# Découverte de motifs centrée sur l'utilisateur

Arnaud Soulet

université de TOURS

Année universitaire : 2019-2020

Discipline : Informatique

**Dissertation en vue d'obtention d'une Habilitation à diriger des recherches**

*Découverte de motifs centrée sur l'utilisateur*

présentée et soutenue publiquement par


## Arnaud Soulet


22 novembre 2019

devant le jury suivant (par ordre alphabétique) :

| | | |
|---|---|---|
| Arnaud GIACOMETTI | Professeur des universités | Université de Tours, France |
| Amedeo NAPOLI | Directeur de recherche CNRS | LORIA (CNRS – INRIA – Université de Lorraine), Nancy, France |
| Céline ROBARDET | Professeur des universités | Institut National des Sciences Appliquées de Lyon, France |
| Marie-Christine ROUSSET | Professeur des universités | Université de Grenoble Alpes, France |
| Gerd STUMME | Professeur des universités | Universität Kassel, Allemagne |
| Christel VRAIN | Professeur des universités | Université d'Orléans, France |

# Remerciements

En premier lieu, j'exprime toute ma gratitude à Céline Robardet, Marie-Christine Rousset et Gerd Stumme qui m'ont fait l'honneur d'être les rapporteurs de mon mémoire. Je les remercie d'avoir pris le temps de rapporter ce manuscrit malgré leurs obligations nombreuses. Je suis également très honoré de la présence dans mon jury d'Amedeo Napoli et Christel Vrain. Je les remercie vivement pour leur participation.

Les contributions de ce mémoire sont avant tout la synthèse d'un travail d'équipe. J'ai eu la chance de collaborer directement avec de nombreux collègues qui appartiennent ou ont appartenu à l'équipe BDTLN, notamment au cours des thèses de Eynollah, Marie, Damien, Adnan et Lamine. Je les remercie tous sans oublier l'équipe administrative et technique, et les autres collègues de l'IUT qui font de Blois un environnement de travail convivial et épanouissant.

Ce mémoire doit aussi beaucoup aux nombreux échanges résultant de collaborations régionales à internationales, courtes ou dans la durée mais toujours enrichissantes. Je tiens tout particulièrement à remercier Bruno Crémilleux avec qui, année après année, nous continuons à échanger et à faire avancer des problématiques (parfois anciennes en faisant le voeu que la bonne science – à la manière du bon vin – se bonifie en vieillissant).

Je suis profondément reconnaissant à Arnaud pour son soutien généreux et indéfectible depuis mon premier jour à Blois. Au-delà de ses qualités scientifiques, je continue d'apprécier sa bienveillance et sa persévérance exemplaires face aux aléas qui jalonnent les années universitaires.

Au-delà du cadre du travail, j'adresse enfin un remerciement à Brunehilde et Ladislas qui m'accompagnent dans la vie. Merci de votre indulgence lorsque le regard perdu, je suis un peu ailleurs.

# Table des matières

# Chapitre 1

# Introduction

**Qu'est-ce que la découverte de motifs ?** En 1993, Rakesh Agrawal, Tomasz Imielinski et Arun N. Swami ont publié l'un des articles phares de la découverte de motifs [2] : « Mining association rules between sets of items in large databases » dans les actes de ACM SIGMOD International Conference on Management of Data en introduisant le problème de l'extraction de règles d'association intéressantes. Formellement, ce problème consiste à énumérer toutes les règles de la forme $X \rightarrow I$ où $X$ est un ensemble d'items et $I$ un item absent de $X$ tel que les probabilités $P(X, I)$ et $P(I|X)$, estimées respectivement par le support et la confiance, soient suffisamment élevés. Depuis, la découverte de motifs est devenu un sous-domaine important de la découverte des connaissances dans les bases de données puisqu'elle concerne environ un article sur six selon notre étude [41, 40]. Environ 20% des auteurs de cinq des conférences majeures ont contribué à au moins une publication sur la découverte de motifs.

Cet article fondateur [2] a surtout initié une école de pensée fortement influencée par le domaine des bases de données. Contrairement aux précédentes approches heuristiques [28], une attention particulière est portée à la complétude de l'énumération en plus de l'exactitude. En effet, la découverte de motifs est vue comme un problème de satisfaction de contraintes où tous les motifs $X$ vérifiant la contrainte $q$ dans $\mathcal{D}$ sont énumérés : $\mathbf{P}_{\text{sat}}(X, \mathcal{D}) \Leftrightarrow q(X, \mathcal{D}) = true$. Localement, cela signifie que pour chaque motif extrait $X$ (satisfaisant $\mathbf{P}_{\text{sat}}$), le motif vérifie $q$ dans le jeu de données $\mathcal{D}$ (sens direct). La complétude de l'extraction garantit qu'un motif non-extrait ne vérifie pas $q$ (sens indirect).

Dans ce mémoire, nous synthétiserons nos travaux en découverte de motifs qui vont au-delà de la satisfaction de contraintes. Néanmoins, dans la lignée de [2], nous montrerons que la singularité des travaux en découverte de motifs est de garantir une propriété *exacte* **P** sur les motifs extraits et surtout, sur les motifs non-extraits.

**Importance de l'utilisateur** La découverte de motifs est basée sur deux dimensions clés que chaque nouvelle proposition doit prendre en compte : le langage et l'intérêt [86]. Fondamentalement, le langage définit la syntaxe des motifs à découvrir, tandis que la mesure de l'intérêt indique la sémantique des motifs recherchés.

— **Langage :** Le langage est le domaine de définition des motifs énumérés. Bien que la plupart des méthodes considèrent les règles d'association et les itemsets, une tendance claire s'est manifestée dès le début des années 2000. L'évolution du nombre d'articles publiés montre une concentration de la communauté sur des langages plus complexes tels que les séquences ou les sous-graphes (cf. le graphique de gauche sur la figure 1.1). A l'instar des travaux en intelligence artificielle, une relation de spécialisation sur le langage permet l'apprentissage de concepts par induction [67]. Cette relation de spécialisation détermine si un motif est plus général qu'un autre. De plus, lorsque les motifs à apprendre ont une nature distincte de celles des données, une relation de couverture renseigne les exemples couverts par un motif donné.

— **Intérêt :** Une fois que le langage et sa relation de spécialisation sont définis, il reste à définir quels sont les motifs intéressants. Dans la plupart des cas, l'intérêt d'un motif est évalué par une mesure. Par exemple, la fréquence d'un motif (i.e., son nombre d'occurrences au sein du jeu de données) est souvent utilisée pour juger de son importance. Intuitivement, un motif qui apparaît dans de nombreuses observations des données est jugé plus intéressant. Toutefois, cette mesure ne couvre pas toutes les sémantiques possibles (par exemple, un contraste ou un motif rare) et la fréquence a tendance à renvoyer des motifs non-significatifs. Ces deux obstacles ont motivé un grand nombre de travaux sur les mesures d'intérêt. Le graphique de droite (cf. figure 1.1) décrit l'évolution de la sémantique des motifs recherchés. *Regularity*, *contrast* et *significant* désignent respectivement les travaux cherchant des régularités (principalement des motifs fréquent), des contrastes entre deux classes et des motifs significatifs. Enfin, *generic* correspond aux travaux dédiés à des classes de contraintes. On observe bien la décroissance des régularités au profit des motifs significatifs.

Les travaux synthétisés dans ce mémoire se concentrent essentiellement sur le second point. Plus précisément, nous cherchons à mieux qualifier l'intérêt des motifs en mettant au centre l'utilisateur. L'idée est de considérer que l'intérêt d'un motif est subjectif et que deux utilisateurs ne seront pas forcément intéressés par les mêmes motifs [23]. De manière générale, cette façon d'envisager l'analyse de données rejoint d'autres travaux que nous avons menés sur le clustering [37, 38] ou la personnalisation de requêtes OLAP [44, 43].

**Contributions** Ainsi, ce mémoire rapporte nos contributions en découverte de motifs centrée sur l'utilisateur en s'attaquant à quatre aspects :

— **Déclarativité :** Les approches déclaratives (e.g., certains travaux des bases de données inductives recensés dans l'ouvrage [34]) ambitionnent d'améliorer l'accessibilité de la découverte de motifs. Ces dernières offrent à l'utilisateur la possibilité d'exprimer ses attentes sur les motifs recherchés sans avoir à se préoccuper de la méthode d'extraction. Dans ce contexte, notre proposition étend l'algèbre relationnelle de sorte à pouvoir directement extraire et manipuler les motifs. Nous la nommons algèbre relationnelle orientée motif (ou plus simplement PORA pour Pattern-Oriented Relational Algebra). Pour montrer l'intérêt de PORA, ce mémoire

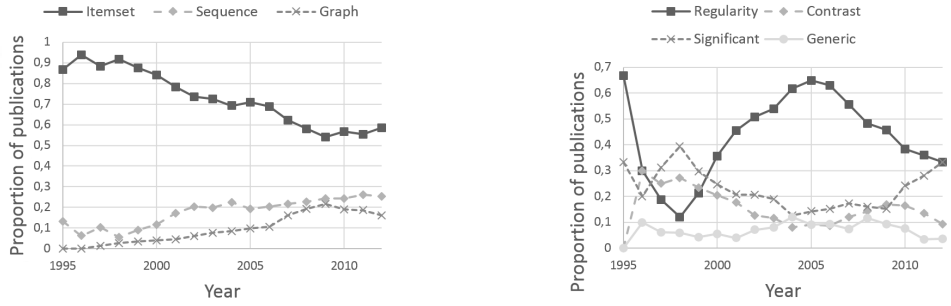



FIGURE 1.1 – Evolution du nombre de publications suivant le langage (à gauche) et l'intérêt (à droite) en se basant sur 1087 articles consacrés à la découverte de motifs (issus des 6 888 articles publiés dans les 5 conférences majeures de l'exploration de données : KDD, PKDD, PAKDD, ICDM et SDM) [40]

revisite la majorité de nos travaux en utilisant directement cette algèbre. Au-delà des aspects déclaratifs, nous verrons qu'il s'agit aussi d'un formalisme puissant pour optimiser la découverte de motifs en réécrivant les requêtes. De plus, définir ce qui peut être exprimé ou non avec PORA permettra de démarquer la découverte de motifs locaux par rapport aux bases de données ou aux modèles globaux.

— **Préférences explicites :** Une façon efficace de satisfaire un utilisateur est de tenir compte de ses préférences. Dans notre contexte, une relation de préférences se résumera à une relation d'ordre partiel $\prec$ où $Y \prec X$ signifie que $X$ est préféré à $Y$. Nous explorerons deux types de relations de préférences explicitées directement par l'utilisateur. Premièrement, la relation d'ordre de Pareto pour un ensemble de mesures $\{m_1, \ldots, m_N\}$ considère que $X$ est préféré à $Y$ si on a $m_i(X) \geq m_i(Y)$ pour chacune des mesures. Deuxièmement, dans le cadre de la construction de modèles, nous considèrerons qu'un motif $X$ est préféré à un motif $Y$ si la qualité de $X$ est supérieure à celle de $Y$ et que $Y$ ne couvre aucun exemple du jeu de données non-couvert par un motif de meilleure qualité.

— **Préférences implicites :** Formuler ses préférences pour un utilisateur est une tâche délicate. Pour cette raison, bon nombre de systèmes visent à construire le modèle de préférences de l'utilisateur en exploitant ses retours. Nous montrerons comment apprendre un profil de préférences contextuelles sur un ensemble de transactions (e.g., des films) à partir d'un ensemble de retours binaires (i.e., ce film $t$ est préféré à ce film $u$). Nous montrerons aussi comment mettre en oeuvre l'apprentissage actif pour apprendre un modèle de préférences sur des motifs (où une partie des transactions est préférée aux autres). Dans ce cadre, le système choisit les motifs pour lesquels il souhaite un retour alors que précédemment cette collection

initiale était donnée.

— **Couplage système-utilisateur :** Une interaction entre un utilisateur et un système nécessite des réponses dans un temps très court. Malgré l'obsession de la vitesse, les méthodes classiques de découverte de motifs ne parviennent pas à atteindre cet objectif. Dans ce contexte, nous proposons plusieurs techniques instantanée de découverte de motifs s'appuyant sur l'échantillonnage de motifs en sortie. Cette technique consiste à tirer aléatoirement un motif avec une probabilité proportionnelle à son intérêt. Nous montrons comment échantillonner des motifs depuis des données complexes ou avec contraintes. Nous utilisons aussi l'échantillonnage afin de construire des modèles fondés sur des motifs de manière anytime (i.e., une réponse disponible à tout instant tend vers la solution exacte). Ce mémoire sera aussi l'occasion d'intégrer l'échantillonnage de motifs à l'algèbre relationnelle orientée motif.

Ces contributions se sont largement appuyées sur les travaux des thèses de Eynollah Khanjari Miyaneh, Marie N'Diaye, Damien Nouvel, Mouhamadou Saliou Diallo, Adnan El Moussawi et Lamine Diop, pour lesquelles j'ai participé à l'encadrement.

**Organisation du mémoire** Ce mémoire se divise en trois parties principales qui synthétisent nos principales contributions concernant la découverte de motifs centrée sur l'utilisateur [1].

Le premier chapitre présente l'algèbre relationnelle orientée motif (PORA) qui sera utilisée dans l'ensemble du mémoire. Après quelques rappels préliminaires, nous introduisons l'opérateur de domaine pour générer des hypothèses sur les données et ce dernier distingue selon nous l'apprentissage de la simple manipulation de données. Il introduit aussi l'opérateur de couverture primordial pour comparer les hypothèses aux données (induction) et pour classer les motifs suivant des préférences (domination). Au-delà de la déclaration d'un processus de fouille, nous montrons qu'il est possible de déduire certaines propriétés de ces requêtes. Par exemple, nous reformulerons de manière algébrique l'algorithme par niveau.

Le second chapitre concerne nos travaux où les préférences de l'utilisateur guident la découverte de motifs. En d'autres termes, la découverte de motifs est envisagée comme un problème d'optimisation où seuls les meilleurs motifs au sens d'une relation de préférences sont retenus. Pour cela, le principal opérateur de l'algèbre relationnelle orientée motif, l'opérateur de couverture, est mis en oeuvre pour jouer le rôle de relation de préférences afin de comparer les motifs deux à deux en vue de conserver les meilleurs. Finalement, nous nous intéressons à la construction de modèle pour avoir une meilleure complémentarité entre les motifs extraits.

Le troisième chapitre concerne nos travaux où la méthode d'analyse des motifs guide leur découverte. Nous faisons l'hypothèse qu'en pratique, les motifs sont analysés avec une acuité proportionnelle à leur mesure d'intérêt. Plutôt que de tous les extraire, il suffit

1. Même si cette synthèse ne développe pas les expérimentations (seules quelques illustrations sont présentées), cela ne signifie pas que nous sous-estimons l'importance des applications et validations. D'ailleurs, nous reviendrons sur les aspects applicatifs en conclusion.

de les tirer et de les présenter à l'utilisateur avec une probabilité proportionnelle à leur mesure d'intérêt. Cette approche correspond exactement à un processus d'échantillonnage de motifs. Nous reformulons cette technique d'échantillonnage de manière algébrique et nous en proposons plusieurs implémentations physiques. Nous montrons comment utiliser l'échantillonnage de motifs pour la construction anytime de modèles fondées sur des motifs. Nous finissons par mettre en oeuvre l'échantillonnage de motifs pour l'apprentissage actif d'une mesure d'intérêt subjective.

Le dernier chapitre conclut sur l'ensemble de notre travail. Nous rappelons alors les résultats obtenus en les discutant. Nous proposons aussi plusieurs directions de recherche comme prolongement.

# Chapitre 2

# Algèbre relationnelle orientée motif

Considérons la tâche populaire d'extraction des motifs fréquents [3] comme un exemple de motivation. La plupart des travaux traitent cette tâche comme une « boîte noire » dont les paramètres d'entrée sont définis par l'utilisateur à la manière de la notion de théorie définie dans [66] : $\mathcal{T}h(\mathcal{L}, freq(X) \geq f, \mathcal{D})$. Au lieu de spécifier uniquement le seuil de fréquence minimal $f$ et le jeu de données $\mathcal{D}$, nous pensons que la requête de l'utilisateur devrait formaliser complètement la notion de motifs fréquents en indiquant comment la fréquence d'un motif est calculée à partir du jeu de données. Idéalement, cette requête serait exprimée en algèbre relationnelle afin de pouvoir manipuler simultanément les données et les motifs [60, 12]. Pour rester déclaratif, un processus d'extraction de motifs doit être entièrement spécifié sans considérer les aspects algorithmiques. Pour cette raison, les opérateurs de boucle ne sont pas pertinents de notre point de vue [18].

*D*

| id | trans |
|---|---|
| $t_1$ | ACD |
| $t_2$ | ABD |
| $t_3$ | ABCE |
| $t_4$ | CD |
| $t_5$ | AB |

Jeu de données

*L*

| patt |
|---|
| ∅ |
| A |
| B |
| C |
| D |
| E |
| AB |
| AC |
| AD |
| AE |
| BC |
| BD |
| BE |
| CD |
| CE |
| DE |
| ABC |
| ... |
| ABCDE |

Langage des itemsets

*I*

| item | type | price |
|---|---|---|
| A | *snack* | 3 |
| B | *snack* | 10 |
| C | *beer* | 5 |
| D | *soda* | 8 |
| E | *soda* | 6 |

Description des items

*F*

| patt | freq |
|---|---|
| A | 4 |
| B | 3 |
| C | 3 |
| D | 3 |
| AB | 3 |
| AC | 2 |
| AD | 2 |
| CD | 2 |

Motifs fréquents

TABLE 2.1 – Exemple illustratif

Faisons l'hypothèse que $L[patt]$ et $D[id, trans]$ sont deux relations [1] qui contiennent respectivement le langage et le jeu de données comme proposées dans la table 2.1. Il faut parvenir à calculer la fréquence de chaque motif de $L$. Le produit cartésien de $L$ par $D$

1. La formalisation de l'algèbre relationnelle est présentée dans la section 2.1.

associe chaque motif de $L$ avec chaque transaction de $D$. Bien sûr, seules les combinaisons où le motif est contenu dans la transaction sont pertinentes : $\sigma_{patt \subseteq trans}(L \times D)$. Finalement, nous comptons pour chaque motif combien de transactions le contiennent et nous ne conservons que les motifs fréquents : $\sigma_{freq \geq f}(\gamma_{patt,\texttt{COUNT}(trans) \rightarrow freq}(\sigma_{patt \subseteq trans}(L \times D)))$. A ce stade, on pourrait croire à tort que l'algèbre relationnelle est suffisamment expressive pour exprimer une requête d'extraction de motifs.

Après quelques rappels préliminaires concernant l'algèbre relationnelle et la découverte de motifs, ce chapitre ajoute deux opérateurs à l'algèbre relationnelle pour former l'algèbre relationnelle orientée motif (*Pattern Oriented Relational Algebra* ou simplement, PORA). Au-delà des aspects déclaratifs, nous montrerons qu'il est possible de déterminer certaines propriétés des requêtes orientées motifs et de les optimiser via des réécritures. Cette contribution publiée dans [46] [2] s'inscrit dans un mouvement plus large sur les langages formels au sein de l'axe Entrepôts et Fouille de données de l'équipe BDTLN (avec d'autres contributions dont [6]).

2. Cet article est présent en annexe.

## 2.1 Préliminaires

Cette section présente les principales définitions et notations qui seront utilisées dans la suite de ce mémoire.

**Algèbre relationnelle** Nous indiquons ici nos notations pour l'algèbre relationnelle principalement inspirées par [1]. Soit un ensemble de littéraux distincts **att**, nommés *attributs*, $\mathbf{dom}(A)$ dénote le *domaine* fini de l'attribut $A \in \mathbf{att}$. La notation $R[U]$ dénote une relation nommée $R$ avec le schéma $U \subset \mathbf{att}$. Une *instance* de $R$ est un sous-ensemble avec répétitions de $\mathbf{dom}(U) = \times_{A \in U} \mathbf{dom}(A)$. Etant donnée une relation $R[A_1, \ldots, A_n]$, $R'$ renomme les attributs $A_1, \ldots, A_n$ en $A'_1, \ldots, A'_n$. Un *schéma de base de données* est un ensemble non-vide et fini $\mathbf{R} = \{R_1[U_1], \ldots, R_n[U_n]\}$ de relations. Une *instance de base de données* de $\mathbf{R}$ est une ensemble $\mathbf{I} = \{I_1, \ldots, I_n\}$ tel que $I_i$ soit une instance de la relation $R_i[U_i]$. Par exemple, la table 2.1 en page 11 présente 3 instances correspondant aux relations $D[id, trans]$, $L[patt]$ et $I[item, type, price]$. Finalement, une *requête* $q$ associe à une instance de base de données une instance de relation. L'ensemble d'attributs de cette relation est noté par $\mathbf{sch}(q)$. Dans la table 2.1, on a $\mathbf{sch}(F) = \{patt, freq\}$ pour la requête des motifs fréquents. Pour deux requêtes $q'$ et $q$ de même schéma, $q'$ est *équivalente* à $q$, noté par $q' \equiv q$, ssi pour n'importe quelle instance de base de données $\mathbf{I}$, on a $q'(\mathbf{I}) = q(\mathbf{I})$.

Soit $I$ une instance de $R$ et $J$ une instance de $S$. Les relations peuvent être manipulées par les opérateurs sur les ensembles tels que le produit cartésien $R \times S$ où $I \times J = \{(t, u) | t \in I \wedge u \in J\}$. Si $R$ et $S$ sont des relations avec un même schéma, alors $R \cup S$, $R \cap S$ et $R - S$ sont respectivement l'union, l'intersection et la différence de $R$ et $S$. *Sélection :* $\sigma_f(I) = \{t | t \in I \wedge f(t)\}$ retient les tuples de $I$ satisfaisant la formule logique $f$ où $f$ est construit avec (i) les opérateurs logiques ($\wedge$, $\vee$ et $\neg$), (ii) les opérateurs de comparaison arithmétique et (iii) les opérandes fondées sur les attributs et les constantes. *Projection étendue :* $\pi_{A_1,\ldots,A_n}(I) = \{t[A_1, \ldots, A_n] | t \in I\}$ conserve seulement les attributs $A_1, \ldots, A_n$ de $R$. De plus, la projection permet aussi d'étendre la relation avec des expressions arithmétiques et de (re)nommer des expressions. Par exemple, $\pi_{A+B \to B', C \to C'}(R)$ créée une nouvelle instance où le premier attribut nommé $B'$ résulte de l'expression arithmétique $A + B$ et le second attribut correspond à $C$, renommé $C'$. *Agrégation :* $\gamma_{A_1,\ldots,A_n,\texttt{AGG}(B)}(I) = \{(a_1, \ldots, a_n, \texttt{AGG}(\pi_B(\sigma_{A_1=a_1 \wedge \cdots \wedge A_n=a_n}(I))) | (a_1, \ldots, a_n) \in \pi_{A_1,\ldots,A_n}(I)\}$ groupe les tuples de $I$ selon les attributs $A_1, \ldots, A_n$ et applique une fonction d'agrégat `AGG` sur les valeurs de $B$.

**Découverte de motifs** Nous introduisons ici quelques notions pour compléter l'algèbre relationnelle [66]. Un *langage* $\mathcal{L}$ est un ensemble de motifs comme les motifs ensemblistes [3] (cf. la table 2.1), les séquences [4] et bien d'autres [9]. Une *relation de spécialisation* $\preceq$ d'un langage $\mathcal{L}$ est une relation d'ordre partiel sur $\mathcal{L}$ [66, 67]. Etant donnée une relation de spécialisation $\preceq$ sur $\mathcal{L}$, $l \preceq l'$ signifie que $l$ est plus général que $l'$, et $l'$ est plus spécifique que $l$. Par exemple, l'inclusion est une relation de spécialisation sur le langage des itemsets $2^{\mathcal{I}}$ où $\mathcal{I}$ est un ensemble de littéraux. De même, nous pouvons considérer le langage des séquences qui sont des ensembles ordonnés d'itemsets. Par exemple, $\langle (ab)c(ac) \rangle$ correspond

à une séquence de 3 itemsets signifiant que l'itemset $ab$ est suivi par $c$ suivi par $ac$. Une sous-séquence $s'$ est incluse dans une séquence $s$, noté $s' \sqsubseteq s$, ssi chaque itemset de $s'$ est inclus dans un itemset de $s$ en maintenant l'ordre. Par exemple, $\langle(ab)(a)\rangle$ est une sous-séquence de $\langle(ab)c(ac)\rangle$ alors que ce n'est pas le cas pour $\langle c(ab)\rangle$. La relation $\sqsubseteq$ est donc une relation de spécialisation pour les séquences.

Etant donnés deux ensembles partiellement ordonnés $(\mathcal{L}_1, \preceq_1)$ et $(\mathcal{L}_2, \preceq_2)$, une relation binaire $\lhd \subseteq \mathcal{L}_1 \times \mathcal{L}_2$ est une *relation de couverture* ssi quand $l_1 \lhd l_2$, on a $l'_1 \lhd l_2$ (resp. $l_1 \lhd l'_2$) pour tout motif $l'_1 \preceq_1 l_1$ (resp. $l_2 \preceq_2 l'_2$). La relation $l_1 \lhd l_2$ signifie que $l_1$ couvre $l_2$, et $l_2$ est couvert par $l_1$. La relation de couverture est utile pour mettre en relation deux langages ensembles (notamment pour lier des motifs aux données). Il est à noter qu'une relation de spécialisation sur $\mathcal{L}$ est aussi une relation de couverture sur $\mathcal{L}$ (e.g., l'inclusion est une relation de couverture pour les itemsets). Mais des relations de couvertures plus complexes peuvent être envisagées. Par exemple, on peut définir la relation de couverture $\lhd_{seq}$ entre les itemsets et les séquences où l'itemset $X$ couvre la séquence $s$ ssi tous les items de $X$ apparaissent dans la séquence $s$ (e.g., $BC \lhd_{seq} \langle(AB)C(AC)\rangle$). Il est clair qu'on a bien les relations suivantes : $(\forall X' \subseteq X)(X \lhd_{seq} s \Rightarrow X' \lhd_{seq} s)$ et $(\forall s \sqsubseteq s')(X \lhd_{seq} s \Rightarrow X \lhd_{seq} s')$.

## 2.2 Déclarer des requêtes de découverte de motifs

Cette section montre comment il est possible de modéliser la découverte de motifs en utilisant l'algèbre relationnelle. Pour cela, elle introduit les deux principaux opérateurs spécifiques à la découverte de motifs et illustre leur utilisation pour la découverte de règles de préférences contextuelles.

**Construire les hypothèses** La découverte de motifs est une technique d'apprentissage inductive où les motifs extraits visent à généraliser les données. Par conséquent, une requête d'extraction de motifs nécessite une relation dont l'instance contient toutes les généralisations possibles des données afin qu'elle puisse jouer le rôle d'espace de recherche. Par chance, c'est le cas de l'instance de la relation $L$ dans la table 2.1. Pour que cela soit toujours le cas, nous introduisons l'opérateur de domaine pour garantir que l'instance d'une table contienne bien toutes les généralisations :

**Définition 1 (Opération de domaine)** *Le domaine d'une relation $R[U]$ est $\delta(R)$ où pour toute instance $I$ de $R$, $\delta(I) = \mathbf{dom}(U)$.*

Par exemple, avec la relation $L[patt]$, la requête $\delta(L)$ correspond à tous les motifs possibles quelque soit l'instance. Evidemment, il n'est pas envisageable de matérialiser $\delta(L)$ en pratique qui nécessiterait une taille considérable en mémoire. Nous verrons comment il est possible de réécrire une requête pour éviter cette matérialisation dans la section 2.3 (cf. le théorème 1).

Cet opérateur qui sature l'instance avec son domaine est très particulier et il étend l'expressivité de l'algèbre relationnelle. Un opérateur comparable avait déjà été envisagé pour l'algèbre imbriquée [56]. Concernant la découverte de motifs, le cadre de la programmation par domination [72] introduit également un tel opérateur pour générer les itemsets. De tels opérateurs évitent de recourir à des opérateurs explicitant une boucle itérative [18].

**Confronter les hypothèses aux faits** Une fois l'espace des hypothèses construit, il faut choisir les bonnes. Pour cela, au sein de la requête $\sigma_{freq \geq s}(\gamma_{patt,\texttt{COUNT}(trans) \rightarrow freq}(\sigma_{patt \subseteq trans}(L \times D)))$, l'opération $\sigma_{patt \subseteq trans}()$ compare le motif à la transaction et retient les lignes où le motif est contenu dans la transaction. C'est cette opération qui modélise l'induction en confrontant les hypothèses aux données. De manière plus générale, nous proposons un opérateur qui peut s'appuyer sur n'importe quelle relation de couverture :

**Définition 2 (Opération de couverture)** *La couverture d'une relation $R[U]$ pour une relation $S[V]$ par rapport à $\triangleleft$ définie sur $\mathbf{dom}(\widetilde{U}) \times \mathbf{dom}(\widetilde{V})$ (où $\widetilde{U} \subseteq U$ et $\widetilde{V} \subseteq V$) est $R \triangleleft S = \sigma_{\widetilde{U} \triangleleft \widetilde{V}}(R \times S)$, i.e. pour toutes instances $I$ de $R$ et $J$ de $S$, $I \triangleleft J = \{(t,u) | t \in I \wedge u \in J \wedge t[\widetilde{U}] \triangleleft u[\widetilde{V}]\}$.*

Ainsi, la requête des motifs fréquents s'écrit $\sigma_{freq \geq f}(\gamma_{patt,\texttt{COUNT}(trans) \rightarrow freq}(\delta(L) \triangleleft D))$ où $\triangleleft$ est la relation de couverture pour comparer les motifs aux données (e.g., l'inclusion

pour les itemsets). Dans cet exemple, on a $\widetilde{\{patt\}} = \{patt\}$ et $\widetilde{\{id, trans\}} = \{trans\}$ car la relation de couverture ne porte pas sur l'identifiant de transaction. L'opération de couverture, qui est une thêta-jointure particulière, n'augmente pas l'expressivité de l'algèbre relationnelle. Pourtant, cette opération est l'opération centrale dans de nombreuses requêtes d'extraction de motifs pour induire à partir des données ou comparer des motifs entre eux.

A l'instar de l'opérateur de jointure en algèbre relationnelle, nous considérons la semi-couverture et l'anti-couverture. L'opérateur de semi-couverture et d'anti-couverture retournent tous les tuples d'une relation qui couvrent respectivement *au moins* un tuple de l'autre relation et *aucun* tuple de l'autre relation. Plus formellement, la *semi-couverture* (resp. l'*anti-couverture*) d'une relation $R[U]$ pour une relation $S[V]$ par rapport à $\lhd \subseteq \mathbf{dom}(\widetilde{U}) \times \mathbf{dom}(\widetilde{V})$ est $R \lhd_{\ltimes} S = \pi_U(R \lhd S)$ (resp. $R \lhd_{\neg} S = R - R \lhd_{\ltimes} S$). Un motif de $L$ est soit présent dans $D$ (i.e., dans $L \lhd_{\ltimes} D$) ou absent de $D$ (i.e., dans $L \lhd_{\neg} D$). Pour cette raison, nous obtenons que $L = L \lhd_{\ltimes} D \cup L \lhd_{\neg} D$ (voir la table 2.2). Plus généralement, la semi-couverture et l'anti-couverture sont complémentaires par définition : $R = R \lhd_{\ltimes} S \cup R \lhd_{\neg} S$ pour toutes relations $R$ et $S$.

| $L \lhd_{\ltimes} D$ |
|---|
| patt |
| $\emptyset$ |
| A |
| B |
| C |
| D |
| E |
| AB |
| AC |
| AD |
| AE |

| |
|---|
| BC |
| BD |
| BE |
| CD |
| CE |
| ABC |
| ABD |
| ABE |
| ACD |
| ACE |
| BCE |
| ABCE |

| $L \lhd_{\neg} D$ |
|---|
| patt |
| DE |
| ADE |
| BCD |
| BDE |
| CDE |
| ABCD |
| ABDE |
| ACDE |
| BCDE |
| ABCDE |

TABLE 2.2 – La semi-couverture et l'anti-couverture de $L$ pour $D$

L'opérateur de couverture autorise des utilisations variées suivant les relations couvertes. Il permet de faire de l'induction avec la requête des motifs fréquents $\sigma_{freq \geq f}(\gamma_{patt, \texttt{COUNT}(trans) \rightarrow freq}(\delta(L) \lhd D))$ où les hypothèses de $\delta(L)$ sont comparées aux données $D$. Cette requête est illustrée dans la table 2.1 avec $f = 2$. Il permet aussi d'utiliser d'autres sources de données comme la relation $I$ de la table 2.1. Par exemple, la requête $(\delta(L) \lhd_{\ltimes} D) \ni_{\neg} \sigma_{type=snack}(I)$ énumère tous les motifs apparaissant dans le jeu de données et ne contenant pas un produit de type ≪ snack ≫. De même, la requête $\sigma_{total \leq t}(\gamma_{patt, \texttt{SUM}(price) \rightarrow total}(I \in (\delta(L) \lhd_{\ltimes} D)))$ énumère tous les motifs apparaissant dans le jeu de données dont la somme des prix est inférieure à $t$. Dans le chapitre 3, nous verrons également que l'opérateur de couverture est utile pour sélectionner les meilleurs motifs au sens d'une relation de préférences.

**Règles de préférences contextuelles** Jusqu'ici, nous avons illustré PORA avec des instances où les motifs sont des itemsets et où la relation $\lhd$ est la relation $\subseteq$. Il est possible de considérer n'importe quel langage muni d'une relation d'ordre partiel (e.g., les motifs

séquentiels [46]). Nous allons maintenant considérer les règles de préférences contextuelles qui seront notamment utiles dans la section 3.2. Une telle règle, représentée par $\langle t, t^+, t^- \rangle$, signifie que pour deux tuples $p$ et $n$, si une transaction $p$ contient les caractéristiques $t \cup t^+$ et que la transaction $n$ contient les caractéristiques $t \cup t^-$, alors $p$ est préférée à $n$. $t$ est appelé le contexte de la règle car il s'agit du contexte dans lequel la règle peut être appliquée. Par exemple, la règle $\langle A, D, E \rangle$ signifie qu'un tuple contenant $D$ est préféré à un tuple contenant $E$ dans le contexte où les deux contiennent $A$. Nous introduisons le langage des règles de préférences avec la relation $L_{pref}[cont, pref, non]$ qui regroupe les tuples de la forme $\langle t, t^+, t^- \rangle$ où $t \cap t^+ = \emptyset$, $t \cap t^- = \emptyset$ et $t^+ \cap t^- = \emptyset$. De plus, en complément du jeu de données $D$ de la table 2.1, nous introduisons une relation $P[pref, non]$ qui contient les préférences d'un utilisateur où le tuple $\langle t_1, t_3 \rangle$ signifie que la transaction $t_1$ est préférée à la transaction $t_3$.

$P$

| pref | non |
|---|---|
| $t_1$ | $t_3$ |
| $t_2$ | $t_3$ |
| $t_2$ | $t_4$ |
| $t_3$ | $t_4$ |
| $t_4$ | $t_5$ |

$D \bowtie P \bowtie D'$

| trans | pref | non | trans' |
|---|---|---|---|
| ACD | $t_1$ | $t_3$ | ABCE |
| ABD | $t_2$ | $t_3$ | ABCE |
| ABD | $t_2$ | $t_4$ | CD |
| ABCE | $t_3$ | $t_4$ | CD |
| CD | $t_4$ | $t_5$ | AB |

TABLE 2.3 – Exemples d'instances pour les préférences

**Définition 3 (Couverture positive et négative)** *Les relations de couverture positive* $\lhd^+$ *et négative* $\lhd^-$ *sont définies pour tout tuples* $\langle t, t^+, t^- \rangle$ *et* $\langle p, n \rangle$ *:*

$$\langle t, t^+, t^- \rangle \lhd^+ \langle p, n \rangle \Leftrightarrow (t \cup t^+) \subseteq p \wedge (t \cup t^-) \subseteq n$$

$$\langle t, t^+, t^- \rangle \lhd^- \langle p, n \rangle \Leftrightarrow (t \cup t^-) \subseteq p \wedge (t \cup t^+) \subseteq n$$

La couverture positive correspond à la couverture des exemples positifs (i.e., qui respectent la règle) tandis que la couverture négative met en relation la règle avec ses contre-exemples. Grâce à ces relations, il est par exemple possible d'extraire les règles dont la couverture positive est jugée suffisamment large :

$$P_{pref} := \sigma_{pos \geq s}(\gamma_{cont,pref,non,\texttt{COUNT}(*) \rightarrow pos}(\delta(L_{pref}) \lhd^+ (D \bowtie P \bowtie D')))$$

Avant de poursuivre, notons que l'expression $D \bowtie P \bowtie D'$ remplace la relation $D$ dans la requête des motifs fréquents vue auparavant et montre bien l'intérêt de pouvoir manipuler les motifs et les données au sein d'un même formalisme. Par exemple, cette requête calculera que la règle $\langle A, D, E \rangle$ a une couverture positive de 2 dans le résultat de la requête ci-dessus car les deux premières préférences de l'instance de $P$ satisfont cette règle. Pour $t_1$ et $t_3$, on constate bien que $A \cup D$ est inclus dans la transaction $t_1$ (i.e., $ACD$) et $A \cup E$ est inclus dans la transaction $t_3$ (i.e., $ABCE$).

Avec des requêtes comme $P_{pref}$, il est aisé de calculer la cardinalité de la couverture positive (ici, *pos*) et négative (*neg*). Nous montrons ainsi comment le cadre du support

($supp = pos/|P|$) et de la confiance ($conf = pos/(pos + neg)$) peut s'étendre aux règles de préférences contextuelles. Par exemple, la règle $\langle A, D, E\rangle$ a un support de $2/5$ et une confiance de 1 car dans le contexte $A$, le tuple contenant $D$ est toujours préféré à celui contenant $E$. Hélas, la requête $P_{pref}$ retourne beaucoup trop de règles et nous verrons dans la section 3.2.2 comment retenir un ensemble compact de règles pour former un profil de préférences qui résume bien l'ensemble des préférences d'un utilisateur.

## 2.3 Raisonner avec les requêtes orientées motifs

L'algèbre relationnelle orientée motif est un langage déclaratif qui se prête bien à la formalisation d'une extraction de motifs. Au delà de la déclaration, il est possible de raisonner sur les requêtes PORA. Cette section introduit la notion de dépendance qui s'inscrit dans une vaste discussion sur la notion de contrainte globale. Nous montrons ensuite comment caractériser l'anti-monotonie d'une requête PORA afin de l'optimiser.

**Dépendance** Une problématique récurrente de la découverte de motifs est de déterminer si les motifs extraits ont une portée locale ou globale [24]. Un premier point de vue est de considérer la signature de la méthode d'extraction ou de la contrainte pour connaître le niveau de la portée [97]. Une limite de cette approche est que les motifs extraits peuvent nécessiter d'exploiter un modèle plus complexe caché. A l'inverse une signature peut faire apparaître des entrées qui ne sont pas utiles. Pour cette raison, un autre point de vue est de considérer le nombre de motifs nécessairement évalués pour retourner le résultat d'une requête [21] [3]. Par exemple, la requête des motifs fréquents nécessite une seule évaluation de valeur de fréquence par tuple de l'instance finale alors que pour calculer les motifs fermés fréquents (cf. page 24 pour une définition), il en faut davantage car chaque motif est comparé à ses spécialisations. La complexité en évaluation des motifs fréquents est en $O(1)$ tandis que la complexité en évaluation des motifs fermés est en $O(k)$ où $k$ est la cardinalité de l'itemset. Dans ce contexte, la complexité en évaluation est d'autant plus élevée que le motif est global.

Avec PORA, nous ne distinguons pas la notion de portée locale ou globale sur la requête dans son ensemble mais sur chacune des relations de la requête i.e., que la requête dépende localement/globalement d'une relation donnée. Nous introduisons même un troisième niveau pour indiquer qu'une requête ne dépend pas d'une relation :

**Définition 4 (Indépendance totale)** *Une requête $q$ est totalement indépendante de $R$ ssi pour toutes instances $I$, $J$ de $R$, on a $q(I) = q(J)$.*

**Définition 5 (Dépendance locale et globale)** *Une requête $q$ est globalement indépendante de $R$ ssi pour toutes instances $I$, $J$ de $R$, on a $q(I \cup J) = q(I) \cup q(J)$. Une requête qui est à la fois globalement indépendante de $R$ mais dépendante de $R$ est dite localement dépendante de $R$.*

Typiquement, la requête des motifs fréquents $\sigma_{freq \geq s}(\gamma_{patt,\texttt{COUNT}(trans) \rightarrow freq}(L \triangleleft D))$ est indépendante de la relation $I$ (dont une instance est montrée dans la table 2.1) car cette relation n'intervient pas dans l'expression de la requête. Evidemment, cette requête dépend à la fois du langage $L$ et des données $D$. Il est possible de tester séparément si deux motifs sont fréquents et donc, il est possible de subdiviser l'instance de $L$ en deux. A l'inverse, le calcul de la fréquence d'un motif requiert de considérer simultanément tous les tuples de $D$. Par conséquent, la requête des motifs fréquents est localement dépendante de $L$ mais globalement dépendante de $D$.

3. Cet article est disponible en annexe.

**Anti-monotonie et optimisation** L'algèbre relationnelle autorise l'optimisation logique des requêtes grâce à la réécriture de requêtes. Par exemple, lorsque c'est possible, il est préférable d'appliquer les restrictions avant les jointures pour diminuer le coût de l'évaluation de la requête. Il est alors naturel de s'interroger sur l'existence de potentielles optimisations spécifiques à PORA. En particulier, ne pourrait-on pas exprimer la condition d'élagage de l'algorithme Apriori comme une simple règle de réécriture ?

Pour ce faire, nous commençons par traduire la notion d'anti-monotonie en PORA :

**Définition 6 (Clôture par le bas)** *Une requête $q$ est close par le bas dans $R[U]$ par rapport à $\preceq$ ssi $U \subseteq \mathbf{sch}(q)$ et $(R \preceq_{\ltimes} q) \equiv \pi_U(q)$.*

En d'autres termes, une requête $q$ est close par le bas dans $R[U]$ si un motif plus général qu'un motif appartenant à la réponse de $q$ appartient aussi à cette réponse. Par exemple, la requête des motifs fréquents est clos par le bas dans $L[patt]$ par rapport à $\preceq$. La clôture par le bas est utilisée par l'algorithme par niveau pour élaguer l'espace de recherche. Plus précisément, tous les motifs plus spécifiques qu'un motif absent de la réponse d'une requête $q$ seront également absent de la réponse. Nous reformulons cette idée avec notre algèbre :

**Théorème 1 (Réécriture par niveau)** *Une requête $q$ close par le bas par rapport à $\preceq$ et globalement indépendante de $R$, vérifie l'égalité suivante pour toute instance $I$ de $R$ :*

$$q(I) = \underbrace{q(\underbrace{I \succ_{\neg} I}_{\mathcal{C}:=})}_{\mathcal{S}:=} \cup q((I \succ_{\ltimes} \mathcal{S}) \succeq_{\neg} (\mathcal{C} \succeq_{\neg} \mathcal{S}))$$

$\mathcal{C} := I \succ_{\neg} I$ correspond aux motifs candidats du premier niveau (i.e., les motifs les plus généraux). La requête $q$ est évaluée sur cet ensemble de candidats dont les motifs retenus sont $\mathcal{S}$. Ensuite, il faut évaluer tous les motifs des niveaux suivants en appliquant la requête sur les motifs à la fois plus spécifiques qu'un motif retenu de $\mathcal{S}$ (i.e., $I \succ_{\ltimes} \mathcal{S}$) et dont aucun sous ensemble n'ait été rejeté (i.e., appartenant à $\mathcal{C} \succeq_{\neg} \mathcal{S}$). Bien sûr, cette règle peut s'appliquer récursivement pour distinguer un deuxième niveau, un troisième, etc.

Comme l'illustre le théorème 1, PORA est un outil puissant pour formaliser ce qui peut parfois être vu comme des astuces algorithmiques. La rigueur d'un tel formalisme est cependant utile pour expliciter certaines hypothèses. Par exemple, le théorème 1 stipule qu'il est nécessaire d'avoir une dépendance globale sur $R$ (ce qui explique qu'il faille adapter l'algorithme par niveau [66] pour les contraintes « globales » telles que la recherche des $k$ motifs les plus fréquents).

## 2.4 De la séparation à la comparaison

Les bases de données inductives prônaient de séparer les motifs (ou modèles) du reste des données. Cette séparation impliquait des opérateurs spécifiques pour générer les motifs et modèles, pour les manipuler et pour les appliquer à nouveau sur les données. A contrepieds, l'algèbre orientée motif met les motifs et les données sur un même plan ; ces derniers sont comparables. Plus précisément, cette algèbre permet de déclarer des requêtes d'extraction de motifs, mais aussi de combiner les motifs aux données. De manière intéressante, cette algèbre peut être utilisée pour manipuler d'autres langages que les motifs ensemblistes comme les séquences ou les règles de préférences. Au-delà des aspects déclaratifs, il s'agit d'un formalisme pour raisonner sur les requêtes. Par exemple, nous avons raffiné la notion de contrainte globale en distinguant plusieurs niveaux de dépendances et ce par rapport à chaque relation. Nous avons aussi montré comment reformuler algébriquement l'élagage de l'algorithme Apriori. Pour montrer les capacités de PORA, nous utiliserons principalement ce formalisme dans la suite [4].

La distinction d'expressivité entre une requête d'extraction de motifs et une requête d'interrogation de données se résume à l'usage de l'opérateur de domaine. Pourtant, l'opérateur de couverture est central car il autorise l'induction (comparaison de motifs aux données) et la domination (comparaisons de motifs entre eux). L'intérêt de la domination qui consiste à retenir les meilleurs motifs au sens d'une relation de préférence, est largement exploré dans le chapitre suivant.

4. Pour alléger le vocabulaire, la distinction entre instance et relation ne sera pas spécifiée si le contexte est clair. Par exemple, nous utiliserons le terme « les données $D$ » aussi bien pour désigner la relation $D$ qu'une instance de $D$.

# Chapitre 3

# Découverte de motifs guidée par les préférences

Ce chapitre concerne nos travaux où les motifs découverts sont les meilleurs au sens d'une relation de préférences. Les méthodes d'extraction de motifs sous contraintes requièrent de fixer des seuils ce qui s'avère souvent difficile en pratique. Il est possible d'aboutir à des résultats extrêmes : soit aucun motif (contraintes trop sélectives et insatisfiables), soit une collection pléthorique où les meilleurs motifs sont noyés au milieu des autres. Plutôt que d'envisager la découverte de motifs comme un problème de satisfaction, l'idée est donc de basculer sur un problème d'optimisation. De manière schématique, les meilleurs motifs correspondent à la plus petite collection qu'on obtiendrait avec des seuils élevés. Plus formellement, tous les motifs maximisant une relation de préférence $\prec$ (où $Y \prec X$ signifie que $X$ est préféré à $Y$) sont extraits : $\mathbf{P}_{\text{opt}}(X, \mathcal{D}) \Leftrightarrow (\forall Y)(X \not\prec Y)$. Localement, cela signifie que pour chaque motif extrait $X$ (satisfaisant $\mathbf{P}_{\text{opt}}$), il n'y a pas un autre motif $Y$ meilleur que lui pour le jeu de données $\mathcal{D}$ (sens direct). La complétude de l'extraction garantit qu'un motif non-extrait est préféré par au moins un motif extrait (sens indirect). La relation de préférence peut être un ordre total/faible induit par une simple mesure d'intérêt ou un ordre partiel plus sophistiqué.

Ce chapitre commence dans la section 3.1 par illustrer l'utilisation de l'opérateur de couverture introduit dans le chapitre précédent pour extraire les motifs préférés, i.e. ceux qui ne sont pas dominés. Nous revisiterons notamment l'extraction de représentations condensées et nous nous intéresserons au cas particulier de la relation Pareto. Pour améliorer la complémentarité entre les motifs extraits, la section 3.2 aborde la construction de modèles. Notre algèbre orientée motif n'est pas suffisamment expressive pour ces méthodes et nous recourons donc à un algorithme procédural. Nous illustrons son fonctionnement avec la construction de profil de préférences.

## 3.1 Préférences pour guider l'extraction

Comme indiqué en introduction de ce chapitre, l'extraction de motifs satisfaisant une contrainte comme la requête $\sigma_{freq \geq s}(\gamma_{patt,\texttt{COUNT}(trans) \rightarrow freq}(\delta(L) \triangleleft D))$ (exigeant une fréquence minimale $s$ sur les motifs) requiert des seuils avec un fort impact sur le volume de motifs retournés, difficile à anticiper par l'utilisateur. Ainsi, de nombreux problèmes en découverte de motifs se modélisent par le calcul des meilleurs motifs de $L$ au sens d'une relation de préférence $\preceq$ (ou relation de domination) i.e., tous ceux qui ne sont pas dominés sont préférés. En PORA, cela se traduit naturellement de la manière suivante :

$$L \prec_{\neg} L$$

A noter que cette expression retenant les plus grands motifs au sens de $\prec$ est traditionnellement dénotée par $\max_{\prec} L$.

Dans ce contexte, l'extraction du meilleur motif (ou top-1) pour une mesure $m$ s'obtient avec la relation de préférence $<^m$ où $a <^m b \Leftrightarrow m(a) < m(b)$. Nous verrons qu'il est possible d'étendre cette approche à un ensemble de mesures $M$ pour extraire les motifs Pareto-optimaux (sous-section 3.1.2). Les représentations condensées peuvent aussi être vues comme une préférence pour les maximaux des classes d'équivalence (motifs fermés) ou pour les minimaux des classes d'équivalence (motifs libres), voir la sous-section 3.1.1.

A nouveau, évaluer cette requête est une tâche extrêmement coûteuse à cause de la taille de $L$ qui est en général très grande et qui conduit à un nombre de comparaisons prohibitif. Il est alors absolument nécessaire de réduire l'espace de recherche en éliminant des motifs assurément dominés et/ou d'éviter des comparaisons superflues. Intuitivement, le calcul des motifs préférés peut se restreindre à n'importe quel sur-ensemble même s'il comporte peu ou pas de motifs dominés. En effet, si un motif est dominé, c'est qu'il existe au moins un motif préféré pour le dominer. Cette propriété se traduit formellement ainsi :

**Propriété 1** *Pour un langage $L$ et une relation de préférence $\prec$, on a la propriété suivante :*

$$(\forall Patt \subseteq L)\big((L \prec_{\neg} L) \subseteq Patt \Rightarrow (L \prec_{\neg} L) \subseteq (Patt \prec_{\neg} Patt)\big)$$

Cette propriété signifie que si un ensemble $Patt \subseteq L$ contient tous les motifs préférés de $L$ (i.e., $L \prec_{\neg} L$), alors il est suffisant d'évaluer $Patt \prec_{\neg} Patt$ pour calculer exactement ces motifs préférés. Nous allons mettre en oeuvre cette propriété de deux manières. Pour commencer, nous expliquerons pour les motifs minimaux comment il est possible de bénéficier de la propriété 1 au fur et à mesure de l'extraction. Ensuite, pour les motifs Pareto-optimaux, nous approximerons $L \prec_{\neg} L$ par un sur-ensemble obtenu grâce à une relaxation de la relation de préférence visée.

Plusieurs travaux présentés dans les sections 3.1.1 et 3.1.2 ont été réalisés en collaboration avec des équipes issues du CERMN, du GREYC, du LIRIS et du LORIA.

### 3.1.1 Représentations condensées adéquates

L'objectif premier de voir la découverte de motifs comme un problème d'optimisation est de réduire significativement le nombre de motifs extraits. La notion de représentation

condensée est l'une des premières propositions en ce sens [80] qui faisait écho à des travaux plus anciens notamment en analyse formelle de concepts [99, 39]. Il s'agit de conserver uniquement les motifs maximaux ou minimaux au sens de l'inclusion pour une mesure $m$. De manière intéressante, ces motifs permettent de retrouver la mesure $m$ de tous les autres motifs.

**Définition 7 (Représentation condensée adéquate)** *La relation de condensation adéquate à un ensemble de mesures $M$ est définie pour tout $t$ et $u$ : $t \subset^M u \Leftrightarrow (t \subset u) \wedge (\forall m \in M)\big(m(t) = m(u)\big)$. La représentation condensée des motifs fermés (resp. libres) adéquate à $M$ se traduit par la requête $L \subset_\neg^M L$ (resp. $L \supset_\neg^M L$).*

Par exemple, dans le jeu de données de la table 2.1, en considérant la mesure de fréquence, $B$ est un motif libre, mais il n'est pas fermé car $AB$ qui est un sur-ensemble a la même fréquence. Dans la littérature, les motifs fermés sont plus populaires que les motifs libres dont l'usage est souvent cantonné aux prémisses des règles d'association [40]. Dans [22], nous montrons que les motifs fréquents libres sont moins faciles à interpréter car une augmentation de la fréquence d'un motif mène dans certains cas à son extraction et dans d'autres cas à son rejet. De plus, les motifs fermés tendent à maximiser les mesures de corrélations. Par exemple, nous utilisons dans [84, 85] des motifs fermés qui maximisent l'hyper-lift (i.e., le lift entre toutes les partitions d'items possibles) et favorise la découverte de motifs pertinents.

**Représentations condensées adéquates aux fonctions conservées** Dans [87], nous avons montré comment évaluer efficacement les requêtes $L \subset_\neg^M L$ et $L \supset_\neg^M L$ pour une mesure conservée avec un élagage anti-monotone qui s'appuie sur la propriété 1 de manière locale.

**Définition 8 (Mesure conservée)** *Une mesure $m$ est conservée si lorsque l'ajout d'un item à un motif $X$ ne modifie pas sa mesure, alors l'addition de cet item ne modifie pas non plus la mesure pour les sur-ensembles de $X$ :*

$$(\forall X \subseteq Y)(\forall i)\big(m(X) = m(X \cup \{i\}) \Rightarrow m(Y) = m(Y \cup \{i\})\big)$$

Typiquement, la fréquence est une mesure conservée. Par exemple, dans le jeu de données de la table 2.1 (page 11), l'ajout de l'item $A$ à l'itemset $B$ conserve la fréquence de 3 (i.e., $\mathit{freq}(B, \mathcal{D}) = \mathit{freq}(AB, \mathcal{D}) = 3$) ce qui garantit que l'ajout de $A$ à un sur-ensemble de $B$ (disons $BD$) conserve aussi la même fréquence (dans ce cas, $\mathit{freq}(BD, \mathcal{D}) = \mathit{freq}(ABD, \mathcal{D}) = 1$).

De manière générale, pour l'extraction des meilleurs motifs au sens de $\preceq$, l'idée intuitive est de comparer localement les motifs $S \subseteq L$ avec leur voisinage, dénoté par $L_{/S}$, de sorte que $(S \preceq_\neg L) \subseteq L_{/S}$. Cela est possible lorsqu'on est certain qu'un motif ne peut être dominé que par un motif de son voisinage ce qui est le cas dans le calcul des représentations condensées où seuls un sur/sous-ensemble peut dominer un motif. Il est alors possible d'utiliser la propriété 1 avec $\mathit{Patt} = L_{/S}$ pour obtenir les motifs préférés de $S$ à savoir

$L_{/S} \prec_\neg L_{/S}$. Cela signifie que si un motif n'est pas dominé localement par les motifs $L_{/S}$, il ne le sera pas sur $L$. La réécriture suivante résume l'approche complète en considérant que $(S \preceq_\neg L) \subseteq L_{/S}$ :

$$L \prec_\neg L = (S \cup (L \setminus S)) \prec_\neg L \tag{3.1}$$
$$= \big(S \prec_\neg L\big) \cup \big((L \setminus S) \prec_\neg L\big) \tag{3.2}$$
$$= \underbrace{\big(L_{/S} \prec_\neg L_{/S}\big)}_{\text{meilleurs motifs sur } L_{/S}} \cup \big((L \setminus S) \prec_\neg L\big) \tag{3.3}$$

La ligne 1 sépare l'espace entre les motifs locaux $S$ et les autres à savoir $L \setminus S$. La ligne 2 distribue l'anti-couverture $\prec_\neg$ par rapport à l'union. Finalement, la propriété 1 est utilisée à la ligne 3 pour optimiser le calcul des meilleurs motifs pour $S$ (i.e., à gauche de l'union). Cette même réécriture peut être appliquée récursivement pour optimiser l'autre partie concernant $L \setminus S$.

Dans le cas des représentations condensées de motifs ensemblistes, $L_{/S}$ correspond aux sous-ensembles (resp. sur-ensembles) directs si l'on cherche les motifs minimaux (resp. maximaux) des classes d'équivalences. De plus, une contrainte anti-monotone évite l'exploration complète de l'espace suggérée par $(L \setminus S) \prec_\neg L$. Dans [87], notre algorithme opte pour une approche par niveau (i.e., dans la réécriture ci-dessus, $S$ correspond à un niveau).

**Système d'ensembles minimisable** Par la suite, nous avons généralisé cette approche pour l'extraction des motifs minimaux pour un système d'ensembles minimisable [92, 91]. Les systèmes d'ensembles [9] permettent de traiter des langages complexes comme les chaines de caractères, certains types de graphes, etc. Notre définition de système d'ensembles minimisable implique les notions de forme canonique (pour que plusieurs représentations d'un motif correspondent à un seul ensemble) et de l'extension $ext$ (qui vérifie $ext(X \cup Y) = ext(X) \cap ext(Y)$ pour tous ensembles $X$ et $Y$). Des langages et extensions variés sont traitables avec ce formalisme incluant les motifs essentiels [19], les règles de classifications [20], les représentations condensées fondées sur des mesures d'agrégats [87], etc. A notre connaissance, l'algorithme proposé (appelé DeFMe) est l'un des plus efficace pour extraire les motifs minimaux des classes d'équivalence (et même les traverses minimales) car l'évaluation de $L_{/S} \prec_\neg L_{/S}$ repose sur un mécanisme de comparaison des valeurs de $ext$ (appelé objets critiques) qui évite de consulter directement les motifs $L_{/S}$. La table 3.1 illustre l'efficacité de DeFMe par rapport aux deux principaux algorithmes de l'état de l'art : ACminer [15] et NDI [17]. Notre approche est à la fois plus rapide et consomme moins de mémoire grâce à un parcours en profondeur ce qui rend faisable l'extraction dans des jeux de données atypiques (par exemple, des jeux avec peu de transactions mais ayant des milliers d'items 90x27679).

Enfin, cette notion de système d'ensembles minimisable est généralisée aux motifs minimaux approchés dans [93]. Dans ce cas, les motifs préférés doivent être suffisamment distincts entre eux et découlent de la relation suivante : $t \subset^{ext,\delta} u \Leftrightarrow (t \subset u) \wedge \big(|ext(t) \setminus ext(u)| \leq \delta\big)$.

| | | | | temps (s) | | | mémoire (ko) | | |
|---|---|---|---|---|---|---|---|---|---|
| jeu de données | trans. | items | minsup | ACminer | NDI | DeFMe | ACminer | NDI | DeFMe |
| 74x822 | 74 | 822 | 88% | fail | fail | **45** | fail | fail | **3,328** |
| 90x27679 | 90 | 27,679 | 91% | fail | fail | **79** | fail | fail | **13,352** |
| chess | 3,196 | 75 | 22% | 6,623 | **187** | 192 | 3,914,588 | 1,684,540 | **8,744** |
| connect | 67,557 | 129 | 7% | 34,943 | **115** | 4,873 | 2,087,216 | 1,181,296 | **174,680** |
| pumsb | 49,046 | 2,113 | 51% | 70,014 | **212** | 548 | 7,236,812 | 1,818,500 | **118,240** |
| pumsb* | 49,046 | 2,088 | 5% | 21,267 | **202** | 4,600 | 5,175,752 | 2,523,384 | **170,632** |

TABLE 3.1 – Caractéristiques des benchmarks et rapidité de l'extraction des motifs libres avec DeFMe par rapport aux algorithmes ACminer [15] et NDI [17]

### 3.1.2 Motifs pareto-optimaux

Les représentations condensées ne sont pas si condensées que cela. D'ailleurs, les modèles de compression de données fondées sur le principe MDL (Minimum Description Length) utilisent comme table de codage un ensemble bien plus réduit de motifs. Pour cette raison, d'autres relations de préférences plus sélectives comme $<^m$ (discutée en introduction de cette section) ont été proposées. L'une des difficultés des top-$k$ motifs est de ranger les motifs suivant une seule mesure $m$. Dans de nombreux problèmes, il est nécessaire d'avoir au moins deux mesures comme la précision et le rappel par exemple. Dans ce cas, une combinaison de ces mesures (comme la F-mesure) tend à donner des valeurs identiques pour des motifs bien différents (e.g., forte précision et faible rappel ou l'inverse) et les motifs les mieux évalués manquent de diversité. Dans ce contexte, le principe des motifs Pareto-optimaux est de conserver tous les motifs qui sont au moins meilleurs pour une des mesures. Ce travail s'inspire des travaux en bases de données sur l'opérateur skyline [14]. Plusieurs propositions spécifiques ont exploité ce principe, mais la nôtre [90] [1] est la première à avoir formaliser le problème pour un large ensemble de mesures d'intérêt à savoir les mesures fondées sur des primitives. Cette classe de mesures combine n'importe quelles primitives (fonction monotone). Elle s'avère large et inclut de nombreuses mesures de la littérature comme par exemple la fréquence, l'aire, la moyenne, etc.

**Définition 9 (Relation Pareto)** *La relation Pareto $\prec^M$ d'un ensemble de mesures $M$ est définie pour tout $t$ et $u$ :*

$$t <^M u \Leftrightarrow (\forall m \in M)\big(m(t) \leq m(u)\big) \wedge (\exists m \in M)\big(m(t) < m(u)\big)$$

*$t \prec^M u$ signifie que $t$ est dominé par $u$ et $u$ est préféré à $t$.*

Les motifs préférés suivant cette relation (i.e., $L <^M L$) sont appelés motifs Pareto-optimaux (ou motifs skylines). Par exemple, la table 3.2 illustre l'utilisation d'une relation Pareto avec la fréquence et l'aire (qui correspond au produit de la fréquence par la longueur). Dans ce cas, seuls deux motifs sont Pareto-optimaux car $A$ et $AB$ sont incomparables pour $<^{\{freq,area\}}$. Ces deux motifs correspondent respectivement au motif le plus fréquent et au motif avec la plus grande aire mais il est possible d'avoir des motifs Pareto-optimaux qui ne maximisent aucune des deux mesures.

1. Cet article est disponible en annexe.

$F$

| patt | freq | area |
|---|---|---|
| A | 4 | 4 |
| B | 3 | 3 |
| C | 3 | 3 |
| D | 3 | 3 |
| AB | 3 | 6 |
| AC | 2 | 4 |
| AD | 2 | 4 |
| CD | 2 | 4 |

$F <^{\{freq,area\}} F$

| patt | freq | area |
|---|---|---|
| A | 4 | 4 |
| AB | 3 | 6 |

TABLE 3.2 – Exemples de motifs Pareto-optimaux

Au-delà de la définition du problème, l'intérêt de [90] est de proposer une méthode efficace de calcul de $L <^M L$ pour n'importe quel ensemble de mesures fondées sur des primitives (classe de mesures définie dans [88]). En effet, cette approche utilise les représentations condensées adéquates à un ensemble $M'$ (judicieusement choisi) comme espace de recherche car il existe des algorithmes très performants pour les extraire comme nous l'avons expliqué dans la section précédente. Pour revenir à l'exemple de la table 3.2, seul un motif fermé (selon la fréquence) peut prétendre à être un motif Pareto-optimal. En effet, le motif $B$ est forcément dominé par sa fermeture à savoir $AB$ (car la fréquence de $AB$ sera la même mais son aire sera plus grande que celle de $B$).

De manière générale, pour extraire les motifs préférés selon $\prec$, l'idée est d'utiliser une relation $\prec'$ qui est une relaxation de $\prec$ (i.e., pour tout $t \prec' u$, on a $t \prec u$ – mais pas forcément la réciproque). Si $\prec'$ est une relaxation de $\prec$ et $R$ un ensemble de motifs, alors l'ensemble de motifs $R \prec' R$ est un sur-ensemble des motifs préférés : $(R \prec_\neg R) \subseteq (R \prec'_\neg R)$. Avec $Patt = (R \preceq'_\neg R)$, il est alors possible d'utiliser la propriété 1. En d'autres termes, la relation $\prec'$ est utilisée comme premier filtre. Si un motif est rejeté par $\prec'$, il l'aurait été par $\prec$. En revanche, les motifs non-rejetés devront être comparés entre eux avec la relation $\prec$. Cette approche avec deux filtres s'avère avantageuse si le coût de l'évaluation de $R \prec'_\neg R$ est faible comparativement à la réduction opérée. Dans notre cas, nous dérivons automatiquement un ensemble de mesures $M'$ à partir de $M$ de sorte que la relation de condensation adéquate à $M'$ (voir la définition 7) soit une relaxation de $\prec^M$ – $M$ est dit $M'$-skylineable. Comme nous l'avons vu dans la section précédente, nous pouvons évaluer efficacement la requête $L \subset^{M'}_\neg L$ ce qui conduit à un calcul efficace de $L \prec^M_\neg L$. Pour la table 3.2 où $M = \{freq, area\}$, on obtient $M' = \{freq\}$.

Dans le cadre de la programmation par contraintes, les motifs déjà explorés sont utilisés pour générer de nouvelles contraintes qui réduiront l'espace de recherche. Cette altération du problème en cours de résolution, appelée Dynamic CSP, améliore encore la réduction de l'espace de recherche [96].

### 3.1.3 De la satisfaction à l'optimisation

Du point de vue formel, les travaux présentés dans cette section montrent bien l'importance de l'opérateur de couverture pour trier les motifs en les comparant deux à deux. Sans

fixer de seuils, la relation de Pareto conduit à extraire un ensemble très réduit de motifs Pareto-optimaux. La programmation par domination [72] s'appuie sur un opérateur comparable dans le cadre de la programmation par contraintes pour reformuler de nombreux problèmes d'extraction de motifs.

Même si utiliser une approche par optimisation plutôt que par satisfaction évite le problème du choix des seuils, il est complexe pour un utilisateur d'expliciter la relation de préférences correspondant à ses attentes. Par exemple, avec la relation de Pareto, il n'est pas si simple de choisir l'ensemble pertinent de mesures d'intérêt comme nous l'avons constaté lors de nos travaux menés avec des chimistes [90]. De plus, tous les motifs extraits sont préférés mais ils ne constituent pas ensemble un modèle holistique contrairement aux travaux sur les ensembles de motifs. En effet, comparer les motifs deux à deux est insuffisant pour construire des ensembles de motifs complémentaires. Nous revenons dès la section suivante sur cette limite.

## 3.2 Construction itérative de modèles

Parallèlement, à l'usage d'une relation de préférences pour sélectionner les motifs, la construction de modèles cherche à trouver une collection réduite de motifs qui soit lisible pour l'utilisateur. En effet, il est possible de construire un ensemble raisonnable de motifs en évitant les redondances entre motifs non-comparables. Typiquement, les motifs Pareto-optimaux reposent uniquement sur des mesures d'intérêt pour retenir les motifs préférés. Du coup, ils peuvent s'avérer être une très mauvaise représentation du jeu de données i.e., certaines parties du jeu de données peuvent ne pas être couvertes (e.g., la transaction $t_4$ du jeu de données de la table 2.1 n'est couverte ni par $A$, ni par $AB$ de la table 3.2). De manière générale, pour garantir une complémentarité entre tous les motifs d'un ensemble, les comparaisons deux à deux sont insuffisantes.

Les modèles fondés sur des motifs locaux sont souvent mis en avant pour leur capacité à bien décrire (même si son usage peut aussi être prédictif dans certains cas). Cette capacité à décrire est double : description des données (dont il sont extraits) et description de la collection complète des motifs (dont ils sont jugés les meilleurs représentants). De manière formelle, étant donnée une collection de motifs $Patt \subseteq L$, un modèle $M$ est l'un des meilleurs ensembles de motifs maximisant un critère $\Phi$ (par exemple, le nombre de transactions couvertes) :

$$\arg \max_{M \subseteq Patt} \Phi(M)$$

Comme dans la section précédente, la découverte de motifs est donc vue comme un problème d'optimisation mais cette fois, l'objectif est de trouver un ensemble de motifs qui doivent être complémentaires. A noter que *Patt* est souvent une collection de motifs extraite exhaustivement (par exemple, le résultat d'une requête PORA). Il est aussi possible d'ajouter une contrainte de taille $k$ en imposant que $\Phi(M) = 0$ si $|M| \neq k$.

Plusieurs travaux se sont intéressés à la résolution exacte de ce problème pour des cas particuliers de $\Phi$ (par exemple, [62, 55] dans le cadre de la programmation par contraintes et [81] en bénéficiant de l'anti-monotonie de l'ensemble de motifs). Pour parvenir à traiter des fonctions $\Phi$ sans bonnes propriétés et des jeux de données de grande taille, plusieurs cadres se sont intéressés à la construction itérative de modèles. Intuitivement, cela consiste à extraire un large réservoir de motifs locaux potentiellement pertinents dans une première étape. Ensuite, cette collection est passée en revue par ordre de pertinence pour construire un modèle. Plus précisément, chaque motif non représenté par le modèle en cours de construction est ajouté à ce modèle. Cette approche originellement présentée pour construire un classifieur CBA [65] a donné lieu à plusieurs cadres généraux [63, 16].

Dans le cadre des travaux de thèse de Eynollah Khanjari Miyaneh [2] [61], nous avons également proposé une méthode générique pour la construction itérative de modèles [48, 49]. Cette dernière modélise également les méthodes heuristiques en autorisant à chaque itération d'extraire exhaustivement une nouvelle collection de motifs. La sous-section 3.2.1 reprend cette contribution en présentant un algorithme simplifié s'appuyant sur PORA. Ensuite, la sous-section 3.2.2 illustre cet algorithme générique avec la construction itérative

2. Thèse dirigée par Arnaud Giacometti et co-encadrée avec Patrick Marcel

d'un profil de préférences. Ce travail a été réalisé dans le cadre des travaux de thèse de Mouhamadou Saliou Diallo [3] [27] en coopération avec l'université d'Uberlândia (Brésil) au sein du projet Stic-Amsud PQuery.

### 3.2.1 Algorithme TwoSteps

Cette section présente un algorithme générique pour résoudre le problème d'optimisation indiquée ci-dessus. L'idée de cet algorithme est de classer les motifs suivant un intérêt modélisé par un ordre total $<^q$. Pour être ajouté au modèle, le meilleur motif courant doit couvrir une partie du jeu de données pas encore couverte par la partie du modèle déjà construite (au sens d'une relation de couverture $\lhd$). L'algorithme TwoSteps (voir algorithme 1) prend en entrée un jeu de données $D$, un ensemble de motifs $L$, un ordre total $<^q$ et une relation de couverture $\lhd$. Après l'initialisation des différentes variables (lignes 1 à 3), la boucle principale construit le modèle en ajoutant un à un les motifs tant que le réservoir de candidats est non vide. La ligne 5 sélectionne le meilleur motif selon $<^q$ et la ligne 6 l'ajoute au modèle. Les transactions couvertes sont supprimées à la ligne 7 et les motifs qui ne couvrent plus de nouvelles données sont retirés du réservoir. A la fin, lorsqu'il n'y a plus de motifs dans $L_k$, le modèle est retourné (ligne 11).

**Algorithm 1** Pattern-based modeling algorithm (TwoSteps)

**Input:** A dataset $D$, a set of patterns $L$, a total quality order $<^q$ and a cover relation $\lhd$
**Output:** A pattern-based model $M$

1: $M := \emptyset$
2: $D_0 := D$ and $L_0 := L \lhd_\ltimes D$
3: $k := 0$
4: **while** $L_k \neq \emptyset$ **do**
5: $\quad Top := (L_k <^q_\neg L_k)$
6: $\quad M := M \cup Top$
7: $\quad D_{k+1} := D_k \rhd_\neg Top$
8: $\quad L_{k+1} := L_k \lhd_\ltimes D_{k+1}$
9: $\quad k := k + 1$
10: **od**
11: **return** $M$

Plutôt que de raffiner $L_k$, la version la plus générale dans [49] propose de calculer une nouvelle collection à chaque itération (ce qui en pratique n'est pas envisageable avec l'extraction de motifs exhaustive car trop onéreux). De même, l'ordre $<^q$ peut être modifié à chaque itération.

Nous avons été forcés de décrire la construction itérative de modèle sous une forme algorithmique car PORA n'est pas suffisamment expressive pour modéliser un processus itératif. Pour deux relations $<^q$ et $\lhd$, il serait bien possible de construire le modèle correspondant avec une comparaison des motifs par paires comme dans la section précédente

3. Thèse dirigée par Arnaud Giacometti et Cheikh Talibouya Diop, et co-encadrée avec Dominique Li

(i.e., $L \prec^* L$) en considérant la relation $\prec^*$ définie ainsi :

$$t \succ^* u \Leftrightarrow (t >^q u) \wedge (\exists t_1 >^q t, \ldots, t_k >^q t) : u \not\triangleleft_{\ltimes} (D \triangleright_{\neg} \{t_1, \ldots, t_k, t\})$$

Malheureusement, il est clair que cette relation ne peut être exprimée avec l'algèbre relationnelle contrairement aux relations présentées dans la section précédente car elle dissimule en fait un ensemble de motifs. Pour exprimer algébriquement la construction de modèles, il serait donc nécessaire d'ajouter la récursion ou un opérateur de point fixe [18].

**Cas particulier de la construction de résumés de motifs** Dans le cadre de la thèse de Marie N'Diaye [4] [68], nous avons utilisé cet algorithme dans le cas particulier où le jeu de données $D$ est un ensemble de motifs d'un langage $\mathcal{L}_P$ que l'on souhaite résumer par un ensemble de motifs issu d'un autre langage $\mathcal{L}_S$. Par exemple, nous nous sommes intéressés à résumer un ensemble de règles d'associations de couples attributs/valeurs par un sous-ensembles de règles respectant un schéma donné [70, 71, 69]. De manière générale, un résumé se définit formellement de la manière suivante :

**Définition 10 (Résumé de motifs)** *Un ensemble de motifs $L_S \subseteq \mathcal{L}_S$ est un résumé d'un ensemble de motifs $L_P \subseteq \mathcal{L}_P$ pour la relation de couverture $\triangleleft^{SP} \subseteq \mathcal{L}_S \times \mathcal{L}_P$ ssi les 3 propriétés suivantes sont vérifiées :*

1. *Tous les motifs de $L_P$ sont couverts par au moins un motif de $L_S$ : $L_P \triangleright_{\ltimes}^{SP} L_S = L_P$*
2. *Tous les motifs de $L_S$ couvrent au moins un motif de $L_P$ :$L_S \triangleleft_{\ltimes}^{SP} L_P = L_S$*
3. *La taille de $L_S$ est plus petite que celle de $L_P$ : $|L_S| \leq |L_P|$*

Pour construire un résumé $L_S$ pour $L_P$, il suffit d'utiliser l'algorithme en deux phases de la manière suivante : $L_S := \textsc{TwoSteps}(L_P, \mathcal{L}_S, <^q, \triangleleft^{SP})$ où $<^q$ est un critère de qualité. Dans nos travaux sur les résumés de règles d'associations, le critère de qualité favorisait le choix de règles qui couvraient des règles similaires en utilisant l'entropie conditionnelle. Par ailleurs, le langage $\mathcal{L}_S$ choisit pour résumer les règles $L_P$ pouvait être modifié par l'utilisateur en sélectionnant les attributs qu'il jugeait comme pertinent.

### 3.2.2 Construction d'un profil de préférences

Cette section illustre la construction itérative d'un profil de préférences en s'appuyant sur l'algorithme TwoSteps présenté dans la précédente section. L'objectif est de modéliser les préférences de l'utilisateur (exprimées dans $P$, voir la table 2.3) sous la forme d'un ensemble de règles de préférences contextuelles comme présenté dans la section 2.2. In fine, l'idée est de déterminer entre deux tuples $t$ et $u$ lequel est préféré par l'utilisateur.

4. Thèse dirigée par Arnaud Giacometti et Cheikh Talibouya Diop, et co-encadrée avec Patrick Marcel

**Construction du modèle** Bien sûr, il est possible d'extraire toutes les règles de préférences ayant un support et une confiance suffisamment élevés. Mais, comme pour les règles d'associations traditionnelles, on obtient alors un ensemble de taille gigantesque contenant de nombreuses redondances et qui s'avère donc inintelligible. Il est possible de le réduire sans perte d'information avec la relation de condensation adéquate à $\{pos, neg\}$ (voir la section 3.1.1). Plus précisément, nous énumérons tous les motifs minimaux $P_{pref} \supset_{\neg}^{\{pos,neg\}} P_{pref}$ qui restent nombreux. Ensuite, l'algorithme TwoSteps est exécuté sur le jeu de données $D \bowtie P \bowtie D'$ en considérant toutes les règles minimales de préférences contextuelles, l'ordre $<_{pref}^{q}$ (défini ci-dessous) et la couverture positive $\lhd^{+}$.

$$\begin{aligned} r <_{pref}^{q} s &\Leftrightarrow conf(r) < conf(s) \\ &\Leftrightarrow conf(r) = conf(s) \wedge supp(r) < supp(s) \\ &\Leftrightarrow conf(r) = conf(s) \wedge supp(r) = supp(s) \wedge r <_{total} s \end{aligned}$$

Cet ordre privilégie les règles ayant une confiance élevée ; puis, un support élevé ; et finalement, un ordre total arbitraire (par exemple, un ordre lexicographique). Cet ordre comparable à celui utilisé dans CBA [65] s'avère efficace en pratique pour construire un modèle intelligible.

**Recommandation** Utiliser un modèle de préférences utilisateur pour recommander le meilleur tuple entre $u_1$ et $u_2$ s'avère plus compliqué que pour les règles d'associations. En effet, utiliser la meilleure règle au sens de $<_{pref}^{q}$ comme dans CBA conduit à un rappel catastrophique [26] (cf. la figure 3.1 ci-après). Par conséquent dans [25], nous avons proposé une méthode pour améliorer la prédiction en ne comparant pas directement les deux tuples. L'idée est que chaque tuple $t$ du jeu de données initial $D$ donne un score à $u_1$ et $u_2$ en utilisant la meilleure règle du profil : 1 point si $u$ est préféré à $t$, 0.5 si pas d'avis et -1 point si $t$ est préféré à $u$. Le tuple avec le meilleur score est celui qui est recommandé. L'avantage de cette approche, appelée vote par valeur, est de construire un ordre faible sur tous les tuples.

Nous avons appliqué notre méthode de construction de profil de préférences sur une base de données cinématographiques recoupant les données sur des films issues de IMDB [5] et les préférences des utilisateurs issues de MovieLens [6]. Ensuite, nous avons comparé la qualité de la classification en utilisant la prédiction par la meilleure règle (BR pour *Best Rule*) ou en utilisant la prédiction par vote par valeurs (BR&RV pour *Best Rule and Range Voting*). Par ailleurs, une classification par machine à vecteurs de support (SVM) est utilisée comme approche de base. La figure 3.1 trace le rappel et la précision en fonction du nombre minimum de paires de préférence que doit couvrir une règle pour être sélectionnée, noté $l$. Cela signifie que la taille du modèle diminue avec $l$ : environ 200 règles pour $l = 0$ et moins de 10 pour $l = 3500$. On constate que le rappel de la méthode BR s'effondre lorsque la taille du modèle diminue (avec un léger gain de précision). A l'inverse, la méthode BR&RV a un rappel qui reste relativement stable et compétitif avec

5. www.imdb.com
6. movielens.org

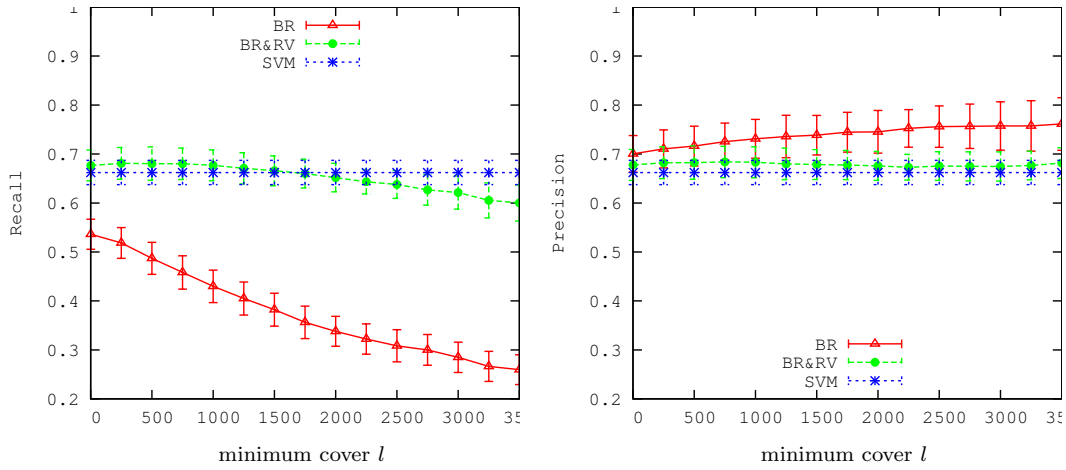



FIGURE 3.1 – Qualité d'une prédiction basée sur un profile de règles de préférences contextuelles en fonction du nombre minimum $l$ de données couvertes (BR : prédiction avec la meilleure règle ; BR&RV : prédiction avec le vote par valeur ; SVM : classification avec une machine à vecteurs de support)

un classifieur SVM. Par contre, notre modèle explicite les préférences contrairement au SVM. Ce compromis entre précision et intelligibilité est important pour proposer des systèmes centrés sur l'utilisateur.

### 3.2.3 Du motif à l'ensemble de motifs

Cette section souligne à nouveau l'importance de la relation de couverture pour découvrir les motifs pertinents. En effet, la construction de modèles peut se formaliser à nouveau par une relation de couverture. Cette dernière dissimule néanmoins un quantificateur et ne peut s'exprimer en utilisant PORA sans ajouter la récursivité ou un opérateur de point fixe. Ce basculement dans l'expressivité explicite une démarcation claire entre motifs locaux et modèles, que nous jugeons importante.

Nous avons aussi identifié plusieurs limites de la construction itérative de modèles indépendantes de l'algébrisation de notre travail :

Premièrement, la communauté de la découverte de motifs a assidument exploré la construction itérative de modèle pour rendre intelligible les énormes collections de motifs inexploitables de prime abord. Pourtant, comme dans la section précédente, la déclaration de ce qui est intéressant aux yeux de l'utilisateur est très difficile à formuler (même choisir une approche parmi la variété des méthodes existantes est un challenge en soi).

Deuxièmement, réduire le nombre de motifs pour l'intelligibilité conduit dans certaines tâches prédictives à réduire aussi le rappel. Dans notre exemple avec les profils de préférences, le vote par valeur a permis de maintenir un rappel raisonnable mais une telle solution n'est pas toujours possible. Pour cette raison, à contrepied de la construction de modèles, certaines approches fondées sur les motifs ont cherché plutôt à conserver

tous les motifs pour construire des classifieurs avec un rappel satisfaisant. Ces modèles massifs se confrontent au difficile problème de l'orchestration des motifs locaux. Dans le contexte du traitement naturel des langues, une large partie des travaux de thèse de Damien Nouvel[7] [73] s'inscrivent dans cette direction pour la reconnaissance des entités nommées [78, 76, 77, 75, 74]. De manière intéressante, il a proposé de conserver tous les motifs (pour maximiser le rappel) et d'utiliser MaxEnt pour pondérer l'importance des motifs dans la prédiction (pour maximiser la précision).

Enfin, l'algorithme TwoSteps s'appuie sur une extraction exhaustive préalable qui construit un énorme réservoir de motifs. La promesse initiale est de disposer de tous les motifs pertinents et donc de construire le meilleur modèle. Pourtant, il est tout de même fréquent de manquer de bons motifs que les approches heuristiques seraient parfois parvenus à extraire. Bref, séparer l'extraction de la construction de modèle semble un choix peu satisfaisant lorsque l'on recherche un ensemble de motifs complémentaires. Il serait judicieux de bénéficier du modèle en construction pour mieux choisir les motifs à y ajouter (comme le faisaient les méthodes heuristiques [28]). Toujours pour avoir le plus grand réservoir possible, il est nécessaire de fixer les seuils d'extraction au plus bas. Cela induit malheureusement un temps d'extraction très coûteux (c'est pour cela qu'il n'est pas répété à chaque étape de la construction du modèle). Pire, il s'agit d'un temps incompressible qui empêche l'utilisateur d'obtenir dans un temps raisonnable une réponse facilitant un processus interactif. Nous reviendrons sur ces aspects dans la section 4.2.

7. Thèse dirigée par Jean-Yves Antoine et co-encadrée avec Nathalie Friburger

# Chapitre 4

# Découverte de motifs guidée par l'analyse

Ce chapitre concerne nos travaux où l'analyse des motifs découverts est le support à leur propre extraction. Quelque soit la technique d'extraction de motifs, une collection de motifs (souvent conséquente) est retournée à l'utilisateur. Pour analyser cette collection, ce dernier n'a pas d'autre choix que de trier les motifs du meilleur au moins bon suivant une mesure d'intérêt. Son intérêt se focalise alors sur un motif particulier, plutôt en haut du classement. Il étudie alors ce motif puis passe à un autre jusqu'à ce qu'il soit satisfait. Partant de ces observations, en 2011, nous avons modélisé un tel processus d'analyse en choisissant le motif à étudier au hasard proportionnellement à la mesure d'intérêt [45, 42]. Dans ce cas, tous les motifs sont analysés par l'utilisateur avec une probabilité proportionnelle à leur intérêt dans le jeu de données $\mathcal{D}$ : $P(X) \sim m(X, \mathcal{D})$ (propriété $\mathbf{P}_{\text{pro}}$). Plutôt que de tirer des motifs depuis la collection de motifs extraits, il s'avère bien plus judicieux de les tirer depuis le jeu de données en garantissant la même propriété. Ainsi, en répétant le tirage proportionnellement à $m$, il est même possible d'approximer la mesure d'intérêt pour tous les motifs ce qui s'avère être une propriété très forte. Comme pour la propriété isolée pour la satisfaction de contraintes en introduction $\mathbf{P}_{\text{sat}}$ ou celle pour l'optimisation $\mathbf{P}_{\text{opt}}$ du chapitre précédent, l'extraction garantit une propriété *exacte* et *globale* sur le langage.

Nous commençons ce chapitre par étendre PORA à l'échantillonnage de motifs en sortie grâce à l'opérateur d'échantillonnage. Ce nouvel opérateur est un outil remarquable pour obtenir instantanément des motifs diversifiés et ainsi, remédier au manque d'interactivité des méthodes d'extraction de motifs exhaustives. Nous décrirons plusieurs implémentations de cet opérateur pour des données complexes par nature (i.e., données numériques et séquentielles) ou par contrainte de stockage (i.e., données distribuées). La section 4.2 illustre directement l'utilisation de l'opérateur d'échantillonnage pour rendre anytime la construction de modèles fondés sur les motifs. Nous montrerons comment retourner à chaque instant un modèle qui tend vers le modèle issu de l'algorithme TwoSteps. Nous revisiterons aussi le calcul du score d'aberration FPOF à la lumière de l'échantillonnage de motifs. Parfois, il est très difficile pour un utilisateur de spécifier la mesure d'intérêt

qui décrit au mieux ses préférences. En admettant que cette mesure existe, la section 4.3 montre finalement comment apprendre cette mesure tout en échantillonnant des motifs proportionnellement à cette mesure. Nous illustrerons ce cycle vertueux dans le cas où l'utilisateur est intéressé par la caractérisation d'une classe positive non-étiquetée.

## 4.1 Echantillonnage de motifs

En introduction de ce chapitre, il a été expliqué que présenter à l'utilisateur des motifs tirés aléatoirement proportionnellement à une mesure d'intérêt $m$ revient finalement à analyser des motifs triés suivant $m$. Précisons que nos travaux concernent l'échantillonnage de motifs en sortie et non, l'échantillonnage de motifs en entrée. L'échantillonnage en entrée [95] consiste à régénérer depuis un échantillon de données tous les motifs qui auraient été extraits depuis le jeu de données complet. L'échantillonnage en sortie [5] consiste à générer un échantillon de motifs parmi les motifs qui auraient été extraits depuis le jeu de données complet. Par exemple, la table 4.1 présente un échantillon de 20 motifs tirés selon la fréquence à partir du jeu de données présenté dans la table 2.1. On constate par exemple que le motif $A$ qui a une fréquence de 4 est deux fois plus présent que le motif $AC$ qui a une fréquence de 2. Plusieurs procédures ont été proposées pour l'échantillonnage de motifs. La première famille [5] repose sur les méthodes de Monte-Carlo par chaînes de Markov. L'idée est que la loi stationnaire de la marche aléatoire corresponde à la distribution à échantillonner. La limite de telles approches stochastiques est la vitesse de convergence qui peut être lente. La seconde famille [13] consiste à tirer une instance du jeu de données, puis à tirer un motif contenu dans cette instance. En choisissant judicieusement les deux distributions de tirage, il est alors possible d'obtenir un tirage exact selon la distribution désirée. Du fait de son exactitude et de sa rapidité, nos travaux se sont focalisés sur cette seconde famille.

$D$

| id | trans |
|---|---|
| $t_1$ | ACD |
| $t_2$ | ABD |
| $t_3$ | ABCE |
| $t_4$ | CD |
| $t_5$ | AB |

Echantillonnage en sortie $\longrightarrow$

| patt |
|---|
| AC |
| CD |
| A |
| ABD |
| C |
| AB |
| AD |
| AE |
| BD |
| ABC |

| |
|---|
| D |
| C |
| AB |
| ACD |
| A |
| ABCE |
| BCE |
| D |
| CE |
| B |

TABLE 4.1 – Exemple d'un échantillon de 20 motifs tirés suivant la fréquence

La première sous-section s'attaque au niveau logique de l'échantillonnage de motifs en s'appuyant sur notre algèbre PORA augmentée d'un nouvel opérateur d'échantillonnage issu des bases de données. Cette sous-section revisite notamment la procédure aléatoire en deux étapes de manière algébrique. La seconde sous-section s'intéresse ensuite au niveau physique en considérant l'implémentation efficace de la procédure aléatoire en deux étapes dans plusieurs contextes.

Les contributions de cette section ont été réalisées dans le cadre de plusieurs projets nationaux (Peps Prefute, Mastodons Decade). Une grande partie de ces contributions résulte des travaux de la thèse de Lamine Diop [1].

1. Doctorant de l'université Gaston Berger de Saint-Louis dirigé par Arnaud Giacometti et Cheikh T.

### 4.1.1 Opérateur d'échantillonnage

Cette première sous-section établit le lien entre les bases de données et l'échantillonnage de motifs. Pour cela, nous allons ajouter à PORA l'opérateur d'échantillonnage introduit dans [79]. Ainsi, nous pourrons à la fois modéliser l'échantillonnage de motifs sous la forme de requêtes, mais aussi réécrire ces requêtes pour les optimiser.

**Définition et expressivité** Nous nous appuyons sur une version de l'opérateur d'échantillonnage de tuples introduit dans [79] :

**Définition 11 (Opération d'échantillonnage [79])** *L'échantillonnage de $k$ tuples d'une relation $R$ suivant l'expression numérique expr est $\psi^k_{expr}(R)$, i.e. pour toute instance $I$ de $R$, $\psi^k_{expr}(I)$ retourne $k$ tuples de $I$ en les tirant proportionnellement par rapport à l'expression expr et avec remise.*

Dans la suite, nous considérons aussi deux notations particulières de $\psi^k_{expr}(R)$ : si $k$ est omis, alors on ne tire qu'un seul tuple (i.e., $\psi^1_{expr}(R)$) et si $expr$ est omis, alors on tire les $k$ tuples avec une distribution uniforme (i.e., $\psi^K_1(R)$). Notons que toute expression constante différente de 0 (même valeur pour tous les tuples) conduit à un échantillonnage uniforme.

Grâce à l'opération de la définition 11, il est aisé de formuler l'échantillonnage de motifs fréquents avec la requête suivante :

$$\psi^k_{freq}(\gamma_{patt,\texttt{COUNT}(trans)\rightarrow freq}(\delta(L) \triangleleft D))$$

Cet échantillonnage de motifs dit en sortie est très différent de l'échantillonnage de motifs en entrée [95] qui se formule ainsi : $\gamma_{patt,\texttt{COUNT}(trans)\rightarrow freq}(\delta(L) \triangleleft \psi^k(D))$. L'objectif de l'échantillonnage en entrée consiste à accélérer l'extraction exhaustive des motifs fréquents en utilisant seulement $k$ transactions.

Bien sûr, l'opération d'échantillonnage est non-déterministe. Dans ce contexte, la relation d'égalité entre deux requêtes est une relation de comparaison trop forte et la notion d'équivalence lui est préférée comme dans [79]. Deux requêtes $q$ et $q'$ sont équivalentes, noté $q(R) \Leftrightarrow q'(R)$ ssi pour toute instance $I$ de $R$, la probabilité qu'un tuple $t$ apparaisse dans $q(R)$ et dans $q'(R)$ est la même. Cette relation d'équivalence est notamment utile pour les règles de réécriture. Par exemple, la règle indiquant qu'il est possible de subdiviser en deux une requête $\psi^k_{expr}(R)$ s'écrit de la manière suivante (pour tout $l \in [1..k-1]$) : $\psi^k_{expr}(R) \Leftrightarrow \psi^{k-l}_{expr}(R) \cup \psi^l_{expr}(R)$. En particulier, la requête $\psi^k_{expr}(R)$ est équivalente à l'union de $k$ requêtes $\psi_{expr}(R)$. Cette règle de réécriture est très intéressante car elle signifie que l'évaluation de la requête $\psi^k_{expr}(R)$ peut facilement être distribuée.

Il n'est pas possible d'écrire des requêtes non-déterministes avec l'algèbre relationnelle traditionnelle et l'ajout de l'opérateur d'échantillonnage augmente donc son expressivité. De même, PORA augmentée de cet opérateur est strictement plus expressif que PORA. Par ailleurs, la complexité en évaluation [22] (brièvement présentée à la page 19) de la

Diop, et co-encadré avec Dominique Li

requête d'échantillonnage est $O(2^n)$ où $n$ est le nombre d'items du langage. En effet, la variation de la fréquence de n'importe quel motif du treillis a un impact sur la probabilité de sélection d'un motif.

**Procédure aléatoire en deux étapes** Cette procédure exacte introduite [13] consiste 1) à tirer une transaction proportionnellement au nombre de motifs qu'elle contient et ensuite, 2) à tirer uniformément un motif au sein de cette transaction. De manière intéressante, cette procédure est très efficace car le nombre d'itemsets contenus dans une transaction $t$ est tout simplement $2^{|t|}$ et le tirage uniforme consiste à retenir chaque item de la transaction $t$ en tirant une pièce. Nous allons maintenant formuler algébriquement cette procédure aléatoire en deux étapes en réécrivant la requête d'échantillonnage suivant la fréquence :

$$\begin{aligned}
&\pi_{patt}(\psi_{freq}(\gamma_{patt,\texttt{COUNT}(trans)\rightarrow freq}(\delta(L) \triangleleft D))) \\
&\quad \Leftrightarrow \pi_{patt}(\psi(\delta(L) \triangleleft D)) && (4.1)\\
&\quad \Leftrightarrow \pi_{patt}(\underbrace{\psi(\delta(L) \triangleleft \underbrace{\psi_{nb}(\gamma_{trans,\texttt{COUNT}(patt)\rightarrow nb}(\delta(L) \triangleleft D)))}_{\text{1) tirage d'une transaction selon } nb}}_{\text{2) tirage uniforme d'un motif}}) && (4.2)
\end{aligned}$$

La première ligne retire le calcul de la fréquence qui se neutralise avec l'échantillonnage proportionnel à la fréquence. Cela revient à échantillonner uniformément la relation $\delta(L) \triangleleft D$. La deuxième équivalence semble complexifier inutilement les choses. Au lieu de partir de l'ensemble du jeu de données $D$, nous choisissons aléatoirement une transaction de $D$ proportionnellement au nombre de motifs qu'elle contient. En réalité, cette réécriture est intéressante car il n'est pas envisageable de matérialiser $\delta(L) \triangleleft D$ pour tirer un tuple. A l'inverse, $\gamma_{trans,\texttt{COUNT}(patt)\rightarrow nb}(\delta(L) \triangleleft D)$ est souvent évaluable sans avoir à matérialiser $\delta(L) \triangleleft D$ (voir la sous-section suivante).

Au final, nous retrouvons bien les deux tirages de la procédure aléatoire en deux étapes au sein de la seconde équivalence. Premièrement, le tirage d'une transaction $t$ proportionnellement à son nombre de motifs correspond à $\psi(\delta(L) \triangleleft \psi_{nb}(\gamma_{trans,\texttt{COUNT}(patt)\rightarrow nb}(\delta(L) \triangleleft D)))$. Deuxièmement, le tirage uniforme d'un motif au sein de $t$ correspond à $\psi(\delta(L) \triangleleft t)$. Tout comme l'approche originelle, la force de cette réécriture est de parvenir à échantillonner un motif par rapport à la fréquence sans parcourir intégralement le jeu de données. En contrepartie, sa principale limite est ne pas expliciter la fréquence du motif échantillonné (i.e., on ne peut pas retirer la projection $\pi_{patt}(\cdot)$ ou la compléter avec la fréquence).

### 4.1.2 Implémentation de l'échantillonnage dans des données complexes

Dans cette sous-section, nous décrivons l'implémentation des deux étapes isolées ci-avant dans des données avec une nature difficile (comme les langages de données numériques ou séquentielles) ou avec une contrainte particulière à savoir la distribution sur plusieurs noeuds.

**Données numériques [54]** Un jeu de données numériques consiste à considérer une relation $D[A_1, \ldots, A_d]$ avec $d$ attributs à valeurs numériques dans $\mathbb{R}$ (i.e., chaque tuple est un point dans l'espace $\mathbb{R}^d$). Dans ce cas, le langage de motifs $L[A'_1, \ldots, A'_d]$ correspond à tous les points de tous les sous-espaces possibles (i.e., chaque tuple est un point de $(\mathbb{R} \cup \{null\})^d$). La valeur *null* permet d'ignorer une dimension. Pour un rayon $r$, notre problème est de tirer un tuple de $\delta(L)$ proportionnellement à sa densité i.e., le nombre de points de $D$ à une distance inférieure à $r$ normalisé par le volume de la boule de rayon $r$. Ce problème soulève un vrai challenge car le langage des motifs numériques est infini. Pour lever ce verrou, nous avons décomposé la seconde étape $\psi(\delta(L) \triangleleft t)$ en deux sous étapes (tandis que la première correspond à un simple tirage uniforme sur $D$) :

— Un tirage d'un motif numérique dans le domaine actif de la relation $D$

— et ensuite, un tirage uniforme dans la boule de rayon $r$.

Nous avons démontré l'exactitude de notre approche. De manière intéressante, cette approche pourrait être généralisée à n'importe quel espace tolérant $(\mathcal{L}, \sim)$ (i.e., la relation $\sim$ est réflexive et symétrique, mais pas transitive). Dans ce cas, le dernier tirage dans la boule de rayon $r$ serait remplacé par un tirage d'un motif par rapport à la relation de tolérance.

**Données séquentielles [30, 29]** Nous avons étendu l'échantillonnage de motifs aux données séquentielles (cf. page 13 pour un rappel des notations) [2]. Cependant, un échantillonnage de motifs naïf en fonction de la fréquence ne serait pas pertinent pour les données séquentielles en raison de l'écueil de la longue traîne. En statistique et en économie, une distribution a une longue traîne si elle comporte un grand nombre d'occurrences éloignées de la partie centrale de la distribution [8]. Dans notre contexte, la longue traîne désigne les motifs séquentiels longs et rares, beaucoup plus nombreux que les motifs courts et fréquents (la « tête »). En conséquence, il est presque impossible de tirer des motifs parmi cette tête, i.e. les plus généraux en dépit du biais de la fréquence. Ce problème est plus grave avec les données séquentielles qu'avec les données transactionnelles, car le nombre de sous-motifs dans une séquence est beaucoup plus élevé que celui d'un itemset de même longueur.

Pour éviter la longue traîne, nous avons ajouté une contrainte sur la norme maximale d'une sous-séquence à tirer (où la norme, dénotée $||\cdot||$, est la somme des cardinalités des itemsets de la séquence). Voici la formalisation sous la forme d'une requête PORA où $M$ est une norme maximale :

$$\psi^k_{freq}(\gamma_{patt,\texttt{COUNT}(trans)\rightarrow freq}(\sigma_{||patt||\leq M}(\delta(L)) \sqsubseteq D))$$

Par rapport aux itemsets, le verrou scientifique est de ne pas biaiser le tirage des sous-séquences qui ont plusieurs occurrences au sein d'une même sous-séquence (e.g., $\langle ac \rangle$ se répète deux fois dans la séquence $\langle (ab)c(ac) \rangle$ : $\langle (\mathbf{a}b)\mathbf{c}(ac) \rangle$ et $\langle (\mathbf{a}b)c(a\mathbf{c}) \rangle$). Par ailleurs, la contrainte $\sigma_{||patt||\leq M}(\cdot)$ complexifie aussi le tirage de la séquence dans le jeu de données et le tirage uniforme de la sous-séquence. Après réécriture, pour

2. L'article [29] est disponible en annexe.

évaluer $\psi_{nb}(\gamma_{trans,\texttt{COUNT}(patt)\rightarrow nb}(\sigma_{||patt||\leq M}(\delta(L)) \sqsubseteq D))$, nous sommes parvenus à compter le nombre de sous-séquences distinctes d'une séquence donnée dont la norme est inférieure à $M$ en généralisant la formule proposée dans [36]. Concernant le tirage uniforme (i.e., $\psi(\sigma_{||patt||\leq M}(\delta(L)) \triangleleft t)$), nous avons proposé une méthode avec rejet qui tire uniquement la première occurrence de chaque sous-séquence (dite forme canonique) afin d'éviter de sur-échantillonner les occurrences multiples. Ainsi, la séquence $\langle ac \rangle$ sera tirée seulement à partir de l'occurrence $\langle (\mathbf{a}b)\mathbf{c}(ac) \rangle$. Cette approche de tirage avec une forme canonique pourrait avantageusement être utilisée pour étendre notre approche à d'autres langages structurés.

**Données distribuées [31]** Dans ce dernier exemple, le langage des motifs est à nouveau celui des itemsets mais nous considérons une contrainte de distribution sur le jeu de données. Une même transaction (i.e., avec la même valeur pour l'attribut *tid*) peut être répartie sur plusieurs tables $D_1, \ldots, D_K$ situées sur des sites distincts. Dans ce contexte, échantillonner les motifs suivant la fréquence revient à évaluer la requête suivante :

$$\psi^{k}_{freq}(\gamma_{patt,\texttt{COUNT}(trans)\rightarrow freq}(\delta(L) \subseteq (D_1 \bowtie \ldots \bowtie D_K)))$$

Malheureusement, l'opération $D_1 \bowtie \ldots \bowtie D_K$ correspondant à une centralisation des données [3] a un coût de communication élevé. Afin d'éviter des échanges nombreux, le travail de thèse de Lamine Diop montre comment évaluer cette requête en utilisant seulement deux primitives (à savoir `sizeOf` et `itemAt`) avec chaque site distant. $\texttt{sizeOf}(D_i, tid)$ retourne la taille de la transaction *tid* de la table $D$. $\texttt{itemAt}(D_i, tid, pos)$ retourne l'item à la position *pos* de la transaction *tid* de la table $D$. Plus précisément, la primitive `sizeOf` est utilisée pour connaitre la taille de chaque transaction sur chaque fragment. Il est ainsi possible d'évaluer $\gamma_{trans,\texttt{COUNT}(patt)\rightarrow nb}(\delta(L) \subseteq D)$ essentiel pour la première étape. Ensuite, le tirage uniforme récupère seulement les items nécessaires avec la primitive `itemAt`. Comme les deux primitives requièrent peu de communications, le tirage d'une collection de motifs à la demande par cette approche est beaucoup moins coûteux que le calcul hors-ligne de tous les motifs fréquents. Nous avons étendu ce principe à une classe de mesures fondées sur la norme (e.g., l'aire ou ajout de contrainte sur la taille).

### 4.1.3 Vers des approches génériques

Nous avons établi le lien entre l'échantillonnage de motifs et l'algèbre relationnelle en ajoutant l'opérateur d'échantillonnage à PORA. Par réécriture, nous avons reformulé la procédure aléatoire en deux étapes originellement proposée par Boley et al [13].

L'un des points forts de l'échantillonnage de motifs est d'offrir un accès direct à l'ensemble des motifs à faible coût. Nous avons notamment montré pour plusieurs cas de langages et données complexes que cette approche retourne instantanément un échantillon de motifs avec une garantie sur sa distribution. Au niveau physique, il reste néanmoins à unifier les propositions actuelles pour étendre l'utilisation à plus de langages et de mesures.

3. Cette centralisation est possible car nous considérons le langage des itemsets qui ne requiert pas de propriété particulière sur les $D_i$ (par exemple, un ordre).

Bien sûr, cette approche fondée sur l'aléatoire n'est pas utilisable dans tous les contextes applicatifs car contrairement à la section précédente, les motifs tirés ne sont pas les motifs maximisant les préférences. Cependant, le biais de tirage concentre l'extraction sur les *bons* motifs (à défaut des *meilleurs*) et évite de se limiter à une infime partie du langage. Cette diversité est un atout considérable lorsque l'on souhaite construire un modèle qui reflète l'intégralité des données comme nous le verrons dans la section suivante. Elle peut même se muer en sérendipité afin de découvrir des motifs originaux de manière interactive (voir la section 4.3).

## 4.2 Système anytime pour les modèles fondés sur les motifs

Un algorithme anytime est un algorithme qui peut fournir une réponse à chaque instant et dont le résultat s'améliore continuellement avec l'augmentation du budget alloué pour tendre vers la solution optimale [100]. Cette propriété anytime est intéressante car elle est propice à l'interactivité. Par exemple, l'utilisateur peut demander une première réponse rapidement pour vérifier que sa requête est la bonne avant de laisser plus de temps au processus pour construire une réponse de meilleure qualité. La section précédente a déjà souligné que l'échantillonnage de motifs est une technique qui retourne instantanément des motifs à la demande. Dans cette section, nous allons tirer profit de cette technique pour proposer des algorithmes anytime pour les approches qui reposaient auparavant sur une collection préalable de motifs extraits.

Pour rappel, les modèles construits en deux étapes sont confrontés à plusieurs limites liées à l'extraction exhaustive de motifs de la première étape (voir la sous-section 3.2.3). Le temps d'extraction de cette phase est incompressible et empêche de construire immédiatement un modèle. Afin de réduire ce temps, l'utilisateur aura donc tendance à fixer des paramètres d'extraction pour réduire l'espace de recherche (e.g., seuil de support minimum élevé) ce qui diminuera potentiellement la qualité du modèle construit.

Intuitivement, la diversité de l'échantillonnage est propice à la construction d'un ensemble de motifs bien représentatif du langage (contrairement aux méthodes d'énumération où les motifs successivement extraits sont plutôt similaires). Plutôt que d'opérer sur la collection complète de motifs $L$, l'idée générale de cette section est de s'appuyer sur un échantillon de taille $k$ à savoir $\psi^k(L)$. Lorsque l'échantillon grossit, le résultat d'une requête PORA exécutée sur l'échantillon sera proche du résultat qu'on aurait obtenu avec la collection complète :

**Propriété 2** *Pour une requête PORA $q$ portant sur une collection de motifs $L[U]$, on a l'égalité suivante :*

$$\lim_{k \to +\infty} q(\gamma_U(\psi^k(L))) = q(L)$$

Cette propriété est évidente puisqu'à la limite, tous les motifs de $L$ seront inclus dans l'échantillon (et même parfois avec des répétitions qui seront supprimées par $\gamma_U(\cdot)$). Bien sûr, il est aisé de réaliser un système anytime en augmentant petit à petit l'échantillon et en évaluant régulièrement la requête $q$.

Malheureusement, l'utilisation de la propriété 2 soulève plusieurs verrous majeurs :

1. Si la collection de motifs $L$ est grande (ce que nous souhaitons pour avoir un modèle de qualité), alors conserver tous les motifs $\gamma_U(\psi^k(L))$ est coûteux en mémoire.
2. Si l'évaluation de $q$ est coûteuse, alors sa répétition successive va entraver les performances du système.
3. La propriété 2 garantit la convergence mais celle-ci peut s'avérer lente.

La sous-section suivante lève en partie ces verrous pour rendre anytime des modèles construits en deux étapes. Ensuite, la sous-section 4.2.2 se concentre sur une requête spécifique pour identifier les données aberrantes. Plus précisément, cette nouvelle méthode

de calcul du FPOF s'appuie sur l'échantillonnage de motifs pour se passer de la première étape. A notre connaissance, il s'agit d'un des premiers travaux à offrir des garanties statistiques sur le modèle produit [51, 52] et à offrir un système anytime [50] pour les modèles fondés sur les motifs.

### 4.2.1 Construction itérative de modèles anytime

Cette partie montre comment construire de manière anytime n'importe quel modèle suivant l'approche en deux étapes décrite dans la section 3.2 (page 30). Même si TwoSteps n'est pas formellement une requête PORA, il s'appuie sur une répétition de requêtes PORA. Par conséquent, il est possible d'utiliser la propriété 2 pour en déduire l'égalité suivante :

$$\lim_{k \to +\infty} \textsc{TwoSteps}(D, \psi^k(L), <^q, \lhd) = \textsc{TwoSteps}(D, L, <^q, \lhd)$$

Inévitablement, cette application naïve de la propriété 2 soulève les trois mêmes limites. L'algorithme 2 montre comment lever les deux premières.

**Algorithm 2** Anytime Pattern-based modeling algorithm (Anytime)

**Input:** A dataset $D$, a pattern set $L$, a total quality order $<^q$ and a cover relation $\lhd$
**Output:** A pattern-based model $M$

1: $M := \emptyset$
2: **repeat**
3: $\quad M := \textsc{TwoSteps}(D, M \cup \psi(L), <^q, \lhd)$
4: **until** the user stops the process
5: **return** $M$

L'algorithme Anytime prend les mêmes paramètres que TwoSteps. Après avoir initialisé le modèle avec l'ensemble vide, une boucle principale est répétée jusqu'à ce que l'utilisateur souhaite avoir un résultat et stoppe l'exécution. A chaque itération, l'algorithme TwoSteps est appliqué sur le modèle courant auquel est ajouté un motif tiré dans la collection (ligne 3). Bien entendu, cet algorithme converge vers la solution qu'on aurait obtenu avec TwoSteps :

**Théorème 2 (Construction anytime)** *L'algorithme* Anytime *tend à retourner le modèle de* TwoSteps *pour tout jeu de données $D$, ensemble de motifs $L$, ordre $<^q$ et relation de couverture $\lhd$.*

En plus de la convergence, l'algorithme Anytime évite de conserver tous les motifs tirés. Seuls les motifs utiles au meilleur modèle courant sont retenus. D'une part, cela diminue fortement la consommation mémoire qui est bornée par $\arg\max_{M \subseteq L} |\textsc{TwoSteps}(D, M, <^q, \lhd)|$ (verrou 1 levé). D'autre part, l'exécution de TwoSteps sur peu de motifs sera plus rapide que sur la collection complète (verrou 2 levé). A noter que pour garantir une amélioration continuelle de la qualité du modèle,

il est nécessaire de garder le précédent modèle en plus de celui en cours de construction. En effet, nous sommes certains que le modèle courant deviendra plus proche du modèle optimal, mais il faudra un certain temps pour cela. Pendant cette durée, il est préférable de retourner le modèle précédent à l'utilisateur.

Clairement le fait de pouvoir tirer plusieurs fois le même motif permet difficilement de conclure sur la rapidité de la convergence. Il faudrait donc l'évaluer pour des instanciations spécifiques. Par simplicité, nous proposons ci-dessus d'effectuer un tirage uniforme sur la collection $L$. Il serait probablement plus judicieux de biaiser le tirage pour favoriser les motifs préférés au sens de $<^q$.

### 4.2.2 Détection de données aberrantes

Cette partie se concentre sur une méthode de détection de données aberrantes fondée sur une collection de motifs fréquents. Il s'agit d'un modèle construit en deux étapes mais pas via une construction itérative comme décrit dans la section 3.2.

**Frequent Pattern Outlier Factor** Intuitivement, si une transaction contient des motifs très fréquents, alors elle partage ses caractéristiques avec de nombreuses autres transactions. En s'appuyant sur ce constat, le score d'aberration d'une transaction est la somme des fréquences des motifs qu'elle contient normalisée par la somme totale des fréquences. Plus ce score est faible, plus la transaction a une chance d'être une donnée aberrante. La définition suivante donne une définition formelle :

**Définition 12 (FPOF [58])** *Pour un jeu de données $D$, le score FPOF (pour* Frequent Pattern Outlier Factor*) se formule algébriquement de la manière suivante :*

$$\gamma_{trans,\frac{\texttt{SUM}(freq)}{\gamma_{\texttt{SUM}(freq)}(F)}\rightarrow FPOF}(F \triangleleft D)$$

*où $F := \gamma_{patt,\texttt{COUNT}(trans)\rightarrow freq}(\delta(L) \triangleleft D)$ liste chaque motif du jeu de données avec sa fréquence.*

Ainsi, chaque tuple calculé est un couple d'une transaction (attribut *trans*) et d'un score d'aberration (attribut *FPOF*). Si une transaction contient peu de motifs dont la fréquence est faible, alors son score sera proche de 0. Inversement, une transaction contenant de nombreux motifs très fréquents aura un score élevé. En pratique, un seuil minimal de fréquence est utilisé pour rendre faisable l'extraction de la collection $F$. Nous verrons dans le paragraphe suivant qu'il n'est pas nécessaire de disposer d'un tel seuil avec notre méthode anytime.

De prime abord, on pourrait imaginer que la complexité de ce problème est exponentielle à cause du calcul de la collection $F$. En réalité, pour les itemsets, nous avons proposé un algorithme avec une complexité quadratique en fonction du nombre de transactions en

démontrant l'égalité suivante dans le cas des itemsets :

$$\begin{aligned}
\gamma_{trans,\texttt{SUM}(freq)}(F \subseteq D) &= \gamma_{trans,\texttt{SUM}(freq)}(\gamma_{patt,\texttt{COUNT}(trans)\rightarrow freq}(\delta(L) \subseteq D') \subseteq D) && (4.1)\\
&= \gamma_{trans,\texttt{COUNT}(*)}(\pi_{patt}((\delta(L) \subseteq D') \subseteq D)) && (4.2)\\
&= \gamma_{trans,\texttt{COUNT}(*)}(\sigma_{patt\subseteq trans}((\delta(L) \subseteq D') \times D)) && (4.3)\\
&= \gamma_{trans,\texttt{COUNT}(*)}(\sigma_{patt\subseteq trans \wedge patt\subseteq trans'}(\delta(L) \times D' \times D)) && (4.4)\\
&= \gamma_{trans,\texttt{COUNT}(*)}(\sigma_{patt\subseteq (trans\cap trans')}(\delta(L) \times D' \times D)) && (4.5)\\
&= \gamma_{trans,\texttt{SUM}(2^{|trans\cap trans'|})}(D \times D') && (4.6)
\end{aligned}$$

La première ligne formule le FPOF pour les itemsets i.e., que la relation de couverture utilisée est la relation $\subseteq$. Ensuite, plutôt que de calculer la fréquence de chaque motif *patt*, la seconde ligne compte directement le nombre de couples $(patt, trans')$ pour une transaction *trans*. Les lignes 3 et 4 utilisent la définition de l'opérateur de couverture pour revenir à des produits cartésiens. On constate qu'il s'agit de compter le nombre de motifs commun aux deux transactions (ligne 5) ce qui peut être calculé efficacement (ligne 6).

**Echantillonnage de motifs pour calculer le FPOF** Malgré une complexité quadratique, l'algorithme utilisant la réécriture ci-dessus reste coûteux pour les grands jeux de données justifiant bien une proposition anytime. Bien sûr, nous pourrions directement utiliser la propriété 2. Cependant, il serait nécessaire de conserver tous les motifs en mémoire à cause du tirage avec remise pour éviter de compter plusieurs fois la fréquence d'un motif pour une même transaction. En tirant les motifs proportionnellement à la fréquence, le nombre d'occurrences d'un motif au sein d'un échantillon approxime sa fréquence. De cette manière, il suffit de compter le nombre de motifs de l'échantillon contenus dans une transaction pour approximer son score (si un même motif est tiré $n$ fois, on le comptera $n$ fois). Plus formellement, on a :

**Théorème 3** *Il est possible d'approximer le FPOF de chaque transaction d'un jeu de données D en s'appuyant sur la relation suivante :*

$$\lim_{k\rightarrow+\infty} \gamma_{trans,\frac{\texttt{SUM}(nb)}{\gamma_{\texttt{SUM}(nb)}(F_k)}\rightarrow FPOF}(F_k \triangleleft D) = \gamma_{trans,\frac{\texttt{SUM}(freq)}{\gamma_{\texttt{SUM}(freq)}(F)}\rightarrow FPOF}(F \triangleleft D)$$

*où* $F_k := \gamma_{patt,\texttt{COUNT}(*)\rightarrow nb}(\psi^k_{freq}(F))$ *et* $F := \gamma_{patt,\texttt{COUNT}(trans)\rightarrow freq}(\delta(L) \triangleleft D)$.

Pour chaque transaction *trans*, l'agrégat $\texttt{SUM}(nb)$ additionne le nombre d'occurrences *nb* (au sein de l'échantillon $\psi^k_{freq}(F)$) de chaque motif *patt* qu'elle contient. Contrairement à la propriété 2, il est à noter que le tirage n'est pas effectué de manière uniforme mais proportionnellement à la fréquence. En pratique, le calcul du FPOF avec cette approche consiste juste à tirer un motif proportionnellement à sa fréquence et d'incrémenter le score de toutes les transactions contenant ce motif. Chaque itération est donc instantanée (verrou 2 levé). De manière intéressante, il n'est pas nécessaire de conserver les motifs tirés en mémoire (verrou 1 levé). Enfin, nous avons montré comment utiliser l'inégalité de

Hoeffding pour approximer l'erreur du score *FPOF* pour chacune des transactions. Nous avons ainsi montré que la convergence de l'approche est rapide (verrou 3 levé).

Pour finir, le résultat du théorème 3 a été décliné en deux variantes algorithmiques. Soit l'utilisateur fixe l'erreur qu'il tolère sur le FPOF, un échantillon adéquat de taille $k$ est tiré pour calculer l'approximation du FPOF [51, 52]. Soit l'algorithme augmente la taille de l'échantillon en maintenant une approximation du FPOF, l'approximation courante est retournée dès que l'utilisateur le souhaite (en lui indiquant l'erreur maximale de cette approximation) [50].

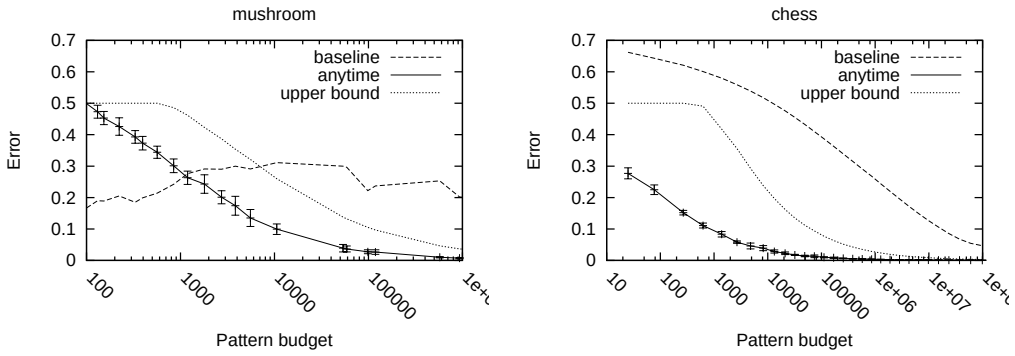



FIGURE 4.1 – Erreur du FPOF calculée avec un échantillon (méthode *anytime*) ou avec les motifs les plus fréquents (méthode *baseline*)

Pour illustrer l'efficacité du calcul du FPOF par échantillonnage, nous avons approximé le FPOF en tirant $k$ motifs selon la fréquence (méthode *anytime*) ou en utilisant les $k$ motifs les plus fréquents (méthode *baseline*). La figure 4.1 présente l'erreur moyenne du FPOF calculée par chacune de ses deux méthodes pour deux jeux de données de la littérature à savoir `mushroom` et `chess`. On constate que la convergence du calcul du FPOF est bien plus rapide avec la méthode par échantillonnage. D'ailleurs, l'erreur pratique de cette approche est bien inférieure à l'erreur théorique estimée avec l'inégalité de Hoeffding (courbe *upper bound*).

### 4.2.3 Vers la robustesse des modèles

Cette section a mis en lumière l'utilité de l'échantillonnage de motifs pour la construction de modèles de manière anytime.

Une leçon importante est qu'un échantillon de motifs aléatoire de taille modeste vaut mieux qu'une énorme collection de motifs extraite de manière exhaustive mais avec un seuil de support minimal. D'une part, l'échantillon est plus représentatif de l'ensemble du langage (y compris les motifs non-fréquents). Dans le cas du FPOF, 1000 motifs donnent une meilleure approximation qu'une collection complète contrainte en contenant mille fois plus. D'autre part, l'échantillonnage est une procédure aléatoire qui autorise l'utilisation d'outils statistiques pour borner l'erreur. Par exemple, il est possible d'offrir des garanties

sur l'approximation du FPOF.

Dans la première sous-section, nous avons montré qu'il était possible de rendre anytime la construction itérative de modèles sans construire une collection complète en mémoire. Il faudrait bien sûr mener une analyse de la rapidité de la convergence de cette approche pour différentes instanciations. Mais, il reste un champ bien plus vaste à explorer. A la ligne 3 de l'algorithme 2, le tirage des motifs se fait indépendamment du modèle déjà construit. Il serait intéressant d'envisager un tirage tenant compte du modèle ce qui permettrait de lever une des principales limites des modèles fondés sur l'extraction de motifs.

## 4.3 Interaction pour guider l'extraction

En pratique, comme nous l'avons déjà évoqué dans le chapitre 3, il est difficile pour un utilisateur d'exprimer son intérêt en formulant soit une contrainte, soit une relation de préférences. Pour remédier à cela, plusieurs travaux dont [98, 35] ont proposé d'apprendre de manière interactive les préférences de l'utilisateur. L'idée est de soumettre des motifs à l'utilisateur final et de bénéficier de ses retours pour mieux cibler ses attentes. En supposant que l'utilisateur dispose d'une relation de préférences sur les motifs, notée $\preceq_{pref}$, l'extraction de motifs interactive vise à apprendre cette relation tout en découvrant les motifs pertinents selon cette relation. La plupart des méthodes suivent un cadre général qui itère 3 phases [98] :

1. **Extraire :** Cette phase produit des motifs intéressants pour l'utilisateur. Si les premières itérations produisent des motifs peu en lien avec ses intérêts, le défi des itérations ultérieures est de parvenir à prendre en compte la dernière relation de préférences apprise $\preceq_{pref_i}$.
2. **Interagir :** Cette phase capture le point de vue de l'utilisateur sur les motifs extraits sous la forme de retours *implicites* (e.g., temps d'observation d'un motif ou clics) ou *explicites* (e.g., notation ou classement de motifs) où les retours explicites procurent les informations les plus précises. Pour aller à l'essentiel, si l'utilisateur indique qu'un motif $X$ est préféré à un autre motif $Y$, $X \preceq_{pref} Y$ est ajouté au retours utilisateur $\mathcal{F}$. Avec une notation, si l'utilisateur donne une meilleure note à $X$ que à $Y$, on pourra aussi ajouter $X \preceq_{pref} Y$ aux retours $\mathcal{F}$.
3. **Apprendre :** Cette phase généralise l'ensemble des retours $\mathcal{F}$ pour itérativement améliorer la relation de préférences $\preceq_{pref_i}$ de sorte que $\lim_{i\to\infty} \preceq_{pref_i} = \preceq_{pref}$. Cette généralisation requiert de disposer d'un modèle de préférences sous-jacent.

Ce processus interactif requiert une boucle courte avec une interaction rapide entre le système de fouille et l'utilisateur tout en relevant deux défis :

Un des défis dans ce cycle est d'assurer un bon *apprentissage actif* [35]. En effet, l'amélioration du modèle de préférences requiert un choix judicieux des motifs à fournir à l'utilisateur. Si l'étape d'extraction produit toujours des motifs similaires, le modèle de préférences ne pourra pas être amélioré. Cela signifie que l'étape d'extraction doit sélectionner des motifs divers et représentatifs (en plus d'avoir un intérêt pour l'utilisateur).

Un autre défi est le choix du modèle de préférences qui détermine la représentation de l'utilisateur. Ce modèle doit être suffisamment large pour ne pas manquer les caractéristiques qui captureront l'intérêt de l'utilisateur. Mais, si le modèle est trop complexe, il sera difficile de l'intégrer dans l'étape d'extraction [11, 83].

### 4.3.1 Echantillonnage de motifs interactif

Il paraît naturel de bénéficier de l'échantillonnage pour mettre en oeuvre le cycle : extraire, interagir et apprendre. En effet, au niveau de l'étape d'extraction, cette technique est suffisamment rapide pour que l'interaction soit de qualité. Au niveau de l'apprentissage, la diversité de cette procédure aléatoire est un atout.

**Algorithm 3** Interactive pattern sampling

**Input:** A dataset $D$, a multiset of feedback $FB$, a scoring query *score* and an oracle $\mathcal{O}$

1: **repeat**
2: $L := score(D, FB)$
3: Draw a pattern $X$ from $D$ according to its preference : $X := \psi_{pref}(L)$
4: Add the user feedback to $FB$ : $FB := FB \cup \{\mathcal{O}(X)\}$
5: **until** The user stops the process

Intuitivement, l'idée de l'échantillonnage de motifs est de tirer un motif proportionnellement à une mesure de préférences (*extraire*), de demander à l'utilisateur d'évaluer ce motif (*interagir*) et finalement, de modifier la mesure d'intérêt des motifs pour prendre en compte cette évaluation (*apprendre*). Le point crucial de cette approche est la mise à jour des préférences que nous modéliserons par une requête de score *score*.

L'algorithme 3 présente le squelette type d'une méthode interactive de découverte de motifs utilisant l'échantillonnage. Il prend en entrée un jeu de données $D$, un ensemble de retours $FB$ pour modéliser l'intérêt initial de l'utilisateur, la requête de score *score* (avec $pref \in \mathbf{sch}(q)$) et un oracle $\mathcal{O}$ pour déterminer les retours de l'utilisateur par un tuple. Plus précisément, cette requête de score retourne un ensemble de motifs dont l'intérêt est mesuré par $pref$. Tout comme pour les algorithmes anytime, l'utilisateur choisit ou non de répéter une boucle d'extraction. Cette boucle calcule d'abord le score pour chaque motif à la ligne 2 (*apprendre*). La ligne 3 tire un motif proportionnellement à la mesure $pref$ (*extraire*) et met à jour l'ensemble des retours à la ligne 4 en tenant compte de l'interaction avec l'utilisateur (*interagir*).

Illustrons cet algorithme générique en instanciant la toute première méthode proposée par [11] qui cherche les meilleurs motifs parmi une collection initiale de motifs fréquents. Cette méthode s'appuie sur un modèle multiplicatif i.e., que le score d'un motif (e.g., itemset) est le produit des scores de ses éléments (e.g., items). Le score de chaque item est calculé à partir des retours binaires sur les motifs qui peuvent être soit positifs, soit négatifs. Si un item est contenu dans des motifs ayant reçu 2 retours positifs et 1 retour négatif, son score sera $b^{4-1}$ où $b$ est une valeur fixe. En considérant une relation $FB[patt, v]$ stockant pour chaque motif $patt$ la valeur $v$ de son retour ($+1$ si positif et $-1$ sinon), il est possible de formuler cette méthode de manière algébrique :

$$score_1 \equiv \gamma_{patt, b^{\texttt{SUM}(v)} \to pref}\big(\underbrace{\gamma_{item, \texttt{SUM}(v) \to v}(\delta(I) \in FB)}_{\text{calcul du score de chaque item}} \in F\big)$$

où $I$ est l'ensemble des items et $F = \sigma_{freq \geq f}(\gamma_{patt, \texttt{COUNT}(*) \to freq}(\delta(L) \triangleleft D))$ est l'ensemble des motifs fréquents.

Pour mettre en oeuvre cette méthode, il sera nécessaire d'initialiser $FB$ avec la valeur 0 pour chaque item : $FB = \delta(I) \times \{\langle 0 \rangle\}$. La table 4.2 détaille les étapes de calcul de la requête $score_1$. Les 5 premières lignes de $FB$ correspondent à l'initialisation des items à 0. Ensuite, 3 retours ont été ajoutés pour les motifs $AB$, $BC$ et $BD$. A la fin du calcul, il est clair que les motifs $AB$ et $AC$ (resp. $D$) sont jugés plus (resp. moins) pertinents.

*FB*

| item | v |
|---|---|
| A | 0 |
| B | 0 |
| C | 0 |
| D | 0 |
| E | 0 |
| AB | 1 |
| BC | 1 |
| BD | -1 |

$\gamma_{item,\texttt{SUM}(v)\rightarrow v}(\delta(I) \in FB) \longrightarrow$

$score_{item}$

| item | v |
|---|---|
| A | 1 |
| B | 2-1 |
| C | 1 |
| D | -1 |
| E | 0 |

$\gamma_{patt,b^{\texttt{SUM}(v)}\rightarrow pref}(score_{item} \in F) \longrightarrow$

$score_1$

| patt | pref |
|---|---|
| A | $b^1$ |
| B | $b^{2-1}$ |
| C | $b^1$ |
| D | $b^{-1}$ |
| E | $b^0$ |
| AB | $b^{3-1}$ |
| AC | $b^2$ |
| AD | $b^{1-1}$ |
| ... | ... |

TABLE 4.2 – Exemple de calcul de la requête $score_1$

Le motif $E$ pour lequel on ne dispose pas d'information est neutre. Une des limites de la proposition de [11] est de se concentrer uniquement sur les motifs fréquents qui sont calculés initialement et indépendamment des préférences utilisateurs.

### 4.3.2 Caractérisation des transactions préférées

Nous avons également proposé une méthode d'échantillonnage interactif de motifs en nous basant sur un modèle de préférences sur les transactions [59, 53]. L'idée est qu'une partie des transactions est préférée à l'autre, mais on ne dispose pas de cet étiquetage au début du processus. L'objectif étant de caractériser les transactions préférées, la méthode propose des motifs à l'utilisateur focalisant sur ces transactions étiquetées au fur et à mesure grâce à ses retours binaires.

Comme dans la section précédente, on dispose d'un ensemble de retours utilisateur $FB[patt, v]$. L'utilisateur retourne $v = 1$ si le motif couvre les transactions qui l'intéressent (i.e., classe cible à découvrir) et $v = 0$, sinon. Ces retours sont utiles pour calculer un score moyen de préférence par transaction. Il ne reste plus qu'à tirer des motifs proportionnellement à la somme des scores moyens des transactions couvertes (i.e., fréquence pondérée par la préférence de chaque transaction). Plus précisément, cette méthode correspond à l'expression suivante :

$$score_2 \equiv \gamma_{patt,\texttt{SUM}(weight)\rightarrow pref}(\delta(L) \subseteq \underbrace{\gamma_{trans,\texttt{AVG}(v)\rightarrow weight}(FB \subseteq D)}_{\text{calcul des préférences}})$$

En réalité, il est nécessaire d'amender cette requête pour garantir une bonne convergence de l'apprentissage des préférences. Premièrement, les transactions non couvertes doivent tout de même avoir un score de 0.5 (en ajoutant $D \supseteq_\neg FB \times \{\langle 0.5 \rangle\}$ dans le calcul des préférences). Deuxièmement, il faut pondérer chaque valeur $v$ par le support au moment du tirage si l'on souhaite avoir une convergence de la méthode (en ajoutant cette information dans la relation $FB$ et en tenant compte lors du calcul de la moyenne). Enfin, une correction statistique de la moyenne est nécessaire pour éviter les cas limites. Pour cela, on peut calculer la moyenne en utilisant l'inégalité de Bennett (à nouveau, il faut améliorer le calcul de l'agrégat).

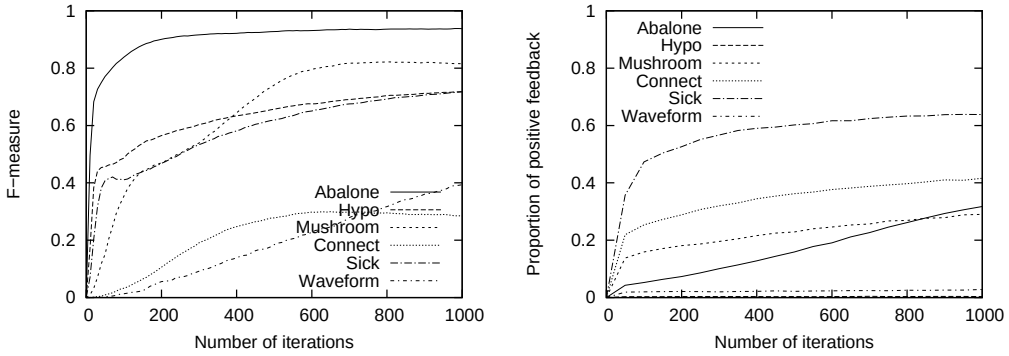



FIGURE 4.2 – Exemples de la caractérisation interactive d'un jeu de données non-étiqueté

Nous avons appliqué notre approche pour caractériser des jeux de données de la littérature tout en apprenant les classes. La figure 4.2 à gauche présente la F-mesure des étiquettes prédites en fonction du nombre de retours de l'utilisateur. Suivant le jeu de données, on retrouve plus ou moins rapidement les classes initialement inconnues. Pour le jeu de données `Abalone`, il suffit de 200 motifs évalués par l'utilisateur pour avoir une F-mesure de 90%, i.e., retrouver la bonne classe pour la plupart des transactions. Parallèlement, sur la courbe de droite, on constate que les motifs retournés à l'utilisateur sont de plus en intéressant puisque la proportion de retours positifs augmente avec le nombre d'itérations.

### 4.3.3 Vers l'apprentissage actif

Comme indiquée en introduction, la découverte de motifs interactive s'inscrit dans le domaine de l'apprentissage actif où le système apprend les préférences de l'utilisateur. Mais, dans notre contexte, la question posée à l'utilisateur ne concerne pas une donnée mais un motif. Ce motif se doit d'être intéressant afin de faciliter l'engagement de l'utilisateur en lui décrivant ses données.

A nouveau, l'échantillonnage s'avère pertinent pour concevoir des méthodes de découverte interactive de motifs. Outre la rapidité du tirage propice à un couplage fort entre l'utilisateur et le processus, la grande diversité des motifs tirés assure un bon apprentissage des préférences. Comme pour la construction de modèles, il est aussi possible d'avoir des garanties sur la convergence et sa rapidité. En revanche, une limite est la faible expressivité des modèles de préférences qui peuvent donner lieu à un échantillonnage de motifs efficace.

L'interaction sert à construire une relation de préférence sur les motifs plutôt que de demander à l'utilisateur de l'expliciter directement. D'apparence, cette approche semble bien plus simple pour l'utilisateur qui n'a pas pas à formaliser son intérêt. Pourtant, les deux méthodes décrites ne peuvent pas découvrir les mêmes motifs car elles reposent sur des hypothèses de modèles de préférences bien distinctes. Par ailleurs, il est clair

qu'il n'existe pas de modèle universel. Par conséquent, l'utilisateur (ou le concepteur) de la méthode doit expliciter le modèle de préférences qu'il juge le plus pertinent pour la tâche visée... Bien que très nettement amoindries, certaines difficultés évoquées dans le précédent chapitre sur la modélisation de l'expertise émergent aussi.

# Chapitre 5

# Conclusion

## Bilan

**Importance de l'utilisateur** Ce mémoire a retracé nos travaux en découverte de motifs où l'importance de l'utilisateur n'a cessé de croître : à travers la déclarativité (chapitre 2) à travers ses préférences explicites (chapitre 3) ou à travers son analyse des motifs (chapitre 4). Cela nous a aussi conduit à nous inscrire dans trois tendances majeures des travaux de la communauté :

- **Plus de rapidité :** La première préoccupation de la découverte de motifs était de développer des algorithmes permettant de renvoyer rapidement une réponse malgré un espace de recherche considérable. La rapidité d'exécution justifiait l'extraction de motifs *fréquents* même s'ils présentaient un intérêt limité pour les utilisateurs finaux. Même si ce n'était pas la finalité, une bonne partie de nos travaux ont porté sur l'optimisation des algorithmes pour extraire efficacement les motifs souhaités. Récemment, l'avènement de la découverte de motifs interactive a suscité l'intérêt pour des réponses encore plus rapides (mais l'exhaustivité n'est plus nécessaire) avec l'échantillonnage de motifs.
- **Plus de qualité :** Dans les années 2000, le passage de l'extraction de motifs fréquents à l'extraction de motifs sous contraintes était une première étape très importante pour améliorer la qualité des motifs extraits. L'extraction de motifs guidée par les préférences comme les motifs Pareto optimaux va un peu plus loin en se concentrant sur les motifs maximisant un critère de qualité. Toutes ces méthodes sont clairement destinées à tirer parti des connaissances explicites fournies par l'utilisateur. Néanmoins, nous jugeons plus prometteur les travaux autour de la découverte de motifs interactive qui s'appuie sur un modèle appris implicitement à partir des retours de l'utilisateur.
- **Plus de simplicité :** Les paramètres d'entrée des méthodes d'extraction de motifs illustrent parfaitement le mouvement de simplification auquel nous avons participé. Les premiers utilisateurs ont été invités à sélectionner l'algorithme approprié pour chaque type de jeu de données. Ensuite, il suffisait à l'utilisateur de formuler ses contraintes et ses seuils. Enfin, la découverte de motifs guidée par les préférences a retiré les seuils. Actuellement, la découverte de motifs interactive élimine même

la nécessité pour l'utilisateur de spécifier explicitement son intérêt. Parallèlement, cette simplification de la spécification du problème s'est accompagnée de travaux sur la simplification des méthodes de résolution grâce à des cadres génériques tel que notre algèbre relationnelle orientée motif.

**Intérêt des motifs découverts** Même si nos travaux se sont largement appuyés sur l'intérêt subjectif d'un utilisateur, ils ont aussi nourri la réflexion concernant un cadre théorique pour l'intérêt objectif de connaissances. Pour commencer, nous avons noté que la découverte de motifs locaux requiert une algèbre plus expressive (comme PORA) que l'algèbre relationnelle traditionnelle dédiée à la seule manipulation de données. En revanche, PORA est insuffisante pour construire des modèles globaux sans ajouter un opérateur supplémentaire (e.g., opérateur de point fixe). La qualité des motifs/modèles extraits augmentent avec l'expressivité de l'algèbre utilisée. Ensuite, pour les motifs locaux, nous avons formalisé l'implication d'une relation au sein d'une requête PORA avec trois degrés (i.e., indépendance, dépendance locale et dépendance globale). Les requêtes les plus élaborées ont tendance à reposer sur plus de dépendances globales mettant en oeuvre de nombreuses comparaisons entre tuples. Récemment, dans [22], nous avons aussi proposé une complexité en évaluation qui mesure le nombre de fréquences nécessaires pour vérifier si la propriété **P** est satisfaite pour un motif $X$ donné. A nouveau, les mesures d'intérêt avec une complexité en évaluation plus forte tendent aussi à extraire les meilleurs motifs. En résumé, les processus découvrant les meilleurs motifs sont ceux qui requièrent le plus de complexité mesurable avec l'expressivité de l'algèbre, avec le nombre de relations en dépendances ou avec le nombre d'évaluations de la fréquence.

**Paradoxe de la fouille de données** Au cours de ces dernières années, nous avons mené plusieurs collaborations pour appliquer nos méthodes en médecine sur le cancer du sein [84, 85] et en chimie [90, 97]. Mais, de nombreuses autres tentatives passionnantes se sont révélées infructueuses. Il s'avère souvent plus facile de redécouvrir des connaissances d'un domaine que d'en découvrir de nouvelles. Ces expériences nous conduisent à formuler un paradoxe : *Plus une connaissance serait intéressante à découvrir, moins on aurait de chance de la découvrir.* En effet, plus une connaissance serait intéressante à découvrir, moins on aurait de chance de disposer des données et de l'expertise adéquates pour la découvrir. Illustrons ce paradoxe avec une collaboration naissante avec des médecins du CHU de Tours concernant la maladie de la sclérose latérale amyotrophique (SLA). Cette maladie est peu connue et les médecins jugent prometteuses nos techniques d'exploration de données pour mieux caractériser cette maladie et son évolution. Malheureusement, du fait de sa méconnaissance (à cause de sa rareté et de sa diversité), il n'y a pas un référentiel définissant les caractéristiques cliniques à recueillir auprès des patients et les données disponibles sont hétérogènes entre les centres hospitaliers. Parallèlement, cette hétérogénéité conduit à un recueil opérationnel peu automatisé propice à des problèmes d'intégration de données (e.g., nombreuses données manquantes ou multiples identifiants pour un même patient). Par conséquent, les données sont peu nombreuses, hétérogènes et difficiles à préparer en vue d'une analyse. Par ailleurs, comme la SLA est une maladie

complexe et mal connue, l'expertise est difficile à formuler. Elle s'avère très parcellaire et le succès de nos approches centrées sur l'utilisateur est incertain.

Dans ce contexte, pour se dispenser de l'expertise, il serait tentant de se tourner vers l'apprentissage profond pour faire un plongement des données sur quelques dimensions. Cette approche nous paraît néanmoins incertaine dans un contexte où le recueil insuffisamment automatisé des données empêche d'atteindre des volumes conséquents. Par ailleurs, un plongement est efficace pour expliquer un problème sous la forme d'un vecteur numérique, mais cela s'oppose à nos méthodes privilégiant une caractérisation discrète et qualitative.

## Perspectives

**Données ouvertes liées** Le web sémantique n'est pas nouveau et son utilisation pour la découverte de connaissances a été envisagée dès son origine [10]. Pourtant, nous estimons que le web sémantique est très largement sous-exploité du fait qu'il soit considéré comme un outil plutôt que comme un objet d'étude. En effet, il est souvent utilisé pour ajouter une couche sémantique afin d'améliorer l'analyse d'autres données textuelles ou relationnelles. Pourtant, les données ouvertes liées forment une double-réponse aux limites révélées par le paradoxe de la fouille de données à savoir le manque de données et d'expertise pour ouvrir sur de nouvelles perspectives de recherche.

Premièrement, les données ouvertes liées (Linked Open Data, LOD) constitue un gisement de données prometteur pour la découverte de connaissances. Bien sûr, de plus en plus de données sont ouvertes et suivent des standards facilitant l'acquisition et la préparation de données en vue d'une exploration. Mais, de notre point de vue, la force des données liées est de connecter des données de sources diverses ouvrant la voie à des découvertes d'associations inattendues. Par exemple, [64] illustre l'intérêt de la fouille de données dans les données ouvertes liées pour procéder à des recoupements interdisciplinaires entre des données médicales et environnementales. En contrepartie, les données ouvertes liées soulèvent des verrous scientifiques d'importance. Du fait qu'elles constituent une base de connaissances, il faut intégrer à la découverte de motifs (processus inductif) le raisonnement (processus déductif) [82]. Mais, en plus, cette tâche est particulièrement ardue car elle doit être opérée sur un volume gigantesque et distribué à l'échelle du web. Cette tâche se complique encore si l'on souhaite éviter la centralisation des données en utilisant uniquement des requêtes SPARQL avec des points d'accès publics [31, 94]. **Une perspective de recherche est donc de rendre possible l'accès intensif aux données requis pour la découverte de motifs à l'échelle du web sémantique en bénéficiant de capacités de raisonnement.**

Deuxièmement, les données ouvertes liées offrent l'opportunité de diffuser les motifs découverts à une large échelle pour utilisation et validation. Si on effectue cet effort sur le partage des motifs comme celui entrepris au niveau des données, un gain comparable peut être espéré. Comme il est inenvisageable de connaitre toutes les personnes intéressées par l'utilisation ou l'expertise des motifs découverts, cette approche maximise les chances de toucher ces bonnes personnes. Par ailleurs, nous pensons que cette approche répond à deux enjeux plus généraux pour la science comme la science ouverte et la science participative.

Cependant, cette approche s'oppose au paradigme de la découverte centrée sur l'utilisateur puisque l'on ne dispose plus de l'utilisateur au moment de la découverte des motifs. Il reste donc à ré-inventer la découverte de motifs de sorte à associer chaque motif découvert à son contexte un peu comme chaque page web est associée à son contexte (i.e., profil de navigation, requête par mots clés). Dans cette direction, il paraît intéressant d'opérer un rapprochement avec les méthodes mises en oeuvre en Recherche d'Information. **Par conséquent, de nouveaux cadres devraient fédérer des expertises afin de proposer des approches de découverte de motifs orientées « communauté » plutôt que orientées « utilisateur ».**

**Incomplétude, biais et véracité** Les données ouvertes liées sont un gisement de données propices aux associations inattendues, mais elles sont loin d'être parfaites. Premièrement, comme la plupart des bases de connaissances, l'hypothèse du monde ouvert complique leur analyse car une information non-renseignée n'est pas forcément inexistante ou fausse. Même si cette problématique n'est pas nouvelle, elle est mal prise en compte par la plupart des méthodes d'apprentissage plutôt conçues pour fonctionner avec l'hypothèse du monde clos. Deuxièmement, les données ouvertes liées sont issues d'agglomérations opportunistes de bases de données et de productions participatives. Par construction, ces données sont donc particulièrement biaisées et notre récent travail [89] [1] qui s'appuie sur la loi de Benford pour mesurer le nombre minimum de faits manquants pour une relation, montre à quel point il est délicat d'ignorer ces biais. Même si ce phénomène est moins accentué dans des bases de données traditionnelles, il demeure épineux. Dans le cas de la SLA, l'expertise sur la maladie évolue ainsi que le périmètre des patients considérés comme souffrant de cette maladie. Bien sûr, ignorer ce biais sur la population étudiée garantit l'apprentissage de connaissances biaisées voire fausses. Détecter et corriger la représentativité d'un échantillon donné s'avère être un verrou scientifique de taille particulièrement négligé en apprentissage où la plupart des résultats utilisent l'hypothèse de variables indépendantes et identiquement distribuées.

De manière générale, quelque soit le jeu de données $\mathcal{D}$ et sa qualité intrinsèque, la découverte de motifs devrait extraire des motifs dont la propriété $\mathbf{P}$ est vérifiée dans le jeu de données idéal $\mathcal{D}^*$ plutôt que $\mathcal{D}$. Ainsi, nous devrions extraire les motifs satisfaisant la contrainte dans le jeu de données idéal $\mathcal{D}^*$ (et non pas $\mathcal{D}$), extraire les motifs non-dominés dans $\mathcal{D}^*$ (et non pas $\mathcal{D}$) et échantillonner par rapport à $\mathcal{D}^*$ (et non pas $\mathcal{D}$). En fait, cette nouvelle façon d'envisager la découverte de motifs signifie que nous ne cherchons plus à vérifier la propriété $\mathbf{P}$ dans $\mathcal{D}$, mais plutôt une propriété voisine $\widetilde{\mathbf{P}}$ qui garantit réellement la propriété $\mathbf{P}$ dans $\mathcal{D}^*$ :

$$\widetilde{\mathbf{P}}(X, \mathcal{D}) \Rightarrow \mathbf{P}(X, \mathcal{D}^*)$$

Il existe déjà des travaux dans cette direction avec l'hypothèse que $\mathcal{D}$ soit un échantillon aléatoire de $\mathcal{D}^*$ : extraction de motifs significatifs [57], extraction de motifs sur un échantillon [95] et même, notre récent travail sur la détection de contraintes de cardinalité maximale [7, 47] [2]. Est-il possible d'aller au-delà lorsque $\mathcal{D}$ est de plus mauvaise

1. L'article est disponible en annexe.
2. L'article [47] est disponible en annexe.

qualité ? Par ailleurs, il est illusoire d'espérer avoir le sens réciproque dans l'équivalence ci-dessus à moins de disposer d'une base de données parfaite. Cela signifie que nous devons définitivement renoncer à la notion d'exhaustivité (sur $\mathcal{D}^*$) en découverte de motifs. **Un défi majeur est de mieux qualifier les données d'apprentissage pour mieux qualifier la qualité des motifs découverts.**

# Chapitre 6

# Annexe

Cette annexe contient une sélection de 6 articles :

— A. Giacometti, P. Marcel, and A. Soulet. A relational view of pattern discovery. In J. X. Yu, M.-H. Kim, and R. Unland, editors, *DASFAA (1)*, volume 6587 of *Lecture Notes in Computer Science*, pages 153–167. Springer, 2011.

— A. Soulet, C. Raïssi, M. Plantevit, and B. Crémilleux. Mining dominant patterns in the sky. In D. J. Cook, J. Pei, W. Wang, O. R. Zaïane, and X. Wu, editors, *ICDM*, pages 655–664. IEEE, 2011.

— B. Crémilleux, A. Giacometti, and A. Soulet. How your supporters and opponents define your interestingness. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 373–389. Springer, 2018.

— L. Diop, C. T. Diop, A. Giacometti, D. Li, and A. Soulet. Sequential pattern sampling with norm constraints. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 89–98. IEEE, 2018.

— A. Soulet, A. Giacometti, B. Markhoff, and F. M. Suchanek. Representativeness of Knowledge Bases with the Generalized Benford's Law. In *ISWC*, 2018.

— A. Giacometti, B. Markhoff, and A. Soulet. Mining significant maximum cardinalities in knowledge bases. In *International Semantic Web Conference, to appear.*

# A Relational View of Pattern Discovery

Arnaud Giacometti, Patrick Marcel, and Arnaud Soulet

Université François Rabelais Tours, LI
3 place Jean Jaurès
F-41029 Blois France
forename.surname@univ-tours.fr

**Abstract.** The elegant integration of pattern mining techniques into database remains an open issue. In particular, no language is able to manipulate data and patterns without introducing opaque operators or loop-like statement. In this paper, we cope with this problem using relational algebra to formulate pattern mining queries. We introduce several operators based on the notion of cover allowing to express a wide range of queries like the mining of frequent patterns. Beyond modeling aspects, we show how to reason on queries for characterizing and rewriting them for optimization purpose. Thus, we algebraically reformulate the principle of the levelwise algorithm.

## 1 Introduction

Pattern discovery is a significant field of Knowledge Discovery in Databases (KDD). A broad spectrum of powerful techniques for producing local patterns has been developed over the two last decades [3–5]. But, it is widely agreed that the need of theoretical fusion between database and data mining still remains a crucial issue [14, 18, 23, 24]. We would force the pattern mining methods to fit in the relational model [1] which is the main database theory. Unlike most of the proposals [6, 10, 14, 16, 20, 23, 28, 33, 34], we desire to only address the pattern mining that we distinguish from the construction of global models [17] like decision trees.

Let us consider the popular task of frequent pattern mining [3] as a motivating example. Most works treat this task as a "black box" which input parameters are defined by the user [6, 7, 14, 16, 20, 28, 32, 34]. Instead of only specifying the minimal frequency threshold and the dataset, we think that the user query should fully formalize the notion of frequent patterns (e.g., it should describe how the frequency of a pattern is computed starting from the dataset). Ideally, we would like to express the frequent pattern mining query in the relational algebra in order to manipulate both the data and the patterns. As declarative aspects should be promoted on physical ones, a pattern discovery process has to be fully specified without considering algorithmic points. For this purpose, loop-like operators [10, 23, 33] are not relevant for us. Furthermore, the improvement of query performances mainly rests on physical optimizations in the field of pattern mining. Typically, the frequent pattern mining is efficiently performed

J.X. Yu, M.H. Kim, and R. Unland (Eds.): DASFAA 2011, Part I, LNCS 6587, pp. 153–167, 2011.

by an adequate implementation [3–5, 25]. Such algorithmic optimizations (even specified at a higher level [10, 23, 33]) reduce the opportunity of integrating other optimizations. We prefer to favor logical reasoning for optimizing query performances. For instance, the rewriting of the naive frequent pattern mining query should enable us to algebraically formulate the levelwise pruning [25].

The main goal of this paper is to propose an algebraic framework for pattern discovery for expressing a wide range of queries without introducing opaque operators or loop-like statements. Our framework brings two meaningful contributions: expressive modeling and logical reasoning. First, it allows a large set of queries manipulating relations which contain both data and patterns. We add to the relational algebra several specific operators, like the cover operator $\triangleleft$, to coherently and easily join such relations. We also define a new operator $\Delta$ for generating a language starting from a relation. Typically, the query $\sigma_{freq \geq f}(\gamma_{patt,\texttt{COUNT}(trans) \rightarrow freq}(\Delta(L) \triangleleft D))$ returns the patterns of language $L$ frequent in dataset $D$. Second, the pattern-oriented relational algebra enables to characterize and rewrite queries in order to optimize their performance. In particular, we formalize the notions of syntactic constraint [9] and global constraint [12] by characterizing the degree of dependence between a query and a relation. Besides, we not only benefit from usual query rewriting methods stemming from the relational model, but we also algebraically reformulate the levelwise pruning.

This paper is organized in the following way. Section 2 introduces basic notions about the relational algebra and the pattern discovery. Section 3 defines the cover-like and domain operators which are at the core of our algebra. We then study the properties of downward closure and independence in Section 4. We rewrite queries satisfying such properties for optimization purpose in Section 5. Finally, Section 6 provides a related work.

## 2 Basic Notions

### 2.1 Relational Algebra

We enumerate here our notations for the relational algebra mainly inspired from [1]. Let $\mathbf{att}$ be a set of distinct literals, named *attributes*, $\mathbf{dom}(A)$ denotes the finite *domain* of the attribute $A \in \mathbf{att}$. The *relation schema* (or relation for brevity) $R[U]$ denotes a relation named by $R$ where $U \subset \mathbf{att}$. An *instance* of $R$ is a subset of $\mathbf{dom}(U) = \times_{A \in U} \mathbf{dom}(A)$. Given a relation $R[A_1, \ldots, A_n]$, $R'$ renames the attributes $A_1, \ldots, A_n$ into $A'_1, \ldots, A'_n$. A *database schema* is a nonempty finite set $\mathbf{R} = \{R_1[U_1], \ldots, R_n[U_n]\}$ of relations. A *database instance* of $\mathbf{R}$ is a set $\mathbf{I} = \{I_1, \ldots, I_n\}$ such that $I_i$ is an instance of the relation $R_i$. Finally, a *query* $q$ maps a database instance to an instance of a relation. The set of attributes of this relation is denoted by $\mathbf{sch}(q)$. A query $q'$ is *equivalent* to $q$, denoted by $q' \equiv q$, iff for any database instance $\mathbf{I}$, one has $q'(\mathbf{I}) = q(\mathbf{I})$.

Let $I$ be an instance of $R$ and $J$ be an instance of $S$. The relations can be manipulated by means of set operators including Cartesian product $R \times S$ where $I \times J = \{(t, u) | t \in I \wedge u \in J\}$. If $R$ and $S$ are relations which have the same schema, then $R \cup S$, $R \cap S$ and $R - S$ are respectively the union, the intersection

and the difference of $R$ and $S$. *Selection:* $\sigma_f(I) = \{t | t \in I \wedge f(t)\}$ selects the tuples of $I$ satisfying the logical formula $f$ where $f$ is built from (i) the logical operators ($\wedge$, $\vee$ or $\neg$), (ii) the arithmetic relational operators and (iii) operands based on attributes and constants. *Extended projection:* $\pi_{A_1,\ldots,A_n}(I) = \{t[A_1,\ldots,A_n] | t \in I\}$ only preserves the attributes $A_1,\ldots,A_n$ of $R$. Besides, the projection also permits to extend the relation by arithmetic expressions and to (re)name expressions. For instance, $\pi_{A+B\rightarrow B',C\rightarrow C'}(R)$ creates a new instance where the first attribute named $B'$ results from the arithmetic expression $A+B$ and the second attribute corresponds to $C$, renamed $C'$. *Grouping:* $\gamma_{A_1,\ldots,A_n,\texttt{AGG}(B)}(I) = \{(a_1,\ldots,a_n,\texttt{AGG}(\pi_B(\sigma_{A_1=a_1\wedge\cdots\wedge A_n=a_n}(I))) \mid (a_1,\ldots,a_n) \in \pi_{A_1,\ldots,A_n}(I)\}$ groups tuples of $I$ by attributes $A_1,\ldots,A_n$ and applies an aggregate function AGG on $B$.

### 2.2 Pattern Discovery

We provide here an overview of pattern discovery based on [25, 32] focusing on the main proposals of the field. A *language* $\mathcal{L}$ is a set of patterns: itemsets $\mathcal{L}_\mathcal{I}$ [3], sequences $\mathcal{L}_\mathcal{S}$ [4] and so on [5]. A *specialization relation* $\preceq$ of a language $\mathcal{L}$ is a partial order relation on $\mathcal{L}$ [25, 27]. Given a specialization relation $\preceq$ on $\mathcal{L}$, $l \preceq l'$ means that $l$ is more general than $l'$, and $l'$ is more specific than $l$. For instance, the set inclusion is a specialization relation for the itemsets. Given two posets $(\mathcal{L}_1, \preceq_1)$ and $(\mathcal{L}_2, \preceq_2)$, a *cover relation* is a binary relation $\lhd \subseteq \mathcal{L}_1 \times \mathcal{L}_2$ iff when $l_1 \lhd l_2$, one has $l'_1 \lhd l_2$ (resp. $l_1 \lhd l'_2$) for any pattern $l'_1 \preceq_1 l_1$ (resp. $l_2 \preceq_2 l'_2$). The relation $l_1 \lhd l_2$ means that $l_1$ covers $l_2$, and $l_2$ is covered by $l_1$. The cover relation is useful to relate different languages together (e.g., for linking patterns to data). Note that a specialization relation on $\mathcal{L}$ is also a cover relation on $\mathcal{L}$ (e.g., the set inclusion is a cover relation for the itemsets).

The pattern can be manipulated by means of three kinds of operators non exhaustively illustrated hereafter. 1) *Pattern mining operators* produce patterns starting from a dataset: theory [25], MINERULE [26] and so on. More precisely, the *theory* denoted by $\mathcal{T}h(\mathcal{L}, q, \mathcal{D})$ returns all the patterns of a language $\mathcal{L}$ satisfying a predicate $q$ in the dataset $\mathcal{D}$ [25]. Typically, the minimal frequency constraint selects the patterns which occur in at least $f$ transactions [3, 4]: $freq(\varphi, D) > f$. As mentioned in introduction, we notice that the query $\mathcal{T}h(\mathcal{L}, freq(\varphi, \mathcal{D}) \geq f, \mathcal{D})$ does not make explicit how the frequency of a pattern is computed from the dataset. Other approaches find the $k$ patterns maximizing a measure $m$ in the dataset $\mathcal{D}$ [12, 15]. 2) *Pattern set reducing operators* compress a collection of patterns. For instance, the minimal and maximal operator denoted by $\mathcal{M}in(\mathcal{S})$ and $\mathcal{M}ax(\mathcal{S})$, return respectively the most general and specific patterns of $\mathcal{S}$ w.r.t. a specialization relation $\preceq$ [25]. The notion of negative and positive borders [25] is very similar. 3) *Pattern applying operators* cross patterns and data. For instance, the data covering operator $\theta_d(P, \mathcal{D}) = \{d \in \mathcal{D} | \exists p \in P : p \lhd d\}$ returns the data of $\mathcal{D}$ covered by at least one pattern of $P$ [32]. Dually, the pattern covering operator $\theta_p(P, \mathcal{D})$ returns the patterns of $P$ covering at least one element of $\mathcal{D}$ [32].

The next sections aim at stating an algebra based on the relational model to simultaneously and homogeneously handle data and patterns. In particular, all the manipulations of patterns described here will be expressed in our algebra.

## 3 Pattern-Oriented Relational Algebra

### 3.1 Pattern-Oriented Attributes

The pattern-oriented relational algebra pays attention to the attributes describing patterns, named *pattern-oriented attributes*. Indeed, several operations are specifically designed to handle such attributes which the domain corresponds to a pattern language together with a specialization relation.

**Definition 1 (Pattern-oriented attributes).** *The pattern-oriented attributes* **patt** *is a subset of the attributes:* **patt** $\subseteq$ **att** *such that for every* $A \in$ **patt**, **dom**$(A)$ *is a poset. Let* $U \subseteq$ **att** *be a set of attributes, the pattern-oriented attributes of* $U$ *is denoted by* $\widetilde{U}$.

For example, Table 1 provides instances of relations $D$, $L$ and $P$ containing pattern-oriented attributes. The relations $D[trans]$ and $L[patt]$ respectively describe a transactional dataset and the corresponding language in the context of (a) itemsets and (b) sequences. The relation $P[item, type, price]$ gives the item identifier, the type and the price of products. We consider that *trans*, *patt* and *item* are pattern-oriented attributes where **dom**$(item) = \mathcal{I}$ and **dom**$(trans) =$ **dom**$(patt) = \mathcal{L}_{\mathcal{I}}$ for itemsets (or $= \mathcal{L}_{\mathcal{S}}$ for sequences). Thereafter, the proposed queries can address instances where the domain of *patt* differs from that of *trans*.

Of course, the relations can be handled with relational operators. For instance, the query $\sigma_{patt \preceq \varphi}(L)$ returns all the patterns of $L$ being more general than the pattern $\varphi$. The formula $patt \preceq \varphi$ is allowed because $\sigma_{patt \preceq \varphi}(L) \equiv \pi_{patt}(\sigma_{patt=left \wedge right=\varphi}(L \times C))$ where the relation $C[left, right]$ extensively enumerates in its instance the tuples $(l, r)$ such that $l \preceq r$. On the contrary, the query $\sigma_{freq(patt,D) \geq f}(L)$ is not correct for computing the frequent patterns

**Table 1.** Instances for pattern discovery

(a) Itemset context

$D$

| trans |
|---|
| ABE |
| ABC |
| ABCD |
| AD |

Dataset

$L$

| patt |
|---|
| $\emptyset$ |
| A |
| B |
| C |
| D |
| E |
| AB |
| AC |
| AD |
| AE |
| BC |
| BD |
| BE |
| CD |
| CE |
| DE |
| ABC |
| ... |
| ABCDE |

Language of itemsets

(b) Sequence context

$D$

| trans |
|---|
| (AB)(E) |
| (AB)(C)(A) |
| (AB)(C)(D) |
| (B)(C)(D)(B) |

Sequential data

$L$

| patt |
|---|
| $\emptyset$ |
| (A) |
| (B) |
| (C) |
| (D) |
| (E) |
| (AB) |
| (A)(B) |
| ... |

Language of sequences

$P$

| item | type | price |
|---|---|---|
| A | *snack* | 3 |
| B | *snack* | 10 |
| C | *beer* | 5 |
| D | *soda* | 8 |
| E | *soda* | 6 |

Product description

because the formula $freq(patt, D)$ requires a relation $D$ and it is not allowed in a selection (see Section 2.1). Besides, we desire to make the computation of frequency explicit. The next section explains how to compute it with the relational algebra.

### 3.2 Cover, Semi-cover and Anti-cover Operators

We now indicate how to formulate the frequent pattern mining query (fpm query in brief) in the relational algebra which illustrates the need of the cover-like operators. Assume that $L[patt]$ and $D[trans]$ are two relations that respectively contain the language and the dataset as proposed in Table 1. The main challenge is to compute the frequency of each pattern of $L$. The Cartesian product of $L$ by $D$ gathers all the patterns of $L$ with all the transactions of $D$. Of course, we only select the relevant tuples such that the pattern covers the transaction: $\sigma_{patt \lhd trans}(L \times D)$. Finally, we count for each pattern how many transactions it covers and we select the frequent ones: $\sigma_{freq \geq s}(\gamma_{patt, \texttt{COUNT}(trans) \rightarrow freq}(\sigma_{patt \lhd trans}(L \times D)))$. As the notion of cover relation plays a central role to relate pattern-oriented attributes, we introduce three operators based on this notion. The cover operator for the pattern discovery is as important as the join operator for classical data manipulations.

*Cover operator.* The result of a cover operation gathers all the combinations of tuples in $R$ and $S$ that have comparable pattern-oriented attributes.

**Definition 2 (Cover operation).** *The cover of a relation $R[U]$ for a relation $S[V]$ w.r.t. a cover relation*[1] $\lhd \subseteq \mathbf{dom}(\widetilde{U}) \times \mathbf{dom}(\widetilde{V})$ *is* $R \lhd S = \sigma_{\widetilde{U} \lhd \widetilde{V}}(R \times S)$, *i.e. for any instances $I$ of $R$ and $J$ of $S$,* $I \lhd J = \{(t, u) | t \in I \wedge u \in J \wedge t[\widetilde{U}] \lhd u[\widetilde{V}]\}$.

As $\theta$-join is a shortcut of $\sigma_f(R \times S)$, the cover operator is derived from primitive operations defined in Section 2.1. In fact, $R \lhd S$ is equivalentl to $\sigma_{\widetilde{U} \lhd \widetilde{V}}(R \times S)$ where the formula $\widetilde{U} \lhd \widetilde{V}$ can be expressed with usual relational operators as done above with $patt \preceq \varphi$. Then, as semi-cover and anti-cover defined below, the cover operator does not increase the expressive power of the relational algebra. However, such operators bring two main advantages. First, algebraic properties of cover-like operators can be formulated, in order to be used by a query optimizer (see Section 5). Second, specialized and efficient query evaluation methods for these operators could be developed.

Let us illustrate the cover operation on several examples of pattern manipulations. Given a dataset $D[trans]$ and a language $L[patt]$, the frequent patterns (with their frequency) correspond to the following query:

$$F = \sigma_{freq \geq f}(\gamma_{patt, \texttt{COUNT}(trans) \rightarrow freq}(L \lhd D))$$

This fpm query fulfills our modeling objective by explicitly and declaratively describing how the frequency is computed. Given the instances of $L$ and $D$

[1] Definitions 2 to 4 consider that the binary relation $\lhd$ is a cover relation w.r.t. the specialization relations $\preceq_{\widetilde{U}}$ and $\preceq_{\widetilde{V}}$ respectively defined on $\mathbf{dom}(\widetilde{U})$ and $\mathbf{dom}(\widetilde{V})$.

**Table 2.** Instances containing mined patterns of instance $D$

(a) Itemset language

$F$

| patt | freq |
|---|---|
| $\emptyset$ | 4 |
| A | 4 |
| B | 3 |
| C | 2 |
| D | 2 |
| AB | 3 |
| AC | 2 |
| AD | 2 |
| BC | 2 |
| ABC | 2 |

Frequent itemsets

$C$

| patt | freq |
|---|---|
| A | 4 |
| AB | 3 |
| AD | 2 |
| ABC | 2 |

Frequent closed itemsets

$M$

| patt | freq |
|---|---|
| AD | 2 |
| ABC | 2 |

Frequent maximal itemsets

(b) Sequence language

$F$

| patt | freq |
|---|---|
| $\emptyset$ | 4 |
| (A) | 3 |
| (B) | 4 |
| (C) | 3 |
| (D) | 2 |
| (AB) | 3 |
| (A)(C) | 2 |
| (B)(C) | 3 |
| (B)(D) | 2 |
| (C)(D) | 2 |
| (AB)(C) | 2 |
| (B)(C)(D) | 2 |

Frequent sequences

provided by Table 1 and $f = 2$, it exactly returns the instance of $F$ (see Table 2). In the fpm query, the relation $\triangleleft \subseteq \mathbf{dom}(patt) \times \mathbf{dom}(trans)$ is a cover relation w.r.t. $\preceq_{patt}$ and $\preceq_{trans}$ (e.g., the inclusion for itemsets [3] or sequences [4]).

As mentioned earlier, a specialization relation is a particular kind of cover relation. Thereby, it can be exactly used as a cover operator. For instance, starting from the frequent patterns $F$, the frequent closed patterns of $D$ [5] are computed as follows: $C = \pi_{patt,freq}(\sigma_{freq>max}(\gamma_{patt,freq,\texttt{MAX}(freq')\rightarrow max}(F \prec F')))$ (we recall that $F'$ renames the attributes *patt* and *freq* into *patt'* and *freq'*). Table 2 illustrates this query applied to a particular instance of $F$ in the case of itemsets. Furthermore, the query $\gamma_{patt,\texttt{MAX}(freq')\rightarrow freq}(L \preceq C')$ regenerates the instance $F$.

*Semi-cover operator.* The semi-cover operator returns all the tuples of a relation covering at least one tuple of the other relation:

**Definition 3 (Semi-cover operation).** *The semi-cover of a relation $R[U]$ for a relation $S[V]$ w.r.t. a cover relation $\triangleleft \subseteq \mathbf{dom}(\widetilde{U}) \times \mathbf{dom}(\widetilde{V})$ is $R \triangleleft_{\ltimes} S = \pi_U(R \triangleleft S)$.*

Definition 3 implicitly means that $R \triangleright_{\ltimes} S$ returns all the tuples of $R$ covered by at least one tuple of $S$. Indeed, $R \triangleright_{\ltimes} S$ has a sense because if the binary relation $\triangleleft$ is a cover relation on $\mathbf{dom}(\widetilde{U}) \times \mathbf{dom}(\widetilde{V})$ w.r.t. $\preceq_{\widetilde{U}}$ and $\preceq_{\widetilde{V}}$, then $\triangleright$ is also a cover relation on $\mathbf{dom}(\widetilde{U}) \times \mathbf{dom}(\widetilde{V})$ w.r.t. $\succeq_{\widetilde{U}}$ and $\succeq_{\widetilde{V}}$. Table 3 illustrates Definition 3 by showing semi-cover operation of $L$ for $D$ which is the whole set of patterns occurring at least once in $D$: $L \triangleleft_{\ltimes} D$. Then, $\sigma_{patt \preceq \varphi}(L \triangleleft_{\ltimes} D)$ returns the patterns being more general than $\varphi$ and present in $D$.

Let us come back to the data and pattern covering operators [32] presented in Section 2.2. The operation $\theta_p(P, D)$ which gives the tuples of $P$ covering at least one tuple of $D$, is equivalent to $P \triangleleft_{\ltimes} D$. Dually, $\theta_d(P, D) = D \triangleright_{\ltimes} P$ returns the tuples of $D$ covered by at least one tuple of $P$.

*Anti-cover operator.* The anti-cover operator returns all the tuples of a relation not covering any tuple of the other relation:

**Table 3.** The semi-cover and anti-cover of $L$ for $D$

| $L \lhd_{\ltimes} D$ |
|---|
| patt |
| $\emptyset$ |
| A |
| B |
| C |
| D |
| E |
| AB |
| AC |
| AD |

| |
|---|
| AE |
| BC |
| BD |
| BE |
| CD |
| ABC |
| ABD |
| ABE |
| ACD |
| BCD |
| ABCD |

| $L \lhd_{\neg} D$ |
|---|
| patt |
| CE |
| DE |
| ACE |
| ADE |
| BCE |
| BDE |
| CDE |
| ABCE |
| ABDE |

| |
|---|
| ACDE |
| BCDE |
| ABCDE |

**Definition 4 (Anti-cover operation).** *The anti-cover of a relation $R[U]$ for a relation $S[V]$ w.r.t. a cover relation $\lhd \subseteq \mathbf{dom}(\widetilde{U}) \times \mathbf{dom}(\widetilde{V})$ is $R \lhd_{\neg} S = R - R \lhd_{\ltimes} S$.*

As for the semi-cover relation, $R \rhd_{\neg} S$ has a sense and returns all the tuples of $R$ not covered by any tuple of $S$. Table 3 gives the patterns of $L$ that do not occur in $D$ by means of the anti-cover of $L$ for $D$: $L \lhd_{\neg} D$. The anti-cover operator enables us to easily express the minimal and maximal pattern operators [25] (see Section 2.2): $\mathcal{M}in(R) = R \succ_{\neg} R$ and $\mathcal{M}ax(R) = R \prec_{\neg} R$. For instance, the frequent maximal itemsets are the frequent itemsets having no more specific frequent itemset: $M = F \prec_{\neg} F$ (see Table 2). A pattern of $L$ is either present in $D$ (i.e., in $L \lhd_{\ltimes} D$) or absent from $D$ (i.e., in $L \lhd_{\neg} D$). Then, we obtain that $L = L \lhd_{\ltimes} D \cup L \lhd_{\neg} D$ (see Table 3). More generally, the semi-cover and anti-cover operator are complementary by definition (see Definitions 3 and 4): $R = R \lhd_{\ltimes} S \cup R \lhd_{\neg} S$ for any relations $R$ and $S$.

### 3.3 Domain Operator

Let us come back to the query $\sigma_{freq \geq f}(\gamma_{patt,\texttt{COUNT}(trans) \rightarrow freq}(L \lhd D))$ that can be applied to any instance of relation $L$. However, in a practical pattern discovery task, the instance of $L$ has to gather all the existing patterns of $\mathbf{dom}(patt)$ (as given by Table 1). To cope with this problem, we introduce a new operator that outputs the domain of the schema for a given relation.

**Definition 5 (Domain operation).** *The domain of a relation $R[U]$ is $\Delta(R)$ where for any instance $I$ of $R$, $\Delta(I) = \mathbf{dom}(U)$.*

As the domain of each attribute is finite, the instance $\Delta(I)$ is finite. Assume that $I = \emptyset$ is an instance of $L[patt]$, $\Delta(I)$ returns the instance depicted by Table 1. The domain operator enables us to complete the frequent pattern mining query: $\sigma_{freq \geq f}(\gamma_{patt,\texttt{COUNT}(trans) \rightarrow freq}(\Delta(L) \lhd D))$. Other practical queries require the use of a language of patterns. For instance, negative border of $R$ [25] can now be formulated: $\mathcal{B}d^{-}(R) = (\Delta(R) - R) \succ_{\neg} (\Delta(R) - R)$. Similarly, the downward and upward closure operators of $R$ are respectively expressed by $\Delta(R) \preceq_{\ltimes} R$ and $\Delta(R) \succeq_{\ltimes} R$.

### 3.4 Scope of the Pattern-Oriented Relational Algebra

The *pattern-oriented relational algebra* which refers to the relational algebra plus the cover-like operators plus the domain operator, is strictly more expressive than the relational algebra. As aforementioned, the cover-like operators do not increase the expressive power of the relational algebra. In contrast, the domain operator cannot be expressed with relational operators because it induces domain dependent queries [1]. Let us note that [10] has already demonstrated that the frequent pattern mining query cannot be formulated in terms of the relational algebra.

From a practical point of view, the large number of query examples illustrating the previous sections (partially reported in Table 4 with $q_1$-$q_5$) highlights the generality of the pattern-oriented relational algebra. The other queries of Table 4 complete this overview by giving examples about the top-k frequent pattern mining with $q_6$ [15], the syntactic pattern mining $q_7$ [9], the utility-based pattern mining $q_8$ or the association rule mining $q_9$ [3]. Note that $\in$ is a cover relation on $\mathbf{dom}(item) \times \mathbf{dom}(patt)$ that relates one item with an itemset or a sequence. The query $q_7$ returns the patterns of $L$ occurring in $D$ and not containing a product of type 'snack'. $q_8$ returns the patterns of $L$ occurring in $D$ such that the sum of product prices is less than a threshold $t$.

**Table 4.** Examples of pattern-oriented queries and their properties

| | Pattern-oriented query | DC | Dependence: Local | Dependence: Global |
|---|---|---|---|---|
| $q_1$ | $\sigma_{freq \geq f}(\gamma_{patt,\texttt{COUNT}(trans) \to freq}(L \triangleleft D))$ | $L$ | $L$ | $D$ |
| $q_2$ | $\pi_{patt,freq}(\sigma_{freq > max}(\gamma_{patt,freq,\texttt{MAX}(freq') \to max}(F \prec F')))$ | | | $F$ |
| $q_3$ | $\sigma_{patt \prec \varphi}(L)$ | $L$ | $L$ | |
| $q_4$ | $\sigma_{patt \prec \varphi}(L \triangleleft_{\Join} D)$ | $L$ | $L$, $D$ | |
| $q_5$ | $F \prec_{\neg} F$ | | | $F$ |
| $q_6$ | $\sigma_{rank \leq k}(\gamma_{patt,freq,\texttt{COUNT}(patt') \to rank}(\sigma_{supp < supp'}(F \times F')))$ | $F$ | | $F$ |
| $q_7$ | $(L \triangleleft_{\Join} D) \ni_{\neg} \sigma_{type=snack}(P)$ | $L$ | $L$, $D$, $P$ | |
| $q_8$ | $\sigma_{total \leq t}(\gamma_{patt,\texttt{SUM}(price) \to total}(P \in (L \triangleleft_{\Join} D)))$ | $L$ | $L$, $D$, $P$ | |
| $q_9$ | $\pi_{patt' \to head, patt \setminus patt' \to body, freq, freq/freq' \to conf}(F' \prec F)$ | | | $F$ |

Most of these typical queries are difficult to evaluate because the handled instances may be very large especially when the domain operator is used for generating the language. The following sections explain how to rewrite queries for optimization purpose.

## 4 Characterizing Pattern-Oriented Queries

In the field of pattern mining, it is well known that some properties are useful to reduce the computation time (e.g., anti-monotone constraint or pre/post-processing ability). This section aims at characterizing such properties in the pattern-oriented relational algebra. More precisely, we first study the structuration of the instance resulting from a query w.r.t. the initial instance. Then, we analyze three levels of dependency between a query and a relation.

Thereafter we assume that $q$ is a query formulated with the pattern-oriented relational algebra and the database schema $\{R_1[U_1], \ldots, R_{n-1}[U_{n-1}], R[U]\}$. Then, this query $q$ is often applied to the database instance $\mathbf{I} = \{I_1, \ldots, I_{n-1}, I\}$.

### 4.1 Downward Closed Query

Intuitively, the notion of downward closed query expresses that of anti-monotone constraints [25] in the pattern-oriented relational algebra. A query $q$ is downward closed in $R$ if for any instance $I$ of $R[U]$, any tuple of $I$ more general than at least one tuple of $\pi_U(q(\mathbf{I}))$ also belongs to $\pi_U(q(\mathbf{I}))$.

**Definition 6 (Downward closed queries).** *A query $q$ is downward closed in $R[U]$ w.r.t. $\preceq$ iff $U \subseteq \mathbf{sch}(q)$ and $(R \preceq_{\ltimes} q) \equiv \pi_U(q)$.*

Definition 6 means that if a tuple $t$ of $R$ is more general than at least one tuple of the answer of $q$, $t$ is also present in this answer. The downward closed property is very interesting for pruning an instance (more details are given in Section 5.2). The query $\sigma_{freq \geq f}(\gamma_{patt,\texttt{COUNT}(trans) \rightarrow freq}(L \triangleleft D))$ is downward closed in $L$ w.r.t. $\preceq$. Indeed, all the generalizations of a frequent pattern are frequent (e.g., $ABC$ is frequent and then, $A$, $B$, $C$, $AB$ and so on are also frequent, see Table 1). Similarly, the top-k frequent pattern query $q_6$ is also downward closed in $F$ w.r.t. $\preceq$. The column 'DC' of Table 4 indicates the relations in which the query is downward closed w.r.t. $\preceq$.

### 4.2 Local and Global Dependent Queries

A query is dependent on the relation $R$ whenever its result varies with the instance of $R$. Whereas the query $\sigma_{patt \preceq \varphi}(L)$ is independent of $D$, $\sigma_{patt \preceq \varphi}(L \triangleleft_{\ltimes} D)$ depends on $D$ because it only returns the tuples of $\sigma_{patt \preceq \varphi}(L)$ that cover at least one tuple of the instance of $D$. Definition 7 formalizes the notion of total independence (or independence in brief):

**Definition 7 (Total independence).** *A query $q$ is totally independent of $R$ iff for any instances $I$, $J$ of $R$, one has $q(\{I_1, \ldots, I_{n-1}, I\}) = q(\{I_1, \ldots, I_{n-1}, J\})$.*

In other words, a query which is independent of $R$ is equivalent to another query not involving $R$. Note that the queries which are totally independent of $D$ correspond to syntactical constraints [9].

We now refine this notion of dependence by introducing the *global independence*. Both queries $\sigma_{patt \preceq \varphi}(L \triangleleft_{\ltimes} D)$ and $\sigma_{freq \geq f}(\gamma_{patt,\texttt{COUNT}(trans) \rightarrow freq}(L \triangleleft D))$ are dependent on $D$. But, the dependency of the second query on $D$ is stronger than that of the first query. Indeed, the computation of the frequency for a tuple of $L$ requires to simultaneously take into account several tuples of $D$.

**Definition 8 (Local/global dependence).** *A query $q$ is globally independent of $R$ iff for any instances $I$, $J$ of $R$, one has $q(\{I_1, \ldots, I_{n-1}, I \cup J\}) = q(\{I_1, \ldots, I_{n-1}, I\}) \cup q(\{I_1, \ldots, I_{n-1}, J\})$. A query being globally independent of $R$ but dependent on $R$ is said to be locally dependent on $R$.*

Definition 8 formalizes the notion of global constraints [12] which compare several patterns together to check whether the constraint is satisfied or not. The queries (like $q_2$, $q_5$, $q_6$ or $q_9$) which are globally dependent on $L$ or $F$ correspond to such global constraints. Besides, the query $q_1$ globally depends on $D$ and locally depends on $L$. It means that $q_1$ can be evaluated by considering separately each tuple of the instance of $L$. Conversely, it is impossible to consider individually each tuple of the instance of $D$. Thus, the higher the overall number of global dependencies, the harder the evaluation of the query. The columns 'Local' and 'Global' of Table 4 indicates the local/global dependent relations for each query. As expected, the queries $q_1$, $q_4$, $q_7$ and $q_8$ depend on $D$ because they benefit from the dataset to select the right patterns. We also observe that the queries $q_2$, $q_5$, $q_6$ and $q_9$ globally depend on $F$ as they postprocess the frequent patterns by comparing them.

## 5 Rewriting Pattern-Oriented Queries

This section examines algebraic equivalences to rewrite queries into forms that may be implemented more efficiently.

### 5.1 Algebraic Laws Involving Cover-Like Operators

Let us consider the query $q_4$: $\sigma_{patt \preceq \varphi}(L \triangleleft_{\ltimes} D)$. As the predicate $patt \preceq \varphi$ is highly selective, it is preferable to first apply it for reducing the language. Thereby, the equivalent query $\sigma_{patt \preceq \varphi}(L) \triangleleft_{\ltimes} D$ may be more efficient than $\sigma_{patt \preceq \varphi}(L \triangleleft_{\ltimes} D)$. The property below enumerates equivalences:

**Property 1 (Laws involving cover-like operators).** *Let $R[U]$ and $S[V]$ be two relation schemas. Let $f$ and $g$ be two predicates respectively on $R$ and $S$. Let $A$ and $B$ be two sets of attributes such that $\widetilde{U} \subseteq A \subseteq U$ and $\widetilde{V} \subseteq B \subseteq V$. One has the following equivalences:*

| | |
|---|---|
| *1.* $\sigma_{f \wedge g}(R \triangleleft S) \equiv \sigma_f(R) \triangleleft \sigma_g(S)$ | $\pi_{A \cup B}(R \triangleleft S) \equiv \pi_A(R) \triangleleft \pi_B(S)$ |
| *2.* $\sigma_f(R \triangleleft_{\ltimes} S) \equiv \sigma_f(R) \triangleleft_{\ltimes} S$ | $\pi_A(R \triangleleft_{\ltimes} S) \equiv \pi_A(R) \triangleleft_{\ltimes} S$ |
| *3.* $\sigma_f(R \triangleleft_{\neg} S) \equiv \sigma_f(R) \triangleleft_{\neg} S$ | $\pi_A(R \triangleleft_{\neg} S) \equiv \pi_A(R) \triangleleft_{\neg} S$ |
| *4.* $R \triangleleft_{\ltimes} S \equiv R \triangleleft_{\ltimes} (S \prec_{\neg} S)$ | $R \triangleleft_{\neg} S \equiv R \triangleleft_{\neg} (S \prec_{\neg} S)$ |

Intuitively, the right hand side of each equivalence listed in Property 1 (proofs are omitted due to lack of space) may lead to optimize the query. Indeed, Lines 1 to 3 "pushes down" the selection and projection operators to reduce the size of the operands before applying a cover-like operator. This technique is successfully exploited in database with Cartesian product or join operator [1]. Besides, Line 4 benefits from the maximal tuples of $S$ (i.e., $S \prec_{\neg} S$) as done in pattern mining [25]. If a tuple $t$ of the instance of $R$ covers a tuple of the instance $J$ of

$S$, then $t$ also covers a tuple of $J \prec_\neg J$. As $|J \prec_\neg J| \leq |J|$, the rewritten query $R \triangleleft_\ltimes (S \prec_\neg S)$ may be less costly than $R \triangleleft_\ltimes S$ provided $J \prec_\neg J$ is not too costly.

### 5.2 Algebraic Reformulation of the Levelwise Algorithm

We now take into account the downward closed and the global independence properties for reformulating queries. For instance, assume that the instance of $L$ is now equal to $\pi_{patt}(F)$. A new computation of $q_1$ again returns $F$: $F = \sigma_{freq \geq 2}(\gamma_{patt,\texttt{COUNT}(trans) \rightarrow freq}(\pi_{patt}(F) \triangleleft D))$. Of course, this query is faster to compute than the original fpm query because the instance of $F$ is very small compared to $\Delta(L)$. We generalize this observation:

**Property 2.** *Let $q$ be a downward closed query in $R[U]$ w.r.t. $\preceq$ and globally independent of $R$ such that $U \subseteq \mathbf{sch}(q)$, one has $q(\mathbf{I}) = q(\mathbf{J})$ for any instances $\mathbf{I} = \{I_1, \ldots, I_{n-1}, I\}$ and $\mathbf{J} = \{I_1, \ldots, I_{n-1}, J\}$ such that $\pi_U(q(\mathbf{J})) \subseteq I \subseteq J$.*

Given a downward closed and independent query $q$, Property 2 demonstrates that $q(\mathbf{I}) = q(\mathbf{J})$ when $I$ is an instance of $R$ such that $\pi_U(q(\mathbf{J})) \subseteq I \subseteq J$. As $I \subseteq J$ and then $|I| \leq |J|$, we suppose that evaluating $q(\mathbf{I})$ is less costly than evaluating $q(\mathbf{J})$ because the cost generally decreases with the cardinality of the instance. Thus, in order to reduce the cost of the evaluation of $q(\mathbf{I})$, we aim at turning $I$ into the smallest instance of $R$ including $q(\mathbf{J})$. Such an approach can be seen as a *pruning* of the instance of $R$.

**Table 5.** Levelwise computation of the fpm query (level 2)

$L$

| patt |
|---|
| AB |
| AC |
| AD |
| BC |
| BD |
| CD |

| ABC |
|---|
| ABD |
| ACD |
| BCD |
| ABCD |

$\mathcal{C} = L \succ_\neg L$

| patt |
|---|
| AB |
| AC |
| AD |
| BC |
| BD |
| CD |

$\mathcal{S}$

| patt | supp |
|---|---|
| AB | 3 |
| AC | 2 |
| AD | 2 |
| BC | 2 |

$L \succ_\ltimes \mathcal{S}$

| patt |
|---|
| ABC |
| ABD |
| ACD |
| BCD |
| ABCD |

$(L \succ_\ltimes \mathcal{S}) \succeq_\neg (\mathcal{C} \succeq_\neg \mathcal{S})$

| patt |
|---|
| ABC |

Table 5 illustrates how to prune the instance $L$ for evaluating the fpm query $q_1$. As $q_1$ is globally independent of $L$, we first divide $L$ into two parts: the most general tuples of $L$ denoted by $\mathcal{C} = L \succ_\neg L$ (i.e., the candidates of the level 2 of Apriori [3]) and others, i.e. $L \succ_\ltimes L$. We then apply the fpm query to $\mathcal{C}$ for computing $\mathcal{S}$: the frequent patterns of $\mathcal{C}$ and their frequency. Finally, we benefit from $\mathcal{S}$ for pruning $L \succ_\ltimes L$ using the downward closed property of $q_1$ in $L$ w.r.t. $\preceq$ (see Definition 6). We only preserve the tuples which are more specific than at least one frequent tuple of $\mathcal{S}$: $L \succ_\ltimes \mathcal{S}$. Finally, we filter out the tuples having a non-frequent generalization: $(L \succ_\ltimes \mathcal{S}) \succeq_\neg (\mathcal{C} \succeq_\neg \mathcal{S})$. As the cardinality of this instance is smaller than $|L \succ_\ltimes L|$, we have achieved our goal.

This principle is generalized with this theorem:

**Theorem 1 (Levelwise equivalence).** *Let $q$ be a downward closed query w.r.t. $\preceq$ and globally independent of $R$, one has the below equality for any database instance $\mathbf{I} = \{I_1, \ldots, I_{n-1}, I\}$:*

$$q(\mathbf{I}) = \underbrace{q(\{I_1, \ldots, I_{n-1}, \underbrace{I \succ_\neg I}_{\mathcal{C}=}\}) \cup q(\{I_1, \ldots, I_{n-1}, (I \succ_\ltimes \mathcal{S}) \succeq_\neg (\mathcal{C} \succeq_\neg \mathcal{S})\})}_{\mathcal{S}=}$$

*Proof.* Let $q$ be a downward closed query w.r.t. $\preceq$ and globally independent of $R$. To alleviate the notations, $q(I)$ refers to $q(\{I_1, \ldots, I_{n-1}, I\})$ where $I$ is any instance of $R$. Besides, we fix that $\mathcal{C} = I \succ_\neg I$ and $\mathcal{S} = q(I \succ_\neg I) = q(\mathcal{C})$:

$$\begin{aligned}
q(I) &= q(I \succ_\neg I \cup I \succ_\ltimes I) = q(\mathcal{C} \cup I \succ_\ltimes I) && (1)\\
&= q(\mathcal{C}) \cup q(I \succ_\ltimes I) && (2)\\
&= q(\mathcal{C}) \cup q(I \succ_\ltimes q(\mathcal{C})) = q(\mathcal{C}) \cup q(I \succ_\ltimes \mathcal{S}) && (3)\\
&= q(\mathcal{C}) \cup q((I \succ_\ltimes \mathcal{S}) \succeq_\neg (\mathcal{C} \succeq_\neg \mathcal{S})) && (4)
\end{aligned}$$

Line 1 stems from the complementary property: $R = R \triangleleft_\ltimes S \cup R \triangleleft_\neg S$. Line 2 is allowed because $q$ is globally independent of $R$. Line 3-4 are due to the downward closed property in $R$ (see Definition 6). □

Theorem 1 can be used for rewriting queries by considering two important points. Firstly, the redundant subqueries as candidate tuples $\mathcal{C} = I \succ_\neg I$ and satisfied tuples $\mathcal{S} = q(\{I_1, \ldots, I_{n-1}, I \succ_\neg I\})$ have to be evaluated only once. Secondly, the practical evaluation of $q$ requires to recursively apply the equality proposed in Theorem 1. Indeed, the subquery $q(\{I_1, \ldots, I_{n-1}, (I \succ_\ltimes \mathcal{S}) \succeq_\neg (\mathcal{C} \succeq_\neg \mathcal{S})\})$ can also be rewritten by a query plan optimizer using the same identity. Therefore, Theorem 1 leads to algebraically reformulate the levelwise algorithm [3, 4, 25]. This algorithm repeats this equality for computing which candidate patterns satisfy the predicate and then, generating those of the next level. Other efficient pruning strategies like depth-first search techniques [5] could also be expressed in pattern-oriented relational algebra. Finally, as observed in [12, 15], we cannot apply Theorem 1 to $q_6$ because it globally depends on $F$.

## 6 Related Work

Inductive databases [18, 24] aims at tightly integrating databases with data mining. Our approach is less ambitious because it is "only" restricted to the pattern mining. Obviously, many proposals provide an environment merging a RDBMS with pattern mining tools: Quest [2], ConQueSt [7], DBminer [16], Sindbad [34] and many other prototypes [6]. In such a context, there are many extensions of the SQL language [31] like DMX or MINERULE [26]. There are also extended relational model [13] like 3W model [20]. However, such methods don't fuse the theoretical concepts stemming from both the relational model and the pattern

discovery. For instance, the query optimizer of DBMS is isolated from pattern mining algorithms. Indeed, most of the approaches consider a pattern mining query as the result of a "black box". Only few works [10, 23, 33] express pattern mining operators by benefiting from the relational algebra. Such approaches add a loop statement for implementing the levelwise algorithm. On the contrary, our proposal extends the relational algebra by still using a declarative approach.

Many frameworks inspired from relational and logical databases, but created from scratch, are proposed during the last decade: constraint-based pattern mining [9, 25], distance-based framework [14], rule-base [19], tuple relational calculus [28], logical database [29], pattern-base [32] and so on. Other directions are suggested in [24] like probabilistic approach or data compression. Besides, constraint programming is another promising way for expressing and mining patterns [21, 30]. Such frameworks are less convenient for handling data (which are often initially stored in relational databases). Besides, they suffer from a lack of simple and powerful languages like the relational algebra (in particular, the manipulation of patterns is frequently separated from that of data).

From a more general point of view, many works add new operators to the relational algebra in order to express more sophisticated queries. Even if such new operators don't necessary increase the expressive power of the relational algebra, most of the time they facilitate the formulation of user queries and provide specific optimizations. Typically, several operators are introduced for comparing tuples with each other, as does a specialization relation with patterns. For instance, the winnow operator is specifically dedicated to handle preferences [11]. Several operators are dedicated for selecting the best tuples by means of relational dominant queries [8] or relational top-k queries [22]. The cover-like operators are very closed to such operators. But, they enable to compare tuples based on different languages, as does a cover relation with patterns. Finally, the domain operator enables us to manipulate values not initially present in the relations. The same concept is used in [13] for generating tables containing patterns.

## 7 Conclusion

In this paper, we have proposed a new and general framework for pattern discovery by only adding cover-like and domain operators to the relational algebra. The pattern-oriented relational algebra interestingly inherits good properties from the relational algebra as closure or declarativity. This framework deals with any language of patterns for expressing a wide spectrum of queries including constraint-based pattern mining, condensed representations and so on. We identify crucial aspects of queries as the downward closed and independence properties. We then benefit from such properties to algebraically reformulate the levelwise algorithm. We think that our algebraisation is an important step towards the elegant integration of pattern discovery in database systems.

Further work addresses the implementation of a complete system based on the pattern-oriented relational algebra. As done in the database field, we project to implement the physical cover operators and to design a query plan optimizer

taking advantage of our proposed algebraic laws. We also study the test of local and global dependence between a query and a relation.

## References

1. Abiteboul, S., Hull, R., Vianu, V.: Foundations of Databases. Addison-Wesley, Reading (1995)
2. Agrawal, R., Mehta, M., Shafer, J.C., Srikant, R., Arning, A., Bollinger, T.: The quest data mining system. In: KDD, pp. 244–249 (1996)
3. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: Bocca, J.B., Jarke, M., Zaniolo, C. (eds.) VLDB, pp. 487–499. Morgan Kaufmann, San Francisco (1994)
4. Agrawal, R., Srikant, R.: Mining sequential patterns. In: Yu, P.S., Chen, A.L.P. (eds.) ICDE, pp. 3–14. IEEE Computer Society, Los Alamitos (1995)
5. Arimura, H., Uno, T.: Polynomial-delay and polynomial-space algorithms for mining closed sequences, graphs, and pictures in accessible set systems. In: SDM, pp. 1087–1098. SIAM, Philadelphia (2009)
6. Blockeel, H., Calders, T., Fromont, É., Goethals, B., Prado, A., Robardet, C.: An inductive database prototype based on virtual mining views. In: KDD, pp. 1061–1064. ACM, New York (2008)
7. Bonchi, F., Giannotti, F., Lucchese, C., Orlando, S., Perego, R., Trasarti, R.: ConQueSt: a constraint-based querying system for exploratory pattern discovery. In: ICDE, p. 159. IEEE Computer Society, Los Alamitos (2006)
8. Börzsönyi, S., Kossmann, D., Stocker, K.: The skyline operator. In: ICDE, pp. 421–430. IEEE Computer Society, Los Alamitos (2001)
9. Boulicaut, J.F., Jeudy, B.: Constraint-based data mining. In: Maimon, O., Rokach, L. (eds.) The Data Mining and Knowledge Discovery Handbook, pp. 399–416. Springer, Heidelberg (2005)
10. Calders, T., Lakshmanan, L.V.S., Ng, R.T., Paredaens, J.: Expressive power of an algebra for data mining. ACM Trans. Database Syst. 31(4), 1169–1214 (2006)
11. Chomicki, J.: Querying with intrinsic preferences. In: Jensen, C.S., Jeffery, K., Pokorný, J., Šaltenis, S., Hwang, J., Böhm, K., Jarke, M. (eds.) EDBT 2002. LNCS, vol. 2287, pp. 34–51. Springer, Heidelberg (2002)
12. Crémilleux, B., Soulet, A.: Discovering knowledge from local patterns with global constraints. In: Gervasi, O., Murgante, B., Laganà, A., Taniar, D., Mun, Y., Gavrilova, M.L. (eds.) ICCSA 2008, Part II. LNCS, vol. 5073, pp. 1242–1257. Springer, Heidelberg (2008)
13. Diop, C.T., Giacometti, A., Laurent, D., Spyratos, N.: Composition of mining contexts for efficient extraction of association rules. In: Jensen, C.S., Jeffery, K., Pokorný, J., Šaltenis, S., Hwang, J., Böhm, K., Jarke, M. (eds.) EDBT 2002. LNCS, vol. 2287, pp. 106–123. Springer, Heidelberg (2002)
14. Dzeroski, S.: Towards a general framework for data mining. In: Džeroski, S., Struyf, J. (eds.) KDID 2006. LNCS, vol. 4747, pp. 259–300. Springer, Heidelberg (2007)
15. Fu, A.W.C., Kwong, R.W., Tang, J.: Mining $n$-most interesting itemsets. In: Ohsuga, S., Raś, Z.W. (eds.) ISMIS 2000. LNCS (LNAI), vol. 1932, pp. 59–67. Springer, Heidelberg (2000)
16. Han, J., Fu, Y., Wang, W., Chiang, J., Gong, W., Koperski, K., Li, D., Lu, Y., Rajan, A., Stefanovic, N., Xia, B., Zaïane, O.R.: DBMiner: a system for mining knowledge in large relational databases. In: KDD, pp. 250–255 (1996)

17. Hand, D.J.: Pattern detection and discovery. In: Hand, D.J., Adams, N.M., Bolton, R.J. (eds.) Pattern Detection and Discovery. LNCS (LNAI), vol. 2447, pp. 1–12. Springer, Heidelberg (2002)
18. Imielinski, T., Mannila, H.: A database perspective on knowledge discovery. Commun. ACM 39(11), 58–64 (1996)
19. Imielinski, T., Virmani, A.: MSQL: a query language for database mining. Data Min. Knowl. Discov. 3(4), 373–408 (1999)
20. Johnson, T., Lakshmanan, L.V.S., Ng, R.T.: The 3W model and algebra for unified data mining. In: Abbadi, A.E., Brodie, M.L., Chakravarthy, S., Dayal, U., Kamel, N., Schlageter, G., Whang, K.Y. (eds.) VLDB, pp. 21–32. Morgan Kaufmann, San Francisco (2000)
21. Khiari, M., Boizumault, P., Crémilleux, B.: Combining CSP and constraint-based mining for pattern discovery. In: Taniar, D., Gervasi, O., Murgante, B., Pardede, E., Apduhan, B.O. (eds.) ICCSA 2010. LNCS, vol. 6017, pp. 432–447. Springer, Heidelberg (2010)
22. Li, C., Chang, K.C.C., Ilyas, I.F., Song, S.: RankSQL: query algebra and optimization for relational top-k queries. In: Özcan, F. (ed.) SIGMOD Conference, pp. 131–142. ACM Press, New York (2005)
23. Liu, H.C., Ghose, A., Zeleznikow, J.: Towards an algebraic framework for querying inductive databases. In: Kitagawa, H., Ishikawa, Y., Li, Q., Watanabe, C. (eds.) DASFAA 2010. LNCS, vol. 5982, pp. 306–312. Springer, Heidelberg (2010)
24. Mannila, H.: Theoretical frameworks for data mining. SIGKDD Explorations 1(2), 30–32 (2000)
25. Mannila, H., Toivonen, H.: Levelwise search and borders of theories in knowledge discovery. Data Min. Knowl. Discov. 1(3), 241–258 (1997)
26. Meo, R., Psaila, G., Ceri, S.: A new SQL-like operator for mining association rules. In: Vijayaraman, T.M., Buchmann, A.P., Mohan, C., Sarda, N.L. (eds.) VLDB, pp. 122–133. Morgan Kaufmann, San Francisco (1996)
27. Mitchell, T.M.: Generalization as search. Artif. Intell. 18(2), 203–226 (1982)
28. Nijssen, S., Raedt, L.D.: IQL: a proposal for an inductive query language. In: Džeroski, S., Struyf, J. (eds.) KDID 2006. LNCS, vol. 4747, pp. 189–207. Springer, Heidelberg (2007)
29. Raedt, L.D.: A logical database mining query language. In: Cussens, J., Frisch, A.M. (eds.) ILP 2000. LNCS (LNAI), vol. 1866, pp. 78–92. Springer, Heidelberg (2000)
30. Raedt, L.D., Guns, T., Nijssen, S.: Constraint programming for itemset mining. In: KDD, pp. 204–212. ACM, New York (2008)
31. Romei, A., Turini, F.: Inductive database languages: requirements and examples. Knowledge and Information Systems 1–34 (2010), http://dx.doi.org/10.1007/s10115-009-0281-4
32. Terrovitis, M., Vassiliadis, P., Skiadopoulos, S., Bertino, E., Catania, B., Maddalena, A., Rizzi, S.: Modeling and language support for the management of pattern-bases. Data Knowl. Eng. 62(2), 368–397 (2007)
33. Wang, H., Zaniolo, C.: ATLaS: a native extension of SQL for data mining. In: Barbará, D., Kamath, C. (eds.) SDM. SIAM, Philadelphia (2003)
34. Wicker, J., Richter, L., Kessler, K., Kramer, S.: SINDBAD and SiQL: an inductive database and query language in the relational model. In: Daelemans, W., Goethals, B., Morik, K. (eds.) ECML PKDD 2008, Part II. LNCS (LNAI), vol. 5212, pp. 690–694. Springer, Heidelberg (2008)

# Mining Dominant Patterns in the Sky

Arnaud Soulet* , Chedy Raïssi†, Marc Plantevit ‡ and Bruno Crémilleux§
* *Université François Rabelais de Tours, LI, EA 2101, F-41029, France*
†*INRIA Nancy Grand-Est, France*
‡*Université de Lyon, CNRS, Université Lyon 1, LIRIS, UMR5205, F-69622, France*
§*Université de Caen Basse-Normandie, CNRS, GREYC UMR6072, F-14032, France*

***Abstract*—Pattern discovery is at the core of numerous data mining tasks. Although many methods focus on efficiency in pattern mining, they still suffer from the problem of choosing a threshold that influences the final extraction result. The goal of our study is to make the results of pattern mining useful from a user-preference point of view. To this end, we integrate into the pattern discovery process the idea of skyline queries in order to mine *skyline patterns* in a threshold-free manner. Because the skyline patterns satisfy a formal property of dominations, they not only have a global interest but also have semantics that are easily understood by the user. In this work, we first establish theoretical relationships between pattern condensed representations and skyline pattern mining. We also show that it is possible to compute automatically a subset of measures involved in the user query which allows the patterns to be condensed and thus facilitates the computation of the skyline patterns. This forms the basis for a novel approach to mining skyline patterns. We illustrate the efficiency of our approach over several data sets including a use case from chemoinformatics and show that small sets of dominant patterns are produced under various measures.**

***Keywords*-Skyline analysis, Pattern mining, user-preferences.**

## I. Introduction

The process of extracting useful patterns from data, called *pattern mining*, is an important tool for data analysis and has been used in a wide range of applications and domains such as bioinformatics [1] or chemoinformatics [2]. Since the pioneering works of Agrawal *et al.* [3], Mannila *et al*. [4], a large amount of work has been developed and many pattern extraction problems are now identified and understood from both theoretical and computational perspectives.

Most existing pattern mining approaches enumerate patterns with respect to a given set of constraints that range from extremely simple to very complex. For instance, given a transaction database, a well-known *"easy"* pattern mining task is to enumerate all itemsets (i.e., sets of items) that appear in at least $s$ transactions. Another mining approach is to extract from a given graph database all subgraphs that have a diameter larger than $l$, connectivity higher than $c$, and where each vertex has a degree bounded by $d$. So far, the community has made great efforts on sophisticated algorithms pushing the constraints deep into the mining process [5]. But it has paid less attention to how to define constraints. In practice, many constraints entail choosing of threshold values such as the well-used minimal frequency. This notion of *"thresholding"* has serious drawbacks. Unless specific domain knowledge is available, the choice is often arbitrary and may lead to a very large number of extracted patterns which can reduce the success of any subsequent data analysis. This drawback is obviously even deeper when several thresholds are needed and have to be combined. A second drawback is the *stringent enumeration aspect*: a pattern is either above or below the thresholds. What about patterns that respect only some thresholds? With this paradigm it is very difficult to apply *subtle selection* mechanisms. There are very few works such as [6] which propose to introduce a softness criterion into the mining process. Other studies blend user preferences in the mining task in order to limit the number of extracted patterns such as the *top-k* patterns [7], [8]. By associating each pattern with a *rank score*, this approach returns an ordered list of the $k$ patterns with the highest score to the user. However, combining several measures to be reflected in a single scoring function is difficult and the performance of top-k approaches are often sensitive to the size of the datasets and to the threshold value, $k$.

In this work, we focus on making the results of pattern mining *useful from a user-preference point of view*. To this end, we integrate into the pattern discovery process the idea of skyline queries [9] in order to mine *skyline patterns* in a threshold-free manner. Such queries have attracted considerable attention due to their importance in multi-criteria decision making. Briefly speaking, in a multidimensional space where a preference is defined for each dimension, a point $a$ dominates another point $b$ if $a$ is better (i.e., more preferred) than $b$ in at least one dimension, and $a$ is not worse than $b$ on every other dimension. For example, a user selecting a set of patterns may prefer a pattern with a low frequency, short length and a high confidence. In this case, we say that pattern $a$ *dominates* another pattern $b$ if $a.frequency \leq b.frequency$, $a.length \leq b.length$, $a.confidence \geq b.confidence$, where at least one strict inequality holds. Given a set of patterns, the skyline set contains the patterns that are not dominated by any other patterns.

We claim that skyline pattern mining is interesting for several reasons: first, skyline processing does not require any threshold selection or ranking function. Second, the

formal property of domination satisfied by the skyline patterns gives to the patterns a global interest with semantics easily understood by the user. However, while this notion of skylines has been extensively developed and researched for database applications, it has remained unused for data mining purposes except for a single work on extracting skyline graphs that maximize two measures: the number of vertices and the edge connectivity [10].

Mining skyline patterns, or skypatterns, can be done in a brute-force manner: i.e., mine all patterns in a first step, then run domination tests with respect to the user preferences and finally output the skyline patterns. However, this naive approach is not feasible in practice as the collection of patterns is often too big to be manageable. Obviously, constraints might be introduced to limit the size of the collection but the consistency of the result may be lost (i.e., some skypatterns may not be produced) and the thresholding problem would remain. A key idea of our work is to take benefit of theoretical relationships between pattern condensed representations and skypatterns. These results improve skypattern extraction and we propose, as a main contribution, an efficient approach which only takes as an input the data set and the measures expressing the user preferences and returns skypatterns. To the best of our knowledge, this is *the first work to study theoretically and empirically the feasibility of the skyline pattern mining in a fully generic way* (i.e., with application to various types of patterns).

The paper is organized as follows. Section II reviews some related work. Section III introduces basic definitions and a formal problem statement. The generic framework of skypattern queries is detailed in Section IV. We report an experimental study on several datasets and a case study from the chemoinformatics domain in section V. We conclude in Section VI.

## II. Related work

The notion of dominance that we introduced above (see Section III for a formal definition) is at the core of the skyline processing. In this paradigm, the retrieved data points are the ones that are not dominated by any other point in the analysis space. These skyline points can be viewed as *compromise points* with respect to a given set of criteria. Skyline computation is strongly related to mathematical and microeconomics problems such as maximum vectors [11], Pareto set [12] and multi-objective optimization [13]. Since its rediscovery within the database community by Börzsönyi *et al.* [9], many methods have been developed for answering skyline queries that can handle various constraints in different computational environments. Another aspect of preference-based processing is the *top-k* procedure [7], [8]. A ranking function $f_r$ is applied to patterns, and the $k$ best patterns with the highest score with respect to $f_r$ are returned. As previously mentioned, this approach suffers from limitations. The choice of $k$ is not trivial (i.e., the *horizon* problem): a low value may miss useful patterns and a too high value introduces redundancy within the produced patterns (i.e., highly similar patterns). This limitation is the main motivation for the most informative patterns (MIP) that have been recently proposed in [14]. MIPs can be seen as patterns that *locally dominate* other patterns according to a scoring function. This approach shares a similar spirit to our work as it also limits the number of enumerated patterns to a more manageable level. However, in contrast to our study, work on MIPs includes a notion of dominance that is only *local and specific* to subsets of patterns.

One of the earliest findings in the data mining community is that a mining process usually produces large collections of patterns. Many researchers have proposed methods to reduce the size of the output: the constraint-based pattern mining framework [15], the condensed representations [16] and the compression of the dataset by exploiting Minimum Description Length Principle [17], to name a few. A general observation is that patterns represent fragmented knowledge, and often there is no clear view of how the pieces of the puzzle interact and combine to produce a global model. Recent approaches have therefore used schemes such as pattern teams [18], constraint-based pattern set mining [19] and pattern selections [20] that aim to minimize the redundancy and the number of patterns. The common theme in these studies is to select patterns from the initial large set of patterns on the basis of their usefulness in a given context. Often, these methods focus on optimizing a global measure on the discovered pattern set and neglect the relationships between patterns. Moreover, these approaches suffer from a lack of flexibility to express the queries requested by the analyst. For each method, the user has to understand its semantics and express queries satisfying its algorithmic properties and constraints. In addition, some studies take advantage of closed patterns (according to the support measure) to maximize a specific measure such as growth rate for emerging patterns [21] and area for tiling [22], [23].

## III. Problem Formulation and Preliminary Definitions

Our study is interesting for several reasons. Firstly, by carefully selecting patterns that are *"the best available"* for a given set of preferences we reduce significantly the output and limit the *"pattern explosion"* curse. The user is *guaranteed* that only the most significant patterns are present in the final result based on his criteria. Secondly, our approach is *parameter-free*. No thresholds are required (solely optional, depending on the analyst needs), and only the preferences and the data set are given as an input.

### A. Preliminary definitions

Although the problem can be formulated for any kind of pattern, for simplicity, we will illustrate our definitions using

Table I: Example of a toy data set and measures

(a) A toy data set $\mathcal{D}$

| Tid | Items |
|---|---|
| $t_1$ | $A$ $B$ $C$ $D$ $E$ $F$ |
| $t_2$ | $A$ $B$ $C$ $D$ $E$ $F$ |
| $t_3$ | $A$ $B$ |
| $t_4$ | $D$ |
| $t_5$ | $A$ $C$ |
| $t_6$ | $E$ |

| Items | $val$ |
|---|---|
| $A$ | 10 |
| $B$ | 55 |
| $C$ | 70 |
| $D$ | 30 |
| $E$ | 15 |
| $F$ | 25 |

(b) Some measures of $\mathcal{M}$

| $Name$ | Definition |
|---|---|
| $area$ | $X \mapsto freq(X) \times length(X)$ |
| $mean$ | $X \mapsto \frac{min(X.val)+max(X.val))}{2}$ |
| $bond$ | $X \mapsto \frac{freq(X)}{freq_\vee(X)}$ |
| $aconf$ | $X \mapsto \frac{freq(X)}{max(X.freq)}$ |
| $gr_1$ | $X \mapsto \frac{\lvert\mathcal{D}_2\rvert}{\lvert\mathcal{D}_1\rvert} \times \frac{freq(X,\mathcal{D}_1)}{freq(X,\mathcal{D}_2)}$ |

Table II: A subset of the primitive-based measures

| Measure $m \in \mathcal{M}$ | Primitive(s) | Operand(s) |
|---|---|---|
| $m_1 \theta m_2$ | $\theta \in \{+,-,\times,/\}$ | $(m_1, m_2) \in \mathcal{M}^2$ |
| $\theta(s)$ | $\theta \in \{freq, freq_\vee, length\}$ | $s \in \mathcal{S}$ |
| $\theta(s.val)$ | $\theta \in \{sum, max, min\}$ | $s \in \mathcal{S}$ |
| constant $r \in \Re^+$ | - | - |

| Syntactic expression $s \in \mathcal{S}$ | Primitive(s) | Operand(s) |
|---|---|---|
| $s_1 \theta s_2$ | $\theta \in \{\cup, \cap, \setminus\}$ | $(s_1, s_2) \in \mathcal{S}^2$ |
| $\theta(s)$ | $\theta \in \{f, g\}$ | $s \in \mathcal{S}$ |
| variable $X \in \mathcal{L}$ | - | - |
| constant $l \in \mathcal{L}$ | - | - |

itemset patterns. Section IV discusses the computational and theoretical aspects associated with the problem when extracting other patterns. Let $\mathcal{I}$ be a set of distinct literals called *items*, an itemset (or pattern) corresponds to a non-null subset of $\mathcal{I}$. These patterns are gathered together in the language $\mathcal{L}$: $\mathcal{L} = 2^{\mathcal{I}} \setminus \emptyset$. A transactional dataset is a multi-set of patterns of $\mathcal{L}$. Each pattern, named *transaction*, is a database entry. Table I(a) presents a transactional dataset $\mathcal{D}$ where 6 transactions denoted by $t_1, \ldots, t_6$ are described by 6 items denoted by $A, \ldots, F$.

All the measures discussed in this study are based on the set of *primitive-based measures* $\mathcal{M}$ that were first defined in the context of constraint-based pattern mining [24]. Table II presents general definitions of measures and Table I(b) gives some specific examples. As presented in [24], $\mathcal{M}$ defines a very large set of interesting measures.

In addition to the classical operators of $\Re^+$ and $\mathcal{L}$, the function $freq$ denotes the frequency of a pattern, and $length$ its cardinality. The disjunctive support is $freq_\vee(X) = |\{t \in \mathcal{D} | \exists i \in X : i \in t\}|$. Given a function $val : \mathcal{I} \to \Re^+$, we extend it to a pattern $X$ and note $X.val$ the multiset $\{val(i) | i \in X\}$. This kind of function is used with the usual SQL-like primitives $sum$, $min$ and $max$. For instance, $sum(X.val)$ is the sum of $val$ for each item of $X$. Finally, $f$ is the intensive function i.e. $f(T) = \{i \in \mathcal{I} | \forall t \in T,\ i \in t\}$, and $g$ is the extensive function i.e. $g(X) = \{t \in Tid | X \subseteq t\}$.

*Definition 1 (Domination):* Given a set of measures $M \subseteq \mathcal{M}$, a pattern $X$ *dominates* another pattern $Y$ with respect to $M$, denoted by $X \succ_M Y$, iff for any measure $m \in M$, $m(X) \geq m(Y)$ and there exists $m \in M$ such that $m(X) > m(Y)$. Two patterns $X$ and $Y$ are said to be *indistinct* with respect to $M$, denoted by $X =_M Y$, iff $m(X)$ equals to $m(Y)$ for any measure $m \in M$ (if $M = \emptyset$, then $X =_\emptyset Y$). Finally, $X \succeq_M Y$ denotes that $(X \succ_M Y) \vee (X =_M Y)$.

Consider our running example using the data set $\mathcal{D}$ in Table I and suppose that $M = \{freq, area\}$, then the pattern $ABCDEF$ dominates $ABC$ because $freq(ABC) = freq(ABCDEF) = 2$ and $area(ABCDEF) > area(ABC)$. Notice in this case that $ABCDEF$ is indistinct to $ABC$ with respect to $\{freq\}$. Similarly, suppose that $M = \{freq, mean, length\}$, the pattern $AC$ dominates $AB$ because $freq(AC) = freq(AB) = 3$, $|AB| = |AC| = 2$ and $mean(AC) > mean(AB)$.

### B. The skypattern mining problem

Given a set of measures $M$, if a pattern is dominated by another, according to all measures of $M$, it is irrelevant and must be discarded in the output. The notion of *skyline pattern* formalizes this intuition.

*Definition 2 (Skypattern operator):* Given a pattern set $P \subseteq \mathcal{L}$ and a set of measures $M \subseteq \mathcal{M}$, a skypattern of $P$ with respect to $M$ is a pattern not dominated in $P$ with respect to $M$. The skypattern operator $\mathcal{S}ky(P, M)$ returns all the skypatterns of $P$ with respect to $M$:

$$\mathcal{S}ky(P, M) = \{X \in P |\ \nexists Y \in P : Y \succ_M X\}$$

Given a set of measures $M \subseteq \mathcal{M}$, the *skypattern mining problem is thus to evaluate the query* $\mathcal{S}ky(\mathcal{L}, M)$. For instance, from the toy data set in Table I, $\mathcal{S}ky(\mathcal{L}, \{freq, length\}) = \{ABCDEF,\ AB, AC, A\}$, as illustrated in Fig. 1.

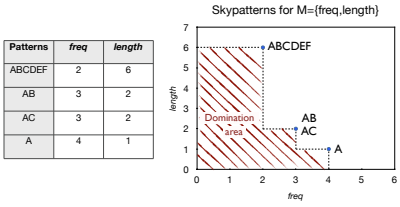



Figure 1: Example of skypattern for a given set of measures

In general, the skypattern mining problem is challenging because of the very high number of candidate patterns (i.e $|\mathcal{L}|$). Indeed, a naive enumeration of $\mathcal{L}$ is not feasible. For example, with 1000 items a naive skypattern approach will need to compute $(2^{1000} - 1) \times |M|$ measures and then compare them. A less naive approach based on heuristics (such as the anti-monotonicity of some measures) may give some results. However, the performance will be closely tied to the underlying properties of the data sets. For instance, in the case of the frequency measure, the density

of the data set plays a major role in the performance and some algorithms are not able to extract frequent patterns at very low thresholds. Nevertheless, considering the following property sheds new insights into an efficient computation of skypattern queries.

*Property 1:* Given a set of measures $M \subseteq \mathcal{M}$, $\mathcal{S}ky(\mathcal{L}, M)$ equals to $\mathcal{S}ky(P, M)$ for any pattern set $P$ containing $\mathcal{S}ky(\mathcal{L}, M)$,

$$(\forall P \subseteq \mathcal{L})(\mathcal{S}ky(\mathcal{L}, M) \subseteq P \Rightarrow \mathcal{S}ky(\mathcal{L}, M) = \mathcal{S}ky(P, M))$$

As $\mathcal{S}ky(\mathcal{L}, M) \subseteq P \subseteq \mathcal{L}$ and $|P| \leq |\mathcal{L}|$, we argue that evaluating $\mathcal{S}ky(P, M)$ is significantly less costly than evaluating $\mathcal{S}ky(\mathcal{L}, M)$ since the cost of $\mathcal{S}ky(x, M)$ generally decreases with the cardinality of $x$. Consequently, we aim to reduce the cost of evaluating $\mathcal{S}ky(P, M)$ by finding a small but relevant set $P$ (i.e. that includes $\mathcal{S}ky(\mathcal{L}, M)$) by means of pattern condensed representations. However, this is not an easy task. A direct approach would be to compute a concise representation for each measure $m \in M$, but this is generally not possible because some measures, such as area or length, are *simply not condensable*. Therefore, our problem can be reformulated as following: *given a set of measures $M$, how can one identify a smaller set of measures $M'$ which allows for the computation of a concise representation on the patterns? In addition, how to use this set of measures to extract efficiently the skypatterns without redundancies?* We address this problem in Section IV.

## IV. Reformulating Skypattern Queries

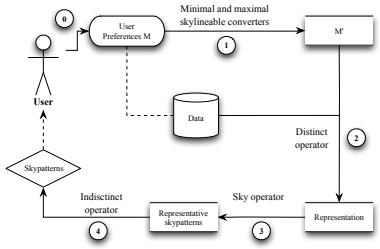



Figure 2: Overview of Aetheris.

In an effort to clarify our methodology, we illustrate in Figure 2 the different processes of our approach called Aetheris. In a first step, and after the user's preferences selection, Aetheris automatically identify a smaller set of measures $M'$ which allows for the computation of a concise representation on the patterns using *converters*. Because of redundancies that may appear in skypatterns, the second step computes a *representative* (i.e., compressed) set of skypatterns. The end-user can either output this compressed representation or the entire list of skypatterns as a final step depending on the application needs. Our methodology revolves around the simple idea that to be able to extract and analyze efficiently skypatterns, one needs to be able to *compress* the patterns that will be used as an input to the skyline operator and then to do a second compression task over the final output (i.e., the skypatterns).

### A. Skylineability of a set of measures

Given some specific measures, it is sometimes easy to point out patterns that are excellent skyline candidates. For instance, let us consider patterns from $\mathcal{D}$ that maximize the cardinality. As the cardinality $length(X)$ *strictly* increases with $X$, the skypattern query $\mathcal{S}ky(\mathcal{L}, \{length(X)\})$ can be defined as a subset of the maximal patterns of $\mathcal{L}$ occurring in $\mathcal{D}$. Unfortunately, this property doesn't hold for other measures such as the frequency (which is only *weakly* decreasing) and the area (which is not monotonic). However, one can notice that the area strictly increases with $X$ when the frequency remains constant. Such a function is said to be *maximally $\{freq\}$-skylineable*.

*Definition 3 (Skylineability):* Given a set of measures $M' \subseteq \mathcal{M}$, a set of measures $M$ is said to be minimally (respectively maximally) $M'$-skylineable iff for any patterns $X =_{M'} Y$ such that $X \subset Y$ (respectively $X \supset Y$), one has $X \succeq_M Y$.

*Definition 4 (Strict skylineability):* Given a set of measures $M' \subseteq \mathcal{M}$ and a set of measures $M$, if $X \succ_M Y$ for any patterns $X =_{M'} Y$ such that $X \subset Y$ (respectively $X \supset Y$), then $M$ is said to be *strictly* minimally (respectively maximally) $M'$-skylineable.

From the previous definitions, given a set of measures $M$ which is maximally $M'$-skylineable, if $X =_{M'} Y$ and $X \supset Y$, it is clear that $X$ cannot be dominated by $Y$ on $M$. For instance, $M = \{freq, area\}$ is strictly maximally $\{freq\}$-skylineable because $area(X)$ strictly increases with the cardinality of $X$ (when the frequency remains constant). Therefore, in our example, $B =_{freq} AB$ and we can directly deduce that $AB \succ_M B$. Notice that $\{freq\}$ is (weakly) maximally (or minimally) $\{freq\}$-skylineable and that $\{length(X)\}$ is strictly maximally $\emptyset$-skylineable. Next subsections will justifiy the notion of *minimal/maximal* in $M'$-skylineability by clearly refering to the minimal/maximal patterns of equivalence classes adequate to $M'$.

*Property 2:* Any set of measures $M$ is minimally and maximally $M$-skylineable.

Property 2 is a very important result as it means that *a set of measures is always skylineable*. Obviously, for a set of measures $M$, the smaller[1] $M'$, the stronger its $M'$-skylineability. For instance, $\{freq\}$-skylineability is more interesting than $\{freq, area\}$-skylineability because $area$ is not a condensable function: there is no pair of distinct patterns $X$ and $Y$ such that $X =_{\{freq,area\}} Y$. How to choose automatically a subset $M'$ is discussed next.

[1]In the sense of cardinality.

### B. Minimal and maximal skylineable converters

Let us first illustrate the general intuition behind an automatic selection technique. Let $M = \{freq\}$ be a set of measures, $X$ and $Y$ be two patterns such that $X \subseteq Y$. Obviously, $M = \{freq\}$ is minimally $\emptyset$-skylineable because $freq$ decreases and $X \succeq_M Y$. Conversely, $M = \{freq\}$ is not *maximally* $\emptyset$-skylineable, but is maximally $\{freq\}$-skylineable. Indeed, if $X =_{\{freq\}} Y$ (i.e., $X$ and $Y$ have the same frequency), then $X \succeq_{\{freq\}} Y$. More generally, any primitive $p$ that is part of the measure $m$ that hinders the $M'$-skylineability of $m$, has to be added to $M'$. We generalize this approach to any primitive-based measure. For this purpose, we define two operators denoted $\underline{c}$ and $\overline{c}$ (see Table III).

Table III: The definition of the minimal and maximal skylineable converters: $\underline{c}$ and $\overline{c}$

| Expr. $e$ | Primitive(s) | $\underline{c}(e)$ | $\overline{c}(e)$ |
|---|---|---|---|
| $e_1 \theta e_2$ | $\theta \in \{+, \times, \cup\}$ | $\underline{c}(e_1) \cup \underline{c}(e_2)$ | $\overline{c}(e_1) \cup \overline{c}(e_2)$ |
| $e_1 \theta e_2$ | $\theta \in \{-, /, \cap\}$ | $\underline{c}(e_1) \cup \overline{c}(e_2)$ | $\overline{c}(e_1) \cup \underline{c}(e_2)$ |
| constant | - | $\emptyset$ | $\emptyset$ |
| $d(X)$ | $d \in \{freq, min, g\}$ | $\emptyset$ | $\{d(X)\}$ |
| $i(X)$ | $i \in \{length, max, sum, freq_\vee, f\}$ | $\{i(X)\}$ | $\emptyset$ |
| $d(e_1)$ | $d \in \{freq, min, g\}$ | $\overline{c}(e_1)$ | $\underline{c}(e_1)$ |
| $i(e_1)$ | $i \in \{length, max, sum, freq_\vee, f\}$ | $\underline{c}(e_1)$ | $\overline{c}(e_1)$ |

Given a primitive-based measure $m \in \mathcal{M}$, the minimal skylineable converter returns a set of measures $M' = \underline{c}(m)$ guaranteeing that for any pattern $X \subset Y$, if $X =_{M'} Y$ then $m(X) \geq m(Y)$. In other words, $X$ dominates $Y$ with respect to $m$. Dually, the maximal converter $\overline{c}$ guarantees that $m(X) \leq m(Y)$ for any pattern $X \subset Y$ such that $X =_{\overline{c}(m)} Y$.

Let us illustrate $\underline{c}$ and $\overline{c}$ on the area measure. The area is defined as a product of the frequency and length. Thus, we report to the first definition in Table III. $\underline{c}(area) = \underline{c}(freq(X)) \cup \underline{c}(length(X)) = \emptyset \cup \{length(X)\} = \{length(X)\}$. Symmetrically, $\overline{c}(area) = \overline{c}(freq(X)) \cup \overline{c}(length(X)) = \{freq(X)\} \cup \emptyset = \{freq(X)\}$. The skylineable converters enable us to automatically find optimization techniques already known for specific measures such as area [22], [23] or growth rate [21] (see Table IV (a)). However, in this work, we *generalize* this principle to cover *any* primitive-based measures. Note that when the converter $\underline{c}$ returns no measure (e.g., $bond$ or $aconf$), it means that the measure decreases with respect to the specialization. Dually, $\overline{c}(m) = \emptyset$ means that $m$ increases with respect to the specialization.

In practice, as the skypatterns are computed for a set of measures, we extend the minimal and maximal converters:

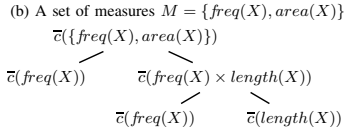
*Definition 5 (Minimal and maximal skylineable converters):* The minimal and maximal skylineable converters defined by Table III for any primitive-based measure are naturally extended to a set of primitive-based measures $M \subseteq \mathcal{M}$: $\overline{c}(M) = \bigcup_{m \in M} \overline{c}(m)$ and $\underline{c}(M) = \bigcup_{m \in M} \underline{c}(m)$.

Table IV: Applying the minimal and maximal converters

(a) Individual measures

| Meas. $m$ | $\overline{c}(m)$ | $\underline{c}(m)$ |
|---|---|---|
| $area$ | $\{freq(X)\}$ | $\{length(X)\}$ |
| $mean$ | $\{min(X.val)\}$ | $\{max(X.val)\}$ |
| $bond$ | $\{freq(X), freq_\vee(X)\}$ | $\emptyset$ |
| $aconf$ | $\{freq(X), max(X.val)\}$ | $\emptyset$ |
| $gr_1$ | $\{freq(X, \mathcal{D}_1)\}$ | $\{freq(X, \mathcal{D}_2)\}$ |

(b) A set of measures $M = \{freq(X), area(X)\}$




For instance, $\overline{c}(\{freq(X), area(X)\}) = \overline{c}(freq(X)) \cup \overline{c}(area(X)) = \{freq(X)\}$ and $\underline{c}(\{freq(X), area(X)\}) = \underline{c}(freq(X)) \cup \underline{c}(area(X)) = \{length(X)\}$. $\overline{c}(\{freq(X), area(X)\}) = \{freq(X)\}$ means that the most specific patterns (when the frequency remains unchanged) maximizes the measures $\{freq(X), area(X)\}$. The following property formalizes this observation:

*Property 3:* A set of primitive-based measures $M \subseteq \mathcal{M}$ is minimally $\underline{c}(M)$-skylineable and maximally $\overline{c}(M)$-skylineable.

In our implementation, the user specified set of measures $M$ is parsed through a syntax tree. Following this step, the minimal and maximal skylineable converters are recursively applied to automically compute $\underline{c}(M)$ and $\overline{c}(M)$ (an example is provided in table IV (b) for $M = \{freq(X), area(X)\}$). This process is illustrated in Figure 2 with the edge labelled 1. From now on, the set of measures $M'$ refers to $\underline{c}(M)$ or $\overline{c}(M)$.

### C. Distinct and indistinct operators

In the previous paragraphs, we remarked the fact that some skypatterns share exactly the same values on the whole set of measures $M'$ (e.g. $B =_{\{freq\}} AB$). This observation leads to the following question: *Is it possible to find some representatives for a group of indistinct skypatterns?* We show that the answer is yes and that instead of directly evaluating the skypattern query on $\mathcal{L}$, we can compute the skypatterns on a condensed representation of $\mathcal{L}$ and then regenerate the entire set of skypatterns. For this end, we introduce the *distinct operator* which produces condensed representations adequate to $M$:

*Definition 6 (Distinct operator):* Given a set of measures $M' \subseteq \mathcal{M}$, the distinct operation for $P \subseteq \mathcal{L}$ with respect to $M'$ and $\theta \in \{\subset, \supset\}$ returns all the patterns $X$ of $P$ such that their generalizations (or specializations) are distinct from $X$

with respect to $M'$:

$$\mathcal{D}is_\theta(P, M') = \{X \in P | \forall Y \theta X : X \neq_{M'} Y\}$$

where $\theta \in \{\subset, \supset\}$.

Given a set of measures $M'$, the set of free (respectively closed) patterns adequate to $M'$ corresponds exactly to $\mathcal{D}is_\subset(\mathcal{L}, M')$ (respectively $\mathcal{D}is_\supset(\mathcal{L}, M')$). For instance, from our toy example, $\mathcal{D}is_\subset(\mathcal{L}, \{freq\}) = \{A, B, C, D, E, F, AD, AE, BC, BD, BE, CD, CE, DE\}$ and $\mathcal{D}is_\supset(\mathcal{L}, \{freq\}) = \{A, D, E, AB, AC, ABCDEF\}$.

We now introduce the *indistinct operator* that enables the retrieval of all the indistinct patterns from their representatives:

*Definition 7 (Indistinct operator):* Given a set of measures $M' \subseteq \mathcal{M}$, the indistinct operation returns all the patterns of $\mathcal{L}$ being indistinct with respect to $M'$ with at least one pattern in $P$.

$$\mathcal{I}nd(\mathcal{L}, M', P) = \{X \in \mathcal{L} | \exists Y \in P : X =_{M'} Y\}$$

For instance, from Table I, the set of patterns that have exactly the same frequency as patterns $B$ and $C$ is $\mathcal{I}nd(\mathcal{L}, \{freq\}, \{AB, AC\}) = \{B, C, AB, AC\}$.

*Property 4:* Given a set of preserving functions $M'$, one has the following relation for any $P \subseteq \mathcal{L}$ and $\theta \in \{\subset, \supset\}$:

$$\mathcal{I}nd(P, M', \mathcal{D}is_\theta(P, M')) = P$$

In other words, the indistinct operator is the inverse function for the distinct operator. For instance, $\mathcal{I}nd(\mathcal{L}, \{freq\}, \mathcal{D}is_\supset(\{B, C, AB, AC\}, \{freq\})) = \{B, C, AB, AC\}$.

### D. Aetheris: Evaluating skypattern query based on skylineability

To compute skypatterns, we would like to confront distinct patterns together instead of individually comparing each pattern. Indeed, the computation of skypatterns with respect to $M = \{freq, area\}$ can be limited to $\mathcal{D}is_\supset(\mathcal{L}, \{freq\})$ because maximal $\{freq\}$-skylineability ensures us that the other patterns are not dominant patterns. For instance, as $AB =_{freq} B$, the $\{freq\}$-skylineability of $M$ gives $AB \succ_M B$ and $B$ cannot be a skypattern. More formally, we know that $\mathcal{S}ky(\mathcal{I}nd(\mathcal{L}, M', \mathcal{D}is_\theta(\mathcal{L}, M')), M) = \mathcal{S}ky(\mathcal{L}, M)$ from Property 4. Theorem 1 now proves that the skypattern operator can be pushed into the indistinct operator:

*Theorem 1 (Operational equivalence):* If a set of measures $M$ is $M'$-skylineable with respect to $\theta \in \{\subset, \supset\}$ and $M'$ is a set of measures, then one has:

$$\mathcal{S}ky(\mathcal{L}, M) = \mathcal{I}nd(\mathcal{L}, M, \mathcal{S}ky(\mathcal{D}is_\theta(\mathcal{L}, M')), M))$$

*Proof:* Let $M$ be a set of measures $M'$-skylineable with $\theta \in \{\subset, \supset\}$.

1.$\mathcal{S}ky(\mathcal{L}, M) \supseteq \mathcal{I}nd(\mathcal{L}, M, \mathcal{S}ky(\mathcal{D}is_\theta(\mathcal{L}, M')), M))$. Let $X \in \mathcal{I}nd(\mathcal{L}, M, \mathcal{S}ky(\mathcal{D}is_\theta(\mathcal{L}, M')), M))$ and $Y \in \mathcal{L}$. There exist $X' \in \mathcal{S}ky(\mathcal{D}is_\theta(\mathcal{L}, M')), M)$ such that $X' =_M X$ and $Y' \in \mathcal{D}is_\theta(\mathcal{L}, M')$ such that $Y' =_{M'} Y$ and $Y' \succeq_M Y$ (i.e., $M'$-skylineability). As $X'$ belongs to $\mathcal{S}ky(\mathcal{D}is_\theta(\mathcal{L}, M'), M)$, it cannot be dominated by any pattern of $\mathcal{D}is_\theta(\mathcal{L}, M')$: $Y' \not\succ_M X$. Thus, $X$ is not dominated by $Y$ (i.e., $X$ is a skyline of $\mathcal{L}$ with respect to $M$) because $X' =_M X$ and $Y' \succeq_M Y$.

2.$\mathcal{S}ky(\mathcal{L}, M) \subseteq \mathcal{I}nd(\mathcal{L}, M, \mathcal{S}ky(\mathcal{D}is_\theta(\mathcal{L}, M')), M))$. Let $Y \in \mathcal{S}ky(\mathcal{L}, M)$. There exists $Y' \in \mathcal{D}is_\theta(\mathcal{L}, M')$ such that $Y' =_{M'} Y$ and $Y' \succeq_M Y$. As $Y$ is a skypattern, one has $Y \succeq_M Y'$ and thus, $Y' =_M Y$. Furthermore, no pattern of $\mathcal{D}is_\theta(\mathcal{L}, M')$ dominates $Y$ nor $Y'$: $Y' \in \mathcal{S}ky(\mathcal{D}is_\theta(\mathcal{L}, M'), M)$. Finally, as $Y' =_M Y$, $Y$ belongs to $\mathcal{I}nd(\mathcal{L}, M, \mathcal{S}ky(\mathcal{D}is_\theta(\mathcal{L}, M')), M))$. ■

It is well-known that the size of adequate condensed representations (i.e., $\mathcal{D}is_\subset(\mathcal{L}, M')$ or $\mathcal{D}is_\supset(\mathcal{L}, M')$) is smaller than the whole collection of patterns [16]. Thus, we have achieved our objective as mentioned in Section III-B. Furthermore, note that if a set of measures is *strictly* $M'$-skylineable, Theorem 1 reduces to the following relation: $\mathcal{S}ky(\mathcal{L}, M) = \mathcal{S}ky(\mathcal{D}is_\theta(\mathcal{L}, M'), M)$ (with $\theta \in \{\subset, \supset\}$). Even if a set of measures is *not* strictly $M'$-skylineable, it is often preferable not to perform the indistinct operation as done in our case study (see Section V-B). In such situation, the skypatterns of $\mathcal{S}ky(\mathcal{D}is_\theta(\mathcal{L}, M'), M)$ form a condensed representation of $\mathcal{S}ky(\mathcal{L}, M)$.

Figure 3 illustrates the computation of the skypatterns with our approach Aetheris. Suppose that $M = \{freq, area\}$, the first step applies the maximal skylineable converter on $M$. Then, the distinct operator preserves the closed itemsets (Step 2). The skyline operator selects the dominant patterns at Step 3 by removing $D$ and $E$ which are dominated by $AB$ (i.e., $area(D) = area(E) = 3 < area(AB) = 6$). Finally, the last step computes the indistinct patterns of skypatterns. Note that this step is unnecessary here because the area is strictly $\{freq\}$-skylineable.

### E. Discussion

As aforementioned, with itemset patterns and the frequency measure, the distinct operator corresponds to the well-known notions of closed or free frequent pattern condensed representations. Indeed, $\mathcal{D}is_\subset(\mathcal{L}, \{freq\})$ is analogous to free frequent itemsets and $\mathcal{D}is_\supset(\mathcal{L}, \{freq\})$ corresponds to closed frequent itemsets. The pattern mining community provides many efficient algorithms to extract these concise representations. In addition, different studies extend the notion of concise representations to any frequency-based measures or condensable function [25]. These theoretical and algorithmic works support our claim that discovery of skypatterns is very efficient, but also extendable to a very large set of measures. This measure genericity allows the end-user to analyze patterns through multiple and useful criteria.

Evaluating efficiently the distinct operator on more complex patterns such as sequences, trees and graphs implies

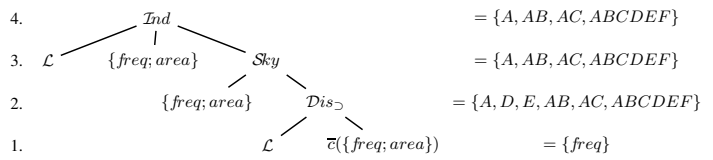



Figure 3: Computing the skypatterns with respect to $\{freq; area\}$ from running example

additional challenges. To cite one example, in the case of sequences, convenient properties such as the *free patterns apriori property* [26], which implies effective search space pruning, cannot be used. Furthermore, in the case of complex patterns, and to the best of our knowledge, no work focused on building concise representations except on the frequency-based measures.

However, it is worth mentioning that Theorem 1 holds for *any* set of measures and *any* language. This means that the efficient extraction of complex skyline patterns (i.e., skyline sequential patterns or skyline graph patterns) is strongly correlated to the advances and progress on complex pattern condensed representations. Last, it is important to notice that Aetheris is not an *exclusive* approach in the sense that it can be coupled with other efficient approaches [27], [28] to extract statistically significant skypatterns.

## V. Experimental Study

We report an experimental study on several benchmarks and a case study from chemoinformatics.

### A. Experiments on UCI benchmarks

*Protocol.* Our approach is the first to mine the whole set of skypatterns in a generic way. As a result, we cannot compare it with earlier methods. Nevertheless, for some data sets, skypatterns can be extracted by applying the skyline operator $\mathcal{S}ky$ as a post-treatment on the collection of itemsets that occurs at least once in the dataset, denoted by $\mathcal{L}$. We call this process the **baseline approach**. Our first batch of experiments focus on comparing runtimes of the baseline approach with respect to **Aetheris**. In our experiments, we limit the set of measures $M'$ to preserving functions only. In this way, we can use any mining algorithm adequate to free and closed itemsets [25]. For a fair comparison, the two approaches use the same implementation of the operator $\mathcal{S}ky$ which is based on the block nested loop (BNL) algorithm [9]. Our second batch of experiments aims at comparing our approach to an optimal constraint-based mining method (with thresholds). For each measure $M_i \in M$, we set the threshold $\sigma_{M_i}$ to $\min_{s \in \mathcal{S}ky\ (\mathcal{L},\ \mathrm{M})}(M_i(s))$. This condition guarantees that no skypatterns will be missed. For instance, in our running example (Figure 1), $\sigma_{freq} = 2$ and $\sigma_{length} = 1$. The set of resulting patterns is called the **optimal constraint-based patterns** (or OCB patterns). This set of patterns needs to be post-processed to find the complete set of skypatterns $\mathcal{S}ky\ (\mathcal{L},\ M)$. Even if this method may seem unrealistic (the user needs to guess optimal thresholds), we still think that this experiment has the benefit of quantifying the reduction of patterns brought by Aetheris even in the scenario where an *ideal end-user* is able to perfectly manage theresholds selections in the constraint-based paradigm.

*Datasets and measures.* Experiments were carried out on 16 various (in terms of dimensions and density) benchmarks from the UCI repository[2]. We considered a number of combinations of primitive-based measures: frequency, area, maximum, minimum, growth rate and mean. Measures using numeric values were applied on attribute values that were randomly generated within the range [0,1] (see Table I). All the tests were performed on a 2.5 GHz Xeon processor with Linux operating system and 2 GB of RAM memory. Running times were averaged over 5 executions.

*Results.* Table V and VI provide an overview of 128 experiments carried out on 16 benchmarks, by aggregating the results for 8 sets of measures. Table V presents averages and maximal results for Aetheris and the baseline approach. Note that runtimes only consider the application of skyline operator and do not take into account mining runtimes to extract collection of itemsets (baseline approach) or the pattern condensed representation (Aetheris approach). Mining condensed representations is generally much more efficient than extracting all itemsets [16]. This means that in practice, the gain of Aetheris on the whole process is even much higher than what is reported. However, because the efficiency of the condensed representations is a well-known result in literature, we prefer in these experiments to focus only on the impact of the skyline operator. It should be noted that in some cases the enumeration of all the itemsets fails (e.g., with `mushroom` and `sick` data sets, see [25] for more details). It means that the baseline approach cannot be applied whereas our approach provides the proper set of skypatterns. This point is a major benefit of our approach.

An important result is that Aetheris always outperforms the baseline approach with at least a factor of 10. The distinct operator used to compute skypatterns speeds up the mining in all cases. The reason is that it drastically reduces the size of the input considered by the skyline operator. However,

[2]http://www.ics.uci.edu/~mlearn/MLRepository.html

when the number of measures increases, the collection returned by the distinct operator becomes less compact and skypattern mining becomes less efficient. Nevertheless, in our experiments, the skyline computation remains extremely fast: there are only 3 experiments requiring more than 1 second with the Aetheris approach (experiments with $M = \{freq; max; area; mean\}$ on `austral`, `crx` and `hepatic`) whereas 61 out of the 128 experiments exceed 1 second for the baseline approach.

Figure 4 (a) depicts the performance of the skyline operator for each of the 128 experiments according to the baseline and Aetheris approaches. As expected, the running time of $\mathcal{S}ky$ increases linearly with the number of itemsets in input. The points corresponding to the Aetheris approach are concentrated on the bottom left corner, showing the efficiency of the method.

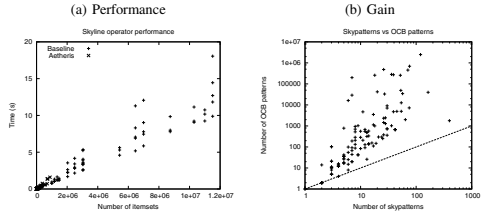



Figure 4: Performance and gain of the skyline patterns.

For each set of measures $M$, Table VI reports the minimal/average/maximal number of skypatterns, the average number of OCB patterns and the average gain of skypatterns (i.e., |# of OCB patterns|/|$\mathcal{S}ky(\mathcal{L}, M)$|). The aim is to illustrate the problem of *"pattern flooding"* that is still appearing even with the optimal constraint-based approach. In contrast, the number of skypatterns is *always extremely low*. At most, there is a maximum of 397 skypatterns (on `anneal` with the frequency and the growth rate measures). Except for the growth rate measure, a higher number of measures leads to a higher number of skypatterns. The explanation is that a pattern rarely dominates all other patterns on the whole set of measures. Interestingly, the gain of a skyline approach (see the last column in Table VI) is always important (greater than 10 and much greater in almost all the cases). Figure 4 (b) summarizes this result by reporting for each experiment the number of OCB patterns compared to the number of skypatterns. The line $y = x$ highlights the gain of our approach: all the points are above the line and in most cases by several orders of magnitude.

### *B. Case Study: discovering toxicophores*

A necessary step in the elaboration of chemicals' protective measures is the thorough identification of their potentially harmful aspects. Consequently, a major issue in chemoinformatics is to establish relationships between chemicals and a given activity (e.g., LC50 in ecotoxicity). Chemical fragments[3] which cause toxicity are called *toxicophores* and their discovery is a major issue as they are at the core of prediction models in (eco)toxicity [2]. The aim of this case study, which is part of a larger research collaboration with a laboratory of medicinal chemistry, is to investigate the use of skypatterns in order to discover toxicophores.

The dataset is collected from the ECB web site[4]. For each chemical, the chemists associate the data with hazard statement codes (HSC) in 3 acute categories: H400 (very toxic, LC50 $\leq$ 1 mg/L), H401 (toxic, 1 mg/L $<$ LC50 $\leq$ 10 mg/L), and H402 (harmful, 10 mg/L $<$ LC50 $\leq$ 100 mg/L). We focus solely on the H400 and H402 classes. The dataset $\mathcal{D}$ consists of 567 chemicals, 372 from the H400 class and 195 from the H402 class. The chemicals are encoded using 129 frequent subgraphs previously extracted from $\mathcal{D}$[5]. The subgraphs are extracted using a 10% relative frequency threshold (experiments with lower thresholds did not bring significant results for the chemists).

The goal of the first experiment is to evaluate the skypattern approach with measures typically used in contrast mining such as the growth rate since toxicophores are linked to a classification problem with respect to the HSC. When associated together, the growth rate and the frequency measures convey the intuitive notion that a candidate toxicophore is a set of fragments whose frequency is strongly higher in the H400 class than the H402 class and is representative enough (i.e., the higher the frequency, the better it is). We do not specify mining runtimes as they are negligible and we only focus on a qualitative analysis for skypatterns.

A first major result is that the number of skypatterns is very small. Using the growth rate and frequency measures, only 8 skypatterns are enumerated and this allows for a direct expert inspection. The chemists emphasize three patterns based on well-known environmental toxicophores, namely the *phenol ring*, the *chloro-substituted aromatic ring*, and the *organo-phosphorus moiety*. The toxicity of the phenol rings is related to hydrophobocity and formation of free radicals [29]. The chloro-substituted aromatic rings and organo-phosphorus moieties are components of widespread pesticides. Moreover, the organo-phosphorus moiety pattern has a high growth rate ($\infty$ value) and a high frequence. This pattern is thus a jumping emerging pattern and the experts compared it furthermore to jumping emerging fragments (JEF) extracted from previous experiments [30]. It appears that the organo-phosphorus moiety pattern is a generalization

[3]A fragment denominates a connected part of a chemical structure containing at least one chemical bond

[4]ECB, European Chemicals Bureau http://ecb.jrc.ec.europa.eu/documentation/ now http://echa.europa.eu/

[5]A chemical $Ch$ contains an item $A$ if $Ch$ supports $A$, and $A$ is a frequent subgraph of $\mathcal{D}$.

Table V: Performance analysis of skypattern mining on UCI benchmarks (time in s)

| Measures $M$ / $\theta$ | Average $\lvert\mathcal{L}\rvert$ | Average $\lvert\mathcal{D}is_\theta(\mathcal{L})\rvert$ | Average time base. | Maximal time base. | Average time Aetheris | Maximal time Aetheris | Average gain of Aetheris |
|---|---|---|---|---|---|---|---|
| $\{freq; area\}$ / maximal (i.e. $\theta = \supset$) | 3,754,792.13 | 63,977.88 | 3.192 | 20.110 | 0.056 | 0.184 | 53.82 |
| $\{freq; min\}$ / minimal (i.e. $\theta = \subset$) | 3,754,792.13 | 187,709.69 | 4.115 | 26.116 | 0.194 | 0.722 | 18.13 |
| $\{freq; max\}$ / maximal | 3,754,792.13 | 92,459.75 | 4.150 | 25.624 | 0.103 | 0.396 | 28.74 |
| $\{freq; max; area\}$ / maximal | 3,754,792.13 | 92,459.75 | 4.808 | 29.562 | 0.122 | 0.446 | 28.76 |
| $\{gr; area\}$ / maximal | 2,559,789.75 | 45,489.94 | 2.280 | 10.180 | 0.050 | 0.176 | 36.94 |
| $\{freq; gr; area\}$ / maximal | 2,559,789.75 | 45,489.94 | 2.709 | 11.146 | 0.059 | 0.184 | 36.97 |
| $\{freq; max; area; mean\}$ / maximal | 3,754,792.13 | 239,017.19 | 6.361 | 39.968 | 0.445 | 1.600 | 10.22 |
| $\{freq; gr\}$ / maximal | 2,559,789.75 | 45,489.94 | 2.274 | 9.242 | 0.046 | 0.144 | 35.95 |

Table VI: Effectiveness of skypattern mining on UCI benchmarks

| Measures $M$ | Minimal # of skypatterns | Average # of skypatterns | Maximal # of skypatterns | Average # of OCB patterns | Average gain of skypatterns |
|---|---|---|---|---|---|
| $\{freq; area\}$ | 1.00 | 4.13 | 8.00 | 91.81 | 13.34 |
| $\{freq; min\}$ | 1.00 | 4.19 | 8.00 | 14403.56 | 2061.81 |
| $\{freq; max\}$ | 2.00 | 10.75 | 42.00 | 46748.50 | 1036.90 |
| $\{freq; max; area\}$ | 2.00 | 14.94 | 57.00 | 52912.13 | 1838.87 |
| $\{gr; area\}$ | 3.00 | 16.06 | 71.00 | 19125.50 | 1021.52 |
| $\{freq; gr; area\}$ | 4.00 | 33.75 | 75.00 | 20453.06 | 399.32 |
| $\{freq; max; area; mean\}$ | 4.00 | 35.06 | 164.00 | 201596.25 | 1905.12 |
| $\{freq; gr\}$ | 6.00 | 48.44 | 397.00 | 2025.94 | 52.79 |

of around 90 JEFs and can be seen as a kind of *maximum common structure* (i.e., consensus structure) of these fragments. The experts highly appreciate that Aetheris is able to provide a synthetic view summarizing the information of a large set of JEFs.

The aim of our second experiment is to integrate and evaluate measures conveying a notion of *background knowledge*. In ecotoxicity, chemists consider that the aromaticity and the density measures may yield an interest for candidate toxicophores. For instance, a common hypothesis is that the higher the chemical density, the stronger its chemical behavior. In addition, chemists know that the aromaticity is a chemical property that favors toxicity since their metabolites can lead to very reactive species which can interact with biomacromolecules in a harmful way. Besides, from a biodegradability point of view, aromatic compounds are among the most recalcitrant of the pollutants. Using chemical knowledge, we are able to compute aromaticity and density on chemical fragments. The aromaticity (or the density) of a pattern is calculated using the $mean$ function defined in Table I based on the aromaticity (or density) of each of the 129 listed subgraphs.

Adding only the density to the growth rate and frequency measures do not deeply change the results: 9 skypatterns are obtained and they are similar to the set of 8 skypatterns previously mined with the growth rate and frequency measures. On the contrary, adding the aromaticity and, even better, both the aromaticity and density, leads to skypatterns with novel chemical characteristics. Once again, the whole set of skypatterns remains small (27 when adding the aromaticity and 38 when adding both the aromaticity and the density) and can be directly analyzed by the chemists. They were especially interested in the following skypattern (provided in Smiles code[6]): {`Clc(ccc)c, cc, ccc, cccc, ccccc, ccc(cc)N`}. This skypattern, including an amine function, was not detected during the first experiment and can be exemplified by the chloroaniline derivatives. Indeed, these derivatives are environmentally hazardous since they are very toxic for aquatic species [31]. The experiment shows that background knowledge can successfully be translated to preferences and that Aetheris is straightforwardly able to discover few and promising patterns.

## VI. Conclusion

In this paper, we introduce the skyline pattern mining problem. Our goal is to make the result of pattern mining useful from a *user-preference point of view*. We propose Aetheris, the first approach to mine skypatterns in a generic way (i.e., with set of measures and applications to various pattern domains). Aetheris is threshold-free and only needs, as parameters, the measures and the data set. Our approach is based on the key notion of skylineability that supports efficient skypattern computation thanks to an adequate condensed representation of patterns. Experiments performed on several datasets and a use case from chemoinformatics show the efficiency of Aetheris according to both quantitative and qualitative aspects.

An important direction for future work is to improve even further the performance of the algorithm. An idea that we want to investigate is the assimilation of the skyline operator with a pruning strategy. Indeed, Aetheris still applies the skyline operators on pattern collections that may be still relatively large. Other perspectives lie in the improvement

[6]http://www.daylight.com/dayhtml/doc/theory/theory.smiles.html

of adequate condensed representations on more complex patterns (i.e., sequences, graphs and dynamic graphs) which is a timely challenge.

*Acknowledgements.* The authors thank the CERM Laboratory (University of Caen, France) for providing the chemical data and in particular Alban Lepailleur for his highly valuable comments. The authors thank Bertrand Cuissart and Guillaume Poezevara for their contribution for major steps of this work and very fruitful discussions. This work is partly supported by the ANR (French Research National Agency) funded projects BINGO*2* ANR-07-MDCO-014 and FOSTER ANR-2010-COSI-012-02.

## REFERENCES

[1] M. J. Zaki and K. Sequeira, "Data mining in computational biology," in *Handbook of Computational Molecular Biology*. Chapman & Hall/CRC Press, 2006, ch. 38, p. 1–26.

[2] J. Auer and J. Bajorath, "Emerging chemical patterns: A new methodology for molecular classification and compound selection," *J. Chem. Inf. Mod*, vol. 46, no. 6, p. 2502–2514, 2006.

[3] R. Agrawal, T. Imielinski, and A. Swami, "Mining association rules between sets of items in large database," in *SIGMOD*, 1993, p. 207–216.

[4] H. Mannila and H. Toivonen, "Levelwise search and borders of theories in knowledge discovery," *DMKD*, vol. 1, no. 3, p. 241–258, 1997.

[5] F. Bonchi, F. Giannotti, C. Lucchese, S. Orlando, R. Perego, and R. Trasarti, "A constraint-based querying system for exploratory pattern discovery," *Inf. Syst.*, vol. 34, no. 1, p. 3–27, 2009.

[6] S. Bistarelli and F. Bonchi, "Soft constraint based pattern mining," *Data Knowl. Eng.*, vol. 62, no. 1, p. 118–137, 2007.

[7] Y. Ke, J. Cheng, and J. X. Yu, "Top-k correlative graph mining," in *SIAM DM*, 2009, p. 1038–1049.

[8] J. Wang, J. Han, Y. Lu, and P. Tzvetkov, "TFP: An efficient algorithm for mining top-k frequent closed itemsets," *TKDE*, vol. 17, p. 652–664, 2005.

[9] S. Börzsönyi, D. Kossmann, and K. Stocker, "The skyline operator," in *ICDE*, 2001, p. 421–430.

[10] A. N. Papadopoulos, A. Lyritsis, and Y. Manolopoulos, "Skygraph: an algorithm for important subgraph discovery in relational graphs," *DMKD*, vol. 17, no. 1, p. 57–76, 2008.

[11] J. Matousek, "Computing dominances in $e^n$," *Inf. Process. Lett.*, vol. 38, no. 5, p. 277–278, 1991.

[12] H. T. Kung, F. Luccio, and F. P. Preparata, "On finding the maxima of a set of vectors," *J. ACM*, vol. 22, no. 4, p. 469–476, 1975.

[13] R. E. Steuer, *Multiple Criteria Optimization: Theory, Computation and Application*. John Wiley, 546 pp, 1986.

[14] F. Pennerath and A. Napoli, "The model of most informative patterns and its application to knowledge extraction from graph databases," in *ECML/PKDD*, 2009, p. 205–220.

[15] R. T. Ng, V. S. Lakshmanan, J. Han, and A. Pang, "Exploratory mining and pruning optimizations of constrained associations rules," in *SIGMOD*, 1998, p. 13–24.

[16] T. Calders, C. Rigotti, and J.-F. Boulicaut, "A survey on condensed representations for frequent sets," in *Constraint-Based Mining and Inductive Databases*. Springer, 2004, p. 64–80.

[17] A. Siebes, J. Vreeken, and M. Van Leeuwen, "Item sets that compress," in *SIAM DM*, 2006.

[18] A. Knobbe and E. Ho, "Pattern teams," in *ECML/PKDD*, 2006, p. 577–584.

[19] L. De Raedt and A. Zimmermann, "Constraint-based pattern set mining," in *SIAM DM*, 2007.

[20] B. Bringmann and A. Zimmermann, "The chosen few: On identifying valuable patterns," in *IEEE ICDM*, 2007, p. 63–72.

[21] G. C. Garriga, P. Kralj, and N. Lavrac, "Closed sets for labeled data," *J. Mach. Learn. Res.*, vol. 9, p. 559–580, 2008.

[22] K.-N. Kontonasios and T. De Bie, "An information-theoretic approach to finding informative noisy tiles in binary databases," in *SIAM DM*, 2010, p. 153–164.

[23] F. Geerts, B. Goethals, and T. Mielikäinen, "Tiling databases," in *Discovery Science*, 2004, p. 278–289.

[24] A. Soulet and B. Crémilleux, "Mining constraint-based patterns using automatic relaxation," *Intell. Data Anal.*, vol. 13, no. 1, p. 109–133, 2009.

[25] A. Soulet and B. Crémilleux, "Adequate condensed representations of patterns," *DMKD*, vol. 17, no. 1, p. 94–110, 2008.

[26] D. Lo, S.-C. Khoo, and J. Li, "Mining and ranking generators of sequential patterns," in *SIAM DM*, 2008, p. 553–564.

[27] N. Tatti, "Probably the best itemsets," in *KDD*, 2010, p. 293–302.

[28] G. I. Webb, "Self-sufficient itemsets: An approach to screening potentially interesting associations between items," *TKDD*, vol. 4, no. 1, 2010.

[29] C. Hansch, S. McCarns, C. Smith, and D. Dodittle, "Comparative qsar evidence for a free-radical mechanism of phenol-induced toxicity," *Chem. Biol. Interact.*, vol. 127, p. 61–72, 2000.

[30] S. Lozano, G. Poezevara, M.-P. Halm, and et al., "Introduction of jumping fragments in combination with QSARs for the assessment of classification in ecotoxicology," *Journal of Chemical Information and Modeling*, vol. 50, no. 8, p. 1330–1339, 2010.

[31] E. Argese, C. Bettiol, F. Agnoli, A. Zambon, M. Mazzola, and A. Ghirardini, "Assessment of chloroaniline toxicity by the submitochondrial particle assay," *Environ. Toxicol. Chem.*, vol. 20, p. 826–832, 2001.

# How Your Supporters and Opponents Define Your Interestingness

Bruno Crémilleux[1], Arnaud Giacometti[2], and Arnaud Soulet[2]

[1]Normandie Univ, UNICAEN, ENSICAEN, CNRS – UMR GREYC, France
bruno.cremilleux@unicaen.fr
[2]Université de Tours – LIFAT EA 6300, France
firstname.lastname@univ-tours.fr

**Abstract.** How can one determine whether a data mining method extracts interesting patterns? The paper deals with this core question in the context of unsupervised problems with binary data. We formalize the quality of a data mining method by identifying patterns – the *supporters* and *opponents* – which are related to a pattern extracted by a method. We define a typology offering a global picture of the methods based on two complementary criteria to evaluate and interpret their interests. The quality of a data mining method is quantified via an evaluation complexity analysis based on the number of supporters and opponents of a pattern extracted by the method. We provide an experimental study on the evaluation of the quality of the methods.

## 1 Introduction

In contrast to a lot of data analysis methods where the goal is to describe all the data with one model, pattern mining focuses on information describing only parts of the data. However, in practice, the number of discovered patterns is huge and patterns have to be filtered or ranked according to additional quality criteria in order to be used by a data analyst. As surveyed by Vreeken and Tatti [26], there exist numerous methods for evaluating the interestingness of extracted patterns, e.g. based on simple measures or the use of statistical testing. However, it remains difficult to clearly identify the advantages and limitations of each approach.

How can one determine whether a data mining method extracts interesting patterns? How can one know and evaluate if a data mining method is better than another for a given task? Our work addresses these core questions. Completely answering those questions is clearly out of the scope of this (or any single) paper but we propose major improvements in these directions in the context of unsupervised problems with binary data.

Our goal is to propose an interestingness theory independent of any assumption about the data such as a model of the data or an expectation using statistical tests. Our key principle to assess the quality of a data mining method extracting a pattern $X$ is to study the relationships between $X$ and the other patterns when $X$ is selected. Roughly speaking, the higher the number of necessary comparisons between $X$ and the other patterns to select $X$, the higher the quality

of the method. As an example, let us consider correlation measures such as the lift [26] and the productive itemset [27]. While calculating lift involves only the individual items contained in $X$ to select $X$, the productive itemset involves all subsets of $X$. A pattern selected by the productive itemset must satisfy more tests and the productive itemset is a more effective selector for mining correlated itemsets. Our framework addresses methods to select patterns such as interestingness measures [25], the constraint-based pattern mining imposing constraints on a single pattern [20] or several patterns such as condensed representations of patterns [5] or *top-k* patterns [9]. We call *selector* a data mining method providing patterns. The goal of our framework is to evaluate the quality of a selector and therefore the interestingness of the patterns extracted by the selector. For that purpose, we introduce the notions of *supporter* and *opponent*. A supporter $Y$ of $X$ is a pattern which increases the interestingness of $X$ when only the support of $Y$ increases while all other patterns' support remains unchanged. In other words, when the support of $Y$ increases, it raises the likelihood of $X$ to be selected and therefore $Y$ supports $X$ to be selected. Analogously, an opponent $Y$ of $X$ is a pattern which decreases the interestingness of $X$ when only the support of $Y$ decreases. We show that the number of supporters and opponents and their relations with $X$ ($Y$ is a generalization or a specialization of $X$, $Y$ and $X$ are incomparable) provide meaningful information about the quality of the selector at hand. Notably, this approach evaluates the quality of a selector only based on the relationships between the patterns from the data without assuming a model or any hypothesis on the data.

This paper formalizes the relationships between patterns to evaluate the quality of a selector through the new notions of supporters and opponents. We present a typology of selectors defined by formal properties and based on two complementary criteria to evaluate and interpret the quality of a selector. This typology offers a global picture of selectors and clarifies their interests and limitations. Highlighting the kinds of patterns' relationships required by a selector helps to compare selectors to each other. We quantify the quality of a selector via an evaluation complexity analysis based on its number of supporters and opponents. This analysis enables us to contrast the quality of a selector with its computing cost. Finally, we conduct an experimental study in the context of correlation measures to evaluate the quality of selectors according to their complexity.

This paper is structured as follows. Section 2 discusses related work. Section 3 introduces preliminaries and defines what a selector is. We present the key notions of supporters and opponents in Section 4 and the typology of selectors in Section 5. We continue with the analysis of the complexity of the selectors in Section 6. Section 7 provides an experimental study on the evaluation of the quality of a few selectors. We round up with discussion and conclusion in Section 8.

## 2 Related Work

As we focus on formal approaches on interestingness, experimental protocols to evaluate the quality of a method like rediscovery [29] or randomization [14]

are out of the scope of this related work. The proposal of a general theory of interestingness was already indicated as a challenge for the past decade [8, 16].

Several approaches have been proposed in the literature to analyze pattern discovery methods. Regarding condensed representations of patterns, the size of a condensed representation is often used as an objective measure to assess its quality [5]. As a condensed representation based on closed patterns is always more compact than a condensed representation based on free (or key) patterns, closed patterns are deemed most interesting. However one of the most compact condensed representations – non-derivable itemsets (NDI) [4] – is little used. The semantics of NDI, which appears complex, may explain this unpopularity. In this paper, we propose a measure to formally identify the complexity of a selector (cf. Section 6). The survey of Vreeken and Tatti [26] presents interestingness measures on patterns by dividing them into two categories, absolute measures and advanced ones. An absolute measure is informally defined as follows: "score patterns using only the data at hand, without contrasting their calculations over the data to any expectation using statistical tests". Advanced measures were introduced to limit redundancy in the results. They are based on statistical models (independence model, partition models, MaxEnt models) having different complexities. Our formalization of complexity classes based on relationships between patterns clarifies the distinction between absolute measures and advanced ones.

A lot of works [12, 18, 23, 24] propose axioms that should be satisfied by an interestingness measure for association rule in order that the measure is considered relevant. These methods state what should be the expected variations of a well-behaved measure under certain conditions (e.g., when the frequency of the body or the head of the association rule increases). More recently, these works were extended to itemsets [15, 27] but only by considering their subsets. Our proposal systematizes this approach by taking into account all patterns of the lattice. Besides, the axioms previously introduced in the literature are mainly focused on correlation measures and there are not such axioms for constraints. In this paper, we generalize these principles to constraints (cf. Section 5).

There are very few attempts to define interactions between patterns when evaluating an interestingness measure. The concept of global constraints has been informally defined in [6]. This notion has been formalized in [13] by defining a relational algebra extended to pattern discovery. Our framework provides a broader and more precise formal definition, especially to better analyze the interrelationships between the patterns.

## 3 Preliminaries

Let $\mathcal{I}$ be a set of distinct literals called *items*, an itemset (or pattern) is a subset of $\mathcal{I}$. The language of itemsets corresponds to $\mathcal{L} = 2^{\mathcal{I}}$. A transactional dataset is a multi-set of itemsets of $\mathcal{L}$. Each itemset, usually called *transaction*, is a dataset entry. For instance, Table 2 gives three transactional datasets with 5 transactions $t_1, \ldots, t_5$ each described by 3 items $A$, $B$ and $C$. Note that the transaction $t_5$ in $\mathcal{D}_1$ is empty. $\mathcal{D}$ denotes a dataset and $\Delta$ all datasets. The *frequency* of an

| Name | Definition |
|---|---|
| | Correlation |
| support [1] | $freq(X)/freq(\emptyset)$ |
| all-confidence [21] | $freq(X)/\max_{i \in X} freq(\{i\})$ |
| bond [21] | $freq(X)/\lvert\{t \in \mathcal{D} : X \cap t \neq \emptyset\}\rvert$ |
| lift [26] | $supp(X)/\Pi_{i \in X} supp(i)$ |
| productive itemset [27] | $(\forall Y \subset X)(prod(X) \Rightarrow supp(X) > supp(Y) \times supp(X \setminus Y))$ |
| | Condensed Representation (CR) |
| maximal itemset [19] | $(\forall Y \supset X)(max(X) \Rightarrow freq(X) \geq \gamma \wedge freq(Y) < \gamma)$ |
| free itemset [2] | $(\forall Y \subset X)(free(X) \Rightarrow freq(X) < freq(Y))$ |
| closed itemset [22] | $(\forall Y \supset X)(closed(X) \Rightarrow freq(X) > freq(Y))$ |
| non-derivable itemset [4] | $(\forall X \in \mathcal{L})(ndi(X) \Leftrightarrow LB(X,\mathcal{D}) \neq UB(X,\mathcal{D}))$ where $LB(X,\mathcal{D})$ and $UB(X,\mathcal{D})$ are respectively lower and upper bounds derived from subsets of $X$ in $\mathcal{D}$ |
| | Other |
| top-$k$ freq. itemset [9] | $(\forall X \in \mathcal{L})(top_k(X) \Leftrightarrow \lvert\{Y \in \mathcal{L} : freq(Y) > freq(X)\}\rvert < k)$ |
| FPOF [17] | $-\sum_{Y \subseteq X} freq(Y)/\sum_{Z \in \mathcal{L}} freq(Z)$ |

**Table 1.** Itemset mining approaches based on frequency

itemset $X$, denoted by $freq(X,\mathcal{D})$, is the number of transactions of $\mathcal{D}$ containing $X$. For simplicity, we write $freq(X)$ when there is no ambiguity.

Constraint-based pattern mining [19] aims at enumerating all patterns occurring at least once in a dataset $\mathcal{D}$ and satisfying a user-defined selection predicate $q$. A well-known example is the minimal support constraint, based on the frequency measure, which provides the patterns having a support greater than a given minimal threshold. Despite the filtering performed by a constraint, the collection of mined patterns is often too large to be managed and interestingness measures are additionally used to rank patterns and focus on the most relevant ones. There are numerous measures [24], several of which (support, bond, lift, all-confidence) are given in Table 1. In this paper, we consider constraint-based pattern mining imposing constraints on a single pattern or several patterns such as condensed representations of patterns or top-k patterns. Table 1 depicts several examples of constraints. The productive itemset is here defined as a constraint.

Let $\mathbb{S}$ be a poset. We formally define the notion of *selector* as follows.

**Definition 1 (Interestingness Selector).** *An interestingness selector $s$ is a function defined from $\mathcal{L} \times \Delta$ to $\mathbb{S}$ that increases when $X$ is more interesting.*

$\mathbb{S}$ is the set of reals $\mathbb{R}$ if the selector is an interestingness measure[1] and booleans $\mathbb{B}$ (i.e. *true* or *false*) with the order $false < true$ if the selector is a constraint. Clearly, selectors define very different views on what should be a relevant pattern. Relevance may highlight correlations between items (regularity), correlations with a class of the dataset (contrast), removing redundancy

[1] The choice of the order $<_{\mathbb{S}}$ has an impact. For instance, in the case of support, $<_{\mathbb{S}}=<_{\mathbb{R}}$ (resp. $<_{\mathbb{S}}=>_{\mathbb{R}}$) enables us to select the positive (resp. negative) correlations.

(condensed representation), complementarity between several patterns (top-$k$), outlier detection such as the FPOF measure (cf. Table 1).

## 4 Framework of Supporters and Opponents

### 4.1 Fundamental definitions and notations

Deciding if a pattern is interesting (and why) generally depends on its frequency, but also on the frequencies of some other patterns. In our framework, we show how the knowledge of those patterns for a given selector makes it possible to qualify this selector and evaluate its quality.

More precisely, in order to isolate the impact of the change in frequency of a pattern $Y$ on the evaluation of the interestingness of a pattern $X$, we propose to compare the interestingness of the assessed pattern $X$ with respect to two very similar datasets $\mathcal{D}$ and $\mathcal{D}'$, where only the frequency of itemset $Y$ varies. Therefore, we introduce the following definition.

**Definition 2 (Increasing at a point).** *Compared to $\mathcal{D}$, a dataset $\mathcal{D}'$ is increasing at a point $Y$, denoted by $\mathcal{D} <_Y \mathcal{D}'$, iff $freq(Y,\mathcal{D}) < freq(Y,\mathcal{D}')$ and $freq(X,\mathcal{D}) = freq(X,\mathcal{D}')$ for all patterns $X \neq Y$.*

For instance, the first two datasets provided by Table 2 satisfy $\mathcal{D}_1 <_{ABC} \mathcal{D}_2$. It means that patterns $\emptyset$, $A$, $B$, $C$, $AB$, $AC$, $BC$ have the same frequency in both datasets, while the frequency of $ABC$ is greater in $\mathcal{D}_2$. Indeed, we have $freq(ABC,\mathcal{D}_1) = 1$, whereas $freq(ABC,\mathcal{D}_2) = 2$. In the same way, we can easily see that $\mathcal{D}_1 <_A \mathcal{D}_3$ due to the addition in $\mathcal{D}_3$ (compared to $\mathcal{D}_1$) of an item $A$ in the fifth transaction $t_5$. Thus, we have $freq(A,\mathcal{D}_1) = 3$, whereas $freq(A,\mathcal{D}_3) = 4$, and $freq(X,\mathcal{D}_1) = freq(X,\mathcal{D}_3)$ for all other patterns $X \neq A$.

$\mathcal{D}_1$

| Trans. | Items |
|---|---|
| $t_1$ | $A$ $B$ $C$ |
| $t_2$ | $A$ $B$ |
| $t_3$ | $A$ $C$ |
| $t_4$ | $B$ $C$ |
| $t_5$ | |

$\mathcal{D}_2$

| Trans. | Items |
|---|---|
| $t_1$ | $A$ $B$ $C$ |
| $t_2$ | $A$ $B$ **C** |
| $t_3$ | $A$ |
| $t_4$ | $B$ |
| $t_5$ | **C** |

$\mathcal{D}_3$

| Trans. | Items |
|---|---|
| $t_1$ | $A$ $B$ $C$ |
| $t_2$ | $A$ $B$ |
| $t_3$ | $A$ $C$ |
| $t_4$ | $B$ $C$ |
| $t_5$ | **A** |

**Table 2.** Three toy datasets with slight variations

Intuitively, given a selector $s$, a *supporter* $Y$ of an assessed pattern $X$ is a pattern that increases the interestingness of $X$ when only the support of $Y$ increases (while all other patterns' support remains unchanged). In other words, when the support of $Y$ increases, it raises the likelihood of $X$ to be selected using $s$. Conversely, if $Y$ is an *opponent* of $X$, when the support of $Y$ increases,

it reduces the likelihood of $X$ to be selected. Using Definition 2, the following definition formalizes these notions of supporter and opponent.

**Definition 3 (Supporters and opponents).** *Given a selector $s$, let $X$ be a pattern in $\mathcal{L}$. $Y$ is a supporter of $X$ for $s$, denoted by $Y \in s^+(X)$, iff there exist two datasets $\mathcal{D}$ and $\mathcal{D}'$ such that $\mathcal{D}' >_Y \mathcal{D}$ and $s(X, \mathcal{D}') > s(X, \mathcal{D})$.*

*Conversely, $Y$ is an opponent of $X$ for $s$, denoted by $Y \in s^-(X)$, iff there exist two datasets $\mathcal{D}$ and $\mathcal{D}'$ such that $\mathcal{D}' >_Y \mathcal{D}$ and $s(X, \mathcal{D}') < s(X, \mathcal{D})$.*

Given a selector, the strength of the notions of *supporter* and *opponent* is to clearly identify the patterns actually involved in the evaluation of an assessed pattern. Moreover, it is important to note that the set of supporters and opponents of a pattern (given a selector) is not dependent on a specific dataset. They are a property of a given selector.

Considering the datasets given in Table 2, let us illustrate Definition 3 with the all-confidence selector. We already noted that $\mathcal{D}_1 <_{ABC} \mathcal{D}_2$. Additionally, we have $\textit{all-conf}(ABC, \mathcal{D}_1) = \frac{\textit{freq}(ABC,\mathcal{D}_1)}{max_{i \in ABC}\textit{freq}(i,\mathcal{D}_1)} = \frac{1}{3}$, whereas $\textit{all-conf}(ABC, \mathcal{D}_2) = \frac{\textit{freq}(ABC,\mathcal{D}_2)}{max_{i \in ABC}\textit{freq}(i,\mathcal{D}_2)} = \frac{2}{3}$. Therefore, we have $\mathcal{D}_1 <_{ABC} \mathcal{D}_2$ and $\textit{all-conf}(ABC, \mathcal{D}_1) < \textit{all-conf}(ABC, \mathcal{D}_2)$, which means that $ABC$ is a supporter of itself for the all-confidence measure. On the other hand, we have $\mathcal{D}_1 <_A \mathcal{D}_3$, $\textit{all-conf}(ABC, \mathcal{D}_1) = \frac{1}{3} > \textit{all-conf}(ABC, \mathcal{D}_3) = \frac{1}{4}$. Thus, by Definition 3, $A$ is an opponent of $ABC$ for the all-confidence measure.

More generally, it is possible to show that for all patterns $X$, $\textit{all-conf}^+(X)$ is equal to $\{X\}$, i.e. $X$ has no supporters other than itself, and that $\textit{all-conf}^-(X) = \{i : i \in X\}$. In the following Section 4.2, we give the set of supporters and opponents for a representative set of usual selectors.

### 4.2 Supporters and opponents of usual selectors

In this section, we give the sets of supporters $s^+$ and opponents $s^-$ for a representative set of selectors $s$. These sets are presented in Table 3 both for interestingness measures (support, all-confidence, bond, lift, etc.) and boolean constraints (productive, free, closed itemset, etc.). Due to lack of space, we do not present a proof for every selector considered in Table 3. Nevertheless, we provide a proof for two examples: the lift measure (see Property 1) and the free constraint (see Property 2). Note that the schema of these proofs could be easily adapted to identify the supporters and opponents of other correlation measures (support, all-confidence, bond, etc.) and other condensed representation constraints (maximal, closed, etc.).

Before detailing the proofs of Properties 1 and 2, given an itemset $X$, Lemma 1 stresses that it is always possible to build two transactional datasets $\mathcal{D}$ and $\mathcal{D}'$ such that $\mathcal{D}'$ is increasing at $X$ in comparison to $\mathcal{D}$, i.e. $\mathcal{D}' >_X \mathcal{D}$. Note that the sets of transactions $\mathcal{D}_X^-$ and $\mathcal{D}_X^+$ introduced in this lemma are crucial to identify the sets of supporters and opponents of a selector.

| Selector $s$ | $s^+(X)$ | $s^-(X)$ | $\lvert s^\pm(X)\rvert$ |
|---|---|---|---|
| support | $\{X\}$ | $\{\emptyset\}$ | $O(1)$ |
| all-confidence | $\{X\}$ | sing. | $O(k)$ |
| bond | $\{X\}$ | sing. | $O(k)$ |
| lift | $\{X\}$ | sing. | $O(k)$ |
| prod. itemset | $\{X\}$ | $X^{\downarrow}$ | $O(2^k)$ |
| max. itemset | $\{X\}$ | $X^{\overline{\uparrow}}$ | $O(n-k)$ |
| free itemset | $X^{\underline{\downarrow}}$ | $\{X\}$ | $O(k)$ |
| closed itemset | $\{X\}$ | $X^{\overline{\uparrow}}$ | $O(n-k)$ |
| NDI | $X^{\downarrow}$ | $X^{\downarrow}$ | $O(2^k)$ |
| top-$k$ frequent | $\{X\}$ | $X^{\leftrightarrow}$ | $O(2^n-2^k$ $-2^{n-k})$ |
| FPOF | $X^{\uparrow}\cup X^{\leftrightarrow}$ | $X^{\downarrow}\cup\{X\}$ | $O(2^n)$ |

| Notation | Formula |
|---|---|
| $k$ | $\lvert X\rvert$ |
| $n$ | $\lvert\mathcal{I}\rvert$ |
| *singletons* | $\{\{i\} : i \in X\}$ |
| *direct sub.* $X^{\underline{\downarrow}}$ | $\{X \setminus \{i\} : i \in X\}$ |
| *direct sup.* $X^{\overline{\uparrow}}$ | $\{X \cup \{i\} : i \in \mathcal{I} \setminus X\}$ |
| subsets $X^{\downarrow}$ | $2^X \setminus \{X\}$ |
| supersets $X^{\uparrow}$ | $\{Y \in \mathcal{L} : X \subset Y\}$ |
| incomp. $X^{\leftrightarrow}$ | $\{Y \in \mathcal{L} : Y \not\subseteq X$ $\wedge X \not\subseteq Y\}$ |
| lattice | $\mathcal{L} \setminus \{X\}$ |

**Table 3.** Analysis of methods based on supporters and opponents

**Lemma 1.** *Given an itemset $X \subseteq \mathcal{I}$, let $\mathcal{D}^-_X$ and $\mathcal{D}^+_X$ be the datasets defined by $\mathcal{D}^+_X = \{Y \subseteq X : |X \setminus Y| \text{ is even}\}$ and $\mathcal{D}^-_X = 2^X \setminus \mathcal{D}^+_X$. We have $\mathcal{D}^-_X <_X \mathcal{D}^+_X$.*

For instance, using datasets shown in Table 2, it is easy to see that $\mathcal{D}_2 = \{ABC\} \cup \mathcal{D}^+_{ABC}$ with $\mathcal{D}^+_{ABC} = \{ABC, A, B, C\}$, and $\mathcal{D}_1 = \{ABC\} \cup \mathcal{D}^-_{ABC}$ with $\mathcal{D}^-_{ABC} = \{AB, AC, BC, \emptyset\}$. Thus, Lemma 1 implies that $\mathcal{D}_1 <_{ABC} \mathcal{D}_2$. Given $\mathcal{D}_0 = \{ABC, AB, AC, BC\}$, we can also check that $\mathcal{D}_1 = \mathcal{D}_0 \cup \mathcal{D}^-_A$ with $\mathcal{D}^-_A = \emptyset$, and $\mathcal{D}_3 = \mathcal{D}_0 \cup \mathcal{D}^+_A$ with $\mathcal{D}^+_A = \{A\}$. Thus, Lemma 1 implies that $\mathcal{D}_1 <_A \mathcal{D}_3$.

*Proof.* First, it is easy to see hat $freq(X, \mathcal{D}^-_X) = 0$ and $freq(X, \mathcal{D}^+_X) = 1$, which shows that $freq(X, \mathcal{D}^+_X) > freq(X, \mathcal{D}^-_X)$. Then, for all itemsets $Y \neq X$, if $Y \not\subseteq X$, we have $freq(Y, \mathcal{D}^-_X) = freq(Y, \mathcal{D}^+_X) = 0$. Otherwise, if $Y \subseteq X$, we can see that $freq(Y, \mathcal{D}^-_X) = freq(Y, \mathcal{D}^+_X) = freq(Y, 2^X)/2$ where $freq(Y, 2^X) = |\{t \in 2^X : Y \subseteq t\}| = |\{Y \cup t : t \in 2^{X \setminus Y}\}| = 2^{|X|-|Y|}$. Thus, for all $Y \neq X$, $freq(Y, \mathcal{D}^-_X) = freq(Y, \mathcal{D}^+_X)$, which completes the proof that $\mathcal{D}^-_X <_X \mathcal{D}^+_X$. □

Using Lemma 1, we now prove Property 1, which defines the supporters and opponents of the lift measure.

*Property 1.* For all itemsets $X$ such that $|X| > 1$, $lift^+(X) = \{X\}$ and $lift^-(X) = \{\{i\} : i \in X\}$.

*Proof.* Given an itemset $X$ such that $|X| > 1$, we distinguish three cases:
**1**. Let $\mathbf{Y = X}$ and two datasets $\mathcal{D}'$ and $\mathcal{D}$ such that $\mathcal{D}' >_Y \mathcal{D}$. By definition we have: $freq(X, \mathcal{D}') > freq(X, \mathcal{D})$ and $freq(Z, \mathcal{D}') = freq(Z, \mathcal{D})$ for all $Z \neq (Y = X)$. Because $|X| > 1$, we also have $\{i\} \neq X$ for all $i \in X$. Therefore, $freq(\{i\}, \mathcal{D}') = freq(\{i\}, \mathcal{D})$ for all $i \in X$, which implies that the denominators of $lift(X, \mathcal{D}')$ and $lift(X, \mathcal{D}')$ are equal. Finally, we have $lift(X, \mathcal{D}') = \frac{supp(X, \mathcal{D}')}{\prod_{i \in X} supp(\{i\}, \mathcal{D}')} > lift(X, \mathcal{D}) = \frac{supp(X, \mathcal{D})}{\prod_{i \in X} supp(\{i\}, \mathcal{D})}$, which shows that $X \in lift^+(X)$, whereas $X \notin lift^-(X)$.
**2**. Let $Y$ be an itemset such that $\mathbf{Y \neq X}$ and $\mathbf{|Y| > 1}$, and two datasets $\mathcal{D}'$ and $\mathcal{D}$ such that $\mathcal{D}' >_Y \mathcal{D}$. By definition, we have: $freq(X, \mathcal{D}') = freq(X, \mathcal{D})$ since $Y \neq X$,

and $freq(\{i\}, \mathcal{D}') = freq(\{i\}, \mathcal{D})$ for all $i \in X$ since $Y \neq \{i\}$ (indeed, we assume that $|Y| > 1$). Thus, we necessarily have $lift(X, \mathcal{D}') = lift(X, \mathcal{D})$ for all datasets $\mathcal{D}'$ and $\mathcal{D}$ such that $\mathcal{D}' >_Y \mathcal{D}$. It implies that for all itemsets $Y$ such that $Y \neq X$ and $|Y| > 1$, $Y \notin lift^+(X)$ and $Y \notin lift^-(X)$.

**3**. Let $Y$ be an itemset such that $\mathbf{Y} \neq \mathbf{X}$ and $|\mathbf{Y}| = \mathbf{1}$, and two datasets $\mathcal{D}'$ and $\mathcal{D}$ such that $\mathcal{D}' >_Y \mathcal{D}$. Using the same reasoning as before, it is easy to see that if $Y \not\subset X$, we necessarily have $lift(X, \mathcal{D}') = lift(X, \mathcal{D})$, which implies that $Y \notin lift^+(X)$ and $Y \notin lift^-(X)$. Dually, if $Y \subset X$, because $|Y| = 1$, there exists $j \in X$ such that $Y = \{j\}$. Since $\mathcal{D}' >_Y \mathcal{D}$ and $X \neq Y$, we have $freq(X, \mathcal{D}') = freq(X, \mathcal{D})$ and $\prod_{i \in X} supp(\{i\}, \mathcal{D}') > \prod_{i \in X} supp(\{i\}, \mathcal{D})$ because $freq(\{j\}, \mathcal{D}') > freq(\{j\}, \mathcal{D}')$ and $j \in X$. Thus, we have $lift(X, \mathcal{D}') = \frac{supp(X, \mathcal{D}')}{\prod_{i \in X} supp(\{i\}, \mathcal{D}')} < lift(X, \mathcal{D}) = \frac{supp(X, \mathcal{D})}{\prod_{i \in X} supp(\{i\}, \mathcal{D})}$, which shows that $Y = \{j\} \subset X$ is an opponent of $X$ for the lift measure (and not a supporter). □

We now consider the case of a condensed representation selector, and prove Property 2, which defines the supporters and opponents of the free constraint.

*Property 2.* For all itemsets $X$ such that $|X| > 1$, $free^+(X) = \{X \setminus \{i\} : i \in X\}$ and $free^-(X) = \{X\}$.

*Proof.* Let $X$ be an itemset such that $|X| > 1$. We first show that $X^{\perp} \subseteq free^+(X)$, i.e. that for all $k \in X$, $Y = X \setminus \{k\} \in free^+(X)$. By definition, we have to find two datasets $\mathcal{D}$ and $\mathcal{D}'$ such that $\mathcal{D}' >_Y \mathcal{D}$, $free(X, \mathcal{D}) = false$, whereas $free(X, \mathcal{D}') = true$. Let $\mathcal{D} = \{X\} \cup \{X \setminus \{i\} : i \in Y\} \cup \mathcal{D}_Y^-$ and $\mathcal{D}' = \{X\} \cup \{X \setminus \{i\} : i \in Y\} \cup \mathcal{D}_Y^+$. First, it is easy to that $\mathcal{D}' >_Y \mathcal{D}$. Moreover, we have $freq(X, \mathcal{D}) = 1$ and $freq(Y, \mathcal{D}) = 1$ since $Y \subseteq X$ and $Y \notin \mathcal{D}_Y^-$. Thus, $X$ is not a free itemset in $\mathcal{D}$, i.e. $free(X, \mathcal{D}) = false$. Then, we can see that $freq(X, \mathcal{D}') = 1$ and $freq(X \setminus \{i\}, \mathcal{D}') = 2$ for all $i \in X$ (in particular, note that $Y = (X \setminus \{k\} \in \mathcal{D}_Y^+)$. Thus, $X$ is a free itemset in $\mathcal{D}'$, i.e. $free(X, \mathcal{D}) = true$, which completes the proof that $Y = (X \setminus \{k\}) \in free^+(X)$.

We now show that $X \in free^-(X)$. We have to find two datasets $\mathcal{D}$ and $\mathcal{D}'$ such that $\mathcal{D}' >_X \mathcal{D}$, $free(X, \mathcal{D}) = true$, whereas $free(X, \mathcal{D}') = false$. Let $\mathcal{D} = \{X\} \cup \mathcal{D}_X^-$ and $\mathcal{D}' = \{X\} \cup \mathcal{D}_X^+$. By construction (see the definitions of $\mathcal{D}_X^-$ and $\mathcal{D}_X^+$ in the proof of Lemma 1), it is clear that $\mathcal{D}' >_X \mathcal{D}$. Moreover, we have $freq(X, \mathcal{D}) = 1$ and $freq(X \setminus \{k\}, \mathcal{D}) = 2$ for all $k \in X$ (because $X \setminus \{k\} \subseteq X \in \mathcal{D}$, and $X \setminus \{k\} \in \mathcal{D}_X^-$). Therefore, $X$ is a free itemset in $\mathcal{D}$, i.e. $free(X, \mathcal{D}) = true$. Then, we can also check that $freq(X, \mathcal{D}') = 2$ and $freq(X \setminus \{k\}, \mathcal{D}) = 2$ for all $k \in X$. Thus, $X$ is not a free itemset in $\mathcal{D}'$, which completes the proof that $X \in free^-(X)$.

To complete the proof, we have to show that any other pattern $Y \notin \{X\} \cup X^{\perp}$ cannot be a supporter or an opponent of $X$. In particular, we have to show that for all $Y \subset X$, if $Y \notin X^{\perp}$, then $Y \notin free^+(X)$, i.e. that for all databases $\mathcal{D}$ and $\mathcal{D}'$ such that $\mathcal{D}' >_Y \mathcal{D}$, it is impossible to have $free(X, \mathcal{D}) = false$ and $free(X, \mathcal{D}') = true$. If $free(X, \mathcal{D}) = false$, it means that there exists $k \in X$ such that $freq(X, \mathcal{D}) = freq(X \setminus \{k\}, \mathcal{D})$. Moreover, because $\mathcal{D}' >_Y \mathcal{D}$ and $Y \notin X^{\perp}$, we have $freq(X, \mathcal{D}') = freq(X \setminus \{k\}, \mathcal{D}')$, which shows that $X$ cannot be free in $\mathcal{D}'$, i.e. $free(X, \mathcal{D}') = false$, and contradicts the hypothesis. Thus, we have shown that only the direct subsets of $X$ are supporters of $X$ for the free constraint. The rest of the proof is omitted for lack of space.

To conclude this section, we stress that the strength of the concept of supporters and opponents is to clearly identify the patterns actually involved in the evaluation of a selector. For instance, whereas the definition of free itemsets

given in Table 1 involves *all strict subsets* of $X$ (with $\forall Y \subset X$), we can see that *only direct subsets* of $X$ are supporters. In the following sections, we show how supporters and opponents can be used to compare selectors (see Section 5), and how the number of supporters and opponents of a selector is related to its effectiveness to select interesting patterns (see Section 6).

## 5 Typology of Interestingness Selectors

### 5.1 Polarity of interestingness selectors

We distinguish two broad categories of selectors according to whether they aim at discovering over-represented phenomena in the data (e.g., positive correlation) or under-represented phenomena in the data (e.g., outlier detection). Naturally, the characterization of these categories is related to the evaluation of the frequency on the pattern to assess. For instance, it is well-known that the interestingness of a pattern $X$ increases with its frequency for finding correlations between items. In order that an interestingness selector $s$ will be sensitive to this variation, it is essential that $s$ increases with the frequency of $X$. This principle has first been proposed for association rules [23] (Property P2) and after, extended to correlated itemsets [15, 27]. Conversely, a selector for outlier detection will favor patterns whose frequency decreases. Indeed, a pattern is more likely to be abnormal as it is not representative of the dataset i.e., its frequency is low.

We formalize these two types of patterns thanks to reflexivity property:

**Definition 4 (Positive and negative reflexive).** *An interestingness selector $s$ is positive (resp. negative) reflexive iff any pattern is its own supporter i.e., $(\forall X \in \mathcal{L})(X \in s^+(X))$ (resp. opponent i.e., $(\forall X \in \mathcal{L})(X \in s^-(X))$).*

As $\textit{all-conf}^+(X) = \{X\}$, the all-confidence selector is positive reflexive. Conversely, the free selector is negative reflexive because $free^-(X) = \{X\}$ (when frequency of $X$ increases, $X$ is less likely to be free because its frequency becomes closer to that of its subsets).

This clear separation based on reflexive property constitutes the first analysis axis of our selector typology. Table 4 schematizes this typology where the polarity is the vertical axis of analysis. The horizontal axis (semantics) will be described in the next section. Note that the correlation measures and the closed itemsets are in the same column. Several works in the literature have shown that closed itemsets maximize classification measures [11] and correlation measures [10]. For instance, the lift of a closed pattern has the highest value of its equivalence class because the frequency of X remains the same (numerator) while the denominator decreases.

Of course, it should not be possible for an interestingness selector to both isolate over-represented phenomena (i.e., positive) and under-represented phenomena (i.e., negative). For this reason, a selector should never be both positive and negative. Besides, the behavior of an interestingness selector is easier to

| Semantics | Polarity: Positive<br>$X \in s^+(X)$ | Polarity: Negative<br>$X \in s^-(X)$ |
|---|---|---|
| **Subsets**<br>$X^{\downarrow}$ | C1: $X^{\downarrow} \cap s^-(X) \neq \emptyset$<br>(all-confidence, bond, productive itemsets, NDI, FPOF, lift) | C2: $X^{\downarrow} \cap s^+(X) \neq \emptyset$<br>(free itemset, NDI, negative border) |
| **Supersets**<br>$X^{\uparrow}$ | C3: $X^{\uparrow} \cap s^-(X) \neq \emptyset$<br>(closed itemsets, maximal itemsets) | $X^{\uparrow} \cap s^+(X) \neq \emptyset$<br>(FPOF) |
| **Incomparable sets**<br>$X^{\leftrightarrow}$ | $X^{\leftrightarrow} \cap s^-(X) \neq \emptyset$<br>(top-k frequent itemsets) | $X^{\leftrightarrow} \cap s^+(X) \neq \emptyset$<br>(FPOF) |

*QC 2: Completemes* $s^+(X) \cup s^-(X) = \mathcal{L}$

*QC 1: Soundness* $s^+(X) \cap s^-(X) = \emptyset$

**Table 4.** Typology of interestingness selectors

understand for the end user if the change in frequency of a pattern $Y$ still impacts $s(X)$ in the same way. In other words, the increase of $freq(X)$ should not decrease $s(X)$ in some cases and increase $s(X)$ in others.

**Quality Criterion 1 (Soundness)** *An interestingness selector $s$ is sound iff no pattern is at the same time a supporter and an opponent of another pattern:* $\forall X \in \mathcal{L}, s^+(X) \cap s^-(X) = \emptyset$.

When Quality Criterion 1 is violated, it makes difficult to interpret a mined pattern. For instance, frequent free itemset mining is not sound. There are two opposite reasons for explaining that a pattern is not extracted: its frequency is too low (non-frequent rejection), or its frequency is too high (non-free rejection). Conversely, for frequent closed patterns, a pattern is not extracted if and only if its frequency is too low (whatever the underlying cause: the pattern is not frequent or non-closed). It means that frequent closed pattern mining is sound. We therefore think that the violation of Quality Criterion 1 (where $s^+(X) = s^-(X) = X^{\downarrow}$) could partly explain the failure of NDI (non-derivable itemsets) even if they form an extremely compact condensed representation.

**Recommendation:** A well-behaving pattern mining method should not mix interestingness selectors with opposite polarities or make possible the existence of patterns that are supporters and opponents of the same pattern.

Before describing the semantics axis of our typology, Table 4 classifies all the selectors presented in Table 1. As expected, all selectors seeking to isolate over-represented phenomena are in the Positive column.

### 5.2 Semantics of interestingness selectors

This section presents three complementary criteria to identify the nature of an interestingness selector. The key idea is to focus on the relationships between patterns to qualify the semantics of the selector. More precisely, the meaning of a positive selector (whose primary objective is to find over-represented patterns) depends strongly on the set of opponents that can lead to the rejection of the assessed pattern. Conversely, a negative reflexive selector relies often on supporters to better isolate under-represented phenomena. For this reason, the positive (resp. negative) column of Table 4 involves opponents $s^-(X)$ (resp. supporters $s^+(X)$).

Furthermore, for two selectors of the same polarity, it is possible to distinguish their goals (e.g., correlation or condensed representation) according to the opponents/supporters that they involve. Thus, we break down the semantics axis into three parts: subsets $X^{\downarrow} = \{Y \subset X\}$, supersets $X^{\uparrow} = \{Y \supset X\}$ and incomparable sets $X^{\leftrightarrow} = \{Y \in \mathcal{L} : Y \not\subseteq X \wedge Y \not\supseteq X\}$. This decomposition of the lattice of the opponents and the lattice of the supporters is useful to redefine coherent classes of usual selectors (these classes are indicated in Table 4):

**Definition 5 (Selector classes).** *An interestingness selector $s$ belongs to:*

- **C1 (Positive correlation)** *iff* $(\forall X \in \mathcal{L})(X^{\downarrow} \cap s^-(X) \neq \emptyset)$
- **C2 (Minimal condensed representation)** *iff* $(\forall X \in \mathcal{L})(X^{\downarrow} \cap s^+(X) \neq \emptyset)$
- **C3 (Maximal condensed representation)** *iff* $(\forall X \in \mathcal{L})(X^{\uparrow} \cap s^-(X) \neq \emptyset)$

Intuitively a pattern is a set of correlated items (or correlated in brief) when its frequency is higher than what was expected by considering the frequency of some of its subsets (this set of opponents varies depending on the statistical model). This means that the increase of the frequency of one of these subsets may lead to the rejection of the assessed pattern. In other words, a correlation measure is based on subsets as opponents. This observation has already been made in the literature for association rules [23] (with Property P3) and itemsets [15, 27]. Table 4 shows that most of correlation measures in the literature satisfy $(\forall X \in \mathcal{L})(X^{\downarrow} \cap s^-(X) \neq \emptyset)$. The extraction of NDI, classified as a condensed representation, also meets this criterion. It is intriguingly since the NDI selector is not usually used as a correlation measure.

A condensed representation is a reduced collection of patterns that can regenerate some properties of the full collection of patterns. Typically, frequent closed patterns enable to retrieve the exact frequency of any frequent pattern. Most approaches are based on the notion of equivalence class where two patterns are equivalent if they have the same value for a function $f$ and if they are comparable. The equality for $f$ and the comparability result in an interrelation between the assessed pattern and its subsets/supersets. Class C3 (i.e., maximal condensed representations) includes the measures that remove the assessed pattern when a more specific pattern provides more information. Closed patterns and maximal patterns satisfy this criterion: $(\forall X \in \mathcal{L})(X^{\uparrow} \cap s^-(X) \neq \emptyset)$. Minimal condensed representations are in the dual class (i.e., Class C2).

Unlike the polarity that opposes two types of irreconcilable patterns, the three parts of the semantics axis (i.e., subsets, supersets and incomparable sets) are simultaneously satisfiable. We think that an ideal pattern extraction method should always belong to these three parts:

**Quality Criterion 2 (Completeness)** *A selector $s$ is complete iff all patterns are either supporter or opponent:* $\forall X \in \mathcal{L}, s^+(X) \cup s^-(X) = \mathcal{L}$.

Let us illustrate the principle behind this quality criterion by considering an ideal pattern mining method that isolates correlations. Of course, this method relies on a selector $s$ that belongs to the class of correlations (for example, the lift). At equal frequency, the longer pattern will be preferred because it will maximize lift. This property corresponds to the criterion $X^{\uparrow} \cap s^-(X) \neq \emptyset$. At this stage, two incomparable patterns can cover the same set of transactions. To retain only one, we must add a new selection criterion that verifies the criterion $X^{\leftrightarrow} \cap s^-(X) \neq \emptyset$. This approach is at the heart of many proposals in the literature [28, 10, 3]: (i) use of a correlation measure, (ii) elimination of non-closed patterns, (iii) elimination of incomparable redundant patterns.

**Recommendation:** All patterns should be either supporters or opponents in a well-behaving pattern mining method. It is often necessary to combine a measure with local and global redundancy reduction techniques.

## 6 Evaluation Complexity of Interestingness Selectors

As Quality Criterion 2 is often violated, we propose to measure its degree of satisfaction to evaluate and compare interestingness selectors. More precisely, we measure the quality of an interestingness selector considering its degree of satisfaction of the semantics criterion. Let us consider the correlation family, it is clear that to detect correlations, support is a poorer measure than lift which is itself less effective than productivity. Whatever the part of the lattice, the more numerous the opponents/supporters of a selector, the better its quality. In other words, a selector is more effective to assess the interestingness of a pattern $X$ when the number of supporters and opponents of $X$ is very high.

**Definition 6 (Evaluation complexity).** *The evaluation complexity of an interestingness selector $s$ is the asymptotic behavior of the cardinality of its supporters/opponents.*

The evaluation complexity of a selector usually depends on the cardinality of the assessed pattern (denoted by $k = |X|$) and the cardinality of the set of items $\mathcal{I}$ (denoted by $n = |\mathcal{I}|$). For instance, $|\textit{all-conf}^{\pm}(X)| = |\textit{all-conf}^{+}(X)| \cup |\textit{all-conf}^{-}(X)| = 1 + k$. Therefore, the behavior of the number of evaluations of all-confidence is linear with respect to itemset size. Similarly, the evaluation complexity of productive itemsets is exponential with respect to the size of the assessed pattern since all subsets are involved in the evaluation of this constraint. According to the evaluation complexity, we say that the quality of the

constraint of productive itemsets is better than that of the all-confidence, because the opponents are more numerous. More generally, this complexity allows to compare several interestingness selectors to each other. The column $|s^{\pm}(X)|$ (where $s^{\pm}(X)$ is the total number of supporters and opponents of $X$) in Table 3 indicates the evaluation complexity of each measure or constraint defined in Table 1. Three main complexity classes emerge: constant, linear and exponential. Although Table 1 is an extremely small sample of measures, we observe that the evaluation complexity of pattern mining methods has increased over the past decades. Interestingly, we also note that the evaluation complexity of global constraints [6, 13] (or advanced measures [26]) is greater than those of local constraints (or absolute measures).

For Classes C2 and C3, the most condensed representations (among those that enable to regenerate the frequency of each pattern) are also those with the greatest evaluation complexity. Indeed, the free itemsets are more numerous than the closed ones, themselves more numerous than the NDIs. For Class C1, it is clear that measures based on more sophisticated statistical models require more relationships [26]. They have therefore an higher evaluation complexity. We will also experimentally verify this hypothesis in the next section.

## 7 Experimental Study

Our goal is to verify whether the quality of the correlated pattern selectors follow the evaluation complexity. In other words, if a correlation measure has a greater evaluation complexity than another measure, it is expected to be more effective.

To verify this hypothesis, we rely on the experimental protocol inspired by [14]. The idea is to compare the extracted patterns in an original dataset $\mathcal{D}$ with the same randomized dataset $\mathcal{D}^*$. Specifically, in the randomized dataset $\mathcal{D}^*$, a large number of items are randomly swapped two by two in order to clear any correlation. Nevertheless, this dataset $\mathcal{D}^*$ retains the same characteristics (transaction length and frequency of each item). So, if a pattern $X$ extracted in the original dataset $\mathcal{D}$ is also extracted in the randomized dataset $\mathcal{D}^*$, $X$ is said to be false positive (FP). Its presence in $\mathcal{D}^*$ is not due to the correlation between items but due to the distribution of items in data. Then we evaluate for each selector how many false positive patterns are extracted on average by repeating the protocol on 10 randomized datasets. Experiments were conducted on datasets coming from the UCI ML Repository [7]. Given a minimum support threshold, we compare 4 selectors: **Support** (all frequent patterns); **All-confidence** (all frequent patterns having at least 5 as all-confidence); **Lift** (all frequent patterns having at least 1.5 as lift); and **Productivity** (all frequent patterns having at least 1.5 as productivity).

Even if arbitrary thresholds are used for the last three selectors, the results are approximately the same with other thresholds because we use the FP rate as evaluation measure. This normalized measure is a ratio, it returns the proportion of FP patterns among all the mined patterns.

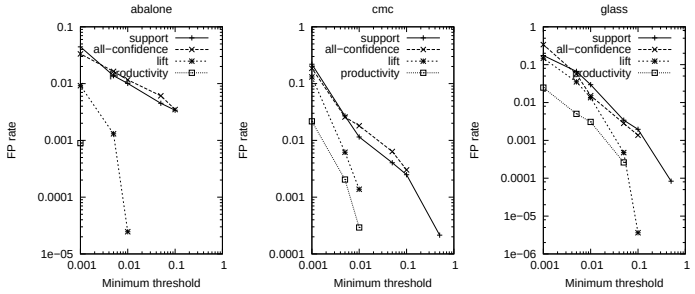



**Fig. 1.** FP rate with minimum support threshold

Figure 1 plots the FP rate of each selector on `abalone`, `cmc` and `glass`. Since lift and productivity measures sometimes do not return FP patterns, there are missing points because the scale is logarithmic. For each dataset, we observe that the FP rate increases when the minimum support threshold decreases regardless of the measure. The evolution of the FP rate for the all-confidence is very similar to that of the support even if the all-confidence has a greater complexity in evaluation. For the other selectors, there is a clear ranking from worst to best: support, lift and productivity. This ranking also corresponds to the classes of complexity from worst to best: constant, linear, exponential. Our framework could be refined to consider a set of patterns as an opponent (or a supporter). Then, the relation between a pattern and its supporters (or opponents) would become a relation between a pattern and a *set* of supporters (or opponents). That would make possible to capture refinements between selectors as follows: the all-confidence depends only on one item at once (due to the maximum of its denominator) whereas the lift can vary according to a set of items (due to the multiplication). Nevertheless, our experiments show that on the whole the correlation measures with the highest evaluation complexity are also the best ones according to the FP rate.

## 8 Conclusion and Discussion

In this paper, we have addressed the question of the quality of a data mining method in the context of unsupervised problems with binary data. A key concept is to study the relationships between a pattern $X$ and the other patterns when $X$ is selected by a method. These relationships are formalized through the notions of supporters and opponents. We have presented a typology of methods defined by formal properties and based on two complementary criteria. This typology offers a global picture and a methodology helping to compare methods to each other. Besides, if a new method is proposed, its quality can be immediately compared

to the quality of the other methods according to our framework. Finally, the quality of a method is quantified via an evaluation complexity analysis based on the number of supporters and opponents of a pattern extracted by the method.

Two recommendations can be drawn from this work. We think that the result of a data mining operation should be understandable by the user. So, our first recommendation is a data mining method should not simultaneously extract over-represented phenomena and under-represented phenomena because mixing these two kinds of phenomena obstructs the understandability of the extracted patterns. This recommendation is formally defined by our soundness criterion. Most of methods satisfy this property, but there are a few exceptions such as the constraints extracting NDI and frequent free patterns. The violation of this recommendation might explain why these patterns are of little use.

Another recommendation is a data mining method should extract patterns for which all patterns contribute to the quality of an extracted pattern. This recommendation is formalized by our completeness criterion stating that all patterns must be either supporters or opponents of a pattern extracted by a method. In practice, this recommendation is not satisfied by a lot of methods. However, a few methods are endowed with this behavior, such as [3, 10, 28] in the context of the correlations. We think that a goal of pattern mining should be to design methods following this recommendation which is more often reached by pattern sets [3] as illustrated by the previous examples [3, 10, 28].

A perspective of this work is to study an interestingness theory for methods producing pattern sets. Pattern sets are a promising avenue since the interestingness of a pattern also depends on the interestingness of the other patterns of the pattern set, thus providing a global quality of the method. Finally, it is important to note that our framework can be generalized to other pattern langages (sequence, graph, etc.) and other basic functions, e.g., observing the variation of support in a target class would extend our approach to the supervised context.

*Acknowledgements.* The authors thank Albrecht Zimmermann for highly valuable discussions. This work has been partly supported by the QCM-BioChem project (CNRS Mastodons).

## References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: VLDB. pp. 487–499. Morgan Kaufmann (1994)
2. Boulicaut, J.F., Bykowski, A., Rigotti, C.: Free-sets: A condensed representation of boolean data for the approximation of frequency queries. Data Min. Knowl. Discov. 7(1), 5–22 (2003)
3. Bringmann, B., Zimmermann, A.: The chosen few: On identifying valuable patterns. In: ICDM. pp. 63–72. Omaha, NE (2007)
4. Calders, T., Goethals, B.: Non-derivable itemset mining. Data Min. Knowl. Discov. 14(1), 171–206 (2007)
5. Calders, T., Rigotti, C., Boulicaut, J.F.: A survey on condensed representations for frequent sets. In: European Workshop on Inductive Databases and Constraint Based Mining. LNCS, vol. 3848, pp. 64–80. Springer (2004)

6. Crémilleux, B., Soulet, A.: Discovering knowledge from local patterns with global constraints. In: ICCSA. pp. 1242–1257. LNCS, Springer (2008)
7. Dheeru, D., Karra Taniskidou, E.: UCI machine learning repository (2017), http://archive.ics.uci.edu/ml
8. Fayyad, U.M., Piatetsky-Shapiro, G., Uthurusamy, R.: Summary from the kdd-03 panel: data mining: the next 10 years. ACM SIGKDD Explorations 5(2) (2003)
9. Fu, A., W., R., Kwong, W., Tang, J.: Mining $n$-most interesting itemsets. In: ISMIS. LNCS, vol. 1932, pp. 59–67. Springer, Charlotte, NC, USA (2000)
10. Gallo, A., De Bie, T., Cristianini, N.: MINI: Mining informative non-redundant itemsets. In: PKDD. pp. 438–445. LNCS, Springer (2007)
11. Garriga, G.C., Kralj, P., Lavrač, N.: Closed sets for labeled data. Journal of Machine Learning Research 9(Apr), 559–580 (2008)
12. Geng, L., Hamilton, H.J.: Interestingness measures for data mining: A survey. ACM Comput. Surv. 38(3) (2006), http://doi.acm.org/10.1145/1132960.1132963
13. Giacometti, A., Marcel, P., Soulet, A.: A relational view of pattern discovery. In: DASFAA. pp. 153–167. Lecture Notes in Computer Science, Springer (2011)
14. Gionis, A., Mannila, H., Mielikäinen, T., Tsaparas, P.: Assessing data mining results via swap randomization. TKDD 1(3), 14 (2007)
15. Hämäläinen, W.: Efficient search for statistically significant dependency rules in binary data. Ph.D. thesis, University of Helsinki (2010)
16. Han, J., Cheng, H., Xin, D., Yan, X.: Frequent pattern mining: current status and future directions. Data Mining and Knowledge Discovery 15(1), 55–86 (2007)
17. He, Z., Xu, X., Huang, Z.J., Deng, S.: FP-outlier: Frequent pattern based outlier detection. Computer Science and Information Systems 2(1), 103–118 (2005)
18. Lenca, P., Meyer, P., Vaillant, B., Lallich, S.: On selecting interestingness measures for association rules: User oriented description and multiple criteria decision aid. EJOR 184(2), 610–626 (2008)
19. Mannila, H., Toivonen, H.: Levelwise search and borders of theories in knowledge discovery. Data Min. Knowl. Discov. 1(3), 241–258 (1997)
20. Morik, K., Boulicaut, J., Siebes, A. (eds.): Local Pattern Detection, International Seminar, Lecture Notes in Computer Science, vol. 3539. Springer (2005)
21. Omiecinski, E.: Alternative interest measures for mining associations in databases. IEEE Trans. Knowl. Data Eng. 15(1), 57–69 (2003)
22. Pasquier, N., Bastide, Y., Taouil, R., Lakhal, L.: Efficient mining of association rules using closed itemset lattices. Inf. Syst. 24(1), 25–46 (1999)
23. Piatetsky-Shapiro, G.: Discovery, analysis, and presentation of strong rules. In: Knowledge Discovery in Databases, pp. 229–248. AAAI/MIT Press (1991)
24. Tan, P.N., Kumar, V., Srivastava, J.: Selecting the right objective measure for association analysis. Inf. Syst. 29(4), 293–313 (2004)
25. Tew, C.V., Giraud-Carrier, C.G., Tanner, K.W., Burton, S.H.: Behavior-based clustering and analysis of interestingness measures for association rule mining. Data Min. Knowl. Discov. 28(4), 1004–1045 (2014)
26. Vreeken, J., Tatti, N.: Interesting patterns. In: Aggarwal, C.C., Han, J. (eds.) Frequent Pattern Mining, pp. 105–134. Springer International Publishing (2014)
27. Webb, G.I., Vreeken, J.: Efficient discovery of the most interesting associations. ACM Trans. Knowl. Discov. Data 8(3), 15:1–15:31 (Jun 2013)
28. Xin, D., Cheng, H., Yan, X., Han, J.: Extracting redundancy-aware top-k patterns. In: KDD. pp. 444–453. ACM (2006)
29. Zimmermann, A.: Objectively evaluating condensed representations and interestingness measures for frequent itemset mining. J. Intell. Inf. Syst. 45(3), 299–317 (2015)

# Sequential Pattern Sampling with Norm Constraints

Lamine Diop*, Cheikh Talibouya Diop*, Arnaud Giacometti†, Dominique Li† and Arnaud Soulet†
*University Gaston-Berger of Saint-Louis, Senegal, {diop.lamine3;cheikh-talibouya.diop}@ugb.edu.sn
†University of Tours, France, firstname.lastname@univ-tours.fr

***Abstract*—In recent years, the field of pattern mining has shifted to user-centered methods. In such a context, it is necessary to have a tight coupling between the system and the user where mining techniques provide results at any time or within a short response time of only few seconds. Pattern sampling is a non-exhaustive method for instantly discovering relevant patterns that ensures a good interactivity while providing strong statistical guarantees due to its random nature. Curiously, such an approach investigated for itemsets and subgraphs has not yet been applied to sequential patterns, which are useful for a wide range of mining tasks and application fields. In this paper, we propose the first method for sequential pattern sampling. In addition to address sequential data, the originality of our approach is to introduce a constraint on the norm to control the length of the drawn patterns and to avoid the pitfall of the "long tail" where the rarest patterns flood the user. We propose a new constrained two-step random procedure, named CSSAMPLING, that randomly draws sequential patterns according to frequency with an interval constraint on the norm. We demonstrate that this method performs an exact sampling. Moreover, despite the use of rejection sampling, the experimental study shows that CSSAMPLING remains efficient and the constraint helps to draw general patterns of the "head". We also illustrate how to benefit from these sampled patterns to instantly build an associative classifier dedicated to sequences. This classification approach rivals state of the art proposals showing the interest of constrained sequential pattern sampling.**

***Keywords*—Pattern Mining, Pattern Sampling, Sequential Data**

## I. INTRODUCTION

In recent years, the field of pattern mining has shifted to user-centered methods [1]. Typically, the idea is to be able to capture the feedback of the user during the analysis of the first mined patterns to better choose the next ones. To guarantee this tight coupling between the system and the user, it is then necessary to use techniques that provide results at any time [2] or within a short response time of only few seconds. Pattern sampling is an efficient approach that instantly returns patterns [3], [4], [5], which enables to produce pattern-based models at any time [6]. Introduced in [7], pattern sampling returns a small set of patterns randomly drawn with a probability proportional to an interestingness measure specified by the user. For instance, with frequency, a pattern twice as frequent will be twice as likely to be picked. Sampling methods are particularly efficient and have the advantage of returning patterns with high diversity. To the best of our knowledge, there is no work addressing pattern sampling in sequential data [8]. Yet sequential pattern mining is useful for a wide range of mining tasks and application fields [9] such as web usage mining, text mining, fraud detection and so on.

Unfortunately, a naive pattern sampling according to frequency is not relevant for sequential data because of the pitfall of the long tail. In statistics and business, the long tail of a distribution is its portion having a large number of occurrences far from the central part of the distribution [10]. In our context, the long tail designates the long and rare sequential patterns far more numerous than the short and frequent ones (the "head"). As a result, it is nearly impossible to draw the most general patterns despite the bias of the frequency. This problem is stronger with sequential data than with transactional data because the number of sub-patterns in a sequence is much higher than that in an itemset of the same length. Figure 1 illustrates the long tail problem on the toy dataset provided in Section III. The top histogram shows the frequency of the 35 patterns of the toy dataset (i.e., bars in dark and light grays). We observe that 23 patterns have a frequency of 1 (the tail). Consequently, the bars in dark gray of the bottom histogram show that 39.6% of the patterns drawn according to frequency belong to this tail (with a frequency of only 1). The real-world datasets reveal even much more problematic situations (see the experimental study in Section V). For instance, each of the 10,000 patterns drawn randomly according to frequency on `bms` dataset appears only in a single sequence of the dataset. Of course, these patterns are useless because they correspond more to noise than true patterns describing the data.

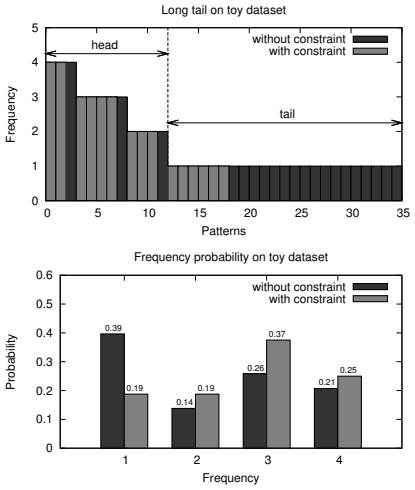



Fig. 1. Impact of the long tail on frequent sequential pattern sampling

To circumvent the pitfall of the long tail, we propose to sample patterns under a constraint on the maximum norm (maximum number of items). This constraint will prevent drawing too specific patterns because too long, but interest-

ingly, still allow to draw non-frequent patterns that describe sequences of rare events. It is really crucial not to force a minimal frequency in order to have a description of rare objects [6]. In Figure 1, a maximum norm constraint of 2 removes all dark gray patterns. Interestingly, much of the tail is cut off. As a result, the bottom histogram shows a significant increase in the probability to draw patterns having frequencies ranging from 2 to 4. Indeed, the probability to draw a pattern with a frequency of 1 has been divided by 2 (the first bar in light gray). To achieve this goal, we would like to use the two-step random procedure [11] which is the most efficient pattern sampling approach in the literature. After a preprocessing phase, this method extracts an exact sample of patterns without rejection. However, extending this approach to sequential patterns is a challenging problem. Indeed, its core requires counting the number of distinct subsequences for each sequence. This task is not easy because a sequence may contain several occurrences of the same subsequence and we want to consider only subsequences of a certain length.

The main contributions of the paper are as follows:

- We propose a new algorithm named CSSAMPLING (Constrained Subsequence Sampling) that samples sequential patterns proportionally to frequency with an interval constraint on the norm. It relies on a constrained two-step random procedure that requires solving two sub-problems: (i) counting the number of distinct subsequences having a maximum norm and (ii) uniformly drawing subsequences. We demonstrate that CSSAMPLING performs an exact sequential pattern sampling according to frequency, and we analyze its complexity on average.
- We present a large set of experimental results for analyzing the behavior of CSSAMPLING. We show on several datasets that our approach is efficient enough to return hundreds of sequential patterns per second. We also highlight the practical interest of norm constraints to better control the quality of the returned patterns and avoid the curse of the long tail.
- Sequence classification is a crucial data mining task useful in a wide range of applications. We investigate how sequential pattern sampling lead to build associative classifiers for sequences. Interestingly, the accuracy of these sample-based classifiers built in a short response time is comparable to that of the methods of the state of the art. Experiments show that it is again essential to use a constraint to draw general patterns contained in the head, and not in the tail.

The outline of this paper is as follows. Section II reviews some related work about pattern sampling methods. Section III introduces basic definitions and the formal problem statement. We present our constrained two-step random procedure for sequential pattern sampling in Section IV. We evaluate our approach in Section V and conclude in Section VI.

## II. RELATED WORK

*a) Instant discovery of sequential patterns:* Sequential pattern mining has been introduced by [8] two decades ago and its usefulness has been widely proved as mentioned in introduction. Since 1995, many methods have optimized the mining of sequential patterns [12], [13], [14] and have introduced variants with constraints [15], [16] or condensed representations [17], [18]. Despite all these advances, sequential pattern mining remains a costly task that often generates too many redundant patterns. Consequently, it is not possible to discover patterns or to build pattern-based models in a short response time. This limit, also reached by other language (e.g., itemset), was circumvented by Monte Carlo tree search [19] or pattern sampling [7]. This kind of instantaneous methods is at the core of many approaches that makes data mining more interactive [3], [4], [5], [6]. But to the best of our knowledge, all these methods have not been applied to sequential patterns. The rest of the related work is devoted to the pattern sampling techniques, which corresponds to our proposal.

*b) Output space sampling:* Importantly, it is necessary to distinguish between input and output space sampling. The *input space* sampling [20] consists in generating from a sample of data all the patterns that would have been mined from the complete dataset. The *output space* sampling [7] consists in generating a sample of patterns among the patterns that would have been mined from the complete dataset. More formally, pattern sampling [7], [11] aims at accessing the pattern space $\mathcal{L}$ by an efficient sampling procedure simulating a distribution $\pi : \mathcal{L} \rightarrow [0, 1]$ that is defined with respect to some interestingness measure $f$, i.e., $\pi(.) = f(.)/Z$ where $Z$ is a normalizing constant. As the pattern language is fully addressed proportionally to $f$, this approach guarantees a good variety of patterns returned to the user unlike heuristic approaches. Several approaches have been proposed for input space sampling of sequential patterns [21], [22], but to the best of our knowledge, this paper proposes the first approach to output space sampling of sequential patterns. Since the complexity of pattern sampling is independent of the language size, it is suitable for structured languages where there is a combinatorial explosion of the number of patterns like subgraphs [23] and even for infinite languages like numerical data [24]. Note that in this paper, we restrict ourselves to frequency as interestingness measure $f$ because we focus more on sequence-specific and constraint-specific issues. It would be natural to extend our approach to other measures (e.g., area or discriminative measures) as done in [11].

*c) Pattern sampling techniques:* Several procedures have been proposed for the output space sampling of patterns. The first kind of procedure [23], [25] randomly draws a pattern from the search space using a heuristic to favor the patterns that are most relevant according to the interestingness measure $f$. In practice, these methods return interesting patterns but they offer no guarantee on the quality of the outputted sample. The second kind of procedure [7], [3], [26] is based on Markov chain Monte Carlo algorithms. The idea is that the equilibrium distribution of a random walk corresponds to the desired probability distribution. The limit of such stochastic methods is the convergence speed, which may be slow. The third kind of procedure [11], [24], [27] consists in drawing an instance of the dataset and then drawing a pattern contained in this instance. By judiciously selecting the two draw distributions, it is possible to obtain an exact sampling according to the desired final distribution. Recently, [24] adds a third step for taking into account numeric data where the pattern language is infinite. We opted for such a multi-step random procedure

for its speed and accuracy. Section IV-A underlines specific challenges for achieving this goal in the case of sequences.

Besides the inherent difficulty of addressing sequences rather than itemsets, we also add an interval constraint on the norm of the returned patterns. In the litterature, there are few proposals adding a binary predicate to restrict the sampling. [25] proposes a framework for sampling of *maximal* itemsets from transactional datasets, but it relies on a heuristic random walk with no guarantee. Based on the SAT framework, [28] requires to have a solver integrating efficiently XOR constraints and in practice, it has been implemented only for itemsets. In addition, the authors emphasize that the efficiency of this generic approach will hardly compete with approaches dedicated to a single language and/or class of constraints. In this paper, we propose an efficient method for integrating only constraints on the norm.

## III. Problem Statement

This section formalizes the problem of sequential pattern sampling under norm constraints. Before, we recall some preliminary definitions about sequences.

### A. Basic definitions

Let $\mathcal{I}$ be a finite set of literals called *items*. An *itemset* $X$ is a subset of $\mathcal{I}$. A *sequence* $s = \langle X_1 \dots X_n \rangle$ defined over $\mathcal{I}$ is an ordered list of non-empty itemsets $X_i \subseteq \mathcal{I}$ ($1 \leq i \leq n$, $n \in \mathbb{N}$). $n$ is the *size* of the sequence $s$ denoted by $|s|$. The *norm* of the sequence $s$, denoted by $\|s\|$, is the sum of the cardinality of all its itemsets, i.e. $\|s\| = \sum_{i=1}^{n} |X_i|$. In the following, $s^l$ denotes the prefix $\langle X_1 X_2 \dots X_l \rangle$ of $s$ ($0 \leq l \leq n$, $l \in \mathbb{N}$), $s^0$ being the empty sequence (represented by $\langle\rangle$) and $s[j] = X_j$ denotes the $j$-th itemset of $s$ ($1 \leq j \leq n$, $j \in \mathbb{N}$). Finally, we denote $\mathbb{S}$ the universal set of all the sequences defined over $\mathcal{I}$, and a sequential dataset $\mathcal{S}$ over $\mathcal{I}$ is a multi-set of sequences defined over $\mathcal{I}$. We recall the definitions of *subsequences* and of *occurrences* of a subsequence:

*Definition 1 (Subsequence):* A sequence $s' = \langle X'_1 \dots X'_m \rangle$ is a subsequence of a sequence $s = \langle X_1 \dots X_n \rangle$, denoted by $s' \sqsubseteq s$, if there exists an index sequence $1 \leq i_1 < i_2 < \dots < i_m \leq n$ such that for all $j \in [1..m]$, one has $X'_j \subseteq X_{i_j}$. We denote $\phi(s)$ the set of subsequences of a sequence $s$, i.e. $\phi(s) = \{s' \in \mathbb{S} : s' \sqsubseteq s\}$, and $\Phi(s)$ its cardinality, i.e. $\Phi(s) = |\phi(s)|$.

*Example 1:* We use the sequential dataset $\mathcal{S}$ presented in Table I as a running example. This dataset contains 4 sequences $s_1$, $s_2$, $s_3$ and $s_4$ defined over the set of items $\mathcal{I} = \{a, b, c, d\}$. For example, the size of $s_1 = \langle (ab)c \rangle$ is equal to 2, i.e. $|s_1| = 2$, whereas its norm is equal to 3, i.e. $\|s_1\| = 2 + 1 = 3$. Moreover, we have $s_1^0 = \langle\rangle$, $s_1^1 = \langle (ab) \rangle$, $s_1^2 = s_1$, $s_1[1] = (ab)$ and $s_1[2] = c$. Finally, the set $\phi(s_1)$ of subsequences of $s_1$ is defined by $\phi(s_1) = \{\langle\rangle, \langle a \rangle, \langle b \rangle, \langle c \rangle, \langle (ab) \rangle, \langle ac \rangle, \langle bc \rangle, \langle (ab)c \rangle\}$. Thus, we have $\Phi(s_1) = 1+3+3+1 = 8$. The number of subsequences $\Phi(s_i)$ of all sequences $s_i \in \mathcal{S}$ is detailed in Table I. The notation $\Phi_{[m,M]}(s_i)$ is formally defined in the Section IV-A. Intuitively, it represents the number of subsequences of a sequence $s_i$ whose norm is between $m$ and $M$.

It is important to note that a subsequence $s' = \langle X'_1 \dots X'_m \rangle$ may occur several times in a sequence $s = \langle X_1 \dots X_n \rangle$ if there exist several index sequences $1 \leq i_1 < i_2 < \dots < i_m \leq n$ such that for all $j \in [1..m]$, one has $X'_j \subseteq X_{i_j}$. In that case, there are several *occurrences* of the subsequence $s'$ in $s$. The next definition explains how each occurrence is represented:

TABLE I. A sequential dataset $\mathcal{S}$ with 4 sequences

| Sid | Sequence of itemsets | #occurrences | $\Phi(s_i)$ | $\Phi_{[1,2]}(s_i)$ |
|---|---|---|---|---|
| $s_1$ | $\langle (ab)c \rangle$ | 8 | 8 | 6 |
| $s_2$ | $\langle (ab)c(ac) \rangle$ | 32 | 25 | 12 |
| $s_3$ | $\langle c(ac) \rangle$ | 8 | 7 | 5 |
| $s_4$ | $\langle (ab)(cd) \rangle$ | 16 | 16 | 10 |

*Definition 2 (Occurrence):* An ordered list of $n$ itemsets $o = \langle Z_1 \dots Z_n \rangle$ is an occurrence of a subsequence $s' = \langle X'_1 \dots X'_m \rangle$ in a sequence $s = \langle X_1 \dots X_n \rangle$ if there exists an index sequence $1 \leq i_1 < \dots < i_m \leq n$ such that for all $j \in \{i_1, \dots, i_m\}$, one has $Z_{i_j} = X'_j \subseteq X_{i_j}$, and for all $j \in [1..n] \setminus \{i_1, \dots, i_m\}$, one has $Z_j = \emptyset$. This index sequence, called signature of $o$, is unique by definition.

*Example 2:* For the sequence $s_2 = \langle (ab)c(ac) \rangle$, $o_1 = \langle (a)(c)\emptyset \rangle$ and $o_2 = \langle (a)\emptyset(c) \rangle$ are two occurrences of its subsequence $s'_2 = \langle (a)(c) \rangle$. Moreover, the index sequences $\langle 1, 2 \rangle$ and $\langle 1, 3 \rangle$ are the signatures of $o_1$ and $o_2$, respectively. In Table I, the number of occurrences of all its subsequences is given for each sequence (e.g., there are 32 occurrences for 25 distinct subsequences in $s_2$).

### B. Problem of sequential pattern sampling under constraint

A pattern sampling method aims at randomly drawing a pattern $X$ from a language $\mathcal{L}$ according to an interestingness measure $f$. $X \sim \pi(\mathcal{L})$ denotes such a pattern where $\pi(.) = f(.)/Z$ is a probability dristribution over $\mathcal{L}$. In our case, we focus on the frequency which is an intuitive interestingness measure for experts and is an essential atomic element to build many other interestingness measures (like area or discriminative measures):

*Definition 3 (Frequency):* The frequency of a subsequence $s \in \mathbb{S}$ in the sequential dataset $\mathcal{S}$, denoted by $freq(s, \mathcal{S})$, is defined by: $freq(s, \mathcal{S}) = |\{s' \in \mathcal{S} : s \sqsubseteq s'\}|$.

Our goal is to randomly draw sequential patterns according to frequency under norm constraints. Given two integers $m$ and $M$ such that $m \leq M$, we denote $\mathbb{S}_{[m,M]}$ the set of sequences of $\mathbb{S}$ whose norm is between $m$ and $M$, i.e. $\mathbb{S}_{[m,M]} = \{s \in \mathbb{S} : m \leq \|s\| \leq M\}$. The problem can finally be stated as follows:

**Given a sequential dataset $\mathcal{S}$, two integers $m$ and $M$, we aim at randomly drawing a subsequence $s \in \mathbb{S}_{[m,M]}$ with a probability distribution $P(s)$ proportional to its frequency in $\mathcal{S}$ i.e., $P(s) = \frac{freq(s,\mathcal{S})}{\sum_{s' \in \mathbb{S}_{[m,M]}} freq(s',\mathcal{S})}$.**

One of the advantages of frequent pattern sampling [11] is to remove the minimum frequency threshold (always difficult to set) while our problem introduces two thresholds: $m$ and $M$. Nevertheless, they are easier to set because their range is much smaller ([1..10] in our experiments) than that of the minimum threshold of frequency.

*Example 3:* Table II represents the set of all subsequences of sequences in $\mathcal{S}$ with a norm between $m = 1$ and $M = 2$, and gives the frequencies in $\mathcal{S}$ of all these subsequences.

TABLE II. SUBSEQUENCES IN $\mathbb{S}_{[1,2]}$ OF SEQUENCES IN $\mathcal{S}$

| **Pattern** $s$ | $freq(s,\mathcal{S})$ | **Pattern** $s$ | $freq(s,\mathcal{S})$ |
|---|---|---|---|
| $\langle a\rangle$ | 4 | $\langle ac\rangle$ | 3 |
| $\langle b\rangle$ | 3 | $\langle ad\rangle$ | 1 |
| $\langle c\rangle$ | 4 | $\langle ba\rangle$ | 1 |
| $\langle d\rangle$ | 1 | $\langle bc\rangle$ | 3 |
| $\langle(ab)\rangle$ | 3 | $\langle bd\rangle$ | 1 |
| $\langle(ac)\rangle$ | 2 | $\langle ca\rangle$ | 2 |
| $\langle(cd)\rangle$ | 1 | $\langle cc\rangle$ | 2 |
| $\langle aa\rangle$ | 1 | | |

For instance, because our problem is to draw a subsequence proportionally to its frequency, and $freq(\langle ac\rangle,\mathcal{S}) = 3 \times freq(\langle ba\rangle,\mathcal{S})$, our objective is to develop a sampling method such that the probability to draw the subsequence $\langle ac\rangle$ is three times greater than the probability to draw the subsequence $\langle ba\rangle$. But, even if the subsequence $\langle(ab)c\rangle$ has a frequency of 3, it will not be drawn because its norm is 3 ($> M$).

## IV. CONSTRAINED TWO-STEP RANDOM PROCEDURE

### A. Overview of the algorithm

To address the problem stated in the previous section, we propose to benefit from a two-step random procedure as done in [11] for sampling itemsets proportionally to their support. But, we constrain this random procedure to consider only the patterns whose norm is satisfactory at both step.

Given a dataset $\mathcal{S}$ and two integers $m$ and $M$ such that $m \leq M$, CSSAMPLING (Constrained Subsequence Sampling) returns a sequential pattern having a norm between $m$ and $M$:

*a) Step 1: Sampling a sequence:* In the first step (lines 1 and 2 of Algorithm 1), we start by counting for each sequence $s \in \mathcal{S}$ the number of subsequences having a norm between $m$ and $M$, i.e. $\Phi_{[m,M]}(s) = |\phi_{[m,M]}(s)|$ where $\phi_{[m,M]}(s) = \{s' \sqsubseteq s : m \leq \|s'\| \leq M\}$. To do this, we show in Section IV-B how to extend the formula given in [29]. Then, this first step continues with the drawing of a sequence $s$ from $\mathcal{S}$ proportionally to its weight $w(s) = \Phi_{[m,M]}(s)$. For instance, Table I provides the weight $\Phi_{[1,2]}(s_i)$ of each sequence $s_i$. It is clear that this weight is different from the number of occurrences $2^{\|s_i\|}$ or the number of distinct subsequences $\Phi(s_i)$ and shows the importance of this calculation so as not to bias the drawing of the subsequence.

*b) Step 2: Sampling a subsequence:* In the second step, we randomly draw the norm $k$ of the subsequence of $s$ which will be returned (line 3 of Algorithm 1). This number $k$ is randomly drawn with a probability proportional to the number of subsequences in $s$ having exactly $k$ as norm, i.e. according to the probability distribution $P_{[m,M]}$ defined for all $k \in [m..M]$ by: $P_{[m,M]}(k) = \frac{\Phi_{[k,k]}(s)}{\Phi_{[m,M]}(s)}$. Finally, Algorithm 1 returns at line 4 a subsequence $s'$ in $s$ of norm $k$ according to a uniform distribution, meaning that each subsequence $s'$ from $s$ of norm $k$ will be drawn with the same probability $\frac{1}{\Phi_{[k,k]}(s)}$. We show in Section IV-C how to perform such a uniform drawing thanks to a rejection sampling. The main challenge is to avoid to pick more often subsequences that have multiple occurrences within the sequence $s$. Typically, even if $\langle(a)(c)\rangle$ has two occurrences in $s_2$, its drawing probability must be the same as that of $\langle(a)(a)\rangle$ (that appears once within $s_2$).

Note that the theoretical study of these two steps (soundness and complexity) will be done in Section IV-D.

**Algorithm 1** CSSAMPLING

**Input:** A sequential dataset $\mathcal{S}$, and two integers $m$ and $M$ such that $m \leq M$

**Output:** A sequence $s \in \mathbb{S}_{[m,M]}$ randomly drawn, i.e. $s \sim freq(\mathbb{S}_{[m,M]},\mathcal{S})$

*// Step 1: Sampling a sequence*

1: Compute for all $s \in \mathcal{S}$, a weight $w$ defined by $w(s) = \Phi_{[m,M]}(s)$

2: Draw a sequence $s$ from $\mathcal{S}$ proportionally to $w$: $s \sim w(\mathcal{S})$

*// Step 2: Sampling a subsequence*

3: Draw an integer $k$ from $m$ to $M$ according to the distribution $P_{[m,M]}(k)$

4: **return** A subsequence $s'$ of norm $k$ randomly drawn from $s$: $s' \sim u(\phi_{[k,k]}(s))$ where $u$ is the uniform distribution

### B. Subsequence counting for drawing a sequence

In this section, we show how to compute the number of distinct subsequences of a sequence with an interval constraint on the norm. We benefit from [29] where a formula counts the number of distinct subsequences in a sequence *without* constraint on the norm. The main difficulty is to avoid to count the same subsequence several times, even if it has several occurrences within the sequence.

To compute the number of distinct subsequences having a norm less than or equal to $j$ contained in a sequence $s = \langle X_1 \dots X_n\rangle$, we start with the empty sequence and then, we concatenate all itemsets $X_i$ one by one. $s \circ Y$ denotes the concatenation of $s$ and $Y$: $s \circ Y = \langle X_1 \dots X_n Y\rangle$. For each new itemset $Y$ concatenated to $s$, we count only subsequences which have a norm less than $j$ and which have not already occurred previously in $s$. For instance, if we add the itemset $ac$ to $\langle(ab)c\rangle$ to count the number of subsequences having a norm less than 2 in $\langle(ab)c(\mathbf{ac})\rangle$, then we avoid counting $\langle(ab)\mathbf{a}\rangle$ whose norm (i.e., 3) is too large and we avoid counting $\langle(a)\mathbf{c}\rangle$ which has already been counted previously (for $\langle(ab)\rangle \circ c$). It is easy to see that the duplicates (here, only $\langle(a)c\rangle$) result from previous occurrences of items in $(ac)$ within sequences $\langle(ab)c\rangle$ (here, $c$ occurs previously at position 2). For this reason, we need the notion of position set:

*Definition 4 (Position set [29]):* Let $s$ be a sequence and $Y$ be an itemset. $L(s,Y) = \{i \in \mathbb{N} : i \leq |s| \wedge s[i] \cap Y \neq 0 \wedge (\forall j > i)(s[i] \cap Y \not\subseteq s[j] \cap Y)\}$ is the position set where $Y$ has a maximal intersection with the different itemsets of $s$.

*Example 4:* Let $s = \langle(ab)c(ac)\rangle$ be a sequence. We have $s^1 = \langle(ab)\rangle$, $s[2] = (c)$ and $L(s^1, s[2]) = \emptyset$ because $s[2]$ intersects no itemset of $s^1$. Now, we are going to compute $L(s^2, s[3])$. $s[3] = (ac)$ intersects at the same time the first itemset $s[1] = (ab)$ of $s$ ($s[1] \cap s[3] = (a)$) and the second itemset $s[2] = (c)$ of $s$ ($s[2] \cap s[3] = (c)$). As these two intersections are disjoint, we obtain $L(s^2, s[3]) = \{1, 2\}$. This means that by concatenating subsets of $s[3]$ to the subsequences in $s^2$, some subsequences of $s^2$ might been counted twice as items of $s[3]$ are also present at positions 1 and 2 in $s^2$.

Using the notion of position set and the inclusion-exclusion principle, we propose a new recursive formula to count the number of distinct subsequences in a sequence $s$ considering a maximum norm as constraint. Intuitively, to construct a

subsequence of $s \circ Y$ having a norm less than $j$, we can concatenate any subset of size $k$ of $Y$ to a subsequence of $s$ having a norm less than $j-k$. Indeed, we are sure to obtain a subsequence of $s \circ Y$ having a norm less than $k+(j-k)=j$, and this principle is repeated for any possible size of a subset of $Y$. Thus, we have: $\phi_{\leq j}(s \circ Y) = \cup_{k=0}^{j} \phi_{\leq j-k}(s) \circ \mathcal{P}_{=k}(Y)$ where $\mathcal{P}_{=k}(Y) = \{X \subseteq Y : |X| = k\}$, which explains the first term of the formula given by Theorem 1. The difficulty is that a subsequence obtained by the concatenation of a subset of $Y$ to a subsequence of $s$ may also occur in $\phi_{\leq j}(s)$. Therefore, we have to take into account these possible redundancies to count the exact number of distinct subsequences of $s$ with a norm less than $j$. This remark explains the correction term $R_{\leq j}(s,Y)$ of the formula given by Theorem 1:

*Theorem 1 (Subsequence number with a maximum norm):* Let $s$ be a sequence, $Y$ be an itemset and $j$ be an integer, the number of distinct subsequences having a norm less or equal to $j$ in $s \circ Y$, denoted by $\Phi_{\leq j}(s \circ Y)$, is defined as follows[1]:

$$\Phi_{\leq j}(s \circ Y) = \left( \sum_{k=0}^{j} \Phi_{\leq j-k}(s) \times \binom{|Y|}{k} \right) - R_{\leq j}(s,Y)$$

where $R_{\leq j}(s,Y)$ is the correction term defined by:

$$R_{\leq j}(s,Y) = \sum_{\emptyset \subset K \subseteq L(s,Y)} (-1)^{|K|+1} R^K_{\leq j}(s,Y)$$

with $R^K_{\leq j}(s,Y) = \sum_{k=1}^{j} \Phi_{\leq j-k}(s^{min(K)-1}) \times \binom{|s[K] \cap Y|}{k}$ where $s[K] = \cap_{k \in K} s[k]$.

This Theorem 1 extends the proposal [29] by setting $j = \infty$.

*Proof:* Let $s$ be a sequence and $Y$ be an itemset. We already explain that to construct a subsequence of $s \circ Y$ having a norm less than $j$, we can concatenate any subset of size $k$ of $Y$ to a subsequence of $s$ having a norm less than $j-k$. Indeed, we are sure to obtain a subsequence of $s \circ Y$ having a norm less than $k+(j-k)=j$. Thus, we have $\phi_{\leq j}(s \circ Y) = \cup_{k=0}^{j} \phi_{\leq j-k}(s) \circ \mathcal{P}_{=k}(Y)$ and $\Phi_{\leq j}(s \circ Y) = \sum_{k=0}^{j} \Phi_{\leq j-k}(s) \times \binom{|Y|}{k} - R_{\leq j}(s,Y)$ where $R_{\leq j}(s,Y)$ is a correction term (to count the number of *distinct* subsequences).

Let $t = \langle T_1 \dots T_m \rangle$ with $|T_m| = k$ be a sequence that is counted multiple times, i.e. $t \in \phi_{\leq j}(s) \cap (\phi_{\leq j}(s) \circ \mathcal{P}_{\geq 1}(Y))$ where $\mathcal{P}_{\geq 1}(Y) = \{X \subseteq Y : |X| \geq 1\}$. Because $t \in (\phi_{\leq j}(s) \circ \mathcal{P}_{\geq 1}(Y))$, we necessarily have $T_m \in \mathcal{P}_{\geq 1}(Y)$, i.e. $T_m \subseteq Y$. Moreover, because $t \in \phi_{\leq j}(s)$, there exists an integer $i \leq |s|$ such that $T_m \subseteq s[i]$. Let $l = max\{i \leq |s| : T_m \subseteq s[i]\}$. Since $T_m \subseteq Y$, we also have $l = max\{i \leq |s| : T_m \subseteq (s[i] \cap Y)\}$. We show now that $l \in L(s,Y)$. First, because $T_m \neq \emptyset$, we have $s[l] \cap Y \neq \emptyset$. Now, assume that there exists $l' > l$ such that $s[l] \cap Y \subseteq s[l'] \cap Y$. Then, we would have $T_m \subseteq s[l'] \cap Y$, which contradicts that $l$ is maximal, and completes the proof that $l \in L(s,Y)$. At this point, we proved that $T \in \phi_{\leq j-k}(s^{l-1}) \circ \mathcal{P}_{=k}(s[l] \cap Y)$ for an integer $l \in L(s,Y)$. Thus, we have $R_{\leq j}(s,Y) = |\bigcup_{l \in L(s,Y)} (\cup_{k=1}^{j} \phi_{\leq j-k}(s^{l-1}) \circ \mathcal{P}_{=k}(s[l] \cap Y))|$.

Using the inclusion-exclusion principle, we rewrite $R_{\leq j}(s,Y)$ as $\sum_{\emptyset \subset K \subseteq L(s,Y)} (-1)^{|K|+1} R^K_{\leq j}(s,Y)$ with $R^K_{\leq j}(s,Y) = |\bigcap_{l \in K} (\cup_{k=1}^{j} \phi_{\leq j-k}(s^{l-1}) \circ \mathcal{P}_{=k}(s[l] \cap Y))|$.

[1]By convention, we consider that $\binom{n}{p} = 0$ if $p > n$.

Now, let $t = \langle T_1 \dots T_m \rangle$ be a sequence in the set $\bigcap_{l \in K} (\cup_{k=1}^{j} \phi_{\leq j-k}(s^{l-1}) \circ \mathcal{P}_{=k}(s[l] \cap Y))$. We necessarily have $t^{m-1} \in \phi_{\leq j-k}(s^{min(K)-1})$ and $T_m \in \cap_{l \in K} \mathcal{P}_{=k}(s[l] \cap Y)$, i.e. $T_m \in \mathcal{P}_{=k}(s[K] \cap Y)$ with $s[K] = \cap_{l \in K} s[l]$. It follows that $R^K_{\leq j}(s,Y) = |\cup_{k=1}^{j} \phi_{\leq j-k}(s^{min(K)-1}) \circ \mathcal{P}_{=k}(s[K] \cap Y))|$. Finally, because the sets $\phi_{\leq j-k}(s^{min(K)-1}) \circ \mathcal{P}_{=k}(s[K] \cap Y))$ are disjoints, we have $R^K_{\leq j}(s,Y) = \sum_{k=1}^{j} \Phi_{\leq j-k}(s^{min(K)-1}) \times \binom{|s[K] \cap Y|}{k}$, which completes the proof of Theorem 1. ■

By continuing Example 4 with the sequence $s = \langle (ab)c(ac) \rangle$, the following example illustrates the principle of the formula given by Theorem 1.

*Example 5:* The set $\phi_{\leq 2}(s^1)$ of subsequences of $s^1 = \langle (ab) \rangle$ with a norm less than 2 is defined by $\phi_{\leq 2}(s^1) = \{\langle\,\rangle, \langle a \rangle, \langle b \rangle, \langle (ab) \rangle\}$. We have $\Phi_{\leq 2}(s^1) = 4$, and it is easy to see that $\Phi_{\leq 1}(s^1) = 3$ (the subsequence $\langle (ab) \rangle$ having a norm strictly greater than 1). As $L(s^1, s[2]) = \emptyset$, we have $R_{\leq 2}(s^1, s[2]) = 0$ and $\Phi_{\leq 2}(s^2) = \sum_{k=0}^{|(c)|} \Phi_{\leq 2-k}(s^1) \times \binom{|(c)|}{k} = \Phi_{\leq 2}(s^1) \times \binom{1}{0} + \Phi_{\leq 1}(s^1) \times \binom{1}{1} = 4 + 3 = 7$. The first term of the sum corresponds to 4 subsequences in $s^3$ obtained by concatenating the empty set to subsequences of $s^2$, while the second term corresponds to 3 subsequences in $s^3$ obtained by concatenating the itemset $(c)$ to each subsequence of $s^2$ having a norm less than 1. Let us detail the calculation of $\Phi_{\leq 2}(s^3) = \sum_{k=0}^{|(ac)|} \Phi_{\leq 2-k}(s^2) \times \binom{|(ac)|}{k} - R_{\leq 2}(s^2, s[3]) = \Phi_{\leq 2}(s^2) + \Phi_{\leq 1}(s^2) \times 2 + \Phi_{\leq 0}(s^2) - R_{\leq 2}(s^2, s[3]) = 7 + 4 \times 2 + 1 - R_{\leq 2}(s^2, s[3])$. For instance, the second term of $\Phi_{\leq 2}(s^3)$, that equals to $4 \times 2$, refers to the number of subsequences in $s^3$ that are obtained by concatenating the two subsets of size 1 of $(ab)$ with a subsequence in $s^2$ having a norm less that 1. Finally, the calculation of the correction term $R_{\leq 2}(s^2, s[3])$ is as follows: $R_{\leq 2}(s^2, s[3]) = (-1)^2 \Phi_{\leq 1}(s^0) \times \binom{|(a)|}{1} + (-1)^2 \Phi_{\leq 1}(s^1) \times \binom{|(c)|}{1} = 1 + 3 = 4$. Thereby, we deduce that $\Phi_{\leq 2}(s^3) = 7 + 4 \times 2 + 1 - 4 = 12$.

The formula given by Theorem 1 is recursive. Nevertheless, given a sequence $s$ and a maximum norm $M$, this recursion can easily be removed by calculating line by line the matrices $T$ and $R$ defined by:

- $T[i][j] = \Phi_{\leq j}(s^i)$ for $i \in [0..|s|]$ and $j \in [0..M]$. $T[i][j]$ is the number of subsequences with a norm less than or equal to $j$ in the sequence $s^i$.
- $R[i][j] = R_{\leq j}(s^{i-1}, s[i])$ for $i \in [2..|s|]$ and $j \in [0..M]$. This correction term is the term required to correct the number of subsequences with a norm less than $j$ of $s^i = s^{i-1} \circ s[i]$ using the number of subsequences with a norm less than $j$ of $s^i$ by concatenating the subsets of $s[i]$.

Algorithm 2 details how the matrices $T$ and $R$ can be computed for a sequence $s$ and a maximum norm $M$. At each iteration of the main loop (lines 5 to 19 of Algorithm 2), it computes the number $T[i][j]$ of subsequences $s^i$ of $s$ with a norm less than or equal to $j$ (for all $j \in [1..M]$) using the previous lines of matrices $T$ and $R$. For each $i \in [2..|s|]$ and $j \in [1..M]$, Algorithm 2 first computes the correction term $R[i][j]$ (lines 7-13). Because $K \subseteq L(s^{i-1}, s[i])$, it is important to note that $m = min(K) \leq i-1 < i$. Thus, at line 11, it ensures that to calculate $R[i][j]$, only previously calculated

terms $T[m-1][j-k]$ of $T$ are used. Then, Algorithm 2 computes (lines 14-17) the value of $T[i][j]$ using only the previous line $i-1$ of matrix $T$ (line 15) and the correction term $R[i][j]$ (line 17). Examples of the matrices $T$ and $R$ are provided by Table III for a sequence $s = \langle (ab)c(ac) \rangle$. In particular, we find the values $R[3][2] = R_{\leq 2}(s^2, s[3])$ and $T[3][2] = \Phi_{\leq 2}(s^3)$ computed in Example 5.

**Algorithm 2** Number of subsequences with a maximum norm

**Input:** A sequence $s$ and a maximal norm $M \leq \|s\|$
**Output:** A matrix $T$ such that $T[i][j] = \Phi_{\leq j}(s^i)$

1: $T[0][0] := T[1][0] = 1$
2: **for** $j = 1$ to $M$ **do**
3: $\quad T[0][j] := 1$ and $T[1][j] := T[1][j-1] + \binom{|s[1]|}{j}$
4: **end for**
5: **for** $i = 2$ to $|s|$ **do**
6: $\quad$ **for** $j = 1$ to $M$ **do**
7: $\quad\quad R[i][j] := T[i][j] = 0$
8: $\quad\quad$ **for all** $K \in \mathcal{P}_{\geq 1}(L(s^{i-1}, s[i]))$ **do**
9: $\quad\quad\quad m := min(K)$ and $k_{max} := |s[K] \cap s[i]|$
10: $\quad\quad\quad$ **for** $k = 1$ to $k_{max}$ **do**
11: $\quad\quad\quad\quad R[i][j] \mathrel{+}= (-1)^{|K|+1} T[m-1][j-k] \times \binom{k_{max}}{k}$
12: $\quad\quad\quad$ **end for**
13: $\quad\quad$ **end for**
14: $\quad\quad$ **for** $k = 0$ to $min\{j, |s[i]|\}$ **do**
15: $\quad\quad\quad T[i][j] \mathrel{+}= T[i-1][j-k] \times \binom{|s[i]|}{k}$
16: $\quad\quad$ **end for**
17: $\quad\quad T[i][j] := T[i][j] - R[i][j]$
18: $\quad$ **end for**
19: **end for**
20: **return**$(T)$

To conclude this section, using Theorem 1, note that we calculate the number of distinct subsequences in a sequence $s$ having a norm between $m$ and $M$ as follows: $\Phi_{[m,M]}(s) = \Phi_{\leq M}(s) - \Phi_{\leq m-1}(s)$. In Algorithm 1, this formula makes it possible to calculate the initial weight $w(s)$ for each sequence $s$ of the sequential database $\mathcal{S}$ (see line 1 of Algorithm 1).

TABLE III. EXAMPLES OF MATRICES $T$ AND $R$

| **T[i][j]** | $\leq \mathbf{0}$ | $\leq \mathbf{1}$ | $\leq \mathbf{2}$ | $\leq \mathbf{3}$ |
|---|---|---|---|---|
| $s^{\mathbf{0}} = \langle\rangle$ | 1 | 1 | 1 | 1 |
| $s^{\mathbf{1}} = \langle (ab) \rangle$ | 1 | 3 | 4 | 4 |
| $s^{\mathbf{2}} = \langle (ab)c \rangle$ | 1 | 4 | 7 | 8 |
| $s^{\mathbf{3}} = \langle (ab)c(ac) \rangle$ | 1 | 4 | 12 | 21 |

| **R[i][j]** | $\leq \mathbf{0}$ | $\leq \mathbf{1}$ | $\leq \mathbf{2}$ |
|---|---|---|---|
| $s^{\mathbf{1}}, s[\mathbf{2}] = c$ | 0 | 0 | 0 |
| $s^{\mathbf{2}}, s[\mathbf{3}] = (ac)$ | 2 | 4 | 5 |

### C. Subsequence sampling by rejection

After randomly drawing a sequence $s \in \mathcal{S}$ proportionally to its weight $w(s)$ (line 2 of Algorithm 1) and an integer $k$ between $m$ and $M$ according to the distribution $P_{[m,M]}(k)$ (line 3 of Algorithm 1), CSSAMPLING aims at returning a subsequence of norm $k$ drawn uniformly from the sequence $s$ (line 4 of Algorithm 1). The difficulty is not to favor the subsequences that have multiple occurrences within the sequence.

To cope with this difficulty, we use a rejection method by uniformly drawing an occurrence of the sequence $s$ and rejecting it if this occurrence is not the first one. As each subsequence has a unique first occurrence, this approach ensures a uniform draw of subsequences. We start by formalizing the notion of first occurrence:

*Definition 5 (First occurrence):* Given a sequence $s$, let $o_1$ and $o_2$ be two occurrences of a subsequence $s'$ within $s$, whose signatures are $\langle i_1^1, i_2^1, \ldots, i_m^1 \rangle$ and $\langle i_1^2, i_2^2, \ldots, i_m^2 \rangle$ respectively. $o_1$ is less than $o_2$, denoted by $o_1 < o_2$, if there exists an index $k \in [1..m]$ such that for all $j \in [1..k-1]$, one has $i_j^1 = i_j^2$, and $i_k^1 < i_k^2$. Finally, we call the *first occurrence* of $s'$ in $s$ its smallest occurrence w.r.t. the order defined previously.

*Example 6:* Let us continue Example 2 where $\langle 1, 2 \rangle$ and $\langle 1, 3 \rangle$ are the signatures of occurrences $o_1 = \langle (a)(c)\emptyset \rangle$ and $o_2 = \langle (a)\emptyset(c) \rangle$ of the subsequence $s' = \langle (a)(c) \rangle$ in $s = \langle (ab)(cd)(ce) \rangle$. As $\langle 1, 2 \rangle$ is less than $\langle 1, 3 \rangle$, we obtain that $o_1 < o_2$. Finally, as $o_1$ and $o_2$ are the only two occurrences of $s'$ in $s$, it means that $o_1$ is the first occurrence of $s'$ in $s$.

In practice, we especially check if an occurrence of the subsequence $s' \sqsubseteq s$ is the first occurrence of $s'$ within the sequence $s$. This can be done efficiently by using Property 1:

*Property 1:* Given an occurrence $o$ of the subsequence $s' \sqsubseteq s$ whose signature is $\sigma = \langle i_1, i_2, \ldots, i_m \rangle$, $o$ is the first occurrence of $s'$ if and only if for all $i_j \in \sigma$, there is no index $l \in [i_{j-1} + 1..i_j - 1]$ such that $o[i_j] \subseteq s[l]$ (with $i_0 = 0$).

*Proof:* Let $\sigma = \langle i_1, \ldots, i_m \rangle$ be the signature of an occurrence $o$ of $s' \sqsubseteq s$. We first show that if there exist $i_j \in \sigma$ and $l \in [i_{j-1}+1..i_j-1]$ such that $o[i_j] \subseteq s[l]$, then $o$ is not the first occurrence of $s'$. Let $1 \leq i'_1 < i'_2 < \cdots < i'_m \leq n$ be the index sequence defined by $i'_j = l$ and for all $k \in [1..m] \setminus \{j\}$, $i'_k = i_k$. Consider now the ordered list $o'$ of $n$ itemsets defined by $o'[l] = o[i_j]$, $o'[i_j] = \emptyset$ and for all $k \in [1..n] \setminus \{l, i_j\}$, $o'[k] = o[k]$. As $o'$ is an occurrence of $s' \sqsubseteq s$ and $o' < o$, it proves that $o$ is not the first occurrence of $s'$. Conversely, we show that if $o$ of signature $\sigma$ is not the first occurence of $s' \sqsubseteq s$, then there exist $i_j \in \sigma$ and $l \in [i_{j-1} + 1..i_j - 1]$ such that $o[i_j] \subseteq s[l]$. By definition, if $o$ is not the first occurrence of $s$, then there exists another occurrence $o'$ of $s'$ such that $o' < o$. So, we know that there exists $k \in [1..n]$ such that $i'_k < i_k$ and for all $j \in [1..k-1]$, $i'_j = i_j$. Thus, there exist indexes $i_k \in \sigma$ and $l = i'_k \in [i'_{k-1} + 1..i_k - 1] = [i_{k-1} + 1..i_k - 1]$ such that $o[i_k] = o[i'_k] \subseteq s[i'_k]$, i.e. $o[i_k] \subseteq s[l]$. ■

Thanks to Property 1, it is finally easy to draw uniformly a subsequence of norm $k$ in a sequence $s$. By randomly drawing $k$ distinct item positions between 1 and $\|s\|$, we start by uniformly drawing an occurrence containing $k$ items from $s$. If this occurrence is a first occurrence, it is accepted and returned. Otherwise we reject it and perform another random draw of a new occurrence of $s$. Although CSSAMPLING relies on a rejection sampling technique, we show in the next section that the average number of draws before acceptance is computable. The experimental section also shows that this average number of draws may be extremely low for real-world datasets.

*Example 7:* In Example 2, assume that we have drawn item positions 1 and 5 within the sequence $s = \langle (\mathbf{ab})(cd)(\mathbf{c}e) \rangle$ in order to build an occurence of a subsequence of $s$ of norm $k = 2$. In this way, we obtain the occurrence $o = \langle (a)\emptyset(c) \rangle$ of signature $\langle 1, 3 \rangle$ of the subsequence $s' = \langle (a)(c) \rangle$ in $s$. In that case, as there exists $l = 2$ in $[1 + 1..3 - 1]$ such that

$o[3] = (c) \subseteq s[2] = (cd)$, we are sure that $o$ is not the first occurrence of $s'$ and this occurrence is rejected.

### D. Theoretical analysis of the method

This property states that CSSAMPLING returns an exact sample of subsequences with norm constraints:

*Property 2 (Soundness):* Let $\mathcal{S}$ be a sequential dataset, $m$ be a minimum norm and $M$ a maximum norm, CSSAMPLING draws a subsequence of $\mathcal{S}$ having a norm between $m$ and $M$ according to a distribution proportional to frequency.

*Proof:* Let $Z$ be the normalizing constant defined by $Z = \sum_{s \in \mathcal{S}} w(s) = \sum_{s \in \mathcal{S}} \Phi_{[m,M]}(s)$. Let $t$ be a subsequence in $\mathbb{S}_{[m,M]}$ and $P(t)$ be the probability to draw subsequence $t$ using Algorithm 1. We have: $P(t) = \sum_{s \in \mathcal{S}} P(t, s) = \sum_{s \in \mathcal{S}, t \sqsubseteq s} P(s) \times P(t/s)$. Considering the second line of Algorithm 1, we have $P(s) = \frac{w(s)}{Z} = \frac{\Phi_{[m,M]}(s)}{Z}$. Then, considering the third and fourth lines of Algorithm 1, if $t$ is a subsequence of norm $k$, we have $P(t/s) = P(k/s) \times P(t/k, s) = \frac{\Phi_{[k,k]}(s)}{\Phi_{[m,M]}(s)} \times \frac{1}{\Phi_{[k,k]}(s)} = \frac{1}{\Phi_{[m,M]}(s)}$. Thus, we have $P(t) = \sum_{s \in \mathcal{S}, t \sqsubseteq s} P(s) \times P(t/s) = \sum_{s \in \mathcal{S}, t \sqsubseteq s} \frac{\Phi_{[m,M]}(s)}{Z} \times \frac{1}{\Phi_{[m,M]}(s)} = \frac{freq(s,\mathcal{S})}{Z}$, which shows that $t$ is drawn proportionnaly to its frequency and completes the proof. ■

We now study the complexity of our method by distinguishing two main phases: the preprocessing (where the distribution of subsequences according to the norm is calculated for each sequence) and the drawing of subsequences.

*a) Preprocessing complexity:* The preprocessing is performed in time $O(|\mathcal{S}| \cdot L \cdot M^2 \cdot 2^P \cdot T^2)$ where $L$ is the maximum length of a sequence, $M$ is the maximum norm of drawn subsequences, $P$ is the maximum size of position sets $L(s^{i-1}, s[i])$ and $T$ is the maximum size of an itemset in a sequence. It is important to note that $P \leq L$ may be very small in practice (see the next section) and that this preprocessing (line 1 of Algorithm 1) is achieved only once before the drawing phase (where a large number of subsequences are drawn from $\mathcal{S}$). Moreover, it is important to note that if the dataset $\mathcal{S}$ contains only sequences of *items* (and not sequences of *itemsets*), then we have $P = 1$. Thus, in that case, the preprocessing can be performed in polynomial time $O(|\mathcal{S}| \cdot L \cdot M^2 \cdot T^2)$.

*b) Drawing complexity:* The draw of subsequences is less expensive. First, the draw of a sequence (line 2 of Algorithm 1) is realized in $O(\ln |\mathcal{S}|)$. It is more difficult to estimate the complexity in the worst case for the draw of a subsequence because the number of rejections is not bounded. Nevertheless, a good way to measure the effectiveness of the approach is to calculate the average number of draws, denoted by $\mu_{[m,M]}(\mathcal{S})$, required to derive a subsequence of $\mathcal{S}$ having a norm between $m$ and $M$. Intuitively, $\mu_{[m,M]}(\mathcal{S})$ depends both on the probability that a sequence $s \in \mathcal{S}$ is drawn and the average number of draws, denoted by $\mu_{[m,M]}(s)$, required to find a first occurrence of a subsequence of $s$. The following property shows how these terms can be calculated:

*Property 3 (Average number of draws):* The average number of draws for the acceptance of a subsequence having a norm between $m$ and $M$ in the sequential dataset $\mathcal{S}$ is defined by: $\mu_{[m,M]}(\mathcal{S}) = \sum_{s \in \mathcal{S}} \frac{\Phi_{[m,M]}(s)}{\sum_{s' \in \mathcal{S}} \Phi_{[m,M]}(s')} \times \mu_{[m,M]}(s)$ where $\mu_{[m,M]}(s) = \frac{\sum_{k=m}^{M} \binom{\|s\|}{k}}{\Phi_{[m,M]}(s)}$.

*Proof:* Using Algorithm 1, it is clear that $\mu_{[m,M]}(\mathcal{S}) = \sum_{s \in \mathcal{S}} P(s) \times \mu_{[m,M]}(s)$ with $P(s) = \frac{\Phi_{[m,M]}(s)}{\sum_{s' \in \mathcal{S}} \Phi_{[m,M]}(s')}$. Then, we have $\mu_{[m,M]}(s) = \sum_{k \in [m..M]} P(k/s) \times N_k(s)$ where $N_k(s)$ is the average number of draws necessary to obtain a subsequence $s'$ of $s$ such that $\|s'\| = k$. When we draw a subsequence $s'$ of norm $k$, the probability that this subsequence is accepted (because it is a first occurrence) is $P_a^k(s) = \frac{\Phi_{[k,k]}(s)}{\binom{\|s\|}{k}}$. Thus, we have $N_k(s) = \sum_{i=1}^{\infty} i \times (1 - P_a^k(s))^{i-1} \times P_a^k(s) = P_a^k(s) \times \sum_{i=1}^{\infty} i \times (1 - P_a^k(s))^{i-1} = P_a^k(s) \times \frac{1}{P_a^k(s)^2} = \frac{1}{P_a^k(s)}$. It follows that $\mu_{[m,M]}(s) = \sum_{k \in [m..M]} P(k/s) \times N_k(s) = \sum_{k \in [m..M]} \frac{\Phi_{[k,k]}(s)}{\Phi_{[m,M]}(s)} \times \frac{\binom{\|s\|}{k}}{\Phi_{[k,k]}(s)} = \frac{\sum_{k \in [m..M]} \binom{\|s\|}{k}}{\Phi_{[m,M]}(s)}$. ■

When the average number of draws is close to 1, it means that the draw of a subsequence is achieved without rejection. For a given sequence, there is no rejection if each occurrence is the first occurrence i.e., there is no duplicate within the sequence. In practice, the average number of draws measured on real-world datasets is often very low. Finally, as the temporal complexity of the draw of an occurrence having a norm equal to $k \in [m..M]$ in a sequence $s$ is in the worst case in $O(M^2)$, the average complexity of drawing $N$ subsequences from a dataset $\mathcal{S}$ (after the preprocessing phase) is in $O(N \cdot M^2 \cdot \mu_{[m,M]}(\mathcal{S}))$.

## V. EXPERIMENTAL STUDY

In the previous section, we proved that our sampling algorithm CSSAMPLING is exact, and studied its complexity. In this section, we evaluate the efficiency of the approach and the interest of the sampled subsequences. More precisely, Section V-A focuses on the speed of CSSAMPLING and its ability to draw patterns that do not belong to the long tail. In Section V-B, in order to illustrate the usefulness of sampled patterns, we show how these patterns can be used to build associative classifiers dedicated to sequences and that our approach rivals state of the art proposal.

### A. Analysis of CSSAMPLING method

This experimental section evaluates the speed of our method and the impact of the norm constraint on the sampled patterns. For this, we use 6 datasets including 2 real life datasets `bms` and `sign`[2] and 4 synthetic datasets generated by IBM data generator[3]. One of the interests of using synthetic datasets is to have examples where the average number of draws $\mu_{[m,M]}(\mathcal{S})$ is ensured to be greater than 1 by adding multiple occurrences within a same sequences. Table V lists basic statistics of all datasets and Table VI compares the average number of draws per subsequence required to extract a pattern with $M \in \{1, 2, 3, 5, 7\}$ (while $m$ is always fixed to 1 in all of our experiments). The prototype of our method is implemented in Python and all experiments are performed on a 2.71 GHz 2 Core CPU with 12 GB of RAM. All experimental datasets used, as well as source code, are available at `https://github.com/LDIOPBSF/CSSampling`.

[2] `http://www.philippe-fournier-viger.com/spmf`
[3] `https://github.com/zakimjz/IBMGenerator`

TABLE IV. EXECUTION TIME FOR SEQUENTIAL PATTERN SAMPLING (AVERAGE AND STANDARD DEVIATION)

| Dataset | Preprocessing time (s) | | | | | Drawing time per pattern (ms) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $M$=1 | $M$=2 | $M$=3 | $M$=5 | $M$=7 | $M$=1 | $M$=2 | $M$=3 | $M$=5 | $M$=7 |
| bms | 0.22±0.00 | 0.29±0.01 | 0.30±0.00 | 0.30±0.01 | 0.31±0.02 | 0.07±0.01 | 0.25±0.01 | 0.43±0.02 | 0.59±0.00 | 0.7±0.01 |
| sign | 0.01±0.00 | 0.02±0.00 | 0.02±0.00 | 0.02±0.00 | 0.03±0.00 | 0.15±0.01 | 0.17±0.01 | 0.19±0.01 | 0.22±0.00 | 0.24±0.00 |
| D10K5S2T6I | 0.81±0.33 | 2.02±0.02 | 2.92±0.03 | 5.02±0.05 | 7.36±0.06 | 0.05±0.00 | 0.07±0.00 | 0.09±0.00 | 0.17±0.00 | 0.24±0.01 |
| D10K6S3T10I | 1.38±0.04 | 3.11±0.10 | 5.16±0.04 | 9.33±0.18 | 14.56±0.10 | 0.06±0.01 | 0.09±0.01 | 0.14±0.01 | 0.24±0.02 | 0.37±0.01 |
| D100K5S2T6I | 5.86±0.26 | 12.34±0.11 | 18.8±0.21 | 32.88±0.27 | 49.24±0.29 | 0.05±0.01 | 0.07±0.01 | 0.09±0.00 | 0.13±0.01 | 0.20±0.02 |
| D100K6S2T6I | 8.44±0.55 | 17.48±0.26 | 27.33±0.74 | 49.48±0.90 | 74.89±0.56 | 0.06±0.01 | 0.07±0.02 | 0.10±0.01 | 0.15±0.01 | 0.21±0.01 |

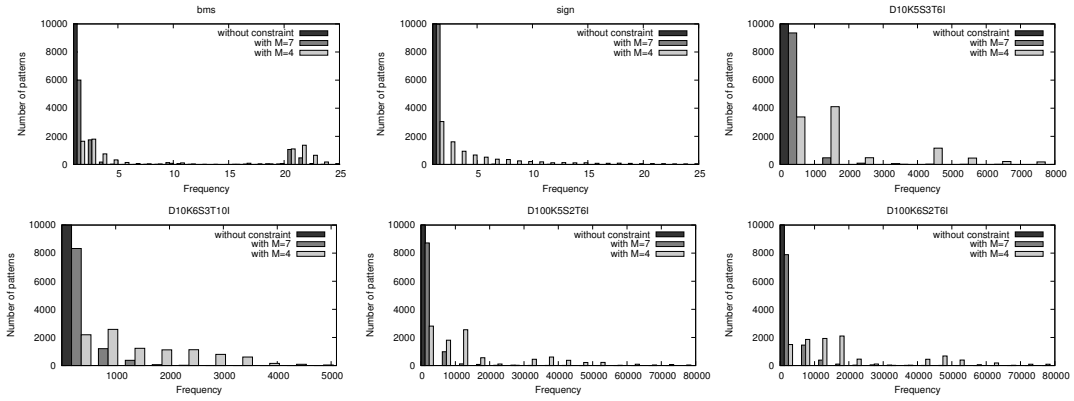



Fig. 2. Distribution of 10,000 sequential patterns according to frequency

TABLE V. STATISTICS OF BENCHMARK DATASETS

| Dataset | $\|\mathcal{S}\|$ | $\|\mathcal{I}\|$ | $\|S\|_{max}$ | $\|S\|_{mean}$ | P | T |
|---|---|---|---|---|---|---|
| bms | 59,601 | 497 | 267 | 2.5 | 1 | 1 |
| sign | 730 | 267 | 94 | 52.0 | 1 | 1 |
| D10K5S2T6I | 10,000 | 6 | 70 | 10.3 | 7 | 6 |
| D10K6S3T10I | 10,000 | 10 | 92 | 15.9 | 10 | 6 |
| D100K5S2T6I | 100,000 | 6 | 72 | 8.5 | 7 | 6 |
| D100K6S2T6I | 100,000 | 6 | 83 | 10.4 | 8 | 9 |

TABLE VI. AVERAGE NUMBER OF DRAWS PER SUBSEQUENCE

| Dataset | $M$=1 | $M$=2 | $M$=3 | $M$=5 | $M$=7 |
|---|---|---|---|---|---|
| bms | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| sign | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| D10K5S2T6I | 4.0 | 7.0 | 11.4 | 23.5 | 38.4 |
| D10K6S3T10I | 3.9 | 6.7 | 10.4 | 18.5 | 25.7 |
| D100K5S2T6I | 3.6 | 5.8 | 8.5 | 14.9 | 23.9 |
| D100K6S2T6I | 4.0 | 7.0 | 11.1 | 21.4 | 32.4 |

*1) Pre-processing and sampling speed:* Table IV indicates the execution time of our method by distinguishing the preprocessing time and the average number of draws of a sequential pattern with $M \in \{1, 2, 3, 5, 7\}$. As expected, the preprocessing time increases with the size of the dataset, the maximum size $P$ of position sets, the maximum size $T$ of an itemset in a sequence, and the maximum norm $M$ of drawn subsequences. However, even for D100K6S2T6I which is large, the execution time of the preprocessing (which can be prepared off-line) is quite reasonable. Regarding the sampling phase, whatever the dataset and the maximum norm $M$, the execution time is always under 1 millisecond. Despite an average number of draws $\mu_{[m,M]}(\mathcal{S})$ greater than 1 (and hence, rejection), performances on synthetic datasets are good.

*2) Impact of norm constraints:* Figure 2 depicts the distribution of 10,000 sequential patterns sampled according to frequency with a maximum norm constraint of 4, 7, and without constraint for different datasets. In all cases, the unconstrained method returns only very low frequent patterns and in particular, with 1 as frequency on real-world datasets. Conversely, the constrained sampling method returns sequential patterns with significantly higher frequency, which shows the importance of introducing constraints on the norm to avoid the problem of the long tail. More precisely, we can see that the lower the value of the $M$ constraint is, the more the method allows to draw patterns with high frequency values. For instance, for D100K6S2T6I, the mean frequency of sampled patterns is equal to $3,770$ using $M = 7$, whereas it is equal to $19,683$ using $M = 4$. Note that for sign, the maximum norm of 7 is not sufficient to return sampled patterns with frequency greater than 1. A norm of at most 4 is necessary so that the frequencies of the subsequences of the sample increase. In that case, the mean frequency of sample patterns is equal to $8.65$.

### B. Accuracy of sampling-based classification

This section shows how sampled subsequences can be used to build associative classifiers dedicated to sequences. Our classification method, called CSSAMPLING+SVM, is a standard two-step approach. In a first step, using a sample $F = \{f_1, \ldots, f_k\}$ of $k$ subsequences obtained using CSSAMPLING, a labeled sequential dataset $\mathcal{S}$ is recoded into a numerical dataset $\mathcal{D}$. More precisely, for each sequence $s \in \mathcal{S}$ labeled by a class $c$, $\mathcal{D}$ contains a tuple of $k+1$ values where $t[j] = 1$ if $f_j \sqsubseteq s$ (0 otherwise) for $j \in [1..k]$, and $t[k+1] = c$. Then, in a second step, using dataset $\mathcal{D}$, we propose to use a SVM as

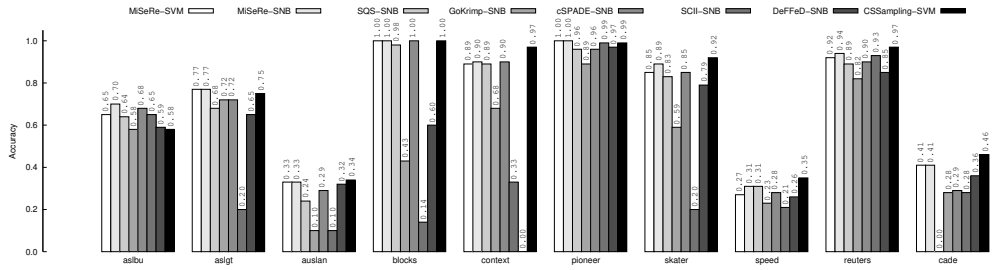



Fig. 3. Comparison of accuracy results between CSSAMPLING with SVM and state-of-the-art sequence classification methods.

TABLE VII. STATISTICS OF BENCHMARK DATASETS

| Dataset | $\|\mathcal{S}\|$ | $\|\mathcal{I}\|$ | $\|S\|_{max}$ | $\|S\|_{mean}$ | $\|\mathcal{C}\|$ |
|---|---|---|---|---|---|
| aslbu | 441 | 132 | 27 | 7.52 | 7 |
| aslgt | 3,493 | 87 | 88 | 22.83 | 40 |
| auslan | 200 | 12 | 24 | 10.00 | 10 |
| blocks | 210 | 8 | 12 | 6.75 | 8 |
| context | 240 | 48 | 123 | 45.20 | 5 |
| pioneer | 160 | 92 | 50 | 21.07 | 3 |
| skater | 530 | 41 | 120 | 25.06 | 6 |
| speed | 530 | 41 | 260 | 64.50 | 7 |
| reuters | 5,459 | 14,577 | 533 | 67.32 | 8 |
| cade | 15,000 | 100,197 | 15,318 | 112.70 | 12 |

classifier for predicting the class of new sequences. Note that in our experiments, we use the SMO algorithm provided by Weka 3.8 and its default options to build SVM classifiers.

In order to evaluate the efficiency of CSSAMPLING+SVM, we use a set of real-world datasets [30][4] that have a wide variety in the number of sequences, items, sequence lengths and classes as well as application domains (see Table VII). For each dataset, we calculate the accuracy of CSSAMPLING+SVM with respect to varied sample sizes and norm constraints, by performing a 10-fold cross-validation.

TABLE VIII. IMPACT OF THE NORM CONSTRAINT ON CLASSIFICATION

| Dataset | $M$=1 | $M$=2 | $M$=3 | $M$=5 | $M$=7 | $M$=10 | **Best** |
|---|---|---|---|---|---|---|---|
| aslbu | 0.57 | **0.58** | 0.56 | 0.55 | 0.42 | 0.38 | 0.58 |
| aslgt | 0.73 | **0.75** | **0.75** | 0.72 | 0.59 | 0.43 | 0.75 |
| auslan | 0.24 | 0.24 | **0.34** | 0.32 | 0.32 | 0.32 | 0.34 |
| blocks | 0.86 | **1.00** | 0.99 | 0.99 | 0.99 | 0.99 | 1.00 |
| context | 0.94 | 0.96 | **0.97** | **0.97** | 0.96 | 0.95 | 0.97 |
| pioneer | **0.99** | **0.99** | 0.98 | 0.87 | 0.74 | 0.66 | 0.99 |
| skater | 0.84 | 0.90 | **0.92** | **0.92** | 0.88 | 0.73 | 0.92 |
| speed | 0.24 | 0.29 | **0.35** | **0.35** | **0.35** | 0.23 | 0.35 |
| reuters | **0.97** | 0.95 | 0.85 | 0.56 | 0.52 | 0.52 | 0.97 |
| cade | **0.46** | 0.33 | 0.25 | 0.22 | 0.22 | 0.21 | 0.46 |
| Average | 0.68 | **0.70** | **0.70** | 0.69 | 0.64 | 0.58 | 0.76 |

*1) Importance of the norm constraint:* As described in previous sections, the norm constraint $M$ is introduced to limit the maximal length of sampled subsequences since too long patterns have been proved less useful in pattern discovery. Table VIII shows that the accuracy of CSSAMPLING+SVM clearly depends on the norm constraint. While the total size of sample is fixed (here, 10,000 patterns), the best classification performance is generally obtained when the maximum norm threshold is strictly larger than 1 (except for datasets `reuters` and `cade`, as observed in [30]) and lower that 10. Given a dataset, the optimal value of $M$ (**Best** column in Table VIII) can be easily identified using cross-validation (evaluating the performance of CSSAMPLING+SVM for $M \in [1..10]$). Finally, note that the performance of classifiers decreases with $M$ when $M$ is greater that its optimal value, which shows the importance to consider maximum norm thresholds to build efficient classifiers. In particular, the performance of classifiers that would be obtained without considering norm constraints (i.e., $M \to \infty$) would therefore be very low.

*2) Comparison with pattern-based sequence classification methods:* We finally compare the accuracy of CSSAMPLING+SVM with the results of 7 state-of-the-art sequence classification methods reported in [30] as baselines with respect to the same datasets: MISERE, SQS, GOKRIMP, CSPADE, SCII and DEFFED. Figure 3 shows that the best accuracies obtained by CSSAMPLING+SVM (column **Best** of Table VIII) are comparable, even better according to datasets, to other pattern-based sequence classification methods reported in [30]. Notice that the goal of this paper is not to propose a new sequence classification method, we just want to illustrate that subsequence sampling is useful in some applications.

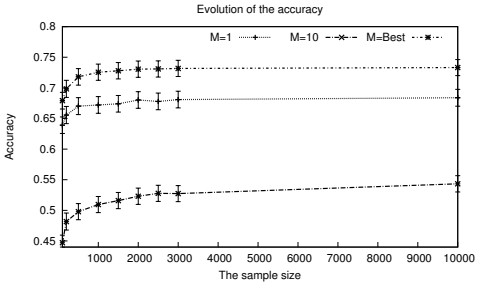



Fig. 4. Impact of the sample size on classification performance.

*3) Impact of the sample size:* Depending on applications, in particular to classification tasks, the impact of sample size shall not be ignored with our classification method. Obviously, the accuracy of the classification increases with the sample size because the sequences are more likely to be covered by at least one subsequence. Figure 4 shows the classification

[4]The datasets `reuters` and `cade` are available at `ana.cachopo.org/datasets-for-single-label-text-categorization` and other ones, at `www.mybytes.de/#data`.

performance, considered as average accuracy values over all datasets, obtained by different sample sizes with respect to norm constraint values 1, 10 and **Best** mentioned in Table VIII. It is easy to observe that the classification performance increases while more sampled sequential patterns are involved (which is useful for developing an anytime approach). Interestingly, the accuracy increases very quickly with the sample size. Thus a classifier built in a short response time considering only 1,000 subsequences competes with methods of the state of the art where all the pattern search space is explored.

## VI. Conclusion

This paper proposes the first output space sampling method for sequential patterns. It also allows to specify an interval constraint on the norm of sequential patterns to better control the returned patterns. We have demonstrated that our sampling algorithm is exact and we have estimated its efficiency with respect to the average number of rejections which increases with the number of occurrences within a sequence. The experimental study shows that the approach is very efficient on real-world datasets where the number of repetitions is low. Moreover, the experiments show that the addition of constraints on the norm avoids returning too many patterns too rare and focuses the sampling on the patterns of the "head" as desired. Finally, we illustrated how to build a classifier in a very short response time by just drawing a sample containing 1,000 patterns. These models still have an accuracy comparable to some methods achieving a complete enumeration of the pattern search space.

We would like to extend our approach to other interestingness measures and to any set system. First, the draw weight of a sequence could be calculated for interestingness measures $u(s) \times freq(s, \mathcal{S})$ (where the utility $u$ depends only on the sequence norm) because the utility can be integrated into the subsequence counting formulas. Second, the uniform drawing within complex structures made possible by a canonical form (here the first occurrence) can be envisaged with other structured languages. As was the case with the itemsets, we think that the results about associative classification are promising for addressing other data mining tasks like detecting outliers in sequential data [6] or for designing interactive systems dedicated to sequential pattern discovery [3].

*Acknowledgements.* This work has been partly supported by the CEA-MITIC (Centre d'Excellence Africain en Mathématiques, Informatique et TIC).

## References

[1] M. van Leeuwen, "Interactive data exploration using pattern mining," in *Interactive Knowledge Discovery and Data Mining in Biomedical Informatics*, 2014, pp. 169–182.

[2] S. Zilberstein, "Using anytime algorithms in intelligent systems," *AI magazine*, vol. 17, no. 3, p. 73, 1996.

[3] M. Bhuiyan, S. Mukhopadhyay, and M. A. Hasan, "Interactive pattern mining on hidden data: a sampling-based solution," in *Proc. of CIKM 2012*, 2012, pp. 95–104.

[4] A. Giacometti and A. Soulet, "Interactive pattern sampling for characterizing unlabeled data," in *Proc. of IDA 2017*, 2017, pp. 99–111.

[5] V. Dzyuba, M. v. Leeuwen, S. Nijssen, and L. De Raedt, "Interactive learning of pattern rankings," *Int. Journal on Artificial Intelligence Tools*, vol. 23, no. 06, p. 32 pages, 2014.

[6] A. Giacometti and A. Soulet, "Anytime algorithm for frequent pattern outlier detection," *International Journal of Data Science and Analytics*, vol. 2, no. 3-4, pp. 119–130, 2016.

[7] M. Al Hasan and M. J. Zaki, "Output space sampling for graph patterns," *Proc. of the VLDB*, vol. 2, no. 1, pp. 730–741, 2009.

[8] R. Agrawal and R. Srikant, "Mining sequential patterns," in *Proc. of ICDE 95*, 1995, pp. 3–14.

[9] J. Han, H. Cheng, D. Xin, and X. Yan, "Frequent pattern mining: current status and future directions," *Data mining and knowledge discovery*, vol. 15, no. 1, pp. 55–86, 2007.

[10] C. Anderson, "The long tail," *Wired magazine*, vol. 12, no. 10, pp. 170–177, 2004.

[11] M. Boley, C. Lucchese, D. Paurat, and T. Gärtner, "Direct local pattern sampling by efficient two-step random procedures," in *Proc. of SIGKDD 2011*, 2011, pp. 582–590.

[12] R. Srikant and R. Agrawal, "Mining sequential patterns: Generalizations and performance improvements," in *Proc. of EDBT 96*, 1996, pp. 3–17.

[13] M. J. Zaki, "SPADE: An efficient algorithm for mining frequent sequences," *Machine Learning*, vol. 42, no. 1-2, pp. 31–60, 2001.

[14] J. Pei, J. Han, B. Mortazavi-Asl, and H. Pinto, "PrefixSpan: Mining sequential patterns efficiently by prefix-projected pattern growth," in *Proc. of ICDE 2001*, 2001, pp. 215–224.

[15] M. N. Garofalakis, R. Rastogi, and K. Shim, "Spirit: Sequential pattern mining with regular expression constraints," in *VLDB*, vol. 99, 1999, pp. 7–10.

[16] J. Pei, J. Han, and L. V. Lakshmanan, "Mining frequent itemsets with convertible constraints," in *Proc. of ICDE 2001*. IEEE, 2001, pp. 433–442.

[17] J. Wang and J. Han, "Bide: Efficient mining of frequent closed sequences," in *Proc. of ICDE 2004*. IEEE, 2004, pp. 79–90.

[18] X. Yan, J. Han, and R. Afshar, "Clospan: Mining: Closed sequential patterns in large datasets," in *Proc. of SDM 2003*. SIAM, 2003, pp. 166–177.

[19] G. Bosc, J.-F. Boulicaut, C. Raïssi, and M. Kaytoue, "Anytime discovery of a diverse set of patterns with monte carlo tree search," *Data Mining and Knowledge Discovery*, pp. 1–47, 2016.

[20] H. Toivonen *et al.*, "Sampling large databases for association rules," in *Proc. of VLDB 96*, vol. 96, 1996, pp. 134–145.

[21] C. Luo and S. M. Chung, "A scalable algorithm for mining maximal frequent sequences using sampling," in *Proc. of ICTAI 2004*. IEEE, 2004, pp. 156–165.

[22] C. Raissi and P. Poncelet, "Sampling for sequential pattern mining: From static databases to data streams," in *Proc. of ICDM 2007*, 2007, pp. 631–636.

[23] A. A. Bendimerad, M. Plantevit, and C. Robardet, "Unsupervised exceptional attributed sub-graph mining in urban data," in *Proc. of ICDM 2016*. IEEE, 2016, pp. 21–30.

[24] A. Giacometti and A. Soulet, "Dense neighborhood pattern sampling in numerical data," in *Proc. of SDM 2018*, 2018, pp. 756–764.

[25] S. Moens and B. Goethals, "Randomly sampling maximal itemsets," in *Proc. of IDEA Workshop 2013*, 2013, pp. 79–86.

[26] M. Boley, T. Gärtner, and H. Grosskreutz, "Formal concept sampling for counting and threshold-free local pattern mining," in *Proc. of SDM 2010*. SIAM, 2010, pp. 177–188.

[27] S. Moens and M. Boley, "Instant exceptional model mining using weighted controlled pattern sampling," in *Proc. of IDA 2014*. Springer, 2014, pp. 203–214.

[28] V. Dzyuba, M. van Leeuwen, and L. De Raedt, "Flexible constrained sampling with guarantees for pattern mining," *Data Mining and Knowledge Discovery*, vol. 31, no. 5, pp. 1266–1293, 2017.

[29] E. Egho, C. Raïssi, T. Calders, N. Jay, and A. Napoli, "On measuring similarity for sequences of itemsets," *Data Mining and Knowledge Discovery*, vol. 29, no. 3, pp. 732–764, 2015.

[30] E. Egho, D. Gay, M. Boullé, N. Voisine, and F. Clérot, "A user parameter-free approach for mining robust sequential classification rules," *Knowl. Inf. Syst.*, vol. 52, no. 1, pp. 53–81, 2017.

# Representativeness of Knowledge Bases with the Generalized Benford's Law

Arnaud Soulet[1], Arnaud Giacometti[1], Béatrice Markhoff[1], and Fabian M. Suchanek[2]

[1] Université de Tours, LIFAT
firstname.lastname@univ-tours.fr
[2] Telecom ParisTech, LTCI
suchanek@telecom-paristech.fr

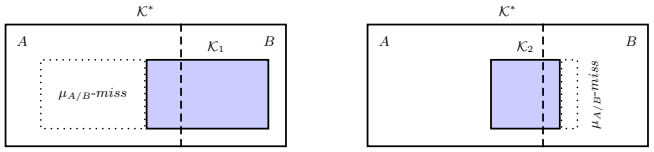
**Abstract.** Knowledge bases (KBs) such as DBpedia, Wikidata, and YAGO contain a huge number of entities and facts. Several recent works induce rules or calculate statistics on these KBs. Most of these methods are based on the assumption that the data is a representative sample of the studied universe. Unfortunately, KBs are biased because they are built from crowdsourcing and opportunistic agglomeration of available databases. This paper aims at approximating the representativeness of a relation within a knowledge base. For this, we use the generalized Benford's law, which indicates the distribution expected by the facts of a relation. We then compute the minimum number of facts that have to be added in order to make the KB representative of the real world. Experiments show that our unsupervised method applies to a large number of relations. For numerical relations where ground truths exist, the estimated representativeness proves to be a reliable indicator.

## 1 Introduction

One of the undisputed successes of the Semantic Web is the construction of huge knowledge bases (KBs). Several recent works use these KBs to derive new knowledge by calculating statistics or deducing rules from the data [7,26,27,29]. For instance, according to DBpedia, 99% of the places in Yemen have a population of more than 1,000 inhabitants. Thus, we could conclude that Yemeni cities usually have more than 1,000 inhabitants. But is that true in the real world?

Naturally, the reliability of such conclusions depends on the quality of the knowledge base [34] namely its correctness (accuracy of the facts) and its completeness. It is well known that KBs are highly incomplete. This is usually not a problem in statistics and in machine learning, where it is rare to have a complete description of the universe under study. Most approaches work on a sample of the data. In such cases, it is crucial that this sample is representative of the entire universe (or at least, that the bias of this sample is known). For example, it is not a problem if the KB contains only half of the cities of Yemen, if their distribution across different sizes corresponds roughly to the distribution in the real world. Figure 1 illustrates this: there is an ideal knowledge base $\mathcal{K}^*$ divided

into two classes $A$ and $B$ that correspond respectively to the places with less than 1,000 inhabitants and other places. The KB $\mathcal{K}_1$ is more complete than the KB $\mathcal{K}_2$. However, $\mathcal{K}_2$ better reflects the distribution between the two classes.




(a) More complete, less representative (b) Less complete, more representative

**Fig. 1.** Completeness vs Representativeness

Unfortunately, it is not clear whether the data in KBs is representative of the real world. For example, several large KBs, such as DBpedia [2] or YAGO [28], extract their data from Wikipedia. Wikipedia, in turn, is a crowdsourced dataset. In crowdsourcing, contributers tend to state the information that interests them most. As a result, Wikipedia exhibits some cultural biases [6,33]. Inevitably, these biases are reflected in the KBs. For instance, 3,922 entities in DBpedia concern the American company "Disney", which is almost as much as the 4,493 entities concerning Yemen (a country with more than 26 million inhabitants). Wikidata [32], likewise, is the result of crowdsourcing, and may exhibit similar biases. In particular, it is likely that countries such as Yemen are less evenly covered than places such as France – due to the population of contributors. Even if the information in these KBs is correct [13], it is not necessarily representative. If we knew how representative a certain KB is, then we could know whether it is reasonable or not to exploit it for deriving statistics. Such an indication should, for example, prevent us from drawing hasty conclusions about the distribution of the population in the cities of Yemen. But, how to estimate whether a knowledge base is representative or not?

This paper proposes to study the representativeness of knowledge bases by help of the generalized Benford's law. This parameterized law indicates the frequency distribution expected by the first significant digit in many real-world numerical datasets. We use this law as a gold standard to estimate how much data is missing in the KB. More specifically, our contributions are as follows:

- We present a method to calculate a lower bound for the number of missing facts for a relation to be representative. This method works in a supervised context (where the relation is known to satisfy the generalized Benford's law), and in an unsupervised context (where the parameter of the law has to be deduced from the data).
- We prove that, under certain assumptions, the calculated lower bounds are correct both in the supervised and the unsupervised context.

– We show with experiments on real KBs that our method is effective for supervised contexts as well as for unsupervised contexts. The unsupervised method, in particular, can audit 63% of DBpedia's facts.

This paper is structured as follows. Section 2 reviews some related work. Section 3 introduces the basic notions of representativeness. In Section 4, we propose our method for approximating representativeness based on the generalized Benford's law. Section 5 provides experimental results. We conclude in Section 6.

## 2 Related Work

To the best of our knowledge, the representativeness of knowledge bases with respect to the real world has not yet been studied. Nevertheless, as mentioned in the introduction, this problem is related to the completeness of KBs.

*Completeness.* Several recent works have studied the completeness of KBs [25,34]. Some works propose to manually add information about the completeness relations [8]. Other approaches mine rules on the data [12] (e.g., people usually live in the city where they work) and propose to add this information where it is missing. For this purpose, the work of [12] makes the Partial Completeness Assumption (PCA): It assumes that, if the KB contains at least one object for a given relation and a given subject, then it contains all of the objects for this context. The PCA has been shown to be reasonably accurate in practice [12]. Newer approaches for rule mining take into account the cardinality of the relations, if it is known [30]. Other work aims to determine more generally whether all objects of a certain relation for a certain subject are present in the KB [11]. For this, the approach uses oracles, such as the PCA and the popularity of the subject in Wikipedia. Again other work [1,14,17,31] mines class descriptions. Such approaches are able to determine that a certain attribute is obligatory for a class – and then allow estimating the number of missing facts per class.

All of these approaches are concerned with completeness in terms of facts with respect to the present entities. Our approach, in contrast, also considers the facts of entities that are missing. Furthermore, none of the above works studies the representativeness of the KB, i.e., whether or not the distribution of entities in the KB corresponds to the distribution in the real world.

*Representative sample.* Completeness is an important notion for estimating the quality of a knowledge base, but it is not necessarily the best indicator when one wants to measure the quality of a distribution. In statistics, several resampling techniques [9] exist to estimate the quality of a sample (median, variance, quantile), in particular by analyzing the evolution of a measure on a subsample or by permuting labels. None of these techniques can be used to check whether a single sample is representative, if the ground truth is unknown – as it is the case in our scenario.

*Benford's law.* When the data is complete, Benford's law [4] is regularly used to detect inconsistencies within the data [22]. If the distribution of the first significant digit of some numerical dataset does not satisfy Benford's law, then

the data is assumed to be faulty. For this reason, Benford's law is regularly used to detect frauds in various kind of data: in accounts [23], in elections [19], or in wastewater treatment plant discharge data [3]. However, in all of these cases, Benford's law is used only to estimate the correctness of the data – not its completeness. The work cannot be used, e.g., to decide how many facts are missing in a KB, or whether a KB is representative of the real world.

## 3 Preliminaries

### 3.1 Representativeness of knowledge bases

For our purposes, a knowledge base (KB) over a set of relations $\mathcal{R}$ and a set of constants $\mathcal{C}$ (representing entities and literals) is a set of *facts* $\mathcal{K} \subseteq \mathcal{R} \times \mathcal{C} \times \mathcal{C}$. We write facts as $r(s, o) \in \mathcal{K}$, where $r$ is the relation, $s$ is the subject, and $o$ is the object. The set of facts for the relation $r$ in $\mathcal{K}$ is denoted by $\mathcal{K}_{|r} = \{r(s, o) \in \mathcal{K}\}$. Given a relation $r$, $r^{-1}(o, s) \in \mathcal{K}$ means that $r(s, o) \in \mathcal{K}$ where $r^{-1}$ is the inverse relation of $r$.

In line with the other work in the area [11,17,18,21,24], we denote with $\mathcal{K}^*$ a hypothetical ideal KB, which contains all facts of the real world. Then, the completeness (also called recall) of $\mathcal{K}$, denoted $comp(\mathcal{K})$, is the proportion of facts of $\mathcal{K}^*$ present in $\mathcal{K}$: $comp(\mathcal{K}) = |\mathcal{K} \cap \mathcal{K}^*|/|\mathcal{K}^*|$. For our work, we will make the following assumption:

**Assumption 1 (Correctness)** *Given a knowledge base $\mathcal{K}$, we assume that all facts of $\mathcal{K}$ are correct i.e., $\mathcal{K} \subseteq \mathcal{K}^*$.*

The correctness assumption is a strong assumption. It has been investigated in [28,34]. In our work, we use it mainly for our theoretical model. Our experiments will show that our method delivers good results even with some amount of noise in the data. Let us now introduce the notion of a *uniform-sampling invariant measure*. A measure $\mu$ maps a knowledge base $\mathcal{K}$ to a frequency vector $(f_1, \ldots, f_n) \in \mathbb{R}^n_{\geq 0}$ where each component $f_i$ is the number of observations of the $i$th characteristic in $\mathcal{K}$. Given a non-zero frequency vector $F = (f_1, \ldots, f_n)$, $\overline{f_i}$ denotes the normalized $i$th component of $F$ where $\overline{f_i} = f_i / \sum_{i=1}^{n} f_i$. We use the mean absolute deviation (MAD) for comparing two non-zero frequency vectors $F = (f_1, \ldots, f_n)$ and $F' = (f'_1, \ldots, f'_n)$:

$$MAD(F, F') = \frac{1}{n} \sum_{i=1}^{n} \left| \overline{f_i} - \overline{f'_i} \right|$$

$F$ and $F'$ are similar for $\epsilon \ll 1$ iff $MAD(F, F') \leq \epsilon$. In such case, we write $F \sim_\epsilon F'$, or simply $F \sim F'$. A measure $\mu$ is uniform-sampling invariant iff for any uniform sample $\mathcal{K}'$ from $\mathcal{K}$ such that $|\mathcal{K}'| \gg 1$, we have $\mu(\mathcal{K}') \sim \mu(\mathcal{K})$. For instance, in Figure 1, counting the number of places with less than 1,000 inhabitants (in part $A$) and more than 1,000 inhabitants (in part $B$) is a measure with two characteristics (denoted by $\mu_{A/B}$). The measure $\mu_{A/B}$ is uniform-sampling

invariant because whatever the uniform sample of a knowledge base $\mathcal{K}$, the proportion of cities with more (or less) than 1,000 inhabitants remains the same. In the following, we consider only uniform-sampling invariant measures.

A knowledge base is *representative* if each measure returns a frequency vector that is proportional to the frequency vector on $\mathcal{K}^*$:

**Definition 1 (Representative KB).** *A knowledge base $\mathcal{K}$ is representative of $\mathcal{K}^*$ iff $\mu(\mathcal{K}) \sim \mu(\mathcal{K}^*)$ for any uniform-sampling invariant measure $\mu$.*

If a knowledge base $\mathcal{K}$ is unrepresentative, there is at least one measure $\mu$ such that $\mu(\mathcal{K}) \nsim \mu(\mathcal{K}^*)$. In this case, since all the facts of $\mathcal{K}$ are correct (Assumption 1), it would be necessary to add new facts to the knowledge base to make it representative for $\mu$. Formally, this number of missing facts of $\mathcal{K}$ for the measure $\mu$, denoted by $\mu\text{-}miss(\mathcal{K})$, is defined as:

$$\mu\text{-}miss(\mathcal{K}) = \min\{|F| : F \subseteq \mathcal{K}^* \wedge \mu(\mathcal{K} \cup F) \sim \mu(\mathcal{K}^*)\}$$

The number of missing facts in $\mathcal{K}$, denoted by $miss(\mathcal{K})$, is the minimum number of facts that have to be added to make the KB representative (whatever the considered measure $\mu$): $miss(\mathcal{K}) = \max_\mu \mu\text{-}miss(\mathcal{K})$. The representativeness of $\mathcal{K}$ estimates whether $\mathcal{K}$ is a representative sample of $\mathcal{K}^*$:

**Definition 2 (Representativeness).** *The representativeness of $\mathcal{K}$, denoted $rep(\mathcal{K})$, is defined as:*

$$rep(\mathcal{K}) = \frac{|\mathcal{K}|}{|\mathcal{K}| + miss(\mathcal{K})}$$

Interestingly, a KB can be representative without being complete. The representativeness of $\mathcal{K}$ is an upper bound of the completeness: $rep(\mathcal{K}) \geq comp(\mathcal{K})$.

### 3.2 Problem statement

The goal of this paper is to approximate the representativeness of a relation $r$ in $\mathcal{K}$ (i.e., the representativeness of $\mathcal{K}_{|r}$) without having a reference knowledge base $\mathcal{K}^*_{|r}$ (which is the most common case in a real-world scenario). This task is ambitious because the calculation of the representativeness of a knowledge base requires to know the distribution of any measure $\mu$ on an unknown knowledge base $\mathcal{K}^*_{|r}$. It is obviously not possible to know the distribution $\mu(\mathcal{K}^*_{|r})$ for any measure. In order to calculate an approximation, we propose to use the following observation, which holds for all measures $\mu$:

$$\mu\text{-}miss(\mathcal{K}_{|r}) \leq miss(\mathcal{K}_{|r})$$

This result (which follows from the definition of $miss(\mathcal{K}_{|r})$) means that it is possible to get a lower bound $l$ of the number of missing facts $miss(\mathcal{K}_{|r})$, if some distributions $\mu_i(\mathcal{K}^*_{|r})$ are known. Such a lower bound is useful for calculating an upper bound of the representativeness and the completeness of the knowledge base: $|\mathcal{K}_{|r}|/(|\mathcal{K}_{|r}| + l)$.

**Given a knowledge base $\mathcal{K}$ and a relation $r$, we aim at estimating the representativeness of the relation $r$ in the knowledge base $\mathcal{K}$ by finding a lower bound $l$ such that $l \leq miss(\mathcal{K}_{|r})$.**

## 4 Our Approach

### 4.1 The generalized Benford's law for KBs

The challenge is to find a set of measures whose distribution is known on the ideal knowledge base $\mathcal{K}^*$. To this end, we propose to rely on Benford's law [4]. This law says that, in many natural datasets, the first significant digit of the numbers is unevenly distributed: Around 30% of numbers will start with a "1", whereas only 5% of numbers will start with a "9". This somehow surprising result follows from the fact that many natural numbers follow a multiplicative growth pattern. For example, a city of 1000 inhabitants may grow by 30% each year, thus passing by the values of 1300, 1690, 2197, 2856, 3712, 4826, 6274, 8157, 10604. These values already show a skewed distribution of the first digit, which will repeat itself in the coming years. There are other reasons for such patterns, and Benford's law has since been observed not just for population sizes, but also for prices, stock markets, death rates, lengths of rivers, and many other real-world phenomena [4] – although not all [20]. Technically, Benford's law is a statistical frequency distribution on the first significant digit of a set of numbers, which may or may not apply to a given dataset. In this paper, we use the generalized Benford's law [16], which is parametrized and can thus apply to more datasets.

**Definition 3 (Generalized Benford's Law [15]).** *A set of numbers is said to satisfy a generalized Benford's law (GBL) with exponent $\alpha \neq 0$ if the first digit $d \in [1..9]$ occurs with probability:*

$$B_d^\alpha = \frac{(1+d)^\alpha - d^\alpha}{10^\alpha - 1}$$

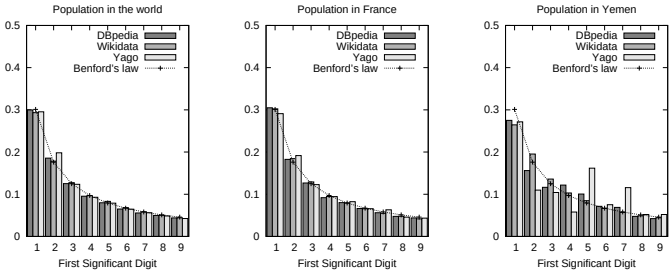



**Fig. 2.** First significant digit distribution for population

The parameter $\alpha$ adds a great flexibility since the choice of this value makes it possible to find Benford's law ($\alpha \to 0$) and the uniform law ($\alpha = 1$). Data

that follows a power law $ax^{-k}$ also follows the GBL approximately with $\alpha = -1/k$ [15]. This is, e.g., the case for the out-degree of Web pages [5], with $k = 2.6$.

The GBL can be applied to KBs. Let us look at the relation `pop`, which links a geographical place to its number of inhabitants (`populationTotal` in DBpedia, `P1082` in Wikidata, and `hasNumberOfPeople` in YAGO). Figure 2 shows the distribution of first digits of this relation, drilled down to places in the world, in France, and in Yemen. We see that the distribution in the KB roughly follows the GBL. Interestingly, the GBL applies better to the French population than to the Yemeni population. We will now take advantage of this information to measure representativeness.

Technically, Figure 2 presents the frequency vector $(f_1, \ldots, f_9)$ of the first digits of the relation `pop`. Of course, it is not possible to directly calculate the ideal frequency vector $(f_1^*, \ldots, f_9^*)$ of $\mathcal{K}^*$. However, in many cases, we know at least the distribution of the ideal frequency vector (thanks to the GBL). If we do not know the distribution, then our idea is to *learn* the exponent $\alpha$ of the GBL from the observed vector. Once the ideal distribution has been determined, we can use the difference between the observed distribution and the estimated distribution to bound the number of missing facts (Figure 3).

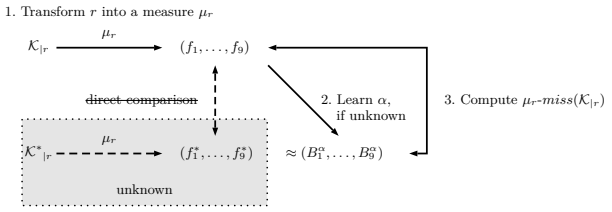



**Fig. 3.** Overview of the method

More precisely, we propose to proceed as follows:

1. **Transforming a relation into a measure:** Benford's law can only work on numerical datasets. Some relations (such as `pop`) are already numerical. Other relations will have to be transformed into numerical datasets (Section 4.2).
2. **Parameterizing the GBL:** To use the GBL, we have to know the parameter $\alpha$. We distinguish two contexts. In a *supervised* context, the parameter $\alpha$ is known upfront in the real world (as it is the case for the population). Otherwise, in an *unsupervised* context, we learn the parameter $\alpha$ that best fits the facts in $\mathcal{K}_{|r}$ assuming it is close to the ideal parameter $\alpha^*$ on $\mathcal{K}^*_{|r}$ (Section 4.3).
3. **Estimating the number of missing facts:** As the knowledge base is correct, only the addition of new facts would make the frequency vector $(f_1, \ldots, f_9)$ coincide with the distribution of $(B_1^\alpha, \ldots, B_9^\alpha)$ which is (approxi-

mately) proportional to $(f_1^*, \ldots, f_9^*)$. The objective of this last step is to calculate the minimum number of facts to add so that $(f_1, \ldots, f_9) \sim (B_1^\alpha, \ldots, B_9^\alpha)$ (Section 4.4).

In the following, when we consider a relation $r$, $\mathcal{K}$ implicitly refers to $\mathcal{K}_{|r}$.

### 4.2 Transforming relations into measures

We show in this section how to transform a relation $r$ into a measure $\mu_r$. The key idea is to transform each relation $r$ into a set of numbers $N_r$ that is a kind of digital signature. Then, we derive a measure $\mu_r$ that counts the frequency of each number in $N_r$ having $d$ as first significant digit:

$$\mu_r(\mathcal{K}) = (\#n : \text{the first significant digit of } n \in N_r(\mathcal{K}) \text{ is equal to } d)_{d \in [1..9]}$$

In our example with the relation `pop`, the measure $\mu_{\texttt{pop}}$ counts the number of places that have a population with $d$ as first significant digit. Let us now generalize this principle to two common types of relations:

– **Numerical transformation:** Given a numerical relation $r$, the numerical transformation keeps all the numbers different from 0:

$$N_r^{num}(\mathcal{K}) = \{number : r(s, number) \in \mathcal{K} \wedge number \neq 0\}$$

Figure 2 illustrates this transformation for relation `pop` by showing the frequency vector resulting from $\mu_{\texttt{pop}}$.

– **Counting transformation:** Given a relation $r$, the counting transformation returns for each object $o$ how many facts it has:

$$N_r^{count}(\mathcal{K}) = \{\#s : r(s, o) \in \mathcal{K} \text{ such that } o \text{ is an object of a fact in } \mathcal{K}_{|r}\}$$

For example, for the relation `starring`, we can count the number of movies for each actor. The left hand-side of Figure 4 illustrates the resulting frequency vector. We choose to count the number of subjects rather than the number of objects, because relations tend to have more subjects per object than vice versa [12]. However, we can also count the number of objects per subject by applying the above method to $r^{-1}$. Figure 4 shows two other histograms, one for the relation `team` (number of players per team) and for `birthPlace` (number of births per place).

This list of transformations is not exhaustive. For instance, it would be possible to count the number of days since today for a date (e.g. for the birth date relation) or to consider the length of strings. Besides, it is possible to transform the same relation in several ways. In this way, it is possible to obtain more frequency vectors.

### 4.3 Parameterizing the generalized Benford's law

The previous section has given us a measure $\mu_r$ that we can apply on the knowledge base $\mathcal{K}$ to calculate a distribution. Now, we want to compare this distribution with the distribution on the ideal KB $\mathcal{K}^*$. This requires knowledge of the

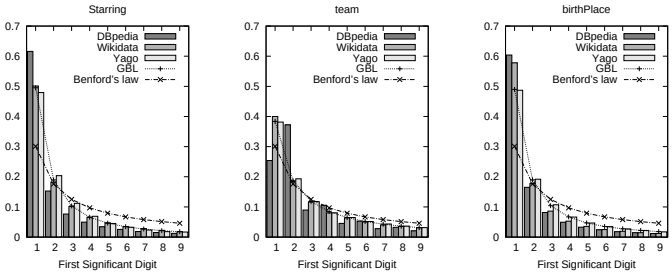



**Fig. 4.** Examples of measures resulting from counting transformation

parameter $\alpha$, which depends on the unknown distribution $\mu_r(\mathcal{K}^*)$. We distinguish two settings.

*Supervised setting.* In some cases, it is known that $\mu_r(\mathcal{K}^*)$ follows the GBL in the real world with a certain parameter $\alpha$. For instance, the population of places, the length of rivers, etc. conform to the GBL in the real world with an exponent tending to 0 (see Table 2 below). In that case, the GBL is already parametrized.

*Unsupervised setting.* If it is not known whether $\mu_r(\mathcal{K}^*)$ follows the GBL, or if its parameter $\alpha$ is not known, we propose to estimate it from the KB. For this purpose, we make the following assumption:

**Assumption 2 (Transferability)** *Given a knowledge base $\mathcal{K}$, we assume that if $\mathcal{K}$ conforms to the GBL with exponent $\alpha$, then the ideal knowledge base $\mathcal{K}^*$ also conforms to the GBL with exponent $\alpha$.*

This assumption may seem strong. However, it is verified in several cases where we have a ground truth available (see experiments in Section 5). The assumption allows us to learn the parameter $\alpha$ that best fits the facts in $\mathcal{K}$. Let us denote by $(f_1, \ldots, f_9)$ the characteristic vector resulting from $\mu_r(\mathcal{K})$ i.e., $f_d$ is exactly the number of occurrences in $N_r(\mathcal{K})$ with $d$ as first significant digit. Let us denote $N = \sum_{d=1}^{9} f_d$. To choose the right parameter $\alpha$, we use the WLS measure (probability weighted least square or Chi square statistics) as goodness-of-fit measure [15]:

$$WLS_{(f_1,\ldots,f_9)}(\alpha) = \sum_{d=1}^{9} \frac{\left(B_d^\alpha - \frac{f_d}{N}\right)^2}{B_d^\alpha}$$

Now, choosing the right parameter $\alpha$ means minimizing the WLS measure for the frequency vector $(f_1, \ldots, f_9)$. For this, we use the gradient descent algorithm. For instance, Figure 4 shows the gap between the GBL and Benford's law for the three relations. For `starring`, $\alpha$ is -1.156 (in DBpedia), -0.759 (in Wikidata)

and -0.750 (in YAGO). Once the parameter $\alpha$ has been obtained, we have to assess whether the frequency vector $\mu_r(\mathcal{K})$ conforms to the generalized Benford's law. For this, we use the mean absolute deviation (MAD) defined in Section 3.1. To know whether the GBL can be used according to the MAD estimator, we distinguish four cases [16,22]: close conformity (C) when $MAD \leq 0.006$, acceptable conformity (AC) when $0.006 < MAD \leq 0.012$, marginal conformity (MC) when $0.012 < MAD \leq 0.015$, and nonconformity (NC) otherwise. In our running examples, the measure $\mu_{\texttt{pop}}$ gives rise to a nonconformity only for Yemeni places in YAGO, because $\alpha = 0.351$ and $MAD(\mu_{\texttt{pop}}(\mathcal{K}), B^{0.351})$ equals 0.035 ($> 0.015$). If a measure $\mu_r$ leads to a nonconformity, then it is not possible to apply the GBL at all. In all other cases, we can estimate the number of missing facts for the relation $r$ as explained in the next section.

### 4.4 Estimating the number of missing facts

The purpose of this section is to estimate the number of missing facts for a relation $r$, knowing that we have an approximation of the expected distribution $(B_1^\alpha, \ldots, B_9^\alpha)$ that is proportional to $(f_1^*, \ldots, f_9^*)$. We assume that all the facts of the knowledge base $\mathcal{K}$ are correct (Assumption 1). Therefore, only the addition of facts can bring the observed distribution of facts $(f_1, \ldots, f_9)$ closer to the expected distribution $(B_1^\alpha, \ldots, B_9^\alpha)$.

*Numerical transformation.* When a relation is numerical, the only way to have a number with a given first significant digit is to add a new fact. Intuitively, it is then enough to add facts for each of the digits where the measured frequency is lower than the expected frequency. The following theorem formalizes this idea:

**Theorem 1.** *Given a knowledge base $\mathcal{K}$ and a measure $\mu_r^{num}$ such that $\mu_r^{num}(\mathcal{K}^*)$ satisfies a generalized Bendford's law with exponent $\alpha$, the number of missing facts for the relation $r$ is:*

$$\mu_r^{num}\text{-}miss(\mathcal{K}) = \max_{d \in [1..9]} \frac{f_d}{B_d^\alpha} - N$$

*where $(f_1, \ldots, f_9) = \mu_r(\mathcal{K})$ and $N = \sum_{d=1}^{9} f_d$.*

This follows from the fact that the expected distribution $f_d/(N+\mu_r^{num}\text{-}miss(\mathcal{K}))$ must be less than $B_d^\alpha$ for each digit $d$. Table 1 indicates the number of missing facts estimated for the relation `pop` with the unsupervised method, and deduces an approximation of the representativeness. Interestingly, the approximation $\mu_r^{num}\text{-}miss$ for Yemeni places of YAGO is very close to what we obtain in a supervised context (where we know that $\alpha \to 0$) – even though the measure is non-conform for that case. In the supervised context, we calculate that 181 facts are missing, while our estimation tells us that 127 facts are missing. Whatever the KB, our estimation of representativeness confirms our intuition mentioned in the introduction: the population of Yemeni places is less well informed than that of French ones.

| | Missing facts | | | Representativeness | | |
|---|---|---|---|---|---|---|
| Measure | DBpedia | Wikidata | YAGO | DBpedia | Wikidata | YAGO |
| $\mu^{num}_{\texttt{pop}}$ in World | 15,789 | 13,720 | 44,223 | 0.954 | 0.961 | 0.895 |
| $\mu^{num}_{\texttt{pop}}$ in France | 1,153 | 1,546 | 18,829 | 0.970 | 0.963 | 0.918 |
| $\mu^{num}_{\texttt{pop}}$ in Yemen | 78 | 4,281 | 127 (NC) | 0.829 | 0.888 | 0.577 (NC) |
| $\mu^{count}_{\texttt{starring}}$ | 51,179 | 10,370 | 2,703 | 0.892 | 0.989 | 0.979 |
| $\mu^{count}_{\texttt{team}}$ | 41,484 | 3,373 | 463 | 0.980 | 0.997 | 0.999 |
| $\mu^{count}_{\texttt{birthPlace}}$ | 38,664 | 25,691 | 470 | 0.971 | 0.986 | 0.998 |

**Table 1.** Representativeness of relations in three KBs (unsupervised context)

*Counting transformation.* For this transformation, the estimation of the number of missing facts is more complicated, because the addition of a fact for an object can change its first significant digit. For instance, if a number starting with 5 is missing, an object with 5 facts has to be added. One can imagine to add 5 new facts for a new object, to add four new facts for an object that has already 1 fact, to add 3 facts for an object that has already 2 facts, etc. We choose the solution that minimizes the total number of added facts:

**Theorem 2.** *Given a knowledge base $\mathcal{K}$ and a measure $\mu_r^{count}$ such that $\mu_r^{count}(\mathcal{K}^*)$ satisfies a generalized Bendford's law with exponent $\alpha$, the number of missing facts for the relation $r$ is:*

$$\mu_r^{count}\text{-}miss(\mathcal{K}) = \sum_{d=1}^{9} ((B_d^\alpha \times m) - f_d) \times d$$

*where* $m = \max_{d\in[1..9]} \frac{\sum_{i\geq d} f_i}{\sum_{i\geq d} B_i^\alpha}$ *and* $(f_1, \ldots, f_9) = \mu_r(\mathcal{K})$.

This follows from the fact that $\sum_{i\geq d} f_i/m \leq \sum_{i\geq d} B_i^\alpha$ for each digit $d$. For the unsupervised context, Table 1 indicates the number of missing facts estimated for the relations `starring`/ `team`/ `birthPlace` with our method and deduces an approximation of the representativeness.

Note that for the same relation $r$, under the two transformations leading to $\mu_r^{num}$ and $\mu_r^{count}$, the number of missing facts is bounded by the maximum result: $\max\{\mu_r^{num}\text{-}miss(\mathcal{K}); \mu_r^{count}\text{-}miss(\mathcal{K})\} \leq miss(\mathcal{K})$. Under the same transformation, the missing facts for two distinct relations $r_1$ and $r_2$ can be added together: $(\mu_{r_1}\text{-}miss(\mathcal{K}) + \mu_{r_2}\text{-}miss(\mathcal{K})) \leq miss(\mathcal{K})$. We will use these properties in Section 5.3 for DBpedia analysis.

### 4.5 Limitations of our approach

Using Theorems 1 and 2, our approach approximates the representativeness of some relation $r$ in the knowledge base $\mathcal{K}$ by finding a lower bound $\mu_r\text{-}miss(\mathcal{K})$ such that $\mu_r\text{-}miss(\mathcal{K}) \leq miss(\mathcal{K}_{|r})$ as requested in Section 3.2. This approach

works only if Assumption 1 (Correctness) holds. For the unsupervised setting, we also need Assumption 2 (Transferability).

Furthermore, for the GBL to be applicable, the set of numbers $N_r$ has to meet the following two conditions. First, the numbers of $N_r$ have to be distributed across several orders of magnitude: $\log_{10}\max(N_r) - \log_{10}\min(N_r) \geq 1$. For instance, the height of people does not meet this criterion because it is between 100 and 199 centimeters for most people. In that case, a numerical transformation would lead to a lot of "1" and "2" as first significant digits. For the same reason, it is also not possible to apply the counting transformation to an inverse functional relation $r$ because in that case, each object has only one subject (i.e., $N_r^{count} = \{1, 1, 1, \dots\}$) and then, its prevalence is 0. Second, the cardinality of $N_r$ has to be sufficiently high: $|N_r| \gg 1$. If we do not have enough numbers in $N_r$, the derived distributions $\mu_r(\mathcal{K})$ will not be reliable enough to learn the parameter $\alpha$. The next section will show where our method can be applied.

## 5 Experiments

These experiments answer the following three questions: Is the unsupervised method reliable? Is the representativeness estimated by our method correct? Is the GBL sufficiently effective to be useful for auditing a knowledge base?

All experimental data (the queries, the distributions, the experimental results, and details of the learning method), as well as the source code, are available here: `http://www.info.univ-tours.fr/~soulet/prototype/iswc18`.

### 5.1 Verification of the transferability assumption

Assumption 2 (Transferability) is a central assumption in the unsupervised approach for learning the GBL parameter. Our first experiment aims to verify if this assumption is true. For this, we compare the parameter $\alpha$ that we obtained by the unsupervised approach to the parameter $\alpha$ of the real world. We found seven relations under the numerical transformation that are known to verify Benford's law in the real world, and that exist in DBpedia and Wikidata. We also found one relation under the counting transformation that exists in our KBs and that is known to follow the GBL in the real world: the out-degree of Wikipedia pages, where $\alpha = -1/2.6 = -0.385$ [5].
Table 2 shows the results obtained for representativeness by Theorem 1 in both supervised and unsupervised contexts. The last column indicates the GBL compliance between the supervised and unsupervised case according to the MAD test (Section 4.3). We see that the learned parameter conforms to the ground truth in all cases: it is very close to zero and does not deviate to values that would have a distorting impact (e.g., $\alpha > 2$, or $\alpha > 5$). For the out-degree of Wikipedia pages, the learned parameter also corresponds well to the real parameter. In addition, the estimator of MAD always indicates a very good conformity ($\leq 0.012$). This entails that the representativeness that we compute in the unsupervised approach is very similar to the supervised value. In all cases except one,

| | | Sup. | | Unsup. | | |
|---|---|---|---|---|---|---|
| Relation | KB | $\alpha^*$ | Rep. | $\alpha$ | Rep. | $MAD(B^{\alpha}, B^{\alpha^*})$ |
| Population of places | DBpedia | 0.001 | 0.949 | -0.020 | 0.954 | C |
| Elevation of places | DBpedia | 0.001 | 0.750 | -0.083 | 0.765 | C |
| Area of places | DBpedia | 0.001 | 0.535 | 0.143 | 0.624 | AC |
| Length of water streams | DBpedia | 0.001 | 0.887 | 0.001 | 0.887 | C |
| Discharge of water streams | DBpedia | 0.001 | 0.938 | -0.105 | 0.930 | AC |
| Number of deaths | Wikidata | 0.001 | 0.909 | -0.106 | 0.908 | AC |
| Number of injured | Wikidata | 0.001 | 0.883 | -0.119 | 0.875 | AC |
| Out-degree of Wikipedia page | DBpedia | -0.385 | 0.999 | -0.486 | 0.999 | AC |

**Table 2.** Conformity of the unsupervised method with the supervised one

there is less than 1% difference. Even for the least correct prediction (`areaTotal`) the difference is at most 10%[3].

Finally, we also applied the unsupervised method to numerical relations whose numbers should *not* verify the GBL. In such a situation, the method should have a MAD test that indicates a nonconformity (i.e. $> 0.015$). This is indeed the case for the following relations: Wikipedia page ID (with MAD 0.029), runtime of films (0.077) or albums (0.090), and weight of persons (0.070).

### 5.2 Validity of representativeness

In Section 3, we postulated that representativeness is an upper bound for completeness. To test this postulation, we simulate an unrepresentative KB as a sample of a known KB. For this purpose, we use the number of inhabitants of French cities from DBpedia as gold standard, because we know that these numbers verify the GBL. We then apply three approaches to degrade this KB:

- **Most-populated:** We removes cities, starting from the least populated to the most populated. This biased sample simulates a KB of Yemeni cities, where only the most populated cities are present.
- **Least-populated:** We remove the most populated cities first. This approach is the opposite of the previous one.
- **Random:** We randomly removes cities. The retained sample of facts is therefore uniformly drawn and it is representative of the original KB.

Our first step is to verify whether our samples conform to Benford's law (Section 4.3). This is indeed the case for 100% of samples for the most-populated approach and the random approach, and for 99% of the samples for the least-populated approach. This validates Assumption 2, and makes our approach applicable. Figure 5 plots the representativeness for the three approaches according to the number of preserved cities in a supervised and unsupervised context. We also plot the real completeness of the sample (w.r.t the original KB).

[3] Different from $\alpha$, the representativeness varies only between 0 and 1.

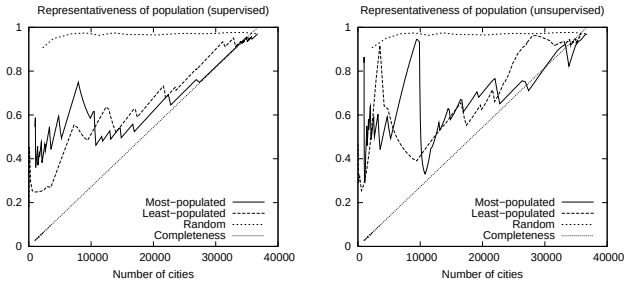



**Fig. 5.** Impact of incompleteness on French cities using `dbo:populationTotal`

We observe that whatever the approach and the context, representativeness is indeed an upper bound for completeness, as postulated. There is only a single major violation at the point of around 34,000 cities for the most-populated approach, which is due to a wrong approximation of the parameter $\alpha$ in that particular sample. Surprisingly, the representativeness is a very good approximation of completeness for the most-populated and the least-populated approaches. In the case of the supervised context, considering a sample $\mathcal{C} = \mathcal{K}_{|\texttt{pop}}$ with more than 22,000 cities, the estimated number of cities (i.e., $P = |\mathcal{C} + \mu^{num}_{\texttt{pop}}\text{-}miss(\mathcal{C})|$) approximates the true number of cities in $\mathcal{K}^*$ (i.e., $T = |\mathcal{K}^*_{|\texttt{pop}}|$) with less than 5% error: $|P - T|/P \leq 0.05$.

Finally, we observe that as long as the number of cities remains large enough (i.e., greater than 2,500), the representativeness of the random approach is high (around 0.95). This is expected for any large random sample from a complete relation, because a random sample has to be representative in our sense.

### 5.3 Effectiveness of the GBL for a KB

We considered in DBpedia (France) all the relations with at least 100 facts. We applied the numerical transformation and the counting transformation. We removed all relations whose numbers are not distributed across several orders of magnitude i.e., $\log_{10} \max(N_r) - \log_{10} \min(N_r) < 1$. Table 3 gives a general overview of the resulting 2,920 relations: the number of considered relations, the number of compliant relations (i.e., with $MAD \leq 0.015$), the number of facts, the proportion of facts in DBpedia, the estimated number of missing facts and finally, the estimated representativeness. Clearly, the counting transformation concerns more relations and facts than the numerical transformation. All in all, our analysis covers about 63% of the facts in DBpedia and we estimate its representativeness at 0.719. To make DBpedia's current relations representative, at least 46 million facts would have to be added.

| Trans. | # of rel. | # of comp. rel. | # of facts | % of DBpedia | Missing facts | Rep. |
|---|---|---|---|---|---|---|
| Counting | 2,920 | 1,461 | 117,349,802 | 0.633 | 45,869,202 | 0.719 |
| Numerical | 108 | 43 | 329,853 | 0.002 | 109,603 | 0.751 |
| Total | 2,920 | 1,487 | 117,461,855 | 0.634 | 45,972,923 | 0.719 |

**Table 3.** Overview of the representativeness of DBpedia (France)

## 6 Conclusion

In this paper, we have introduced the first method to analyze how representative a knowledge base is for the real world. We believe that representativeness is a dimension of data quality in its own right (along with correctness and completeness), because it is essential for applying statistical or machine learning methods. Our approach quantifies a minimum number of facts that must complement the knowledge base in order to make it representative. Experiments on DBpedia validate our proposal in a supervised and unsupervised context on several relations. Using our method, we estimate that at least 46 million facts are missing for DBpedia to be a representative knowledge base. In future work, we would like to take into account representativeness to correct the result of queries on knowledge bases much like this has been done recently for completeness [10].

## References


1. Alam, M., Buzmakov, A., Codocedo, V., Napoli, A.: Mining Definitions from RDF Annotations Using Formal Concept Analysis. In: IJCAI (2015)
2. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: A nucleus for a web of open data. In: The semantic web, pp. 722–735. Springer (2007)
3. Beiglou, P.H.B., Gibbs, C., Rivers, L., Adhikari, U., Mitchell, J.: Applicability of benfords law to compliance assessment of self-reported wastewater treatment plant discharge data. Journal of Environmental Assessment Policy and Management p. 1750017 (2017)
4. Benford, F.: The law of anomalous numbers. Proceedings of the American philosophical society pp. 551–572 (1938)
5. Broder, A., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., Tomkins, A., Wiener, J.: Graph structure in the web. Computer networks 33(1-6), 309–320 (2000)
6. Callahan, E.S., Herring, S.C.: Cultural bias in wikipedia content on famous persons. Journal of the Association for Information Science and Technology 62(10), 1899–1915 (2011)
7. de la Croix, D., Licandro, O.: The longevity of famous people from hammurabi to einstein. Journal of Economic Growth 20(3) (Sep 2015)
8. Darari, F., Razniewski, S., Prasojo, R.E., Nutt, W.: Enabling fine-grained RDF data completeness assessment. In: ICWE. pp. 170–187. Springer (2016)
9. Efron, B.: The jackknife, the bootstrap, and other resampling plans, vol. 38. Siam (1982)

10. Galárraga, L., Hose, K., Razniewski, S.: Enabling completeness-aware querying in SPARQL. In: Proceedings of the 20th International Workshop on the Web and Databases. pp. 19–22. ACM (2017)
11. Galárraga, L., Razniewski, S., Amarilli, A., Suchanek, F.M.: Predicting completeness in knowledge bases. In: WSDM. pp. 375–383. ACM (2017)
12. Galárraga, L., Teflioudi, C., Hose, K., Suchanek, F.M.: Fast rule mining in ontological knowledge bases with AMIE++. The VLDB Journal 24(6), 707–730 (2015)
13. Giles, J.: Internet encyclopaedias go head to head (2005)
14. Hellmann, S., Lehmann, J., Auer, S.: Learning of OWL class descriptions on very large knowledge bases. Int. J. Semantic Web Inf. Syst. 5 (04 2009)
15. Hürlimann, W.: A first digit theorem for powers of perfect powers. Communications in Mathematics and Applications 5(3), 91–99 (2014)
16. Hürlimann, W.: Benfords law in scientific research. Int J Sci Eng Res 6(7), 143–148 (2015)
17. Lajus, J., Suchanek, F.M.: Are All People Married? Determining Obligatory Attributes in Knowledge Bases . In: WWW (2018)
18. Levy, A.Y.: Obtaining complete answers from incomplete databases. In: VLDB (1996)
19. Mebane Jr, W.R.: Election forensics: Vote counts and benfords law. In: Summer Meeting of the Political Methodology Society, UC-Davis, July. pp. 20–22 (2006)
20. Morzy, M., Kajdanowicz, T., Szymański, B.K.: Benfords distribution in complex networks. Scientific reports 6, 34917 (2016)
21. Motro, A.: Integrity = Validity + Completeness. TODS (1989)
22. Nigrini, M.: Benford's Law: Applications for forensic accounting, auditing, and fraud detection, vol. 586. John Wiley & Sons (2012)
23. Nigrini, M.J.: A taxpayer compliance application of benford's law. The Journal of the American Taxation Association 18(1), 72 (1996)
24. Razniewski, S., Korn, F., Nutt, W., Srivastava, D.: Identifying the extent of completeness of query answers over partially complete databases. In: SIGMOD (2015)
25. Razniewski, S., Suchanek, F., Nutt, W.: But what do we actually know? In: Proceedings of the 5th Workshop on Automated Knowledge Base Construction. pp. 40–44 (2016)
26. Rebele, T., Nekoei, A., Suchanek, F.M.: Using YAGO for the Humanities . In: WHISE workshop (2017)
27. Schich, M., Song, C., Ahn, Y.Y., Mirsky, A., Martino, M., Barabási, A.L., Helbing, D.: A network framework of cultural history. Science 345(6196) (2014)
28. Suchanek, F.M., Kasneci, G., Weikum, G.: YAGO: a core of semantic knowledge. In: WWW. pp. 697–706. ACM (2007)
29. Suchanek, F.M., Preda, N.: Semantic culturomics. Proceedings of the VLDB Endowment 7(12), 1215–1218 (2014)
30. Tanon, T.P., Stepanova, D., Razniewski, S., Mirza, P., Weikum, G.: Completeness-aware rule learning from knowledge graphs. In: ISWC. pp. 507–525. Springer (2017)
31. Völker, J., Niepert, M.: Statistical schema induction. In: ESWC (2011)
32. Vrandečić, D., Krötzsch, M.: Wikidata: a free collaborative knowledgebase. Communications of the ACM 57(10), 78–85 (2014)
33. Wagner, C., Garcia, D., Jadidi, M., Strohmaier, M.: It's a man's wikipedia? assessing gender inequality in an online encyclopedia. In: ICWSM. pp. 454–463 (2015)
34. Zaveri, A., Rula, A., Maurino, A., Pietrobon, R., Lehmann, J., Auer, S.: Quality assessment for linked data: A survey. Semantic Web 7(1), 63–93 (2016)

# Mining Significant Maximum Cardinalities in Knowledge Bases

Arnaud Giacometti[0000−0003−0270−5146], Béatrice Markhoff[0000−0002−5171−8499], and Arnaud Soulet[0000−0001−8335−6069]

Université de Tours, LIFAT, France
firstname.lastname@univ-tours.fr

**Abstract.** Semantic Web connects huge knowledge bases whose content has been generated from collaborative platforms and by integration of heterogeneous databases. Naturally, these knowledge bases are incomplete and contain erroneous data. Knowing their data quality is an essential long-term goal to guarantee that querying them returns reliable results. Having cardinality constraints for roles would be an important advance to distinguish correctly and completely described individuals from those having data either incorrect or insufficiently informed. In this paper, we propose a method for automatically discovering from the knowledge base's content the maximum cardinality of roles for each concept, when it exists. This method is robust thanks to the use of Hoeffding's inequality. We also design an algorithm, named C3M, for an exhaustive search of such constraints in a knowledge base benefiting from pruning properties that drastically reduce the search space. Experiments conducted on DBpedia demonstrate the scaling up of C3M, and also highlight the robustness of our method, with a precision higher than 95%.

**Keywords:** Cardinality Mining, Contextual Constraint, Knowledge Base

## 1 Introduction

With the rise of the Semantic Web, knowledge bases (that we will denote KB) are growing and multiplying. At the worldwide level knowledge hubs are built from collaborative platforms, either by extraction from Wikipedia as DBpedia [1] or collaboratively collecting knowledge as for Wikidata [6], or integrating various sources using information retrieval algorithms as for YAGO [21]. These very large KB represent a wealth of information for applications, as this is the case with Wikipedia for human beings. On a smaller scale, more and more knowledge bases are published on the Web, built from diverse data sources following Extract-Transform-Load integration processes that are based on a shared ontology (ontology-based data integration).

Due to the way they are generated, all of these KB need to be enriched with more information to evaluate their quality with respect to the represented reality,

and reverse engineering techniques have already been considered to automatically obtain useful declarations such as keys [16,19]. In this paper we propose to automatically discover another kind of useful declaration about the represented data in a given KB: *role maximum cardinalities*. In knowledge representation, numerical restrictions which specify the number of occurrences of a role are particularly useful [2]. For example, a numerical restriction can be used to describe a concept[1] $C$ as the set of individuals who have at most 3 children. Moreover, a numerical restriction can be used to declare a *maximum cardinality constraint on the role $R$ in the context $C$*, for instance on the role `parent` in the context `Person`, for declaring that individuals of concept `Person` have at most twice the role `parent`. Such a declaration allows reasoners to infer whether all the assertions on role $R$ exist in the KB for any individual belonging to $C$. This can be used to supplement the answers to queries with precise information on their quality in terms of *recall* with respect to reality [20].

`Person` / `birthYear`

| $i$ | $n_i$ | $\tau_i$ | $\widetilde{\tau}_i$ |
|---|---|---|---|
| **1** | **159,841** | **0.999** | **0.996** |
| 2 | 91 | 0.928 | 0.775 |
| 3 | 4 | 0.571 | 0.000 |
| 4 | 2 | 0.667 | 0.000 |
| 5 | 1 | 1.000 | 0.000 |

`Person` / `parent`

| $i$ | $n_i$ | $\tau_i$ | $\widetilde{\tau}_i$ |
|---|---|---|---|
| 1 | 10,643 | 0.529 | 0.518 |
| **2** | **9,392** | **0.991** | **0.975** |
| 3 | 75 | 0.882 | 0.718 |
| 4 | 9 | 0.900 | 0.420 |
| 6 | 1 | 1.000 | 0.000 |

$\top$ / `team`

| $i$ | $n_i$ | $\tau_i$ | $\widetilde{\tau}_i$ |
|---|---|---|---|
| 1 | 1,221,202 | 0.901 | 0.900 |
| 2 | 20,505 | 0.153 | 0.148 |
| 3 | 16,876 | 0.148 | 0.144 |
| ... | ... | ... | ... |
| 20 | 2 | 1.000 | 0.000 |

`FootballMatch` / `team`

| $i$ | $n_i$ | $\tau_i$ | $\widetilde{\tau}_i$ |
|---|---|---|---|
| 1 | 26 | 0.008 | 0.000 |
| **2** | **3,092** | **0.998** | **0.971** |
| 3 | 3 | 0.500 | 0.000 |
| 4 | 2 | 0.667 | 0.000 |
| 5 | 1 | 1.000 | 0.000 |

**Table 1.** Cardinality distributions for some contexts/roles in DBpedia (with the role cardinality $i$, the number of individuals $n_i$ having $i$ times this role, the likelihood $\tau_i$ and the pessimistic likelihood $\widetilde{\tau}_i$ that are defined in Section 4.1)

To the best of our knowledge there is only one work dedicated to the extraction of cardinality constraint from a KB [15], maybe because compared to the traditional database framework, extracting *significant* cardinality constraints from a KB is a far more challenging task. Indeed, we are facing three important challenges. A *first challenge* is that a KB generally contains *inconsistent* data, either because of errors or because of duplicate descriptions. Due to these in-

[1] We use the Description Logics (DL) [2] terminology, as DL are the theoretical foundations of OWL, so we use the terms *concept* (i.e. class), *role* (i.e. property), *individual* and *fact* (i.e. instances).

consistencies, the *observed* maximum cardinality for a role in a KB cannot be considered to be its true maximum cardinality. For example, it is expected that a person will have at most one birth year and two parents. However, considering the roles `birthYear` and `parent` in DBpedia (see Table 1), some persons have 5 birth years or 6 parents. These few inconsistent assertions should not influence the maximum cardinality discovery. Then, a *second challenge* is that a KB is often *incomplete* for a given role. For this reason, the *most frequently observed* cardinality for a role in a KB cannot be considered to be its true maximum cardinality. Typically, most people described in DBpedia have only one informed parent. Nonetheless, we have to take into account that many people have two informed parents for not underestimating the maximum cardinality of the role `parent`. Finally, a *third challenge* is that the expected constraints depend on a *context*. For instance in DBpedia the role `team` is used to inform the teams to which a person has belonged and the teams of a football match. Thus, it is not possible to determine the maximum cardinality of the role `team` in DBpedia (context $\top$), but its maximum cardinality is expected to be 2 in the context of `FootballMatch`. Consequently, instead of exploring each role of a knowledge base, we have to explore each role for each concept. This leads to a huge search space and therefore it is necessary to prune it without missing relevant constraints. But, conversely, we have to avoid extracting redundant constraints. If we identify that a person has at most one birth year, it would be a shame to overwhelm the end user with the cardinality of `birthYear` for artists, scientists and so on.

Taking into account these challenges, we present in this paper two main contributions. Our first contribution is to propose *a method for computing a significant maximum cardinality*. The significance is guaranteed by the use of Hoeffding's inequality for computing corrected likelihood estimates of maximum cardinality. We show with experiments using DBpedia that we extract only reliable maximum cardinalities. More precisely, contrary to [15] it is important to note that we output a maximum cardinality if and only if it is actually significant. Our second contribution is C3M[2], *an algorithm for enumerating the set of all contextual maximum cardinalities* that are minimal (Definition 2) and significant (Definition 4). We use two sound pruning criteria that drastically reduce the exploration space, and ensure the scalability of C3M with large KB. It is also interesting to notice that we implemented C3M in such a way that it explores Web KB via their public SPARQL endpoints without centralizing data.

This paper is structured as follows. Section 2 reviews some related works. In Section 3, we first introduce some basic notions and formalize the problem. Then, in Section 4, we show how to detect a significant maximum cardinality of a role. Next, in Section 5, we present our algorithm C3M. Section 6 provides experimental results on DBpedia that shows its efficiency and its scalability, together with the meaningfulness of discovered constraints. We conclude in Section 7.

[2] The prototype and the results are available at https://github.com/asoulet/c3m, both in CSV and in RDF (Turtle); we provide also the schema of our constraints expressed in RDF.

## 2 Related Work

To increase knowledge about the quality of data contained in KB, some proposals calculate quality indicators like completeness [17] or representativeness [18], while others are interested in the enrichment of individuals or concepts with fine-grained assertions or constraints. Our proposal is in the line of these works, which we detail in what follows.

*Works on Mining Role Cardinality for Individuals* Several works consist in enriching the set of assertions on individuals (ABox), and we can distinguish the *endogenous* approaches [9] relying on the assertions already present in the ABox, from the *exogenous* approaches [13] relying on external sources. [9] shows that it is important to determine when a particular role (such as `parent`) is missing for a particular individual (such as *Obama*). Their proposal of Partial Completeness Assumption states that when at least one assertion about a role $R$ is informed for an individual $s$, then all assertions for this role $R$ are informed for this individual $s$. In [13], the authors benefit from text mining applied on Wikipedia for improving the completeness of individuals described in Wikidata. This exogenous approach relies on syntactical patterns to identify cardinalities on individuals. More generally, in [8], the authors propose various kinds of endogenous and exogenous heuristics for characterizing the completeness of individuals, called Completeness Oracles, as for instance taking into account the popularity of individuals (i.e., a famous individual is more likely to have complete information). Our proposal is endogenous as it processes the facts already contained in the KB that we want to enrich. Nevertheless, it does not characterize the role cardinality for a specific *individual* but for a *concept*. It is therefore more general as the constraints for concept $C$ apply for all the individuals of $C$.

*Works on Mining Role Cardinality for Concepts* Other proposals have focused on the enrichment of the schema part (TBox) with new assertions or axioms allowing to partially or completely specify the cardinality of a role. In particular, several works [16,19] address the automated discovery of contextual keys in RDF datasets as it was done in relational databases. They find axioms stating that individuals of a concept $C$ must have only one tuple of values for a given tuple of roles. The same kind of cardinality information is induced by [12]. Indeed, the authors propose to discover roles that are mandatory for individuals of a concept $C$. For this purpose, they compare the density of the role $R$ for individuals of the concept $C$ with the densities of $R$ for other concepts in the concept hierarchy. Our proposal focuses on mining the maximum cardinality for a role $R$ in a context $C$ (if it exists). But, contrary to the previous work, we can get information about cardinalities greater than 1 (e.g., 2 parents for a child). To the best of our knowledge, [15] is the only work explicitly dedicated to the detection of minimum/maximum cardinalities. This approach proceeds in two stages: removal of outliers and calculation of bounds. Unfortunately, KB are so incomplete that the filtering of outliers is ineffective (e.g., there are more children with only one parent than children with 2 parents). Moreover, their filtering method implicitly assumes that the cardinalities follow a normal distribution, or a distribution

that is moderately asymmetric, which is not always the case (see the examples of Table 1). Consequently, for DBpedia their approach finds that a person has at most 2 years of birth (instead of 1) and 3 parents (instead of 2); and a football match has 3 teams (instead of 2). It is also important to note that the method extracts a cardinality constraint for every concept and role of the KB, whatever the number of observations and the distribution (e.g., a constraint for team is found in the context $\top$). Thus, many of these constraints are not significant. On the contrary, our approach benefits from Hoeffding's inequality for ensuring statisical significance. Finally, contrary to our approach, the authors do not envisage an algorithm to systematically explore the roles and concepts of the KB. An exploration strategy is yet crucial and not trivial in practice due to the huge search space.

*Interest of Role Cardinality* Whatever the approach, all information extracted about role cardinalities is useful for improving many methods, as they reduce the uncertainty imposed by the open-world assumption. [9,20] show the necessity of reducing this uncertainty for data mining applied to KB. In particular, [9,8] propose to benefit from the previously mentioned Partial Completeness Assumption for improving the confidence estimation of association rules. More recently, [20] has further improved the confidence estimation of a rule by exploiting the bounds on the cardinality for an individual. Data mining is not the only field where insights about cardinalities are useful. [3,17,4] and more recently [10] propose to characterize query answers benefiting from the completeness degree of the queried data. Most of these methods can therefore directly benefit from the constraints that we investigate in this paper.

## 3 Preliminaries and Problem Formulation

### 3.1 Basic Notations

For talking about KB components, we use Description Logics (DL) [2] terminology. For instance DBpedia is a KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, where $\mathcal{T}$ denotes its TBox and $\mathcal{A}$ denotes its ABox. One example of assertion in $\mathcal{T}$ is $\texttt{Artist} \sqsubseteq \texttt{Person}$, meaning that the concept `Artist` is subsumed by the concept `Person`, i.e. all artists are persons. $\mathcal{T}$ also includes assertions like $\exists\texttt{birthYear} \sqsubseteq \texttt{Person}$, meaning that the role `birthYear` is defined for persons. Note that the only part of the TBox used by our approach is the named hierarchies of concepts. Besides, $\texttt{Person}(Obama)$ and $\texttt{birthYear}(Obama, 1961)$ are assertions of DBpedia's ABox $\mathcal{A}$. The former indicates that *Obama* is a person, while the latter states that *Obama* was born in 1961. In this paper, we assume that a KB $\mathcal{K}$ contains only one hierarchy of concepts and we use the general top concept $\top$ which subsumes every concept in $\mathcal{K}$. In DL, a maximum cardinality $M$ on the role $R$ may be represented using the numerical restriction constructor $\leq M\ R$. $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ implies[3] the constraint $\top \sqsubseteq (\leq M\ R)$, if for all subjects $s$, the number of objects $o$ such that $R(s, o)$

[3] DL formal semantics are given in terms of interpretations, see [2].

belongs to $\mathcal{K}$ (i.e., $R(s,o) \in \mathcal{A}$ or $R(s,o)$ can be inferred from $\mathcal{T}$ and $\mathcal{A}$) is equal to or fewer than $M$.

We focus on cardinality constraints that are *contextual*, as stated in Definition 1. Intuitively, these constraints are not necessarily satisfied for all subjects of a role $R$, but for all the subjects of $R$ that belong to a concept $C$.

**Definition 1 (Contextual Constraint).** *Given an integer $M \geq 1$, a role $R$ and a concept $C$ of a KB $\mathcal{K}$, a contextual maximum cardinality constraint defined on $R$ for $C$ is an expression of the form: $C \sqsubseteq (\leq M\ R)$.*

The concept $C$ is called the context of the constraint $C \sqsubseteq (\leq M\ R)$. For example, the contextual constraint $\texttt{Person} \sqsubseteq (\leq 1\ \texttt{birthYear})$ means that each person has at most 1 birth year, while $\texttt{FootballMatch} \sqsubseteq (\leq 2\ \texttt{team})$ means that a football match has at most 2 teams. Note that asserting that an artist has at most one year of birth (i.e., $\texttt{Artist} \sqsubseteq (\leq 1\ \texttt{birthYear})$) is true, but less general than $\texttt{Person} \sqsubseteq (\leq 1\ \texttt{birthYear})$ because $\texttt{Artist} \sqsubset \texttt{Person}$. Similarly, asserting that 1,000 is a maximum cardinality for the parent role (i.e., $\texttt{Person} \sqsubseteq (\leq 1,000\ \texttt{parent})$) is true, but less specific than $\texttt{Person} \sqsubseteq (\leq 2\ \texttt{parent})$. We want to discover contextual maximum cardinality constraints that have a context as general as possible and a cardinality as small as possible. For this purpose, we introduce the notion of minimal contextual constraint:

**Definition 2 (Minimal Contextual Constraint).** *The contextual constraint $\gamma_1 : C_1 \sqsubseteq (\leq M_1\ R)$ is more general than the contextual constraint $\gamma_2 : C_2 \sqsubseteq (\leq M_2\ R)$, denoted by $\gamma_2 \sqsubset \gamma_1$, iff $C_2 \sqsubset C_1$[4] and $M_1 \leq M_2$, or $C_2 \equiv C_1$ and $M_1 < M_2$. For a given set of contextual constraints $\Gamma$, constraint $\gamma_1 \in \Gamma$ is minimal in $\Gamma$ if there is no constraint $\gamma_2 \in \Gamma$ more general than $\gamma_1$: $(\nexists \gamma_2 \in \Gamma)(\gamma_1 \sqsubset \gamma_2)$.*

The notion of minimality restricts the mining to a set of constraints that is not redundant, meaning that we do not want to extract a maximum cardinality constraint $\gamma_2$ if it is logically implied by another maximum cardinality constraint $\gamma_1$. More precisely, it is easy to see that if a maximum cardinality constraint $\gamma_1 : C_1 \sqsubseteq (\leq M_1\ R)$ is more general than a maximum cardinality constraint $\gamma_2 : C_2 \sqsubseteq (\leq M_2\ R)$, then for all interpretation $\mathcal{I}$ of a KB $\mathcal{K}$, if $\mathcal{I}$ is a model of $\gamma_1$, then $\mathcal{I}$ is also a model of $\gamma_2$. Indeed, if $\mathcal{I}$ is a model of $\gamma_1$, we have $C_1^{\mathcal{I}} \subseteq \{o : \#\{o' : (o,o') \in R^{\mathcal{I}}\} \leq M_1\}$. Moreover, since $\gamma_1$ is more general than $\gamma_2$, we have $C_2^{\mathcal{I}} \subseteq C_1^{\mathcal{I}}$ and $M_1 \leq M_2$. Thus, we have $C_2^{\mathcal{I}} \subseteq C_1^{\mathcal{I}} \subseteq \{o : \#\{o' : (o,o') \in R^{\mathcal{I}}\} \leq M_1\} \subseteq \{o : \#\{o' : (o,o') \in R^{\mathcal{I}}\} \leq M_2\}$, which shows that $\mathcal{I}$ is a model of $\gamma_2$.

Note that our method relies on a named concept hierarchy for exploring possible contexts and using their subsumption relations. However, it is possible to generate such a hierarchy to explore more complex contexts in a pre-processing step. Such an approach is useful to analyze data by expressing the background knowledge of an expert through an analytical hierarchy.

[4] We denote $C \sqsubset C'$ when $C \sqsubseteq C'$ and $C' \not\sqsubseteq C$.

### 3.2 Problem Statement

Considering the statistics in DBpedia provided by Table 1, we do not want to discover the contextual constraints $\texttt{Person} \sqsubseteq (\leq 6\ \texttt{birthYear})$ or $\texttt{Person} \sqsubseteq (\leq 5\ \texttt{parent})$ even if these constraints are satisfied and minimal in $\mathcal{K}$. We would intend to extract the contextual constraints $\texttt{Person} \sqsubseteq (\leq 1\ \texttt{birthYear})$ or $\texttt{Person} \sqsubseteq (\leq 2\ \texttt{parent})$. Therefore, as defined in [14], we assume an ideal description of the world or ideal KB, denoted $\mathcal{K}^*$, in the sense that $\mathcal{K}^*$ is *correct* (it does not contain any inconsistancies) and *complete*. Note that in general, we have neither $\mathcal{K} \subseteq \mathcal{K}^*$, nor $\mathcal{K}^* \subseteq \mathcal{K}$, because $\mathcal{K}$ is inconsistent or incomplete. In this context, our problem can be formalized as follows:

*Problem 1.* Given a knowledge base $\mathcal{K}$, we aim at discovering the set of all contextual maximum cardinality constraints $C \sqsubseteq (\leq M\ R)$ where $C$ and $R$ are concept and role of $\mathcal{K}$, that are *satisfied* in $\mathcal{K}^*$ and *minimal* with respect to the concept hierarchy of $\mathcal{K}$.

In order to solve Problem 1 we have to deal with the two following challenges: (i) discover constraints that would be satisfied in $\mathcal{K}^*$ whereas this knowledge base is hypothetical and unknown (see Section 4), and (ii) efficiently explore the search space knowing that the number of possible contextual maximum cardinality constraints is huge (see Section 5).

## 4 Detecting Significant Maximum Cardinalities

This section use a probability framework relying on the hypothesis that the degree of completeness of a role is in general higher than its level of inconsistencies. For instance, this assumption is reasonable for DBpedia. Indeed, even if it is difficult to evaluate the completeness and the semantic accuracy of a knowledge base because it requires a gold standard [5], several results of the literature tend to show that the semantic accuracy of DBpedia is better than its completeness [7].

More formally, let us assume that $M$ is the *true* maximum cardinality of the role $R$ in the context $C$, meaning that the maximum cardinality constraint $\gamma : C \sqsubseteq (\leq M\ R)$ is satisfied in $\mathcal{K}^*$. In practice, the ideal KB $\mathcal{K}^*$ is unknown and we only have a sample $\mathcal{K}$ of the reality. Let $X$ be the random variable that denotes for a subject $s$ the number of assertions $R(s, o)$ observed in $\mathcal{K}$. We assume that:

- The level of inconsistencies in $\mathcal{K}$ is not significant, i.e. the probability $\mathbf{P}(X > M)$ to observe a cardinality greater than $M$ for role $R$ is low. For example, in Table 1, we can see that 85 individuals of context `Person` have more than 2 parents, but they represent less than 0.43% of the observed individuals.
- The degree of completeness (present roles) is significantly higher, i.e. the probability $\mathbf{P}(X = M)$ to observe the maximum cardinality $M$ is significantly higher than $\mathbf{P}(X > M)$. For example, in Table 1, we can see that $9,342$ individuals of context `Person` have 2 parents, which represents more than 46.7% of the observed individuals.

Under these hypotheses, the following property states that if $M$ is the true maximum cardinality of the role $R$ in the context $C$, then $M$ is the integer $i$ that maximizes the conditional probability $\mathbf{P}(X = i|X \geq i)$:

*Property 1.* Let $M$ be the *true* maximum cardinality of the role $R$ in the context $C$. If $\mathbf{P}(X = M) \geq \lambda$ and $\mathbf{P}(X > M) \leq \epsilon$, then we have $\mathbf{P}(X = M|X \geq M) \geq \frac{\lambda}{\lambda+\epsilon}$ and $\mathbf{P}(X = i|X \geq i) \leq (1 - \lambda)$ for $i \in [1..M[$. Moreover, if $\lambda > 1/2(\sqrt{\epsilon^2 + 4\epsilon} - \epsilon)$, we have: $M = \arg\max_{i \in \mathbb{N}^+} \{\mathbf{P}(X = i|X \geq i) : \mathbf{P}(X = i) > \epsilon\}$.

Due to lack of space, we omit the proofs. Assuming an inconsistency level $\epsilon$ equal to 0.1% (resp. 1%), Property 1 states that it is possible to detect a true maximum cardinality if the degree of completeness $\lambda$ is greater than $1/2(\sqrt{0.001^2 + 4 \cdot 0.001} - 0.001) = 3.2\%$ (resp. 9.5%). Moreover, a true maximum cardinality constraint $M$ will be detected if $\mathbf{P}(X = M|X \geq M) \geq \frac{\lambda}{\lambda+\epsilon} \geq 97\%$ (resp. 90%). Finally, note that when there is no inconsistency (i.e., $\mathbf{P}(X > M) = 0$ and $\epsilon = 0$), if $M$ is a true maximum cardinality, then $\mathbf{P}(X = M|X \geq M) = 1$.

Now, based on this assumption, we define in Section 4.1 the measure of *likelihood* to detect maximum cardinality constraints, and show how to use Hoeffding's inequality to obtain more accurate decisions. Besides, we introduce in Section 4.2 the notion of *significant constraint*.

### 4.1 Likelihood Measure

We now introduce the notion of likelihood to measure a frequency estimation of the conditional probability $\mathbf{P}(X = i|X \geq i)$ involved in Property 1 (for deciding whether a cardinality $i$ for the role $R$ in the context $C$ is likely to be maximum):

**Definition 3 (Likelihood).** *Given a knowledge base $\mathcal{K}$, the likelihood of the maximum cardinality $i$ of the role $R$ for the context $C$ is the ratio defined as follows: $\tau_i^{C,R}(\mathcal{K}) = \frac{n_i^{C,R}}{n_{\geq i}^{C,R}}$ if $n_{\geq i}^{C,R} > 0$ (0 otherwise) where $n_i^{C,R}$ (resp. $n_{\geq i}^{C,R}$) is the number of individuals $s$ of the context $C$ such that $i$ facts $R(s, o)$ (resp. $i$ facts or more) are stated in $\mathcal{K}$.*

When the context and the role are clear, we omit them in notations. In that case, $n_i$, $n_{\geq i}$ and $\tau_i(\mathcal{K})$ respectively denote $n_i^{C,R}$, $n_{\geq i}^{C,R}$ and $\tau_i^{C,R}(\mathcal{K})$.

For example, let us consider the context `Person` and the role `parent`. Using Table 1, it is easy to see that $n_{\geq 2}^{\texttt{Person,parent}} = 9,477$ $(9,477 = 9,392 + 75 + 9 + 1)$. Thereby, the likelihood $\tau_2^{\texttt{Person,parent}}(\mathcal{K})$ is 0.991 (i.e., $9,392/9,477$). Note that this measure ignores the $10,643$ persons that have only one informed parent (to evaluate if 2 is the true maximum cardinality for parents). Then, it is also easy to see that we have $\tau_6^{\texttt{Person,parent}}(\mathcal{K}) = 1$, whereas 6 is not the true maximum cardinality for the role `parent`. Intuitively, if the likelihood $\tau_6^{\texttt{Person,parent}}(\mathcal{K}) = 1$ does not make sense, it is due to an insufficient number of individuals for reinforcing this hypothesis (here, only 1 individual has 6 parents). In general, the estimation of $\mathbf{P}(X = i|X \geq i)$ by $\tau_i(\mathcal{K})$ must be corrected to be statistically

valid. For this purpose, we benefit from the Hoeffding's inequality [11] which has the advantage of being true for any distribution. It provides an upper bound on the probability that an empirical mean (in our case, a likelihood $\tau_i(\mathcal{K})$) deviates from its expected value (the conditional probability $\mathbf{P}(X = i|X \geq i)$) by more than a given amount. More formally, we have the following property:

*Property 2 (Lower bound).* Given a knowledge base $\mathcal{K}$ and a confidence level $1 - \delta$, assuming that all the observations are independently and identically distributed, the conditional probability $\theta_i = \mathbf{P}(X = i|X \geq i)$ is greater than the pessimistic likelihood $\widetilde{\tau}_i(\mathcal{K})$ defined by (if $n_{\geq i} > 0$):

$$\widetilde{\tau}_i(\mathcal{K}) = \max\left\{\frac{n_i}{n_{\geq i}} - \sqrt{\frac{\log(1/\delta)}{2n_{\geq i}}}, 0\right\}$$

with a probability greater than $(1 - \delta)$, i.e. $\mathbf{P}(\theta_i \geq \widetilde{\tau}_i(\mathcal{K})) \geq (1 - \delta)$.

This property provides us an efficient tool to make confident decisions. For instance, for the role `parent` in Table 1, we observe that the correction strongly reduces the likelihood $\tau_i(\mathcal{K})$ for cardinalities 3, 4 and 6 (e.g., $\widetilde{\tau}_6^{\texttt{Person,parent}}(\mathcal{K}) = 0$). Conversely, we have $\widetilde{\tau}_2^{\texttt{Person,parent}}(\mathcal{K}) = 0.975$, a strong indicator to consider that 2 is the true maximum cardinality for the role `parent` in the context `Person`.

### 4.2 Significant Maximum Cardinality

Using Property 1 and 2, we finally propose to detect a maximum cardinality $M$ for a confidence level $1 - \delta$ if (i) the pessimistic likelihood $\widetilde{\tau}_M(\mathcal{K})$ is maximum, i.e. $\widetilde{\tau}_M(\mathcal{K}) = \max_{i>0} \widetilde{\tau}_M(\mathcal{K})$, and (ii) the pessimistic likelihood $\widetilde{\tau}_M(\mathcal{K})$ is greater than a minimum likelihood threshold $min_\tau$. Based on this heuristic, we introduce the notion of *significant* maximum cardinality constraint:

**Definition 4 (Significant Constraint).** *Given a minimum likelihood threshold $min_\tau$, a confidence level $1-\delta$ and a knowledge base $\mathcal{K}$, a contextual maximum cardinality constraint $C \sqsubseteq (\leq M\ R)$ is significant w.r.t. $\mathcal{K}$ iff $\widetilde{\tau}_M(\mathcal{K}) \geq min_\tau$ and $\widetilde{\tau}_M(\mathcal{K}) = max_{i\geq 1}\widetilde{\tau}_i(\mathcal{K})$.*

Compared to Property 1, note that in our heuristic, we do not test whether $\widetilde{\tau}_M$ is greater than $\epsilon$, or not. However, it is easy to see that if $\widetilde{\tau}_M = \tau_M - \sqrt{\frac{\log(1/\delta)}{2n_{\geq M}}} \geq min_\tau$, then we necessarily have $n_{\geq M} \geq \frac{\log(1/\delta)}{2(1-min_\tau)^2}$, which guarantees that we will not make a decision if the number of observations $n_{\geq M}$ is too low. For example, with $1 - \delta = 99\%$ and $min_\tau = 0.97$, we will consider that $M$ is a true maximum cardinality only if $n_{\geq M} \geq 2,558$.

In DBpedia for a confidence level $1 - \delta = 99\%$ and a threshold $min_\tau = 0.97$, we observe that the detected maximum cardinalities of the roles `birthYear` and `parent` in the context `Person` are 1 and 2 respectively (bold values in Table 1). Interestingly, with these same thresholds, no maximum cardinality is detected

for the role `team` when no context is considered. This is because this role is used both to inform the teams to which a player has belonged and the teams present in a sport event. Thence, our method manages to detect the cardinality of 2 in the context of football matches.

By Definition 4, if a constraint is *significant* w.r.t. $\mathcal{K}$, it means that its pessimistic likelihood is greater than $min_\tau$ and that it is probably satisfied in $\mathcal{K}^*$ (using Property 1 and Property 2). Now, our problem is expressed as follows:

*Problem 2.* Given a knowledge base $\mathcal{K}$ satisfying the assumptions expressed in Section 4 about its consistency and its completeness, a confidence level $1-\delta$ and a minimum likelihood threshold $min_\tau$, we aim at discovering the set of all contextual maximum cardinality constraints $C \sqsubseteq (\leq M\ R)$ where $C$ and $R$ are concept and role of $\mathcal{K}$, that are *significant* w.r.t. $\mathcal{K}$ and *minimal* w.r.t. the concept hierarchy defined in the TBox of $\mathcal{K}$.

## 5 Extracting Maximum Cardinality Constraints

### 5.1 Pruning Criteria

For discovering all the contextual constraints of a knowledge base $\mathcal{K}$, a naive approach would consist in testing each role for each concept with our detection method. If $N_C$ is the number of concepts and $N_R$ the number of roles, this naive approach would require $N_C \times N_R$ tests. This is unfeasible for large knowledge bases such as DBpedia, containing more than 483k concepts and 60k roles. We design two pruning criteria (Properties 3 and 4) taking advantage of the two conditions that a constraint $\gamma$ must satisfy to be mined: (i) the constraint $\gamma$ has to be *significant* i.e., its pessimistic likelihood has to be greater than the minimum likelihood threshold $min_\tau$, and (ii) the constraint $\gamma$ has to be *minimal* with respect to the hierarchy of concepts defined in the TBox of $\mathcal{K}$.

First, we show that a constraint $C \sqsubseteq (\leq M\ R)$ cannot be significant if the number of individuals of the context $C$ in $\mathcal{K}$ is too small. Indeed, if $|C|$ is too small, the confidence interval computed with Hoeffding's inequality is very large and consequently, the pessimistic likelihood is lower than the minimum threshold $min_\tau$. This intuition is formally presented in this property:

*Property 3 (Significance pruning).* Given a confidence level $1-\delta$ and a minimum likelihood threshold $min_\tau$, if one has $|C \sqcap (\exists R.\top)| < \frac{\log(1/\delta)}{2(1-min_\tau)^2}$ for the context $C$ and the role $R$, then no contextual constraint $C' \sqsubseteq (\leq M\ R)$ with $C' \sqsubseteq C$ can be significant w.r.t. the knowledge base $\mathcal{K}$.

This property is very important to reduce the search space because if the number of individuals in $\mathcal{A}$ that belong to $C \sqcap (\exists R.\top)$, for a context $C$ and a role $R$, is not large enough (if it is lower than $\log(1/\delta)/2(1-min_\tau)^2$), then it is impossible to find a significant constraint $C' \sqsubseteq (\leq M\ R)$ where $C'$ is a concept more specific than $C$ in the hierarchy of $\mathcal{K}$. For example, we use a minimum

likelihood threshold $min_\tau$ of 97% and a confidence $1-\delta$ of 99% to extract constraints in DBpedia (see experimental sections), which means that at least 2,558 observations are needed for a role $R$ in a context $C$. For this reason, since there are only 896 facts for the role `beatifiedDate` describing the context `Person`, we are sure that it is not necessary to explore this role for the sub-concepts like `Artist` or `Scientist`.

Assume now that we have extracted the constraint $C \sqsubseteq (\leq 1\ R)$ from the knowledge base $\mathcal{K}$. It is not possible to find another *minimal* constraint $C' \sqsubseteq (\leq M'\ R)$ with a context $C'$ more specific than $C$ because the cardinality $M'$ cannot be smaller than 1. This property, which is a direct consequence of minimality (see Definition 2), is formalized as follows:

*Property 4 (Minimality pruning).* Let $\Gamma$ be a set of contextual maximum cardinality constraints. If $\Gamma$ contains a contextual constraint $C \sqsubseteq (\leq 1\ R)$, then no contextual constraint $C' \sqsubseteq (\leq M'\ R)$ with $C' \sqsubset C$ can be minimal in $\Gamma$.

Property 4 is also useful to reduce the search space because if a constraint $C \sqsubseteq (\leq 1\ R)$ has been detected as significant, then it is useless to explore all the constraints $C' \sqsubseteq (\leq M'\ R)$ where $C' \sqsubset C$. As soon as the constraint $\texttt{Person} \sqsubseteq (\leq 1\ \texttt{birthYear})$ has been detected (meaning than a person has at most one birth year), it is no longer necessary to explore the constraint $\texttt{Artist} \sqsubseteq (\leq M\ \texttt{birthYear})$ which is more specific.

### 5.2 C3M: Contextual Cardinality Constraint Mining

Properties 3 and 4 are implemented in our algorithm called C3M (*C3M* for *Contextual Cardinality Constraint Mining*). Its main function, called *C3M-Main*, takes as input a knowledge base $\mathcal{K}$, a confidence level $1-\delta$ and a minimum likelihood threshold $min_\tau$. The exploration of the search space is performed independently for each role $R$ of the knowledge base $\mathcal{K}$ (see the main loop of Algorithm 1 at line 2). In a first phase, given a role $R$ of $\mathcal{K}$, Algorithm 1 carries out a depth-first exploration of cardinality constraints for $R$ (line 4). This exploration starts from the top concept of $\mathcal{K}$, denoted by $\top$, by calling the recursive function *C3M-Explore*. Because the concepts of $\mathcal{K}$ may have multiple more general concepts, the set $\Gamma_R$ of maximum cardinality constraints returned by function *C3M-Explore* may contain constraints that are not minimal. Therefore, in a second phase (line 6), the function *C3M-Main* checks for each constraint $\gamma \in \Gamma_R$ if $\Gamma_R$ contains a constraint $\gamma'$ that is more general than $\gamma$. When it is not the case constraint $\gamma$ is added to the set of maximum cardinality constraints $\Gamma_m$ that are minimal. $\Gamma_m$ is finally returned by function *C3M-Main* (line 8).

The recursive function *C3M-Explore* benefits from the pruning criteria presented in Properties 3 and 4 during a depth-first exploration of the search space. First, it evaluates if the number of observations in $C \sqcap (\exists R.\top)$ is sufficiently important. If it is not the case, we know that there is no maximum cardinality constraint $C' \sqsubseteq (\leq M\ R)$ with $C' \sqsubseteq C$ that can be significant w.r.t. $\mathcal{K}$ (see Property 3) and the depth-first exploration is stopped (line 2 of Algorithm 2).

**Algorithm 1** C3M-Main

**Input:** A knowledge base $\mathcal{K}$, a confidence level $1-\delta$ and a minimum likelihood threshold $min_\tau$

**Output:** The set $\Gamma_m$ of all maximum cardinality constraints that are significant and minimal w.r.t. $\mathcal{K}$

1: $\Gamma_m := \emptyset$
2: **for all** role in $\mathcal{K}$ **do**
3: {Depth-first exploration of maximum cardinality constraints}
4: $\Gamma_R := C3M\text{-}Explore(\mathcal{K}, R, \top, \infty, \delta, min_\tau)$
5: {Computation of maximum cardinality constraints that are minimal}
6: $\Gamma_m := \{\gamma \in \Gamma_R : (\nexists \gamma' \in \Gamma_R)(\gamma \sqsubset \gamma')\} \cup \Gamma_m$
7: **end for**
8: **return** $\Gamma_m$

Otherwise, the pessimistic likelihood $\widetilde{\tau}_i$ is computed for each cardinality value $i$ (lines 4-6) and the most likely cardinality $i_M$ is selected (line 7). If the corresponding pessimistic likelihood $\widetilde{\tau}_{i_M}$ is lower than $min_\tau$, it means that no maximum cardinality constraint is detected (for this level of the hierarchy of $\mathcal{K}$) and $i_M$ is set to $\infty$ (line 8). Otherwise, if $i_M$ is strictly lower than $M$ (the maximum cardinality detected at a previous level of the hierarchy), it means that we detect a maximum constraint cardinality $\gamma : C \sqsubseteq (\leq i_M\ R)$ that is *potentially* minimal. As already mentioned, as a concept of the knowledge base $\mathcal{K}$ may have multiple super-concepts, we will have to check whether $\gamma$ is really minimal in the second phase of function *C3M-Main*. Finally, using Property 4, we know that if $i_M = 1$, it is not necessary to explore the descendants $C' \sqsubset C$ to detect other constraints $C' \sqsubseteq (\leq M'\ R)$. Otherwise, *C3M-Explore* is recursively called (line 12) to explore all the direct sub-concepts of $C$ (identified using the hierarchy in the TBox of $\mathcal{K}$).

**Theorem 1.** *Given a knowledge base $\mathcal{K}$, a confidence level $1-\delta$ and a minimum likelihood $min_\tau$, our algorithm C3M-Main returns the set of all contextual cardinality constraints $C \sqsubseteq (\leq M\ R)$ that are significant w.r.t. $\mathcal{K}$ and minimal w.r.t. the hierarchy of concepts defined in the TBox of $\mathcal{K}$.*

Theorem 1 straightforwardly stems from Properties 3 and 4. Although these pruning criteria are not heuristic, we will see in the experimental section that algorithm *C3M-Main* is efficient enough to handle knowledge bases as large as DBpedia. Note that we have implemented the functions *C3M-Main* and *C3M-Explore* (client side) such that they consume a SPARQL endpoint (server side) to query the knowledge base $\mathcal{K}$. More precisely, given a context $C$ and a role $R$, a SPARQL query is built and executed to compute the cardinality distribution $n_i^{C,R}$ ($i \in \mathbb{N}$), which is useful for calculating pessimistic likelihoods (see line 5 of Algorithm 2). Therefore, for each role $R$ in $\mathcal{K}$, the server side executes $N_C$ queries where $N_C$ represents the number of concepts in the hierarchy of concepts of $\mathcal{K}$. It means that the complexity of our approach in number of queries is in $\mathcal{O}(N_C)$. On the other hand, on the client side (where the functions *C3M-Main* and *C3M-*

**Algorithm 2** C3M-Explore

**Input:** A knowledge base $\mathcal{K}$, a role $R$, a context $C$, a cardinality $M$, a confidence level $1-\delta$ and a minimum likelihood threshold $min_\tau$
**Output:** A set $\Gamma$ of constraints

1: $\alpha := \frac{log(1/\delta)}{2(1-min_\tau)^2}$ and $n^{C,R}_{\geq 0} := |C \sqcap (\exists R.\top)|$
2: **if** $(n^{C,R}_{\geq 0} < \alpha)$ **then return** $\emptyset$
3: $\Gamma := \emptyset$ and $i_{max} := \arg\max_{i \in \mathbb{N}} \{n^{C,R}_i > 0\}$
4: **for all** $i \in [1..min\{M, i_{max}\}]$ **do**
5: $\quad \widetilde{\tau}_i := \max\left\{ \frac{n^{C,R}_i}{n^{C,R}_{\geq i}} - \sqrt{\frac{\log(1/\delta)}{2n^{C,R}_{\geq i}}}; 0 \right\}$
6: **end for**
7: $i_M := \arg\max_{i \in [1..min\{M, i_{max}\}]} \{\widetilde{\tau}_i\}$
8: **if** $(\widetilde{\tau}_{i_M} < min_\tau)$ **then** $i_M := \infty$
9: **if** $(i_M < M)$ **then** $\Gamma := \{C \sqsubseteq (\leq i_M\ R)\}$
10: **if** $(i_M > 1)$ **then**
11: $\quad$ **for all** direct sub-concept $C' \sqsubset C$ not yet explored **do**
12: $\quad\quad \Gamma := \Gamma \cup$ *C3M-Explore*$(\mathcal{K}, R, C', i_M, \delta, min_\tau)$
13: $\quad$ **end for**
14: **end if**
15: **return** $\Gamma$

*Explore* are executed), given a role $R$ of $\mathcal{K}$, the complexity of our approach (in number of operations) is in $\mathcal{O}(N_C \times i_{max})$ where $i_{max} = \arg\max_{i \in \mathbb{N}} \{n^{\top,R}_i > 0\}$. Intuitively, $i_{max}$ represents the maximum integer for which there is at least one subject $s$ such that $i_{max}$ facts $R(s, o)$ belong to $\mathcal{K}$.

## 6 Experiments

The goal of this experimental study is mainly to evaluate the scaling of the C3M algorithm with a large knowledge base, the interest of minimality and the precision of the mined constraints. In this paper, we present and analyze experimental results using only DBpedia. This KB contains more than 500 million triples with more than 480k distinct concepts and 60k distinct roles. However, the Github repository at https://github.com/asoulet/c3m provides the execution of C3M on 3 other SPARQL endpoints, YAGO, BNF and EUROPEANA.

Our algorithm is implemented in Java with the Apache Jena Library. In our experiments, DBpedia is directly queried via its SPARQL endpoint[5]. Note that we virtually add an element $\top$ that subsumes all concepts without parents including `owl:Thing`, and the confidence level is $1 - \delta = 99\%$ for all experiments[6]. Figure 1 varies the minimum likelihood threshold $min_\tau$ from 0.90 to 0.99 to observe the evolution of the collection of contextual maximum cardinality constraints.

[5] http://jena.apache.org and https://dbpedia.org
[6] The results for $min_\tau = 0.97$ and the ground truth used to evaluate the precision are available at https://github.com/asoulet/c3m.

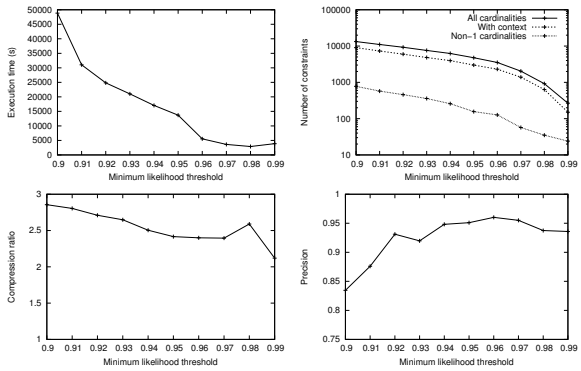



**Fig. 1.** Impact of the minimum likelihood threshold

*Scalability* Figure 1 (left top) reports the execution time, which increases very rapidly when the likelihood threshold decreases. This is due to a very rapid increase of the size of the search space because the pruning properties are less selective. As a result, the number of extracted contextual constraints also increases with the decrease of the threshold $min_\tau$ as shown in Figure 1 (right top). More precisely, it reports the total number of mined constraints, the number of constraints with a non-$\top$ context (i.e., with context different from $\top$), and the number of non-1 constraints (i.e., with maximum cardinality greater than 1). First, it is clear that a majority of constraints have 1 as cardinality. For a minimum likelihood threshold equal to 0.97, there are 1,979 constraints with 1 as maximum cardinality (see Figure 2 (left) that details the distribution of constraints with cardinality). Second, we also observe that most of constraints have a non-$\top$ context that shows the usefulness of our approach based on contexts. For a minimum likelihood threshold equal to 0.97, Figure 2 (right) plots the distribution of the constraints with the level of their context in the DBpedia hierarchy.

*Minimality* Figure 1 (left bottom) plots the compression ratio due to minimality (i.e., number of minimal and non-minimal constraints divided by the number of minimal constraints) by varying the likelihood threshold. Interestingly, the reduction of the number of constraints thanks to minimality is important regardless of the threshold (between 2 and 3 times smaller). It is slightly less effective when the likelihood threshold is high, but much fewer constraints are identified. As a reminder, the non-minimal pruned constraints are not informative because redundant with more general ones. In other words, they are not useful for an

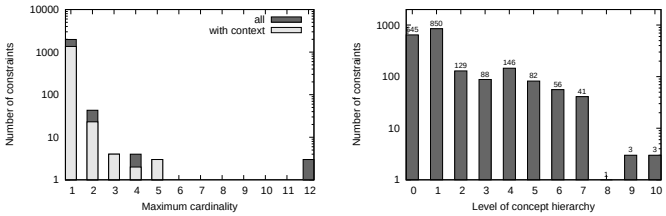



**Fig. 2.** Distribution of constraints for $min_\tau = 0.97$

inference system and in addition, they reduce the readability of the extraction for end users.

*Precision* In order to evaluate the quality of the mined constraints, we built a ground truth from a set $\mathcal{C}^*$ of 5,041 constraints selected from the 13,313 constraints extracted with $min_\tau = 0.90$. We first used common sense knowledge and information from the DBpedia pages to determine the maximum cardinalities of certain relations. For instance, since we have a single birth, the maximum cardinality for all birth dates and places has been set to 1. For some relations like `rdfs:label` or `rdfs:abstract`, the maximum cardinality has been set to 12 according to the documentation[7]. In a second step, we automatically extended the maximum cardinality constraints to the different contexts. The set $\mathcal{C}^*$ covers 667 distinct roles and 2,150 distinct concepts. Thereby, the precision of a set of constraints $\mathcal{C}$ corresponds to the proportion of correct constraints out of the number of constraints that are annotated (i.e., $\mathcal{C} \cap \mathcal{C}^*$). Figure 1 (right bottom) plots the precision of the set of constraints returned by C3M according to the minimum likelihood threshold $min_\tau$[8]. We observe that precision increases with this threshold, but drops off for thresholds greater than 0.96. This is due to correct cardinality constraints which are not recognized as the needed number of individuals is too high. However, it is important to note that this decrease is not very significant because the number of mined constraints becomes very small for thresholds greater than 0.96. Interestingly, for a threshold greater than or equal to 0.94, the precision of our approach is excellent since about 95% of the constraints are correct.

We also qualitatively analyzed the maximum cardinality constraints for a minimum likelihood threshold equal to 0.97. We observe that the erroneous constraints often result from construction or representation biases. For instance, the method found the constraint `http://schema.org/School` $\sqsubseteq (\leq 2$ `country`$)$ that is wrong because a school is located in a single country. But we observe in DBpedia

[7] https://wiki.dbpedia.org/services-resources/datasets/dbpedia-datasets

[8] We do not compare our method with [15] because in the case of DBpedia, this method systematically returns a *wrong maximum* cardinality for all constraints.

that many English schools are attached to both England and the United Kingdom. It is clear that a single affiliation to England (part of the United Kingdom) would have been sufficient. Besides, at physical level, while each individual has a unique date of birth,we identify a cardinality of 2 because many dates are represented with two distinct encoding formats.

To summarize, our approach scales well on DBpedia with about 500 million triples thanks to the advanced pruning techniques used by C3M. The majority of the extracted constraints have a context demonstrating the interest of benefiting from the concept hierarchy of the knowledge base. Importantly, the precision of the mined constraints is about 95% for $min_\tau \geq 0.94$.

## 7 Conclusion

This paper provides the first proposal for a complete exploration of significant constraints of maximum cardinality in a knowledge base. We show how to find, from a knowledge base $\mathcal{K}$ that satisfies assumptions about its completeness and consistence degrees, a minimal set of contextual constraints $C \sqsubseteq (\leq M\ R)$ that are *significant*, i.e. that can be expected to occur in reality. Our experiments demonstrate the feasibility of a systematic exploration of large knowledge bases such as DBpedia (about 500 million triples) for the discovery of minimal contextual constraints of maximum cardinality thanks to the C3M algorithm. With a high minimum likelihood threshold, the precision of the mined constraints is about 95%, which is excellent. Additionally, the minimality exploited by our algorithm drastically reduce the number of obtained constraints, so that they can be manually analyzed by end users. In future work, we would intend to extend our approach to minimum cardinality constraints. This task is not completely symmetrical because under the open-world assumption, it is difficult to know if facts are missing or if the minimum cardinality is reached. For instance, a majority of people have only one informed parent in DBpedia but, of course, the true minimum cardinality is 2. Another future work is to improve C3M by benefiting more from reasoning capabilities. For the moment, we take into account the hierarchy of concepts to reduce the set of constraints, but we could improve our approach by fully exploiting OWL (e.g., with equivalent classes or properties).

## References

1. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: Dbpedia: A nucleus for a web of open data. In: The semantic web, pp. 722–735. Springer (2007)
2. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press, New York, NY, USA (2003)
3. Darari, F., Nutt, W., Pirrò, G., Razniewski, S.: Completeness statements about rdf data sources and their use for query answering. In: Proc. of International Semantic Web Conference. pp. 66–83. Springer Berlin Heidelberg (2013)

4. Darari, F., Razniewski, S., Prasojo, R.E., Nutt, W.: Enabling Fine-Grained RDF Data Completeness Assessment. In: Proc. of International Conference on Web Engineering. pp. 170–187. Springer International Publishing, Cham (2016)
5. Debattista, J., Lange, C., Auer, S., Cortis, D.: Evaluating the quality of the LOD cloud: An empirical investigation. Semantic Web **9**(6), 859–901 (2018)
6. Erxleben, F., Günther, M., Krötzsch, M., Mendez, J., Vrandečić, D.: Introducing wikidata to the linked data web. In: Proc. of International Semantic Web Conference. pp. 50–65. Springer (2014)
7. Färber, M., Bartscherer, F., Menne, C., Rettinger, A.: Linked data quality of DBpedia, Freebase, OpenCyc, Wikidata, and YAGO. Semantic Web **9**(1), 77–129 (2018)
8. Galárraga, L., Razniewski, S., Amarilli, A., Suchanek, F.M.: Predicting completeness in knowledge bases. In: Proc. of the 10th ACM International Conference on Web Search and Data Mining. pp. 375–383. ACM (2017)
9. Galárraga, L.A., Teflioudi, C., Hose, K., Suchanek, F.: AMIE: Association rule mining under incomplete evidence in ontological knowledge bases. In: Proc. of World Wide Web Conference. pp. 413–422. ACM (2013)
10. Galárraga, L., Hose, K., Razniewski, S.: Enabling Completeness-aware Querying in SPARQL. In: Proc. of the 21st Workshop on the Web and Databases. pp. 19–22. ACM (2017)
11. Hoeffding, W.: Probability inequalities for sums of bounded random variables. Journal of the American Statistical Association **58**(310), 13–20 (1963)
12. Lajus, J., Suchanek, F.M.: Are all people married? Determining obligatory attributes in knowledge bases. In: Proc. of World Wide Web conference. pp. 1115–1124 (2018)
13. Mirza, P., Razniewski, S., Darari, F., Weikum, G.: Enriching knowledge bases with counting quantifiers. In: Proc. of International Semantic Web Conference. pp. 179–197. Springer (2018)
14. Motro, A.: Integrity = validity + completeness. ACM Transactional Database Systems **14**(4), 480–502 (Dec 1989)
15. Muñoz, E., Nickles, M.: Mining cardinalities from knowledge bases. In: Proc. of International Conference on Database and Expert Systems Applications. pp. 447–462. Springer (2017)
16. Pernelle, N., Saïs, F., Symeonidou, D.: An automatic key discovery approach for data linking. Web Semantics: Science, Services and Agents on the World Wide Web **23**, 16–30 (2013)
17. Razniewski, S., Korn, F., Nutt, W., Srivastava, D.: Identifying the extent of completeness of query answers over partially complete databases. In: Proc. of the ACM SIGMOD. pp. 561–576. ACM (2015)
18. Soulet, A., Giacometti, A., Markhoff, B., Suchanek, F.M.: Representativeness of knowledge bases with the generalized benford's law. In: Proc. of International Semantic Web Conference. pp. 374–390. Springer (2018)
19. Symeonidou, D., Galárraga, L., Pernelle, N., Saïs, F., Suchanek, F.: Vickey: Mining conditional keys on knowledge bases. In: Proc. of International Semantic Web Conference. pp. 661–677. Springer (2017)
20. Tanon, T.P., Stepanova, D., Razniewski, S., Mirza, P., Weikum, G.: Completeness-aware rule learning from knowledge graphs. In: Proc. of International Semantic Web Conference. pp. 507–525. Springer (2017)
21. Weikum, G., Hoffart, J., Suchanek, F.M.: Ten years of knowledge harvesting: Lessons and challenges. IEEE Data Engineering Bulletin **39**(3), 41–50 (2016)

# Bibliographie


[1] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of databases : the logical level*. Addison-Wesley Longman Publishing Co., Inc., 1995.

[2] R. Agrawal, T. Imieliński, and A. Swami. Mining association rules between sets of items in large databases. In *Acm sigmod record*, volume 22, pages 207–216. ACM, 1993.

[3] R. Agrawal, R. Srikant, et al. Fast algorithms for mining association rules. In *Proc. 20th int. conf. very large data bases, VLDB*, volume 1215, pages 487–499, 1994.

[4] R. Agrawal, R. Srikant, et al. Mining sequential patterns. In *icde*, volume 95, pages 3–14, 1995.

[5] M. Al Hasan and M. J. Zaki. Output space sampling for graph patterns. *Proceedings of the VLDB Endowment*, 2(1) :730–741, 2009.

[6] J. Aligon, D. Li, P. Marcel, and A. Soulet. Towards a logical framework for olap query log manipulation. PersDB, 2012.

[7] E. A. S. Aly, M. L. Diakité, A. Giacometti, B. Markhoff, and A. Soulet. Découverte de cardinalité maximale contextuelle dans les bases de connaissances(mining contextual maximum cardinality in knowledge bases). In *Actes de la Conférence Nationale d'Intelligence Artificielle et Rencontres des Jeunes Chercheurs en Intelligence Artificielle (CNIA+RJCIA 2018), Nancy, France, 4-6 Juillet 2018.*, pages 86–93, 2018.

[8] C. Anderson. The long tail. *Wired magazine*, 12(10) :170–177, 2004.

[9] H. Arimura and T. Uno. Polynomial-delay and polynomial-space algorithms for mining closed sequences, graphs, and pictures in accessible set systems. In *Proceedings of the 2009 SIAM International Conference on Data Mining*, pages 1088–1099. SIAM, 2009.

[10] B. Berendt, A. Hotho, and G. Stumme. Towards semantic web mining. In *International Semantic Web Conference*, pages 264–278. Springer, 2002.

[11] M. Bhuiyan, S. Mukhopadhyay, and M. A. Hasan. Interactive pattern mining on hidden data : a sampling-based solution. In *Proc. of ACM CIKM*, pages 95–104, 2012.

[12] H. Blockeel, T. Calders, E. Fromont, B. Goethals, A. Prado, and C. Robardet. A practical comparative study of data mining query languages. In *Inductive databases and constraint-based data mining*, pages 59–77. Springer, 2010.

[13] M. Boley, C. Lucchese, D. Paurat, and T. Gärtner. Direct local pattern sampling by efficient two-step random procedures. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 582–590. ACM, 2011.

[14] S. Borzsony, D. Kossmann, and K. Stocker. The skyline operator. In *Data Engineering, 2001. Proceedings. 17th International Conference on*, pages 421–430. IEEE, 2001.

[15] J.-F. Boulicaut, A. Bykowski, and C. Rigotti. Approximation of frequency queries by means of free-sets. In *PKDD*, pages 75–85, 2000.

[16] B. Bringmann, S. Nijssen, and A. Zimmermann. Pattern-based classification : a unifying perspective. *arXiv preprint arXiv :1111.6191*, 2011.

[17] T. Calders and B. Goethals. Depth-first non-derivable itemset mining. In *Proceedings of the 2005 SIAM international conference on data mining*, pages 250–261. SIAM, 2005.

[18] T. Calders, L. V. Lakshmanan, R. T. Ng, and J. Paredaens. Expressive power of an algebra for data mining. *ACM Transactions on Database Systems (TODS)*, 31(4) :1169–1214, 2006.

[19] A. Casali, R. Cicchetti, and L. Lakhal. Essential patterns : A perfect cover of frequent patterns. In *International Conference on Data Warehousing and Knowledge Discovery*, pages 428–437. Springer, 2005.

[20] B. Crémilleux and J.-F. Boulicaut. Simplest rules characterizing classes generated by $\delta$-free sets. In *Research and development in intelligent systems XIX*, pages 33–46. Springer, 2003.

[21] B. Crémilleux, A. Giacometti, and A. Soulet. How your supporters and opponents define your interestingness. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 373–389. Springer, 2018.

[22] B. Crémilleux, A. Giacometti, and A. Soulet. How Your Supporters and Opponents Define Your Interestingness. In *ECML PKDD*, 2018.

[23] B. Crémilleux, M. Plantevit, and A. Soulet. Preferencebased pattern mining. In *14th International Conference on Formal Concept Analysis, Rennes, France*, pages 1–171, 2017.

[24] B. Crémilleux and A. Soulet. Discovering knowledge from local patterns with global constraints. In *International Conference on Computational Science and Its Applications*, pages 1242–1257. Springer, 2008.

[25] S. de Amo, M. S. Diallo, C. T. Diop, A. Giacometti, D. H. Li, and A. Soulet. Contextual preference mining for user profile construction. *Inf. Syst.*, 49 :182–199, 2015.

[26] S. de Amo, M. S. Diallo, C. T. Diop, A. Giacometti, H. D. Li, and A. Soulet. Mining contextual preference rules for building user profiles. In A. Cuzzocrea and U. Dayal, editors, *DaWaK*, volume 7448 of *Lecture Notes in Computer Science*, pages 229–242. Springer, 2012.

[27] M. S. Diallo. *Découverte de règles de préférences contextuelles : application à la construction de profils utilisateurs.* PhD thesis, Tours, 2015.

[28] T. G. Dietterich and R. S. Michalski. A comparative review of selected methods for learning from examples. In *Machine Learning*, pages 41–81. Springer, 1983.

[29] L. Diop, C. T. Diop, A. Giacometti, D. Li, and A. Soulet. Sequential pattern sampling with norm constraints. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 89–98. IEEE, 2018.

[30] L. Diop, C. T. Diop, A. Giacometti, D. H. Li, and A. Soulet. Echantillonnage de motifs séquentiels sous contrainte sur la norme. In *EGC*, pages 35–46, 2018.

[31] L. Diop, C. T. Diop, A. Giacometti, D. H. Li, and A. Soulet. Découverte de motifs à la demande dans une base de données distribuée. In *Extraction et Gestion des connaissances, EGC 2019, Metz, France, January 21-25, 2019*, pages 21–32, 2019.

[32] K.-C. Duong, M. Bamha, A. Giacometti, D. Li, A. Soulet, and C. Vrain. MapFIM : Memory aware parallelized frequent itemset mining in very large datasets. In *Database and Expert Systems Applications - 28th International Conference, DEXA 2017, Lyon, France, August 28-31, 2017, Proceedings*, 2017.

[33] K.-C. Duong, M. Bamha, A. Giacometti, D. Li, A. Soulet, and C. Vrain. Mapfim+ : Memory aware parallelized frequent itemset mining in very large datasets. In *Transactions on Large-Scale Data-and Knowledge-Centered Systems XXXIX*, pages 200–225. Springer, 2018.

[34] S. Džeroski, B. Goethals, and P. Panov. *Inductive databases and constraint-based data mining.* Springer Science & Business Media, 2010.

[35] V. Dzyuba, M. v. Leeuwen, S. Nijssen, and L. De Raedt. Interactive learning of pattern rankings. *Int. Journal on Artificial Intelligence Tools*, 23(06) :32 pages, 2014.

[36] E. Egho, C. Raïssi, T. Calders, N. Jay, and A. Napoli. On measuring similarity for sequences of itemsets. *Data Mining and Knowledge Discovery*, 29(3) :732–764, 2015.

[37] A. El Moussawi, A. Cheriat, A. Giacometti, N. Labroche, and A. Soulet. Clustering par apprentissage de distance guidé par des préférences sur les attributs. In *EGC'2016*, volume 30, pages 333–344, 2016.

[38] A. El Moussawi, A. Cheriat, A. Giacometti, N. Labroche, and A. Soulet. Clustering with quantitative user preferences on attributes. In *Internationale Conference on Tools with Artificial Intelligence*, 2016.

[39] B. Ganter, G. Stumme, and R. Wille. *Formal concept analysis : foundations and applications*, volume 3626. springer, 2005.

[40] A. Giacometti, D. H. Li, P. Marcel, and A. Soulet. 20 years of pattern mining : a bibliometric survey. *SIGKDD Explorations*, 15(1) :41–50, 2013.

[41] A. Giacometti, D. H. Li, and A. Soulet. 20 ans de découverte de motifs : une étude bibliographique quantitative. In *Extraction et gestion des connaissances (EGC'2013), Actes, 29 janvier - 01 février 2013, Toulouse, France*, pages 133–144, 2013.

[42] A. Giacometti, D. H. Li, and A. Soulet. Balancing the analysis of frequent patterns. In V. S. Tseng, T. B. Ho, Z. Zhou, A. L. P. Chen, and H. Kao, editors, *Advances in Knowledge Discovery and Data Mining - 18th Pacific-Asia Conference, PAKDD 2014, Tainan, Taiwan, May 13-16, 2014. Proceedings, Part I*, volume 8443 of *Lecture Notes in Computer Science*, pages 53–64. Springer, 2014.

[43] A. Giacometti, P. Marcel, E. Negre, and A. Soulet. Query recommendations for olap discovery driven analysis. In *Proceedings of the ACM twelfth international workshop on Data warehousing and OLAP*, pages 81–88. ACM, 2009.

[44] A. Giacometti, P. Marcel, E. Negre, and A. Soulet. Query recommendations for olap discovery-driven analysis. *IJDWM*, 7(2) :1–25, 2011.

[45] A. Giacometti, P. Marcel, and A. Soulet. Equilibrer l'analyse des motifs fréquents. In A. Khenchaf and P. Poncelet, editors, *EGC*, volume RNTI-E-20 of *Revue des Nouvelles Technologies de l'Information*, pages 47–52. Hermann-Éditions, 2011.

[46] A. Giacometti, P. Marcel, and A. Soulet. A relational view of pattern discovery. In J. X. Yu, M.-H. Kim, and R. Unland, editors, *DASFAA (1)*, volume 6587 of *Lecture Notes in Computer Science*, pages 153–167. Springer, 2011.

[47] A. Giacometti, B. Markhoff, and A. Soulet. Mining significant maximum cardinalities in knowledge bases. In *International Semantic Web Conference, to appear*.

[48] A. Giacometti, E. K. Miyaneh, P. Marcel, and A. Soulet. A generic framework for rule-based classification. *Proceedings of LeGo*, pages 37–54, 2008.

[49] A. Giacometti, E. K. Miyaneh, P. Marcel, and A. Soulet. A framework for pattern-based global models. In *International Conference on Intelligent Data Engineering and Automated Learning*, pages 433–440. Springer, 2009.

[50] A. Giacometti and A. Soulet. Anytime algorithm for frequent pattern outlier detection. *International Journal of Data Science and Analytics*, 2(3-4) :119–130, 2016.

[51] A. Giacometti and A. Soulet. Détection de données aberrantes à partir de motifs fréquents sans énumération exhaustive. In *EGC'2016*, volume 30, pages 51–62, 2016.

[52] A. Giacometti and A. Soulet. Frequent pattern outlier detection without exhaustive mining. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 196–207. Springer, 2016.

[53] A. Giacometti and A. Soulet. Interactive pattern sampling for characterizing unlabeled data. In *Advances in Intelligent Data Analysis XV - 15th International Symposium, IDA 2017, London, England, October, 2017, Proceedings*, 2017.

[54] A. Giacometti and A. Soulet. Dense neighborhood pattern sampling in numerical data. In *Proceedings of the 2018 SIAM International Conference on Data Mining, SDM 2018, May 3-5, 2018, San Diego Marriott Mission Valley, San Diego, CA, USA.*, pages 756–764, 2018.

[55] T. Guns, S. Nijssen, and L. De Raedt. k-pattern set mining under constraints. *IEEE Transactions on Knowledge and Data Engineering*, 25(2) :402–418, 2013.

[56] M. Gyssens and D. Van Gucht. The powerset algebra as a natural tool to handle nested database relations. *Journal of Computer and System Sciences*, 45(1) :76–103, 1992.

[57] W. Hämäläinen and G. I. Webb. A tutorial on statistically sound pattern discovery. *Data Mining and Knowledge Discovery*, 33(2) :325–377, 2019.

[58] Z. He, X. Xu, J. Z. Huang, and S. Deng. Fp-outlier : Frequent pattern based outlier detection. *Comput. Sci. Inf. Syst.*, 2(1) :103–118, 2005.

[59] M. Hewasinghage, S. Isaj, A. Giacometti, and A. Soulet. Caractérisation interactive des transactions préférées d'un utilisateur par l'échantillonnage. In *EGC'2017*, pages 285–296, 2017.

[60] T. Imielinski and H. Mannila. A database perspective on knowledge discovery. *Communications of the ACM*, 39(11) :58–64, 1996.

[61] E. Khanjari Miyaneh. *Un cadre générique pour les modèles globaux fondés sur les motifs locaux*. PhD thesis, Tours, 2009.

[62] M. Khiari, P. Boizumault, and B. Crémilleux. Constraint programming for mining n-ary patterns. In *International Conference on Principles and Practice of Constraint Programming*, pages 552–567. Springer, 2010.

[63] A. Knobbe, B. Crémilleux, J. Fürnkranz, and M. Scholz. From local patterns to global models : The lego approach to data mining. *LeGo*, 8 :1–16, 2008.

[64] A. Lausch, A. Schmidt, and L. Tischendorf. Data mining and linked open data–new perspectives for data analysis in environmental research. *Ecological Modelling*, 295 :5–17, 2015.

[65] B. L. W. H. Y. Ma and B. Liu. Integrating classification and association rule mining. In *Proceedings of the fourth international conference on knowledge discovery and data mining*, pages 24–25, 1998.

[66] H. Mannila and H. Toivonen. Levelwise search and borders of theories in knowledge discovery. *Data mining and knowledge discovery*, 1(3) :241–258, 1997.

[67] T. M. Mitchell. Generalization as search. *Artificial intelligence*, 18(2) :203–226, 1982.

[68] M. Ndiaye. *Exploration de grands ensembles de motifs.* PhD thesis, Tours, 2010.

[69] M. Ndiaye, C. T. Diop, A. Giacometti, P. Marcel, and A. Soulet. Construction et exploration de résumés de grands ensembles de règles d'association. In *Bases de données avancées*, 2009.

[70] M. Ndiaye, C. T. Diop, A. Giacometti, P. Marcel, and A. Soulet. Construction et exploration de résumés de grands ensembles de règles d'association. In *Colloque National sur la Recherche en Informatique et ses Applications (CNRIA 2010)*, 2010.

[71] M. Ndiaye, C. T. Diop, A. Giacometti, P. Marcel, and A. Soulet. Cube based summaries of large association rule sets. In *International Conference on Advanced Data Mining and Applications*, pages 73–85. Springer, 2010.

[72] B. Negrevergne, A. Dries, T. Guns, and S. Nijssen. Dominance programming for itemset mining. In *2013 IEEE 13th International Conference on Data Mining*, pages 557–566. IEEE, 2013.

[73] D. Nouvel. *Reconnaissance des entités nommées par exploration de règles d'annotation-Interpréter les marqueurs d'annotation comme instructions de structuration locale*. PhD thesis, Université François Rabelais-Tours, 2012.

[74] D. Nouvel, J. Antoine, N. Friburger, and A. Soulet. Fouille de règles d'annotation pour la reconnaissance d'entités nommées. *TAL*, 54(1) :13–41, 2013.

[75] D. Nouvel, J.-Y. Antoine, N. Friburger, and A. Soulet. Recognizing named entities using automatically extracted transduction rules. In *Language & Technology Conference (LTC'11)*, 2011.

[76] D. Nouvel, J.-Y. Antoine, N. Friburger, and A. Soulet. Coupling knowledge-based and data-driven systems for named entity recognition. In *Innovative hybrid approaches to the processing of textual data (EACL'12 workshop)*, 2012.

[77] D. Nouvel and A. Soulet. Annotation d'entités nommées par extraction de règles de transduction. In *11ème Conférence Internationale Francophone sur l'Extraction et la Gestion des Connaissances (EGC'10)*, 2011.

[78] D. Nouvel, A. Soulet, J.-Y. Antoine, D. Maurel, and N. Friburger. Reconnaissance d'entités nommées : enrichissement d'un système à base de connaissances à partir de techniques de fouille de textes. In *Traitement Automatique des Langues Naturelles*, 2010.

[79] F. Olken. *Random sampling from databases*. PhD thesis, University of California, Berkeley, 1993.

[80] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Discovering frequent closed itemsets for association rules. In *International Conference on Database Theory*, pages 398–416. Springer, 1999.

[81] L. D. Raedt and A. Zimmermann. Constraint-based pattern set mining. In *proceedings of the 2007 SIAM International conference on Data Mining*, pages 237–248. SIAM, 2007.

[82] M.-C. Rousset, M. Atencia, J. David, F. Jouanot, O. Palombi, and F. Ulliana. Datalog revisited for reasoning in linked data. In *Reasoning Web International Summer School*, pages 121–166. Springer, 2017.

[83] S. Rueping. Ranking interesting subgroups. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 913–920. ACM, 2009.

[84] B. Séroussi, A. Soulet, N. Messai, C. Laouénan, F. Mentré, and J. Bouaud. Patient clinical profiles associated with physician non-compliance despite the use of a guideline-based decision support system : a case study with oncodoc2 using data mining techniques. volume 2012, page 828. American Medical Informatics Association, 2012.

[85] B. Séroussi, A. Soulet, J. Spano, J. Lefranc, I. Cojean-Zelek, B. Blaszka-Jaulerry, L. Zelek, A. Durieux, C. Tournigand, N. Messai, A. Rousseau, and J. Bouaud. Which patients may benefit from the use of a decision support system to improve compliance of physician decisions with clinical practice guidelines : A case study with breast cancer involving data mining. In *MEDINFO 2013 - Proceedings of the 14th World Congress on Medical and Health Informatics, 20-13 August 2013, Copenhagen, Denmark*, pages 534–538, 2013.

[86] A. Soulet. Two decades of pattern mining : Principles and methods. In *Business Intelligence - 6th European Summer School, eBISS 2016, Tours, France, July 3-8, 2016, Tutorial Lectures*, pages 59–78, 2016.

[87] A. Soulet and B. Crémilleux. Adequate condensed representations of patterns. *Data Min. Knowl. Discov.*, 17(1) :94–110, 2008.

[88] A. Soulet and B. Crémilleux. Mining constraint-based patterns using automatic relaxation. *Intell. Data Anal.*, 13(1) :109–133, 2009.

[89] A. Soulet, A. Giacometti, B. Markhoff, and F. M. Suchanek. Representativeness of Knowledge Bases with the Generalized Benford's Law. In *ISWC*, 2018.

[90] A. Soulet, C. Raïssi, M. Plantevit, and B. Crémilleux. Mining dominant patterns in the sky. In D. J. Cook, J. Pei, W. Wang, O. R. Zaïane, and X. Wu, editors, *ICDM*, pages 655–664. IEEE, 2011.

[91] A. Soulet and F. Rioult. Efficiently depth-first minimal pattern mining. In V. S. Tseng, T. B. Ho, Z. Zhou, A. L. P. Chen, and H. Kao, editors, *Advances in Knowledge Discovery and Data Mining - 18th Pacific-Asia Conference, PAKDD 2014, Tainan, Taiwan, May 13-16, 2014. Proceedings, Part I*, volume 8443 of *Lecture Notes in Computer Science*, pages 28–39. Springer, 2014.

[92] A. Soulet and F. Rioult. Extraire les motifs minimaux efficacement et en profondeur. In C. Reynaud, A. Martin, and R. Quiniou, editors, *14èmes Journées Francophones Extraction et Gestion des Connaissances, EGC 2014, Rennes, France, 28-32 Janvier, 2014*, volume E-26 of *Revue des Nouvelles Technologies de l'Information*, pages 383–394. Hermann-Éditions, 2014.

[93] A. Soulet and F. Rioult. Exact and approximate minimal pattern mining. In *Advances in Knowledge Discovery and Management*, pages 61–81. Springer International Publishing, 2017.

[94] A. Soulet and F. M. Suchanek. Anytime large-scale analytics of linked open data. In *International Semantic Web Conference, to appear.*

[95] H. Toivonen et al. Sampling large databases for association rules. In *VLDB*, volume 96, pages 134–145, 1996.

[96] W. Ugarte, P. Boizumault, B. Crémilleux, A. Lepailleur, S. Loudni, M. Plantevit, C. Raïssi, and A. Soulet. Skypattern mining : From pattern condensed representations to dynamic constraint satisfaction problems. *Artificial Intelligence*, 244 :48 – 69, 2017. Combining Constraint Solving with Mining and Learning.

[97] W. Ugarte, P. Boizumault, S. Loudni, and B. Crémilleux. Modeling and mining optimal patterns using dynamic csp. In *2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 33–40. IEEE, 2015.

[98] M. Van Leeuwen. Interactive data exploration using pattern mining. In *Interactive knowledge discovery and data mining in biomedical informatics*, pages 169–182. Springer, 2014.

[99] R. Wille. An approach based restructuring lattice theory : hierarchies of concepts. *Reidel*, 1(982,445), 1982.

[100] S. Zilberstein. Using anytime algorithms in intelligent systems. *AI magazine*, 17(3) :73, 1996.

## Résumé

La découverte de motifs est une technique d'énumération utilisée pour extraire des connaissances à partir de bases de données. Ce mémoire synthétise nos principaux résultats concernant la découverte de motifs centrée sur l'utilisateur. Premièrement, nous introduisons l'algèbre relationnelle orientée motif (PORA) qui est le formalisme utilisé dans l'ensemble du mémoire. Nous ajoutons à l'algèbre relationnelle un opérateur de domaine pour générer des hypothèses sur les données et un opérateur de couverture pour comparer les hypothèses aux données. Au-delà de la déclaration d'un processus de fouille, cette algèbre permet de raisonner sur les requêtes pour déduire des propriétés ou procéder à des optimisations par règles de réécriture. Une seconde partie présente nos travaux où les préférences de l'utilisateur guident la découverte de motifs. En d'autres termes, la découverte de motifs est envisagée comme un problème d'optimisation où seuls les meilleurs motifs au sens d'une relation de préférences sont retenus. Pour cela, l'opérateur de couverture est mis en oeuvre pour jouer le rôle de relation de préférences en comparant les motifs deux à deux en vue de conserver les meilleurs. Finalement, nous nous intéressons à la construction de modèle pour améliorer la complémentarité entre les motifs extraits. La dernière partie détaille nos contributions où la méthode d'analyse des motifs guide leur découverte. Avec cette vision, les motifs sont analysés avec une acuité proportionnelle à leur intérêt. Plutôt que d'extraire tous les motifs, il suffit alors de les échantillonner pour les présenter à l'utilisateur avec une probabilité proportionnelle à leur intérêt. Nous étendons PORA pour reformuler algébriquement le principe de l'échantillonnage de motifs. Nous montrons l'intérêt de l'échantillonnage de motifs pour la construction d'approches anytime et la mise en place de système interactif. Enfin, une conclusion dresse un bilan et discute de plusieurs perspectives de recherche.

**Mots-clés :** Fouille de données, Découverte de motifs

---

## Abstract

Pattern mining is an enumeration technique used to discover knowledge from databases. This Habilitation thesis summarizes our main contributions regarding user-centric pattern mining. First, we introduce the pattern-oriented relational algebra (PORA), which is the formalism used throughout the thesis. We add a domain operator to the relational algebra to generate hypotheses about the data and a cover operator to compare the hypotheses to the data. Beyond the declaration of mining processes, this algebra makes it possible to reason on the queries to deduce properties or to optimize queries with rewriting rules. A second part presents our work where the user's preferences guide the mining of patterns. In other words, pattern mining is seen as an optimization problem where only the best patterns in the sense of a preference relation are preserved. For this purpose, the cover operator is implemented to play the role of preference relation by comparing patterns two-by-two in order to retain the best ones. Finally, we are interested in model construction to improve the complementarity between mined patterns. The last part details our contributions where the method of analysis of the patterns guides their discovery. With this vision, the patterns are analyzed with a sharpness proportional to their interest. Rather than mining all the patterns, it is then sufficient to sample them with a probability proportional to their interest for presenting them to the user. We are extending PORA to reformulate the principle of pattern sampling algebraically. We show the interest of pattern sampling for the construction of anytime and interactive systems. Finally, a conclusion summarizes and discusses several research perspectives.

**Keywords :** Data mining, Pattern mining