Modélisation et extraction de motifs optimaux

Willy Ugarte Rojas, Patrice Boizumault, Bruno Crémilleux, Samir Loudni

Modélisation et extraction de motifs optimaux

Willy Ugarte Patrice Boizumault Bruno Crémilleux Samir Loudni

GREYC (CNRS UMR 6072) – Université de Caen Basse-Normandie
Campus Côte de Nacre, 14032 Caen cedex 5 - France
{prénom.nom}@unicaen.fr

Résumé

Cet article introduit la notion de Motif Optimal (MO) selon une préférence définie par l'utilisateur. La notion de MO permet de modéliser de nombreux problèmes d'extraction de motifs : motifs libres/fermés/maximaux, skypatterns, top-k*, motifs pics, miki, sous-groupes, . . . Nous montrons comment extraire tous les MO avec une approche générique fondée sur les CSP dynamiques. Une étude expérimentale, menée sur plusieurs MO, compare les performances obtenues par rapport aux méthodes ad hoc.*

Mots Clef

Fouille de données, Extraction de motifs, CSP Dynamiques

Abstract

This paper introduces the notion of Optimal Patterns according to a user preference. Optimal patterns enable to model many pattern mining problems such as free/closed/maximal patterns, skypatterns, top-k*, peaks, miki, subgroups, . . . We propose a generic approach based on Dynamic CSP for mining optimal patterns. Finally, we perform an experimental study comparing our approach vs ad hoc methods on several MO.*

Keywords

Pattern mining, Dynamic CSP.

1 Introduction

L'extraction de motifs contraints est un champ majeur de la fouille de données. L'intérêt des motifs produits est garanti par le point de vue de l'analyste exprimé à travers la sémantique de la contrainte. De plus, la complétude des méthodes mises en œuvre assure qu'aucun motif estimé pertinent ne sera manqué. Cependant, le grand nombre de motifs généralement obtenus est un frein à leur utilisation. C'est pourquoi la communauté effectue d'importants efforts pour définir et produire des *ensembles de motifs* où l'intérêt d'un motif n'est pas uniquement lié au motif mais dépend aussi des autres motifs extraits. Malgré les difficultés d'extraction dues à la prise en compte de plusieurs motifs au lieu d'un seul, cette voie connaît actuellement un vif intérêt [1, 2] parce qu'elle permet de définir un résultat (i.e., un ensemble de motifs) exprimant un point de vue avec une certaine globalité sur les données qui est plus proche des intérêts de l'utilisateur. Notons que cette idée de comparer des motifs entre eux pour obtenir des motifs plus significatifs est déjà présente dans les meilleurs motifs selon un certain critère [3] ou les représentations condensées de motifs [4, 5].

Dans cet article, nous proposons la notion de motifs optimaux (MO), que nous définissons comme les meilleurs motifs selon une préférence donnée par l'utilisateur. Une contribution forte est que les MO englobent de nombreux problèmes de fouille de données. Nous proposons une méthode générique pour extraire les MO. Celle-ci repose sur les Problèmes de Satisfaction de Contraintes (CSP) dynamiques. L'idée majeure est la suivante : dès qu'une solution est trouvée, une contrainte est ajoutée dynamiquement au CSP courant afin de rechercher une solution de meilleure qualité et de réduire l'espace de recherche. Le processus s'arrête lorsqu'aucune meilleure solution ne peut être obtenue. En particulier, nous montrons que les MO sont caractérisés par une contrainte de base et un ensemble de contraintes ajoutées dynamiquement.

La section 2 définit la notion de MO. La section 3 présente notre approche générique extrayant les MO. La section 4 synthétise les travaux relatifs. La section 5 décrit l'étude expérimentale menée sur plusieurs MO et compare les performances obtenues par rapport aux méthodes ad hoc.

2 Motifs optimaux

2.1 Contexte

Nous adoptons les définitions classiques en fouille de données. $\mathcal{I}$ est un ensemble d'*items* binaires. Un motif est un sous-ensemble non-vide de $\mathcal{I}$ et est un élément du langage de motifs $\mathcal{L}_\mathcal{I} = 2^\mathcal{I} \backslash \emptyset$. Un jeu de données transactionnel $\mathcal{T}$ est un multi-ensemble de motifs de $\mathcal{L}_\mathcal{I}$. Une entrée de $\mathcal{T}$ est appelée transaction et est un élément de $\mathcal{L}_\mathcal{I}$.

Définition 1 (Couverture)**.** *La couverture d'un motif x est l'ensemble de transactions de $\mathcal{T}$ qui couvrent x : $T(x) = \{t \in \mathcal{T} \mid x \subseteq t\}$.*

Définition 2 (Fréquence)**.** *La fréquence d'un motif x est le nombre de transactions couvrant x : $freq(x) = \mid T(x) \mid$.*

Un motif x est fréquent ssi $freq(x) \geq \alpha$, pour un seuil α. La fréquence d'un motif x par rapport à une classe i est

$freq_i(x) =\mid \{t \mid t \in \mathcal{T}_i, x \subseteq t\} \mid$ où $\mathcal{T}_i$ est l'ensemble des transactions de la classe i.

Définition 3 (Taux de croissance)**.** *Le taux de croissance d'un motif x par rapport à une classe i est :*

$$gr_i(x) = \begin{cases} 0 & si\ freq_i(x) = 0 \\ \infty & si\ freq(x) = freq_i(x) \\ \frac{|\mathcal{T} \setminus \mathcal{T}_i| \times freq_i(x)}{|\mathcal{T}_i| \times (freq(x) - freq_i(x))} & sinon \end{cases}$$

Un motif x est émergent pour une classe i ssi $gr_i(x) \geq \gamma$, pour un seuil γ. En particulier, x est un *Jumping Emerging Pattern* (JEP) ssi $freq(x)=freq_i(x)$ i.e $gr_i(x) = \infty$.

Définition 4 (taille et k-itemset)**.** *La taille d'un motif est sa cardinalité. Un motif x est un k-itemset ssi $taille(x) = k$.*

2.2 Définition des MO

Cette section introduit la notion de motif optimal (MO). Les MO vont permettre de modéliser les problèmes de fouille dont les solutions sont optimales pour une préférence donnée.

Définition 5 (Préférence)**.** *Une **préférence** $\rhd$ est une relation d'ordre partiel sur $\mathcal{L}_\mathcal{I}$. Soient x et y deux motifs, $x \rhd y$ indique que x est préféré à y.*

Exemples :

1. Soit m une **mesure**, $x \rhd y = m(x) > m(y)$.
2. **Dominance :** soit M un ensemble de mesures, un motif x domine un autre motif y par rapport à M (noté par $x \succ_M y$) ssi $\forall m \in M, m(x) \geq m(y)$ et $\exists m' \in M, m'(x) > m'(y)$. Ainsi, $x \rhd y = x \succ_M y$.

Définition 6 (Motif optimal)**.** *Soit $\mathcal{T}$ un jeu de données, et $\rhd$ une préférence. Un motif x est optimal (selon $\rhd$) ssi :* $\nexists y_1, \ldots y_p, \in \mathcal{L}_\mathcal{I}, \forall 1 \leq j \leq p,\ y_j \rhd x$.

Ainsi, x est optimal s'il n'existe pas p motifs qui soient préférés à x. Un ensemble de MO est défini comme suit :

$$\{x \in \mathcal{L}_\mathcal{I} \mid \ \texttt{élémentaire}(x) \wedge \nexists y_1, \ldots y_p, \in \mathcal{L}_\mathcal{I}, \forall 1 \leq j \leq p,\ y_j \rhd x\}$$

La contrainte $\texttt{élémentaire}(x)$ permet de spécifier que le motif x doit satisfaire une propriété de base. Elle peut être vide, mais dans la pratique, elle s'avère très utile. En effet, de nombreux problèmes de fouille imposent que les motifs recherchés aient une propriété particulière : avoir une fréquence minimale, être un fermé, ...
Cette définition, qui peut apparaître simple, est puissante pour une double raison. D'une part, elle est générale et permet de modéliser de nombreux problèmes de fouille (sa force réside dans les comparaisons multiples autorisées entre x et les y_i). D'autre part, elle est propice à une méthode efficace d'extraction fondée sur les CSP dynamiques (cf. section 3).

2.3 Exemples de MO

Nous présentons succinctement plusieurs problèmes de fouille connus et montrons qu'il s'agit de MO. Par manque de place, nous ne détaillons par l'intérêt de ces problèmes, et renvoyons le lecteur aux articles les introduisant.

2.3.1 Motifs fermés [4], libres [5] et maximaux. Ces motifs forment une couverture (approximative pour les maximaux) du jeu de données. Si on prend comme exemple la fréquence, alors les motifs fermés sont les motifs x dont aucune spécialisation $y \supset x$ a une fréquence égale à celle de x. Ils s'expriment sous forme de MO comme suit, où α est un seuil (éventuellement nul) de fréquence :

$$\{x \in \mathcal{L}_\mathcal{I} \mid \ freq(x) \geq \alpha \ \wedge \nexists y \in \mathcal{L}_\mathcal{I} : y \supset x \wedge freq(y) = freq(x)\}$$

Les motifs libres et maximaux s'expriment selon de simples variantes de cette définition :

$$\text{motifs libres} \quad \{x \in \mathcal{L}_\mathcal{I} \mid \ freq(x) \geq \alpha \wedge \nexists y \in \mathcal{L}_\mathcal{I} : y \subset x \wedge freq(y) = freq(x)\}$$

$$\text{motifs maximaux} \quad \{x \in \mathcal{L}_\mathcal{I} \mid \ freq(x) \geq \alpha \wedge \nexists y \in \mathcal{L}_\mathcal{I} : y \supset x \wedge freq(y) \geq \alpha\}$$

2.3.2 Skypatterns [6, 7]. Soit M un ensemble de mesures. Un skypattern est un motif non-dominé par rapport à M :

$$\{x \in \mathcal{L}_\mathcal{I} \mid \texttt{fermé}_M(x) \wedge \nexists y \in \mathcal{L}_\mathcal{I} : y \succ_M x\}$$

La contrainte $\texttt{fermé}_M(x)$ impose que x soit un motif fermé pour M pour éviter les skypatterns redondants.

2.3.3 Motifs pics [8]. Soient $d(x, y) = |x \setminus y| + |y \setminus x|$ une distance, m une mesure, ρ un entier et δ un réel. Un motif pic possède une valeur selon m jugée grande par rapport à celles de ses voisins (au sens de d), alors :

$$\{x \in \mathcal{L}_\mathcal{I} \mid \ freq(x) \geq 1 \wedge \nexists y \in \mathcal{L}_\mathcal{I} : d(x, y) \leq \delta \wedge \rho \times m(y) > m(x)\}$$

2.3.4 top-k [8]. Soit m une mesure, et k un entier. top-k est l'ensemble des k meilleurs motifs selon m :

$$\{x \in \mathcal{L}_\mathcal{I} \mid \ freq(x) \geq 1 \wedge \nexists y_1, \ldots, y_k \in \mathcal{L}_\mathcal{I} : \forall 1 \leq j \leq k, m(y_j) > m(x)\ \}$$

2.3.5 The N-most interesting k-itemsets [9]. Soit N un entier. N-*most* est l'ensemble des N plus fréquents k-itemsets :

$$\{x \in \mathcal{L}_\mathcal{I} \mid \ taille(x) = k \wedge \nexists y_1, \ldots, y_N \in \mathcal{L}_\mathcal{I} : \forall 1 \leq j \leq N, freq(y_j) > freq(x)\ \}$$

2.3.6 Découverte de sous-groupes pertinents [10]. Un sous-groupe pertinent rassemble des motifs qui discriminent T_1 de T_2, où T_1 de T_2 sont 2 classes formant une partition de $\mathcal{T}$ (e.g., positive et négative) :

$$\{x \in \mathcal{L}_\mathcal{I} \mid \ freq_1(x) \geq \alpha \wedge \nexists y \in \mathcal{L}_\mathcal{I} : \ T_1(y) \supseteq T_1(x) \wedge T_2(y) \subseteq T_2(x) \wedge (T(y) = T(x) \Rightarrow y \subset x)\}$$

2.3.7 Pattern compression problem [11]. Soit $d(x,y) = 1 - \frac{|T(x)\cap T(y)|}{|T(x)\cup T(y)|}$ une distance, et δ un seuil. Un motif x est *représentatif* ssi aucun motif y dans son voisinage ($d(x,y) \leq \delta$) n'est inclus dans x. Le problème de compression de motifs consiste à trouver l'ensemble des motifs représentatifs :

$$\{x \in \mathcal{L}_\mathcal{I} \mid \texttt{fermé}(x) \wedge \nexists y \in \mathcal{L}_\mathcal{I} : d(x,y) \leq \delta \wedge y \not\subset x\}$$

2.3.8 Maximally informative k-itemset [12]. Soit x un k-itemset et B=$\{b_1, \ldots, b_k\}$ un tuple de k valeurs binaires. L'entropie conjointe de x est définie par :

$$H(x) = -\sum_{B\in\{0,1\}^k} p(x=B)\log p(x=B)$$
où $p(x=B)$ est la probabilité jointe de$(x=B)$

Soit x un k-itemset, x est un maximally informative k-itemset (*miki*) ssi aucun k-itemset y a une plus grande entropie conjointe ($H(y) > H(x)$). Un miki est un motif de la taille spécifiée qui maximise l'entropie conjointe :

$$\{x \in \mathcal{L}_\mathcal{I} \mid taille(x) = k \wedge \nexists y \in \mathcal{L}_\mathcal{I} : H(y) > H(x)\}$$

2.3.9 Optimal Risk Patterns [13]. Le risque relatif d'un motif x est défini par $RR(x) = \frac{freq_1(x)(|\mathcal{T}|-freq(x))}{(|\mathcal{T}|-freq_1(x))freq(x)}$ Soit π un seuil. Un motif fréquent x est un *risk pattern* ssi $RR(x) \geq \pi$, et x est un *optimal risk pattern* ssi il n'existe aucun motif plus petit qui ait un risque relatif plus grand :

$$\{x \in \mathcal{L}_\mathcal{I} \mid \quad freq(x) \geq \alpha \wedge RR(x) \geq \pi \wedge \nexists y \in \mathcal{L}_\mathcal{I} : y \subset x \wedge RR(y) \geq RR(x)\}$$

Enfin, d'autres problèmes d'ensembles de motifs que nous ne détaillons pas ici (e.g., les Strong Emerging Patterns [14] et les Essential Jumping Emerging Patterns [15]) sont aussi des MO.

3 DCSP pour l'extraction des MO

Cette section montre comment l'extraction des MO peut être modélisée et résolue à l'aide des CSP dynamiques [16]. L'idée majeure est, qu'à chaque étape, dès qu'une solution est trouvée, une nouvelle contrainte est ajoutée dynamiquement afin de rechercher une meilleure solution (au sens de $\rhd$). Le processus s'arrête lorsque aucune meilleure solution ne peut plus être obtenue. Notre approche est *générique* et peut donc s'appliquer à l'extraction de n'importe quel MO. Notre approche est *complète* car le solveur utilisé est complet.

La section 3.1 rappelle brièvement la notion de CSP dynamique. La section 3.2 montre comment l'extraction d'un MO peut être modélisée et résolue à l'aide d'un CSP dynamique. La section 3.3 décrit plus particulièrement trois exemples.

3.1 CSP dynamiques

Un CSP $P = (X, D, C)$ est défini par :
- un ensemble de variables X,
- un ensemble de domaines D, qui à chaque variable $x\in X$ associe un ensemble fini de valeurs $D(x)$,
- un ensemble de contraintes C.

Un CSP dynamique (DCSP) [16] est une séquence $P_1, P_2, ..., P_n$ de CSP, où chaque CSP P_i résulte de changements apportés au précédent P_{i-1}. Ces changements peuvent affecter les variables, les domaines et les contraintes. Pour notre approche, les seuls changements effectués sont l'ajout de nouvelles contraintes. Chaque fois qu'une solution du CSP courant est obtenue, de nouvelles contraintes sont postées dynamiquement. Ces contraintes persistent lors du retour arrière afin que les nouvelles solutions satisfassent à la fois l'ensemble initial de contraintes ainsi que celles ajoutées.

3.2 Extraction des MO à l'aide des DCSP

Cette section montre comment l'extraction des MO (selon une préférence $\rhd$) peut être modélisée et résolue à l'aide d'un CSP dynamique. Considérons la séquence $P_1, P_2, ..., P_n$ de CSP où chaque $P_i = (\{x\}, \mathcal{L}_\mathcal{I}, q_i(x))$ et
- $q_1(x) = \texttt{élémentaire}(x)$,
- $q_{i+1}(x) = q_i(x) \wedge \phi(s_i, x)$ où s_i est la première solution à la requête $q_i(x)$.

Les contraintes $\phi(s_i, x)$ imposent successivement que tous les motifs s_i obtenus ne soient pas meilleurs (au sens de la préférence $\rhd$) que le motif x (si un s_i était meilleur que x, x ne pourrait pas être un MO). Ainsi, à l'étape $(i+1)$, la contrainte suivante $\phi(s_i, x)$ sera ajoutée :

$$\phi(s_i, x) = \neg(s_i \rhd x)$$

En conséquence, aucun des motifs s_1, s_2, ..., s_i obtenus ne peut être meilleur que x (preuve immédiate par induction). Les nouvelles contraintes $\phi(s_i, x)$ ajoutées dynamiquement permettent de réduire l'espace de recherche. L'extraction s'arrête lorsque aucun meilleur motif ne peut être obtenu, i.e. il existe n tel que $q_{n+1}(x)$ n'a pas de solution. Cependant, tous les motifs extraits s_1, s_2, ..., s_n ne sont pas nécessairement optimaux selon $\rhd$. Certains peuvent être des motifs *intermédiaires*, i.e. utiles uniquement pour améliorer l'élagage de l'espace de recherche. Ces derniers (i.e. les motifs s_i pour lesquels il existe s_j ($1\leq i<j\leq n$) tel que $s_j \rhd s_i$) sont retirés en post-traitement. Ainsi, l'extraction s'effectue en deux étapes :

1. Calculer l'ensemble $\{s_1, s_2, \ldots, s_n\}$ des candidats à l'aide d'un CSP dynamique,
2. Supprimer tous les s_i qui sont des motifs *intermédiaires* (ils ne peuvent être des MO).

3.3 Exemples

Cette section présente plus particulièrement trois exemples d'extraction de MO utilisant les CSP dynamiques. Les trois MO choisis figurent parmi ceux décrits à la section 2.3 et nous semblent significatifs de l'ensemble. Chacun de ces trois exemples est traité en lui associant une contrainte de base $\texttt{élémentaire}(x)$ et en définissant les contraintes $\phi(s_i, x)$ ajoutées dynamiquement.

MO	élémentaire(x)	$\phi(s_i, x)$
Motifs fermés	$freq(x) \geq \alpha$	$(s_i \not\supset x) \vee (freq(s_i) \neq freq(x))$
Motifs maximaux	$freq(x) \geq \alpha$	$s_i \not\supset x$
Motifs libres	$freq(x) \geq \alpha$	$(s_i \not\subset x) \vee (freq(s_i) \neq freq(x))$
Skypatterns	fermé$_M(x)$	$\left(\bigvee_{m\in M} m(s_i) < m(x)\right) \vee \left(\bigwedge_{m\in M} m(s_i) = m(x)\right)$
Motifs pics	$freq(x) \geq 1$	$(d(x, s_i) > \delta) \vee (\rho \times m(s_i) \leq m(x))$
top-k	$freq(x) \geq 1$	$m(x) \geq \min_{s_j\in S} m(s_j)$ if $i \geq k$ $vrai$ sinon
The N-most interesting k-itemsets	$taille(x) = k$	$freq(x) \geq \min_{s_j\in S} freq(s_j)$ if $i \geq N$ $vrai$ sinon
Découverte de sous-groupes relevants	$freq(x) \geq \alpha$	$T_1(x) \not\subseteq T_1(s_i) \vee$ $T_2(x) \not\supseteq T_2(s_i) \vee$ $(T(s_i) = T(x) \wedge s_i \not\subset x)$
Pattern compression problem	fermé(x)	$(d(x, s_i) > \delta) \vee (s_i \subset x)$
Maximally informative k-itemset	$taille(x) = k$	$H(s_i) \leq H(x)$
Optimal Risk Patterns	$(freq(x) \geq \alpha) \wedge (RR(x) \geq \pi)$	$(s_i \not\subset x) \vee (RR(s_i) < RR(x))$

TABLE 1 – Contrainte élémentaire et contraintes ajoutées dynamiquement pour chaque MO.

Les motifs fermés. Soit α un seuil de fréquence et x le motif inconnu. Alors, les contraintes associées sont :
– élémentaire$(x) = freq(x) \geq \alpha$,
– $\phi(s_i, x) = \neg(s_i \rhd x) = \neg(s_i \supset x \wedge freq(s_i){=}freq(x))$
soit encore $\phi(s_i, x) \equiv (s_i \not\supset x) \vee (freq(s_i) \neq freq(x))$.
Sur cet exemple, les fermés sont définis suivant la mesure de fréquence. Pour rechercher les fermés pour une mesure m quelconque, il suffit de remplacer $freq$ par m.

Les skypatterns. Soit M un ensemble de mesures et x le motif inconnu. Alors, les contraintes associées sont :
– élémentaire$(x) =$ fermé$_M(x)$,
– $\phi(s_i, x) = s_i \not\succ_M x$ où s_i est la première solution de la requête $q_i(x)$.
La contrainte fermé$_M(x)$ impose que x soit un motif fermé pour M (voir section 2.3.2).
La contrainte $\phi(s_i, x){=}(s_i \not\succ_M x)$ impose que le motif suivant recherché x ne soit pas dominé par s_i : $\phi(s_i, x){=}(\bigvee_{m\in M} m(s_i) < m(x)) \vee (\bigwedge_{m\in M} m(s_i) = m(x))$
Bien que le nombre de candidats puisse être potentiellement très grand, il reste de taille raisonnable dans la pratique comme le montrent les expérimentations menées à la section 5.

Les top-**k.** Soit m une mesure d'intérêt. L'ensemble des top-k (voir section 2.3.4) peut aussi se formuler par :

$$\{x \in \mathcal{L}_\mathcal{I} \mid \quad freq(x) \geq 1 \wedge \nexists y_1, \ldots, y_k \in \mathcal{L}_\mathcal{I} : \min_{1\leq j\leq k} m(y_j) > m(x)\}$$

Chaque solution trouvée s_i est stockée dans une liste S. Au départ, le premier motif x tel que $freq(x) \geq 1$ est recherché. Tant que $(i < k)$, le nombre de motifs recherchés n'est pas encore atteint. Il est alors trop tôt pour contraindre la recherche et en conséquence $\phi(s_i, x){=}vrai$. Dès que la k-ème solution est obtenue, nous pouvons imposer que le nouveau motif x recherché soit meilleur (selon m) qu'au moins un des k meilleurs motifs déjà obtenus en postant la contrainte $\phi(s_i, x)$ et en retirant la solution avec la plus petite valeur selon m i.e $S \leftarrow S \setminus \{\arg\min_{s_j\in S} m(s_j)\}$.
D'où $\phi(s_i, x)$ est définie par :

$$\phi(s_i, x) = \begin{cases} m(x) \geq \min_{s_j\in S} m(s_j) & if \quad i \geq k \\ vrai & sinon \end{cases}$$

Autres exemples. Tous les MO présentés à la section 2.3 peuvent être extraits par notre approche. La table 1 indique, pour chaque MO, les contraintes élémentaire(x) et $\phi(s_i, x)$ qui lui sont associées. Pour un MO défini par la préférence $\rhd$, les contraintes $\phi(s_i, x)$ traduisent la relation $\neg(s_i \rhd x)$. Les CSP dynamiques fournissent ainsi une approche générique capable d'extraire n'importe quel type de MO.

4 Travaux relatifs

Approximer et Pousser. Dans [8], l'idée d'une approche générique pour traiter des problèmes d'extraction de motifs a été introduite, mais l'implémentation repose sur une contrainte locale qui est poussée, limitant les exemples pouvant être traités. Dans [17], une algèbre fondée sur la notion de dominance **binaire** a été définie, mais celle-ci ne peut pas être appliquée aux cas n-aire (e.g., top-k, N-most).
Programmation par contraintes. Notre proposition bénéficie des progrès récents sur la fertilisation croisée entre la fouille de données et la PPC [18, 19, 20]. Le point commun de toutes ces méthodes est de modéliser de manière déclarative l'extraction de motifs comme un CSP, dont la résolution fournit un ensemble complet de solutions satisfaisant toutes les contraintes.

5 Expérimentations

Nos expérimentations portent sur trois types de MO : les fermés/fréquents/maximaux, les skypatterns et les motifs

pics. Pour chacun d'eux, nous comparons les performances de notre approche générique (PPC+MO) avec une ou plusieurs méthodes dédiées. Les jeux de donnés utilisés sont ceux de l'UCI [1]. Toutes les expérimentations ont été réalisées sous Linux avec un processeur Core i3 à 2,13 GHz et une mémoire vive de 4 Go. PPC+MO a été développé en Gecode [2] en étendant l'extracteur de motifs (basé sur les CSP) développé par [19].

Motifs fermés, libres et maximaux. Nous comparons PPC+MO avec les *meilleurs* extracteurs : Eclat [21] et LCM [22] (uniquement pour les motifs fermés). Les expérimentations ont été menées en faisant varier le seuil de fréquence min_{freq}. La figure 1 montre que les méthodes ad hoc pour ces motifs (Eclat et LCM) sont plus performantes que notre approche générique. Ce résultat était attendu tellement les chercheurs ont fourni d'efforts depuis 15 ans sur l'extraction de représentations condensées. Malgré tout, ce résultat montre la faisabilité de notre méthode.

Skypatterns. Nous comparons PPC+MO avec Aetheris [6] qui est la seule approche ad hoc extrayant les skypatterns. Aetheris calcule une représentation condensée faite de motifs fermés par rapport à l'ensemble de mesures considéré, puis filtre ceux-ci pour obtenir les skypatterns. Nos expérimentations portent sur 23 jeux de données de l'UCI (voir la colonne gauche de la table 2) pour lesquels nous avons considéré les mesures M={$freq$, max, $area$, $mean$, gr_i}. Les skypatterns sont extraits suivant 6 différents ensembles de mesures : les 5 sous-ensembles de 4 mesures issus de M notés $M_1 \ldots M_5$ et $M_6 = M$. Pour $mean$, des valeurs d'attributs ont été générées aléatoirement dans l'intervalle $[0..1]$. Pour chaque méthode, les temps CPU reportés incluent l'ensemble du processus d'extraction.

La table 2 indique, pour chaque jeu de données et pour chaque ensemble de mesures : le nombre de skypatterns, pour PPC+MO : le nombre de candidats et le temps CPU associé, pour Aetheris : le nombre de motifs fermés et le temps CPU associé. Pour 16 jeux de données (sur les 23), les temps de calcul sont très faibles (moins de 30 secondes) et similaires, quel que soit l'ensemble de mesures.

La figure 2 montre le nuage de points de temps CPU pour les 7 jeux de données restants. Chaque point correspond à l'extraction des skypatterns pour un jeu de données et un ensemble de mesures : sa coordonnée x est le temps CPU pris par PPC+MO et sa cordonnée y celui pris par Aetheris. Les 2 échelles sont logarithmiques.

1. http://www.ics.uci.edu/~mlearn/MLRepository.html
2. http://www.gecode.org/

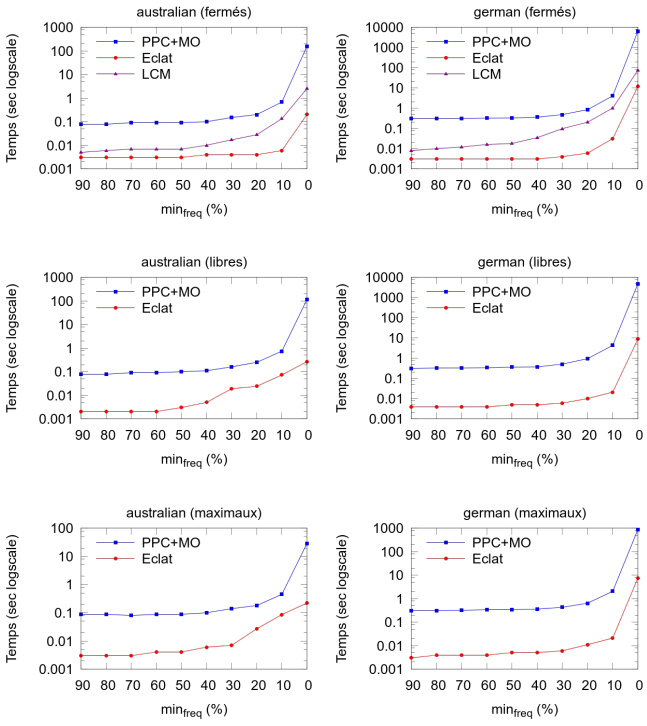

FIGURE 1 – Comparaison des temps CPU (motifs fermés, libres et maximaux).

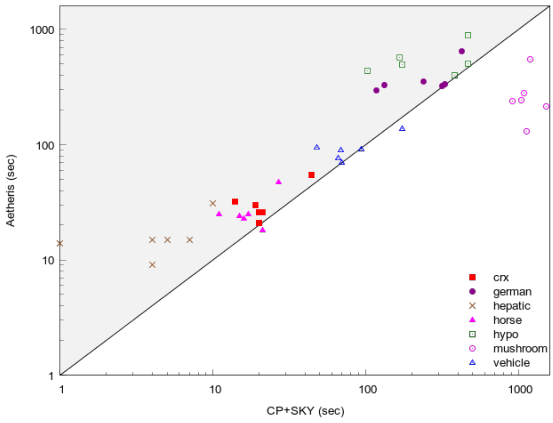

FIGURE 2 – Comparaison des temps CPU (skypatterns).

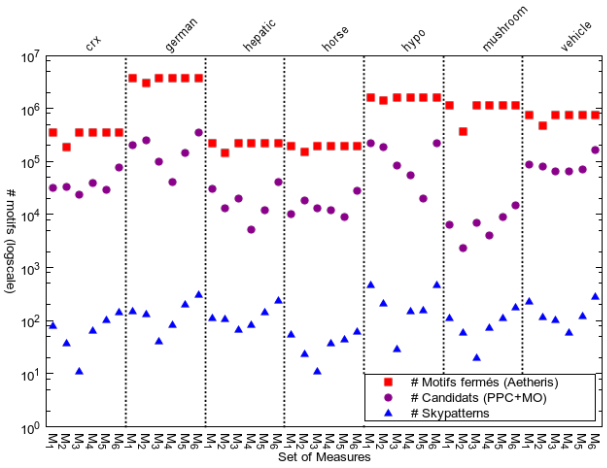

FIGURE 3 – Etude sur les 7 jeux de données retenus et les 6 ensembles de mesures considérés (skypatterns).

Jeu de données				$M_6 = \{freq, max, area, mean, gr_i\}$					Moyennes sur $\{M_1,M_2,M_3,M_4,M_5\}$				
					PPC+MO		Aetheris			PPC+MO		Aetheris	
	# items	# transactions	densité	# de Skypatterns	# de Candidats	Temps (sec)	# de motifs fermés	Temps (sec)	# de Skypatterns	# de Candidats	Temps (sec)	# de motifs fermés	Temps (sec)
abalone	28	4,178	0.321	76	5,255	25	9,947	**1**	43.20	3,393.20	6	9,808.80	**1**
anneal	68	798	0.195	187	13,903	12	35,152	**3**	80.60	6,748.20	6	30,606.00	**1**
austral	55	690	0.272	172	49,379	24	243,156	**20**	77.00	19,626.80	18	228,115.20	13
breast	43	286	0.231	38	2,311	**1**	7,721	**1**	23.60	1,374.20	**1**	7,369.20	**1**
cleve	43	303	0.325	97	19,370	**8**	77,203	**8**	51.80	11,215.80	5	74,670.80	**4**
cmc	28	1,474	0.357	62	12,760	12	25,649	**1**	36.00	7,702.80	9	25,408.60	**1**
crx	59	690	0.269	143	78,327	**44**	349,721	55	59.40	31,482.80	**19**	317,090.20	27
german	76	1,000	0.276	308	347,957	**426**	3,662,911	652	121.80	148,255.40	**249**	3,525,072.40	305
glass	34	216	0.295	52	2,633	**1**	7,165	**1**	31.80	1,587.80	**1**	6,920.80	**1**
heart	38	270	0.368	154	16,960	**6**	72,618	8	73.60	9,118.00	**4**	70,124.20	**4**
hepatic	45	155	0.421	237	41,096	**10**	222,333	31	103.20	16,156.20	**4**	206,742.00	13
horse	75	300	0.235	63	28,275	**27**	191,177	47	34.20	12,609.60	**16**	182,982.20	23
hypo	47	3,163	0.389	478	221,032	**469**	1,604,864	893	204.80	145,359.40	**303**	1,565,797.20	481
iris	15	151	0.333	7	86	**1**	93	**1**	5.80	72.40	**1**	91.80	**1**
lymph	59	142	0.322	161	26,200	**7**	116,030	26	64.00	12,532.40	**4**	105,260.00	11
mushroom	119	8,124	0.193	176	14,599	1,186	1,153,229	**548**	75.40	5,767.60	1137	995,808.40	**221**
new-thyroid	21	216	0.287	15	200	**1**	288	**1**	12.00	151.80	**1**	285.00	**1**
page	35	941	0.314	92	4,482	8	21,121	**2**	49.00	2,576.00	6	20,207.80	**1**
pima	26	768	0.346	53	1,400	3	12,559	**1**	32.40	943.80	3	12,439.40	**1**
tic-tac-toe	29	259	0.344	82	11,206	11	43,318	**3**	51.20	7,867.40	9	42,902.20	**2**
vehicle	58	846	0.327	280	164,152	172	745,353	**138**	126.20	73,390.60	**69**	689,937.80	84
wine	45	179	0.311	43	7,780	**2**	36,671	4	25.40	4,538.60	**2**	34,397.00	**2**
zoo	43	102	0.394	56	4,654	**1**	14,431	**1**	34.20	2,642.60	**1**	12,851.20	**1**

TABLE 2 – Comparaison détaillée sur les 23 jeux de données de l'UCI (skypatterns).

La figure 2 montre que PPC+MO surpasse Aetheris, presque tous les points étant dans la partie grise : la seule exception est mushroom. Ce jeu de données a certes un nombre de fermés proche de celui des autres, mais mushroom est un jeu de données de plus grande taille que les autres (notamment en nombre d'items). Or, la taille de l'espace de recherche pour extraire les skypatterns est en $O(2^{|\mathcal{I}|})$ (cf [7]). Ainsi, le nombre faible de fermés pour mushroom (au regard de sa taille) favorise Aetheris.
La figure 3 compare, pour chacun des 7 jeux de données et pour chacun des 6 ensembles de mesures, le nombre de motifs fermés calculés par Aetheris avec le nombre de candidats générés par PPC+MO. Ces 2 valeurs sont rapportées au nombre total de skypatterns. Le nombre de candidats demeure faible (milliers) par rapport au nombre important de motifs fermés (millions) produits par Aetheris. Enfin, le nombre de skypatterns (qui est le même quelle que soit l'approche) demeure peu élevé (centaines).

Motifs pics. Nous comparons PPC+MO avec QeCode [23], qui est la seule méthode connue pour extraire les motifs pics. Les expérimentations ont été menées en faisant varier le seuil de fréquence min_{freq}, pour différentes valeurs de ρ. La figure 4 montre que PPC+MO surpasse nettement Qecode : l'évolution des temps CPU semble quasi-linéaire pour PPC+MO alors qu'elle est clairement exponentielle pour Qecode.

Synthèse. Notre méthode générique est moins performante pour les motifs "simples" dont l'extraction a été profondément étudiée (e.g., fermés). En revanche, pour l'extraction d'ensembles de motifs plus élaborés comme les skypatterns, notre approche s'avère aussi compétitive voire plus performante que les méthodes ad hoc existantes (cf. l'exemple des motifs pics).

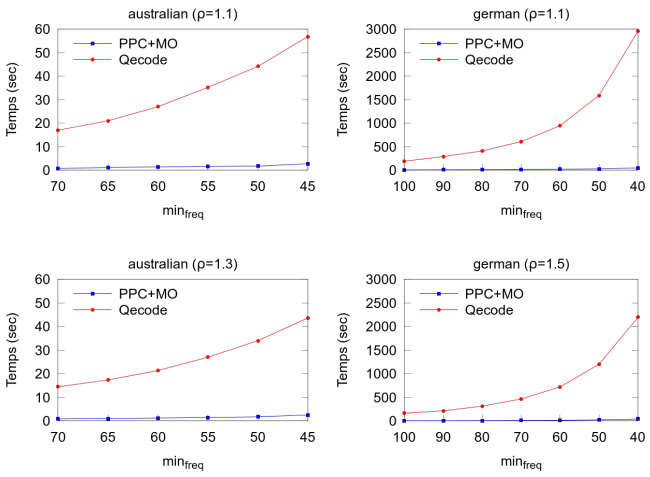

FIGURE 4 – Comparaison des temps CPU (motifs pics).

6 Conclusion

Dans cet article, nous avons introduit la notion de MO et nous avons montré que les MO permettaient de modéliser de nombreux problèmes de fouille de motifs : skypatterns, top-k, motifs fermés,... La résolution, à l'aide des

CSP dynamiques, rend notre approche générique. Les comparaisons effectuées avec les méthodes ad hoc ont montré qu'elle était tout à fait concurrentielle. La déclarativité de notre approche ouvre la voie à la définition et la découverte de nouveaux ensembles de motifs.

Remerciements

Ce travail a été soutenu par l'Agence Nationale de la Recherche, projets Ficolofo ANR-10-BLA-0214 et Hybride ANR-11-BS02-002.

Références

[1] Knobbe, A., Crémilleux, B., Fürnkranz, J., Scholz, M. : From local patterns to global models : The lego approach to data mining. In : LeGo Workshop co-located with ECML/PKDD'08. (2008)

[2] Raedt, L.D., Zimmermann, A. : Constraint-based pattern set mining. In : SDM, SIAM (2007)

[3] Fu, A.W.C., w. Kwong, R.W., Tang, J. : Mining *n*-most interesting itemsets. In : ISMIS. (2000)

[4] Pasquier, N., Bastide, Y., Taouil, R., Lakhal, L. : Discovering frequent closed itemsets for association rules. In : ICDT. (1999)

[5] Boulicaut, J.F., Bykowski, A., Rigotti, C. : Free-sets : a condensed representation of boolean data for the approximation of frequency queries. Data Mining and Knowledge Discovery (2003)

[6] Soulet, A., Raïssi, C., Plantevit, M., Crémilleux, B. : Mining dominant patterns in the sky. In : ICDM. (2011)

[7] Ugarte, W., Boizumault, P., Loudni, S., Crémilleux, B., Lepailleur, A. : Mining (soft-) skypatterns using dynamic CSP. In : CPAIOR'14. Volume 8451 of LNCS. (2014) 71–87

[8] Crémilleux, B., Soulet, A. : Discovering knowledge from local patterns with global constraints. In : ICCSA. (2008)

[9] Cheung, Y.L., Fu, A.W.C. : Mining frequent itemsets without support threshold : With and without item constraints. TKDE **16** (2004)

[10] Novak, P.K., Lavrac, N., Webb, G.I. : Supervised descriptive rule discovery : A unifying survey of contrast set, emerging pattern and subgroup mining. Journal of Machine Learning Research **10** (2009)

[11] Xin, D., Han, J., Yan, X., Cheng, H. : Mining compressed frequent-pattern sets. In : VLDB. (2005)

[12] Knobbe, A.J., Ho, E.K.Y. : Maximally informative k-itemsets and their efficient discovery. In : KDD. (2006)

[13] Li, J., Fu, A.W.C., He, H., Chen, J., Jin, H., McAullay, D., Williams, G.J., Sparks, R., Kelman, C. : Mining risk patterns in medical data. In : KDD. (2005)

[14] Soulet, A., Crémilleux, B., Rioult, F. : Condensed representation of emerging patterns. In : PAKDD. (2004)

[15] Fan, H., Ramamohanarao, K. : An efficient single-scan algorithm for mining essential jumping emerging patterns for classification. In : PAKDD. (2002)

[16] Verfaillie, G., Jussien, N. : Constraint solving in uncertain and dynamic environments : A survey. Constraints **10**(3) (2005)

[17] Negrevergne, B., Dries, A., Guns, T., Nijssen, S. : Dominance programming for itemset mining. In : ICDM. (2013)

[18] Raedt, L.D., Guns, T., Nijssen, S. : Constraint programming for itemset mining. In : KDD. (2008)

[19] Khiari, M., Boizumault, P., Crémilleux, B. : Constraint programming for mining n-ary patterns. In : CP. (2010)

[20] Guns, T., Nijssen, S., Raedt, L.D. : Itemset mining : A constraint programming perspective. AIJ **175** (2011)

[21] Borgelt, C. : Efficient implementations of apriori and eclat. In : ICDM Workshop FIMI. (2003)

[22] Uno, T., Asai, T., Uchida, Y., Arimura, H. : An efficient algorithm for enumerating closed patterns in transaction databases. In : DS. (2004)

[23] Khiari, M., Lallouet, A., Vautard, J. : Extraction de Motifs sous Contraintes Quantifiées. In : JFPC 2012