Description interactive de l'intérêt de l'utilisateur via l'échantillonnage de motifs

Moditha Hewasinghage, Suela Isaj, Arnaud Giacometti, Arnaud Soulet

Université François-Rabelais de Tours, LI EA 6300
Campus de Blois, 41000 Blois
prenom.nom@univ-tours.fr

Résumé. La plupart des méthodes d'extraction de motifs requièrent que l'utilisateur formalise son intérêt avec une mesure d'intérêt et des seuils. L'utilisateur est souvent incapable d'expliciter son intérêt mais il saura juger si un motif donné est pertinent ou non. Dans cet article, nous proposons une nouvelle méthode de découverte de motifs interactive en supposant que seule une partie des données est intéressante pour l'utilisateur. En intégrant le retour utilisateur de motifs proposés un à un, notre méthode vise à échantillonner des motifs avec une probabilité proportionnelle à leur fréquence d'apparition au sein des transactions implicitement préférées par l'utilisateur. Nous démontrons que notre méthode identifie exactement les transactions implicitement préférées par l'utilisateur sous réserve de la consistance de ses retours. Des expérimentations montrent les bonnes performances de l'approche en terme de précision et rappel.

1 Introduction

La découverte de motifs est un outil puissant pour extraire des motifs intéressants au sein d'un jeu de données. Néanmoins, la plupart des approches reposent sur le paradigme de la recherche par requête qui peut être une contrainte à satisfaire ou une mesure à maximiser. Il est difficile à l'utilisateur final d'exprimer explicitement son intérêt sous la forme d'une telle requête. Pour cette raison, l'extraction interactive de motifs van Leeuwen (2014) vise à capturer cette requête en observant le retour de l'utilisateur au fur et à mesure que des motifs lui sont présentés. La plupart des méthodes itèrent un cycle en trois phases. Une phase d'extraction extrait des motifs pertinents. Les premières itérations passées, cette pertinence intègre les préférences apprises. En effet, une phase d'interaction glane les retours de l'utilisateur (e.g., notation des motifs) qui seront généralisés par une phase d'apprentissage en un modèle de préférence.

Dans ce contexte, le choix du modèle de préférence s'avère déterminant pour bien identifier l'intérêt de l'utilisateur. Par exemple, le modèle du produit pondéré associe un poids à chaque item et considère alors le score d'un motif comme le produit des poids de ses items Bhuiyan et al. (2012). Dans ce cas, aucune distinction sur les transactions ne sera faite. Les autres propositions de la littérature associent à chaque motif un vecteur de caractéristiques. Grâce aux retours de l'utilisateur, certains vecteurs seront jugés plus pertinents que d'autres et l'apprentissage de ce classement construira un modèle de préférence sur l'ensemble des motifs Rueping

(2009); Dzyuba et al. (2014). Dans cet article, nous faisons l'hypothèse que les transactions sont divisées en deux parties : les transactions préférées et les autres. Ainsi, l'utilisateur souhaite extraire les motifs qui décrivent ses transactions préférées. Il s'agit en quelque sorte d'une généralisation de la découverte de contrastes (qui vise aussi à caractériser une partie des transactions). Mais dans notre cas, la variable cible est découverte au cours de l'interaction.

Au-delà de ce modèle de préférence, cet article présente une méthode originale basée sur l'échantillonnage de motifs. Grâce à cette technique, nous montrons comment les préférences apprises peuvent être directement utilisées pour extraire les motifs à présenter à l'utilisateur. Si les retours de l'utilisateur correspondent bien à son intérêt, nous garantissons à la fois que les transactions préférées seront bien identifiées et que les motifs extraits décriront exactement ces transactions. Enfin, une étude expérimentale montre l'augmentation rapide du rappel en maintenant une bonne précision (même si les transactions préférées sont peu nombreuses).

2 Travaux relatifs

L'extraction interactive de motifs van Leeuwen (2014) est un problème d'apprentissage bidirectionnel entre l'utilisateur et le système. D'une part, le système apprend les préférences de l'utilisateur à partir de ses retours (sens "de l'utilisateur vers le système"). D'autre part, l'utilisateur apprend de nouvelles connaissances à partir du jeu de données à travers les motifs fournis par le système (sens "du système vers l'utilisateur"). Ce problème diffère donc de l'apprentissage actif traditionnel Settles (2010) qui n'inclut pas le sens "du système à l'utilisateur". Cette distinction est importante. Premièrement, les requêtes demandées à l'utilisateur sont des motifs et non des transactions. Dans la plupart des tâches d'apprentissage actif, le retour demandé par l'utilisateur porte directement sur les objets et non sur la généralisation de ces objets (bien qu'il existe quelques exceptions notables Rashidi et Cook (2011); Bessiere et al. (2013)). Deuxièmement, le choix de la requête présentée à l'utilisateur ne peut pas seulement viser à améliorer le modèle de préférence contrairement à l'apprentissage actif traditionnel. Pour que l'utilisateur continue d'interagir avec le système, ce dernier doit fournir des motifs intéressants à l'utilisateur (i.e., du point de vue de ses préférences). Troisièmement, la requête présentée à l'utilisateur à chaque itération doit être calculée en quelques secondes pour maintenir une interaction satisfaisante. Cette contrainte n'est pas forte dans l'apprentissage actif traditionnel puisque l'espace de recherche des requêtes est petit. C'est beaucoup plus difficile d'extraire des motifs intéressants en un temps limité du fait de la grande taille de l'espace de recherche.

Un des défis de l'extraction interactive de motifs est donc d'extraire des motifs pertinents pour l'utilisateur (sens "du système vers l'utilisateur") tout en améliorant son modèle de préférences (sens "de l'utilisateur vers le système"). En pratique, les premières méthodes Xin et al. (2006); Rueping (2009) ignoraient l'utilisation d'un critère pour favoriser la diversité des requêtes qui aurait permis l'acquisition complète des préférences. Une approche récente Dzyuba et al. (2014) a néanmoins montré l'intérêt de tenir compte de la diversité à l'instar de l'apprentissage actif. Ce travail a aussi montré l'importance de l'alétoire pour améliorer la diversité. Ce besoin d'aléatoire justifie l'utilisation de l'échantillonnage de motifs Bhuiyan et al. (2012). En plus de sa diversité intrinsèque, nous montrons ici comment tirer parti des propriétés statistiques de l'échantillonnage pour contrôler l'erreur du modèle de préférences appris afin de mieux choisir les requêtes (motifs extraits) comme dans Giacometti et Soulet (2016).

Un autre défi est d'extraire de nouveaux motifs à chaque itération en seulement quelques secondes afin de maintenir une interaction satisfaisante. Ce besoin n'est pas rempli par les méthodes traditionnelles d'extraction de motifs. Ainsi, les premières méthodes Xin et al. (2006); Rueping (2009) étaient basées sur une extraction préliminaire et ensuite, elles ré-ordonnaient cette collection de motifs selon le critère mis à jour en tenant compte de l'évolution du modèle de préférence. Cette approche par post-traitement n'autorise pas la découverte de nouveaux motifs qui auraient été précédemment omis. Plus récemment, une approche par recherche en faisceau Dzyuba et al. (2014) a été proposée pour extraire à chaque itération de nouveaux motifs qui maximisent le critère mis à jour (combinant qualité et diversité dans ce cas). Une telle approche reste lente et elle ne parvient pas toujours à trouver des motifs très diversifiés. Dans ce contexte, l'échantillonnage de motifs Bhuiyan et al. (2012) est une technique attractive puisqu'elle donne un accès rapide à tous les motifs ayant une valeur non nulle par rapport au citère mis à jour, garantissant une très bonne diversité. Dans cet article, nous adoptons une procédure aléatoire en deux étapes Boley et al. (2011) dont la complexité est linéaire avec la taille du jeu de données.

3 Formulation du problème

Cette section formule en deux temps le problème de la description de l'intérêt de l'utilisateur en exploitant ses retours. Auparavant nous rappelons des définitions préliminaires. Soit $\mathcal{I}$ un ensemble de littéraux nommés *items*, un itemset (ou un motif) est un sous-ensemble de $\mathcal{I}$. Le langage des itemsets correspond à $\mathcal{L} = 2^{\mathcal{I}}$. Un jeu de données transactionnel $\mathcal{D}$ est un multi-ensemble d'itemsets de $\mathcal{L}$. Chacune des observations de $\mathcal{D}$ est appelée *transaction*, et Δ représente l'ensemble de tous les jeux de données possibles. La découverte de motifs tire avantage de mesures d'intérêt pour évaluer la pertinence d'un motif. Plus précisément, une mesure d'intérêt pour un langage $\mathcal{L}$ est une fonction définie de $\mathcal{L} \times \Delta$ dans $\Re$. Typiquement, le support d'un motif X dans un jeu de données $\mathcal{D}$ est la proportion de transactions de $\mathcal{D}$ qui contiennent le motif X : $supp(X, \mathcal{D}) = |\{t \in \mathcal{D} | X \subseteq t\}|/|\mathcal{D}|$.

3.1 Description de l'intérêt de l'utilisateur

En général, on considère que tous les utilisateurs portent un même intérêt pour toutes les transactions. Dans notre approche, nous considérons que l'intérêt de l'utilisateur n'est pas le même pour toutes les transactions (et évidemment différent pour chaque utilisateur). Une partie des transactions, dénotée par $\mathcal{D}^1$, est implicitement préférée par l'utilisateur par rapport aux autres transactions (partie dénotée par $\mathcal{D}^0$). Par exemple, le tableau 1 propose un jeu de données transactionnel avec 4 transactions $t_1, \ldots, t_4$ décrites par 5 items A, B, C, D et E où les transactions t_1 et t_2 appartiennent à $\mathcal{D}^1$ car elles sont préférées par l'utilisateur.

L'échantillonnage de motifs Boley et al. (2011) donne un accès au langage $\mathcal{L}$ en tirant des motifs selon une distribution $p : \mathcal{L} \rightarrow [0, 1]$ qui est définie par rapport à une mesure d'intérêt $m : p(.) = m(.)/Z$ où Z est une constante de normalisation. Ainsi, l'utilisateur a un accès rapide à l'ensemble des motifs du langage et ce sans paramètre (excepté la taille potentielle de la réponse). Dans notre contexte, puisque l'utilisateur n'est pas intéressé par toutes les transactions de $\mathcal{D}$ (mais seulement par un sous-ensemble $\mathcal{D}^1$ de $\mathcal{D}$), nous ne voulons pas extraire l'échantillon de motifs par rapport à une mesure d'intérêt m évaluée sur $\mathcal{D}$, mais sur $\mathcal{D}^1$. En

$\mathcal{D}$

Trans.	Items					Préférence
t_1	A	B			E	1
t_2	A	B				1
t_3		B	C	D		0
t_4			C	D		0
	Connu par avance					Inconnu

TAB. 1: Exemple jouet de jeu de données

effet, la mesure d'intérêt m évaluée sur $\mathcal{D}^1$ est plus appropriée car cela focalise l'extraction sur les motifs décrivant les transactions préférées de l'utilisateur. Cependant, les transactions préférées dans $\mathcal{D}^1$ ne sont pas connues par avance. Nous formalisons notre problème ainsi :

Etant donné un jeu de données $\mathcal{D}$ contenant un ensemble inconnu de transactions préférées $\mathcal{D}^1$, notre problème consiste à construire une séquence de motifs $\langle X_1, \ldots, X_k \rangle$ telle que la probabilité $\mathbf{P}_i(X)$ de tirer un motif X à l'étape i tend vers $supp(X, \mathcal{D}^1)/Z$ lorsque i tend vers $+\infty$ où Z est une constante de normalisation ($Z = \sum_{X \in \mathcal{L}} supp(X, \mathcal{D}^1)$).

3.2 Interaction avec l'utilisateur

Dans la formulation du problème ci-dessus, nous n'avons pas précisé comment parvenir à découvrir l'intérêt de l'utilisateur i.e., les transactions préférées $\mathcal{D}^1$ qui sont initialement inconnues. Comme en apprentissage actif, une stratégie serait de demander directement à l'utilisateur si certaines transactions sont intéressantes ou non pour découvrir la classe $\mathcal{D}^1$. Dans notre approche, parce que le jeu de données $\mathcal{D}$ peut être très large (surtout par rapport à la taille réduite du jeu de données $\mathcal{D}^1$), nous préférons demander à l'utilisateur si un motif extrait est bien une généralisation des transactions préférées.

Plus précisément, nous modélisons le retour de l'utilisateur par un oracle $\mathcal{O}$ qui est une fonction de $\mathcal{L}$ vers $\{0, 1\}$. Etant donné un motif X, nous avons $\mathcal{O}(X) = 1$ (resp. 0) si l'oracle donne un retour positif (resp. négatif) pour un motif X. Comme le retour utilisateur au sujet d'un même motif X peut changer durant le processus, nous pouvons considérer que $\mathcal{O}$ est une variable aléatoire. Dans cette configuration, $\mathbf{P}(1/X)$ représente la probabilité d'avoir un retour positif sachant le motif X quand l'oracle est consulté. Par exemple, $\mathbf{P}(1/AB) = 1$ signifie que l'oracle donne toujours un retour positif pour AB.

Maintenant, nous devons faire le lien entre les retours de l'utilisateur sur les motifs et les transactions. En général, nous considérons qu'un utilisateur peut observer soit une transaction $t \in \mathcal{D}$, soit un motif $X \in \mathcal{L}$, soit une paire (t, X) où $X \subseteq t$ est une généralisation de t. Dans cette configuration, $\mathbf{P}(t)$ (resp. $\mathbf{P}(X)$) est la probabilité que l'utilisateur observe une transaction $t \in \mathcal{D}$ (resp. un motif $X \in \mathcal{L}$). De plus, $\mathbf{P}(t, X)$ est la probabilité que l'utilisateur observe conjointement une transaction t et un motif X. En utilisant la formule des probabilités totales et la définition d'une probabilité conditionnelle, nous avons :

$$\mathbf{P}(1/t) = \frac{\mathbf{P}(1,t)}{\mathbf{P}(t)} = \frac{\sum_{X \in \mathcal{L}} \mathbf{P}(1,t,X)}{\mathbf{P}(t)} = \frac{\sum_{X \in \mathcal{L}} \mathbf{P}(t) \times \mathbf{P}(X/t) \times \mathbf{P}(1/t,X)}{\mathbf{P}(t)}$$
$$= \sum_{X \in \mathcal{L}} \mathbf{P}(X/t) \times \mathbf{P}(1/X)$$

en faisant l'hypothèse que l'oracle considère seulement le motif X pour déterminer son retour (et pas la transaction t qui par la suite n'est pas montrée à l'utilisateur), i.e. $\mathbf{P}(1/t, X) = \mathbf{P}(1/X)$. En fixant que $\mathbf{P}(X/t) = 0$ si $X \not\subseteq t$, nous obtenons finalement :

$$\mathbf{P}(1/t) = \sum_{X \subseteq t} \mathbf{P}(X/t) \times \mathbf{P}(1/X) \tag{1}$$

Cette équation montre comment les préférences de l'utilisateur sur les transactions peuvent être inférées de ses retours sur les motifs. Cela montre aussi qu'il ne sera pas possible d'apprendre les transactions préférées des retours sur les motifs si ces retours ne sont pas consistants. Par exemple, si $P(1/X) = 0$ pour tous les motifs $X \subseteq t$, alors nous calculons que $P(1/t) = 0$ que t soit une transaction préférée ou non. Pour cette raison,nous introduisons la notion de consistance de l'oracle par rapport aux transactions préférées $\mathcal{D}^1$:

Définition 1 (Consistance de l'oracle) *Etant donné un sous-ensemble de transactions préférées $\mathcal{D}^1 \subseteq \mathcal{D}$, un oracle $\mathcal{O}$ est consistant avec $\mathcal{D}^1$ si et seulement si pour toute transaction $t \in \mathcal{D}$, nous avons soit $P(1/t) > 0.5$ si $t \in \mathcal{D}^1$, soit $P(1/t) < 0.5$ sinon.*

Ainsi, si le retour de l'utilisateur est consistant avec ses préférences, le problème formulé dans la section précédente peut être résolu en estimant les probabilités $P(1/t)$ pour toutes les transactions de $t \in \mathcal{D}$. En effet, si nous parvenons à estimer cette distribution, alors il sera possible de tirer des motifs suivant leur support dans $\mathcal{D}^1$. Maintenant, nous reformulons notre problème de la manière suivante :

Etant donné un jeu de données $\mathcal{D}$ contenant un ensemble inconnu de transactions préférées $\mathcal{D}^1$ et un oracle oracle $\mathcal{O}$ consistant avec $\mathcal{D}^1$, le problème consiste à construire en même temps :

- **une séquence $\langle w_1, \ldots, w_k \rangle$ de vecteurs de poids ω_i définis de $\mathcal{D}$ vers $\Re$ tels que pour chaque transaction $t \in \mathcal{D}$, nous ayons : $\lim_{i \to +\infty} w_i(t) = \mathbf{P}(1/t)$, et**
- **une séquence $\langle X_1, \ldots, X_k \rangle$ de motifs X_i telle que si $\mathbf{P}_i(X)$ dénote la probabilité de tirer le motif X à l'étape i, alors $\lim_{i \to +\infty} \mathbf{P}_i(X) = supp(X, \{t \in \mathcal{D} | \mathbf{P}(1/t) > 0.5\})/Z$ où Z est une constante de normalisation.**

4 Apprentissage à partir de retours de l'utilisateur

4.1 Algorithme interactif de description des transactions préférées

Dans le but de résoudre le problème posé dans la section 3.2, nous proposons d'utiliser une procédure aléatoire en deux étapes pour générer les motifs présentés à l'utilisateur.

Dans Boley et al. (2011), les auteurs montrent comment échantillonner des motifs selon une distribution proportionnelle au support des motifs. Dans notre approche, nous proposons d'échantillonner des motifs suivant une distribution proportionnelle à leur support *pondéré*. Plus formellement, étant donné un jeu de données $\mathcal{D}$ et un vecteur de poids $w : \mathcal{D} \to \Re$, le support pondéré du motif X dans $\mathcal{D}$, noté $supp_\omega(X, \mathcal{D})$, est défini par : $supp_\omega(X, \mathcal{D}) = \sum_{t \in \mathcal{D}, X \subseteq t} \omega(t) / \sum_{t \in \mathcal{D}} \omega(t)$. Il est facile de voir que si toutes les transactions de $\mathcal{D}^1$ ont un poids de 1 et que toutes les autres ont un poids de 0, alors $supp_\omega(X, \mathcal{D}) = supp(X, \mathcal{D}^1)$.

Algorithm 1 Echantillonnage de motifs selon un support pondéré

Input: Un jeu de données $\mathcal{D}$ et un vecteur de poids ω
Output: Un itemset tiré aléatoirement $X \sim supp_\omega(\mathcal{L}, \mathcal{D})$
1: Soient les poids ω' définis par $\omega'(t) := 2^{|t|} \times \omega(t)$ pour tout $t \in \mathcal{D}$
2: Tirer une transaction proportionnellement à $\omega' : t \sim \omega'(\mathcal{D})$
3: **return** un itemset tiré proportionnellement à la distribution uniforme $u : X \sim u(2^t)$

L'algorithme 1 adapte la procédure aléatoire en deux étapes Boley et al. (2011) pour générer des motifs avec une probabilité proportionnelle à celle de leur support pondéré.

Maintenant que nous savons comment tirer les motifs présentés à l'utilisateur, nous devons montrer comment la probabilité $\mathbf{P}(1/t)$ de chaque transaction peut être estimée. Si les motifs présentés à l'utilisateur sont générés en utilisant l'algorithme 1, on a pour tout motif X : $\mathbf{P}(X/t) = \frac{1}{|2^t|}$. Ainsi, en utilisant l'équation 1, pour chaque transaction $t \in \mathcal{D}$, nous avons :

$$\mathbf{P}(1/t) = \sum_{X \subseteq t} \mathbf{P}(X/t) \times \mathbf{P}(1/X) = \frac{1}{|2^t|} \sum_{X \subseteq t} \mathbf{P}(1/X) \tag{2}$$

Pour estimer les probabilités $\mathbf{P}(1/t)$, nous proposons la méthode détaillée par l'algorithme 2. A chaque étape i, cet algorithme commence par tirer un motif X_i (ligne 4) selon son support pondéré $s_i = supp_{\omega_i}(X_i, \mathcal{D})$ (voir l'algorithme 1). Il est ensuite demandé à l'oracle (ligne 5) si le motif X_i est intéressant ou non. Alors, en utilisant la séquence de retours de l'utilisateur $\langle (X_1, f_1, s_1), \ldots, (X_i, f_i, s_i) \rangle$, la pondération $\bar{\omega}_{i+1}(t)$ est mise à jour pour chaque transaction $t \in \mathcal{D}$ afin d'approximer $\mathbf{P}(1/t)$ (ligne 7). Ensuite, une estimation corrigée de $\mathbf{P}(1/t)$, notée $\tilde{\omega}_{i+1}(t)$, est calculée à la ligne 8 en exploitant l'inégalité de Bennett. Enfin, une binarisation de $\tilde{\omega}_{i+1}(t)$, notée $\omega_{i+1}(t)$, est effectuée à la ligne 9 pour mettre à 1 (resp. 0) le poids d'une transaction très certainement préférée (resp. non-préférée). Le principe de l'intégration des retours utilisateur pour successivement calculer l'estimation moyenne de $\mathbf{P}(1/t)$, son estimation corrigée puis enfin, sa binarisation est détaillé dans la section suivante.

Algorithm 2 Echantillonnage interactif de motif

Input: Un jeu de données $\mathcal{D}$ et un oracle $\mathcal{O}$
1: $i := 1$ et $\omega_1(t) := \bar{\omega}_1(t) := 0.5$ for all $t \in \mathcal{D}$
2: Soit F une séquence vide d'observations
3: **repeat**
4: Tirer un motif X_i de $\mathcal{D}$ selon $supp_{w_i}$
5: Ajouter le retour utilisateur $(X_i, \mathcal{O}(X_i), supp_{w_i}(X_i, \mathcal{D}))$ à la séquence F
6: **for all** $t \in \mathcal{D}$ **do**
7: $\bar{\omega}_{i+1}(t) := \frac{\sum_{(X_j, f_j, s_j) \in F, X_j \subseteq t} f_j / s_j}{\sum_{(X_j, f_j, s_j) \in F, X_j \subseteq t} 1/s_j}$
8: Calculer l'estimation corrigée $\tilde{\omega}_{i+1}(t)$ de $\mathbf{P}(1/t)$ en utilisant $\bar{\omega}_{i+1}(t)$
9: Calculer l'estimation binarisée $\omega_{i+1}(t)$ de $\tilde{\omega}_{i+1}(t)$
10: **end for**
11: $i := i + 1$
12: **until** L'utilisateur arrête le processus

4.2 Intégrer les retours utilisateur à la pondération

Estimation Soit une séquence de retours utilisateur F formée des triplets (X_j, f_j, s_j) où pour chaque motif X_j, f_j est le retour de l'utilisateur et s_j est le support du motif au moment du tirage. A partir de F, une première estimation de la probabilité $\mathbf{P}(1/t)$ est définie comme suit :

$$\bar{\omega}_F(t) = \frac{\sum_{(X_j,f_j,s_j)\in F, X_j \subseteq t} f_j/s_j}{\sum_{(X_j,f_j,s_j)\in F, X_j \subseteq t} 1/s_j}$$

Intuitivement, l'estimation calcule la proportion de retours positifs au sein de l'échantillon en pondérant le retour par sa probabilité de tirage. De cette manière, un motif qui a deux fois plus de chance d'être tiré a un poids deux fois moins important. Avec ce biais de tirage des motifs, cette pondération garantit que $\bar{\omega}_F(t)$ est une bonne approximation de $\mathbf{P}(1/t)$:

Propriété 1 *Etant donnés un jeu de données $\mathcal{D}$ et une séquence de retours utilisateur F, pour chaque transaction $t \in \mathcal{D}$, le poids $\bar{\omega}_F(t)$ converge vers $\mathbf{P}(1/t)$ quand $|F|$ tend vers l'infini.*

En pratique, cette propriété signifie que l'ajout de nouveaux retours tend à améliorer l'estimation de la probabilité $\mathbf{P}(1/t)$. Cependant, dans l'algorithme 2, comme le tirage d'un motif X_i dépend de l'estimation calculée à l'étape précédente, cela peut provoquer des effets de bord. Par exemple, en faisant l'hypothèse que le motif $X_0 = \emptyset$ soit initialement tiré et reçoive un retour négatif $\mathcal{O}(\emptyset) = 0$, l'estimation $\bar{\omega}_{\langle(\emptyset,0,1/Z)\rangle}(t)$ de chaque transaction sera $\frac{0/(1/Z)}{1/(1/Z)} = 0$ (parce que l'ensemble vide est inclus dans chacune des transactions et la probabilité de le tirer est $1/Z$). Dans ce cas, aucun motif ne pourrait être tiré à l'étape suivante. Il ne serait alors pas possible d'améliorer l'estimation initiale faute de pouvoir obtenir de nouveaux retours. Pour éviter cela, il est essentiel de réaliser une correction statistique sur l'estimation.

Estimation corrigée L'idée est de bénéficier de l'estimation ci-dessus en calculant un intervalle de confiance qui borne la probabilité $\mathbf{P}(1/t)$. Pour cela, nous choisissons d'utiliser l'inégalité de Bennett Maurer et Pontil (2009) pour estimer l'erreur courante car elle est vraie quelque soit la distribution de probabilité. Après k observations indépendantes d'une variable aléatoire r à valeur réelles dans l'intervalle $[0, 1]$, l'inégalité de Bennett garantit que, avec une confiance $1 - \delta$, la vraie moyenne de r est au moins $\overline{r} - \epsilon$ où $\overline{r}$ et $\overline{\sigma}$ sont respectivement la moyenne et l'écart-type observés dans l'échantillon et $\epsilon = \sqrt{\frac{2\overline{\sigma}^2 \ln(1/\delta)}{k}} + \frac{\ln(1/\delta)}{3k}$. Nous utilisons ce résultat statistique afin de borner la vraie valeur de $\mathbf{P}(1/t)$ à partir de la séquence F des retours utilisateurs :

Propriété 2 (Bornes de $\mathbf{P}(1/t)$) *Etant donnés un jeu de données $\mathcal{D}$, une séquence de retours F et une confiance $1 - \delta$, la probabilité $\mathbf{P}(1/t)$ de la transaction t est bornée ainsi :*

$$\underbrace{\max\{0, \bar{\omega}_F(t) - \epsilon\}}_{inf_F(t)} \leq \mathbf{P}(1/t) \leq \underbrace{\min\{\bar{\omega}_F(t) + \epsilon, 1\}}_{sup_F(t)}$$

avec $\epsilon = \sqrt{2\overline{\sigma}\ln(1/\delta)/k} + \ln(1/\delta)/3k$ où $\overline{\sigma} = \sqrt{\bar{\omega}_F(t) - \bar{\omega}_F(t)^2}$ est l'écart-type de $\bar{\omega}_F(t)$.

Cette propriété est importante car elle approxime l'erreur de notre estimation $\bar{\omega}_F$. Reprenons l'exemple précédent où le tirage de l'ensemble vide à la première itération donnait

$\bar{\omega}_{\{(\emptyset,0,1/Z)\}}(t) = 0$ pour chaque transaction t. Dans ce cas, les bornes inférieure et supérieure sont respectivement 0 et 1. Bien sûr, la valeur $\bar{\omega}_F(t)$ est dans cet intervalle mais il est plus prudent de minimiser l'erreur (ici, 0.5). De manière générale, pour estimer $\mathbf{P}(1/t)$, nous proposons finalement de prendre la valeur moyenne entre ces deux bornes en calculant :

$$\tilde{\omega}_F(t) = \frac{inf_F(t) + sup_F(t)}{2}$$

Cette estimation corrigée tend vers la probabilité $\mathbf{P}(1/t)$ quand le nombre de retours utilisateur augmente (puisque les deux bornes tendent vers cette même probabilité). Néanmoins, même si l'oracle $\mathcal{O}$ est consistant avec $\mathcal{D}^1$, la probabilité $\mathbf{P}(1/t)$ n'est pas égale à 1 pour une transaction préférée et 0, pour les autres. Donc, en utilisant cette estimation corrigée, le tirage réalisé ne convergerait pas vers un tirage proportionnel au support dans $\mathcal{D}^1$.

Binarisation de l'estimation La binarisation de l'estimation consiste juste à mettre un poids de 1 lorsqu'il est certain que la probabilité $\mathbf{P}(1/t)$ est supérieure à 0.5 ou à l'inverse de mettre un poids de 0 si $\mathbf{P}(1/t)$ est inférieure à 0.5 avec certitude. Cette notion de certitude repose sur l'erreur estimée avec la propriété 2 :

$$\omega_F(t) = \begin{cases} 1 & \text{si } inf_F(t) > 0.5 \\ 0 & \text{si } sup_F(t) < 0.5 \\ \frac{inf_F(t)+sup_F(t)}{2} & \text{sinon} \end{cases} \quad (3)$$

Avec cette pondération, si l'oracle est consistant avec $\mathcal{D}^1$, alors le poids d'une transaction préférée de $\mathcal{D}^1$ tend vers 1 et celui d'une transaction non-préférée tend vers 0. De manière intéressante, la diminution à 0 des transactions non-préférées favorise le tirage d'autres transactions dont l'estimation n'est pas encore suffisamment affinée. A l'inverse l'augmentation à 1 des transactions préférées défavorise le tirage d'autres transactions encore incertaines. Ainsi, la préoccupation de fournir des motifs pertinents à l'utilisateur est soutenue.

Grâce à l'utilisation de propriétés statistiques, il est possible de conclure sur la bonne convergence de la méthode :

Théorème 1 (Convergence) *Etant donnés un ensemble de transactions préférées $\mathcal{D}^1 \subseteq \mathcal{D}$ et un oracle $\mathcal{O}$ consistant avec $\mathcal{D}^1$, pour chaque transaction $t \in \mathcal{D}$, les poids $\bar{\omega}_i(t)$ et $\tilde{\omega}_i(t)$ calculés par l'algorithme 2 convergent vers $\mathbf{P}(1/t)$ quand i tend vers l'infini. De plus, $\omega_i(t)$ converge vers 1 ssi $t \in \mathcal{D}^1$ (sinon vers 0) quand i tend vers l'infini.*

Par ailleurs, les complexités temporelle et en espace de cette approche en $O(k|\mathcal{D}||\mathcal{I}|)$ (où k est le nombre de motifs tirés) sont excellentes. En effet, les pondérations sont calculées incrémentalement en conservant un nombre fixe d'informations pour le numérateur et le dénominateur. Il n'est pas nécessaire de conserver le détail de tous les retours de l'utilisateur.

5 Expérimentations

Dans cette section, nous rapportons les évaluations expérimentales réalisées sur 6 jeux de données provenant de l'UCI Machine Learning repository (`archive.ics.uci.edu/ml`), le tableau 2 en donnant les principales caractéristiques. Les jeux de données sont utilisés sans

$\mathcal{D}$	$\lvert\mathcal{D}\rvert$	$\lvert\mathcal{I}\rvert$	$E[\lvert\mathcal{D}^1\rvert]/\lvert\mathcal{D}\rvert$	$\lvert\mathcal{D}^1_{min}\rvert$	$\mathcal{D}$	$\lvert\mathcal{D}\rvert$	$\lvert\mathcal{I}\rvert$	$E[\lvert\mathcal{D}^1\rvert]/\lvert\mathcal{D}\rvert$	$\lvert\mathcal{D}^1_{min}\rvert$
Abalone	4 177	28	10.9%	1 374	Mushroom	8 124	119	0.9%	3,646
Connect	67 557	129	0.23%	129	Sick	2 800	58	2.97%	450
Hypo	3 163	47	5.19%	1 035	Waveform	5 000	67	0.09%	1,662

TAB. 2: Caractéristiques des jeux de données

pré-traitements particuliers, un item correspondant généralement à un couple $(attribut, valeur)$. Notons enfin que par construction, les motifs retournés par l'algorithme 1 ne peuvent contenir deux items portant sur le même attribut.

Protocole expérimental Pour chaque jeu de données $\mathcal{D}$, le sous-ensemble $\mathcal{D}^1 \subseteq \mathcal{D}$ des transactions préférées est construit de deux manières différentes. Pour les expériences sur la vitesse de convergence, les transactions préférées sont générées aléatoirement. Un motif X est tiré aléatoirement avec une probabilité proportionnelle à son support dans $\mathcal{D}$, et $\mathcal{D}^1$ est définie comme étant les transactions contenant X, i.e. $\mathcal{D}^1 = \{t \in \mathcal{D} \ : \ X \subseteq t\}$. Pour chaque jeux de données, 1 000 motifs et sous-ensembles $\mathcal{D}^1$ sont générés aléatoirement, et les différentes mesures reportées sont les moyennes arithmétiques des 1 000 expérimentations réalisées. Pour les expériences sur l'influence de $1 - \delta$, 1 000 expérimentations sont également réalisées où la classe minoritaire est le sous-ensemble de transactions préférées. La cardinalité de cet ensemble, noté $\mathcal{D}^1_{min}$, est précisée dans la dernière colonne du tableau 2.

Nous utilisons un oracle déterministe modélisant un utilisateur pour lequel un motif est intéressant si sa fréquence d'apparition dans le sous-ensemble $\mathcal{D}^1$ des transactions préférées est supérieure à sa fréquence d'apparition dans l'ensemble de toutes les transactions : $\mathcal{O}(X) = 1$ si $supp(X, \mathcal{D}^1) > supp(X, \mathcal{D})$, 0 sinon.

A chaque itération de l'algorithme 2, la qualité est évaluée avec trois mesures : $Precision$, $Rappel$ et $F\text{-}mesure$. Elles comparent les transactions prédites comme préférées $\mathcal{D}^*$ par rapport aux transactions réellement préférées $\mathcal{D}^1$. Plus précisément, on a : $Precision = \frac{TP}{TP+FP}$, $Rappel = \frac{TP}{P}$ et $F\text{-}mesure = \frac{2 \times Precision \times Rappel}{Precision + Rappel}$ où $TP = |\mathcal{D}^* \cap \mathcal{D}^1|$ est le nombre de vrais positifs, $FP = |\mathcal{D}^* \cap \mathcal{D}^0|$ est le nombre de faux positifs, et $P = |\mathcal{D}^1|$ est le nombre de transactions préférées de $\mathcal{D}$. Par ailleurs, sur une fenêtre glissante de longueur 50, nous mesurons la proportion $\mathbf{P}(1)$ de motifs présentés à l'utilisateur qui l'intéressent, i.e. les motifs présentés qui sont évalués positivement par l'oracle. Soit une séquence $F = \langle (X_1, f_1, s_1), \ldots, (X_k, f_k, s_k) \rangle$ de $k > 50$ retours utilisateur, on a : $\mathbf{P}(1) = \frac{\sum_{i=k-49}^{k} f_k}{50}$. Enfin, sauf si indication contraire, le niveau de confiance $1 - \delta$ est fixé à 90%.

Convergence de la méthode Le premier résultat important de la figure 1 est que *la mesure de rappel* est monotone croissante quel que soit le jeu de données. De plus, cette croissance peut être très rapide. Pour trois des jeux de données ($Abalone$, $Hypo$ et $Sick$), la mesure de rappel dépasse 50% après moins de 50 itération, et 70% après 200 itérations. Elle dépasse même 90% pour 4 des 6 jeux de données après 500 itérations, alors que la complexité du problème à résoudre est importante. La troisième colonne du tableau 2 représente pour chaque jeu de données la taille relative moyenne de $\mathcal{D}^1$ (sur les 1 000 itérations) par rapport à la taille totale du jeu de données $\mathcal{D}$. On constate ainsi que la taille des sous-ensembles de transactions $\mathcal{D}^1$ à identifier est toujours très faible par rapport à la taille totale des jeux de données : 10.9% pour $Abalone$, 5.2% pour $Hypo$ et un peu moins de 3% pour $Sick$.

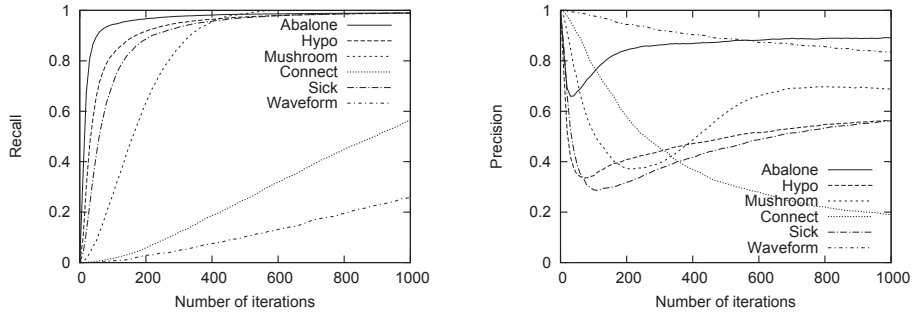

FIG. 1: Evolution des rappel et précision avec le nombre d'itérations.

La convergence est plus lente pour deux des jeux de données ($Connect$ et $Waveform$). Pour $Connect$, le rappel dépasse néanmoins 90% après $1\,800$ itérations. Enfin, il dépasse 90% pour $Waveform$ après $3\,900$ itérations. Ces convergences plus lentes s'expliquent par une taille encore plus faible des sous-ensembles $\mathcal{D}^1$ de transactions préférées à identifier. Dans la troisième colonne du tableau 2, on note en effet que la taille relative des sous-ensembles de transactions à identifier est de seulement 0.23% et 0.09% pour les jeux de données $Connect$ et $Waveform$. Ces pourcentages signifient qu'il s'agit (en moyenne) d'identifier par interaction de l'ordre de 150 transactions préférées sur $67\,557$ pour $Connect$ et de l'ordre de 4 transactions préférées sur $5\,000$ pour $Waveform$.

La mesure de précision commence toujours par décroître avant d'augmenter. Néanmoins, la croissance de la mesure de précision débute après moins de 200 itérations pour 4 des jeux de données ($Abalone$, $Hypo$, $Mushroom$ et $Sick$) et après environ $2\,000$ itérations pour deux autres jeux de données. Enfin, le niveau de précision final atteint reste satisfaisant pour tous les jeux de données au regard des excellents niveaux de rappel obtenus et de la difficulté d'identifier avec des interactions un sous-ensemble de transactions préférées de taille très faible.

L'évolution de la $F\text{-}mesure$ est présentée en figure 2, ainsi que le taux de motifs intéressants $\mathbf{P}(1)$ présentés à l'utilisateur (décrivant ses transactions préférées). On constate une croissance monotone de ces taux pour tous les jeux de données (la croissance est beaucoup plus lente pour $Connect$ et $Waveform$ et non visible sur les $1\,000$ premières itérations). Pour les autres jeux de données, ce taux reste très variable après $1\,000$ itérations (entre 29% pour $Sick$ et 63% pour $Abalone$). En croisant ces taux avec la taille relative des ensembles de transactions préférées (troisième colonne du tableau 2), on note que le taux de motifs intéressants présentés à l'utilisateur est d'autant plus important que cette taille relative est importante.

Impact du niveau de confiance Cette deuxième série d'expériences a pour objectif de mesurer l'impact du niveau de confiance dans la convergence de la méthode. Pour rappel, ce taux est utilisé pour estimer les bornes dans lesquelles se situent les probabilités $\mathbf{P}(1/t)$ et réaliser une binarisation des probabilités estimées (voir section 4.2). Les évolutions étant similaires sur les 6 jeux de données, la figure 3 présente uniquement les résultats pour $Mushroom$. Lorsque le niveau de confiance exigé augmente, les bornes calculées pour corriger les poids estimés des transactions sont moins étroites. Ainsi, la convergence des poids est plus lente et leur binarisation est retardée. Il en résulte une croissance plus lente du rappel lorsque le niveau de confiance

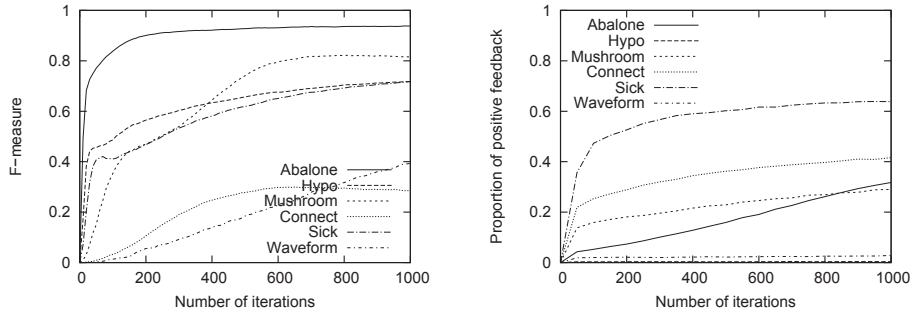

FIG. 2: Evolution de la $F\text{-}mesure$ et du taux de motifs intéressants avec le nombre d'itérations.

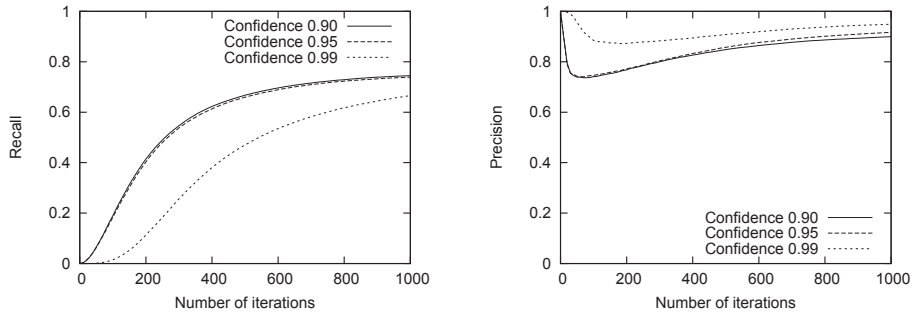

FIG. 3: Précision et rappel suivant la confiance pour $Mushroom$

est élevé. Mais en contrepartie, cette croissance plus lente du rappel est contrebalancée par une décroissance plus faible de la précision en première phase de convergence. On notera que cette réduction de la décroissance de la précision peut être très significative, i.e. de près de 10% sur les deux jeux de données, en passant d'un niveau de confiance de 95% à 99%.

Pour finir, on notera que si les mesures de rappel et précision évoluent différemment suivant le niveau de confiance (tout particulièrement en début de séquence), les rappel et précision tendent finalement vers des valeurs limites similaires (les courbes se rapprochant en fin de séquence). Ainsi, l'influence du niveau de confiance sur les évolutions des mesures de rappel et précision constitue une piste intéressante pour analyser comment contrôler et accélérer la convergence de la méthode.

6 Conclusion et perspectives

Cet article présente une nouvelle méthode d'extraction interactive de motifs en exploitant l'échantillonnage de motifs. Au-delà de l'efficacité, cette technique offre des garanties statistiques sur l'apprentissage des préférences et donc, sur la convergence de l'approche. Les

expérimentations illustrent cette bonne convergence sur plusieurs jeux de données. Le rappel de la méthode augmente rapidement et la précision reste raisonnable même si l'intérêt de l'utilisateur ne porte que sur quelques transactions. Le problème peut se généraliser facilement à d'autres mesures d'intérêt que le support. On pense évidemment à une mesure pour identifier des contrastes entre $\mathcal{D}^1$ et $\mathcal{D}^0$. La modélisation de l'intérêt de l'utilisateur pourrait considérer une pondération non-binaire et de manière plus ambitieuse, cette pondération pourrait même être étendue aux items voire à l'ensemble des couples item/transaction.

Remerciements. Ce travail a été partiellement soutenu par le CNRS, PEPS 2016, projet Préfute.

Références

Bessiere, C., R. Coletta, E. Hebrard, G. Katsirelos, N. Lazaar, N. Narodytska, C.-G. Quimper, et T. Walsh (2013). Constraint acquisition via partial queries. In *IJCAI'2013*, pp. 7.

Bhuiyan, M., S. Mukhopadhyay, et M. A. Hasan (2012). Interactive pattern mining on hidden data : a sampling-based solution. In *Proc. of the 21st ACM CIKM 2012*, pp. 95–104. ACM.

Boley, M., C. Lucchese, D. Paurat, et T. Gärtner (2011). Direct local pattern sampling by efficient two-step random procedures. In *Proc. of the 17th ACM SIGKDD 2011*, pp. 582–590.

Dzyuba, V., M. v. Leeuwen, S. Nijssen, et L. De Raedt (2014). Interactive learning of pattern rankings. *International Journal on Artificial Intelligence Tools 23*(06), 1460026.

Giacometti, A. et A. Soulet (2016). Frequent pattern outlier detection without exhaustive mining. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 196–207.

Maurer, A. et M. Pontil (2009). Empirical bernstein bounds and sample variance penalization. *arXiv preprint arXiv :0907.3740*.

Rashidi, P. et D. J. Cook (2011). Ask me better questions : active learning queries based on rule induction. In *Proc. of the 17th ACM SIGKDD 2011*, pp. 904–912.

Rueping, S. (2009). Ranking interesting subgroups. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 913–920. ACM.

Settles, B. (2010). Active learning literature survey. *University of Wisconsin, Madison 52*(55-66), 11.

van Leeuwen, M. (2014). Interactive data exploration using pattern mining. In *Interactive Knowledge Discovery and Data Mining in Biomedical Informatics*, pp. 169–182. Springer.

Xin, D., X. Shen, Q. Mei, et J. Han (2006). Discovering interesting patterns through user's interactive feedback. In *Proc. of the 12th ACM SIGKDD 2006*, pp. 773–778. ACM.

Summary

This paper proposes an interactive pattern mining method assuming that only some transactions are interesting for the user. By integrating his feedback, our method aims at sampling patterns with a probability proportional to their frequency in these preferred transactions. We demonstrate that our method accurately identifies the preferred transactions if user feedback are consistent. Experiments show the good performances of the approach.