Extraction de chroniques discriminantes

Yann Dauxais, David Gross-Amblard, Thomas Guyet, André Happe

Extraction de chroniques discriminantes

Yann Dauxais*, David Gross-Amblard*, Thomas Guyet** et André Happe***

*Université Rennes-1/IRISA
prenom.nom@irisa.fr
**Agrocampus-Ouest/IRISA
*** CHRU Brest - équipe REPERE

Résumé. L'extraction de motifs séquentiels vise à extraire des comportements récurrents dans un ensemble de séquences. Lorsque ces séquences sont étiquetées, l'extraction de motifs discriminants engendre des motifs caractéristiques de chaque classe de séquences. Cet article s'intéresse à l'extraction des chroniques discriminantes où une chronique est un type de motif temporel représentant des durées inter-évènements quantitatives. L'article présente l'algorithme DCM dont l'originalité réside dans l'utilisation de méthodes d'apprentissage automatique pour extraire les intervalles temporels. Les performances computationnelles et le pouvoir discriminant des chroniques extraites sont évalués sur des données synthétiques et réelles.

1 Introduction

La fouille de données temporelles, *e.g.* séries temporelles ou des séquences, est dédiée à l'analyse de données portant une information temporelle. De telles données sont largement rencontrées dans les domaines tels que la médecine, l'ingénierie ou la finance. Pour ces domaines, l'aspect temporel est primordial et il est crucial de proposer des approches capables d'analyser ces données en tenant compte de cette spécificité. Dans cet article, nous nous concentrons sur l'extraction de motifs dans des séquences d'évènements datés. Nous cherchons à extraire des motifs pouvant servir à discriminer précisément des comportements associés aux séquences Fradkin et Mörchen (2015). Il s'agit, par exemple, d'associer une suite d'évènements à l'état pathologique d'un patient.

L'extraction de motifs séquentiels, *i.e.* tenant compte de la séquentialité des évènements, mais pas de leurs dates, a été largement étudiée et fait l'objet de plusieurs états de l'art (Masseglia et al., 2004; Mooney et Roddick, 2013). L'extraction de motifs séquentiels est efficace pour une contrepartie non-négligeable : l'information temporelle n'est pas prise en compte dans son intégralité. Les motifs décrivant une information temporelle riche tels que les motifs d'intervalles (Guyet et Quiniou, 2011) ou les chroniques (Dousson et Duong, 1999; Cram et al., 2012; Huang et al., 2012) captent une information plus riche. De ce fait, leur utilisation permet des prédictions plus précises.

D'autre part et bien que l'objectif d'extraction de motifs séquentiels soit la prédiction d'évènements, la plupart des approches de fouille se sont intéressées à extraire des motifs fréquents, *i.e.* qui apparaissent fréquemment dans la base d'exemples. L'extraction de motifs

fréquents *et discriminants* semble être une approche plus intéressante en vue de proposer des motifs qui permettront une prédiction plus précise. De plus, les motifs discriminants sont intéressants pour réduire le nombre de motifs à extraire. Face au problème bien connu du déluge de motifs, cette approche semble pertinente pour se focaliser uniquement sur des motifs d'intérêt.

Dans l'objectif de proposer aux utilisateurs des motifs qui permettent une discrimination précise de séquences, nous nous intéressons à la fouille de motifs temporels fréquents et discriminant. Pour la grande expressivité des chroniques et leurs propriétés algorithmiques, nous nous intéressons plus précisément à l'extraction de chroniques fréquentes et discriminantes.

Cet article propose l'algorithme DCM pour extraire des chroniques discriminantes d'un jeu de données temporelles étiqueté. Sa contribution majeure réside dans l'utilisation d'un algorithme d'apprentissage de règles relationnelles pour extraire les contraintes temporelles discriminantes.

2 Travaux antérieurs

Fradkin et Mörchen (2015) ont comparé plusieurs méthodes d'extraction de motifs séquentiels discriminants allant du post-traitement de l'ensemble des motifs fréquents à la construction d'un arbre de décision utilisant les motifs séquentiels. Ce sont les algorithmes auxquels nous nous sommes comparés par la suite (voir section 5).

Des ordres partiels discriminants fermés sont extraits par Fabrègue et al. (2014) pour fouiller des données liées à des écosystèmes aquatiques pollués. Pour extraire ces motifs, les ordres partiels fermés sont d'abord extraits de chacune des bases de données puis un post-traitement est effectué pour ne retenir que ceux qui sont discriminants. L'ensemble des ordres partiels discriminants est un sous-ensemble de celui des chroniques discriminantes. La définition d'ordre partiel correspond à celle d'épisode (Mannila et al., 1997).

Des chroniques discriminantes ont déjà été extraites par Carrault et al. (2003). Ces chroniques permettaient de décrire des problèmes d'arythmies cardiaques sur les données ECG. L'inconvénient majeur de cette approche réside dans l'apport de connaissances expertes pour l'extraction et la reconnaissance des chroniques. En particulier, les intervalles temporels discriminants ne sont pas directement extraits mais spécifiés initialement suivant différentes étiquettes tels que « court », « normal » et « long ».

Les motifs utilisés dans ces travaux n'apportent pas la richesse d'information contenue par les chroniques ou ne l'extrait pas directement des données. De plus, ces travaux utilisent des approches basées principalement sur un post-traitement, c'est-à-dire que l'ensemble des motifs discriminants est extrait à partir d'un plus gros ensemble de motifs et non directement comme ce qui a pu être fait avec les motifs émergents (Dong et Li, 1999).

3 Définitions

Cette section commence par introduire les définitions utiles puis définit le problème d'extraction de chroniques discriminantes.

3.1 Séquences et chroniques

Soit $\mathbb{E}$ un ensemble de types d'évènement et $\mathbb{T}$ un domaine temporel tel que $\mathbb{T} \subseteq \mathbb{R}$, un **évènement** est un couple (e, t) tel que $e \in \mathbb{E}$ et $t \in \mathbb{T}$. L'ensemble $\mathbb{E}$ est supposé totalement ordonné et est noté $\leq_{\mathbb{E}}$. Une **séquence** est un triplet $\langle SID, \langle (e_1, t_1), (e_2, t_2), ..., (e_n, t_n) \rangle, C \rangle$ tel que SID est l'indice de la séquence, $\langle (e_1, t_1), (e_2, t_2), ..., (e_n, t_n) \rangle$ est une séquence finie d'évènements et $L \in \mathbb{L}$ est une étiquette. Les éléments de la séquence sont ordonnés selon un ordre $\prec$ défini par $\forall i, j \in [1, n],\ (e_i, t_i) \prec (e_j, t_j) \Leftrightarrow (i < j \wedge t_i < t_j) \vee (i = j \wedge e_i <_{\mathbb{E}} e_j)$.

Exemple 1 (Ensemble de séquences, $\mathcal{S}$)**.** Le tableau 1 représente un ensemble de 6 séquences contenant 5 types d'évènement (A, B, C, D et E) et étiquetées par deux labels différents ($\mathbb{L} = \{+, -\}$). Ce jeu de données sera réutilisé pour les exemples suivants.

SID	Séquence	Label
1	$(A,1), (B,3), (A,4), (C,5), (C,6), (D,7)$	+
2	$(B,2), (D,4), (A,5), (C,7)$	+
3	$(A,1), (B,4), (C,5), (B,6), (C,8), (D,9)$	+
4	$(B,4), (A,6), (E,8), (C,9)$	-
5	$(B,1), (A,3), (C,4)$	-
6	$(C,4), (B,5), (A,6), (C,7), (D,10)$	-

TAB. 1 – *Ensemble de six séquences appartenant à deux classes.*

Une **contrainte temporelle** est un quadruplet (e_1, e_2, t^-, t^+), noté $e_1[t^-, t^+]e_2$, tel que $e_1, e_2 \in \mathbb{E}$, $e_1 \leq_{\mathbb{E}} e_2$ et $t^-, t^+ \in \mathbb{T}$, $t^- \leq t^+$. Une contrainte temporelle $e_1[t^-, t^+]e_2$ est dite satisfaite par un couple d'évènements $((e, t), (e', t'))$ ssi $e = e_1$, $e' = e_2$ et $t' - t \in [t^-, t^+]$. On notera $e_1[a, b]e_2 \subseteq e_1'[a', b']e_2'$ ssi $e_1 = e_1'$, $e_2 = e_2'$ et $[a, b] \subseteq [a', b']$.

Une **chronique** est un couple $(\mathcal{E}, \mathcal{T})$ tel que $\mathcal{E} = \{\!\{e_1 ... e_n\}\!\}$, $e_i \in \mathbb{E}$ et $\forall i, j, 1 \leq i < j \leq n$, $e_i \leq_{\mathbb{E}} e_j$ et tel que $\mathcal{T}$ est un ensemble de contraintes temporelles tel qu'il existe une contrainte temporelle dans $\mathcal{T}$ pour chaque paire d'éléments de $\mathcal{E}$, *i.e.* $\forall e, e' \in \mathcal{E},\ e \leq_{\mathbb{E}} e',\ e[a, b]e' \in \mathcal{T}$. L'élément $\mathcal{E}$ sera appelé un **multiset**, *i.e.* $\mathcal{E}$ peut contenir plusieurs occurrences d'un même type d'évènement. Dans la mesure où la contrainte $e[a, b]e'$ est équivalente à $e'[-b, -a]e$, on utilise l'ordre sur les items, $\leq_{\mathbb{E}}$, pour orienter la contrainte entre deux évènements d'une chronique.

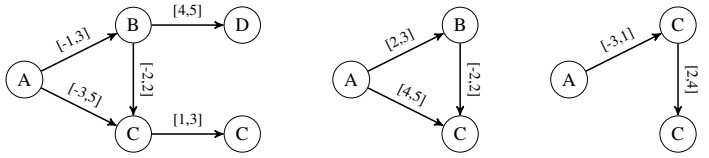

FIG. 1 – *Exemple de trois chroniques apparaissant dans le tableau 1 (cf. exemples 2 et 3). L'absence d'arc entre deux évènements traduit une contrainte de la forme* $[-\infty, \infty]$.

Exemple 2. La figure 1 illustre sous forme de graphe trois chroniques. La chronique $\mathcal{C} = (\mathcal{E}, \mathcal{T})$ où $\mathcal{E} = \{\!\{e_1 = A, e_2 = B, e_3 = C, e_4 = C, e_5 = D\}\!\}$ et $\mathcal{T} = \{e_1[-1, 3]e_2, e_1[-3, 5]e_3, e_2[-2, 2]e_3, e_2[4, 5]e_5, e_3[1, 3]e_4\}$ y est représentée à gauche. On remarque que ce graphe n'est pas complet. En l'absence d'arc entre deux évènements traduit une contrainte de la forme $[-\infty, \infty]$, *i.e.* qu'il n'y a pas de contrainte.

3.2 Support d'une chronique

Soient $s = \langle (e_1, t_1), ..., (e_n, t_n) \rangle$ une séquence et $\mathcal{C} = (\mathcal{E} = \{\!\{e'_1, ..., e'_m\}\!\}, \mathcal{T})$ une chronique. Une **occurrence** de $\mathcal{C}$ dans s est une sous-séquence d'évènements $\tilde{s} = \langle (e_{f(1)}, t_{f(1)}), ..., (e_{f(m)}, t_{f(m)}) \rangle$ tel qu'il existe une fonction $f : [1, m] \mapsto [1, n]$ injective telle que 1) $\forall i,\ e'_i = e_{f(i)}$ et 2) $\forall i, j,\ t_{f(j)} - t_{f(i)} \in [a, b]$ où $e'_i[a, b]e'_j \in \mathcal{T}$. Il faut noter que f n'est pas nécessairement croissante. Ceci résulte de la différence entre (i) l'ordre du multiset d'une chronique défini sur les items et (ii) l'ordre des évènements dans une séquence, $\prec$, définie par le domaine temporel. La chronique $\mathcal{C}$ **apparaît** dans s, noté $\mathcal{C} \in s$, s'il existe au moins une occurrence de $\mathcal{C}$ dans s. Le **support** d'une chronique $\mathcal{C}$ dans un ensemble de séquences $\mathcal{S}$ est le nombre de séquences dans lesquelles $\mathcal{C}$ apparaît : $support(\mathcal{C}, \mathcal{S}) = |\{S \mid S \in \mathcal{S}\ et\ \mathcal{C} \in S\}|$.

Exemple 3. La chronique $\mathcal{C}$, à gauche de la figure 1, apparaît dans les séquences 1, 3 et 6 du tableau 1. On remarque qu'il existe deux occurrences de $\mathcal{C}$ dans la séquence 1. On a $support(\mathcal{C}, \mathcal{S}) = 3$ et cette chronique est fréquente dans $\mathcal{S}$ pour tout seuil de fréquence minimal σ_{min} inférieur ou égal à 3. Les deux autres chroniques de la figure 1, que l'on nommera $\mathcal{C}_1$ et $\mathcal{C}_2$ de gauche à droite, apparaissent respectivement dans les séquences 1 et 3 et dans la séquence 6. On a donc $support(\mathcal{C}_1, \mathcal{S}) = 2$ et $support(\mathcal{C}_2, \mathcal{S}) = 1$.

3.3 Extraction de chroniques discriminantes

Soient deux bases de séquences $\mathcal{S}^+$ et $\mathcal{S}^-$ et deux paramètres σ_{min} et g_{min} donnés. Une chronique est dite **discriminante** pour $\mathcal{S}^+$ ssi $support(\mathcal{C}, \mathcal{S}^+) \geq \sigma_{min}$ et $support(\mathcal{C}, \mathcal{S}^+) \geq g_{min} \times support(\mathcal{C}, \mathcal{S}^-)$. Le **taux de croissance** $g(\mathcal{C}, \mathcal{S})$ d'une chronique est défini comme égal à $\frac{support(\mathcal{C}, \mathcal{S}^+)}{support(\mathcal{C}, \mathcal{S}^-)}$ si $support(\mathcal{C}, \mathcal{S}^-) > 0$ et égal à $+\infty$ si $support(\mathcal{C}, \mathcal{S}^-) = 0$.

Exemple 4. En reprenant la chronique $\mathcal{C}$ de la figure 1, $support(\mathcal{C}, \mathcal{S}^+) = 2$, $support(\mathcal{C}, \mathcal{S}^-) = 1$ donc $g(\mathcal{C}, \mathcal{S}) = 2$, c'est-à-dire que cette chronique est discriminante si $g_{min} \leq 2$. Pour les chroniques $\mathcal{C}_1$ et $\mathcal{C}_2$, $support(\mathcal{C}_1, \mathcal{S}^+) = 2$ et $support(\mathcal{C}_1, \mathcal{S}^-) = 0$ donc $g(\mathcal{C}_1, \mathcal{S}) = +\infty$ et $support(\mathcal{C}_2, \mathcal{S}^+) = 0$ et $support(\mathcal{C}_2, \mathcal{S}^-) = 1$ donc $g(\mathcal{C}_2, \mathcal{S}) = 0$. Donc, pour toute valeur de g_{min}, $\mathcal{C}_1$ est discriminante, mais pas $\mathcal{C}_2$.

L'utilisation d'une contrainte de fréquence en complément de la contrainte de discriminance évite des chroniques trop peu fréquentes et donc insignifiantes. Par exemple, une chronique telle que $\mathcal{C}_\infty$ pour laquelle $support(\mathcal{C}, \mathcal{S}^+) = 1$ et $support(\mathcal{C}, \mathcal{S}^-) = 0$ est considérée comme discriminante mais n'a que peu d'intérêt. Le second avantage de cette contrainte de fréquence minimale est de réduire fortement le nombre de motifs à extraire en tronquant une partie généralement importante de l'espace de recherche contenant les motifs non fréquents. Cette contrainte de fréquence est monotone et, en particulier, si une chronique [1] $(\mathcal{E}, \mathcal{T}_\infty)$ n'est pas fréquente, alors aucune chronique de la forme $(\mathcal{E}, \mathcal{T})$ ne le sera.

Extraire l'ensemble complet des chroniques discriminantes n'est pas intéressant à cause de nombreuses chroniques discriminantes similaires. Dans ce cas, il est préférable d'extraire les chroniques pour lesquelles les contraintes temporelles sont les plus larges, *i.e.* plus généralisatrices. L'approche proposée dans la section suivante est incomplète. Elle se focalise sur l'extraction efficace d'un sous-ensemble des chroniques discriminantes que nous cherchons à être signifiant.

1. $\mathcal{T}_\infty$ est l'ensemble de contraintes temporelles dont toutes les bornes sont fixées à ∞.

4 Algorithme DCM

Cette section présente l'algorithme DCM pour l'extraction des chroniques discriminantes.

Algorithme 1 Algorithme DCM pour l'extraction de chroniques discriminantes

Require: $\mathcal{S}^+, \mathcal{S}^-$: ensembles de séquences, σ_{min} : fréquence minimum, g_{min} : croissance minimum
1: $\mathbb{M} \leftarrow$ FINDMULTISET($\mathcal{S}^+, \sigma_{min}$) ▷ $\mathbb{M}$ est l'ensemble des multisets fréquents
2: $\mathbb{C} \leftarrow \emptyset$ ▷ $\mathbb{C}$ est l'ensemble des chroniques discriminantes
3: **for all** $ms \in \mathbb{M}$ **do**
4: **if** $support\left(\mathcal{S}^+, (ms, \mathcal{T}_\infty)\right) > g_{min} \times support\left(\mathcal{S}^-, (ms, \mathcal{T}_\infty)\right)$ **then**
5: $\mathbb{C} \leftarrow \mathbb{C} \cup \{(ms, \mathcal{T}_\infty)\}$ ▷ La chronique sans contrainte temporelle est discriminante
6: **else**
7: **for all** $\mathcal{T} \in$ EXTRACTDISCRCONSTRAINTS($\mathcal{S}^+, \mathcal{S}^-, ms, g_{min}, \sigma_{min}$) **do**
8: $\mathbb{C} \leftarrow \mathbb{C} \cup \{(ms, \mathcal{T})\}$ ▷ Ajout d'une nouvelle chronique discriminante
9: **return** $\mathbb{C}$

L'algorithme 1 illustre la procédure d'extraction des chroniques fréquentes discriminantes. Cette dernière comporte deux étapes : l'extraction des multisets fréquents, puis la spécification des contraintes temporelles des multisets.

Dans un premier temps, ligne 1, FINDMULTISET extrait les multisets d'items fréquents, $\mathbb{M}$. Ceci est réalisé par un algorithme de fouille d'itemsets fréquents en construisant un jeu de données pour lequel chaque occurrence d'un même item a été numérotée. Un item $a \in \mathbb{E}$ apparaissant n fois dans une séquence est encodé par n items : $I_1^a, ..., I_n^a$. Un itemset fréquent de taille m, $(I_{i_k}^{e_k})_{1 \leq k \leq m}$, extrait à partir de ce jeu de données est transformé en le multiset contenant, pour tout k, i_k occurrences de l'évènement e_k. Tous les itemsets fréquents contenant plusieurs occurrences d'un même item (*i.e.* $\exists i, j \in [1, m],\ i \neq j$, tq $e_i = e_j$) avec une cardinalité différente ont été préalablement supprimés car redondants.

Dans un second temps, les lignes 3 à 11 extraient les contraintes temporelles de chaque multiset. L'approche naïve dans laquelle les contraintes temporelles discriminantes sont extraites pour chaque multiset fréquent a l'inconvénient de générer un grand nombre de chroniques. L'information de la discriminance du multiset est considérée plus généralisatrice, et seule cette chronique est conservée. Aucune contrainte temporelle n'est spécifiée pour ces multisets discriminants. Pour cela, la ligne 4 teste si le multiset ms est discriminant. Si tel est le cas, ms est ajouté à l'ensemble des motifs discriminants. Et seulement dans le cas contraire, les lignes 7 à 9 engendrent des chroniques à partir des contraintes temporelles discriminantes identifiées par EXTRACTDISCRCONSTRAINTS.

4.1 Extraction de contraintes temporelles

L'idée générale de EXTRACTDISCRCONSTRAINTS est de ramener l'extraction des intervalles décrivant les contraintes temporelles à une tâche classique d'apprentissage relationnel.

Pour chaque extraction on construit un jeu de données relationnelles tel que ses attributs sont les paires d'évènements du multiset et tel que ses exemples sont les occurrences du multiset. Les valeurs d'un attribut pour un exemple sont les durées inter-évènements de la paire d'évènements au sein d'une occurrence. Un exemple est étiqueté par le SID séquence. Cette étiquette est conservée afin de faire correspondre la définition de support d'une chronique avec la comparaison de ces occurrences.

SID	A→B	B→C	A→C	Label
1	2	2	4	+
1	−1	2	1	+
2	5	−2	3	+
3	3	0	3	+
5	−1	3	1	-
6	6	−1	5	-

TAB. 2 – *Jeu de données associé au multiset* $\{A, B, C\}$.

Exemple 5. Le tableau 2 correspond au jeu de données obtenu à partir des occurrences du multiset $\{\!\{A, B, C\}\!\}$ dans le tableau 1. L'attribut « **A→B** » désigne les durées entre **A** et **B**. On remarque sur cet exemple que plusieurs exemples peuvent provenir de la même séquence.

Ce jeu de données relationnelles est traitable par un algorithme d'apprentissage relationnel dont les résultats seront des conjonctions de règles de la forme $e_1 \rightarrow e_2 \geq x$ ou $e_1 \rightarrow e_2 \leq y$ où e_1, e_2 sont des évènements et $(x, y) \in \mathbb{R}$. Ces règles sont alors traduites comme des contraintes temporelles, $e_1[x, y]e_2$.

Exemple 6. La conjonction de règles $A \rightarrow B \leq 5 \wedge B \rightarrow C \leq 2 \implies +$ caractérisant parfaitement les exemples étiquetés par $+$ dans le tableau 2 est traduite par l'ensemble de contraintes temporelles $\mathcal{T} = \{A[-\infty, 5]B, B[-\infty, 2]C\}$ ce qui donne la chronique discriminante $\mathcal{C} = (\mathcal{E} = \{\!\{e_1 = A, e_2 = B, e_3 = C\}\!\}, \mathcal{T} = \{e_1[-\infty, 5]e_2, e_2[-\infty, 2]e_3\})$.

Apprentissage de règles ($Ripper_k$) L'apprentissage de règles est effectué en pratique par l'algorithme $Ripper_k$ (Cohen, 1995). Cet algorithme a été choisi parmi l'état de l'art des algorithmes d'apprentissage de règles relationnelles, d'une part, parce qu'il est l'un des plus performants et, d'autre part, parce qu'il permet l'extraction de règles non-ordonnées. Le problème des extracteurs de règles ordonnées (*e.g.* CN2, C4.5) est qu'ils extraient une liste de conjonctions de règles dont la discriminance de la règle ordonnée à la position n n'est valable qu'en dehors des cas décrits par $n-1$ règles précédentes. La discriminance d'une chronique obtenue à partir de l'une de ces règles ne serait valide que pour un sous-ensemble des séquences du jeu de données. Ceci ne correspond pas à notre définition de discriminance.

Pour une classe à apprendre, $Ripper_k$ sépare le jeu de données en deux : $Growth$ et $Prune$. Le premier permet de construire la conjonction de règles discriminant les exemples d'une classe aux autres. La construction s'arrête lorsqu'il n'est plus possible d'ajouter une règle à la conjonction qui améliore la discriminance. Le second est utilisé pour élaguer la règle construite. Si la conjonction est plus discriminante sans sa dernière règle, on lui retire et on réessaie jusqu'à ce qu'elle ne puisse plus être améliorée. Si la discriminance n'est pas satisfaite pour cette conjonction, la recherche s'arrête pour cette étiquette. Sinon la conjonction est retournée, les exemples associés à celle-ci sont retirés de $Growth$ et la recherche recommence. Les étapes précédentes sont répétées pour chaque étiquette présente dans le jeu de données.

Le choix d'utiliser un algorithme d'apprentissage basé sur une heuristique incomplète, en l'occurrence une heuristique basée sur le principe MDL (*Minimum Description Length*), se montre ici indispensable pour des raisons calculatoires. Néanmoins, $Ripper_k$ combine (1) une complexité algorithmique raisonnable – les temps de calculs restent donc raisonnables, (2) des performances en classification intéressantes – les chroniques extraites sont donc bien

représentatives du jeu de données – et (3) des ensembles de règles réduits – les chroniques extraites restent facilement interprétables (Lattner et al.).

Post-traitement des chroniques La mesure de taux de croissance utilisée par notre approche n'est pas directement intégrée dans $Ripper_k$. Pour assurer que les chroniques extraites soient correctes, il est alors nécessaire de faire un post-traitement.

La définition du paramètre de coût des faux positifs de $Ripper_k$ comme égal à g_{min} permet de retrouver une contrainte similaire à notre contrainte de discriminance. Néanmoins, cette approche ne prend pas en compte les motifs validant cette contrainte sur la totalité du jeu de données mais indépendamment sur $Growth$ et $Prune$. Des motifs pourraient être oubliés.

La solution choisie post-traite l'ensemble des conjonctions retournées par $Ripper_k$ et ne conserve que celles validant notre contrainte de discriminance. Cette solution a l'avantage de fonctionner dans le cas où les règles sont déjà post-traitées pour vérifier que le nombre d'exemples n'a pas faussé la mesure de support.

Limite des instances multiples Le problème des instances multiples (Foulds et Frank, 2010) désigne les problèmes de comptage rencontrés lorsqu'un motif apparaît plusieurs fois dans un objet. Dans notre cas, un objet est une séquence. Cette situation est rencontrée lors de la constitution du jeu de données (*cf.* table 2) et fausse le comptage du nombre d'objets validant une conjonction de règles fait par $Ripper_k$. Lorsque les lignes 1 et 2 de la table 2 sont couvertes par une règle, elle ne doivent ici compter que pour un objet (*i.e.* un SID). Ce cas d'utilisation n'est pas prévu par $Ripper_k$.

Cette fois-ci, la solution consistant à post-traiter les conjonctions de règles de $Ripper_k$ et à ne conserver que celles qui valident effectivement les contraintes de fréquence minimale et de discriminance, n'assure pas que des chroniques discriminantes soient extraites.

Une seconde solution à ce problème serait de modifier le système de comptage de $Ripper_k$ afin qu'il évalue une conjonction de règles en comptant les SID distincts. Mais l'heuristique de $Ripper_k$ n'est pas faite pour écarter des exemples après en avoir sélectionné d'autres et donc l'efficacité serait faible dans le cas de jeux de données contenant de nombreux exemples portant les mêmes SID.

La troisième solution serait d'utiliser des méthodes adaptées à ce type de problème. Par exemple, Doran et Ray (2014) sélectionnent un exemple témoin pour chaque objet afin de se ramener à une tâche d'apprentissage de règles classique. Le jeu de données fourni à $Ripper_k$ peut être prétraité suivant ces méthodes afin de ne lui transmette qu'un seul exemple par séquence. Cette solution paraît être la plus fiable mais ajoute une complexité importante au processus d'extraction de contraintes temporelles.

La prise en compte des instances multiples se ferait donc nécessairement au prix d'un ajout de complexité calculatoire. Les instances multiples étant rarement rencontrées dans nos applications, nous avons privilégié l'efficacité calculatoire en ne mettant pas en place de solution spécifique à ce problème.

5 Expériences et résultats

L'implémentation de DCM, écrite en $C++$, repose sur les implémentations préexistantes de LCM (Uno et al., 2004) et de $Ripper_k$ (Cohen, 1995).

5.1 Données synthétiques

Les bases de données sont générées par un simulateur qui introduit des chroniques fréquentes aux séquences. Le simulateur gère des bases contenant deux classes $\mathbb{L} = \{\mathcal{S}^+, \mathcal{S}^-\}$. Les expérimentations sur données synthétiques permettent de valider l'extraction des chroniques discriminantes au sein d'un jeu de données.

Le principe général du simulateur est de générer des séquences basées sur deux chroniques $\mathcal{C}^+$ et $\mathcal{C}^-$ puis de les bruiter aléatoirement. Afin de générer un jeu de données dont il est possible de discriminer les classes positives et négatives, $\mathcal{C}^+$ et $\mathcal{C}^-$ sont majoritairement introduites dans deux ensembles respectivement $\mathcal{S}^+$ et $\mathcal{S}^-$. Deux paramètres gèrent l'introduction de ces chroniques : $f^{\mathcal{D}}_{min}$ et $g^{\mathcal{D}}_{min}$. $f^{\mathcal{D}}_{min}$ définit le pourcentage de séquences de la classe majoritaire dans lesquelles une chronique apparaîtra. $g^{\mathcal{D}}_{min}$ définit le rapport entre les nombres de séquences des deux classes dans lesquelles apparaît cette chronique. On ne cherche pas à extraire $\mathcal{C}^+$ et $\mathcal{C}^-$, on cherche à extraire les motifs décrivant le plus d'occurrences de $\mathcal{C}^+$ et le moins d'occurrences de $\mathcal{C}^-$.

Le simulateur est paramétré pour générer par défaut des jeux de données contenant 800 séquences de chaque classe de longueur moyenne 10. La taille du vocabulaire est fixée à 50 items et $f^{\mathcal{D}}_{min}$ et $g^{\mathcal{D}}_{min}$ sont fixés respectivement à 80% et 10%.

Les jeux de données $BaseN$ ont été créés à partir de chroniques basées sur le même multiset $\{\!\{A, B\}\!\}$ de taille 2. Elles comportent donc chacune un unique intervalle temporel. Les deux intervalles se chevauchent plus ou moins. Ainsi, pour tous ces jeux $\mathcal{C}^+ = (\{\!\{A, B\}\!\}, A[3, 10]B)$ et $\mathcal{C}^- = (\{\!\{A, B\}\!\}, A[7, 8]B)$ puis les contraintes $A[6, 8]B$, $A[4, 8]B$, $A[4, 9]B$ et $A[4, 10]B$ ont été utilisées pour générer respectivement $Base1$, $Base2$, $Base3$, $Base4$ et $Base5$.

Pour chaque jeu de données, les résultats sont moyennés sur 20 exemplaires de bases. Les résultats présentés regrouperont donc jusqu'à 40 motifs du fait de ces 20 générations. La capacité de l'extracteur à extraire les bons motifs peut être évaluée en comparant les ensembles des motifs extraits et des motifs recherchés puisque les motifs discriminants recherchés sont connus à l'introduction des chroniques. Deux mesures ont pour cela été utilisées : $\Delta c_{(m,m^t)}$ et $\Delta g_{(m,m^t)}$. $\Delta c_{(m,m^t)}$, le rapport de couverture, représente le rapport entre le nombre de séquences de la classe $\mathcal{S}^+$ contenant une occurrence partagée par m et m^t et le nombre de celles contenant m^t. $\Delta g_{(m,m^t)}$, le rapport de croissance, représente le rapport entre les taux de croissance de m et de m^t. Si $support(m, \mathcal{S}^-) = support(m^t, \mathcal{S}^-) = 0$ alors $\Delta g_{(m,m^t)} = 1$ sinon si $support(m^t, \mathcal{S}^-) = 0$ alors $\Delta g_{(m,m^t)} = 0$. Pour chacun des motifs recherchés m^t, un seul couple $(\Delta c_{(m,m^t)}, \Delta g_{(m,m^t)})$ est retenu tel que $\Delta c_{(m,m^t)}$ soit le plus élevé pour tout motif extrait m et que $\Delta g_{(m,m^t)}$ soit le plus élevé s'il existe plusieurs $\Delta c_{(m,m^t)}$ maximaux.

Le tableau 3 présente les résultats d'extraction de DCM sur les 20 jeux de données générés pour chaque $BaseN$. Le rapport de couverture est donné en ligne et celui de croissance en colonne. Chaque cellule du tableau correspond au nombre de motifs extraits pour un rapport de couverture, un rapport de croissance et un type d'expérience donnés.

On remarque sur le tableau 3 que DCM extrait pour au moins 75% des motifs discriminants introduits dans chaque jeu de données un motif discriminant dont les rapport de couverture et de croissance sont parfaits ($= 1$). De plus, il n'y a que pour les jeux de données $Base4$ et $Base5$ que notre extracteur n'extrait aucun motif pour certains motifs discriminants introduits, 1 sur 40 pour $Base4$ et 2 sur 20 pour $Base5$.

$\mathcal{C}^-$	$Base1$	$Base2$	$Base3$				$Base4$					$Base5$			
$\Delta c \backslash \Delta g$	**1**	**1**	**1**	**0.96**	**0.89**	**0.86**	**1**	**0.96**	**0.93**	**0.89**	**0**	**1**	**0.93**	**0.91**	**0**
1	31	40	30	2	5	3	30	3	3	1	0	15	1	1	0
0.9	6														
0.85	1														
0.72							1								
0.66	2														
0.5												1			
0.47							1								
0											1				2

TAB. 3 – *Nombre de motifs extraits par couple* $(\Delta c, \Delta g)$ *pour chaque motifs discriminants introduits. L'absence de chiffre correspond à une valeur 0.*

Afin d'expliquer ces erreurs, l'extraction a été relancée avec un taux de croissance minimal de 1.1. Avec cette configuration, tous les motifs de $Base4$ sont extraits (avec $\Delta c = 1$ et $\Delta g = 0.83$). Il est à noter que le motif correspondant à $\Delta c = 0.47$ pour la première extraction des motifs de $Base4$ obtient les mêmes rapports Δc et Δg.

Ces résultats sur des données synthétiques simples montrent que notre extracteur extrait effectivement les motifs discriminants avec une certaine robustesse.

5.2 Expérimentations sur données réelles

Deux types de jeux de données réelles ont été utilisés : nous utilisons tout d'abord les jeux de données utilisés pour évaluer $BIDE\text{-}D$ (Fradkin et Mörchen, 2015) afin de nous comparer à cette approche. Nous illustrons ensuite la qualité des chroniques extraites à partir de jeux de données ECG (*i.e.* électrocardiogrammes).

Comparaison avec $BIDE$**-**D Les premières données réelles utilisées pour tester notre extracteur sont les données proposées pour tester les algorithmes de $BIDE\text{-}D$. Ces données proviennent d'applications variées. Afin de se comparer avec $BIDE\text{-}D$ sur un sous-ensemble des jeux de données utilisés, ni trop simples comme $blocks$ où les taux de prédiction approchent les 100% ni trop difficiles comme $Auslan2$ où les approches de $BIDE\text{-}D$ dépassent difficilement les 30%, nous avons choisi de nous concentrer sur $asl\text{-}bu$, $asl\text{-}gt$ et $context$. L'évaluation des résultats se fera en comparaison avec ceux présentés pour $BIDE\text{-}D$ (*cf.* annexe à (Fradkin et Mörchen, 2015)).

Les résultats suivants comparent les performances en classification des chroniques discriminantes avec les séquences discriminantes extraites par $BIDE\text{-}D$. Les paires $\langle \mathcal{C}, L \rangle$ de chroniques discriminantes, $\mathcal{C}$ et d'étiquettes, L, ont été utilisées pour prédire l'étiquette d'une séquence. On prédira donc une étiquette, L, pour une séquence dans le cas où cette séquence contiendrait la chronique $\mathcal{C}$. Dans le cas de plusieurs chroniques apparaissant dans la séquence on retiendra l'étiquette associée à la chronique ayant le taux de croissance le plus élevé. Cette méthode naïve a été jugée plus juste que l'utilisation d'un classifieur pour évaluer les chroniques extraites.

Les résultats commentés ci-dessous sont présentés dans le tableau 4. Une contrainte de taille maximale des motifs a été fixée à 5 pour faciliter leur extraction et limiter leur nombre.

Sur $asl\text{-}bu$ les résultats sont quelque peu meilleurs que ceux de $BIDE\text{-}D$. On remarque, que ce soit pour $\sigma_{min} = 0.3$ ou $\sigma_{min} = 0.6$, que le taux de prédiction pour $g_{min} = 2$ est équivalent alors que le nombre de motifs extraits est réduit de plus de 30000 à 1600.

	$asl\text{-}bu$				$asl\text{-}gt$				$context$				
$\sigma_{min} \backslash g_{min}$	**2**	**3**	**4**	**5**	**2**	**3**	**4**	**5**	**2**	**3**	**4**	**5**	**6**
0.2					0.09	0.07	0.07	0.06	0.06	0.58	0.57	0.59	0.55
0.3	0.66	0.65	0.63		0.08	0.06	0.05	0.04	0.58	0.56	0.56	0.52	0.52
0.4	0.66	0.66	0.65		0.04	0.04	0.03	0.02	0.58	0.54	0.49	0.48	0.45
0.5	0.66	0.64	0.58	0.50	0.03	0.03	0.02	0.02	0.58	0.55	0.52	0.48	0.43
0.6	0.65	0.55	0.51	0.37	0.03	0.02	0.02	0.01	0.55	0.56	0.49	0.42	0.43

TAB. 4 – *Évolution de la précision en fonction de σ_{min}, de g_{min} et du jeu de données.*

Sur $asl\text{-}gt$ les résultats sont malheureusement très mauvais. Là où les taux de prédictions vont de 0.27 pour $\sigma_{min} = 0.6$ à 0.83 pour $\sigma_{min} = 0.2$ pour $BIDE\text{-}D$, ceux-ci n'atteignent pas 0.1 pour les chroniques discriminantes. De nombreux motifs sont extraits mais sont mal répartis entre les étiquettes du jeu de données. Ces résultats proviennent peut-être de la contrainte de taille maximale des chroniques ou montrent la limite de l'utilisation de $Ripper_k$ dans un contexte d'instances multiples.

Finalement pour $context$, les résultats sont similaires à ceux de $asl\text{-}bu$ pour notre approche. Ce sont donc nos meilleurs résultats puisque les résultats présentés par $BIDE\text{-}D$ oscillent entre 0.26 et 0.53. On remarque que les faibles résultats de $BIDE\text{-}D$ sont sûrement liés à l'utilisation de seuils de support minimaux différents pour ce jeu de données. L'écart du nombre de motifs extraits par les paramètres $\sigma_{min} = 0.2$ ou $\sigma_{min} = 0.6$ est beaucoup moins important que pour $asl\text{-}bu$. Il va de 360 motifs pour $\sigma_{min} = 0.2$ et $g_{min} = 2$ à 145 motifs pour $\sigma_{min} = 0.6$ et $g_{min} = 4$.

Analyse d'électrocardiogrammes Ces mêmes expériences ont été réalisées sur les données ECG, des données d'électrocardiogramme contenant majoritairement des cycles cardiaques problématiques. Chaque jeu de données concerne le patient et ses cycles cardiaques. Le jeu de données présenté dans les résultats est celui concernant le patient 214 dont les cycles cardiaques sont étiquetés « bloc de branches » et « extrasystole », deux problèmes cardiaques. Ces données ont déjà été utilisées pour extraire des chroniques discriminantes par Carrault et al. (2003). Ces données ont été prétraitées afin que chaque séquence corresponde à un cycle cardiaque. L'intérêt de ces données est la très faible diversité d'événements, et l'équivalence de la plupart des séquences, quelle que soit leur classe, en terme d'items. Elles ne contiennent que quatre types d'évènements, les ondes cardiaques p et qrs annotées des mentions $normal$ ou $abnormal$. En effet, aucun multiset d'items ne peut être discriminant dans de telles conditions et les évènements cardiaques se déroulant toujours dans le même ordre, un motif purement séquentiel ne pourrait pas non plus être discriminant. L'aspect temporel devient donc le seul moyen d'établir une discriminance [2]. Ceci met en évidente l'utilité des chroniques discriminantes face à d'autres types de motifs moins riches.

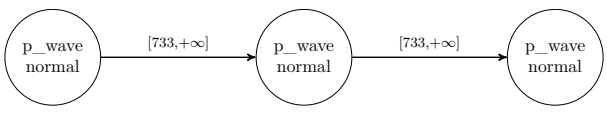

FIG. 2 – *Chronique discriminant les blocs de branche des extrasystoles du patient* 214.

2. Ces données correspondent à un cas réel des jeux de données synthétiques de la section 5.1.

Pour des jeux de données ECG rééquilibrés le taux de prédiction est toujours supérieur à 80% pour un taux de croissance minimal fixé à 2. Le bloc de branche est un problème cardiaque qui rend les cycles anormalement longs et l'extrasystole correspond à une contraction prématurée du cœur. On remarque que la chronique discriminante de la Figure 2 représente un cycle cardiaque anormalement long dont les bornes inférieures (733) correspondent finalement à des cycles normaux. Cette chronique est un exemple de l'intérêt de l'information temporelle pour la discriminance. En effet, les trois évènements sont les mêmes, leur ordre n'a aucune importance.

6 Conclusion et perspectives

L'extraction de chroniques était déjà connue mais il n'existait pas de travaux concernant l'extraction de chroniques discriminantes. L'algorithme DCM a été proposé pour résoudre ce problème. Il ajoute, d'une part, la prise en compte d'une information temporelle riche par rapport aux algorithmes d'extraction de motifs séquentiels discriminants et il ajoute, d'autre part, la contrainte de discriminance à l'extraction de chroniques. Cet algorithme se base sur les travaux antérieurs d'apprentissage de règles relationnelles. Les expérimentations réalisées utilisant l'algorithme $Ripper_k$ ont montré que DCM était capable d'extraire efficacement des chroniques discriminantes. Les comparaisons en terme de pouvoir de prédiction entre DCM et les algorithmes de $BIDE$-D (Fradkin et Mörchen, 2015) ont montré que les chroniques discriminantes extraites par DCM étaient capables de concurrencer les méthodes de l'état de l'art, et ce, sans avoir à entraîner de classifieur.

Toutefois, pour certains jeux de données, DCM n'extrait pas de chroniques suffisamment discriminantes pour prédire au moins aussi bien que les algorithmes de $BIDE$-D. Une perspective d'amélioration serait de traiter le problème des extractions des contraintes temporelles avec une approche gérant les instances multiples. L'ensemble des motifs extraits reste important. L'extraction de représentations condensées de chroniques, *e.g.* des chroniques minimales, ou un post-traitement permettant d'identifier des chroniques similaires permettraient une meilleure lecture des résultats. Finalement, les extractions de contraintes temporelles étant indépendantes les unes des autres, la performance computationnelle de DCM pourrait être améliorées au travers du parallélisme.

?

Remerciements Ce travail a été financé par l'ANSM dans le cadre de la plate-forme PEPS.

Références

Carrault, G., M.-O. Cordier, R. Quiniou, et F. Wang (2003). Temporal abstraction and inductive logic programming for arrhythmia recognition from electrocardiograms. *Artificial intelligence in medicine 28*(3), 231–263.

Cohen, W. W. (1995). Fast effective rule induction. In *Machine Learning, Proceedings of the International Conference on Machine Learning*, pp. 115–123.

Cram, D., B. Mathern, et A. Mille (2012). A complete chronicle discovery approach : application to activity analysis. *Expert Systems 29*(4), 321–346.

Dong, G. et J. Li (1999). Efficient mining of emerging patterns : Discovering trends and differences. In *Proceedings of ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 43–52.

Doran, G. et S. Ray (2014). A theoretical and empirical analysis of support vector machine methods for multiple-instance classification. *Machine Learning 97*(1), 79–102.

Dousson, C. et T. V. Duong (1999). Discovering chronicles with numerical time constraints from alarm logs for monitoring dynamic systems. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 620–626.

Fabrègue, M., A. Braud, S. Bringay, C. Grac, F. Le Ber, D. Levet, et M. Teisseire (2014). Discriminant temporal patterns for linking physico-chemistry and biology in hydro-ecosystem assessment. *Ecological Informatics 24*, 210–221.

Foulds, J. et E. Frank (2010). A review of multi-instance learning assumptions. *The Knowledge Engineering Review 25*(01), 1–25.

Fradkin, D. et F. Mörchen (2015). Mining sequential patterns for classification. *Knowl. Inf. Syst. 45*(3), 731–749.

Guyet, T. et R. Quiniou (2011). Extracting temporal patterns from interval-based sequences. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 1306–1311.

Huang, Z., X. Lu, et H. Duan (2012). On mining clinical pathway patterns from medical behaviors. *Artificial Intelligence in Medicine 56*(1), 35–50.

Lattner, A. D., S. Kim, G. Cervone, et J. J. Grefenstette. Experimental comparison of symbolic learning programs for the classification of gene network topology models. *Center for Computing Technologies–TZI 2*, 1.

Mannila, H., H. Toivonen, et A. I. Verkamo (1997). Discovery of frequent episodes in event sequences. *Data mining and knowledge discovery 1*(3), 259–289.

Masseglia, F., M. Teisseire, et P. Poncelet (2004). Recherche des motifs séquentiels. *Revue Ingénierie des Systemes d'Information (ISI), numéro spécial "Extraction de motifs dans les bases de données 9*(3-4), 183–210.

Mooney, C. H. et J. F. Roddick (2013). Sequential pattern mining – approaches and algorithms. *ACM Journal of Computing Survey 45*(2), 1–39.

Uno, T., M. Kiyomi, et H. Arimura (2004). LCM ver. 2 : Efficient mining algorithms for frequent/closed/maximal itemsets. In *FIMI*, Volume 126.

Summary

Sequential pattern mining attempts to extract frequent behaviours from sequential dataset. When sequences are labeled, it is interesting to extract characteristic behaviors for each sequence class. This task is called discriminant pattern mining. In this paper, we introduce discriminant chronicle mining. Conceptually, a chronicle is a graph whose vertices are events and edges represent quantitative time constraints between events. We also propose DCM, an algorithm dedicated to mining of discriminant chronicles. It is based on rule learning methods to extract the temporal constraints. Computational performances and discriminant power of extracted chronicles are evaluated on artificial and real data.