Motifs Séquentiels δ-Libres

Marc Plantevit*, Chedy Raïssi**, Bruno Crémilleux***

* Université de Lyon, CNRS, INRIA
Université Lyon 1, LIRIS Combining, UMR5205, F-69622, France
Marc.Plantevit@liris.cnrs.fr,
**INRIA Nancy Grand-Est
Chedy.Raissi@loria.fr
*** Université de Caen Basse-Normandie
GREYC, UMR6072, F-14032, France
Bruno.Cremilleux@info.unicaen.fr

Résumé. Bien que largement étudiée, l'extraction de motifs séquentiels reste une tâche très difficile et pose aussi le défi du grand nombre de motifs produits. Dans cet article, nous proposons une nouvelle approche extrayant les motifs séquentiels les plus généraux à fréquence similaire. Nous montrons en quoi l'extension de cette notion, déjà connue pour les motifs ensemblistes, est un problème particulièrement difficile pour les séquences. Les motifs δ-libres ainsi produits sont en nombre réduit et facilitent les usages d'un processus de fouille et nous montrons leur apport comme descripteurs dans un contexte de classification de séquences.

1 Introduction

La fouille de données temporelles est une problématique rencontrant un vif succès notamment parce qu'elle est motivée par de nombreuses applications telles que l'analyse de données clients ou financières, le web usage mining ou encore l'analyse de séquences biologiques. Pour un aperçu des méthodes de fouilles de séries temporelles ou de flots de données, voir Dong et Pei (2007). Les données représentées par des séquences d'événements discrets sont un cas particulier très étudié de données temporelles. Une tâche importante dans de telles données est de découvrir les régularités présentes en extrayant des *motifs séquentiels*, tâche introduite par Agrawal et Srikant (1995). Un motif séquentiel est un motif local représentant une séquence d'itemsets régulièrement observée dans les données. Au-delà de leur intérêt intrinsèque, ceux-ci sont aussi exploités à des fins de clustering ou de classification.

Il est bien connu que l'extraction de motifs pose le problème de l'abondance des résultats produits : l'utilisateur a peu de guide pour découvrir les motifs qui lui seront utiles parmi la masse de motifs proposés. Cela a conduit la communauté à mener de nombreux travaux pour offrir une vision plus synthétique des motifs extraits (représentations condensées Mannila et Toivonen (1996), compression des données van Leeuwen et al. (2009)) ou se focaliser sur les plus pertinents suivant des critères a priori (telles que l'extraction sous contraintes de motifs locaux ou d'ensembles de motifs, voir Bonchi et Lucchese (2007)).

Ces différentes approches ont largement été explorées dans le cas des motifs ensemblistes (itemsets) et, par exemple, de nombreuses représentations condensées ont été proposées afin d'éviter d'extraire des motifs redondants (voir Calders et al. (2004) pour un état de l'art). Ces méthodes mettent en place des techniques algorithmiques sophistiquées pour faire face au grand espace de recherche à explorer. Dans le cas des séquences, la combinatoire est encore plus forte et la tâche est beaucoup plus ardue. Il n'est donc pas surprenant qu'on ne dispose encore que de peu de résultats pour ce type de données. Ainsi, pour les séquences, les motifs séquentiels fermés sont la représentation condensée la plus répandue (Yan et al. (2003)) et, ce n'est que très récemment que les motifs séquentiels libres ont été proposés par Gao et al. (2008) et Lo et al. (2008). Tout comme pour les itemsets, ces deux représentations restent particulièrement sensibles à la présence de bruit dans les données qui fragmentent les motifs produits.

Dans cet article, nous proposons un nouveau type de motifs séquentiels, les motifs séquentiels δ-libres. Ceux-ci, à fréquence similaire, favorisent les motifs les plus généraux. Ces motifs présentent le double avantage d'être moins nombreux que les motifs libres et d'être plus adaptés aux données bruitées. Même si la notion de motifs δ-libres était déjà connue dans le cas des itemsets, l'extraction des motifs séquentiels δ-libres pose un véritable défi. En effet, nous montrons que, contrairement au cas des itemsets, la contrainte de δ-liberté n'est pas anti-monotone dans le cas des séquences. L'absence de cette propriété rend la conception d'un algorithme efficace beaucoup plus difficile. Nous montrons alors comment exhiber des propriétés intéressantes d'élagage des motifs non-pertinents qui nous permettent de définir un algorithme d'extraction efficace correct et complet. D'autre part, les motifs séquentiels δ-libres ainsi obtenus facilitent les usages d'un processus de fouille et nous montrons leur apport comme descripteurs dans un contexte de classification de séquences.

Le reste de l'article s'organise de la façon suivante. Après avoir défini les motifs séquentiels δ-libres à la section 2, nous proposons à la section 3 un algorithme permettant leur extraction. La section 4 présente des expériences menées sur des données réelles montrant l'efficacité de l'algorithme et l'utilité des motifs extraits dans un contexte de classification de séquences. Enfin, la section 5 situe notre proposition par rapport à l'existant.

2 Motifs séquentiels et motifs minimaux

2.1 Préliminaires

Soit $\mathcal{I} = \{i_1, i_2 \dots i_m\}$ un ensemble fini de littéraux appelés items. Un *itemset* est un ensemble non vide d'items. Une *séquence* S sur $\mathcal{I}$ est une liste ordonnée d'itemsets $\langle it_1, \dots, it_l \rangle$ où les it_j sont des itemsets de $\mathcal{I}$ et $j = 1 \dots l$. Une séquence est aussi appelée *motif séquentiel*. Une *k-séquence* est une séquence de k items (i.e., de longueur k) contenus dans au plus k itemsets. On note $|S|$ la longueur d'une séquence S et $S[0, l]$ la l-séquence identifiée comme préfixe de S. L'ensemble infini de tous les séquences possibles sur $\mathcal{I}$ est noté $\mathbb{T}(\mathcal{I})$. Soit $s = \langle e_1, e_2, \dots, e_n \rangle$ une séquence, on note $s^{(i)} = \langle e_1, \dots, e_{i-1}, e_{i+1}, \dots, e_n \rangle$, la séquence s où le $i^{\text{ème}}$ élément est supprimé. L'opérateur '$\cdot$' représente l'opération de concaténation d'itemsets entre deux séquences et '$-\cdot$' représente la concaténation d'items entre deux séquences. Par exemple, $\langle (a)(b) \rangle \cdot \langle (b)(a) \rangle = \langle (a)(b)(b)(a) \rangle$ et $\langle (a)(b) \rangle - \cdot \langle (c)(a) \rangle = \langle (a)(b, c)(a) \rangle$.

Définition 1 (*Inclusion*)**.** *Une séquence $S' = \langle is'_1\ is'_2\ \ldots\ is'_n\rangle$ est une sous-séquence d'une autre séquence $S = \langle is_1\ is_2\ \ldots\ is_m\rangle$, noté $S' \preceq S$, s'il existe des entiers $i_1 < i_2 < \ldots < i_j < \ldots < i_n$ tels que $is'_1 \subseteq is_{i_1}$, $is'_2 \subseteq is_{i_2} \ldots is'_n \subseteq is_{i_n}$.*

Une base de séquences $\mathcal{D}$ est une collection de paires (SID, T) où SID est un identifiant et T une séquence de $\mathbb{T}(\mathcal{I})$.

Définition 2 (*Support, fréquence et fréquents*)**.** *Le support d'une séquence S dans une base de séquences $\mathcal{D}$, noté $Support(S, \mathcal{D})$, est défini de la façon suivante : $Support(S, \mathcal{D}) = |\{(SID, T) \in \mathcal{D} | S \preceq T\}|$.*

La fréquence de S dans $\mathcal{D}$, notée $freq^{\mathcal{D}}_S$, est $freq^{\mathcal{D}}_S = \frac{Support(S,\mathcal{D})}{|\mathcal{D}|}$.

Étant donné un seuil minimum de fréquence σ, le problème de l'extraction de motifs séquentiels fréquents est d'extraire l'ensemble complet des séquences S dans $\mathcal{D}$ telles que $freq^{\mathcal{D}}_S \geq \sigma$. L'ensemble complet des séquences fréquentes pour un seuil σ dans une base $\mathcal{D}$ est noté $FSeqs(\mathcal{D}, \sigma)$ [1]*,*

$$FSeqs(\mathcal{D}, \sigma) = \{S \mid freq^{\mathcal{D}}_S \geq \sigma\}$$

S_1	$\langle(a)(b)(c)(d)(a)(b)(c)\rangle$
S_2	$\langle(a)(b)(c)(b)(c)(d)(a)(b)(c)(d)\rangle$
S_3	$\langle(a)(b)(b)(c)(d)(b)(c)(c)(d)(b)(c)(d)\rangle$
S_4	$\langle(b)(a)(c)(b)(c)(b)(b)(c)(d)\rangle$
S_5	$\langle(a)(c)(d)(c)(b)(c)(a)\rangle$
S_6	$\langle(a)(c)(d)(a)(b)(c)(a)(b)(c)\rangle$
S_7	$\langle(a)(c)(c)(a)(c)(b)(b)(a)(e)(d)\rangle$
S_8	$\langle(a)(c)(d)(b)(c)(b)(a)(b)(c)\rangle$

TAB. 1: Base de séquences $\mathcal{D}_{ex}$

Exemple 1 (Exemple « fil conducteur »)**.** *La base de séquences $\mathcal{D}_{ex}$ (cf. table 1) va servir tout au long de cet article comme base « jouet » pour illustrer les différentes notions. $\mathcal{D}_{ex}$ contient 8 séquences de données avec $\mathcal{I} = \{a, b, c, d, e\}$. La séquence $\langle(a)(b)(a)\rangle$ est incluse dans S_1. On dit aussi que S_1 supporte $\langle(a)(b)(a)\rangle$. Remarquons que S_5 ne supporte pas $\langle(b)(d)\rangle$ ($\langle(b)(d)\rangle \not\preceq S_5$).*

Pour faciliter la présentation, les exemples de cet article portent sur des séquences d'items. Cependant, tous les théorèmes et propositions sont énoncés et vérifiés dans le cadre général des séquences d'itemsets.

Nous rappelons maintenant la notion de base projetée, notion qui nous est nécessaire pour l'obtention de nos résultats.

Définition 3 (Base projetée Pei et al. (2004))**.** *Soit s_p, un motif séquentiel dans une base de séquences $\mathcal{D}$. La base projetée par rapport à s_p, notée $\mathcal{D}_{|s_p}$, est l'ensemble des suffixes des séquences de $\mathcal{D}$ par rapport au préfixe s_p.*

Notons que le préfixe de s_p dans une séquence de données S est égal à la sous-séquence de S commençant au début de S et finissant strictement après la première *occurrence minimale* (voir Mannila et al. (1997)) de s_p dans S.

1. Dans le cas où σ est un entier, $freq^{\mathcal{D}}_S$ est défini par rapport à $Support(S, \mathcal{D})$. Dans la suite, σ est entier si non spécifié.

2.2 Motifs séquentiels δ-libres

La notion de motifs minimaux par rapport à une contrainte donnée a été intensivement étudiée pour les motifs ensemblistes (voir section 5). Les motifs libres (aussi appelés *générateurs* ou *clés*) sont les motifs minimaux à fréquence égale. Une définition d'un tel motif est que X est libre si et seulement si il n'existe pas de règle exacte (i.e., confiance = 100%) entre n'importe quelle paire de ses sous-ensembles. Cette notion a ensuite été généralisée pour donner lieu aux motifs δ-libres et aux représentations condensées approximatives fondées sur ces motifs Boulicaut et al. (2003). Cette généralisation permet d'obtenir des représentations condensées plus concises mais la fréquence d'un motif y est connue avec une incertitude contrôlée. D'autre part, un attrait majeur des motifs δ-libres est leur capacité à autoriser quelques exceptions parmi les exemples supportant ces motifs. Cette caractéristique rend ces motifs particulièrement bien adaptés pour traiter des jeux de données réels qui, de par leur nature, contiennent des éléments s'écartant des règles générales et/ou des erreurs.

Nous étendons cette définition aux séquences et nous définissons les motifs séquentiels δ-libres similairement aux motifs ensemblistes δ-libres. Les séquences δ-libres sont des motifs *minimaux*. À fréquence similaire, elles favorisent les séquences les plus générales par rapport aux séquences les plus spécifiques.

Définition 4 (Séquence δ-libre)**.** *Étant donné une base de séquences $\mathcal{D}$, une séquence s est δ-libre si :*

$$\forall s' \prec s,\ Support(s', \mathcal{D}) > Support(s, \mathcal{D}) + \delta$$

À partir de l'exemple de la table 1, la figure 1 représente toutes les séquences 1-libres (**en gras**) ayant un support supérieur ou égal à 3. Par exemple, $\langle (b)(d)(b) \rangle_3$ (l'indice en fin de séquence représente le support) est une séquence 1-libre alors que la séquence $\langle (b) \rangle_8$ ne l'est pas, puisqu'elle a le même support que sa sous-séquence $\langle \rangle_8$.

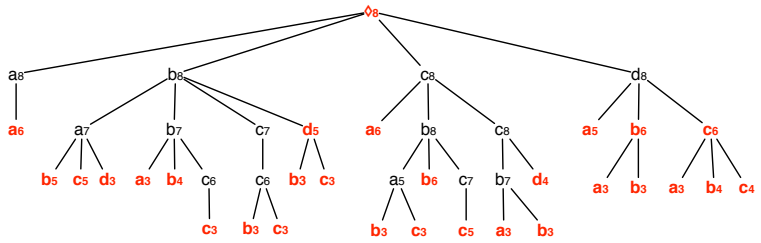


FIG. 1: L'arbre d'énumération des motifs séquentiels 1-libres (en gras) de $\mathcal{D}$ ($\sigma = 3$)

De par la définition d'une séquence δ-libre (cf. définition 4), pour tout $\delta \in \mathbb{N}^+$, un motif $\delta + 1$ libre est nécessairement δ-libre. Comme la réciproque n'est pas vraie, cela signifie que plus δ est grand, plus la taille de la collection de motifs δ-libres est réduite. Les expériences montrent (cf. section 4) qu'augmenter δ réduit très fortement le nombre de motifs produits.

La section suivante précise la difficulté d'extraction des motifs séquentiels δ-libres et propose une méthode d'extraction efficace.

3 Extraction de motifs séquentiels δ-libres

L'extraction des motifs δ-libres étant bien maîtrisée pour les données ensemblistes, il est alors naturel de s'interroger sur l'absence de résultats pour ce type de motifs sur les séquences. Une explication est qu'il est délicat, de façon générale, d'étendre aux séquences une méthode fonctionnant sur des données ensemblistes. La difficulté réside principalement dans la combinatoire, encore plus élevée dans le cas des séquences.

Nous commençons par montrer que la δ-liberté n'est plus une contrainte antimonotone pour les séquences. L'antimonotonie d'une contrainte étant au cœur de puissantes techniques d'élagage en extraction de motifs, elle a permis de concevoir des solveurs génériques qui poussent les contraintes satisfaisant cette propriété. Nous pensons que la perte de cette propriété explique en grande partie le manque de techniques d'extraction de séquences δ-libres. Ce constat motive notre recherche de nouvelles propriétés pour extraire ces motifs.

Propriété 1. *La contrainte de δ-liberté n'est pas antimonotone dans le cadre des séquences.*

Il est simple de constater la perte de l'antimonotonie de la δ-liberté. Par exemple, à partir des données de la table 1, la séquence $\langle (a)(a) \rangle_6$ est 1-libre alors que $\langle (a) \rangle_8$ ne l'est pas. Cette perte souligne la différence fondamentale entre séquences et motifs ensemblistes pour leur extraction et la complexité algorithmique qui va en résulter. Ainsi, avec les séquences, il devient impossible d'utiliser des mécanismes qui sont efficaces pour les motifs ensemblistes, tels que l'inférence du support de certains motifs. Notons que ce constat rejoint le résultat pessimiste de Raïssi et al. (2008).

Récemment, Gao et al. (2008) et Lo et al. (2008) ont simultanément introduit une propriété de monotonie pour un ensemble particulier de séquences non-génératrices. Dans cet article, nous étendons ce résultat au cas des séquences δ-libres et nous montrons une nouvelle propriété qui nous permettra d'éviter le test de séquences « non prometteuses » (cf. propriété 2 ci-dessous). Cette généralisation s'appuie sur la notion de δ-équivalence de bases projetées. Notons que les générateurs de séquences de Gao et al. (2008) et Lo et al. (2008) sont un cas particulier de séquences δ-libres (i.e., $\delta = 0$).

Définition 5 (δ-équivalence de bases projetées)**.** *Soient deux séquences s et s', leurs bases projetées respectives $\mathcal{D}_{|s}$ et $\mathcal{D}_{|s'}$ sont dites δ-équivalentes (notée $\mathcal{D}_{|s} \equiv_\delta \mathcal{D}_{|s'}$) si ces dernières ont au plus δ suffixes différents.*

Cette définition peut être pleinement exploitée pour produire une propriété monotone de certaines séquences non δ-libres :

Propriété 2. *Soient deux séquences s et s'. Si $s' \prec s$ et $\mathcal{D}_{|s} \equiv_\delta \mathcal{D}_{|s'}$, alors il n'existe pas de séquence de préfixe s qui puisse être δ-libre.*

Démonstration. (Par contradiction) Soient deux séquences s et s' telles que $s' \prec s$, $Support(s', \mathcal{D}) - Support(s, \mathcal{D}) \leq \delta$ et $\mathcal{D}_{|s} \equiv_\delta \mathcal{D}_{|s'}$. Supposons qu'il existe une séquence $s_p = s \cdot s_c$ δ-libre. Puisque $\mathcal{D}_{|s} \equiv_\delta \mathcal{D}_{|s'}$, il existe une séquence $s'' = s' \cdot s_c$ telle que $Support(s'', \mathcal{D}) - Support(s_p, \mathcal{D}) \leq \delta$. Ce qui nous emmène à une contraction par rapport au fait que s_p est supposée δ-libre. □

La propriété 2 est très intéressante car elle permet d'éviter l'exploration de séquences « non prometteuses ». De plus, la vérification de la δ-équivalence des bases projetées peut être restreinte aux sous-séquences de longueur $n-1$ comme la propriété 3 l'indique :

Propriété 3 (Backward pruning)**.** *Soit une séquence préfixe $s_p = \langle e_1, e_2, \ldots, e_n \rangle$. S'il existe un entier i ($1 \leq i < n-1$) tel que $\mathcal{D}_{|s_p} \equiv_\delta \mathcal{D}_{|s_p^{(i)}}$, alors l'exploration de la séquence s_p peut être stoppée puisqu'aucun motif séquentiel de préfixe s_p peut être δ-libre dans $\mathcal{S}$.*

La propriété 3 peut être facilement poussée dans un algorithme d'extraction de motifs séquentiels δ-libres, elle permet alors un élagage efficace des séquences *non prometteuses*. En effet, la propriété 4 assure que l'élagage issu de la propriété 3 ne peut pas entraîner l'élagage de séquences δ-libres.

Propriété 4. *Soit une séquence préfixe $s_p = \langle e_1, e_2, \ldots, e_n \rangle$. Si s_p est δ-libre alors s_p ne peut pas être élaguée (i.e., elle n'est pas considérée comme non prometteuse).*

Démonstration. Si s_p est δ-libre alors, par définition de la δ-liberté, il n'existe pas d'entier i tel que $Support(s_p, \mathcal{D}) + \delta < Support(s_p{}^{(i)}, \mathcal{D})$. Par conséquent, il n'existe pas d'entier i tel que $s_p \equiv_\delta s_p{}^{(i)}$ et l'élagage de s_p ne peut donc pas être appliqué. □

L'algorithme que nous proposons pour extraire les séquences δ-libres s'appuie sur les propriétés précédentes afin de pleinement exploiter la propriété de monotonie de certaines séquences non δ-libres. D'autre part, il est aussi possible de tirer parti de la combinaison conjointe des contraintes de δ-liberté et de fréquence. En effet, dans le cas où les séquences explorées sont dans le voisinage de la bordure positive des séquences fréquentes, alors la combinaison de ces deux contraintes peut être utilisée pour réduire l'espace de recherche comme la propriété suivante le montre.

Propriété 5. *Soient le seuil de support minimum σ et la séquence préfixe s_p. Si $\sigma \leq Support(s_p, \mathcal{D}) \leq \sigma + \delta$, alors l'exploration de s_p peut être stoppée.*

Démonstration. On montre aisément que les séquences de préfixe s_p ne peuvent pas être à la fois fréquentes et δ-libres. □

Nous présentons maintenant DEFFED, notre algorithme d'extraction de séquences δ-libres fréquentes. DEFFED (DElta Free Frequent sEquence Discovery) intègre les propriétés 3 et 5. Dans le même esprit que l'algorithme d'extraction de motifs séquentiels fermés Bide de Wang et al. (2007), DEFFED est basé sur le paradigme *pattern growth* (voir Pei et al. (2004)) et effectue l'extraction de séquences δ-libres sans gestion d'un ensemble de motifs potentiellement δ-libres. Il adopte ainsi une vérification *bi-directionnelle* pour élaguer efficacement l'espace de recherche. DEFFED ne conserve que les séquences fréquentes qui sont δ-libres. C'est un avantage très important par rapport aux méthodes qui s'appuient sur la gestion d'un ensemble de motifs potentiellement δ-libres (e.g. Clospan pour les fermés), ce qui nécessite un post-traitement quadratique en la cardinalité de l'ensemble et qui peut s'avérer prohibitif lorsque cette cardinalité devient trop importante.

Pour découvrir l'ensemble correct et complet des séquences δ-libres fréquentes d'une base de séquences $\mathcal{D}$ (i.e., toutes les séquences δ-libres fréquentes de préfixe $\langle\rangle$), l'algorithme DEFFED est lancé de la façon suivante : DEFFED$(\sigma, \delta, \langle\rangle, \mathcal{D}, \{\langle\rangle_{|\mathcal{D}|}\})$. En effet, $\langle\rangle_{|\mathcal{D}|}$ est, par définition, la plus petite séquence δ-libre. DEFFED parcourt d'abord la base de séquences pour extraire toutes les 1-séquences fréquentes (ligne 2). Ensuite, chaque 1-séquence fréquente (ligne 5) est considérée comme séquence préfixe. L'algorithme vérifie si la séquence préfixe est δ-libre (ligne 9). Finalement, si la séquence préfixe satisfait les conditions des lignes 13

Algorithme 1 : DEFFED (DElta Free Frequent sEquence Discovery)

Data : σ, δ, séquence préfixe s_p et sa base projetée $\mathcal{D}_{|s_p}$, FFS
Result : $FFS \cup$ L'ensemble des séquences fréquentes δ-libres de préfixe s_p

```
1  begin
2      LFI ← frequent 1-sequences(D_|sp, σ) ;
3      is_free ← ⊥;
4      unpromising ← ⊥;
5      foreach item e ∈ LFI do
6          s'p = ⟨sp · e⟩;
7          D_|s'p ← pseudo_projected_database(D_|sp , s'p);
8          if Support(s'p, D) + δ < Support(sp, D) then
               // potentiellement δ-libre
9              if ∄ integer i and Support(s'p^(i), D) − δ > Support(s'p, D) then
10                 FDS ← FDS ∪ {s'p};
11                 is_free ← ⊤;
12         if ¬is_free then
13             if ∄ integer i | D_|s'p ≡δ D_|s'p^(i) then
14                 unpromising ← ⊤;
15         if ¬unpromising then
               // vérifier s'il est possible de trouver des séquences
                  fréquentes δ-libres(propriété 5)
16             if Support(s'p, D) > σ + δ then
17                 Call DEFFED(σ, δ, s'p, D_|s'p , FFS);
18 end
```

et 16 (on ne peut pas élaguer), l'algorithme est appelé récursivement sur la séquence préfixe. D'autre part, combinée avec la bordure positive des motifs séquentiels fréquents, DEFFED permet d'obtenir la représentation condensée approximative des séquences δ-libres fréquentes.

4 Expériences

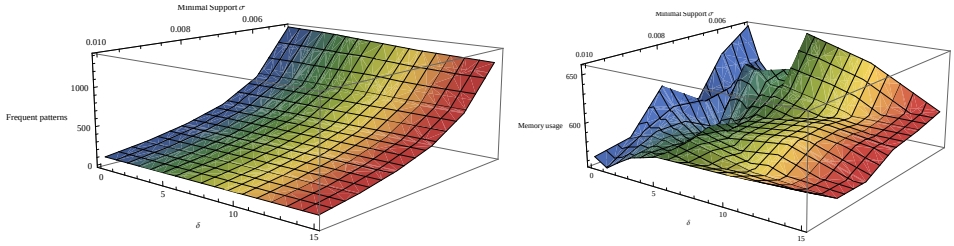
Notre algorithme DEFFED, ainsi que les algorithmes BIDE Wang et al. (2007) et PrefixSpan Pei et al. (2004), ont été implémentés en Java. Les expérimentations ont été effectuées sur un cluster où un ordonnanceur a alloué les "jobs". Les nœuds de calcul sont équipés de 8 processeurs cadencés à 2.5 GHz et de 16Go de mémoire sous CentOS. Afin de dresser un bilan complet de notre approche, nous menons des expériences sur un jeu de données synthétique et un jeu de données réel. Plusieurs autres jeux de données ont été testés, les résultats ne sont pas rapportés ici par manque de place.

Le jeu de données synthétique C100I20L30 (100000 séquences de taille moyenne 30 sur un alphabet de 20000 items) a été créé à l'aide du générateur d'IBM QUEST [1]. Le jeu de données réels contient les résultats des quinze dernières années des rencontres de football de la *Premier League* [2], première division anglaise de football. Ce jeu de données contient 280 séquences

1. http ://www.almaden.ibm.com/cs/projects/iis/hdb/Projects/data_mining/datasets/syndata.html
2. http ://www.premierleague.com/

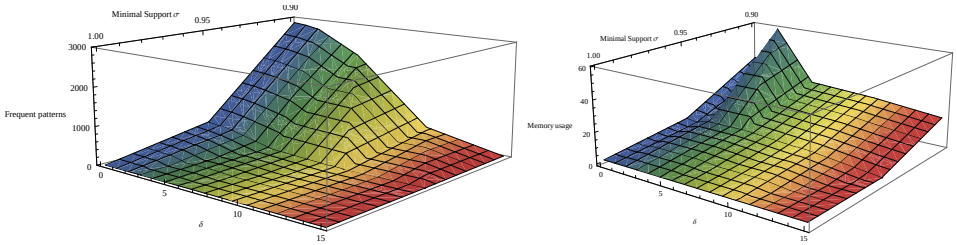
contenant chacune 38 itemsets de taille 2 (cette taille est fixe et ne varie pas). Chaque itemset représente une journée du championnat avec trois issues possibles encodées dans le premier item : victoire, défaite ou nul. Le deuxième item représente les buts marqués et autres détails de la partie. De par son encodage, ce jeu de données est considéré comme très dense.

Ces expérimentations ont pour but de répondre aux questions suivantes : *comment se comporte* DEFFED *en fonction de ses différents paramètres et par rapport aux autres solveurs ? Quel est l'apport des motifs extraits dans des tâches de classification ?*



(a) #δ-libres (b) Utilisation mémoire (Mo)

FIG. 2: Comportement de DEFFED sur C100I20L30 en fonction du support minimum et δ

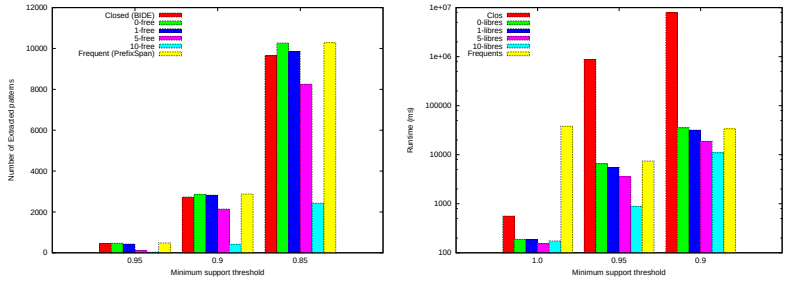


(a) #δ-libres (b) Utilisation mémoire (Mo)

FIG. 3: Comportement de DEFFED sur PremierLeague en fonction du support minimum et δ

Comportement de DEFFED. Les deux jeux de données sont utilisés dans cette expérimentation. Nous comparons les nombres de motifs δ-libres produits suivant différentes valeurs de δ, par rapport aux nombres de motifs clos et fréquents, ainsi que les temps d'exécution (l'algorithme BIDE est utilisée pour extraire les clos et PrefixSpan pour les fréquents).

Les résultats sont présentés dans les figures 2, 3 et 4. Les figures 2 et 3 montrent l'importance de la valeur δ qui permet de limiter l'espace de recherche et donc le nombre de motifs extraits et par conséquence la taille mémoire utilisée ainsi que le temps d'execution. Plus δ est grand, plus l'élagage sera important. Le temps d'extraction des motifs et l'utilisation mémoire pour ce processus sont ainsi diminués grâce au seul paramètre δ et permet de pousser le processus global d'extraction vers des supports intéressants pour les jeux de données choisis. La figure 4 présente la comparaison avec les algorithmes BIDE et PrefixSpan. Le lecteur appréciera le gain gagné lorsque que δ augmente. Notons également le fait que BIDE obtient de plus mauvaises performances que PrefixSpan car le nombre de fermés est proche du nombre de fréquents, ce qui ne rentabilise pas le coût des vérifications bidirectionnelles de BIDE.



(a) Nombre de motifs (b) Temps d'exécution

FIG. 4: Comparaison de DEFFED, avec PrefixSpan et BIDE sur PremierLeague en fonction du support minimum

Descripteurs	Support	SVM	Nb descripteurs	Temps d'obtention du classifieur (s)
Motifs fréquents	0.9	98.5714	2879	2.55
Motifs fermés	0.9	98.5714	2737	2.58
Motifs 0-libres	0.9	98.5714	2864	2.22
Motifs 5-libres	0.9	98.2143	2144	2.10
Motifs 10-libres	0.9	**99.2857**	416	0.62
Motifs 15-libres	0.9	98.9286	27	0.09
Motifs 17-libres	0.9	98.9286	**9**	**0.11**
Motifs fréquents	0.85	97.5	10289	9.89
Motifs 0-libres	0.85	97.5	10274	8.88
Motifs 5-libres	0.85	97.5	9479	8.82
Motifs 10-libres	0.85	97.8571	5227	3.89
Motifs 15-libres	0.85	98.2143	957	0.7
Motifs 17-libres	0.85	**98.9286**	**357**	**0.39**
Motifs fréquents	0.8	95.3571	29597	35.53
Motifs 0-libres	0.8	95.3571	29582	36.77
Motifs 5-libres	0.8	95.3571	28746	32.42
Motifs 10-libres	0.8	94.6429	22466	36.67
Motifs 15-libres	0.8	96.7857	8935	11.65
Motifs 17-libres	0.8	**97.1429**	**4795**	**3.59**

TAB. 2: Pourcentage d'éléments correctement classifiés.

Apport des motifs δ-libres dans un but de classification. Afin d'évaluer l'intérêt des motifs δ-libres dans un but de classification, nous les avons utilisés comme descripteurs dans une tâche de classification. Le jeu de données PremierLeague comporte 5 classes : les équipes finissant dans les 4 premières, les équipes classées entre la cinquième et huitième position, celles finissant entre la neuvième et quatorzième, puis celles terminant entre les quinzièmes et dix-septièmes positions et enfin les équipes reléguées.

Une fois les motifs δ-libres extraits, nous les avons utilisés comme descripteurs de ce jeu de données. Le but de notre démarche n'est pas une comparaison entre classifieurs, mais la comparaison et l'étude de l'impact des δ-libres, en tant que descripteurs, par rapport à d'autres types de motifs, sur les performances d'un classifieur donné. Pour cela, nous avons utilisé un classifieur SVM. Les performances du classifieur ont été évaluées avec une 10 validation croisée. Nous comparons les résultats obtenus avec les motifs δ-libres par rapport aux motifs fréquents et aux motifs fermés qui sont généralement utilisés dans ces tâches de classification. Le tableau 2 rapporte les résultats [1]. Hormis le fait que les meilleurs résultats sont obtenus à

1. Les supports hauts (entre 80% et 100%) choisis dans cette expérimentation reflètent la densité et la complexité

partir de certaines valeurs de δ pour des seuils de support donnés, on remarque que les δ-libres produisent des résultats de classification comparables aux autres types de descripteurs. Plus remarquable, on note que le nombre de descripteurs est beaucoup plus faible avec les motifs δ-libres et le temps de calcul du classifieur est aussi plus court. Par exemple, pour un seuil de support $\sigma = 0.9$ et $\delta = 17$, il faut uniquement 9 descripteurs pour obtenir un classifieur obtenant un taux de bien classés de 98.9286% en 0.11s alors qu'il en faut 2879 avec les motifs fréquents pour une performance équivalente. De plus, remarquons que le temps de construction du classifieur ne prend pas en compte le temps d'extraction qui dans ce cas là est d'environ 34 secondes pour Prefixspan alors que dans le cas de notre algorithme, il est d'environ 3 secondes. Les δ-libres permettent ainsi d'obtenir des précisions comparables avec beaucoup moins de descripteurs et un temps de calcul réduit.

5 Travaux connexes

La recherche de représentations concises des motifs fréquents a donné lieu à de nombreux travaux depuis l'article fondateur de Mannila et Toivonen (1996). Une telle représentation est composée d'un sous-ensemble des motifs fréquents qui forme une synthèse exacte de l'ensemble des motifs (toute l'information véhiculée par les motifs fréquents peut être retrouvée). La plupart des travaux portent sur les motifs ensemblistes (i.e., itemsets), principalement parce qu'il existe des relations fortes entre les motifs ensemblistes et de puissants outils mathématiques comme la théorie des ensembles, la combinatoire et les correspondances de Galois. Ces outils jouent un rôle important dans la construction des représentations condensées fondées sur les motifs clos Pasquier et al. (1999), les motifs essentiels Casali et al. (2005), les motifs δ-libres de Boulicaut et al. (2003) (également appelés clés ou générateurs dans le cas particulier où $\delta = 0$) et les motifs non-dérivables de Calders et Goethals (2002). Calders et al. (2004) proposent un état de l'art de ce domaine.

En revanche, obtenir des représentations condensées dans des données structurées s'avère nettement plus complexe. Nous avons vu que la (δ) liberté ne satisfait plus la propriété de monotonie avec les séquences (cf. section 3). Les motifs fermés ont été étendus aux séquences avec des algorithmes efficaces comme Clospan de Yan et al. (2003) et Bide de Wang et al. (2007). Récemment, les générateurs ont été proposés pour les séquences par Baralis et al. (2008), Gao et al. (2008) et Lo et al. (2008). Notre contribution est une généralisation de ces travaux. L'algorithme DEFFED permet d'extraire toutes les séquences d'itemsets δ-libres alors que les précédents travaux correspondent au cas particulier où $\delta = 0$. La faisabilité de DEFFED est obtenue par l'apport de nouveaux critères d'élagage pour pallier la perte de l'antimonotonie et la combinaison des contraintes de δ-liberté et de fréquence pour élaguer l'espace de recherche. Remarquons que l'état de l'art inclut aussi des résultats pessimistes, comme celui de Raïssi et al. (2008) qui démontrent qu'une représentation à base de séquences non-dérivables n'est pas possible.

En s'appuyant sur le principe du MDL Grunwald et al. (2005)), Li et al. (2006) démontrent que les motifs libres sont de meilleurs candidats que les motifs fermés dans des tâches de classification. Paradoxalement, dans le contexte des séquences, les motifs libres n'ont pas été utilisés à des fins de classification hormis par Gao et al. (2008). Depuis la première proposition

inhérente au jeu de données PremierLeague, ainsi pour l'algorithme BIDE, l'implémentation dépasse largement le temps alloué pour les extractions à partir de 85%

de Lesh et al. (1999), les principaux travaux s'appuient sur des séquences fréquentes. Park et Kanehisa (2003) et She et al. (2003) combinent méthodes de sélection de descripteurs et classifieur fondé sur les SVM pour prédire les caractéristiques des membranes des protéines. Exarchos et al. (2009) et Tseng et Lee (2009) proposent de construire directement un classifieur associatif à partir de motifs séquentiels. Ces travaux considèrent ainsi l'ensemble complet des motifs séquentiels fréquents, ce qui diverge du fait que l'ensemble des motifs d'un classifieur associatif doit être concis et sans redondance d'après Bringmann et al. (2009). Comme la partie expérimentale l'a montré, les motifs séquentiels δ-libres peuvent être utilisés comme descripteurs pour la classification et leur nombre réduit en font de bons candidats pour les tâches de classification associative.

6 Conclusion

Nous avons proposé un nouveau type de motifs séquentiels, les motifs séquentiels δ-libres, ainsi qu'un algorithme correct et complet qui les extrait efficacement. Leur extraction n'est pas une tâche simple parce que, contrairement aux données ensemblistes, la δ-liberté ne vérifie plus la propriété d'antimonotonie pour les séquences. La réussite de leur extraction s'appuie sur la mise en évidence de propriétés d'élagage fondées sur les bases projetées. Les expérimentations montrent que ces motifs sont nettement moins nombreux que les motifs séquentiels libres et donc encore moins nombreux que l'ensemble des motifs séquentiels. Elles montrent aussi les apports de ces motifs en tant que descripteurs en classification. De plus, ces motifs possèdent une capacité à autoriser quelques exceptions dans la production de règles, ce qui s'avère particulièrement utiles dans un contexte de classification. Associés au fait que leur nombre soit réduit, nous pensons que ces motifs sont particulièrement appropriées pour faciliter l'étape de sélection et combinaison de séquences en classification associative, étape qui reste l'une des difficultés majeures de ce type de méthodes.

Références

Agrawal, R. et R. Srikant (1995). Mining sequential patterns. In *Proceedings of the 11th International Conference on Data Engineering (ICDE 95)*, Tapei, Taiwan, pp. 3–14.

Baralis, E., S. Chiusano, R. Dutto, et L. Mantellini (2008). Compact representations of sequential classification rules. In T. Y. Lin, Y. Xie, A. Wasilewska, et C.-J. Liau (Eds.), *Data Mining : Foundations and Practice*, Volume 118 of *Studies in Computational Intelligence*, pp. 1–30. Springer.

Bonchi, F. et C. Lucchese (2007). Extending the state-of-the-art of constraint-based pattern discovery. *Data Knowl. Eng. 60*(2), 377–399.

Boulicaut, J.-F., A. Bykowski, et C. Rigotti (2003). Free-sets : A condensed representation of boolean data for the approximation of frequency queries. *Data Min. Knowl. Discov. 7*(1), 5–22.

Bringmann, B., S. Nijssen, et A. Zimmermann (2009). Pattern-based classification : A unifying perspective. In *From Local Patterns to Global Models : Proceedings of the ECML PKDD 2009 Workshop*.

Calders, T. et B. Goethals (2002). Mining all non-derivable frequent itemsets. In *PKDD*, pp. 74–85.

Calders, T., C. Rigotti, et J.-F. Boulicaut (2004). A survey on condensed representations for frequent sets. In *Constraint-Based Mining and Inductive Databases*, pp. 64–80.

Casali, A., R. Cicchetti, et L. Lakhal (2005). Essential patterns : A perfect cover of frequent patterns. In *DaWaK*, pp. 428–437.

Dong, G. et J. Pei (2007). *Sequence Data Mining*, Volume 33 of *Advances in Database Systems*. Springer.

Exarchos, T. P., M. G. Tsipouras, C. Papaloukas, et D. I. Fotiadis (2009). An optimized sequential pattern matching methodology for sequence classification. *Knowl. Inf. Syst. 19*(2), 249–264.

Gao, C., J. Wang, Y. He, et L. Zhou (2008). Efficient mining of frequent sequence generators. In *WWW*, pp. 1051–1052.

Grunwald, P., I. Myung, et M. Pitt (2005). *Advances in Minimum Description Length*. MIT Press.

Lesh, N., M. J. Zaki, et M. Ogihara (1999). Mining features for sequence classification. In *KDD*, pp. 342–346.

Li, J., H. Li, L. Wong, J. Pei, et G. Dong (2006). Minimum description length principle : Generators are preferable to closed patterns. In *AAAI*. AAAI Press.

Lo, D., S.-C. Khoo, et J. Li (2008). Mining and ranking generators of sequential patterns. In *SDM*, pp. 553–564.

Mannila, H. et H. Toivonen (1996). Multiple uses of frequent sets and condensed representations (extended abstract). In *KDD*, pp. 189–194.

Mannila, H., H. Toivonen, et A. I. Verkamo (1997). Discovery of frequent episodes in event sequences. *Data Min. Knowl. Discov. 1*(3), 259–289.

Park, K.-J. et M. Kanehisa (2003). Prediction of protein subcellular locations by support vector machines using compositions of amino acids and amino acid pairs. *Bioinformatics 19*(13), 1656–1663.

Pasquier, N., Y. Bastide, R. Taouil, et L. Lakhal (1999). Discovering frequent closed itemsets for association rules. In *ICDT'99*, pp. 398–416.

Pei, J., J. Han, B. Mortazavi-Asl, J. Wang, H. Pinto, Q. Chen, U. Dayal, et M. Hsu (2004). Mining sequential patterns by pattern-growth : The prefixspan approach. *IEEE Trans. Knowl. Data Eng. 16*(11), 1424–1440.

Raïssi, C., T. Calders, et P. Poncelet (2008). Mining conjunctive sequential patterns. *Data Min. Knowl. Discov. 17*(1), 77–93.

She, R., F. C. 0002, K. Wang, M. Ester, J. L. Gardy, et F. S. L. Brinkman (2003). Frequent-subsequence-based prediction of outer membrane proteins. In *KDD*, pp. 436–445.

Tseng, V. S. et C.-H. Lee (2009). Effective temporal data classification by integrating sequential pattern mining and probabilistic induction. *Expert Syst. Appl. 36*(5), 9524–9532.

van Leeuwen, M., J. Vreeken, et A. Siebes (2009). Identifying the components. *Data Min. Knowl. Discov. 19*(2), 176–193.

Wang, J., J. Han, et C. Li (2007). Frequent closed sequence mining without candidate maintenance. *IEEE Trans. Knowl. Data Eng. 19*(8), 1042–1056.

Yan, X., J. Han, et R. Afshar (2003). Clospan : Mining closed sequential patterns in large databases. In *SDM*.

Summary

Sequential pattern mining is a challenging task with important locks like the size of the output. In this paper, we propose a new approach that extract the more general patterns and suppress the more specific patterns with similar frequencies. We defined δ-sequential patterns that enable to reduce the output. Even if this notion is already known for itemsets, we show that its extension to the sequence framework is very difficult. The approach produces few and useful patterns for data mining tasks like sequence classification.