Catégorisation de séquences temporelles - Application à l'analyse de parcours de soins

Enoal Gesny, Pierre Pinson, Thomas Guyet

Catégorisation de séquences temporelles – Application à l'analyse de parcours de soins

Enoal Gesny[*,**], Pierre Pinson[**]
Thomas Guyet[*]

[*]Inria, Hospices Civils de Lyon, Université Claude Bernard Lyon 1
[**]AP-HP/URC Cochin

Résumé. En vue de l'amélioration des prises en charge futures, il est intéressant de proposer aux cliniciens des vues objectives sur leurs pratiques. La catégorisation des parcours de soins répond à cet objectif de révéler des groupes homogènes de prise en charge. La difficulté de catégoriser des parcours de soins, représentés par des séquences d'évènements datés, réside dans la définition d'une métrique appropriée sémantiquement et pour les algorithmes de catégorisation. Dans cet article, nous proposons une méthode qui croise l'utilisation de la métrique Drop-DTW et l'approche DBA pour la construction de séries temporelles moyennes. Ces approches sont adaptées pour des séquences d'évènements datés et nous en dérivons l'algorithme HIERASTISEQ qui catégorise des séquences temporelles. Cette approche est évaluée sur des données synthétiques et réelles.

1 Introduction

Les séquences temporelles, composées d'évènements datés, sont des types de données fréquemment rencontrés dans des domaines d'applications variés tels que l'analyse de logs informatiques, l'analyse de parcours de vie en science sociale (Gabadinho et al., 2011) ou encore l'analyse de parcours de soins (Rama et al., 2019). Une particularité de ce type de données est qu'il n'existe pas de représentation vectorielle naturelle pour représenter l'information qu'elles contiennent. En particulier, elles sont formées par deux dimensions très différentes : symboliques et temporelles ; et, d'autre part, leur taille varie généralement d'un exemple à l'autre (e.g. dans un parcours de soins, certains patients ont plus d'évènements médicaux que d'autres). Cette difficulté à modéliser les données dans un espace vectoriel rend difficile l'adaptation des méthodes usuelles d'analyse de données (méthodes de classification, de catégorisation ou de prévision). Si certaines approches se passent d'une représentation vectorielle (Egho et al., 2015; Gay et al., 2015), cet article vise l'adaptation de méthodes classiques.

Au cœur de l'adaptation des méthodes d'analyse de données se trouve la question de la formalisation d'une topologie des objets à représenter. Ces méthodes s'appuient sur les propriétés des espaces métriques pour généraliser des observations éparses et bruitées. La formalisation d'un tel espace revient à définir une métrique entre les séquences temporelles. Ce problème de la définition de métrique pour des séquences temporelles est très similaire à celui rencontré pour l'analyse de séries temporelles (données temporelles où les valeurs sont numériques,

et non évènementielles, et généralement avec un échantillonnage temporel régulier et dense). Ainsi, pour l'analyse de séries temporelles, toute une collection de métriques a été proposée et permet de les analyser (Tavenard et al., 2020). Un des enjeux de l'utilisation de ces méthodes est d'identifier la métrique qui dispose des propriétés appropriées à la tâche d'analyse que l'on souhaite mener (p. ex., les propriétés d'une distance) et dont la sémantique donnera les résultats attendus. Intuitivement, il faut qu'une métrique de similarité produise une valeur de similarité haute lorsque les exemples sont considérés similaires par les experts du domaine.

Pour le cas des données séquentielles, peu de travaux se sont intéressés à l'adaptation des méthodes d'analyse de données ; mais il existe des besoins, notamment pour exploiter des métriques dont la sémantique puisse mieux prendre en compte la dimension temporelle (et en particulier les délais entre la survenue des évènements). En effet, la plupart des solutions actuelles ne prennent généralement en compte que la séquentialité dans les données.

Dans ce travail, nous étendons la métrique de Drop-DTW (Dvornik et al., 2021) qui a été proposée pour évaluer des distances entre séquences en tenant compte à la fois des aspects séquentiels, en adaptant la Dynamic Time Warping (Sakoe et Chiba, 1978), et en y ajoutant une prise en charge d'évènements anodins, qui peuvent être écartés – *dropped*. Cette métrique dispose des caractéristiques sémantiques qui nous semblent adéquates pour notre cas d'application – *i.e.*, la catégorisation de parcours de soins, mais cette métrique n'étant pas une distance, elle ne permet pas de construire une séquence moyenne ni d'être utilisée dans ces méthodes classiques de catégorisation telles que l'algorithme des K-Means ou la classification ascendante hiérarchique.

Dans cet article, nous proposons donc HIERASTISEQ (*catégorisation Hiérarchique Ascendant de Time Sequences*) qui est une adaptation d'une classification ascendante hiérarchique pour des séquences temporelles. La contribution principale est dans la proposition de la construction d'une séquence temporelle moyenne, inspirée de la méthode DBA (Petitjean et al., 2011) pour les séries temporelles. La construction de séquence moyenne répond à la fois aux besoins des algorithmes de catégorisation et aux besoins d'interprétation des groupes de séquences. Nous proposons ensuite l'application de cette méthode pour cartographier des parcours patients types. Dans le cadre du projet OPTISOINS [1] nous nous sommes en particulier intéressés aux patients ayant bénéficié d'une chirurgie d'exérèse pulmonaire à visée carcinologique. Les parcours de soins ont été reconstruits à partir des données collectées dans le système d'information (*Electronic Health Record*) du regroupement des hopitaux Franciliens (APHP). Le parcours de soins est ici la séquence temporelle des étapes clées de la prise en charge d'une pathologie.

2 Préliminaires : DTW, Drop-DTW et DBA

Dans cette section, on introduit les notations et on rappelle les principes de la DTW et de Drop-DTW. La DTW ayant été initialement définie pour des séries temporelles, nous l'adaptons pour des séquences temporelles.

Soit Σ un ensemble fini de n types d'évènement. Une séquence temporelle est une suite de paires $\langle (s_i, t_i) \rangle$ où $s_i \in \Sigma$ est un type d'évènement et $t_i \in \mathbb{R}$ est la date de l'évènement.

1. https://www.bernoulli-lab.fr/project/optisoins/

(s_i, t_i) dénote un évènement de la séquence. Par convention, les évènements d'une séquence sont ordonnés par les dates.

La Dynamic Time Warping (DTW) (Sakoe et Chiba, 1978) calcule l'alignement optimal entre deux séries soumises à certaines contraintes. On adapte ici les définitions au cas des séquences temporelles. Soit $X = \langle (x_1, t_1^x), \ldots, (x_N, t_N^x) \rangle$ et $Z = \langle (z_1, t_1^z) \ldots, (z_K, t_K^z) \rangle$ les séquences d'entrée, où N, K sont les longueurs respectives des séquences. Un alignement entre les deux séquences est défini comme une matrice binaire, $M \in \{0,1\}^{K \times N}$ où $M_{i,j} = 1$ si z_i est apparié à x_j, et 0 sinon. L'appariement d'un élément z_i à un élément x_j a un coût $C_{i,j}$ entre les éléments i et j des séquences. La DTW trouve l'alignement M^* entre les séquences Z et X qui minimise le coût global :

$$M^* = \underset{M \in \mathcal{M}}{\operatorname{argmin}} \langle M, C \rangle = \underset{M \in \mathcal{M}}{\operatorname{argmin}} \sum_{i,j} M_{i,j} C_{i,j}, \tag{1}$$

où $\langle M, C \rangle$ est le produit de Frobenius et $\mathcal{M}$ est l'ensemble de tous les alignements réalisables qui satisfont aux contraintes suivantes : monotonie, continuité et correspondance des extrémités ($M_{1,1} = M_{K,N} = 1, \forall M \in \mathcal{M}$).

Drop-DTW (Dvornik et al., 2021) étend l'ensemble des alignements réalisables, $\mathcal{M}$, à ceux qui adhèrent uniquement à la contrainte de monotonie. Par conséquent, contrairement à DTW, des évènements peuvent être éliminés du processus d'alignement. Drop-DTW résout l'alignement temporel optimal tout en permettant d'écarter certains éléments. Plus précisément, la mise à l'écart d'un élément x_j signifie qu'il n'y a pas d'élément dans Z qui a été apparié à x_j. Pour tenir compte des éléments non-appariés dans le calcul du coût d'alignement, Dvornik et al. (2021) ajoutent un nouveau coût de mise à l'écart : $\delta \in \mathbb{R}^+$. L'appariement optimal peut alors être défini comme suit :

$$M^* = \underset{M \in \overline{\mathcal{M}}}{\operatorname{argmin}} \langle M, C \rangle + \delta . (P_z(M) + P_x(M)) \tag{2}$$

où $\overline{\mathcal{M}}$ est l'ensemble des matrices binaires satisfaisant uniquement la contrainte de monotonie, et $P_x(M)$ est un vecteur dont le j-ième élément est égal à 1 si $M_{:,j} = \mathbf{0}$ et à 0 dans le cas contraire ; $P_z(M)$ est défini de la même manière, mais sur les lignes. De même que la DTW, la Drop-DTW peut être évaluée efficacement par des techniques de programmation dynamique.

Le problème de la génération d'une série temporelle moyenne au sens de la DTW – qui n'a pas les propriétés d'une distance – a été abordé notamment au travers de l'algorithme DBA (Petitjean et Gançarski, 2012). L'algorithme DBA est une méthode de calcul de série « moyenne » qui consiste à affiner itérativement une série (initiale aléatoire), afin de minimiser sa distance quadratique (DTW) par rapport aux séries à moyenner. À chaque itération, l'algorithme effectue des barycentres « verticaux » des éléments alignés des séries temporelles.

La méthode DBA a été proposée pour des ensembles de séries temporelles pour lesquels la définition d'un barycentre « vertical » correspond à faire des moyennes dans l'espace des réels (valeurs des séries). Il ne s'adapte donc pas immédiatement à des séquences temporelles. Dans la suite, nous proposons donc une méthode, nommée HIERASTISEQ, qui s'inspire de DBA pour créer des séquences moyennes au sens de la Drop-DTW, tenant compte de la possibilité de mise à l'écart.

3 Méthode HIERASTISEQ

Nous présentons HIERASTISEQ (*Hierarchical Ascendant clustering of Time Sequences*), une méthode de catégorisation de séquences temporelles selon leurs points communs entre évènements tout en prenant en compte leurs différences temporelles. Cette méthode s'articule autour de 4 propositions :

1. une représentation probabiliste des séquences temporelles ; Cette représentation modélise les évènements dans un espace vectoriel pour adapter la technique des moyennes « verticales ».
2. l'utilisation de la Drop-Dynamic Time Warping pour comparer les séquences.
3. l'adaptation de DBA (Petitjean et al., 2011) pour le calcul d'une séquence moyenne sur la base de la métrique Drop-DTW.
4. l'utilisation d'une catégorisation hiérarchique exploitant le calcul des séquences temporelles moyennes. Le choix d'une catégorisation hiérarchique est motivé par la facilité d'interprétation des résultats.

HIERASTISEQ désigne ici la méthode de catégorisation qui construit des groupes de séquences temporelles en tenant compte à la fois du type et des dates des évènements. Dans la suite de cette partie, on détaille les trois premières propositions.

3.1 Représentation probabiliste des séquences

Afin de se ramener à la situation proche de DBA dans laquelle les éléments sont représentés dans un espace vectoriel, on propose de plonger l'espace des séquences, introduit dans la section précédente, dans un espace plus grand où les évènements sont représentés par leur *one-hot encoding* qui est une représentation vectorielle. À la place de représenter une séquence d'évènements, on représente une séquence de distributions d'évènements :

$$\langle([d_1^1, \dots d_1^n], t_1), \dots, ([d_K^1, \dots d_K^n], t_K)\rangle$$

où $[d_\cdot^1, \dots d_\cdot^n] \in [0,1]^n$, tel que $\sum_i d_\cdot^i = 1$ est une distribution de probabilités des évènements.

Exemple 1 (Représentation probabiliste d'une séquence temporelle)**.** *Soient* $\Sigma = \{A, B, C, D, E, F\}$ *un ensemble de types d'évènements et* $s_1 = \langle(A,0),(B,6)\rangle$*,* $s_2 = \langle(A,0),(B,6),(C,1000)\rangle$ *et* $s_3 = \langle(D,10),(E,16),(F,1000)\rangle$ *trois séquences temporelles. Ces séquences peuvent être représentées en séquences probabilistes par :*

$$s_1 = \langle([1,0,0,0,0,0],0.0),([0,1,0,0,0,0],6.0)\rangle$$
$$s_2 = \langle([1,0,0,0,0,0],0.0),([0,1,0,0,0,0],6.0),([0,0,1,0,0,0],1000.0)\rangle$$
$$s_3 = \langle([0,0,0,1,0,0],10.0),([0,0,0,0,1,0],16.0),([0,0,0,0,0,1],1000.0)\rangle$$

3.2 Adaptation de la Drop-DTW pour les séquences temporelles

L'adaptation proposée de Drop-DTW porte sur la proposition d'une métrique entre évènements et sur l'ajout de contraintes temporelles à l'algorithme Drop-DTW.

L'algorithme du Dynamic Time Warping requiert la définition d'un coût entre deux évènements pour comparer les séquences. Afin de prendre en compte le temps et l'évènement, nous

Algorithme 1 : Drop-DTW avec contraintes de proximité

Données : $C \in \mathbb{R}^{M \times N}$: matrice de coûts ; $\delta \in \mathbb{R}^+$: drop cost ; $\sigma \in \mathbb{N}^+$, $\tau \in \mathbb{R}^+$: contraintes de proximité séquentielle/temporelle

$D^+_{0,0} \leftarrow 0$; $D^+_{i,0} \leftarrow +\infty$; $D^+_{0,j} \leftarrow +\infty$;
$D^-_{0,0} \leftarrow 0$; $D^-_{i,0} \leftarrow \sum_1^i d$; $D^-_{0,j} \leftarrow \sum_1^j d$;
for $i = 1$ *to* $N + 1$ **do**
  for $j = max(1, i - \sigma + 1)$ *to* $min(m + 1, i + \sigma)$ **do** // Contraintes Sakoe-Chiba
    if $distdate(i - 1, j - 1) \leq \tau$ **then** // Contrainte temporelle
      $D^+_{i,j} \leftarrow C_{i-1,j-1} + min(D_{i-1,j}, D_{i,j-1}, D^+_{i-1,j-1})$; // Coût alignement
      $D^-_{i,j} \leftarrow \delta + D_{i,j-1}$; // Coût de *drop*
      $D_{i,j} \leftarrow min(D^+_{i,j}, D^-_{i,j})$; // Coût final
    else
      $D_{i,j} \leftarrow +\infty$;
return $D_{M,N}$

avons décidé de pondérer : un coût de distance en temps c_t, et un coût de différence d'évènement c_e. Ayant introduit la représentation probabiliste des séquences temporelles, la distance entre deux évènements $e_a = ([d^1_a, \dots d^n_a], t_a)$ et $e_b = ([d^1_b, \dots d^n_b], t_b)$, est donnée par :

$$d(e_a, e_b) = p_t.||\boldsymbol{d}_a - \boldsymbol{d}_b||_2 + p_e.(t_b - t_a)^2$$

où $p_t, p_e \in \mathbb{R}_+$ sont des pondérations de deux termes : une distance entre évènements probabilistes et un écart de temps. Élever l'écart de temps au carré a un sens car plus deux évènements sont éloignés et plus il est important qu'ils ne soient pas facilement associables. d est une métrique qui dispose des propriétés d'une distance.

Nous pouvons désormais adapter la métrique entre deux séquences temporelles. Nous avons retenu l'utilisation de la Drop-DTW pour sa capacité à prendre en compte des évènements aberrants ou manquants. Nous proposons également d'ajouter des contraintes sur les appariements en ne permettant des appariements qu'entre évènements proches dans la séquence (contrainte déjà introduite dans la DTW) et également proches dans le temps. L'algorithme 1 présente l'algorithme de calcul de la distance proposée entre deux séquences temporelles où la matrice de coût a été calculée à partir de la distance entre évènements, d.

Exemple 2 (Illustration de l'intérêt du drop-cost)**.** *Considérant les séquences de l'exemple 1, les séquences s_1 et s_2 semblent proches et pourraient présenter un intérêt à être regroupées tandis que la séquence 3 semble très différente. Sans drop-cost, les séquences 2 et 3 seraient regroupées avant les séquences s_1 et s_2. En utilisant le drop-cost, le dernier évènement de la séquence 3 ne serait pas pris en compte et le coût serait simplement remplacé par δ. Ainsi, elles seraient jugées plus ressemblantes que les séquences 2 et 3.*

La métrique proposée est paramétrée par le rapport $\frac{p_t}{p_e}$ et par le *drop-cost* δ. Nous explicitons ici comment déterminer ces valeurs relativement à une limite de délai d'indifférence entre évènements, notée Δt. Cette limite est estimable par les experts. Ensuite, le rapport $\frac{p_t}{p_e}$ peut être déduit du délai Δt. Pour les évènements localisés dans l'intervalle de temps $[t-\Delta t, t+\Delta t]$ l'importance du type d'évènement doit être plus grande que l'importance du temps. Il faut alors choisir p_e au moins Δt fois supérieur à p_t.

Algorithme 2 : TSR : calcul d'une séquence temporelle moyenne

Données : $\mathcal{S} = \{s_1, s_2, ..., s_m\}$: liste de séquences temporelles
$s_r \leftarrow alea(\mathcal{S})$;
while do
  $\mathcal{E} \leftarrow [\emptyset, \ldots, \emptyset]$; // $|r|$ ensembles vides
  $\mathcal{T} \leftarrow [\emptyset, \ldots, \emptyset]$; // idem
  for $s = s_1, \ldots s_{r-1}, s_{r+1}, \ldots, s_m$ **do**
    $M^* \leftarrow DropDTW(s_r, s)$; // calculer l'alignement entre s_r et s
    for $(i_r, i_s) \in M^*$ **do**
      $\mathcal{E}_{i_r} \leftarrow \mathcal{E}_{i_r} \cup \{s[i_s].\boldsymbol{d}\}$; // ajout evt de s aligné à i_r
      $\mathcal{T}_{i_r} \leftarrow \mathcal{T}_{i_r} \cup \{s[i_s].t\}$; // ajout date de s alignée à i_r
  $\forall i,\ \tau_i \leftarrow \frac{1}{|\mathcal{T}_{i_r}|} \sum_{t \in \mathcal{T}_{i_r}} t$; // calcul des moyennes verticales
  $\forall i,\ \mathcal{H}_i \leftarrow \frac{1}{|\mathcal{T}_{i_r}|} \sum_{\boldsymbol{d} \in \mathcal{E}_{i_r}} \boldsymbol{d}$;
  $s_r \leftarrow (\mathcal{H}, \tau)$; // séquence moyenne affinée
return s_r

Le *drop-cost* se déduit également du Δt, *i.e.* le délai à partir duquel il n'y a plus d'intérêt à associer deux évènements. On propose ainsi de définir le *drop-cost* par :

$$\delta = \frac{1}{\Delta t^2}(\Delta t + t_{max})^2 + 1$$

avec t_{max} la distance en temps maximum à prendre en compte.

3.3 Construction d'une séquence moyenne

Soit $\mathcal{S} = \{s_1, \ldots, s_m\}$ un ensemble de séquences temporelles, l'Algorithme 2 détaille les étapes de la génération d'une séquence moyenne en s'inspirant du principe des moyennes « verticales » de DBA (Petitjean et al., 2011). Dans un premier temps, une séquence temporelle de référence s_r est choisie dans $\mathcal{S}$. Ensuite, la distance de s_r avec chacune des autres séquences est calculée pour en récupérer l'alignement optimal M^* représenté par les indices i_r et i_s. Pour chaque évènement de s_r, les lignes 7 à 9 collectent, d'une part, la liste des évènements appariés dans les séquences et, d'autre part, leurs dates pour en construire des moyennes (l. 12-13). La moyenne des évènements est construite comme l'histogramme des distributions des évènements. Finalement, la ligne 14 crée une nouvelle séquence moyenne qui peut être affinée dans une itération suivante, jusqu'à convergence. Dans notre implémentation, on suppose la convergence après un nombre fixe d'itérations.

De manière similaire à DBA, les évènements des séquences probabilistes sont représentés dans un espace vectoriel. Ainsi, la nouvelle séquence moyenne est toujours plus proche (selon la Drop-DTW) des séquences dont elle fait la moyenne, et l'algorithme converge bien. [2] De plus, cet algorithme préserve la capacité de Drop-DTW à permettre les décalages temporels et à traiter les événements manquants ou aberrants.

Le choix de la séquence temporelle de la ligne 1 est déterminant pour le résultat. Nous avons opté pour prendre aléatoirement une séquence de la taille la plus longue.

2. La preuve détaillée est proposée en matériel supplémentaire.

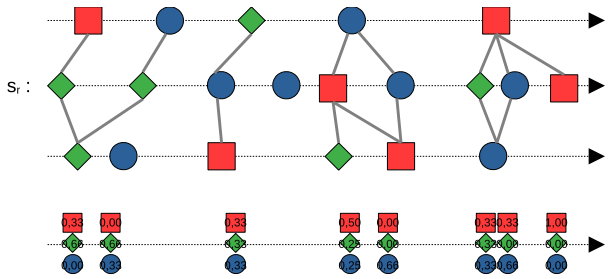

FIG. 1 – *Exemple de calcul d'une séquence moyenne (en bas) de trois séquences temporelles (en haut). Le type des évènements est donné par la couleur et la forme géométrique, leur position horizontale indique leur date. Pour la séquence probabiliste, un évènement est une distribution des trois types d'évènements. Les traits grisés illustrent les appariements des séquences avec* s_r.

La Figure 1 illustre une itération du calcul d'une moyenne pour trois séquences. Les appariements par la drop-DTW sont représentés en gris. En se servant de s_r comme référence, la moyenne obtenue est la séquence probabiliste représentée en bas de l'illustration. On constate que l'utilisation de drop-DTW permet d'écarter certains évènements de l'appariement.

Cette méthode est utilisée pour construire le représentant d'un groupe dans une classification hiérarchique ascendante. Ainsi, HIERASTISEQ agrège récursivement les séquences en groupes homogènes, selon la métrique proposée. Pour des raisons de place, nous ne détaillons pas plus cet algorithme classique de catégorisation.

4 Expérimentations et résultats

Dans cette section, nous présentons tout d'abord des expérimentations sur données synthétiques dont nous connaissons les résultats attendus. Nous appliquons ensuite la méthode proposée sur des parcours de soins réels et nous comparons les résultats obtenus à ceux de l'analyse de séquences d'état (SSA) (Gabadinho et al., 2011). La SSA, et en particulier l'outil TraMineR, est usuellement utilisée pour catégoriser des parcours de soins (Roux et al., 2019).

4.1 Tests sur des données synthétiques

Nous avons, dans un premier temps, effectué plusieurs tests sur des données synthétiques afin d'évaluer l'efficacité de la méthode en fonction de plusieurs cas de figures.

4.1.1 Effets du ratio $\frac{p_t}{p_e}$ sur les regroupements

Pour ce premier test, nous créons deux séquences temporelles modèles qui diffèrent selon une ou plusieurs des dimensions que l'on souhaite étudier : la taille de la séquence, la nature des évènements et les dates des évènements. Il y a donc $8 = 2^3$ variations possibles. Un jeu de données est constitué de 15 séquences de chacune des 8 variations possibles et la séquence

		Prédit	
Réel		0	15
		15	15

		Prédit	
Réel		15	0
		0	30

TAB. 1 – *Matrices de confusion pour l'expérimentation avec évènements aberrants, à gauche pour* $p_t = \frac{1}{9}$, $p_e = 1$,$\delta = +\infty$, *à droite pour* $p_t = \frac{1}{9}$, $p_e = 1$, $\delta = 4$

de base (soit $135 = 9 \times 15$ séquences au total). Les dates de ces séquences temporelles sont obtenues et rendues aléatoires par l'ajout d'un bruit uniforme sur la date. L'objectif de cette expérimentation est d'évaluer la précision de l'identification des 9 catégories de séquences.

Nous avons testé deux jeux de paramètres, tout d'abord, nous avons choisi le ratio $\frac{p_t}{p_e} = \frac{1}{9}$, avec $\delta = +\infty$. Dans cette configuration, le coût d'un évènement est plus important que le coût en temps pendant 3 jours autour de l'évènement. Le second jeu de paramètres utilisé est $\frac{p_t}{p_e} = \frac{1}{400}$, toujours avec $\delta = +\infty$. Cette fois, le coût de différence entre évènements est plus important que le coût en temps au cours des 20 premiers jours.

Dans le cas de la première configuration, nous observons que l'ensemble des groupes ont été correctement identifiés (score de Kappa de 1). En revanche, dans le cas de la seconde configuration, nous avons observé une dégradation des résultats avec un score de Kappa de 0.62. Cette dégradation était attendue car les distances en temps maximum observées entre évènements sont autour de 20. Par conséquent, la notion de temps n'est pratiquement jamais prise en compte par la métrique.

Cette expérimentation montre ainsi que le choix des paramètres est primordial mais qu'il peut se faire en prenant en compte la distance entre les évènements des séquences.

4.1.2 Effet du drop-cost

Nous pouvons ensuite tester l'effet du drop-cost sur un jeu de données synthétique, avec et sans drop-cost. On prend 3 séquences modèles dont 2 qui se ressemblent, excepté que l'une d'elle présente une valeur aberrante.

1. $\langle (D, 0.0)(E, 3.0)(F, 5.0)(D, 7.0)(D, 10.0)(E, 12.0)(F, 15.0)\rangle$
2. $\langle (E, 2.0)(A, 4.0)(D, 8.0)(C, 12.0)\rangle$
3. $\langle (D, 0.0)(E, 3.0)(F, 5.0)(D, 7.0)(D, 10.0)(E, 12.0)(F, 15.0)(D, 1500.0)\rangle$

Intuitivement, on souhaite rassembler les séquences 1 et 3. On applique HIERASTISEQ sans drop-cost et on obtient la matrice de confusion de la Table 1 à gauche dont le score de Kappa est de -0.5 (désaccord). On constate que l'algorithme a associé l'ensemble issu de la séquence 1 et l'ensemble issu de la séquence 2.

L'ajout du drop-cost coïncidant avec les distances entre les séquences permet d'obtenir la matrice de confusion de la Table 1 à droite, dont le score de Kappa est de 1 (accord total). Le drop-cost permet de négliger le dernier évènement considéré comme aberrant et les résultats correspondent aux attentes.

Nous avons également effectué une expérimentation similaire avec des valeurs manquantes. Cette fois, les séquences modèles sont les suivantes :

1. $\langle (D, 0.0), (E, 3.0), (F, 5.0), (D, 7.0)\rangle$

		Prédit	
Réel		0	15
		15	15

		Prédit	
Réel		15	0
		0	30

TAB. 2 – *Matrices de confusion avec évènements manquants, à gauche pour* $p_t = \frac{1}{9}$, $p_e = 1$,$\delta = +\infty$, *à droite pour* $p_t = \frac{1}{9}$, $p_e = 1$, $\delta = 4$

2. $\langle (E, 2.0), (A, 4.0), (E, 7.0), (D, 8.0), (D, 10.0), (E, 12.0) \rangle$
3. $\langle (D, 0.0), (E, 2.0), (E, 3.0), (F, 5.0), (D, 7.0), (F, 9.0), (A, 13.0) \rangle$

On remarque alors que l'ensemble des évènements datés de la séquence 1 sont présents dans la séquence 3 mais qu'ils sont tous différents de la séquence 2. Nous voudrions alors rassembler les séquences issues de 1 et 3. En gardant les mêmes paramètres que précédemment. Sans drop-cost, on obtient la matrice de confusion de la Table 2 à gauche. Ce résultat correspond aux attentes car les séquences issues de 2 et 3 sont alors associées car plus proches en temps. Cependant, en ajoutant un $\delta = 4$, nous obtenons la matrice de confusion de la Table 2 à droite, ce qui est logique car les évènements les plus éloignés en temps ne sont plus associés.

4.2 Application aux parcours de soins

Les tests sur valeurs synthétiques présentent les résultats attendus et valident ainsi l'efficacité de la catégorisation dans les situations décrites. On poursuit à présent l'étude de la méthode par une expérimentation sur données réelles pour évaluer sa capacité à identifier des groupes pertinents. Dans cette expérimentation, on compare les résultats de HIERASTISEQ avec ceux de TraMineR, approche déjà utilisée pour la catégorisation de parcours de soins.

Cette expérimentation est réalisée sur l'Entrepôt de Données de Santé (EDS) de l'AP-HP. Nous disposons des évènements ayant eu lieu dans les 120 jours autour d'une exérèse pulmonaire ainsi que la date de décès s'il y en a une. La majorité des évènements ayant lieu dans les 90 jours autour de l'exérèse de référence, nous avons décidé de ne garder que les évènements de cet intervalle. Les évènements plus éloignés sont cliniquement moins intéressants à étudier. Pour des raisons cliniques, nous avons également décidé de ne garder, pour chaque patient, que la première occurrence de chimiothérapie. Nous avons fait le même choix pour les radiothérapies et immunothérapies. Les décès sont également retirés car ils ne font pas partie du parcours de soins. Nous conservons néanmoins l'ensemble de ces données pour l'affichage de résultats. Pour des raisons de temps de calculs, nous ne présentons que les résultats des tests sur 500 personnes choisis aléatoirement. Le traitement des données complètes nécessite entre 2 et 3 heures.

Les paramètres retenus sont 5 groupes, $\Delta t = 7$, $t_{max} = 16$, et $\delta \approx 11.8$ (calculé selon la méthode proposée). Pour le choix du nombre de groupes, différentes valeurs ont été testées. En augmentant le nombre de groupe, on remarque que ce sont avant tout des groupes de petites tailles qui se scindent. Il n'y a donc pas nécessairement d'intérêt à augmenter ce nombre. Le Δt est établi à 7 jours car, dans notre jeu de données, il peut être intéressant d'associer les mêmes types d'évènements ayant eu lieu à moins d'une semaine d'intervalle. Enfin, toujours à partir du jeu de données, nous considérons qu'il n'y a plus d'intérêt à associer deux évènements éloignés de plus de 16 jours. De son côté, TraMineR nécessite la conversion des séquences

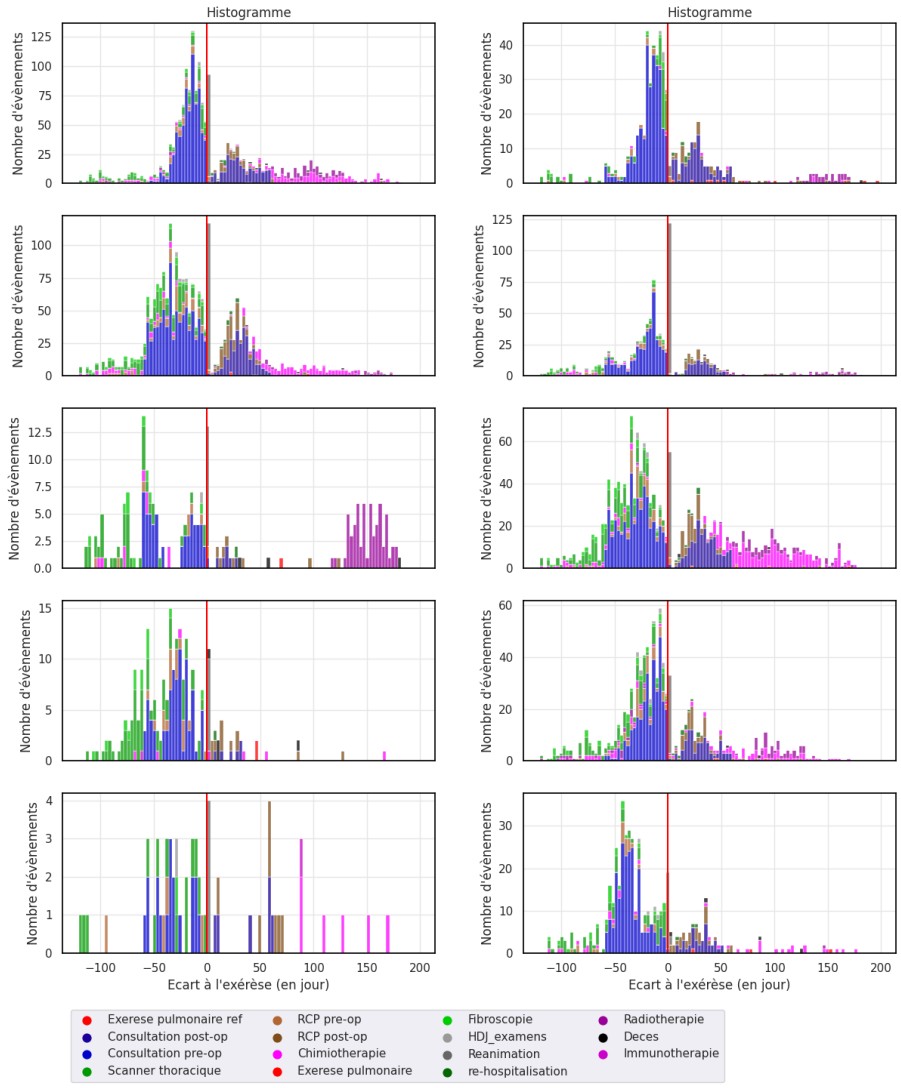

FIG. 2 – *Résultats de la catégorisation avec* HIERASTISEQ *(à gauche) et TraMineR (à droite) pour 5 groupes et 500 patients*

temporelles en séries temporelles. TraMineR étant sensible aux séries éparses, nous avons augmenté l'échantillonnage en temps, passant de 1 à 3 jours pour les calculs et nous avons ajouté l'évènement vide qui correspond à l'absence de données sur une période de 3 jours.

Les Figures 2 correspondent aux groupes représentés comme des histogrammes d'évènements dans le temps pour chaque groupe, respectivement pour HIERASTISEQ et TraMineR. On remarque que deux groupes principaux se détachent pour HIERASTISEQ. Les différences entre les groupes se font notamment en fonction du temps et des différences d'évènements

comme attendu. Par exemple, le pic de consultation pre-op survient plus tard dans le groupe 1 que dans le groupe 2 ou bien qu'il y en a 2 différents dans le groupe 3. On remarque également que les chimiothérapies sont réparties dans les deux premiers groupes. Enfin, on voit que le nombre de fibroscopies est une différence majeure entre le groupe 1 et 2. Ces groupes semblent donc révélateurs de certains parcours de soins.

Ensuite, on remarque pour les groupes de TraMineR que la présence de certains évènements a un fort impact dans les regroupements obtenus. On voit par exemple que la quasi-totalité des chimiothérapies est regroupée dans le groupe 4 ou que les réanimations sont majoritairement dans le groupe 2 et 3. En revanche, on ne remarque pas de différences temporelles majeures entre les groupes.

En complément des résultats de la Figure 2, nous avons procédé à une analyse quantitative (matrices de confusion) et qualitative (analyse des groupes). L'analyse des confusions montre qu'il n'y a pas de correspondance entre les groupes bien qu'il semble y avoir des exclusions communes. La première différence entre TraMineR et notre proposition est l'impact de la donnée temporelle. Cette importance du temps peut être une explication de la différence de taille entre les groupes. On remarque également que les évènements pris en compte dans la catégorisation sont plus scindés avec TraMineR que pour nos résultats. La répartition des chimiothérapies dans les groupes est un bon exemple de ce phénomène. Dans notre cas, les valeurs sont comprises entre 9 et 20% des patients tandis que dans le cas de TraMineR un groupe en contient 44%, un autre à 28% et les 3 autres moins de 7%. Notre algorithme semble prendre en compte à la fois des données sur les évènements, des données sur le temps et l'ordre d'apparition grâce à la DTW, alors que TraMineR priorise les évènements semblables et l'ordre d'apparition.

5 Conclusion

Nous avons proposé une méthode de catégorisation de séquences temporelles inspirée des techniques existantes pour les séries temporelles. Ainsi, l'utilisation de ces objets permet la superposition d'évènements, un échantillonnage en temps irrégulier et l'utilisation de séquences clairsemées. Tous ces avantages sont très intéressants d'un point de vue clinique pour la description des parcours de soins. Nous avons montré qu'en utilisant la Drop-DTW, l'algorithme devenait moins sensible aux valeurs manquantes et aberrantes, ce qui est également utile dans l'application aux parcours de soins. Puis, nous avons testé notre méthode sur des données réelles de l'étude OPTISOINS pour notre approche comparée à TraMineR. À la vue des premiers résultats, HIERASTISEQ semble être cliniquement prometteur pour identifier des parcours moyens.

Cependant, le grand nombre de paramètres et le temps de calcul nécessaire sont des limites de la méthode. Pour le nombre de paramètres, notre effort s'est porté sur un guide du choix des valeurs des paramètres, que les expérimentations tendent à valider. En perspective, une étude plus poussée de l'effet des paramètres doit encore est menée. De plus, nous devons à présent faire valider l'intérêt clinique des parcours moyens par des cliniciens.

Remerciements Une partie des recherches présentées dans cet article est subventionnée par la Fondation de l'AP-HP, dans le cadre de la Chaire AI-RACLES et a reçu l'accord du Comité scientifique et éthique du CDW de l'AP-HP (CSE-20-01-OPTISOINS).

Références

Dvornik, M., I. Hadji, K. G. Derpanis, A. Garg, et A. Jepson (2021). Drop-DTW : Aligning common signal between sequences while dropping outliers. *Advances in Neural Information Processing Systems (NIPS) 34*, 13782–13793.

Egho, E., D. Gay, M. Boullé, N. Voisine, et F. Clérot (2015). A parameter-free approach for mining robust sequential classification rules. In *Proceedings of International Conference on Data Mining (ICDM)*, pp. 745–750.

Gabadinho, A., G. Ritschard, N. S. Müller, et M. Studer (2011). Analyzing and visualizing state sequences in *R* with TraMineR. *Journal of Statistical Software 40*(4), 10.

Gay, D., R. Guigourès, M. Boullé, et F. Clérot (2015). TESS : temporal event sequence summarization. In *Proceedings of the International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 1–10.

Petitjean, F. et P. Gançarski (2012). Summarizing a set of time series by averaging : From steiner sequence to compact multiple alignment. *Theoretical Computer Science 414*(1), 76–91.

Petitjean, F., A. Ketterlin, et P. Gançarski (2011). A global averaging method for dynamic time warping, with applications to clustering. *Pattern Recognition 44*(3), 678–693.

Rama, K., H. Canhão, A. M. Carvalho, et S. Vinga (2019). AliClu - temporal sequence alignment for clustering longitudinal clinical data. *BMC Medical Informatics and Decision Making 19*(1), 289.

Roux, J., O. Grimaud, et E. Leray (2019). Use of state sequence analysis for care pathway analysis : The example of multiple sclerosis. *Statistical Methods in Medical Research 28*(6), 1651–1663.

Sakoe, H. et S. Chiba (1978). Dynamic programming algorithm optimization for spoken word recognition. *IEEE transactions on acoustics, speech, and signal processing 26*(1), 43–49.

Tavenard, R., J. Faouzi, G. Vandewiele, F. Divo, G. Androz, C. Holtz, M. Payne, R. Yurchak, M. Rußwurm, K. Kolar, et al. (2020). Tslearn, a machine learning toolkit for time series data. *The Journal of Machine Learning Research 21*(1), 4686–4691.

Summary

With the aim to improve care in the future, it is worth offering clinicians an objective view of their practices. The clustering of care pathways meets this objective of revealing homogeneous groups of patients. The difficulty in clustering care pathways, represented by sequences of timestamped events, lies in defining a semantically appropriate metric and clustering algorithms. In this article, we propose a method that combines the use of the Drop-DTW metric and the DBA approach for the construction of average time series. These approaches are adapted for sequences of timestamped events, and we derive the HIERASTISEQ algorithm for clustering time sequences. This approach is evaluated on synthetic and real data.