Multiple Co-clustering de séries temporelles. Application à la validation de systèmes d'aide à la conduite

Etienne Goffinet [1,2] & Mustapha Lebbah [1] & Hanane Azzag [1] & Loïc Giraldi [2]

[1] *Université Sorbonne Paris Nord, CNRS, Laboratoire Informatique Paris Nord, 93430 Villetaneuse*

[2] *Groupe Renault, 78280 Guyancourt*

Résumé. Le développement de systèmes d'aide à la conduite demeure un défi technique pour les constructeurs automobiles. La validation de ces systèmes nécessite de les éprouver dans un nombre considérable de contextes de conduites. Pour ce faire, le Groupe Renault a recourt à la simulation massive, qui permet de reproduire précisément la complexité des conditions physiques de conduite et produit une grande quantité de séries temporelles multivariées. Nous présentons les contraintes opérationnelles et défis scientifiques liées à ces jeux de données, ainsi que notre proposition d'une approche de classification probabiliste adaptée, qui crée plusieurs partitions indépendantes en regroupant les variables redondantes. Nous présentons les résultats obtenus avec cette nouvelle méthode sur un jeu de données issu d'un cas d'usage industriel.

Mots-clés. Multi-Coclustering, Séries temporelles multivariées, Multi-Clustering

Abstract.

Advanced driver-assistance systems development remains a technical challenge for automobile manufacturers. The reliable validation of these systems requires testing them in a considerable number of driving contexts. The numerical simulation helps Groupe Renault validate such systems, and recreates the complexity of physical driving conditions. These simulations produce large quantities of high-dimensional multivariate time series. We detail the operational constraints associated to these datasets, and a suited model-based classification method able to handle. Based on a hierarchical structure, it creates several independent partitions while grouping redundant variables. We present the results obtained on a dataset from an industrial use case.

Keywords. Multi-Coclustering, Multivariate time series, Multi-Clustering

1 Introduction

Avant de pouvoir être mis sur le marché, les systèmes d'aide à la conduite sont rigoureusement étudiés et testés par les constructeurs automobiles. Pour garantir la qualité de ces tests, fonction de leur exhaustivité, le Groupe Renault a recourt à la simulation massive. Pour un cas d'usage donné, cette validation produit un nombre important

d'observations (en centaines de milliers) décrites par un grand nombre de capteurs (en centaines). L'exploration de ces données permet de déterminer précisément les capacités d'un système d'aide à la conduite et de raffiner son paramétrage par l'expert. Le comportement de la voiture simulée ne peut pas toujours être prédit, ce qui rend le recours aux méthodes non-supervisées indispensable. La classification non-supervisée, ou *clustering*, regroupe des observations similaires en *clusters*. Le clustering de séries temporelles basé sur des modèles (Goffinet (2020)) nous est particulièrement utile, en particulier la production d'intervalles de confiance pour la détection probabiliste des valeurs aberrantes et la production d'intervalles de confiances sur les paramètres inférés. Certaines des variables obtenues en sortie de simulation sont corrélées (par exemple la vitesse d'une voiture et la vitesse de rotation d'une de ses roues), positivement ou négativement, et parfois même dupliquées. D'autres, bien que non-dupliquées et non-corrélées, produisent des partitions similaires lorsqu'elles sont traitées individuellement par une méthode de partitionnement donnée (par exemple position et accélération). Nous proposons une nouvelle méthode qui regroupe les variables de manière hiérarchique: basé sur leur partition ligne, puis sur leur distribution. Le co-clustering (Figure 1 - a)) réalise simultanément un clustering d'observations (aussi appelé partition ligne dans la suite) et un clustering de variables (aussi appelé partition colonne dans la suite). Ce modèle fait l'hypothèse que toutes les variables partagent la même partition ligne. Lorsque les variables ne satisfont pas cette hypothèse de partage de partition ligne (e.g. les variables décrivent des physiques ou des systèmes différents), le co-clustering n'est plus adapté.

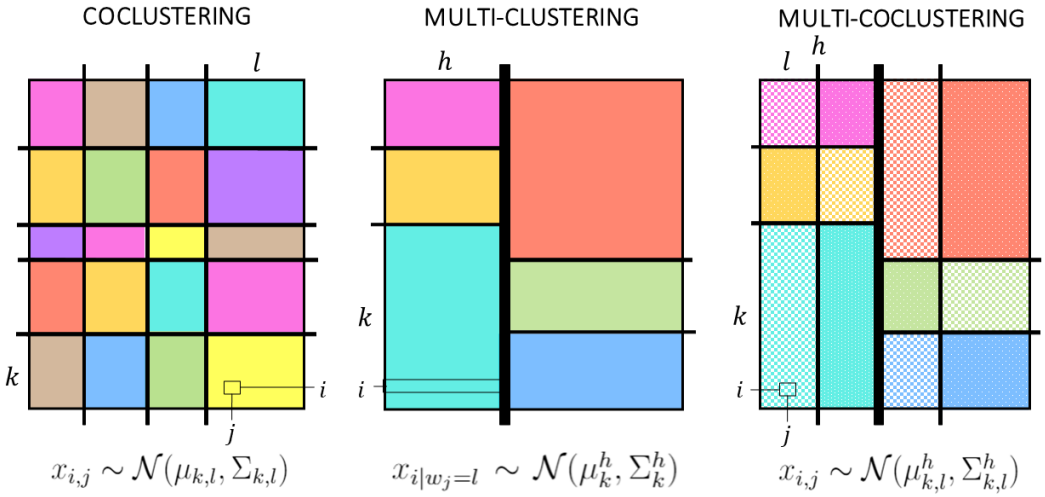

Figure 1: Illustration du Co-clustering, du Multi-Clustering et du Multi-Coclustering. k, l, h sont les indices d'un cluster d'observations, de variables corrélées et de variables redondantes (respectivement). Couleurs et motifs indiquent les appartenances aux blocs.

Cette contrainte peut être relâchée avec le *Multi-Clustering* ((Hu, J. (2018), Marbac

(2019), Vandewalle, V. (2020)) où plusieurs partitions lignes différentes sont inférées, comme illustré sur la Figure 1 - b). Dans ce cas, les partitions ligne ne sont plus mélangées, mais définies indépendamment dans chaque cluster de variables. Dans un contexte paramétrique, la combinatoire liée au Multi-Clustering rend difficile la sélection de modèle par recherche exhaustive (il y a, par exemple, 184755 modèles différents avec au maximum 10 clusters de variables et 10 clusters d'observations). Des approches heuristiques sont alors envisagées, (par exemple des stratégies gloutonnes) au prix d'hypothèses sur la structure du modèle. L'approche non-paramétrique permet de contourner cette contrainte en intégrant nativement une sélection de modèle. Un modèle de Multi-Clustering Bayésien Non-paramétrique a été développée par Guan (2010), pour le traitement de données continues multivariées. Dans ce modèle, les lignes appartenant à un bloc suivent une distribution multivariée indépendante (c.f. Figure 1-b)). Cette méthode de Multi-Clustering regroupe les variables en fonction de leur partition ligne, mais ne regroupe pas les variables de distribution similaire. Nous proposons dans ce papier une nouvelle méthode de Multi Co-Clustering (MCC) non-paramétrique qui regroupe les variables partageant la même partition d'observations et, dans chaque groupe de variables, infère un modèle de blocs latents non-paramétriques (NPLBM) (Meeds (2010)). Cette approche permet de discriminer plus finement les variables : parmi celles qui partagent les mêmes partitions lignes, la méthode regroupe les variables ayant des distributions identiques (c.f. Figure 1-c)). À notre connaissance, il n'existe qu'un seul travail comparable sur le sujet par Tokuda (2017) mais qui ne s'applique pas aux séries temporelles.

2 Multi-Coclustering de séries temporelles multivariées

Dans cette partie nous définissons le modèle (Section 2.1), son inférence (Section 2.2) et présentons une application sur un jeu de donnée issu d'un cas d'usage industriel.

2.1 Définition

Chaque cellule du jeu de données final est un vecteur de coefficients issu d'une PCA fonctionnelle appliquée aux séries exprimées en base polynomiale, traitement classique dans le cas du co-clustering fonctionnel, (Bouveyron (2018), Slimen (2018)). Dans la suite, X désigne le dataset complet, de dimension $n \times p \times d$, avec n le nombre de simulations, p le nombre de variables, d la dimension de l'espace des observations. Soit H le nombre total de clusters de variables, tel que les variables dans un cluster partagent la même partition ligne. Ces clusters seront indicés par la variable $h \in \{1, \dots, H\}$. Pour chacun de ces clusters, soit p_h le nombre de clusters de variables possédant une distribution identique. Le vecteur $\mathbf{v} \in \mathbb{N}^p$ représente les appartenances aux clusters de variables partageant une même partition ligne, et le vecteur $\mathbf{w}_h \in \mathbb{N}^{p_h}$ l'appartenance aux clusters de variables

distribution identique. Ainsi, la paire $(\mathbf{v}, \mathbf{w}_h)$ est nécessaire pour identifier l'appartenance d'une variable à un cluster de colonnes. La matrice $Z \in \mathbb{N}^{H \times n}$ désigne les appartenances aux partitions lignes. Le modèle est défini par

$$x_{i,j} \mid v_j, w_j, z_{i,v_j}, \theta^{v_j}_{z_{i,v_j},w_j} \sim \mathcal{N}(\theta^{v_j}_{z_{i,v_j},w_j}),\ \theta^{v_j}_{z_{i,v_j},w_j} \sim G_0$$

$$v_j \sim Mult(\eta),\ \eta_j(\mathbf{r}) = r_j \prod_{j'=1}^{j-1} (1 - r_{j'}),\quad r_j \overset{\text{i.i.d.}}{\sim} Beta(1, \gamma)\ , \gamma \sim Gamma(a_r, b_r)$$

$$w_j \sim Mult(\rho_h),\ \rho^h_j(\mathbf{s}^h) = s^h_j \prod_{j'=1}^{j-1} \left(1 - s^h_{j'}\right),\quad s^h_j \overset{\text{i.i.d.}}{\sim} Beta(1, \beta_h)\ , \beta_h \sim Gamma(a_c, b_c)$$

$$z^{v_j}_j \sim Mult(\pi_h),\ \pi^h_j(\mathbf{t}^h) = t^h_j \prod_{j'=1}^{j-1} \left(1 - t^h_{j'}\right),\quad t^h_j \overset{\text{i.i.d.}}{\sim} Beta(1, \alpha_h),\ \alpha_h \sim Gamma(a_l, b_l).$$

Dans chaque dimension $(\mathbf{v}, W, Z)$, les proportions des appartenances suivent une construction *stick-breaking* (Sethuraman (1994)) dont le paramètre de concentration suit une loi Gamma. Les paramètres des distributions gaussiennes multivariées de chaque bloc (dont l'ensemble est noté Θ) suivent un prior conjugué Normal-inverse-Wishart, notée G_0. Dans la suite, on note $\chi = (G_0, a_r, b_r, a_c, b_c, a_l, b_l)$ l'ensemble des hyper-paramètres.

2.2 Inférence

Nous proposons une inférence en deux étapes: d'abord une étape de Multi-Clustering qui regroupe les variables partageant la même partition ligne, puis l'inférence d'un modèle NPLBM dans chacun de ces clusters.

Multi-Clustering Dans cette première étape, la distribution postérieure $p(\mathbf{v}, Z \mid X, \chi)$ est approchée par un échantillonneur de Gibbs. Chaque itération de l'algorithme se décompose en trois étapes: 1) Échantillonnage de $\mathbf{v}$ sachant Z; 2) Pour $h = \{1, \ldots, H\}$, mise à jour de Z^h sachant $\mathbf{v}$; 3) Mise à jour des paramètres de concentration. Dans l'étape 1), pour $j = 1, \ldots, p$, v_j est échantillonnée selon une loi multinomiale de proportions

$$p(v_j \mid \mathbf{v}_{-j}, Z, \chi) \propto \begin{cases} \dfrac{p_h}{p - 1 + \gamma}\ p(\mathbf{x}_{.,j} \mid \mathbf{z}^h, \chi), & \text{cluster existant } h, \quad (1) \\ \dfrac{\gamma}{n - 1 + \gamma}\ \mathrm{p}(\mathbf{x}_{.,j} \mid \chi), & \text{nouveau cluster}, \quad (2) \end{cases}$$

avec $\mathbf{v}_{-j} = \{v_i : i \neq j\}$ et $p(\mathbf{x}_{.,j} \mid \mathbf{z}^h)$ la distribution prédictive a priori de $x_{.,j}$ dans $\mathbf{z}^h$. Dans l'équation (2), $\mathrm{p}(\mathbf{x}_{.,j} \mid \chi)$ est estimé par inférence d'un modèle de mélange de processus de Dirichlet (DPM), (une fois par variable, avant l'inférence du MCC), qui produit également la partition ligne optimale $\hat{\mathbf{z}}^j$ associée au nouveau cluster échantillonné. Dans l'étape 2), les appartenances $\mathbf{z}^h$ sont mis à jour par un échantillonneur de Gibbs multivarié. L'étape 3) met à jour γ et α_h suivant la procédure définie par West (1998).

Coclustering Pour $h = \{1, \ldots, H\}$, un modèle NPLBM est inféré, à $\mathbf{z}^h$ fixé, par un algorithme de Gibbs approchant $p(\mathbf{w}^h \mid \mathbf{v}, \mathbf{z}^h)$. Pour chaque variable $j' = 1, \ldots, p_h$ du jeu de données $X^h = (x_{i,j} : v_j = h)$, une nouvelle appartenance est échantillonnée selon :

$$p(w^h_{j'} \mid \mathbf{w}^h_{-j'}, \mathbf{z}^h_{-j'}, \chi) \propto \begin{cases} \dfrac{p^h_l}{p_h - 1 + \beta} \displaystyle\prod_{k=1}^{K^h} p(\mathbf{x}^h_{k,j'} \mid \mathbf{x}^h_{k,l}, \chi), & \text{cluster existant } l, \quad (3) \\ \dfrac{\beta}{p_h - 1 + \beta} \displaystyle\prod_{k=1}^{K^h} p(\mathbf{x}^h_{k,j'} \mid \chi), & \text{nouveau cluster}, \quad (4) \end{cases}$$

avec $\mathbf{w}^h_{-j'}$ et $\mathbf{z}^h_{-j'}$ les appartenances aux blocs dans le cluster h sans la variable j', $q^h_l = \sum_{i \neq j'} \mathbb{I}_l(w_i)$, $\mathbf{x}^h_{k,j'} = \{x_{i,j'} : v_{j'} = h, z^h_i = k\}$, et $p(\mathbf{x}^h_{k,j'} \mid \mathbf{x}^h_{k,l}, \chi)$ et $p(\mathbf{x}^h_{k,j'} \mid \chi)$, respectivement, les distributions jointes prédictives a posteriori et a priori dans le bloc (k, l). Après chaque itération, le paramètre de concentration β_h est également mis à jour.

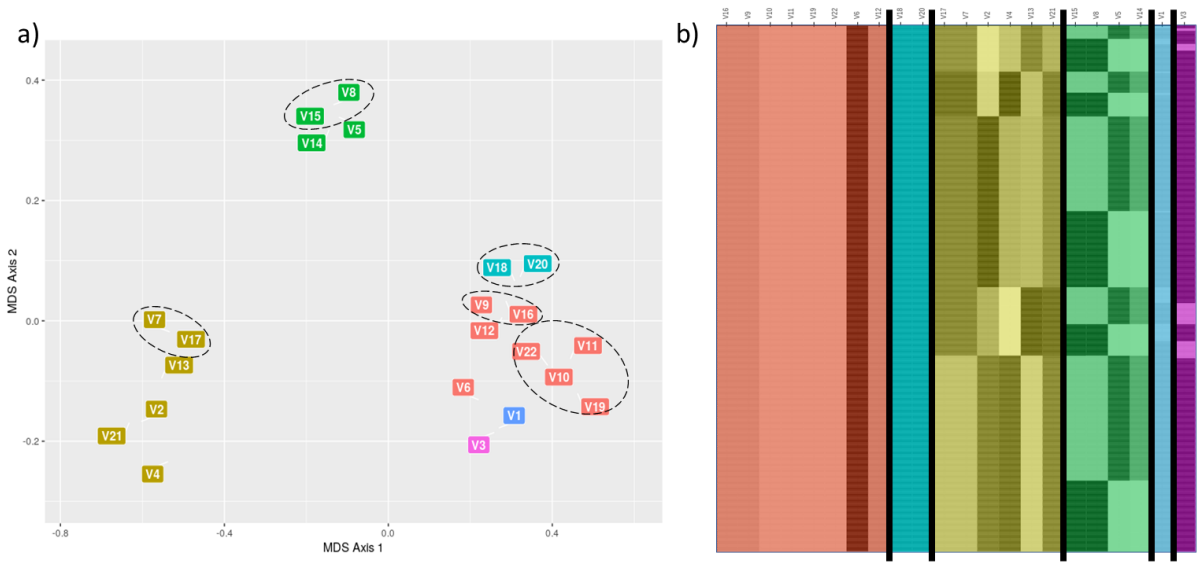

Figure 2: Résultats de l'application sur données réelles

Application sur données réelles Après validation de MCC sur des données issues du modèle génératif, nous l'appliquons sur un jeu de données réel issu de la validation d'un système d'aide au maintien dans la voie. Les groupes de variable partageant le même clustering ligne sont regroupés par couleur dans le graphique 2-a) et par distribution (pointillés). Le graphique 2-b) représente les partitions lignes. En plus de discriminer les variables non-informatives et dupliquées, ce graphique fait apparaître une structure hiérarchique entre les groupes de variables "Kaki" (qui regroupe des variables de direction), "Vert" (regroupant des variables de positions dans la voie) et l'activation des

systèmes d'aide à la conduite (V1 et V3). Les prochaines étapes sont maintenant la mise en relation de ces partitions avec les paramètres d'entrée de la simulation pour construire un modèle explicatif.

Bibliographie

Goffinet, E, Lebbah, M, Azzag, H., Giraldi, L. (2020). Autonomous Driving Validation With Model-Based Dictionary Clustering. ECML-PKDD, 2020.

Marbac, M., Vandewalle, V. (2019). A tractable multi-partitions clustering. Computational Statistics & Data Analysis, 132, 167-179.

Vandewalle, V. (2020). Multi-Partitions Subspace Clustering. Mathematics, 8(4), 597.

Hu, J., and Pei, J. (2018). Subspace multi-clustering: a review. Knowledge and information systems, 56(2), 257-284.

Guan, Y., Dy, J. G., Niu, D., and Ghahramani, Z. (2010). Variational inference for nonparametric multiple clustering. In MultiClust Workshop, KDD-2010.

Meeds, E., Roweis, S. (2007). Nonparametric bayesian biclustering. Technical report, University of Toronto.

Tokuda, T., Yoshimoto, J., Shimizu, Y., Okada, G., Takamura, M., Okamoto, Y., ... and Doya, K. (2017). Multiple co-clustering based on nonparametric mixture models with heterogeneous marginal distributions. PloS one, 12(10), e0186566.

Bouveyron, C., Bozzi, L., Jacques, J., Jollois, F. X. (2018). The functional latent block model for the co-clustering of electricity consumption curves. Journal of the Royal Statistical Society: Series C (Applied Statistics), 67(4), 897-915.

Slimen, Y. B., Allio, S., Jacques, J. (2018). Model-based co-clustering for functional data. Neurocomputing, 291, 97-108.

Neal, R. M. (2000). Markov chain sampling methods for Dirichlet process mixture models. Journal of computational and graphical statistics, 9(2), 249-265. Meeds, E., Roweis, S. (2007). Nonparametric bayesian biclustering. Technical report, University of Toronto.

West, M. (1992). Hyperparameter estimation in Dirichlet process mixture models. ISDS Discussion Paper 92-A03: Duke University.

Williamson, S., Dubey, A., Xing, E. (2013, February). Parallel Markov chain Monte Carlo for nonparametric mixture models. In International Conference on Machine Learning.

Meguelati, K., Fontez, B., Hilgert, N., Masseglia, F. (2019, April). Dirichlet process mixture models made scalable and effective by means of massive distribution. In Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing (pp. 502-509).

Sethuraman, J. (1994). A constructive definition of Dirichlet priors. Statistica sinica, 639-650.