant la même méthodologie que pour l'étape précédente.

Pour chaque élément $(\mathfrak{S}_{\mathfrak{d},\mathfrak{j}}, \mathfrak{S}_{\mathfrak{d}',\mathfrak{j}'})$ de E, nous associons la valeur du BIC qui correspond à $\widehat{K}_{(\mathfrak{d},\mathfrak{j})\cup(\mathfrak{d}',\mathfrak{j}')}$. L'arête de $\mathcal{G}$ ayant la plus grande valeur du BIC est ensuite supprimée, et les sommets associés sont joints. Il y a donc un groupe en moins. Cette procédure est lancée récursivement jusqu'à ce qu'aucun noeud ne puisse être joint avec un autre ou bien qu'il n'y ait plus qu'un noeud dans l'arbre.

Une fois la partition $\mathcal{U}$ créée, nous pouvons utiliser celle-ci pour classifier de nouvelles observations. Cette classification se fait par descente de l'arbre $\mathfrak{T}$, et nous pouvons donc calculer les probabilités d'appartenance à chacune des classes pour les nouvelles observations.

4 Analysis empirique

Montrons les performances de notre algorithme sur un exemple et fixons $K = 5$, $P = 2$, $\mathcal{T}_1 = \mathcal{T}_2 = [0, 1]$. Un échantillon de $N = 1000$ courbes bivariées indépendantes est simulé suivant le modèle : pour $t_1, t_2 \in [0, 1]$,

$$\begin{array}{lll}
\text{Cluster 1:} & X^{(1)}(t_1) = h_1(t_1) + b_{0.9}(t_1), & \\
& X^{(2)}(t_2) = h_3(t_2) + 1.5 \times b_{0.8}(t_2) & \\
\text{Cluster 2:} & X^{(1)}(t_1) = h_2(t_1) + b_{0.9}(t_1), & \\
& X^{(2)}(t_2) = h_3(t_2) + 0.8 \times b_{0.8}(t_2), & \\
\text{Cluster 3:} & X^{(1)}(t_1) = h_1(t_1) + b_{0.9}(t_1), & \\
& X^{(2)}(t_2) = h_3(t_2) + 0.2 \times b_{0.8}(t_2), & \\
\text{Cluster 4:} & X^{(1)}(t_1) = h_2(t_1) + 0.1 \times b_{0.9}(t_1), & \\
& X^{(2)}(t_2) = h_2(t_2) + 0.2 \times b_{0.8}(t_2), & \\
\text{Cluster 5:} & X^{(1)}(t_1) = h_3(t_1) + b_{0.9}(t_1), & \\
& X^{(2)}(t_2) = h_1(t_2) + 0.2 \times b_{0.8}(t_2). &
\end{array}$$

Les fonctions h sont définies, par $h_1(t) = (6 - |20t - 6|)_+ /4$, $h_2(t) = (6 - |20t - 14|)_+ /4$ et $h_3(t) = (6 - |20t - 10|)_+ /4$, pour $t \in [0, 1]$. Les fonctions b_H sont définies, pour $t \in [0, 1]$, par $b_H(t) = (1+t)^{-H} B_H(1+t)$ avec $B_H(\cdot)$ est un mouvement brownien fractionnaire avec un coefficient de Hurst H. Les proportions du mélange sont supposées égales.

Les données sont obtenues suivant le modèle (2). Chaque courbe est observée à 101 points répartis aléatoirement sur $[0, 1]$. Les vecteurs d'erreurs sont supposés être de lois normales centrées et de variance $1/2$. Pour chaque $n \in \{1, \ldots, N\}$, nous observons une réalisation du vecteur $X = \left(X^{(1)} + \alpha X^{(2)}, X^{(2)}\right)^\top$, avec $\alpha = 0.4$.

Notre méthode est comparée aux algorithmes `FunHHDC` (Schmutz et al. (2020)), `Funclust` (Jacques et Preda (2014)) et k-`means` (Ieva et al. (2013)) sur les courbes (k-`means`-d_1) et leur dérivées (k-`means`-d_2)). Nous évaluons aussi notre performance par rapport à un GMM après une FPCA (`FPCA+GMM`) et à notre algorithme si on ne fait que la construction de l'arbre sans l'étape de jointure (`Growing`). Notre algorithme montre de bonnes performances en terme d'index de Rand (cf. Table 1) et d'estimation du nombre de groupes dans le jeu de données (cf. Figure 1).

Method	1	2	3	4	5	6	7+
`fCUBT`	-	-	-	-	0.664	0.238	0.098
`Growing`	-	-	-	-	0.604	0.182	0.214
`FPCA+GMM`	-	-	-	-	0.414	0.396	0.19
`FunHDDC`	0.508	0.492	-	-	-	-	-
`Funclust`	-	0.066	0.182	0.192	0.200	0.196	0.164
k-means-d_1	-	-	-	-	0.034	0.144	0.822
k-means-d_2	-	0.004	0.01	0.094	0.874	0.010	0.008

Table 1: Nombres de groupes

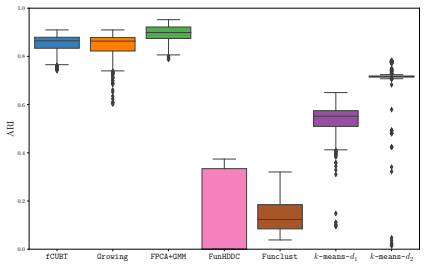

Figure 1: Index de Rand

Enfin, la méthodologie est dévelopée pour des données fonctionnelles multivariées pouvant être définies sur des domaines différents et potentiellement de différentes dimensions. Ainsi, nous pouvons considérer des processus définis sur un carré dans le plan, $\mathcal{T} = [0, 1]^2$ par exemple. Dans ce cas, une décomposition de ce processus peut être faite, par exemple, grâce à l'algorithme FCP-TPA pour une décomposition tensorielle (Allen (2013)) et donc être utilisé dans la MFPCA. Une version complète de l'article est disponible à l'adresse suivante : arxiv:2012.05973.

Bibliographie

Allen, G. I. (2013). Multi-way functional principal components analysis. In 2013 *5th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, 220-223.

Fraiman, R., Ghattas, B. and Svarc, M. (2013). Interpretable clustering using unsupervised binary trees. *Advances in Data Analysis and Classification*, 7.

Happ, C. and Greven, S. (2018). Multivariate Functional Principal Component Analysis for Data Observed on Different (Dimensional) Domains. *Journal of the American Statistical Association*, 113 649-659.

F. Ieva, A. M. Paganoni, D. Pigoli, and V. Vitelli. (2013). Multivariate functional clustering for the morphological analysis of electrocardiograph curves. *Journal of the Royal Statistical Society.* Series C (Applied Statistics), 62(3):401-418, 2013.

Jacques, J. and Preda, C. (2014b). Model-based clustering for multivariate functional data. *Computational Statistics and Data Analysis*, 71 92-106.

Schmutz, A., Jacques, J., Bouveyron, C., Cheze, L. and Martin, P. (2020). *Clustering multivariate functional data in group-specific functional subspaces.* Computational Statistics

Yao, F., Müller, H.-G. and Wang, J.-L. (2005). Functional Data Analysis for Sparse Longitudinal Data. *Journal of the American Statistical Association*, 100 577-590.