Wave Top-k Random-d Family Search : comment guider un expert dans un espace structuré

Etienne Lehembre, Bruno Cremilleux, Bertrand Cuissart, Abdelkader Ouali, Albrecht Zimmermann

Wave Top-k Random-d Family Search : comment guider un expert dans un espace structuré

Etienne Lehembre*, Bruno Cremilleux*
Bertrand Cuissart*, Abdelkader Ouali*, Albrecht Zimmermann*

*UNICAEN, ENSICAEN, CNRS - UMR GREYC, Normandie Univ 14000 Caen, France
{prenom.nom}@unicaen.fr

Résumé. Dans cet article, nous développons une méthode (WTRFS) incluant le retour utilisateur dans le but de le guider parmi les résultats d'une fouille de motifs. Ce travail vise à remplacer l'étape de déclaration des descripteurs utilisée dans la fouille interactive de motifs. Pour cela, la méthode s'appuie sur l'existence hypothétique d'un lien entre les différents motifs intéressants un expert. Nous montrons empiriquement que WTRFS renvoie rapidement les résultats les plus pertinents pour l'utilisateur. De plus, même si les retours de l'utilisateur sont imparfaits, le comportement de WTRFS n'en est pas altéré.

1 Introduction

Le but de la fouille de données est d'aider les experts de domaines applicatifs (ils ou elles) à analyser leurs données en leur montrant des associations d'intérêt. Lorsque ces résultats sont fournis sous la forme d'un ensemble de motifs saillants, un problème récurrent est la grande quantité de solutions fournies, souvent impossible à appréhender par un humain. Différentes approches traitent ce problème comme les représentations condensées de motifs qui synthétisent *l'espace des solutions* (Pasquier et al., 1999), les nombreuses *mesures de qualité* (Tan et al., 2004) et, plus récemment, les techniques de *fouille d'ensembles de motifs* (De Raedt et Zimmermann, 2007). Cependant, la combinaison de ces résultats reste insuffisante à rendre l'espace des solutions humainement abordable. Aussi, une proposition est d'intégrer l'expert au processus via une fouille qualifiée *d'interactive*.

Alors que plusieurs méthodes de fouille interactive de motifs traitent les données sous forme d'itemsets (Boley et al., 2013; Van Leeuwen, 2014), peu de travaux portent sur la recherche interactive de motifs à partir de données structurées, comme la fouille interactive de sous-graphes (Bhuiyan et Hasan, 2016; Bhuiyan et Al Hasan, 2016). De plus, même dans ces travaux, les sous-graphes sont traités comme des itemsets et les relations entre motifs sont peu exploitées. Les algorithmes considèrent les motifs comme un ensemble, sans exploiter la taille des sous-graphes pour induire leur degré de spécificité. Bien que certains travaux (van Leeuwen et al., 2016) travaillent à retranscrire l'intérêt subjectif dans la distribution de l'échantillonnage. Cette dernière est généralement impactée globalement et non localement. Pourtant, l'expert est sensible à ces paramètres locaux et son intérêt peut diverger lorsqu'il étudie deux régions distinctes de l'espace des solutions.

L'approche standard en fouille interactive apprend une approximation des préférences de l'expert en encodant les motifs via des descripteurs prédéfinis pour lesquels des poids sont appris. La création de descripteurs est ainsi une phase cruciale de ces méthodes. Si les descripteurs créés ne retranscrivent pas fidèlement les points saillants de l'ensemble des graphes étudiés, alors, cela conduira à un impact négatif sur le résultat de la fouille. De plus la méthode de définition des descripteurs peut elle-même être un obstacle. Ils peuvent être définis dans le code de l'algorithme, par l'expert à travers un éditeur fourni, ou produit par un réseau neuronal (Bhuiyan et Al Hasan, 2016). La première méthode requiert une compréhension du langage utilisé pour développer l'algorithme créant de ce fait une barrière pour modifier les descripteurs.La seconde méthode contraint l'expert à travers les outils de définition des descripteurs. Ces outils peuvent manquer de flexibilité ou de précision pour traduire convenablement l'intérêt de l'expert. De plus, ces deux méthodes requièrent de l'expert qu'il sache déjà ce qu'il recherche dans le jeu de données. Enfin, les réseaux neuronaux produisent généralement des vecteurs comme descripteurs dont l'interprétation est difficile et donc peu explicable. Les trois méthodes partagent un manque de flexibilité au cours de l'exploration. En effet, elles ne permettent pas de redéfinir les descripteurs au cours de l'exploration ce qui rend impossible toute adaptation à un changement d'avis de l'expert.

Notre méthode se concentre sur l'exploitation des propriétés de la fouille de sous-graphes. Son but est de structurer l'espace des solutions en exploitant les sous-graphes afin de l'utiliser pour échantillonner efficacement les propositions soumises à un expert. Nous identifions trois points cruciaux. Premièrement, il est important que la recherche des solutions ne soit pas restreinte à une sous-partie de la structure. Deuxièmement, il est possible d'exploiter la relation d'ordre partiel structurant l'espace des solutions afin diffuser l'intérêt subjectif de l'expert. Troisièmement, afin de diffuser correctement cet intérêt il est essentiel de fournir une interaction nuancée et graduée à l'utilisateur.

L'article est organisé comme suit. Le section 2 introduit les notions et notations nécessaires pour comprendre l'article. La section 3 détail l'algorithme. La section 4 décrit une expérience clef permettant de prendre du recul sur les résultats de la méthode. La section 5 résume les contributions de l'article.

2 Notations et notions préliminaires

Soit $\mathcal{D}$ l'ensemble de données, $\mathcal{L}$ le langage de motifs, et $\mathbb{G}(\mathbb{V}, \mathbb{E})$ un graphe où $\mathbb{V}$ est l'ensemble de sommets et $\mathbb{E}$ est l'ensemble d'arcs. Un POG (Partial Order Graph) modélise l'espace partiellement ordonné des motifs solutions (poset) de $\mathcal{L}$ dont l'ordre partiel est noté $<$. Par exemple, en analyse formelle de concepts, le POG serait un treillis et $<$ serait l'opérateur de fermeture (Kuznetsov et Obiedkov, 2001). Pour chaque sommet $v \in \mathbb{V}$, v contient un motif X pouvant être dans notre cas un sous-graphe ou un ensemble de sous-graphes. Chaque motif X possède un ensemble nommé support noté $Supp(X)$ contenant les éléments de $\mathcal{D}$ dans lesquels il apparaît : $Supp(X) : \{t \in \mathcal{D} \mid X < t\}$. On définit l'ensemble des arcs tel que :

$$\mathbb{E} = \{(v_1, v_2) \mid v_1, v_2 \in \mathbb{V}, v_1 < v_2, \nexists v_3 \in \mathbb{V} : v_1 < v_3 < v_2\}.$$

On appelle v_1 *parent* et v_2 *enfant*. On étend cette relation par transitivité aux parents des parents et aux enfants des enfants appelés respectivement *ancêtres* et *descendants*. On définit alors la *lignée* de v comme l'ensemble de ses ancêtres et descendants. On définit les *racines*

de $\mathbb{G}$ comme l'ensemble des sommets $v \in \mathbb{V}$ ne possédant aucun parent, la *distance* comme le nombre minimal d'arcs entre deux sommets du POG et une couche L comme un ensemble de v ayant la même distance des racines du POG, où la *profondeur* de la couche est définie par cette distance. Soit L une couche du graphe. On dit que la couche composée des parents des sommets de L et la couche composée des enfants de L sont ses *couches adjacentes*.

3 Méthode

Dans cette section, nous proposons un algorithme aillant pour but de retranscrire avec nuance l'intérêt de l'expert afin de l'accompagner dans son exploration. La méthode WTRFS (Wave Top-k Random-d Family Search) repose sur un ensemble de principes fondamentaux. Le premier est l'exploration en *vague* de la structure. Le second est la diffusion de l'intérêt de l'expert à travers son interaction en adoptant des actions radicales lorsque l'expert est certain de ses choix et des actions plus nuancées lorsque le doute est présent. Le dernier est l'exploitation de l'intérêt subjectif afin de modifier l'espace de recherche et composer les échantillons proposés à l'expert. Durant son exécution, l'algorithme conduit l'expert à des motifs portant son intérêt ou permettant de discriminer l'espace selon ce dernier, positivement ou négativement. Durant l'algorithme ou une fois celui-ci arrêté, l'utilisateur peut étudier une explication de son intérêt à travers le POG impacté par ses choix.

Afin d'intégrer l'intérêt de l'expert dans le POG, nous devons y ajouter quelques notions. Pour cela nous définissons le graphe d'ordre partiel de l'intérêt subjectif (SIPOG). Le SIPOG est défini comme $\mathbb{G}(\mathbb{V}, \mathbb{E}, \mathbb{V}^+, \mathbb{V}^-, poids)$ où $\mathbb{V}^+ \subset \mathbb{V}$ est le sous-ensemble des sommets prioritaires pour l'exploration, $\mathbb{V}^- \subset \mathbb{V}$ est le sous-ensemble des sommets exclus de l'exploration, et $poids : \mathbb{G} \to \mathbb{R}$ est la fonction $poids(v)$ qui associe à chaque sommet $v \in \mathbb{V}$ un nombre réel $j \in \mathbb{R}$ appelé poids. Nous détaillons par ce qui suit les principes sur lesquels repose l'algorithme WTRFS.

Le parcours en vague. Le parcours en vague commence aux éléments les plus généraux, ici les sous-graphes d'ordre minimal, et parcourt les couches du graphe jusqu'à atteindre les sous-graphes d'ordre maximal. Ce parcours amène l'expert à examiner des éléments de plus en plus spécifiques afin de confirmer sa compréhension des concepts généraux. Lors de la remontée, le parcours amène l'expert à confronté sa compréhension des éléments spécifique avec des éléments plus généraux qui les contiennent. Ce parcours itératif permet de diffuser l'intérêt de l'expert dans le graphe sans sauter d'étape. C'est-à-dire, sans observer d'élément décorrélé de l'espace observé.

Interaction	Conséquence·s	Couleur
Rejeté	Zone d'exclusion & Diminution des poids	Rouge
Non-intéressé	Diminution des poids	Orange
Incertain	rien	Violet
Intéressé	Augmentation des poids	Bleu
Accepté	Zone prioritaire & Augmentation des poids	Vert

TAB. 1 – *Interaction expert et son impact*

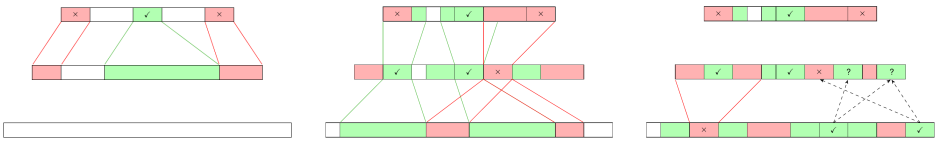
Interaction expert et diffusion de l'intérêt subjectif. Afin de diffuser l'intérêt de l'expert, il faut d'abord lui fournir un médium pour le transmettre : une interaction. Dans le tableau 1, on liste dans la colonne *Interaction* les réponses possibles de l'expert à un sommet proposé et dans la colonne *Conséquence·s* la ou les conséquences liées à l'interaction.

La conséquence la plus drastique présentée est la définition de zones prioritaires ou exclues de l'exploration. En effet, ces conséquences ont un impact important car les échantillons sont d'abord extraits dans les zones prioritaires puis dans les zones non-marquées mais jamais dans les zones exclues. Ces conséquences doivent donc être contenues aux descendants et ancêtres directs des sommets acceptés ou rejetés. Soit v_1 le motif présenté à l'expert :
Si il accepte v_1 :

$$\mathbb{V}^+ \leftarrow \mathbb{V}^+ \cup \{\forall v_2 \in \mathbb{V} | \exists (v_1, v_2) \in \mathbb{E} \text{ or } \exists (v_2, v_1) \in \mathbb{E}\} \tag{1}$$

Si il rejette v_1 :

$$\mathbb{V}^- \leftarrow \mathbb{V}^- \cup \{\forall v_2 \in \mathbb{V} | v_2 \notin \mathbb{V}^+ \text{ and } \exists (v_1, v_2) \in \mathbb{E} \text{ or } \exists (v_2, v_1) \in \mathbb{E}\} \tag{2}$$



(a) *(b)* *(c)*

FIG. 1 – *Illustration de la définition de zones prioritaires ou exclues dans le SIPOG.*

La figure 1 illustre la modification des zones de recherche décrites respectivement dans l'équation 1 et l'équation 2 comme conséquence des interactions "accepté" et "rejeté". L'étape *(a)* situe l'algorithme dans la couche la plus haute de l'espace des solutions, c'est-à-dire la couche contenant les sous-graphes les plus génériques. On soumet un échantillon de trois motifs à l'expert qui accepte un motif (marqué par une encoche) et en rejette deux (marqués par des croix). On définit alors dans la couche adjacente inférieur une zone de recherche prioritaire (en vert) et deux zones exclues de la recherche (en rouge). L'étape *(b)* place l'algorithme dans la couche intermédiaire. On échantillonne deux motifs provenant de la zone de recherche prioritaire, et un provenant d'une zone neutre. Cette fois, la définition des zones prioritaires ou exclues de la recherche se fait sur les deux couches adjacentes. L'étape *(c)*, situe l'algorithme dans la couche la plus basse de l'espace des solutions, contenant les éléments les plus spécifiques. Les interactions de l'expert définissent ici des zones spécifiques dans des couches adjacentes supérieures. Une fois l'interaction finie, l'algorithme WTRFS échantillonne dans la couche adjacente supérieure et remonte itérativement jusqu'à la première couche de l'espace de recherche. On note alors des conflits et des confirmations provenant de l'intérêt subjectif pouvant être exploités pour amorcer la compréhension de l'espace.

Notre seconde méthode de diffusion de l'intérêt est plus subtile. Elle consiste à modifier le poids des ancêtres et descendants du sommet portant l'interaction de manière à impacter les futurs échantillonnages. Cette modification dépendra donc de l'interaction de l'expert, à savoir si celle ci est positive (réponses "accepté" et "intéressé") ou négative (réponses "rejeté" et "non-intéressé") :

Soit v un sommet, A une interaction expert et λ un modificateur.

$$ponderation(v, A, \lambda) = \begin{cases} poids(v) + \lambda & \text{if } A \in \{\text{accepté, intéressé}\} \\ poids(v) - \lambda & \text{if } A \in \{\text{rejeté, non-intéressé}\} \end{cases} \quad (3)$$

Néanmoins, plus la distance entre deux sommets est grande, plus les motifs contenus diffèrent. Il faut ainsi considérer cette variation dans l'application de l'équation 3. Nous définissons la formule 4 où k est la distance entre un sommet v et v' son ancêtre (resp. descendant) tel que :

$$\forall v' \in \text{Ancêtres}(v) \cup \text{Descendants}(v) \cup \{v\}, ponderation(v', A, |poids(v)| * \frac{1}{2^k}) \quad (4)$$

Notons que la modification des poids affectés aux sommets modifie aussi leur impact lorsque l'expert interagit avec eux. Plus la valeur absolue du poids d'un sommet sera importante, plus l'interaction liée aura d'impact. Donc plus les sous-graphes et les sur-graphes d'un motifs auront un étiquetage uniforme, plus le motif aura d'impact, favorisant la non-ambiguïté.

Échantillonage des motifs. Disposant d'une propagation efficace de l'intérêt expert dans le graphe relationnel d'intérêt subjectif, nous exploitons cet intérêt pour échantillonner l'espace des solutions. Pour cela, nous exploitons le graphe d'ordre partiel d'intérêt subjectif et ses poids afin de calculer l'intérêt potentiel des sommets.

La probabilité d'un motif d'être échantillonné doit augmenter

— avec le poids cumulatif de ces ancêtres/descendants ayant des interactions positives (accepté/intéressant) parce que ce type d'interactions signifie une meilleure probabilité d'être accepté

— avec le poids cumulatif de ses ancêtres/descendants avec lesquels l'expert n'a pas encore interagi afin d'aider à l'exploration de l'espace

— si les poids cumulatif des ancêtres/descendants ayant des interactions positives et de ceux ayants des interactions négatives (rejeté, non-intéressant) sont similaires parce que ces informations contradictoires exigent plus d'exploration

À partir de ces considération, nous calculons l'intérêt d'un sommet v selon :

$$\begin{gathered} f_p(v, \mathbb{G}) = \sum_{i=0}^{k} (poids(L_i^+(v, \mathbb{G})) + poids(L_i^?(v, \mathbb{G})) + \\ (poids(L_i^\star(v, \mathbb{G})) - |poids(L_i^+(v, \mathbb{G})) - poids(L_i^-(v, \mathbb{G}))|) * \tfrac{1}{2^i} \end{gathered} \quad (5)$$

Dans la formule 5, $L_i(v, \mathbb{G})$ indique la lignée de v à distance i dans $\mathbb{G}$ avec :

— $poids(L_i^+(v, \mathbb{G}))$ la somme des poids de ceux qui portent une interaction positive,

— $poids(L_i^?(v, \mathbb{G}))$ la somme de poids de ceux qui ne portent pas d'interaction,

— $poids(L_i^*(v, \mathbb{G}))$ la somme des poids de ceux qui portent n'importe quelle interaction,

— $poids(L_i^-(v, \mathbb{G}))$ la somme des poids de ceux qui portent une interaction négative

— le facteur $\frac{1}{2^i}$ permettant de réduire l'influence des sommets en fonction de leur distance

Cette valeur est exploitée de deux manières dans l'échantillonnage. Premièrement, dans une volonté d'exploitation, l'échantillonnage sélectionne les k premiers motifs classés par f_p. Secondement, dans une volonté d'exploration, l'échantillonnage tire d motifs de manière pseudo aléatoire où la probabilité de sélection sera déterminé par f_p.

Soit $L \in \mathbb{G}$ une couche du graphe. La partie aléatoire de l'échantillonnage suit les probabilités définies comme suit :

$$\forall v \in L, P(v) = \frac{f_p(v)}{\sum\limits_{\forall v' \in L} f_p(v')} \tag{6}$$

Au début de l'exploration, les sommets favorisés par l'échantillonnage sont les sommets ayant le plus de parents, d'enfants, d'ancêtres et de descendants proches. Puisque qu'aucune interaction n'a encore eu lieu, la seule somme non-nulle lors du calcul de f_p est $poids(L_i^?(v, \mathbb{G}))$. Tous les sommets étant initialisés avec un poids identique cela implique que les sommets ayants la connexité proche la plus forte auront les meilleurs valeurs d'intérêt potentiel. Ce comportement nous intéresse car il permet de favoriser les sommets ayant un impact plus grand sur le SIPOG en début de recherche et donc d'atteindre rapidement une meilleure discrimination de l'espace de recherche.

Complexité des opérations sur les lignées. Soit n le nombre de sommets de $\mathbb{G}$ et m le nombre de sommets de la couche L de $\mathbb{G}$. La complexité de la pondération de la lignée et du calcul de f_p est de $O(n - m)$ par sommet de L.

Algorithme WTRFS. Ayant défini les éléments nécessaires à la compréhension de l'algorithme WTRFS, nous pouvons à présent le détailler.

L'algorithme 1 prend en entrée un SIPOG $\mathbb{G}$, un facteur d'exploitation k déterminant le nombre de tirage en tête, un facteur d'exploration d déterminant le nombre de tirage pseudo-aléatoires, et la profondeur minimale et maximale des couches à considérer. Tant que l'expert ne met pas fin au processus et qu'il reste des motifs à explorer, la boucle ligne 1 à 20 continue. Dans la boucle ligne 2 à 18 la valeur i commence à la profondeur f et termine à la profondeur l, i est incrémentée à chaque itération si $f < l$ ou décrémentée si $f > l$. À la ligne 3 on affecte à L la couche de $\mathbb{G}$ de profondeur i. Puis on assigne à chaque sommet v dans L son intérêt potentiel à travers la boucle ligne 4 à 6 en utilisant la formule 5. À la ligne 7, on affecte à $\mathbb{S}$ un échantillon de L en tirant k premiers motifs et d pseudo-aléatoires basés sur leur intérêt potentiel. La boucle ligne 8 à 17 itère sur chaque sommet v de l'échantillon et modifie le graphe $\mathbb{G}$ selon l'interaction A obtenue à la ligne ligne 9. Les poids de la lignée du sommet sont modifiés ligne 10 en utilisant la formule 4 et les ensembles de zones prioritaires $\mathbb{V}^+$ et de zones exclues $\mathbb{V}^-$ sont mis à jour ligne 12 and 15 en utilisant respectivement l'équation 1 et l'équation 2. Après une descente ou une montée complète, on échange les valeurs de f et l de manière à fouiller les couches dans le sens contraire à la ligne 19. Cela permet d'effectuer la recherche de haut en bas puis de bas en haut de manière alternative donnant une forme de vague à l'exploration. Une fois que l'expert est satisfait ou qu'il n'y a plus de motif pouvant être traité, on renvoi le graphe modifié ligne 21.

WTRFS produit deux résultats. Le premier est l'ensemble des motifs échantillonnés et étiquetés par l'expert. Le second est le graphe relationnel $\mathbb{G}$ sculpté par l'interaction de l'expert. Ce graphe, à travers ses zones de recherche prioritaires, ses zones d'exclusions, les poids des sommets, leur valeur d'intérêt potentiel, et leur étiquette est une représentation structurée de l'intérêt expert. Cette représentation peut être observée et étudiée et offre une vision *globale* de l'espace de recherche à partir des interactions *locales*. À travers cette représentation, l'expert

Algorithm 1 Wave Top-k Random-d Family Search

Require: $\mathbb{G}(\mathbb{V}, \mathbb{E}, \mathbb{V}^+, \mathbb{V}^-, poids)$ un graphe, k le nombre de tirages en tête, d le nombre de tirages aléatoires, f la première couche à explorer, l la dernière couche à explorer.
Ensure: $\mathbb{G}$ le graphe modifier par les interactions experts.

while $\exists v \in \mathbb{V} | v \notin \mathbb{V}^- \ \& \ v$ non exploré **do**
  for $i : f \rightarrow l$ **do**
    $L \leftarrow Layer_i(\mathbb{G})$
    for $v \in L$ **do**
      $v \leftarrow f_p(v, \mathbb{G})$ (Équation 5)
    end for
    $\mathbb{S} = \{$les k v'ayant le plus grand $f_p(v', \mathbb{G})\}$
    $\mathbb{S} = \mathbb{S} \cup \{d$ v''aléatoirement choisis selon Équation $6\}$
    for $v \in \mathbb{S}$ **do**
      $A \leftarrow Interaction(v)$
      Pondération_de_la_lignée$(\mathbb{V}, v, A)$ (Équation 4)
      if $A =$ accepté **then**
        $\mathbb{V}^+ \leftarrow \mathbb{V}^+ \cup \{\forall v_2 \in \mathbb{V} | \exists (v, v_2) \in \mathbb{E}$ or $\exists (v_2, v) \in \mathbb{E}\}$
      end if
      if $A =$ rejeté **then**
        $\mathbb{V}^- \leftarrow \mathbb{V}^- \cup \{\forall v_2 \in \mathbb{V} | v_2 \notin \mathbb{V}^+$ and $\exists (v, v_2) \in \mathbb{E}$ or $\exists (v_2, v) \in \mathbb{E}\}$
      end if
    end for
  end for
  $t \leftarrow f$; $f \leftarrow l$; $l \leftarrow t$
end while

peut explorer son propre intérêt, voir la relation entre les motifs qu'il a choisit de mettre en avant et ceux qui on été mis en retrait. Mais cette structure permet aussi d'évaluer les éléments qui n'ont pas été observés car leur poids et l'intérêt potentiel qui leur est assigné ont aussi été affectés durant l'exploration.

4 Expériences et résultats

Une difficulté intrinsèque à l'évaluation des méthodes de fouille interactive est que l'évaluation devrait solliciter un expert afin d'interagir avec le système et évaluer les résultats. Or, typiquement pour les jeux de données publiques, des experts ne sont pas disponibles. De plus, demander à l'expert d'effectuer des évaluations répétées afin d'avoir des résultats fiables et les valider d'une manière non-numérique, par exemple par des expériences biologiques, exige un investissement en temps et peut être extrêmement coûteux. La protocole adopté dans la littérature est donc de simuler les retours d'utilisateur par un *oracle omniscient* qui exploite une mesure de qualité objective afin d'étiqueter les motifs avec fidélité (Bhuiyan et Al Hasan, 2016; Bhuiyan et Hasan, 2016; Gyongyi et al., 2004).

Dans notre travail, on utilise la mesure de qualité Weighted Relative Accuracy WRAcc (Todorovski et al., 2000). Cette mesure est calculée à partir des classes des graphes formant l'en-

semble des données et est définie tel que :

$$WRAcc(X, \mathcal{D}) = \frac{Supp(X)}{|\mathcal{D}|} * (\frac{Supp(X)^+}{Supp(X)} - \frac{|\mathcal{D}^+|}{|\mathcal{D}|}),$$

où $\mathcal{D}^+$ est un sous-ensemble de $\mathcal{D}$ qui contient les données qui font partie d'une classe cible et $Supp(X)^+$ le support de X dans ce sous-ensemble.

Néanmoins, l'utilisation d'un tel oracle risque de donner une évaluation trop optimiste, particulièrement pour une méthode exploratrice comme la nôtre. Afin de pallier ce défaut, nous éprouvons notre méthode avec cinq autres types *oracles* simulant plusieurs comportements experts possibles. À notre connaissance, c'est la première fois qu'une méthode de fouille interactive est évaluée d'une telle manière.

Nous assignons à chaque sommet du graphe une *étiquette cachée* déterminée par la qualité du motif contenu. L'oracle, lui, assigne une *étiquette découverte* déterminée par la combinaison du type d'oracle et la valeur de qualité.

Les valeurs de la mesure de qualité sont transcrites avec les étiquettes cachées de manière à ce que les valeurs les plus basses de l'espace des solutions soient rejetées, les valeurs suivantes soient inintéressantes, etc. Les valeurs seuils sont calculées pour chaque espace des solutions de manières à respecter autant que possible la distribution suivante : 2.00% d'étiquettes *Rejeté*, 18.00% d'étiquettes *Inintéressant*, 60.00% d'étiquette *Incertain*, 18.00% d'étiquettes *Intéressant*, et 2.00% d'étiquettes *Accepté*. Cette distribution a pour but de représenter le fait qu'un expert n'est pas intéressé par l'ensemble des résultats, il utilisera moins les actions ayant beaucoup de conséquences et plus les autres.

	Rejeté	Inintéressant	Incertain	Intéressant	Accepté
Rejeté	80%	15%	5%	0%	0%
Inintéressant	10%	75%	10%	5%	0%
Incertain	5%	10%	70%	10%	5%
Intéressant	0%	10%	70%	10%	0%
Accepté	0%	0%	5%	15%	80%

TAB. 2 – *Distribution des probabilités de réponses pour l'oracle probabiliste*

Les cinq oracles sont :

1. *l'oracle omniscient* : il assigne à chaque sommet présenté son étiquette cachée.
2. *l'oracle probabiliste* : il possède pour chaque étiquette un vecteur de probabilité de réponse induisant un pourcentage d'erreur fixe. Afin de rester cohérent, chaque vecteur possède des probabilité de choix concernant chacune des étiquettes de façon à éviter les réponses improbables. L'idée est de donner le plus de probabilité à la bonne étiquette et des probabilités positives aux étiquettes similaires. Plus un choix aura d'impact moins la probabilité de se tromper sera grande car on considère que ces choix sont faits lorsque l'expert se sent sûr de lui. Les probabilités sont décrites dans la table 2 où chaque ligne correspond à un vecteur de probabilités dans lequel chaque colonne contient la probabilité que l'étiquette soit choisie.
3. *l'oracle biaisé* : il modélise un a priori de l'expert provenant de ses connaissances concernant des jeux de données étudiés par le passé on choisit d'utiliser une seconde

mesure de qualité dont le comportement diverge de celle choisie pour déterminer les étiquettes cachées. De cette façon, les erreurs commises par l'oracle gardent une cohérence par rapport au support des motifs. Dans cet article, la mesure choisie pour représenter le biais est la confiance. On fixe arbitrairement la marge d'erreur à 20%, c'est-à-dire de donner 20% de chance à l'oracle de choisir sa réponse d'après la valeur de qualité du biais plutôt que de la valeur de qualité de la vérité terrain.

4. *l'oracle localement subjectif* : il modélise le comportement qu'un expert a en ne prenant en compte un échantillon de manière locale. Cela l'amène à classifier l'échantillon proposé en considérant que le meilleur motif est au moins intéressant et que le pire motif est au moins inintéressant. L'oracle étiquette donc le motif dont la qualité est la plus haute comme intéressant si il n'est pas accepté et le motif dont la qualité est la plus basse comme inintéressant si il n'est pas rejeté.
5. *l'oracle subjectivement surpris* : il modélise un expert voulant explorer les motifs qui le surprennent, qu'ils aient une bonne qualité ou non. Afin de calculer cette surprise avec cohérence nous utilisons le sélecteur *Outstanding Pattern Selector* introduit dans l'article (Lehembre et al., 2022) dont les motifs sélectionnés seront automatiquement étiquetés comme *accepté* par l'oracle.

4.1 Jeux de données

Dataset	Graphes	Fréquence	Sous-graphes	Classes d'équivalence
AIDS	2,000	10%	192	192
BZR_MD	306	10%	3,249	2,147
MUTAG	188	10%	603	110
MCF-7	27,770	10%	1,024	1,024
Mutagenicity	4,337	10%	1,904	1,880
NCI-H23	40,353	10%	1,001	1,001

TAB. 3 – *Informations essentielles concernant les jeux de données TUDatasets, les sous-graphes extraits, et les classes d'équivalence formant le POG.*

Nous étudions six jeux de données ayant des caractéristiques différentes du répertoire TUDataset [1] possédant deux classes pour une facilité d'utilisation expérimentale. Les sous-graphes fréquents sont extraits avec *quickSpan* [2] avec un support minimal de 10%. On limite la taille des sous-graphes à sept sommets en supposant que des sous-graphes plus grands seront difficiles à interpréter. La table 3 liste les noms des jeux de données, leur taille, ainsi que le nombre de sous-graphes extraits et le nombre de classes d'équivalence.

Les classes d'équivalence sont calculées comme suit : si deux sous-graphes p et q ont le même support $Supp(p) = Supp(q)$ et qu'ils sont liés dans le POG par un chemin passant uniquement par des sous-graphes p_i ayant le même support $Supp(p) = Supp(p_i)$ alors ils font partie de la même classe d'équivalence. Ces classes permettent d'éviter les informations redondantes dans le POG pour ne pas montrer deux fois la même information à l'expert. Par la suite, chaque sommet du POG contiendra une classe d'équivalence qu'on appellera motif.

1. https ://chrsmrrs.github.io/datasets/docs/datasets/
2. https ://gitlab.inria.fr/Quickspan/quickspan

La dimension des espaces étudiés variant d'environ 200 motifs à quelque milliers. La méthode WTRFS explore avec 300 interactions entre 10% et 100% de l'espace de recherche lors des expériences. La variation de la proportion explorée de l'espace des motifs nous permet d'observer les comportements variables ou non de WTRFS par rapport à son espace d'application et d'avoir un indice sur son adaptabilité.

Protocole expérimental. Dans le but d'évaluer l'efficacité de WTRFS, on étiquette chaque classe d'équivalence dans le POG avec l'une des cinq interactions : Rejeté, Inintéressé, Incertain, Intéressé, Accepté. On soumet 100 échantillons de 3 motifs à chacun des oracles décrits plus tôt. On choisit de répartir les 3 motifs de l'échantillon en 2 motifs exploités ($k = 2$) et 1 motif exploré ($d = 1$). Comme les résultats incluent des éléments aléatoires, chaque couple jeu de données - oracle est réalisé une centaine de fois et les résultats observés sont la moyenne des résultats de ces cent itérations.

Afin d'interpréter nos résultats on étudie les étiquettes découvertes proposées à l'oracle et les étiquettes cachées. Soit A un type d'étiquette, on définit le *Rappel* tel que :

$$Rappel(A) = \frac{\text{Découvertes}(A)}{\text{Cachées}(A)}$$

Résultats. Poser 300 questions est difficile. On espère donc aider rapidement l'expert à découvrir le plus grand nombre possible de motifs acceptés, tout en lui soumettant quelques motifs à rejeter. Si d'autres motifs sont présentés à l'expert, ils doivent être intéressants.

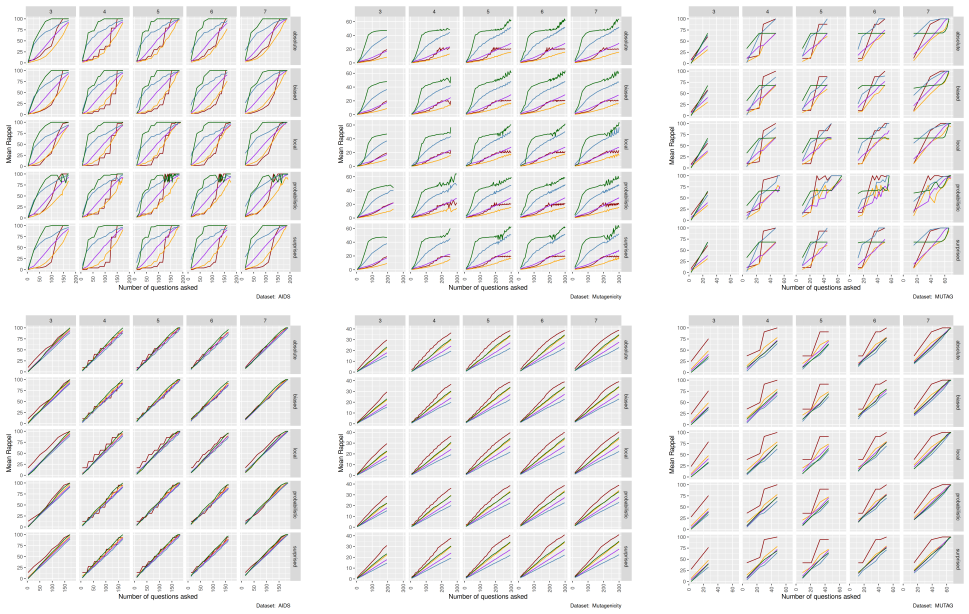
Dans la figure 2, pour chaque illustration, l'axe des abscisses indique le nombre de motifs proposés à l'oracle, et l'axe des ordonnés indique le rappel. Les colonnes correspondent aux couches du POG et les lignes correspondent aux types d'oracle. Les couleurs correspondent aux types des étiquettes (voir table 1). [3]

Les résultats montrent la quasi-constante progression du pourcentage d'étiquettes *accepté* retrouvées. Même si dans les réseaux les plus denses ces étiquettes ne sont pas toujours toutes retrouvées, on remarque que leur pourcentage de découverte reste plus haut que ceux des autres étiquettes quel que soit l'oracle observé. En comparant ces résultats aux résultats de la contenant les courbes d'un parcours WTRFS où les motifs sont échantillonnés au hasard, on note que les résultats de la figure 2 sont nettement meilleurs.

On note que pour le jeu de données *AIDS* la progression du Rappel, dans le cas de l'échantillonnage aléatoire, est linéaire et quasi-équivalente pour chaque type d'étiquette. Ce qui signifie que pour chaque couche, la distribution des étiquettes est équivalente.

Même si l'oracle omniscient arrive toujours aux meilleurs résultats, les autres oracles ne dégradent pas fortement la qualité des résultats. De plus, la courbe des *intéressants* augmente généralement plus rapidement que les autres. Les courbe des *rejetés* restent basses, soit pour la totalité de l'expérience, soit pendant une longue période. Le jeu MUTAG est une exception, sa courbe des éléments étiquetés *rejeté* excède celles des *acceptés* et *intéressants* après 20 à 30 questions selon oracle. Cela peut-être expliqué par la dimension remarquablement petite de l'espace des solutions. En effet, la courbe des éléments *accepté* stagne lorsque celle des *rejeté*

3. Prises indépendamment, les courbes de rappel sont strictement croissantes. Mais chaque parcours n'étant pas identique, toutes les courbes ne sont pas considérées au même moment dans la même couche. C'est pourquoi la courbe de la moyenne des résultats n'est pas forcément strictement croissante.



AIDS *Mutagenicity* *MUTAG*

FIG. 2 – *Moyenne du Rappel pour AIDS, MUTAG, et Mutagenicity avec WTRFS en haut et un parcours en vague avec échantillonage aléatoire en bas.*

s'envole, ce qui signifie que des zones sont exclues de la recherche. L'espace des solutions étant restreint, ces coupes impactent une proportion importante de l'espace de recherche et donc la découverte des autres éléments.

Les jeux de données, exécutables et l'ensemble des résultats sont disponibles à l'adresse suivante : `https://github.com/wtrfs/Wave-Top-k-Random-d-Family-Search/`.

5 Conclusion

Dans cet article nous présentons un algorithme dont le but est d'accompagner un expert durant son exploration d'un espace de solutions. Nos travaux se concentrent sur les motifs et espaces structurés, en particulier sur les motifs de graphes. Pour cela, nous présentons un parcours adapté aux objets étudiées se découpant en trois points essentiels : le parcours de la structure, la gestion des zones de recherche et l'échantillonnage. Nous déterminons cinq interactions et leur conséquences. Chaque couple interaction-conséquence influe sur l'espace de recherche accessible ou l'échantillonnage des motifs.

Pour évaluer notre méthode, nous avons simulé les retours d'un expert en utilisant des oracles se basant sur une mesure de qualité objective. Nous avons montré que la méthode recouvrent un nombre important de motifs de haute qualité, en fonction du nombre d'interactions avec l'oracle, même si les retours des oracles sont bruités. De plus, la méthode échantillonne peu de motifs de mauvaise qualité.

Références

Bhuiyan, M. et M. A. Hasan (2016). Interactive knowledge discovery from hidden data through sampling of frequent patterns. *Statistical Analysis and Data Mining : The ASA Data Science Journal 9*(4), 205–229.

Bhuiyan, M. A. et M. Al Hasan (2016). Priime : A generic framework for interactive personalized interesting pattern discovery. In *2016 IEEE International Conference on Big Data (Big Data)*, pp. 606–615. IEEE.

Boley, M., M. Mampaey, B. Kang, P. Tokmakov, et S. Wrobel (2013). One click mining : Interactive local pattern discovery through implicit preference and performance learning. In *Proceedings of the ACM SIGKDD workshop on interactive data exploration and analytics*, pp. 27–35.

De Raedt, L. et A. Zimmermann (2007). Constraint-based pattern set mining. In *Proceedings of the Seventh SIAM International Conference on Data Mining*. SIAM.

Gyongyi, Z., H. Garcia-Molina, et J. Pedersen (2004). Combating web spam with trustrank. In *Proceedings of the 30th international conference on very large data bases (VLDB)*.

Kuznetsov, S. O. et S. A. Obiedkov (2001). Algorithms for the construction of concept lattices and their diagram graphs. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pp. 289–300. Springer.

Lehembre, E., R. Bureau, B. Crémilleux, B. Cuissart, J.-L. Lamotte, A. Lepailleur, A. Ouali, et A. Zimmermann (2022). Selecting outstanding patterns based on their neighbourhood. In *International Symposium on Intelligent Data Analysis*, pp. 185–198. Springer.

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

Tan, P., V. Kumar, et J. Srivastava (2004). Selecting the right objective measure for association analysis. *Inf. Syst. 29*(4), 293–313.

Todorovski, L., P. Flach, et N. Lavrač (2000). Predictive performance of weighted relative accuracy. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pp. 255–264. Springer.

Van Leeuwen, M. (2014). Interactive data exploration using pattern mining. In *Interactive knowledge discovery and data mining in biomedical informatics*, pp. 169–182. Springer.

van Leeuwen, M., T. De Bie, E. Spyropoulou, et C. Mesnage (2016). Subjective interestingness of subgraph patterns. *Machine Learning 105*(1), 41–75.

Summary

In this paper, we develop a method (WTRFS) that includes the user interaction in order to guide her through data mining results. This work aims to replace the descriptor declaration step used in interactive data mining. For this we exploit the hypothetical relation between experts' patterns of interest. We empirically demonstrate that WTRFS returns first the most relevant results for the user. Moreover, even if the users' interactions are not perfect, we observed that WTRFS behavior isn't altered.