Nouveaux Descripteurs Discriminants pour la Fouille Interactive de Motifs

Arnold Hien*, Samir Loudni*
Noureddine Aribi** Abdelkader Ouali*** Albrecht Zimmermann***

*TASC – DAPI, IMT-Atlantique, LS2N – CNRS, Nantes, France
{arnold.hien, samir.loudni}@imt-atlantique.fr
**Lab. LITIO, Université Oran1, Oran, Algérie
aribi.noureddine@univ-oran1.dz
***CNRS - UMR GREYC, Normandie Univ., UNICAEN, Caen, France
{abdelkader.ouali, albrecht.zimmermann}@unicaen.fr

Résumé. Le présent article est un résumé de l'article de Hien et al. (2023b) (publié à la conférence PAKDD 2023). Nous y introduisons la notion de motif discriminant que nous exploitons dans un cadre interactif de fouille de motifs. Il s'agit d'une approche de fouille de motifs centrée sur l'utilisateur qui exploite les préférences de ce dernier pour guider la recherche vers des motifs pertinents pour lui. Cela est rendu possible par l'introduction de mécanismes de feedback qui permettent à l'utilisateur de spécifier ses préférences sur les motifs extraits. Les approches existantes présentent une faiblesse consistant en l'utilisation de descripteurs statiques de bas niveau qu'elles exploitent pour apprendre des poids indépendants représentant l'importance de ces descripteurs pour l'utilisateur. Nous introduisons de nouveaux descripteurs plus complexes qui sont dérivées directement du feedback de l'utilisateur. Ces descripteurs sont utilisés pour apprendre des poids que nous agrégeons à ceux des descripteurs de bas niveau avec pour objectif d'améliorer l'apprentissage des préférences de l'utilisateur.

1 Introduction

La fouille interactive de motifs intègre l'utilisateur dans le processus de fouille afin de prendre en compte les préférences de celui-ci pour guider la recherche vers des motifs intéressants. Pour cela, le processus de fouille se déroule de manière itérative (Rüping, 2009) dans un framework où les actions se succèdent dans une boucle avec les étapes suivantes : (1) extraction d'un ensemble relativement restreint de motifs $\mathcal{X}$; (2) interaction dans laquelle l'utilisateur exprime ses préférences sur les motifs extraits ; (3) apprentissage des préférences sous forme d'un modèle qui sera par la suite exploitée pour extraire des nouveaux motifs plus intéressants.

Plusieurs approches de fouille interactive existent dans la littérature et se différencient par la manière de traiter chacune des trois étapes précédentes. Dans ce document, nous avons mis l'accent sur les méthodes qui effectuent un rangement des motifs à l'étape (2) de la fouille.

Pour l'extraction des motifs, Rüping (2009) utilise une approche interactive pour la découverte de sous-groupes tandis que Bhuiyan et Hasan (2016) introduisent une méthode interactive d'échantillonnage de motifs fréquents. Pour l'étape d'apprentissage, Joachims (2002) propose RANKSVM pour apprendre les préférences que l'utilisateur exprime. Dzyuba *et al.*, eux, proposent dans Dzyuba et al. (2014) et Dzyuba et van Leeuwen (2017) d'utiliser plutôt les méthodes d'optimisation par descente des coordonnées.

Pour apprendre les préférences de l'utilisateur, les différentes approches de l'état de l'art nécessitent de représenter les motifs par des *descripteurs* représentant leurs caractéristiques (items, transactions, fréquences, etc). L'apprentissage consiste alors à associer à chaque descripteur des poids afin d'apprendre une fonction de préférences.

Cependant, une des limites des descripteurs utilisés vient de leur utilisation comme des éléments indépendants les uns des autres. En effet, étant donné que chaque descripteur décrit une caractéristique particulière du motif, il est naturel de les considérer comme étant indépendants les uns des autres. Toutefois, il serait intéressant de considérer d'éventuelles interdépendances qui peuvent exister entre ces différents descripteurs. Typiquement, un utilisateur pourrait être « intéressé par des combinaisons particulières d'items » ou être « désintéressé par des transactions particulières ».

Dans cet article, nous introduisons une nouvelle classe de descripteurs dynamiques qui permettent d'"expliquer le rangement de l'utilisateur" et ainsi de saisir l'importance des interactions entre les éléments. Ces descripteurs exploitent le concept de *motifs discriminants* qui sépare les motifs auxquels l'utilisateur attribue un faible rang de ceux qui ont un rang élevé. En ajoutant temporairement ces descripteurs aux descripteurs statiques, nous pouvons leur associer des poids qui sont ensuite agrégés à ceux des autres descripteurs.

2 Préliminaires

Fouille de motifs. Soit un jeu de données transactionnelle $\mathcal{D}$, $\mathcal{I}$ l'ensemble des n *items* de $\mathcal{D}$ et $\mathcal{T} = \{1, ..., m\}$ l'ensemble des transactions. Chaque transaction t est un sous-ensemble d'items, *i.e.,* $t \subseteq \mathcal{I}$. Nous définissons un motif X comme un sous-ensemble non vide de $\mathcal{I}$ et sa couverture $\mathcal{V}_\mathcal{D}(X)$ est égale à l'ensemble des transactions qui le supportent, *i.e.,* $\mathcal{V}_\mathcal{D}(X) = \{t \in \mathcal{D} \mid X \subseteq t\}$. Le langage des motifs correspond à $\mathcal{L}_\mathcal{I} = 2^\mathcal{I} \backslash \emptyset$. La fouille de motifs consiste alors à trouver les motifs de la théorie $\mathcal{T}h(\mathcal{L}, \mathcal{D}, q) = \{\phi \in \mathcal{L} \mid \ q(\mathcal{D}, \phi)\ is\ true\}$. L'une des tâches les plus connues de la fouille est l'*extraction de motifs fréquents* présentée par Agrawal et Srikant (1994) et qui consiste à trouver l'ensemble des motifs X tels que $sup_\mathcal{D}(X) = |\mathcal{V}_\mathcal{D}(X)| \geq \theta$, θ étant le seuil de fréquence minimal.

Apprentissage des Préférences. Soit $\Phi : \mathcal{L}_\mathcal{I} \rightarrow \mathbb{R}$ une fonction représentant les préférences de l'utilisateur sur les motifs. La fouille interactive consiste à approximer itérativement les préférences de ce dernier avec une fonction φ. Pour cela, à chaque itération t du processus, l'algorithme de fouille interactive sélectionne k motifs $\mathcal{X}$ à présenter à l'utilisateur. Ces motifs sont alors rangés selon l'ordre de préférences donné par Φ, puis, une nouvelle approximation φ^{t+1} est faite et exploitée à l'itération suivante. Des explications plus détaillées sur les différentes étapes sont données dans notre article accepté (Hien et al., 2023b) mais également par Dzyuba et van Leeuwen (2017).

Algorithme 1 : DISPALE (**Di**scriminating **S**ub-**Pa**ttern feature **Le**arning)

Entrées : Base transactionnelle $\mathcal{D}$, ensemble de motifs $\mathcal{X}$
Données : Taille de requête k, nombre d'itérations T, par. de rétention ℓ, descripteurs $\mathcal{F}$
Sorties : φ : Fonction de classement
début
  $\mathcal{U} \leftarrow \emptyset$, $w^0_{\mathcal{F}} \leftarrow \mathbf{0}$, $\mathcal{X}^0 \leftarrow \emptyset$, $\varphi^0 = \varphi_{log}(w^0_{\mathcal{F}})$
  pour $t = 1, 2 \dots T$ **faire**
    $\mathcal{X}^t \leftarrow \text{TOP}(\mathcal{X}^{t-1}, \ell) \cup (\text{SAMPLEPATTERNS}(\mathcal{D}, \varphi^{t-1}) \times (k - \ell))$
    $\widehat{\mathcal{R}}^t \leftarrow \text{RANK}(\mathcal{X}^t)$, $disc \leftarrow \text{MINEDISCRIMINATING}(\mathcal{X}^t, \widehat{\mathcal{R}}^t)$, $\mathcal{U} \leftarrow \mathcal{U} \cup \widehat{\mathcal{R}}^t$
    $\langle w^t_{\mathcal{F}}, w^t_{\mathcal{F}_{disc}} \rangle \leftarrow \text{LEARNWEIGHTS}(\mathcal{U}, \mathcal{F} \cup \mathcal{F}_{disc})$
    $w^t_{\mathcal{F}} \leftarrow \text{UPDATEWEIGHTS}(w^t_{\mathcal{F}}, w^t_{\mathcal{F}_{disc}})$
    $\varphi^t \leftarrow \varphi_{log}(w^t_{\mathcal{F}})$
  retourner φ^T

3 DiSPaLe : descripteurs discriminants et fouille interactive

Dans cette section, nous présentons DISPALE, notre outil de fouille interactive de motifs exploitant des descripteurs complexes issus d'une combinaison des descripteurs statiques de bas niveau avec des *descripteurs discriminants*. Le processus de fouille de DISPALE est décrit par l'algorithme 1 dans lequel la fonction d'apprentissage est une fonction logistique φ_{log}.

3.1 Vers une description plus expressive et compréhensible des Motifs

Pour représenter les motifs, les méthodes de l'état de l'art exploitent leurs caractéristiques statiques comme les items qu'ils incluent, les transactions couvertes ou encore leur longueur (Bhuiyan et Hasan, 2016; Dzyuba et van Leeuwen, 2017). Or, comme nous l'avons indiqué, ces descripteurs sont considérés comme étant indépendants les uns des autres quelle que soit la fonction d'apprentissage utilisée. Cette approche permet de trouver les motifs qui sont *globalement* intéressant pour l'utilisateur, mais ne permet pas d'identifier certaines relations plus complexes. Par exemple, un utilisateur peut être " intéressé par les motifs contenant l'item i_1 si l'item i_3 est présent mais pas l'item i_4". Par ailleurs, les descripteurs traditionnels des motifs sont définis en amont du processus de fouille et restent figés pendant tout l'apprentissage. Les retours de l'utilisateur n'ont alors aucune influence sur leur structure.

Pour pallier à cela, nous proposons donc une nouvelle approche permettant une description plus expressive des motifs afin d'améliorer l'apprentissage des préférences de l'utilisateur. Pour cela, nous introduisons la notion de *motifs discriminants* qui sont des motifs corrélés au rangement effectué et qui ont été déterminant dans le retour de l'utilisateur.

a) Variance Interclasse. Comme nous l'avons expliqué plus haut, les motifs discriminants permettent d'expliquer le rangement de l'utilisateur et de déterminer pourquoi un ou plusieurs motifs ont obtenu un bon/mauvais rang. Une manière de modéliser la recherche de motifs discriminants consiste à considérer les rangs numériques donnés aux motifs individuels comme des étiquettes numériques et de considérer la recherche des discriminants comme un problème de régression. Dans notre cas, l'objectif n'est pas de construire un modèle de régression complet, mais seulement d'extraire un modèle individuel qui est en corrélation avec l'étiquette

Algorithme 2 : Extraction de motifs discriminants

```
1  Fonction MineDiscriminating(𝒳, R̂)
2      ICV_max ← 0, disc ← ∅, I_𝒳 ← {i ∈ X | X ∈ 𝒳}
3      𝒮 ← I_𝒳
4      pour chaque i ∈ I_𝒳 faire
5          si ICV(i, R̂) ≥ ICV_max alors
6              ICV_max ← ICV(i, R̂), disc ← {i}
7      pour chaque Y ∈ 𝒮 faire
8          tant que (∃i ∈ I_𝒳 ∧ ∃X ∈ 𝒳 st. Y ∪ {i} ⊆ X ∧ i ∉ Y) faire
9              si ICV(Y ∪ {i}, R̂) ≥ ICV_max alors
10                 ICV_max ← ICV(Y ∪ {i}, R̂), disc ← Y ∪ {i} , 𝒮 ← 𝒮 ∪ disc
11     retourner disc
```

numérique. Pour cela, nous exploitons la notion de variance inter-classe proposée par Morishita et Sese (2000).

Définition 1 *Soit $\mathcal{X}$ un ensemble de k motifs ordonnés selon l'ordre $\widehat{\mathcal{R}}$ de l'utilisateur. On note $\mathcal{X}_Y = \{X \in \mathcal{X} | X \supseteq Y\}$ l'ensemble des motifs X de $\mathcal{X}$ qui contiennent le sous-motif Y, et $\overline{\mathcal{X}}_Y = \mathcal{X} - \mathcal{X}_Y$. La variance interclasse du sous-motif Y est définie comme suit :*

$$ICV(Y, \widehat{R}) = |\mathcal{X}_Y| \cdot (\mu(\mathcal{X}) - \mu(\mathcal{X}_Y))^2 + |\overline{\mathcal{X}}_Y| \cdot (\mu(\mathcal{X}) - \mu(\overline{\mathcal{X}}_Y))^2$$

avec $\mu(\mathcal{X}) = \frac{1}{|\mathcal{X}|} \cdot \sum_{X \in \mathcal{X}} r(X)$, et $r(X)$ le rang du motif X dans $\widehat{\mathcal{R}}$.

b) Extraction de motifs discriminants. La variance inter-classe permet d'évaluer la corrélation des motifs avec le rangement de l'utilisateur, les motifs ou sous-motifs les plus corrélés étant ceux ayant les plus grandes valeurs d'ICV. Notre objectif est alors de trouver les sous-motifs $Y \subseteq X \in \mathcal{X}$ dont la présence dans un ou plusieurs motifs X a influencé leur rangement. Ainsi, si $Y \subseteq X$, on peut dire que le rangement de X à la $r(X)^e$ position dans $\mathcal{X}$ est plus susceptible d'être expliqué par la présence du sous-motif Y.

L'algorithme 2 décrit la procédure d'extraction d'un motif discriminant dont la recherche est effectuée par la fonction MINEDISCRIMINATING (voir Algorithme 1, ligne 1). Il prend en entrée un ensemble de motifs $\mathcal{X}$ ainsi que le rangement associé $\widehat{\mathcal{R}}$ et commence par évaluer l'ICV de tous les items des motifs $X \in \mathcal{X}$ (lignes 4 à 6). Puis, les items sont combinés au fur et à mesure afin d'obtenir des discriminants plus grand et plus précis (lignes 7 à 10). Le sous-motif inclus dans l'un des motifs $X \in \mathcal{X}$ et qui maximise l'ICV est alors enregistré dans $disc$ et l'ensemble des sous-motifs pouvant être étendus est mis à jour en y ajoutant $disc$ (lignes 9 à 10). À la fin, le motif discriminant $disc$ est retourné (ligne 11) et pourra être utilisé comme descripteur dans le processus de fouille interactive de motifs (voir Algorithme 1, lignes 1 et 1).

3.2 Motifs discriminants comme descripteurs

L'utilisation des motifs discriminants comme descripteurs apporte une sémantique nouvelle dans la description des motifs. En effet, ces nouveaux descripteurs permettent d'expliquer

le rang obtenu par un motif X pendant l'itération courante en identifiant une combinaisons d'items corrélés positivement ou négativement avec le rangement de l'utilisateur. Nous utiliserons donc les motifs discriminants comme des descripteurs numériques binaires indiquant la présence/absence du discriminant dans les motifs décrits.

Une première exploitation des motifs discriminants consiste donc à étendre les descripteurs initiaux $\mathcal{F}$ avec les descripteurs discriminants trouvés à chaque itération. Cette stratégie d'exploitation prend en compte la sémantique de tous les rangements précédants dans la description des futurs motifs et permet une description de plus en plus précise des motifs d'un point de vue syntaxique (avec $\mathcal{F}$) et sémantique (avec les descripteurs discriminants). Cependant, l'augmentation sans limite du nombre de descripteurs pourraient complexifier le processus de fouille et l'exposer à un risque de sur-apprentissage. Pour pallier à cela, nous proposons d'utiliser les motifs discriminants comme des **descripteurs temporaires** $\mathcal{F}_{disc}$. Il s'agit d'ajouter temporairement ces descripteurs à $\mathcal{F}$ afin d'apprendre des poids $w_{\mathcal{F}}$. Les poids appris seront alors exploités pour mettre à jour la fonction d'apprentissage φ. Trois types de descripteurs discriminants peuvent être ajoutés à $\mathcal{F}$:

- F_{disc_X} : c'est un descripteur binaire utilisé pour marquer la présence/absence du discriminant dans un motif $X \in \mathcal{X}$: $\mathbf{F_{disc_X}} = \begin{cases} 1 \; si \; disc \subseteq X \\ 0 \; sinon \end{cases}$

- $F_{disc_\mathcal{T}}$: il s'agit d'un descripteur numérique représentant la fréquence du motif discriminant. $F_{disc_\mathcal{T}}$ peut alors prendre les valeurs suivantes : $\mathbf{F_{disc_\mathcal{T}}} = \begin{cases} sup_{\mathcal{D}}(disc)/|\mathcal{D}| \; si \; disc \subseteq X \\ 0 \; sinon \end{cases}$

- $F_{disc_\mathcal{I}}$ est un descripteur numérique qui représente la longueur du motif discriminant. $disc$; $F_{disc_\mathcal{I}}$ peut alors prendre les valeurs suivantes : $\mathbf{F_{disc_\mathcal{I}}} = \begin{cases} |disc|/|\mathcal{I}| \; si \; disc \subseteq X \\ 0 \; sinon \end{cases}$

En notant $\mathcal{F}_{disc}$ l'ensemble des descripteurs discriminants ajoutés à $\mathcal{F}$, on obtient le vecteur de descripteur temporaire suivant : $\mathcal{F}^* = \langle \underbrace{F_1, \ldots, F_n}_{\mathcal{F}}, \underbrace{F_{disc_X}, F_{disc_\mathcal{T}}, F_{disc_\mathcal{I}}}_{\mathcal{F}_{disc}} \rangle$.

3.3 Exploitation des descripteurs discriminants

Afin de mettre en exergue l'intérêt des discriminants $disc$ pour l'utilisateur, nous proposons de les utiliser dans une double mise à jour des poids associées aux descripteurs : (i) une première mise à jour classique effectuée avec une méthode d'apprentissage comme SCD (présenté par Shalev-Shwartz et Tewari (2011)) pour mettre à jour φ_{log} et $w_{\mathcal{F}}$; (ii) une deuxième mise à jour qui est spécifique aux poids des discriminants $\mathcal{F}_{disc}$ et qui consiste à agréger ces poids à ceux de $\mathcal{F}$. Pour la deuxième mise à jour, notre approche est inspirée de Arora et al. (2012), qui utilise une méthode multiplicative pour la mise à jour de poids w_i associés à des objets i. A chaque itération t, cette méthode met à jour les poids w_i^t des objets i de la manière suivante : $w_i^t = Agreg(w_i^t, c_i^t) = w_i^t \cdot \bigotimes(w_i^t, c_i^t)$ où $\bigotimes(w_i^t, c_i^t)$ est un facteur d'agrégation et c_i^t un gain (resp. pénalité) si $c_i^t \geq 1$ (resp. $c_i^t < 1$).

Nous proposons alors deux types d'agrégation des poids. Les détails sur la fonction f^{ag} utilisée sont donnés dans Hien et al. (2023b) et dans Hien (2022).

- Pour les descripteurs binaires (items et transactions), l'agrégation s'effectue comme suit :
  - o $w_{F_i}^t = f^{ag}(w_{F_i}^t, w_{F_{disc_X}}^t) \; \forall F_i \in \mathcal{F}_\mathcal{I} \wedge i \in \mathcal{I}[disc]$
  - o $w_{F_i}^t = f^{ag}(w_{F_i}^t, w_{F_{disc_X}}^t) \; \forall F_i \in \mathcal{F}_\mathcal{T} \wedge i \in \mathcal{V}_\mathcal{D}(disc)$

avec $\mathcal{F}_{\mathcal{I}}$ et $\mathcal{F}_{\mathcal{T}}$ les descripteurs des items et des transactions et $\mathcal{I}[disc]$ les items de $disc$.
- Pour les descripteurs numériques (fréquence, longueur, ...), on procède comme suit :
  - o $w^t_{F_{freq}} = f^{ag}(w^t_{F_{freq}}, w^t_{F_{disc_{\mathcal{T}}}})$, où F_{freq} représente le descripteur sur la fréquence.
  - o $w^t_{F_{length}} = f^{ag}(w^t_{F_{length}}, w^t_{F_{disc_{\mathcal{I}}}})$, avec F_{length} le descripteur de la longueur.

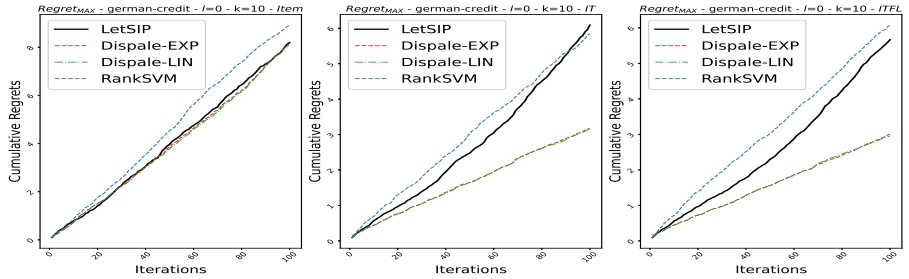
TAB. 1 – Évaluation et comparaison de DISPALE-EXP ($\eta = 0.13$) avec LETSIP et RANKSVM pour $k = 10$. (1) : LETSIP, (2) : DISPALE-EXP, (3) : RANKSVM.

	$\ell = 0$						$\ell = 1$					
	Regret : $Regret_{\max}$			Regret : $Regret_{Avg}$			Regret : $Regret_{\max}$			Regret : $Regret_{Avg}$		
Descripteurs	(1)	(2)	(3)	(1)	(2)	(3)	(1)	(2)	(3)	(1)	(2)	(3)
I	**112.137**	114.567	123.130	554.816	**553.050**	582.303	**10.438**	11.438	11.465	**496.918**	499.151	521.634
IT	108.446	**91.528**	101.635	543.556	**492.967**	542.595	10.761	11.465	**9.192**	483.689	**449.444**	491.014
ITLF	106.006	**88.391**	100.162	538.848	**487.537**	540.844	11.275	11.579	**9.601**	490.818	**450.202**	490.649

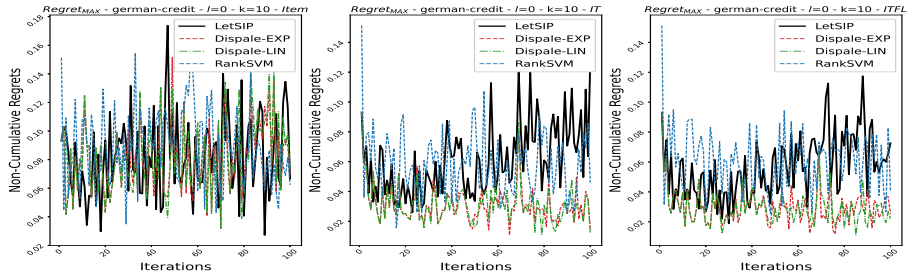
4 Expérimentations

a) Protocole expérimental. Pour évaluer DISPALE, notre outil de fouille interactive de motifs, nous émulons les préférences de l'utilisateur à l'aide d'une mesure de qualité Φ cachée. Pour chaque jeu de données, un ensemble $\mathcal{P}$ de motifs fréquents est extrait sans connaissances préalables des préférences de l'utilisateur. Nous supposons alors l'existence d'un ordre de préférence $\widehat{R}$ de l'utilisateur dérivé de Φ permettant de ranger les motifs de $\mathcal{P}$. Plus précisément : $\forall X, Y \in \mathcal{P}, X \succ Y \Rightarrow \Phi(X) > \Phi(Y)$. Ainsi, l'objectif est d'apprendre une fonction logistique φ_{log} qui approxime Φ, *i.e.,* $X \succ Y \Rightarrow \varphi_{log}(X) > \varphi_{log}(Y)$. Pour nos expérimentations, nous avons utilisé comme Φ la fonction *surprisingness* (surp) définie par : $surp(X) = \max\{sup_{\mathcal{D}}(X) - \prod_{i=1}^{|X|} sup_{\mathcal{D}}(\{i\}), 0\}$. Notre approche est alors évaluée en comparant ses performances avec celles de LETSIP (l'outil de Dzyuba et van Leeuwen (2017)) et RANKSVM (de Joachims (2002)). Pour cela, nous utilisons la mesure du regret qui évalue la capacité des méthodes à apprendre une fonctionφ_{log} permettant de sélectionner des motifs de $\mathcal{P}$ possédant les meilleurs rangs : plus la valeur obtenue est basse, plus la méthode est performante. Par soucis d'espaces, les détails de cette évaluation peuvent être retrouvées dans Hien et al. (2023b) et dans Hien (2022).
b) Résultats expérimentaux. Étant donné le nombre important de paramètres à évaluer pour DISPALE, la première étape de notre analyse a consisté à déterminer les paramètres permettant d'obtenir les meilleurs regrets. Ainsi, nous relevons que les meilleures valeurs de regret sont obtenues pour $\eta = 0.13$, pour une taille de requête $k = 10$ et l'utilisation de l'agrégation exponentielle dans DISPALE que nous notons DISPALE-EXP. Nos expérimentations, résumées dans la Table 1, montrent alors que DISPALE-EXP permet globalement d'avoir de meilleurs résultats que LETSIP et RANKSVM.
Dans la Figure 1, nous présentons une vue plus détaillée des résultats obtenus par les trois méthodes précédentes avec deux jeux de données (les autres résultats sont donnés dans Hien et al. (2023a)). Les graphiques montrent ainsi l'évolution du regret (de manière cumulative et non cumulative) au fil de 100 itérations d'apprentissage. Ils permettent de confirmer que les meilleures valeurs de regret sont obtenues avec DISPALE-EXP. En analysant les temps d'exécution des jeux de données GERMAN-CREDIT et CHESS (Figures 1c et 1d), on constate que,

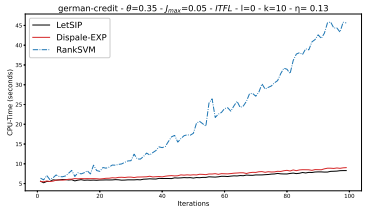
dans l'ensemble, l'ajout des descripteurs discriminants ne pénalise pas significativement les performances de notre outils DISPALE.



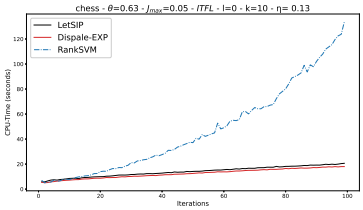
(a) GERMAN-CREDIT : regret cumulatif $\text{Regret}_{\max}$.



(b) GERMAN-CREDIT : regret non cumulatif $\text{Regret}_{\max}$.



(c) German-credit : temps d'exécution



(d) Chess : temps d'exécution

FIG. 1 – Vue détaillée des résultats de deux jeux de données avec $k = 10$ et $\ell = 0$.

5 Conclusion

Dans cet article, nous avons proposé une nouvelle approche de fouille interactive avec l'introduction de nouveaux descripteurs discriminants permettant une représentation plus expressive et dynamique des motifs. Les expérimentations que nous avons mené montrent la pertinence de notre outils, DISPALE, face aux méthodes de l'état de l'art, et sa capacité à améliorer la précision de l'apprentissage des préférences de l'utilisateur. Les descripteurs discriminants pourraient donc être utilisés dans d'autres contexte comme la fouille de graphes ou de motifs séquentiels.

Références

Agrawal, R. et R. Srikant (1994). Fast algorithms for mining association rules in large databases. In *Proceedings of 20th Int. Conference on Very Large Data Bases*, pp. 487–499.

Arora, S., E. Hazan, et S. Kale (2012). The multiplicative weights update method : a meta-algorithm and applications. *Theory Comput. 8*(1), 121–164.

Bhuiyan, M. et M. A. Hasan (2016). Interactive knowledge discovery from hidden data through sampling of frequent patterns. *Stat. Anal. Data Min. 9*(4), 205–229.

Dzyuba, V. et M. van Leeuwen (2017). Learning what matters - sampling interesting patterns. In *PAKDD 2017, Proceedings, Part I*, pp. 534–546.

Dzyuba, V., M. van Leeuwen, S. Nijssen, et L. D. Raedt (2014). Interactive learning of pattern rankings. *Int. J. Artif. Intell. Tools 23*(6), 1460026.

Hien, A., S. Loudni, N. Aribi, A. Ouali, et A. Zimmermann (2023a). Code and supplementary material. `https://gitlab.com/phdhien/dispale`.

Hien, A., S. Loudni, N. Aribi, A. Ouali, et A. Zimmermann (2023b). Interactive pattern mining using discriminant sub-patterns as dynamic features. In *Proceedings of the 27th PAKDD 2023*, pp. 252–263.

Hien, L. (2022). *Cadre interactif de fouille de motifs avec prise en compte des préférences de l'utilisateur*. Theses, Normandie Université.

Joachims, T. (2002). Optimizing search engines using clickthrough data. In *Proceedings of the 8th ACM SIGKDD Int. Conference on Knowledge Discovery and Data Mining*, pp. 133–142.

Morishita, S. et J. Sese (2000). Traversing itemset lattice with statistical metric pruning. In *Proceedings of the 19th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*.

Rüping, S. (2009). Ranking interesting subgroups. In A. P. Danyluk, L. Bottou, et M. L. Littman (Eds.), *Proceedings of ICML 2009*, Volume 382, pp. 913–920.

Shalev-Shwartz, S. et A. Tewari (2011). Stochastic methods for l_1-regularized loss minimization. *J. Mach. Learn. Res. 12*, 1865–1892.

Summary

Recent years have seen a shift from a pattern mining process that has users define constraints beforehand, and sift through the results afterwards, to an interactive one. This new framework depends on exploiting user feedback to learn a quality function for patterns. Existing approaches have a weakness in that they use static pre-defined low-level features, and attempt to learn independent weights representing their importance to the user. As an alternative, we propose to work with more complex features derived directly from the pattern ranking imposed by the user. Those features are used to learn weights to be aggregated with low-level features and help to drive the quality function in the right direction. Experiments on UCI datasets show that using higher-complexity features leads to the selection of patterns that are better aligned with a hidden quality function while being competitively fast when compared to state-of-the-art method