Clustering de séries temporelles par construction de dictionnaire

Étienne Goffinet[*,**] Mustapha Lebbah[*]
Hanane Azzag[*], Loïc Giraldi[**]

[*]Laboratoire Informatique de Paris Nord
99 Avenue Jean Baptiste Clément
93430 Villetaneuse
www.lipn.univ-paris13.fr
[**]Renault SAS
1 Avenue du Golf, 78280 Guyancourt

Résumé. La classification non-supervisée est un domaine qui regroupe les méthodes d'analyses de données dont l'objectif est la recherche de groupes d'observations similaires dans un jeu de données. Lorsque les données considérées sont issues de l'observation d'un phénomène à différents instants, elles sont appelées des séries temporelles : par exemple l'évolution du cours du temps d'une action boursière, de données météorologiques... Dans certains cas, ces séries peuvent alterner différentes phases de fonctionnement distinctes, que l'on appelle des régimes : par exemple, l'observation de la vitesse d'une voiture qui peut montrer des phases d'accélération, une vitesse de croisière, des phases de freinage, etc... Nous présentons dans cet article une méthode dédiée à l'analyse de ce dernier type de séries temporelles et qui est basée sur la combinaison de trois étapes : la segmentation individuelle des séries temporelles, le recodage dans un dictionnaire de régimes communs et le clustering des séquences catégorielles ainsi produites. Notre contribution inclut également une stratégie innovante de sélection de modèle pour la segmentation. Nous présentons les différents avantages de cette méthode et les résultats obtenus sur des jeux de données publics.

1 Introduction

L'attention portée à l'analyse de séries temporelles a augmenté drastiquement ces dernières décennies en même temps que la capacité à les produire. Dans le cas industriel cela peut s'expliquer par la baisse des coûts des capteurs ajoutée au besoin de certifier les équipements avec toujours plus de rigueur. L'augmentation de la taille et de la complexité des données à analyser ont rendu nécessaire le développement d'outils pour fournir une aide précieuse à la décision des experts du métier.

La classification non-supervisée (ou clustering) de séries temporelles est un outil qui a pour but de partitionner un jeu de données en des groupes d'observations temporelles "similaires", ce qui constitue une première phase dans la compréhension de sa structure. Définir la similarité

entre les séries est un point crucial car elle détermine à la fois la construction du clustering et l'interprétation des résultats.

Dans certains cas d'usage, les séries sont issues de l'enchaînement de différentes phases, par exemple dans le domaine de la reconnaissance d'activité humaine basée sur l'observation des mouvements. Il est alors possible de se baser sur l'enchaînement de ces phases (l'ordre, la fréquence, l'amplitude. . .) pour caractériser les séries temporelles et ainsi les différencier. Plusieurs travaux ont été publiés sur ce thème durant les dernières décennies. La problématique a, par exemple, été traitée par la famille des méthodes à base de dictionnaire telles que proposées par Lin et Li (2009) et Schäfer (2015), qui se basent sur une extraction de descripteurs par segmentation à pas de temps uniforme. Plusieurs travaux ont été parallèlement publiés sur le sujet de la détection de points de changements de régimes optimaux, qui peuvent se voir comme autant de méthodes d'estimation de modèles de régressions polynomiales par morceaux.

Ces méthodes estiment les points de changements de régime par l'optimisation de fonctions objectifs basées sur la qualité d'approximation, et ce de différentes façons : fenêtre glissante et de taille croissante comme dans Keogh et al. (2004) et Fuchs et al. (2010), par programmation dynamique comme dans Lavielle et Moulines (2000), par des modèles de Markov cachés dans Kehagias (2004) ou par des modèles de mélange de régression par morceaux chez Samé et al. (2011). C'est ce dernier modèle, basé sur l'hypothèse de l'existence de processus logistiques latents, que nous avons choisi de considérer pour la suite. Ce choix est motivé d'une part par les avantages liés à l'utilisation d'un modèle de mélange : la facilité à produire des intervalles de confiance ainsi que la capacité à faire de la sélection de modèles sans intervention de l'expert via le recours à des critères stochastiques étudiés dans la littérature (c.f. section suivante). Ce choix est également motivé par les performances particulières de ce modèle, illustrées dans Chamroukhi et al. (2009) et Chamroukhi et al. (2010) en comparaison de la modélisation par modèle de Markov caché et son efficacité computationnelle par rapport à d'autres méthodes de type programmation dynamique.

La première étape de notre méthode est basée sur ce modèle de segmentation, allié à une stratégie originale de sélection de modèle. Dans la seconde phase nous construisons un dictionnaire de régimes en réalisant un clustering des régimes précédemment extraits. Pour ce faire, nous normalisons chacun des régimes et les exprimons dans une base polynomiale commune avant d'estimer un modèle de mélange gaussien. Les séries temporelles sont alors recodées dans le dictionnaire ainsi produit et sont transformées en séquences catégorielles. Le résultat final est obtenu après une classification hiérarchique ascendante de ces séquences avec la distance d'édition sur l'espace des chaînes de caractères. Notre méthode présente les avantages suivants :

— Le clustering basé sur la détection de régimes est intuitif et facilement interprétable par un expert, tant que ces régimes sont associables à des phénomènes qui lui sont connus.
— La méthode n'est pas affectée par d'éventuelles différences de longueur des séries temporelles. Par ailleurs, elle est indépendante de la synchronicité des séries temporelles ainsi que de la synchronicité de l'apparition de ces régimes.
— La phase de segmentation est applicable séparément et indépendamment sur chaque série temporelle, ce qui rend le calcul sur un jeu de données trivialement distribuable. Cette phase permet une réduction drastique de la dimension des données.
— La stratégie de segmentation permet la recherche automatique du nombre optimal de

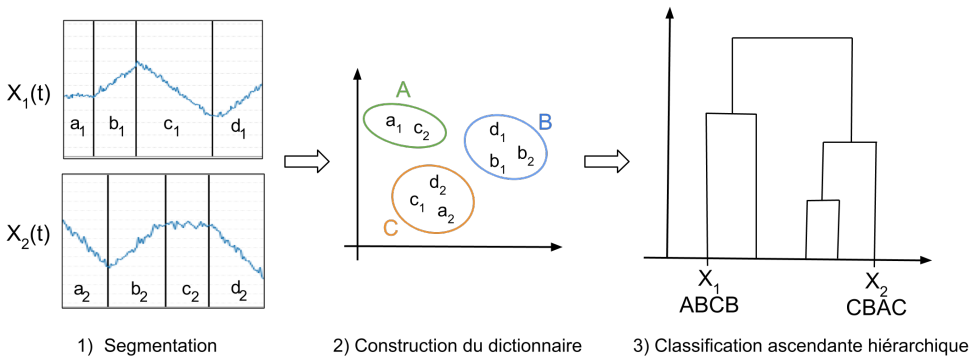

FIG. 1 – *Illustration des étapes de la méthode*

segments et l'optimisation du degré de la régression polynomiale sur chaque segment. Cette méthode permet de réduire fortement la complexité du modèle en relâchant l'hypothèse d'égalité des ordres des régressions sur chaque sous-segments.

— La construction du dictionnaire est également automatique et ne nécessite pas d'être ajustée par une intervention extérieure.

La seconde section de cet article présente en détail les étapes de la méthode, qui sont illustrées sur la figure 1, et les hypothèses associées à chacune d'elles. Dans la troisième section nous présentons les résultats obtenus sur plusieurs jeux de données publics, avant de présenter nos conclusions et interprétations des capacités de notre méthode dans la dernière partie. Ces capacités découlent principalement des propriétés des modèles de mélange.

2 Un clustering en trois étapes

Nous avons retenu le principe des modèles de mélange pour les parties segmentation et création de dictionnaire, qui est une approche standard de classification non-supervisée.

2.1 Segmentation par modèle de mélange

Soit $x = (x_t)_{t\in\{1,..,T\}}$ est un ensemble de réalisations de la variable aléatoire X et $\phi = (\phi_r(t) = t^r)_{r\in\{1,...,R\}}$ une base polynomiale d'ordre $R \in \mathbb{N}$. Un modèle de régression polynomial de la séquence x dans cette base est décrit par :

$$\forall t \in \{1,..,T\}, x_t \sim \mathcal{N}(\beta^T \Phi(t), \sigma^2), \tag{1}$$

avec $\beta = (\beta_s)_{s\in\{1,...,S\}} \in \mathbb{R}^S$ le vecteur de coefficients de régression et $\sigma^2 \in \mathbb{R}$. Le modèle de segmentation consiste en un mélange de modèles de régression polynomiale.

Soit $K \in \mathbb{N}$ le nombre de clusters, et $z = (z_t)_{t\in\{1,..,T\}}$ le vecteur décrivant les appartenances des points $(x_t)_{t\in\{1,..,T\}}$ à ces clusters. $\forall t \in \{1,..,T\}$, z_t suit une distribution

multinomiale de paramètres $\pi(t) = (\pi_k(t))_{k\in\{1,\dots,K\}}$. La distribution de x à l'instant t est alors défini par :

$$p(x_t) = \sum_K \pi_k(t) f_k(t;\theta), \tag{2}$$

et la log-vraisemblance de la séquence x est donnée par :

$$l(x;\theta) = \sum_{t=1}^{T} log\left(\sum_{k=1}^{K} \pi_k(t) f_{\theta_k}(x_t)\right), \tag{3}$$

avec $f_{\theta_k}(x_t)$ les densités des régressions associées à chaque composant. Les proportions du mélange $\pi(t) = (\pi_k(t))_{k\in\{1,..,K\}}$ sont des fonctions du temps qui varient selon un processus logistique : $\forall\ t \in \{1,..,T\}, \pi(t) = (\pi_k(t))_{k\in\{1,..,K\}}$ telles que $\sum_K \pi_k(t) = 1$. Chaque composant du mélange $(f_k(t))_{k\in\{1,..,K\}}$ est alors défini par la fonction de densité d'un modèle de régression polynomiale. Pour chaque composant k, ce processus est décrit par :

$$\pi_k(t) = p(z_t = k) = \frac{\exp(\sum_{s=1}^{S} w_{k,s}\phi_s(t))}{\sum_{h=1}^{K}\exp(\sum_{s=1}^{S} w_{h,s}\phi_s(t))}, \tag{4}$$

paramétré par les poids logistiques $w_k = (w_{k,s})_{s\in\{1,\dots,S\}}$. On note dans la suite $w = (w_k)_{k\in\{1,\dots,K\}}$ et $\theta = (w, \beta, \sigma)$ l'ensemble des paramètres de la loi de mélange. L'expression (1) met en évidence l'un des avantages de cette méthode de segmentation : elle inclut une estimation de la variance $(\sigma_k^2)_K$, ce qui rend le modèle robuste en présence de bruit. L'utilisation de ce modèle implique implicitement une première hypothèse de forme sur les régimes : la base polynomiale doit être adaptée à cette estimation. À base polynomiale et nombre de composants fixés, la log-vraisemblance (3) est optimisée par un algorithme EM (décrite dans sa version générale dans Dempster et al. (1977)) dans une version spécifique proposée dans Samé et al. (2011).

Expectation-Maximization (EM) algorithm L'algorithme EM est une approche classique de maximisation de la vraisemblance en présence de données manquantes. Dans notre cas, ces données manquantes sont les appartenances aux clusters latents décrites par le vecteur z. L'algorithme est itératif, chaque itération comprenant deux étapes.

Étape Expectation (E) Étant donné un état des paramètres θ, la première étape consiste en l'estimation de l'espérance, conditionnelle à θ, de la log-vraisemblance suivante :

$$\begin{aligned}\mathbb{E}_{x,\theta}\left[l(x,z;\theta)\right] &= \mathbb{E}_{x,\theta}\left[\sum_{t=1}^{T}\sum_{k=1}^{K}\mathbb{I}_{z_t=k} log\left(p(x_t, z_t = k;\theta)\right)\right] \\ &= \sum_{t=1}^{T}\sum_{k=1}^{K}\tau_{t,k} log\left(\pi_{t,k} f_{\theta_k}(x_t)\right).\end{aligned} \tag{5}$$

Le développement de cette log-vraisemblance en l'expression (5) montre que cette étape conduit à l'estimation des valeurs $\tau_{t,k} = p(z_t = k|x_t;\theta)$, la distribution de z conditionnelle à x. Ces probabilités s'estiment par l'expression suivante :

$$\tau_{t,k} = p(z_t = k|x_t;\theta) = \frac{p(z_t = k, x_t;\theta)}{p(x_t)} = \frac{\pi_{t,k} f_{\theta_k}(x_t)}{\sum_{h=1}^{K} \pi_{t,h} f_{\theta_h}(x_t)}.$$

Étape Maximisation (M) Durant cette étape, les paramètres du modèle sont mis à jour. Les nouvelles valeurs des paramètres sont obtenues en maximisant l'espérance de la log-vraisemblance décomposée selon l'expression :

$$\begin{aligned}
\mathbb{E}_{x,\theta}\left[l(x,z;\theta)\right] &= \sum_{t=1}^{T}\sum_{k=1}^{K} \tau_{t,k} log\left(\pi_k f_{\theta_k,t}(x_t)\right) \\
&= \sum_{t=1}^{T}\sum_{k=1}^{K} \tau_{t,k} log\pi_k + \sum_{t=1}^{T}\sum_{k=1}^{K} \tau_{t,k} log f_{\theta_k,t}(x_t) \\
&= Q_1(\pi) + Q_2((\theta_k)_{k\in\{1,\dots,K\}}).
\end{aligned}$$

Avec $\tau_{t,k} = p(z_t = k|x_t,\theta)$ les probabilités d'appartenance conditionnelles estimées à l'étape E. L'optimisation de cette espérance passe donc par la maximisation de Q_1 et de Q_2. L'optimisation de la première expression Q_1 est réalisée par l'algorithme Iterative Reweighted Least Square, et produit les nouvelles valeurs des poids logistiques w. Les paramètres de composants $(\beta_k, \sigma_k)_K$ sont obtenus via l'optimisation de Q_2, décrite par les expressions suivantes :

$$\tilde{\beta}_k = arg\min_{\beta_k} \sum_{t=1}^{T} \tau_{t,k}(x_t - \sum_{r=1}^{R} \beta_k \phi_r(t))^2, \tag{6}$$

$$\tilde{\sigma}_k^2 = \frac{1}{\sum_{t=1}^{T} \tau_{t,k}} \sum_{t=1}^{T} \tau_{t,k}(x_t - \tilde{x}_k(t))^2, \tag{7}$$

avec $\tilde{x}_k(t) = \sum_{s=1}^{S} \tilde{\beta}_{k,s}\phi_s(t)$ la valeur de x_t estimée au point t par le composant k.

Stratégie adaptative de sélection de modèle La méthode proposée initialement dans Chamroukhi et al. (2009) décrit un modèle avec un nombre fixe de composants et une base polynomiale commune à tous les composants. En pratique, le nombre de composants est rarement connu à l'avance, et l'ordre des régression peut varier d'un régime à l'autre. Nous avons donc développé une stratégie de type "top-down" qui permet d'adapter à la fois le nombre de composants et l'ordre du modèle de régression de chaque composant. La stratégie est itérative et consiste, à chaque étape, en la mise à jour du composant le moins performant, au sens de la vraisemblance partielle définie ainsi :

$$l_k(x;\theta) = \frac{1}{\sum_{t=1}^{T} \pi_{t,k}} \sum_{t=1}^{T} \pi_{t,k} log\left(f_{\theta_k}(x_t)\right), k \in \{1,..,K\}. \tag{8}$$

Ce critère quantifie la qualité de représentation d'un composant pondérée par les probabilités d'appartenance des observations. Deux modèles sont créés à partir du composant (noté $k_{old} \in \{1,\dots,K\}$) qui minimise ce score.

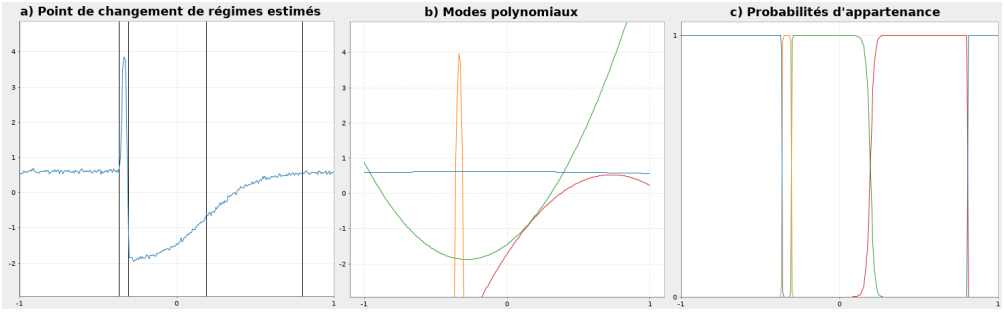

FIG. 2 – *Résultat de la stratégie de segmentation sur une série temporelle issue du jeu de données* Trace*; a) montre les points de changement, la deuxième partie b) superpose la courbe initiale avec le mode de chaque composant du mélange ; dans le troisième cadre c) sont représentées les probabilités d'appartenance à posteriori des points de la série temporelle*

La premier modèle candidat est obtenu en scindant k_{old} en deux sous-composants, noté k_1 et k_2. Notons par ailleurs t_m la médiane de la séquence $\{1, \ldots, T\}$ pondérée par les probabilités $\pi_{k_{old}}$. Nous considérons ce point t_m comme point de changement de régimes entre k_1 et k_2. Les probabilités d'appartenance des nouveaux composants sont basées sur celles de k_{old} ; pour k_1, elles sont définies selon l'expression :

$$\pi_{k_1} = \begin{cases} \pi_{t,k_{old}} & , t \in \{1, \ldots, t_m\} \\ \epsilon & , t \in \{t_m + 1, \ldots, T\} \end{cases} , \tag{9}$$

avec ϵ la précision machine minimale. Les probabilités d'appartenance associées au composant k_2 sont obtenues par la même expression, avec indices de temps inversés. Après cette transformation, il est nécessaire de régulariser les paramètres $(\pi_k)_K$ afin que la contrainte : $\forall t \in \{1, \ldots, T\}, \sum_{k=1} \pi_k(t) = 1$ soit respectée.

Le second modèle candidat est obtenu en augmentant l'ordre de la régression associée à k_{old} d'une unité. On lance ensuite deux EMs : chacun prenant l'un des modèles candidats comme état initial. Après convergence de chaque EM, nous obtenons deux modèles différents qu'il faut départager. Ce choix est basé sur le score BIC (issu de Schwarz et al. (1978)). Le modèle ainsi sélectionné est alors considéré comme point de départ pour l'itération suivante, et ainsi de suite jusqu'à ce que l'amélioration du score BIC ne soit plus significative. Les points de changement de régimes sont estimés par maximum des probabilités d'appartenance.

L'ensemble de la méthode de segmentation est appliquée individuellement sur chaque série temporelle et les transforme en suites de sous-séquences. La distribution du nombre de segments obtenus sur le jeu de données *Trace*, par clusters finaux obtenus, est illustrée sur la figure 5. Un exemple de résultat de segmentation est également disponible sur la figure 2 : la séquence de points de rupture est cohérente et le degré de chaque composant s'est effectivement adapté à chacun de régimes.

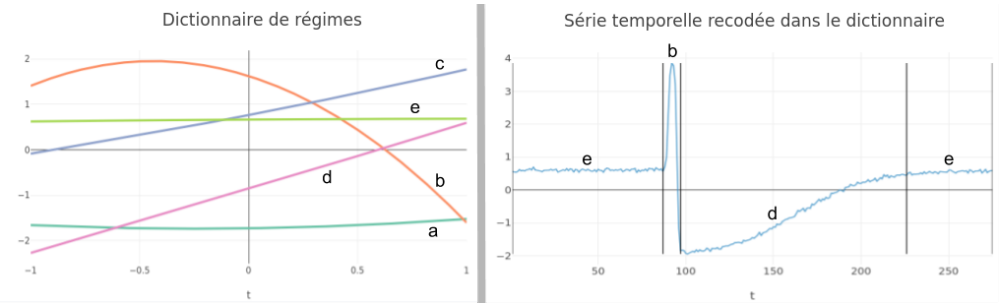

FIG. 3 – *Modes obtenus après la construction du dictionnaire des segments issus du jeu de données* Trace *et exemple de recodage d'une série temporelle dans le dictionnaire*

2.2 Construction du dictionnaire

Pour pouvoir comparer entre eux les régimes des séries temporelles, il faut que ceux-ci s'expriment dans une base commune. C'est le but de cette deuxième étape, qui va réaliser un clustering de l'ensemble des sous-séquences précédemment extraites pour trouver des groupes de régimes similaires. Ces groupes formeront le dictionnaire dans lequel seront recodées les séquences de régime. Pour ce faire nous transformons d'abord les sous-séquences de telle manière qu'elles aient le même support et estimons pour chacune un modèle de régression dans une base polynomiale commune. Chaque sous-séquence est transformée en un vecteur de coefficients dans cette base.

Nous estimons ensuite les paramètres d'un modèle de mélange gaussien (GMM) hétéroscédastique sur ce nouveau jeu de de données. Nous évoquions dans la partie 2.1 une hypothèse implicite sur la base de polynômes utilisée. Dans cette seconde partie nous faisons une deuxième hypothèse implicite : les sous-segments correspondant aux "mêmes" régimes (au sens de l'expert) sont "suffisamment" proches et ceux correspondant à des régimes "différents" sont "suffisamment" différents pour être respectivement regroupés ou discriminés par le GMM.

La sélection du nombre de classes est à nouveau basée sur l'optimalité du critère BIC. L'algorithme EM est initialisé dans ce cas de manière classique par l'algorithme K-means++, dont l'efficacité a été prouvée dans (Blömer et Bujna, 2013).

Dans notre cas d'application nous avons choisi de ne pas standardiser les sous-séquences, cependant c'est une hypothèse possible, qui revient à ne pas donner d'importance à la moyenne ni à la variance des sous-séquences dans la création du dictionnaire. De la même façon, il est possible d'ajouter l'information de la durée des sous-séquences.

Les séries sont alors recodées dans le dictionnaire ainsi créé pour les transformer en séquences catégorielles. Un exemple de résultat de dictionnaire et de recodage est illustré sur la figure 3. Après cette phase de recodage la dimension des données a été largement réduite : pour une série temporelle de taille n, la dimension passe de $\mathbb{R}^n$ à D^k, avec D le dictionnaire des régimes et k le nombre de régimes composant les séquences. Le clustering de ces séquences est l'objet de la troisième étape de notre méthode.

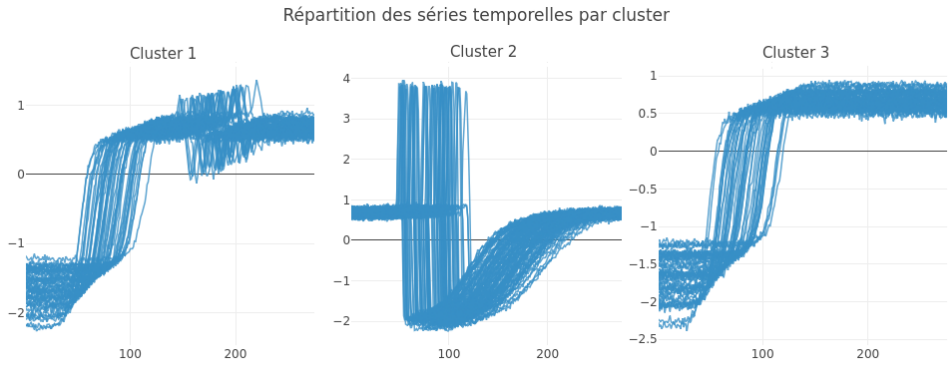

FIG. 4 – *Distribution des séries temporelles du jeu de données* Trace *dans un clustering à trois composants*

2.3 Clustering de séquences categorielles

Dans cette dernière partie les séquences de régimes sont assimilées à des chaînes de caractère. Dans cet espace, nous utilisons la distance d'édition alliée à une classification ascendante hiérarchique selon la méthode de Ward pour obtenir les clusters finaux. De manière analogue aux deux premières parties de l'algorithme de clustering, l'hypothèse que nous faisons ici est la suivante : la distance d'édition est adaptée et fournit des clusters qui sont intéressants au regard de l'objectif d'interprétation de l'expert. Un exemple de résultats de cette méthode de clustering hiérarchique (avec élagage de l'arbre hiérarchique à trois clusters) est illustré sur la figure 4, ainsi que la distribution du nombre de régimes détectés par clusters sur la figure 5.

Le jeu de données *Trace* contient des séries temporelles idéales pour notre méthode : elles sont formées de suites de régimes similaires et les classes observées peuvent être expliquées par ces enchaînements. La figure 5 montre en outre que, dans le cas de *Trace*, la distribution du nombre de segments détectés dépend fortement du cluster. Ce résultat était attendu, la différence de taille entre chaînes de caractère étant une borne inférieure de la distance de Levenshtein associée.

Ce cas correspond au cas d'usage que nous ciblons, cependant la question se pose : que donnerait la méthode sur des jeux de données ne possédant pas ces particularités ? Pour y répondre, nous avons testé la méthode sur les jeux de données issus de Dau et al. (2018) et comparé les résultats à ceux d'autres méthodes de l'état de l'art.

3 Expérimentations

Dans cette partie nous exposons les résultats obtenus sur les jeux de données de l'archive UCR. Pour établir une base de comparaison, nous nous comparons à deux méthodes de référence dans le domaine du clustering de séries temporelles. La première méthode DD_{DTW}

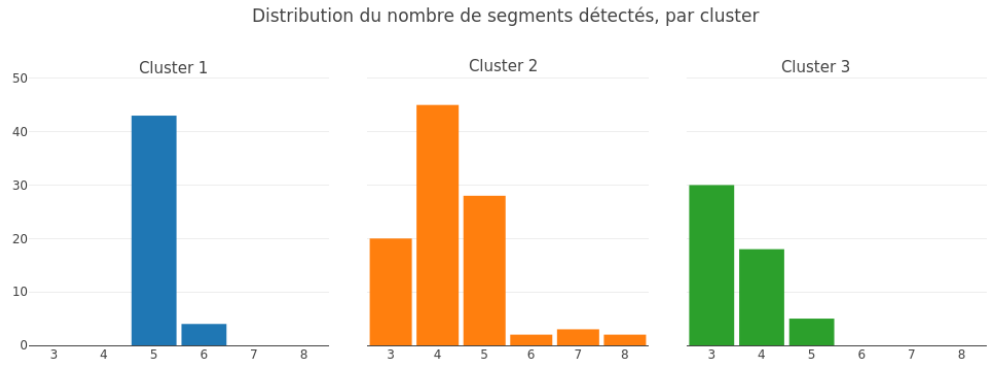

FIG. 5 – *Distribution du nombre de segment détectés durant l'étape de segmentation des time series de* Trace*, par clusters, dans un clustering à trois composants*

présentée dans Łuczak (2016) consiste en un clustering hiérarchique alliée à une distance combinant DTW et DTW appliquée sur la différentielle des séries. La deuxième méthode à laquelle nous nous comparons est KSC ; elle a été présentée dans Yang et Leskovec (2011) et consiste en un clustering par partitionnement avec une construction de centroides basée sur une distance spécifique invariante par translation et changement d'échelle des séries temporelles. Ces solutions se basent sur des hypothèses différentes des nôtres, cette comparaison permet donc de mettre en relief l'intérêt de l'hypothèse de construction par régimes et du modèle de dictionnaire.

La comparaison entre les méthodes est réalisée sur la base de l'Index de Rand (RI). Ce critère, classique dans le cadre de la validation de méthode de clustering, mesure la similarité entre clusters créés et classes observées en calculant le pourcentage d'observations correctement regroupées/séparées conformément aux classes observées.

Tandis que la méthode DD_{DTW} est déterministe, KSC et notre méthode (que nous nommerons $SD-LHC$ pour Segmentation, Dictionnaire, Levenshtein Hierarchical Clustering) sont stochastiques et intègrent donc une part d'aléatoire. La part d'aléatoire dans notre méthode vient de la stratégie d'initialisation de l'algorithme EM lors de la phase de construction de dictionnaire. On retiendra dans chaque cas la moyenne du score obtenu sur trente lancers.

Notre méthode est globalement performante et, en comparaison un à un, donne des résultats compétitifs avec les méthodes de l'état de l'art. Dans de nombreux cas, le rand index obtenu par $SD-LHC$ est équivalent à celui obtenu par les autres méthodes, et il est strictement meilleur que DD_{DTW} sur 62% des jeux de données, et meilleur que KSC dans 54% des cas.

Il n'est pas possible, par manque de place, de montrer ici l'ensemble des résultats sur les 84 jeux de données UCR. Nous produisons cependant, dans le tableau 1, un extrait des scores obtenus. Les exemples retenus sont illustratifs des cas d'usage favorables et défavorables, et sont représentés sur les figure 6 et figure 7.

Nous pouvons faire plusieurs remarques en mettant en relation les résultats obtenus avec le profil des séries temporelles qui composent les datasets. La figure 6 nous donne un aperçu

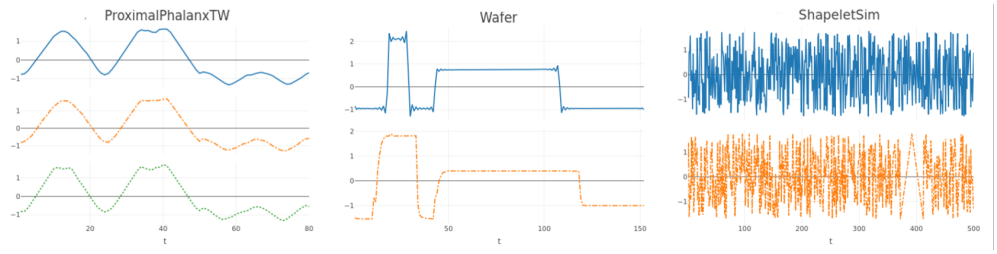

FIG. 6 – *Échantillons représentant quelques classes de jeux de données sur lesquels $SD-LHC$ est peu performant.*

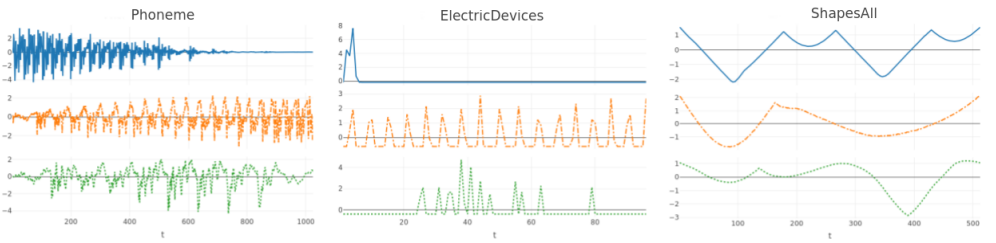

FIG. 7 – *Échantillons représentant quelques classes de jeux de données sur lesquels $SD-LHC$ est performant.*

de certains jeux de données représentatifs sur lesquels $SD-LHC$ est peu performant. Ces mauvaises performances s'expliquent ici par la proximité des segments détectés (pour *ProximalPhalanxTW* et *Wafer*) et par l'inadéquation de la base polynomiale utilisée lors de la segmentation (pour *ShapeletSim*), c'est à dire le non-respect de nos hypothèses.

Sur la figure 7 sont représentés les profils de jeux de données sur lesquels la méthode $SD-LHC$ est particulièrement performante. Les profils de ces séries sont cohérentes avec nos hypothèses. Davantage d'illustrations et d'informations sur les jeux de données sont disponibles sur la page web de l'archive UCR : Dau et al. (2018).

Name	DD_DTW_HC	KSC	SD-LHC
ElectricDevices	0.441	0.362	**0.725**
Phoneme	0.453	0.508	**0.875**
ShapesAll	0.838	0.628	**0.960**
ProximalPhalanxTW	**0.880**	0.805	0.758
Wafer	0.534	**0.591**	0.531
ShapeletSim	0.498	0.498	**0.501**

TAB. 1 – *Rand Index obtenu sur des jeux de données illustratifs issus de l'archive UCR*

4 Conclusions et perspectives

Dans le cadre de la classification non-supervisée de séries temporelles montrant des signes de construction par régimes, nous proposons une méthode proche des méthodes par dictionnaire et qui se compose de trois étapes : la segmentation automatique de chaque série temporelle, la construction d'un dictionnaire de régimes communs et le clustering des séquences de régimes ainsi produites.

La méthode montre de bons résultats lorsqu'elle est appliquée à des séries qui suivent l'hypothèse de construction par régime, mais est également compétitive de manière globale face à d'autres méthodes de l'état de l'art. Par ailleurs elle possède de nombreux avantages, en particulier l'indépendance à la différence de longueur, la synchronicité des séries et régimes, ainsi que les avantages liés au principe du modèle de mélange (stratégie de sélection de modèle, intervalles de confiance, etc...). Tandis ce principe est appliqué dans les deux premières parties de la méthode, la troisième partie est basée sur une classification hiérarchique, qui est une heuristique.

Pour pouvoir accéder à des stratégies de sélection de modèle dans cette dernière partie, l'idée vient naturellement de chercher à étendre cette troisième partie en un modèle de mélange. Ce développement nécessite la définition d'une fonction de densité adaptée dans l'espace des séquences catégorielles.

Références

Blömer, J. et K. Bujna (2013). Simple methods for initializing the em algorithm for gaussian mixture models. *CoRR*.

Chamroukhi, F., A. Samé, G. Govaert, et P. Aknin (2009). Time series modeling by a regression approach based on a latent process. *Neural Networks 22*(5-6), 593–602.

Chamroukhi, F., A. Samé, G. Govaert, et P. Aknin (2010). A hidden process regression model for functional data description. application to curve discrimination. *Neurocomputing 73*(7-9), 1210–1221.

Dau, H. A., E. Keogh, K. Kamgar, C.-C. M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, Yanping, B. Hu, N. Begum, A. Bagnall, A. Mueen, et G. Batista (2018). The ucr time series classification archive. `https://www.cs.ucr.edu/~eamonn/time_series_data_2018/`.

Dempster, A. P., N. M. Laird, et D. B. Rubin (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society : Series B (Methodological) 39*(1), 1–22.

Fuchs, E., T. Gruber, J. Nitschke, et B. Sick (2010). Online segmentation of time series based on polynomial least-squares approximations. *IEEE Transactions on Pattern Analysis and Machine Intelligence 32*(12), 2232–2245.

Kehagias, A. (2004). A hidden markov model segmentation procedure for hydrological and environmental time series. *Stochastic Environmental Research and Risk Assessment 18*(2), 117–130.

Keogh, E., S. Chu, D. Hart, et M. Pazzani (2004). Segmenting time series : A survey and novel approach. In *Data mining in time series databases*, pp. 1–21. World Scientific.

Lavielle, M. et E. Moulines (2000). Least-squares estimation of an unknown number of shifts in a time series. *Journal of time series analysis 21*(1), 33–59.

Lin, J. et Y. Li (2009). Finding structural similarity in time series data using bag-of-patterns representation. In *International conference on scientific and statistical database management*, pp. 461–477. Springer.

Łuczak, M. (2016). Hierarchical clustering of time series data with parametric derivative dynamic time warping. *Expert Systems with Applications 62*, 116–130.

Samé, A., F. Chamroukhi, G. Govaert, et P. Aknin (2011). Model-based clustering and segmentation of time series with changes in regime. *Advances in Data Analysis and Classification 5*, 301–321.

Schäfer, P. (2015). The boss is concerned with time series classification in the presence of noise. *Data Mining and Knowledge Discovery 29*(6), 1505–1530.

Schwarz, G. et al. (1978). Estimating the dimension of a model. *The annals of statistics 6*(2), 461–464.

Yang, J. et J. Leskovec (2011). Patterns of temporal variation in online media. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pp. 177–186. ACM.

Summary

Clustering is a particular subset of the data analysis methods, which aims at researching and discriminating groups (clusters) of similar observations in a dataset. Often these data can be observed at different step forming time series as the evolution of a stock market value or meteorological phenomena. In some time series, the sequences of observations exhibit distinct and interpretable phases, which we call "regimes". For instance, a car speed can show acceleration, cruise speed and breaking phases. In this article we propose a method dedicated to the clustering of this particular kind of time series. It consists in the combination of three steps: an individual segmentation of the time series, the construction of a common regimes dictionary, and the final clustering of categorical sequences produced from the recoding of the time series in this dictionary. We present the different advantages of this method and the results obtained on several public datasets.