Extension des bases de données inductives pour la découverte de chroniques

Alexandre Vautier*, Marie-Odile Cordier*, René Quiniou*
* Irisa - Projet DREAM Campus de Beaulieu 35042 RENNES Cedex, France
{Alexandre.Vautier,Marie-Odile.Cordier,Rene.Quiniou}@irisa.fr

Résumé. Les bases de données inductives intègrent le processus de fouille de données dans une base de données qui contient à la fois les données et les connaissances induites. Nous nous proposons d'étendre les données traitées afin de permettre l'extraction de motifs temporels fréquents et non fréquents à partir d'un ensemble de séquences d'évènements. Les motifs temporels visés sont des chroniques qui permettent d'exprimer des contraintes numériques sur les délais entre les occurrences d'événements.

1 Introduction

L'enjeu de la *fouille de données* est d'extraire des connaissances cachées à partir d'une grande quantité de données disponibles. En ne disposant, a priori, d'aucune information particulière sur les données, on souhaite découvrir des connaissances cachées sous forme de *motifs*. Dans ce cadre, on appelle motif un concept (une loi, une règle, une classe,...) synthétisant un ensemble de données.

Les bases de données inductives (BDIs), introduites par Imielinski et Mannila (Imielinski et Mannila 1996), sont nées de la nécessité de formaliser la fouille de données et d'établir des liens clairs avec les concepts et les algorithmes utilisés dans le domaine des bases de données. L'un des buts des BDIs, est de récupérer l'efficacité de cette gestion au profit de la fouille de données. Dans une BDI, les données et leurs motifs sont gérés de la même façon. Ainsi, le processus de fouille de données devient un processus interactif dans lequel l'utilisateur soumet des requêtes à la BDI qui lui fournit en résultat des motifs qui sont, soit entreposés directement dans la BDI, soit « induits » par un processus basé sur la recherche de motifs intéressants, par exemple fréquents.

Dans les BDIs actuelles (Lee et De Raedt 2003, Dzeroski 2002, Imielinski et Mannila 1996, De Raedt 2002) les données sont des séquences de symboles et les motifs sont des séquences complexes de symboles. Nous proposons d'adjoindre une dimension temporelle numérique aux BDIs. En effet, de nombreux problèmes de découverte de connaissances nécessitent d'intégrer d'autres phénomènes qu'une simple succession de symboles. Ainsi, dans une BDI étendue au temps, une donnée est une *séquence d'évènements* et un motif est une *chronique* (Dousson et Ghallab 1994) constituée d'un ensemble de variables temporelles typées et soumis à des contraintes numériques.

La partie 2 expose les techniques d'extraction de motifs temporels proches des chroniques. La partie 3 détaille l'extension des BDIs : une relation de généralité entre chroniques permettant de structurer l'espace de recherche et un critère de reconnaissance utilisé pour définir la fréquence d'une chronique dans une séquence d'évènements sont introduits. La partie 4 expose comment traiter une requête à la base de donnée. L'algorithmes de l'espace des versions (Mitchell 1997) est modifié pour utiliser les chroniques minimales et fréquentes d'une séquence d'évènements calculées par une adaptation d'un outil de fouille de données FACE (Dousson et Duong 1999). Enfin une illustration et des expérimentations montrent l'intérêt de cette approche.

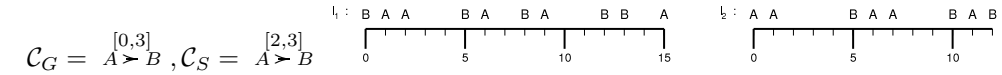
2 Le temps en fouille de données

En fouille de données, la prise en compte du temps dans les connaissances à extraire rend plus complexe la recherche des motifs intéressants. Pour gérer cet attribut particulier, différentes méthodes ont été proposées dans la littérature. Tout d'abord, il existe des techniques d'extraction de motifs qui utilisent le temps comme un élément permettant d'ordonner des symboles. Par exemple, SeqLog (Lee et De Raedt 2003) extrait des séquences complexes de symboles dans les séquences d'une BDI ou encore Winepi et Minepi (Mannila et al. 1997) permettent d'extraire des motifs dont les symboles sont soit séquentiels soit parallèles. Les chroniques permettent la combinaison de ces deux formes en rajoutant des contraintes numériques sur le délai entre évènements. La recherche de motifs dans les séries temporelles utilisent, après une discrétisation du signal, des techniques relativement proches de la notre. Par exemple, Lin et al. (Lin et al. 2002) extraient des motifs fréquents qui représentent un ensemble de portions continues du signal proches les unes des autres selon une mesure de distance. Comparée à cette technique, l'application de notre méthode sur une série temporelle discrétisée permet de découvrir des portions fréquentes et non continues sans mesure de distance.

Le temps est parfois traité comme un attribut numérique géré d'une façon particulière. Par exemple, Yoshida (Yoshida et al. 2000) ajoute la dimension temporelle à la découverte d'itemsets fréquents. Les enregistrements *datés* de la base de données sont regroupés, par exemple selon leur appartenance à un client. Leur algorithme permet d'extraire des séquences d'itemsets contraints temporellement ayant un support minimal : la séquence fréquente $< \{(a), -, -, -\}\{(b, c), 17, 18, 19\}\{(b), 5, 10, 12\} >$, où a, b et c sont des items, indique, par exemple, qu'il existe suffisamment de groupes d'enregistrements pour lesquels un enregistrement contenant l'item a est suivi d'un enregistrement contenant (b, c) lui-même suivi d'un enregistrement contenant l'item (b). Le délai entre les deux premiers enregistrements est compris entre 17 et 19 unités de temps et le délai entre les deux derniers enregistrements est compris entre 5 et 12. Les valeurs 18 et 10 représentent les délais moyens entre les enregistrements considérés dans la base de données. Cette méthode fait apparaître, sous forme d'intervalles entre itemsets, une notion de contrainte temporelle semblable à celle que nous utilisons. Par rapport à cette approche, nous nous limitons à des itemsets composés d'un seul item (les évènements), en revanche nous ajoutons la notion de parallélisme entre itemsets.

Lin propose une méthode (Lin 2003) permettant d'extraire tous les intervalles maximalement spécifiques et fréquents dans un ensemble d'intervalles à partir d'une relation de généralité sur les intervalles (l'inclusion par exemple). Un intervalle est maximalement spécifique et fréquent si toutes ses spécialisations (des réductions de l'intervalle) ne sont pas fréquentes. Nous nous plaçons dans le cas de multi-intervalles maximalement spécifiques et fréquents, ce qui rend la tâche beaucoup plus complexe.

On peut remarquer que les différentes formes de motifs utilisées dans les méthodes présentées plus haut sont toutes sont des spécialisations du modèle de chronique (Dousson et Ghallab 1994). C'est ce type de motif fréquent que recherche FACE (Dousson et Duong 1999), un outil de fouille de données destiné à analyser des journaux d'alarmes (séquence d'évènements) d'un réseau de télécommunications. Notre approche nécessite le calcul des chroniques maximalement spécifiques et fréquentes, or FACE retourne des chroniques fréquentes mais non maximalement spécifiques. Nous verrons dans la suite comment adapter FACE pour qu'il calcule de l'ensemble des chroniques attendues.



Instances de $\mathcal{C}_S$ ou $\mathcal{C}_G$ dans l_1 : $\{((A,2),(B,5)),((A,6),(B,8)),((A,9),(B,12))\}$
Instance de $\mathcal{C}_S$ ou $\mathcal{C}_G$ dans l_2 : $\{((A,7),(B,10))\}$

FIG. 1 – Un exemple de recherche de chroniques sous contraintes fréquentielles

$$\mathcal{C}_1 = (\mathcal{G}, \mathcal{A}), \qquad \mathcal{G} : t_1 \underset{[-2;1]}{\overset{[-1;2]}{\rightrightarrows}} t_2 \underset{[1;4]}{\rightarrow} t_3, \qquad \mathcal{A} : t_1 \rightarrow A \quad t_2 \rightarrow B \quad t_3 \rightarrow B$$

FIG. 2 – Une chronique $\mathcal{C}_1$ de taille 3. Les intervalles étiquetant les arcs représentent les délais minimaux et maximaux entre variables temporelles, par exemple : $1 \leq t_3 - t_2 \leq 4$.

3 Base de données inductive et données temporelles

Notre problème consiste à retrouver l'ensemble des chroniques qui ont une fréquence minimale dans certaines séquences d'évènements mais une fréquence maximale dans d'autres séquences. Par exemple, soit les séquences l_1 et l_2 de la figure 1, on souhaite retrouver les chroniques qui satisfont les contraintes $freq(\mathcal{C}, l_1) \geq 3 \wedge freq(\mathcal{C}, l_2) \leq 1$. Les chroniques $\mathcal{C}_S$ et $\mathcal{C}_G$ satisfont ces contraintes car elles ont 3 instances dans l_1 et 1 seule dans l_2. Il faut aussi remarquer que $\mathcal{C}_G$ est plus générale que $\mathcal{C}_S$ et qu'il n'existe pas de chronique solution plus spécifique que $\mathcal{C}_S$ ou plus générale que $\mathcal{C}_G$.

Une base de données inductive $I(\mathcal{D}, \mathcal{P})$ comprend deux composants. $\mathcal{D}$ contient les données regroupées en ensembles et $\mathcal{P}$ contient les motifs, eux aussi, regroupées en ensembles. Certaines BDIs (Lee et De Raedt 2003, Kramer et al. 2001) considèrent des données sous forme de séquences de symboles, par exemple $e_1 = ababcc$; $e_2 = abbc$; $e_3 = bb$; $e_4 = abc$; $e_5 = bc$; $e_6 = cc$. Les motifs sont des séquences complexes qui autorisent la succession non immédiate de symboles, par exemple la séquence complexe $ba < cd$ indique que les symboles cd ne se situent pas forcément juste après les symboles ba dans une séquence. À la requête $freq(p, e_1) \geq 2$, où p est l'ensemble recherché des séquences complexes de fréquence supérieure ou égale à 2 dans e_1, on obtient en résultat $p = \{a, b, ab, a < c, b < c, c\}$. Nous proposons d'associer des informations temporelles numériques à ce type de données et motifs. Nous reprenons la définition d'une séquence d'évènements utilisée par Mannila et al. (Mannila et al. 1997) qui étend les séquences de symboles. Nous définissons une chronique qui est un motif temporel et qui étend une séquence complexe. Un exemple de chronique est donné dans la figure 2.

3.1 Chroniques : définitions

Définition 1 (Évènement)
Soit S un ensemble de types d'évènements. *Un* évènement *e est une paire (s, d), où $s \in S$ est un type d'évènement et d est un entier dénotant la* date d'occurrence *de e.*

Définition 2 (Séquence d'évènements)
Une séquence d'évènements est une liste d'évènements ordonnés selon leur date.

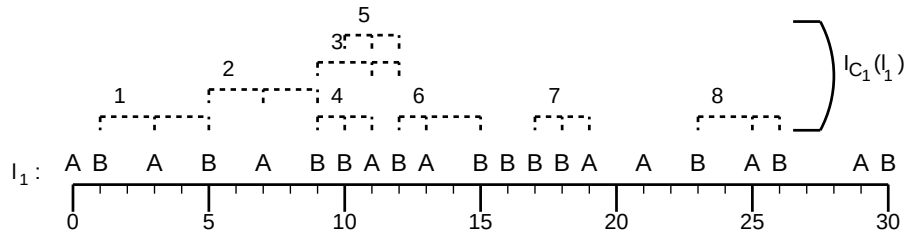


FIG. 3 – Séquence d'évènements l_1 et les instances $\mathcal{I}_{\mathcal{C}_1}(l_1)$

Définition 3 (Chronique)
Une chronique $\mathcal{C}$ est un doublet $(\mathcal{G}, \mathcal{A})$, où
- *$\mathcal{G} = (\mathcal{V}, \mathcal{E})$ est un graphe de contraintes temporelles minimal où les nœuds (notés t) de $\mathcal{V}$ représentent les variables temporelles et les arcs (notés e) de $\mathcal{E}$ représentent les contraintes temporelles (notés c_e), toutes sous la forme d'un intervalle unique.*
- *$\mathcal{A} : \mathcal{V} \rightarrow S$ est une fonction de typage qui associe chaque variable temporelle de $\mathcal{V}$ à un type d'évènement de S.*

On note $|\mathcal{C}| = |\mathcal{V}|$ la taille de $\mathcal{C}$.

Les occurrences d'une chronique dans une séquence d'évènements sont dénotées par des sous-séquences appelées *instances* de chronique. Par exemple, la séquence d'évènements l_1 (figure 3) comporte huit instances de la chronique $\mathcal{C}_1$ (figure 2).

Définition 4 (Instance de chronique)
Soient une chronique $\mathcal{C} = (\mathcal{G}, \mathcal{A})$ telle que $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ et une séquence d'évènements l. Une sous-séquence c de l est une instance de $\mathcal{C}$ dans l si et seulement s'il existe une fonction bijective $f : \mathcal{V} \rightarrow c$ et une substitution $\sigma : \mathcal{V} \rightarrow \mathbb{N}$ telles que

$(\forall t \in \mathcal{V}, f(t) = (s, d) \Rightarrow \mathcal{A}(t) = s \wedge \sigma(t) = d) \wedge (\forall e \in \mathcal{E}, e = (t, t') \Rightarrow \sigma(t') - \sigma(t) \in c_e)$

On note $\mathcal{I}_{\mathcal{C}}(l)$ l'ensemble des instances de $\mathcal{C}$ dans l.

3.2 Définition d'une relation de généralité entre chroniques

Établir une relation de généralité sur les chroniques est nécessaire pour réutiliser le concept d'espace des versions (Mitchell 1997) utilisé pour le calcul des requêtes sur une BDI. Auparavant, nous introduisons la notion d'*appariement* entre deux chroniques.

Définition 5 (Appariement entre deux chroniques)
Soient deux chroniques $\mathcal{C} = (\mathcal{G}, \mathcal{A})$ et $\mathcal{C}' = (\mathcal{G}', \mathcal{A}')$ telles que $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ et $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$. Une fonction $\mathfrak{a} : \mathcal{V} \rightarrow \mathcal{V}'$ (dont le domaine de définition est noté $D_{\mathfrak{a}}$) est un appariement entre $\mathcal{C}$ et $\mathcal{C}'$ si et seulement si :
- *$\forall t \in D_{\mathfrak{a}}, t' = \mathfrak{a}(t) \wedge \mathcal{A}'(t') = \mathcal{A}(t)$*
- *La restriction de $\mathfrak{a}$ à $D_{\mathfrak{a}}$ est une fonction bijective.*

Définition 6 (Appariement complet entre deux chroniques)
L'appariement $\mathfrak{a}$ est complet si et seulement s'il ne peut pas exister d'appariements entre $(\mathcal{G} \setminus D_{\mathfrak{a}})$ et $(\mathcal{G}' \setminus D_{\mathfrak{a}^{-1}})$. On note $\mathbb{A}_{\mathcal{C}}^{\mathcal{C}'}$ l'ensemble de tous les appariements complets entre deux chroniques $\mathcal{C}$ et $\mathcal{C}'$. Soit $n = |\mathcal{C}| \geq k = |\mathcal{C}'|$, on peut noter que $|\mathbb{A}_{\mathcal{C}}^{\mathcal{C}'}| \leq \frac{n!}{(n-k)!}$.

Une chronique est *plus générale* qu'une autre si on peut apparier tous les nœuds de son graphe à ceux du graphe de la seconde en respectant le typage des variables et l'inclusion des contraintes (cf. exemple de la figure 4). Ceci amène à définir des relations de généralité sur les contraintes temporelles puis sur les graphes de contraintes temporelles avant de définir celle sur les chroniques.

Définition 7 (Relation de généralité sur les contraintes temporelles)
Soient deux contraintes temporelles $c_e = [a, b]$ *et* $c_{e'} = [c, d]$. c_e *est plus générale que* $c_{e'}$ *(noté* $c_{e'} \subseteq c_e$*) si et seulement si* $a \leq c \leq d \leq b$.

Définition 8 (Relation de généralité sur les graphes temporels)
Soit $\mathfrak{a}$ *un appariement entre les deux graphes* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ *et* $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$. $\mathcal{G}$ *est plus général que* $\mathcal{G}'$ *selon* $\mathfrak{a}$ *(noté* $\mathcal{G}' \subseteq_{\mathfrak{a}} \mathcal{G}$*) si et seulement si :*

$$D_a = \mathcal{V} \wedge \forall (t_1, t_2) \in \mathcal{V} \times \mathcal{V}, e' = \mathcal{E}'(\mathfrak{a}(t_1), \mathfrak{a}(t_2)), e = \mathcal{E}(t_1, t_2), c_{e'} \subseteq c_e$$

Définition 9 (Relation de généralité entre chroniques)
Une chronique $\mathcal{C} = (\mathcal{G}, \mathcal{A})$ *est plus générale qu'une chronique* $\mathcal{C}' = (\mathcal{G}', \mathcal{A}')$*, noté* $\mathcal{C} \sqsubseteq \mathcal{C}'$*, si et seulement si*

$$\exists \mathfrak{a} \in \mathbb{A}_{\mathcal{C}}^{\mathcal{C}'}, \mathcal{G}' \subseteq_{\mathfrak{a}} \mathcal{G},$$

Pour tester la relation de généralité sur les graphes de contraintes temporelles, il suffit de tester toutes les contraintes, ainsi il existe un algorithme en $O(n^2)$. La vérification de la relation de généralité entre deux chroniques nécessite au pire l'énumération de tous les appariements complets. Ce test a donc au pire une complexité en $O(n!)$. On introduit les notions de chroniques minimales et maximales d'un ensemble selon la relation de généralité décrite.

Définition 10 (Chroniques minimales et maximales d'un ensemble)
Soit F *un ensemble de chroniques. Les chroniques minimales et maximales de* F *sont définies par :*

$$min(F) = \{f \in F | \forall q \in F, f \sqsubseteq q \Rightarrow f = q\}, max(F) = \{f \in F | \forall q \in F, q \sqsubseteq f \Rightarrow f = q\}$$

3.3 Fréquence de chroniques dans une séquence d'évènements

La mesure d'intérêt des motifs la plus utilisée en fouille de données est basée sur la fréquence. Pour utiliser la relation de généralité lors de la recherche de chroniques, les propriétés de monotonie et d'anti-monotonie doivent être satisfaites par les contraintes

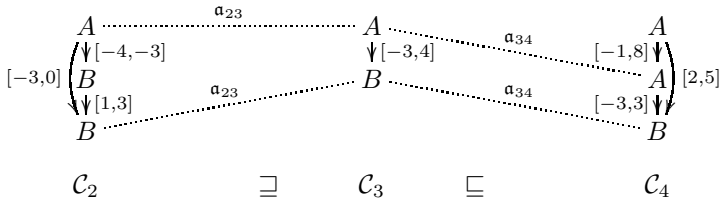


FIG. 4 – Relations de généralité entre trois chroniques $\mathcal{C}_3 \sqsubseteq \mathcal{C}_2$ et $\mathcal{C}_3 \sqsubseteq \mathcal{C}_4$

portant sur la fréquence. Ceci est vérifié si une chronique $\mathcal{C}'$ plus spécifique (resp. plus générale) qu'une chronique $\mathcal{C}$ a une fréquence égale ou moins (resp. plus) élevée. On introduit un critère de reconnaissance Q qui assure cette propriété en restreignant l'ensemble des instances comptabilisées lors du calcul de la fréquence.

Définition 11 (Fréquence de chroniques dans une séquence d'évènements)
La fréquence $freq^Q(\mathcal{C}, l)$ d'une chronique $\mathcal{C}$ dans la séquence d'évènements l selon un critère de reconnaissance Q est le cardinal de l'ensemble des instances de $\mathcal{C}$ dans l reconnues par Q.

$$E \subseteq \mathcal{I}_{\mathcal{C}}(l) \wedge Q(E) \wedge freq^Q(\mathcal{C}, l) = |E|$$

Définition 12 (Critère de reconnaissance)
Soit $2^{\mathbb{I}}$ l'ensemble des parties de l'ensemble des instances d'une chronique quelconque $\mathcal{C}$ dans l. $Q : 2^{\mathbb{I}} \rightarrow \{vrai, faux\}$ est un critère de reconnaissance si et seulement si :
- $\forall \mathcal{C}\ \forall l\ \exists!\ E \subseteq \mathcal{I}_{\mathcal{C}}(l) : Q(E)$,
- $\forall \mathcal{C}\ \forall \mathcal{C}', \mathcal{C} \sqsubseteq \mathcal{C}' \Rightarrow freq^Q(\mathcal{C}, l) \geq freq^Q(\mathcal{C}', l)$.

On peut trouver différents critères de reconnaissance dans la littérature. Le critère Q_m d'instances minimales (Mannila et al. 1997) extrait toutes les instances les plus courtes, en terme de durée dans la séquence d'évènements, d'un épisode (une collection d'évènements partiellement ordonnée). Le critère $Q_{d\&t}$ d'instances disjointes reconnues au plus tôt (Dousson et Duong 1999) extrait toutes les instances d'une chronique qui ne se chevauchent pas dans la séquence d'évènements et qui arrivent le plus tôt possible dans cette séquence. Par exemple dans la figure 3, $freq^{Q_{d\&t}}(\mathcal{C}_1, l_1) = 5$ selon le critère de reconnaissance $Q_{d\&t}$, car $Q_{d\&t}(c)$ est satisfait pour $c = \{c_1, c_4, c_6, c_7, c_8\}$.

3.4 Union de chroniques

On définit l'union de deux chroniques $\mathcal{C}$ et $\mathcal{C}'$ comme l'ensemble des chroniques $\mathcal{C}_i$ plus générales que $\mathcal{C}$ et $\mathcal{C}'$ tout en étant le plus spécifique possible. Cette opération correspond à la *lgg* (least general generalization) en apprentissage traditionnel. D'une part, les types des variables de $\mathcal{C}_i$ sont à la fois des types de variables de $\mathcal{C}$ et de $\mathcal{C}'$. D'autre part, les contraintes temporelles de toute chronique $\mathcal{C}_i$ sont plus générales que celles de $\mathcal{C}$ et $\mathcal{C}'$. La complexité de l'union de deux graphes de contraintes est en $O(n^2)$ alors que la complexité de l'union de chroniques est en $O(n!)$ car cette opération nécessite l'énumération de tous les appariements complets entre les deux chroniques.

Définition 13 (Union de contraintes temporelles)
Soient deux contraintes temporelles $c_e = [a, b]$ et $c_{e'} = [c, d]$. Leur union correspond à la contrainte $c = c_e \cup c_{e'}$ telle que $c = [min(a, c), max(b, d)]$

Définition 14 (Union de graphes de contraintes temporelles)
Soient deux graphes $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ et $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$ et un appariement $\mathfrak{a}$ entre eux, leur union selon $\mathfrak{a}$ correspond au graphe $\mathcal{G}'' = (\mathcal{V}'', \mathcal{E}'') = \mathcal{G} \cup_{\mathfrak{a}} \mathcal{G}$ tel que :

$$\forall e = (t_1, t_2) \in D_{\mathfrak{a}} \times D_{\mathfrak{a}}, \exists e' = (t'_1 = \mathfrak{a}(t_1), t'_2 = \mathfrak{a}(t_2)), \exists e'' = (t^{t'_1}_{t_1}, t^{t'_2}_{t_2}) \in \mathcal{E}'', c_{e''} = c_e \cup c_{e'}$$

Définition 15 (Union de chroniques)
Soit deux chroniques $\mathcal{C} = (\mathcal{G}, \mathcal{A})$ et $\mathcal{C}' = (\mathcal{G}', \mathcal{A}')$. On définit l'union de $\mathcal{C}$ et $\mathcal{C}'$ par :

$$\mathcal{C} \bigcup \mathcal{C}' = min\{\mathcal{C}_i | \exists \mathfrak{a} \in \mathbb{A}^{\mathcal{C}'}_{\mathcal{C}}, \mathcal{C}_i = (\mathcal{G} \cup_{\mathfrak{a}} \mathcal{G}', \mathcal{A} \cap_{\mathfrak{a}} \mathcal{A}')\}$$

où l'intersection des deux fonctions de typage est une nouvelle fonction de typage : $\mathcal{A}_\mathfrak{a} = \mathcal{A} \cap_\mathfrak{a} \mathcal{A}' \Leftrightarrow \forall t \in D_\mathfrak{a}, \mathcal{A}_\mathfrak{a}(t_t^{\mathfrak{a}(t)}) = \mathcal{A}(t) = \mathcal{A}'(t)$.

4 Requêtes sur des séquences d'évènements

Les opérations définies ci-dessus vont servir à spécifier la recherche de chroniques correspondant à une requête sur une BDI. Soit deux ensembles P et N de séquences d'évènements et un seuil T_l minimal ou maximal pour chacune des séquences de P et N. On souhaite extraire, sous forme de chroniques, tous les phénomènes fréquents dans au moins une séquence d'évènements de l'ensemble P et non fréquents dans toutes les séquences de l'ensemble N. Une requête Rq a la forme générale :

$$Rq(P, N, T) = \Big(\exists l \in P, freq^Q(\mathcal{C}, l) \geq T_l\Big) \wedge \Big(\forall l \in N, freq^Q(\mathcal{C}, l) < T_l\Big) \qquad (1)$$

Deux approches du calcul des solutions de Rq sont possibles. La première consiste à considérer globalement la requête (Bucila et al. 2002) en utilisant subtilement chaque partie de la requête. La deuxième méthode consiste à calculer toutes les chroniques fréquentes de chaque séquence d'évènements séparément, à les mémoriser, puis à fusionner les résultats pour obtenir les solutions de la requête Rq. Pour une requête donnée, l'efficacité de la première méthode provient du fait que de nombreux calculs sont évités. Pour un ensemble de requêtes portant sur les mêmes séquences d'évènements et dont les seuils ont les mêmes valeurs mais sont indifféremment minimaux ou maximaux, la deuxième méthode est plus efficace car seule la fusion des résultats diffère d'une requête à l'autre. Nous avons privilégié l'aspect base de données et ainsi développé la deuxième méthode. Nous présentons tout d'abord une adaptation de l'algorithme de Mitchell qui sera utilisé par notre algorithme général afin de calculer les bornes d'espaces des versions puis nous introduisons la notion de chronique maximalement fréquente (CMF).

4.1 Calcul de l'espace des versions : algorithme de Mitchell

L'algorithme de Mitchell déjà utilisé pour calculer l'espace des versions de séquences de symboles (De Raedt et Kramer 2001) est étendu à la recherche de chroniques et présenté dans l'algorithme 1. Il calcule les bornes S et G de l'espace des versions correspondant aux solutions d'une conjonction de contraintes $c = c_0 \wedge c_1 \wedge \ldots \wedge c_n$. S et G peuvent être définies à partir des opérateur min et max (définition 10) et de l'ensemble des solutions de c : $S(c) = min(sol(c))$, $G(c) = max(sol(c))$.

Les contraintes admissibles sont de deux formes différentes : $\mathcal{C} \sqsubseteq \mathcal{C}'$ ou $\mathcal{C} \not\sqsubseteq \mathcal{C}'$ où on dispose de $\mathcal{C}'$ a priori et où $\mathcal{C}$ représente toute chronique solution. Pour les contraintes de forme $\mathcal{C} \sqsubseteq \mathcal{C}'$, l'opérateur utilisé (ligne 1 de l'algorithme 1) est l'union de chroniques (définition 15). Dans le cas des contraintes de forme $\mathcal{C} \not\sqsubseteq \mathcal{C}'$, on utilise l'opérateur osd :

Définition 16 (Opérateur de spécialisation dirigé)
Soit trois chroniques g, s et d, tel que $g \sqsubseteq s$, on définit l'opérateur

$$osd(g, s, d) = max\{\mathcal{C} | g \sqsubseteq \mathcal{C} \wedge \mathcal{C} \not\sqsubseteq d \wedge \mathcal{C} \sqsubseteq s\}$$

Comme $g \sqsubseteq s$, il existe un appariement complet $\mathfrak{a}$ entre g et s. En spécialisant g par ajout des variables temporelles de s qui ne sont pas appariés par $\mathfrak{a}$ (ainsi que

Algorithme 1: Algorithme de Mitchell

Entrées : $\mathcal{AM}$: une conjonction de contraintes
Sorties : $S = min(Sol(\mathcal{AM}))$ et $G : max(Sol(\mathcal{AM}))$
$S = \{\bot\}$ $G = \{\top\}$
pour chaque $AM_i \in \mathcal{AM}$ **faire**
  cas où AM_i *de forme* $\mathcal{C} \sqsubseteq \mathcal{C}'$
    $G = \{g \in G | g \sqsubseteq \mathcal{C}'\}$
1   $S = min\{c | c \in (\mathcal{C}' \cup s) \wedge \exists s \in S \wedge \exists g \in G : g \sqsubseteq c\}$
  cas où AM_i *de forme* $\mathcal{C} \not\sqsubseteq \mathcal{C}'$
    $S = \{s \in S | s \not\sqsubseteq \mathcal{C}'\}$
2   $G = max\{c | c \in osd(g, s, \mathcal{C}') \wedge \exists g \in G \wedge \exists s \in S\}$

les contraintes temporelles associées), on obtient un ensemble de chroniques duquel on extrait les chroniques qui ne sont pas plus générales que d. Les chroniques maximales de cet ensemble constituent le résultat de $osd(g, s, d)$. Cet opérateur remplace avantageusement l'opérateur $mgs(g, d)$ (De Raedt et Kramer 2001). En effet, l'extension de $mgs(g, d)$ aux chroniques est complexe car il retourne les chroniques maximales parmi les chroniques plus spécifiques que g mais pas plus générales que d. Or, on ne sait comment spécialiser g et l'ensemble des chroniques résultat peut être très grand.

L'opérateur *osd* impose $s \neq \bot$. En effet, si $s = \bot$, alors g peut être spécialisé de n'importe quelle façon et on revient dans le cas de *mgs*. Dans l'algorithme 1, $S = \{\bot\}$ si aucune contrainte de forme $\mathcal{C} \sqsubseteq \mathcal{C}'$ n'a été traitée avant une contrainte de forme $\mathcal{C} \not\sqsubseteq \mathcal{C}'$. On peut éviter ce cas car pour chaque espace des versions à calculer on dispose d'une contrainte de forme $\mathcal{C} \sqsubseteq \mathcal{C}'$ qu'il suffit de traiter avant toutes les autres contraintes.

4.2 Algorithme général

La requête (1) peut être réécrite de façon à utiliser l'algorithme de Mitchell. Soit $\text{Cmf}(l, T) = min\{\mathcal{C} : freq^Q(\mathcal{C}, l) \geq T\}$ l'ensemble des chroniques minimales et fréquentes (*CMFs*) dans une séquence d'évènements l selon un seuil de fréquence minimum T. Les sous-requêtes qui imposent un seuil minimum de fréquence sont des sous-requêtes anti-monotones donc toute solution de la requête (1) est plus générale qu'au moins une CMF d'une séquence d'évènements de l'ensemble P. Les sous-requêtes qui imposent un seuil maximum de fréquence sont des sous-requêtes monotones donc toute solution de la requête (1) n'est pas plus générale que chacune des CMFs des séquences d'évènements de l'ensemble N. Ainsi, on peut réécrire la requête (1) de la façon suivante :

$$Rq = \Big(\bigvee_{\mathcal{B} \in \text{Cmf}(P,T)} \mathcal{C} \sqsubseteq \mathcal{B} \Big) \wedge \Big(\bigwedge_{\overline{\mathcal{B}} \in \text{Cmf}(N,T)} \mathcal{C} \not\sqsubseteq \overline{\mathcal{B}} \Big) \text{ où } \text{Cmf}(E, T) = \bigcup_{l \in E} \text{Cmf}(l, T_l) \quad (2)$$

Le calcul des bornes de l'espace des versions par l'algorithme proposé ne peut s'effectuer qu'à partir d'une conjonction de contraintes. L'expression (2) est réécrite en une disjonction de conjonctions :

$$Rq = \bigvee_{\mathcal{B} \in \text{Cmf}(P,T)} \Big(\mathcal{C} \sqsubseteq \mathcal{B} \bigwedge_{\overline{\mathcal{B}} \in \text{Cmf}(N,T)} \mathcal{C} \not\sqsubseteq \overline{\mathcal{B}} \Big) \quad (3)$$

Algorithme 2: Algorithme général

Entrées : P, N : Ensembles de séquences d'évènements
T : Ensemble des seuils associés aux séquences d'évènements

Sorties : (S, G) : l'espace des versions correspondant aux solutions de (1)

$S = \emptyset$ $G = \emptyset$ $\mathcal{M} = \emptyset$

1 Calcul de Cmf(P, T) et Cmf(N, T)

pour chaque $\overline{\mathcal{B}} \in$ *Cmf*(N, T) **faire** $\mathcal{M} = \mathcal{M} \cup \{\mathcal{C} \not\sqsubseteq \overline{\mathcal{B}}\}$

pour chaque $\mathcal{B} \in$ *Cmf*(P, T) **faire**

$\{S_{\mathcal{B}}, G_{\mathcal{B}}\} = mitchell(\{\mathcal{C} \sqsubseteq \mathcal{B}\} \cup \mathcal{M})$

$S = min(S \cup S_{\mathcal{B}})$

$G = max(G \cup G_{\mathcal{B}})$

Pour chaque chronique appartenant à Cmf(P, T), on calcule les bornes d'un espace des versions d'une conjonction de contraintes à l'aide de l'algorithme de Mitchell. Ensuite, il suffit d'effectuer respectivement l'union des bornes S et G de ces espaces de versions pour obtenir les bornes de l'ensemble des solutions de la requête (3). L'algorithme 2 détaille ce procédé. L'un des principaux intérêts de cette méthode est qu'elle nécessite seulement le calcul de CMFs dans chaque séquence d'évènements (ligne 1 de l'algorithme 2). La partie suivante détaille cette recherche de CMFs en utilisant FACE.

4.3 Calcul des CMFs

La recherche des CMFs s'effectue grâce à une adaptation de l'outil de fouille de données FACE (Dousson et Duong 1999). La figure 5 illustre ce nouveau fonctionnement. La génération des chroniques candidates de taille n repose sur la notion de monotonie. Une chronique ne peut être fréquente que si toute chronique plus générale est fréquente. Or pour une chronique $\mathcal{C}$ de taille n, il existe n chroniques plus générales qui comportent $n-1$ variables de $\mathcal{C}$. Ainsi, à chaque étape d'apprentissage n, on génère les chroniques candidates à partir des chroniques fréquentes trouvées à l'étape précédente. De plus, pour tout couple de variables temporelles des chroniques générées, on impose que la contrainte temporelle associée soit incluse dans l'intervalle $[-W, W]$. La valeur W, fixée par l'utilisateur, représente la durée maximale d'une instance de chronique.

La phase de reconnaissance consiste à rechercher l'ensemble des instances de chaque

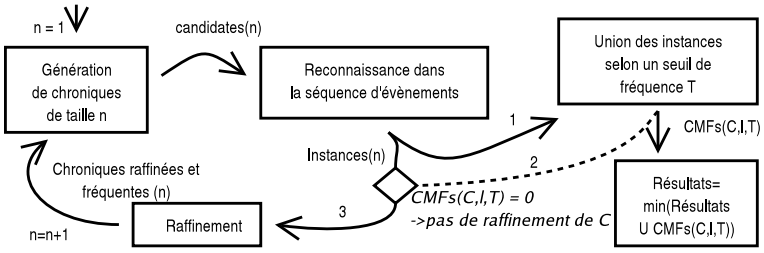


FIG. 5 – Algorithme de FACE

chronique. Ces instances ont deux utilités : elles servent de base à la génération des CMFs (arc 1 de la figure 5) et sont utilisées pour raffiner les chroniques de taille n (arc 3 de la figure 5). La génération de l'ensemble des CMFs, noté $\mathrm{Cmf}(\mathcal{C}, l, T)$, à partir des instances d'une chronique $\mathcal{C}$ dans une séquence d'évènements l par rapport à un seuil T consiste à regrouper ces instances en ensembles. Chacun de ces ensembles E est associé à une chronique $\mathcal{C}_E \in \mathrm{Cmf}(\mathcal{C}, l, T)$ qui est la plus spécifique possible et qui couvre l'ensemble des instances de E. Un ensemble E respecte les conditions suivantes : $\mathcal{C}_E \in \mathrm{Cmf}(\mathcal{C}, l, T) \Leftrightarrow |E| \geq T \wedge Q(E) \wedge \neg\exists \mathcal{C}' \in \mathrm{Cmf}(\mathcal{C}, l, T) : \mathcal{C}_E \sqsubset \mathcal{C}'$. Si aucun ensemble d'instances ne satisfait ces conditions alors la chronique reconnue est non fréquente et n'est pas raffinée (arc 2 de la figure 5). Le raffinement d'une chronique $\mathcal{C}$ consiste en la recherche de la chronique la plus spécialisée généralisant l'ensemble des instances de $\mathcal{C}$. On a prouvé dans (Vautier 2004) que l'algorithme présenté extrait l'ensemble correct et complet des CMFs d'une durée maximale W selon un critère de reconnaissance Q dans une séquence d'évènements l.

4.4 Illustration

Une BDI contient les séquences d'évènements l_1 et l_2 de la figure 6. On souhaite connaître les chroniques de fréquence supérieure à 5 dans l_1 et inférieure à 3 dans l_2. Pour cela, on pose la requête $freq^Q(\mathcal{C}, l_1) \geq 5 \wedge freq^Q(\mathcal{C}, l_2) < 3$ à la BDI. Le critère de reconnaissance Q utilisé est celui d'instances disjointes au plus tôt ($Q_{d\&t}$) et la durée des instances recherchées n'excède pas $W = 5$ $u.t.$

Soit la requête $Rq(P, N, T)$ tel que $P = \{l_1\}, N = \{l_2\}$ *et* $T = \{l_1 \rightarrow 5, l_2 \rightarrow 3\}$ posée à l'algorithme général 2. Celui-ci demande le calcul à FACE de $\mathrm{Cmf}(l_1, 5)$ et $\mathrm{Cmf}(l_2, 3)$ qui retourne les ensembles présentés dans la figure 6. À partir des cinq chroniques de $\mathrm{Cmf}(l_2, 3)$, l'algorithme 2 crée la contrainte $\mathcal{M} = \bigwedge_{\overline{\mathcal{B}} \in \mathrm{Cmf}(l_2,3)} \mathcal{C} \not\sqsubseteq \overline{\mathcal{B}}$. Pour chaque chronique $\mathcal{B}$ de $\mathrm{Cmf}(l_1, 5)$ il ajoute à $\mathcal{M}$ la contrainte $\mathcal{C} \sqsubseteq \mathcal{B}$ et calcule les bornes (S, G) de l'espace des versions grâce à l'algorithme 1. L'union de ces espaces des versions donne le résultat suivant :

$$S = \{ \quad A \xrightarrow{[-1,2]}_{[-2,-1]\,[1,4]} B \xrightarrow{} B, \quad A \xrightarrow{[-2,-2]} B \ \}, \ G = \{ \quad A \xrightarrow{[-1,2]}_{[-2,-1]\,[1,4]} B \xrightarrow{} B, \quad A \xrightarrow{[-\infty,-2]} B \ \}.$$

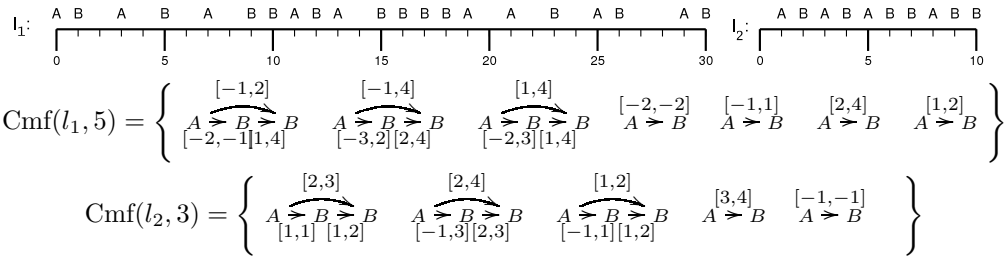


FIG. 6 – Les CMFs de deux séquences d'évènements l_1 et l_2

4.5 Expérimentations

Les premières expérimentations en matière de caractérisation de pathologies (arythmies) cardiaques démontrent l'intérêt des algorithmes proposés. La BDI contient des séquences d'évènements cardiaques élaborées à partir d'électrocardiogrammes (Carrault et al. 2003). Chacune de ces séquences est étiquetée par un expert en fonction de l'arythmie constatée sur l'ECG. On a recherché les chroniques fréquentes dans les séquences d'évènements associées à un type d'arythmie mais qui ne sont pas fréquentes dans les autres séquences. Les résultats sont prometteurs : la méthode décrite ici a permis de retrouver des chroniques correspondant à celles obtenues par une méthode d'apprentissage supervisé. Dans un proche avenir, la méthode sera évaluée de manière plus systématique sur des jeux de données plus complexes provenant du domaine des télécommunications.

5 Conclusion et perspectives

Nous avons présenté une méthode originale qui extrait l'information sous forme de motifs appelés chroniques prenant en compte des contraintes temporelles numériques sur des évènements. Ils permettent également l'expression de la séquentialité et du parallélisme entre évènements et généralisent les motifs temporels utilisés par Mannila et De Raedt (Mannila et al. 1997, Lee et De Raedt 2003), entre autres.

Nous avons étendu le concept de bases de données inductives à la fouille de données temporelles. Des chroniques, qui comportent des contraintes temporelles numériques explicites, sont extraites à partir d'une requête sur une BDI contenant des séquences d'évènements. La requête d'un utilisateur fixe les seuils de fréquence minimale ou maximale des chroniques devant être extraites de la BDI. Notre contribution introduit la notion de critère de reconnaissance qui généralise la façon dont la fréquence d'un motif est calculée sur des données temporelles. De plus, la recherche de solutions utilise une relation de généralité sur les chroniques qui permet la réutilisation et l'adaptation d'algorithmes de l'espace des versions pour gérer la dimension temporelle numérique. Ces algorithmes nécessitent le calcul préalable des chroniques maximales et fréquentes de chaque séquence d'évènements utilisée dans la requête. Ce calcul est effectué par une adaptation d'un outil de fouille de données.

Notre approche fournit l'ensemble correct et complet des solutions. Or, afin de traiter des gros volumes de données en un temps raisonnable, nombre d'outils de fouille de données fournissent une approximation de l'ensemble des solutions en sacrifiant la correction et/ou la complétude. Nous nous proposons, dans des travaux futurs, d'introduire la possibilité de moduler la complétude des résultats notamment par l'introduction d'algorithmes de clustering sur les instances de chroniques. Une telle méthode permettrait, en particulier, de rendre plus efficace la recherche de CMFs.

Références

Bucila C., Gehrke J., Kifer D. et White W. (2002). Dualminer : a dual-pruning algorithm for itemsets with constraints. Dans *Proc. of the eighth ACM SIGKDD Int. Conf. on Knowledge discovery and data mining*, pages 42–51. ACM Press.

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

De Raedt L. (2002). A perspective on inductive databases. *SIGKDD Explor. Newsl.*, 4(2) :69–77.

De Raedt L. et Kramer S. (2001). The levelwise version space algorithm and its application to molecular fragment finding. Dans *Proc. of IJCAI 2001*, pages 853–862, Seattle, USA. Morgan Kaufmann.

Dousson C. et Duong T. V. (1999). Discovering chronicles with numerical time constraints from alarm logs for monitoring dynamic systems. Dans *Proc. of IJCAI 1999*, pages 620–626.

Dousson C. et Ghallab M. (1994). Suivi et reconnaissance de chroniques. *Revue d'intelligence artificielle*, 8 :29–61.

Dzeroski S. (2002). Computational scientific discovery and inductive databases. Dans *International Workshop on Active Mining (AM-2002)*, pages 4–15, Maebashi City, Japan. IEEE Computer Society.

Imielinski T. et Mannila H. (1996). A database perspective on knowledge discovery. *Comm. of The ACM*, 39 :58–64.

Kramer S., De Raedt L. et Helma C. (2001). Molecular feature mining in HIV data. Dans *Proc. of the seventh ACM SIGKDD Int. Conf. on Knowledge discovery and data mining*, pages 136–143. ACM Press.

Lee S. D. et De Raedt L. (2003). *Database Support for Data Mining Applications*, volume 2682 de *LNCS*, chapitre Constraint Based Mining of First Order Sequences in SeqLog. Springer-Verlag.

Lin J., Keogh E., Lonardi S. et Patel P. (2002). Finding motifs in time series. Dans *Proceedings of the Second Workshop on Temporal Data Mining*, Edmonton, Canada.

Lin J.-L. (2003). Mining maximal frequent intervals. Dans *Proc. of the 2003 ACM symposium on Applied computing*, pages 426–431. ACM Press.

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

Mitchell T. M. (1997). *Machine Learning*. McGraw-Hill, New York.

Vautier A. (2004). Réconciliation fouille de données et programmation logique inductive. Rapport technique, Irisa.

Yoshida M., Iizuka T., Shiohara H. et Ishiguro M. (2000). Mining sequential patterns including time intervals. Dans *Proc. SPIE Vol. 4057, p. 213-220, Data Mining and Knowledge Discovery : Theory, Tools, and Technology II, Belur V. Dasarathy Ed.*

Summary

Inductive databases integrate databases with data mining. This paper proposes an inductive database extension to mine temporal patterns in events sequences. Temporal patterns discovered are event sets temporally constrained by numerical intervals. We proposed a generality relation defined on this kind of patterns and related adaptations to version space algorithms. These algorithms have been added to FACE, a data mining tool on stream data. First results show not only better discovered knowledge, but also upholding efficiency and effectiveness.