Une nouvelle approche pour l'extraction efficace des quadri-concepts fréquents

Mohamed Nader Jelassi[1,2,3]

[1] Université Tunis El Manar. Faculté des Sciences de Tunis, Tunis, Tunisie.
[2] Clermont Université, Université Blaise Pascal, LIMOS, BP 10448, F-63000 Clermont-Ferrand, France.
[3] CNRS, UMR 6158, LIMOS, F-63171 Aubière, France.
nader.jelassi@isima.fr

Résumé

L'extraction des patterns fréquents à partir de relations binaires a été largement étudiée depuis de nombreuses années. L'extension du contexte diadique classique par la considération de nouvelles dimensions est donc un challenge intéressant. Dans ce papier, nous proposons une approche efficace pour la fouille des relations 4-aires afin de découvrir les concepts quadratiques fréquents en considérant une nouvelle dimension comme information supplémentaire dans les *folksonomies*. Nous introduisons un nouvel algorithme capable d'extraire ce genre de concepts à partir de *folksonomies* étendues. Nous introduisons également un nouvel opérateur de fermeture qui partitionne l'espace de recherche en classes d'équivalences dont les plus petits éléments sont les quadri-générateurs minimaux. L'étude expérimentale que nous menons met en évidence les bonnes performances de notre algorithme par rapport au seul concurrent de la littérature.

Mots Clef

Contexte quadratique, générateur minimal, fermeture, classe d'équivalence

Abstract

Mining frequent patterns from binary relations has been widely studied since many years. Then, extending the classical dyadic context by considering new dimension(s) is an interesting challenge. In this paper, we propose an efficient approach for mining 4-ary relations to discover frequent quadratic concepts. So, we consider a *folksonomy* commonly composed of triples <users, tags, resources> and we shall consider a new dimension which represents further information about users. We present a formal definition of the problem and introduce an efficient algorithm for its solution to allow for mining extended *folksonomies*. We also introduce a new closure operator that splits the induced search space into equivalence classes whose smallest elements are minimal generators. Carried out experiments on real-world datasets highlight good performances of our algorithm versus the only concurrent of the literature.

Keywords

Quadratic context, minimal generator, closure, equivalence class

1 Introduction et Motivations

Une *folksonomie* est un néologisme pour désigner un système de classification collaborative par les internautes [7]. L'idée est de permettre à des utilisateurs de partager et de décrire des objets via des mots-clés (tags) librement choisis. L'essor des folksonomies, dû au succès des systèmes de partages (*e.g.,* FLICKR, BIBSONOMY, YOUTUBE, etc.), a suscité l'intérêt des chercheurs pour le domaine de *Folksonomy mining*. Cependant, en raison des très grandes tailles des *folksonomies*, plusieurs travaux se sont focalisées sur l'extraction de représentation concises (sans perte d'information) des patterns intéressants, *i.e.,* concepts triadiques [3] [2]. Ce dernier algorithme (DATA PEELER) a la particularité d'être générique, *i.e.,* capable d'extraire des concepts n-aires à partir de relations n-aires, et donc capable d'extraire des concepts quadratiques à partir de relations 4-aires. Récemment, dans [8], l'algorithme TRICONS a surpassé ses concurrents grâce à un balayage intelligent de l'espace de recherche à travers la localisation des générateurs minimaux triadiques. Dans ce papier, nous posons une nouvelle problématique en introduisant une nouvelle dimension dans les folksonomies afin de traiter avec les concepts quadratiques. Ainsi, nous introduisons un algorithme capable d'extraire ces concepts. Nous introduisons également un nouvel opérateur de fermeture qui partitionne l'espace de recherche en classes d'équivalences dont les plus petits éléments sont les quadri-générateurs minimaux (*QGs*) ; ces *QGs* permettent un balayage intelligent de l'espace de recherche [8]. Ensuite, nous comparons les performances de notre nouvel algorithme, ainsi introduit, à celles de DATA PEELER. DATA PEELER est un algorithme générique pour les relations n-aires qui peut donc être appliqué pour les relations 4-aires. La motivation d'introduire une quatrième dimension aux folksonomies vient du fait que plusieurs applications (*e.g.,* tâches de recommandations, proposition d'amis, détection de tendances, pour ne citer que) nécessitent des informations supplémentaires en plus des trois ensembles constituant une *folksonomie*. Ainsi, en

plus des relations 3-aires (utilisateur, tag, ressource), une quatrième dimension serait d'une grande utilité. Cette quatrième dimension peut recouvrir différents aspects : par exemple le profil (genre, âge, profession, ...), ou le temps si on veut étudier la dynamique temporelle des *folksonomies*. Dans ce papier, nous traitons la quatrième dimension de manière indifférente pour l'aspect méthodologique, mais afin d'extraire des résultats à partir du jeu de données choisi, nous focaliserons sur l'aspect profil. Par suite, l'algorithme QUADRICONS, que nous introduisons pour cette tâche spécifique, cherche à extraire un ensemble de quadruplets fréquents, dont chaque quadruplet (U, T, R, P) consiste en un ensemble U d'utilisateurs, un ensemble T de tags, un ensemble R de ressources et un ensemble P de profils. Ces quadruplets, appelés *quadri-concepts fréquents*, vérifie la propriété suivante : chaque utilisateur de U avec un profil de P a taggué chaque ressource de R avec tous les tags de T, et on ne peut ajouter des éléments à un de ces ensembles sans avoir à en retirer à un des trois autres ensembles. De plus, nous pouvons ajouter des contraintes de support sur chacune des quatre dimensions afin d'extraire les quadri-concepts fréquents. Le reste du papier est organisé comme suit : dans la section suivante, nous proposons une définition formelle du problème d'extraction des quadri-concepts fréquents. Nous introduisons un algorithme dédié pour cette tâche ainsi qu'un nouvel opérateur de fermeture dans la Section 3. Dans la Section 4, nous menons une étude expérimentale afin de mesurer les performances de notre algorithme en termes de temps d'exécution et de mémoire consommée. Enfin, nous concluons notre papier avec des perspectives pour nos travaux futurs dans la Section 5.

2 Le problème d'extraction des quadri-concepts fréquents

Nous commençons par une adaptation de la notion de *folksonomie* [3] au contexte quadratique [5].

Définition 1 *Une **v-folksonomie** est un ensemble de tuples $\mathcal{F}_v = (\mathcal{U}, \mathcal{T}, \mathcal{R}, \mathcal{V}, \mathrm{Y})$ où $\mathcal{U}$, $\mathcal{T}$, $\mathcal{R}$ et $\mathcal{V}$ sont des ensembles finis dont les éléments sont appelés **utilisateurs**, **tags**, **ressources** et **variables**. $\mathrm{Y} \subseteq \mathcal{U} \times \mathcal{T} \times \mathcal{R} \times \mathcal{V}$ représente une relation quadratique où chaque élément $y \subseteq \mathrm{Y}$ peut être représenté par un quadruplet : $y = \{(u, t, r, v) \mid u \in \mathcal{U}, t \in \mathcal{T}, r \in \mathcal{R}, v \in \mathcal{V}\}$ ce qui veut dire que l'utilisateur u a annoté la ressource r via le tag t à travers la variable v. Nous considérons que deux utilisateurs sont **proches** s'ils partagent au moins une même variable en commun. Dans le reste du papier, cette variable v peut être modélisée de manière indifférente pour l'aspect méthodologique, mais afin d'extraire des résultats à partir d'un jeu de données du monde réel, nous focaliserons sur l'aspect profil. Ainsi, dans ce qui suit, nous remplaçons la variable v par p (i.e., profil), et nous notons la **p-folksonomie** $\mathcal{F}_p$. Enfin, nous considérons maintenant que deux utilisateurs sont **proches** s'ils partagent au moins une même information de profil en commun (e.g., même âge, même profession, etc.).*

Exemple 1 *Le Tableau 1 montre un exemple d'une p-folksonomie $\mathcal{F}_p$ avec $\mathcal{U} = \{u_1, \ldots, u_4\}$, $\mathcal{T} = \{t_1, \ldots, t_4\}$, $\mathcal{R} = \{r_1, r_2, r_3\}$ et $\mathcal{P} = \{p_1, p_2\}$. Chaque croix désigne une opération de tagging faite par un utilisateur de $\mathcal{U}$, avec une information de profil de $\mathcal{P}$, utilisant un tag de $\mathcal{T}$ sur une ressource de $\mathcal{R}$. Par exemple, l'utilisateur u_1 qui possède les informations de profils p_1 (étudiant) et p_2 (27 ans) a tagué toutes les ressources avec les tags t_2, t_3 et t_4.*

$\mathcal{F}_p$	$\mathcal{R}$	r_1				r_2				r_3			
$\mathcal{P}$	$\mathcal{U}/\mathcal{T}$	t_1	t_2	t_3	t_4	t_1	t_2	t_3	t_4	t_1	t_2	t_3	t_4
	u_1		×	×	×		×	×	×		×	×	×
p_1	u_2		×	×	×	×	×	×	×	×	×	×	×
	u_3		×	×	×	×	×	×	×	×	×	×	×
	u_4					×			×	×			×
	u_1		×	×	×		×	×	×		×	×	×
p_2	u_2		×	×	×	×			×	×	×	×	×
	u_3		×	×	×	×	×	×	×	×	×		
	u_4					×			×	×			×

TABLE 1 – Un exemple d'une *p-folksonomie*.

Un quadri-set est l'extension d'un tri-set ([8]) à notre cas quadratique. Étant donné que l'ensemble des quadri-sets fréquents est très redondant, nous considérons, dans ce qui suit, une représentation condensée de cet ensemble, *i.e.,* un sous-ensemble contenant la même information : l'ensemble des **quadri-concepts fréquents**. La définition d'un quadri-concept est donnée comme suit :

Définition 2 *(*CONCEPT QUADRATIQUE (FRÉQUENT)*) Un concept quadratique (ou quadri-concept) d'une p-folksonomie $\mathcal{F}_p = (\mathcal{U}, \mathcal{T}, \mathcal{R}, \mathcal{P}, \mathrm{Y})$ est un quadruplet (U, T, R, P) avec $U \subseteq \mathcal{U}$, $T \subseteq \mathcal{T}$, $R \subseteq \mathcal{R}$ et $P \subseteq \mathcal{P}$ avec $U \times T \times R \times P \subseteq \mathrm{Y}$ tel que le quadruplet (U, T, R, P) est maximal,* i.e., *aucun de ces ensembles ne peut être augmenté sans diminuer un des trois autres ensembles.*

Les *p-folksonomies* ont quatre dimensions qui sont complètement symétriques. Ainsi, nous pouvons définir des seuils minimaux de supports sur chacune des quatre dimensions. Ces seuils de support sont antimonotones : Si (A_1, B_1, C_1, E_1) avec A_1 maximal pour $A_1 \times B_1 \times C_1 \times E_1 \subseteq \mathrm{Y}$ n'est pas *u-fréquent* (par rapport à la dimension $\mathcal{U}$), alors tous les (A_2, B_2, C_2, E_2) avec $B_1 \subseteq B_2$, $C_1 \subseteq C_2$ et $E_1 \subseteq E_2$ ne sont également pas *u-fréquent*. Il en est de même pour les trois autres dimensions. Dans [6], les auteurs ont démontré qu'au delà d'un contexte à deux dimensions, la symétrie directe entre monotonie et antimonotonie est rompue. À cet effet, ils ont introduit un lemme résultant de la connexion triadique de Galois [1] induite par un contexte triadique. Dans ce qui suit, nous adaptons ce lemme pour notre cas à quatre dimensions.

Lemme 1 *(Voir aussi [9], Proposition 2) Soient (A_1, B_1, C_1, E_1) et (A_2, B_2, C_2, E_2) deux quadri-sets avec A_i maximal pour $A_i \times B_i \times C_i \times E_i \subseteq \mathrm{Y}$, pour i = 1,2. Si $B_1 \subseteq B_2$, $C_1 \subseteq C_2$ et $E_1 \subseteq E_2$ alors $A_2 \subseteq A_1$. Il en est*

de même pour les trois autres dimensions. Dans le reste du papier, l'inclusion (A_1, B_1, C_1, E_1) $\subseteq$ (A_2, B_2, C_2, E_2) est vérifiée si et seulement si $B_1 \subseteq B_2$, $C_1 \subseteq C_2$, $E_1 \subseteq E_2$ et $A_2 \subseteq A_1$.

Problème 1 ***(Extraction des quadri-concepts fréquents)****[3] Soit $\mathcal{F}_p$ = ($\mathcal{U}$, $\mathcal{T}$, $\mathcal{R}$, $\mathcal{P}$, Y) une p-folksonomie et soient $minsupp_u$, $minsupp_t$, $minsupp_r$ et $minsupp_p$ quatre seuils minimaux de supports définis par l'utilisateur. La tâche d'extraction des quadri-concepts fréquents consiste à déterminer* ***tous*** *les quadri-concepts (U, T, R, P) de $\mathcal{F}_p$ tels que $\mid U \mid \geq minsupp_u$, $\mid T \mid \geq minsupp_t$, $\mid R \mid \geq minsupp_r$ et $\mid P \mid \geq minsupp_p$. L'ensemble des quadri-concepts fréquents de $\mathcal{F}_p$ est égal à $\mathcal{QC}$ = {$qc \mid qc$ = (U, T, R, P) est un quadri-concept fréquent}.*

Dans ce qui suit, nous proposons un opérateur de fermeture dédié au cas quadratique, *i.e.,* une *p-folksonomie* :

Définition 3 (OPÉRATEUR DE FERMETURE D'UNE *p-folksonomie*) *Soit S = (A, B, C, E) un quadri-set de $\mathcal{F}_p$ avec A maximal pour $A \times B \times C \times E \subseteq$ Y. L'opérateur de fermeture h de $\mathcal{F}_p$ est défini comme suit :*
$h(S) = h(A, B, C, E) = (U, T, R, P) \mid U = A$
$\wedge\ T = \{t_i \in \mathcal{T} \mid (u_i, t_i, r_i, p_i) \in$ Y $\forall\ u_i \in U, \forall\ r_i \in C, \forall\ p_i \in E\}$
$\wedge\ R = \{r_i \in \mathcal{R} \mid (u_i, t_i, r_i, p_i) \in$ Y $\forall\ u_i \in U, \forall\ t_i \in T, \forall\ p_i \in E\}$
$\wedge\ P = \{p_i \in \mathcal{P} \mid (u_i, t_i, r_i, p_i) \in$ Y $\forall\ u_i \in U, \forall\ t_i \in T, \forall\ r_i \in R\}$

Remarque 1**.** En raison d'un manque d'espace, nous n'avons pu mettre la preuve que h est un opérateur de fermeture (preuves d'extensivité, d'idempotence et d'isotonie). Cependant, cette preuve peut-être consultée dans [4]. L'application de l'opératue de fermeture h sur un quadri-set résulte en un *quadri-concept* qc = (U, T, R, P). Dans la suite du papier, les parties U, R, T et P sont respectivement appelés ***Extent***, ***Intent***, ***Modus*** et ***Variable***. Conformément aux cas diadiques et triadiques, l'opérateur de fermeture partitionne l'espace de recherche en classes d'équivalences, que nous introduisons maintenant :

Définition 4 (CLASSE D'ÉQUIVALENCE) *Soient S_1 = (A_1, B_1, C_1, E_1), S_2 = (A_2, B_2, C_2, E_2) deux quadri-sets de $\mathcal{F}_p$ et qc un quadri-concept fréquent. S_1 et S_2 appartiennent à la même classe d'équivalence représentée par le quadri-concept qc, i.e., $S_1 \equiv_{qc} S_2$ ssi $h(S_1) = h(S_2) = qc$.*

Les générateurs minimaux (*GMs*) jouent un rôle important dans plusieurs problèmes théoriques et pratiques impliquant des systèmes de fermeture. Ils offrent un moyen plus simple de définir un concept étant donné qu'ils contiennent beaucoup moins d'attributs qu'un concept fermé. En effet, les *GMs* représentent les plus petits éléments dans une classe d'équivalence et leur détection devient beaucoup plus facile. En effet, contrairement aux quadri-concepts qui sont des ensembles maximuax d'utilisateurs, tags, ressources et profils, les *GMs* ne contiennent qu'un seul ensemble maximal (*i.e.,* les utilisateurs) commun à un seul tag, une seule ressource et une seule information de profil. Dans ce qui suit, nous introduisons l'extension de la définition d'un *GM* à une *p-folksonomie*.

Algorithme 1: QUADRICONS

Data :
1. $\mathcal{F}_p$ ($\mathcal{U}$, $\mathcal{T}$, $\mathcal{R}$, $\mathcal{P}$, Y) : Une *p-folksonomie*.
2. $minsupp_u$, $minsupp_t$, $minsupp_r$, $minsupp_p$: Seuils minimaux de support.

Result : $\mathcal{QC}$: {Quadri-concepts fréquents}.

begin
  */*Étape 1 : L'extraction des quadri-générateurs*/*
  FINDMINIMALGENERATORS($\mathcal{F}_p$, $\mathcal{MG}$, $minsupp_u$) ;
  */*Étape 2 : le calcul de la partie modus*/*
  foreach *quadri-gen* $g \in \mathcal{MG}$ **do**
    CLOSURECOMPUTE($\mathcal{MG}$, $minsupp_u$, $minsupp_t$, $minsupp_r$, g, $\mathcal{QS}$, 1) ;
  PRUNEINFREQUENTSETS($\mathcal{QS}$,$minsupp_t$) ;
  */*Étape 3 : Le calcul de la partie intent*/*
  foreach *quadri-set* $s \in \mathcal{QS}$ **do**
    CLOSURECOMPUTE($\mathcal{QS}$, $minsupp_u$, $minsupp_t$, $minsupp_r$, s, $\mathcal{QS}$, 2) ;
  PRUNEINFREQUENTSETS($\mathcal{QS}$,$minsupp_r$) ;
  */*Étape 4 : Le calcul de la partie variable*/*
  foreach *quadri-set* $s \in \mathcal{QS}$ **do**
    CLOSURECOMPUTE($\mathcal{QS}$, $minsupp_u$, $minsupp_t$, $minsupp_r$, s, $\mathcal{QC}$, 3) ;
  PRUNEINFREQUENTSETS($\mathcal{QC}$,$minsupp_p$) ;
return $\mathcal{QC}$;

Définition 5 *(*QUADRI-GÉNÉRATEUR MINIMAL*) Soient g =(A, B, C, E) un quadri-set de $\mathcal{F}_p$ tel que $A \subseteq \mathcal{U}$, $B \subseteq \mathcal{T}$, $C \subseteq \mathcal{R}$ et $E \subseteq \mathcal{P}$ et qc un quadri-concept fréquent. Le quadruplet g est un quadri-générateur minimal (ou quadri-générateur) de qc ssi $h(g) = qc$ et $\nexists\ g_1 = (A_1, B_1, C_1, E_1)$ tel que (i) $A = A_1$, (ii) $(B_1 \subseteq B \wedge C_1 \subseteq C \wedge E_1 \subset E) \vee (B_1 \subset B \wedge C_1 \subset C \wedge E_1 \subseteq E)$, et (iii) $h(g) = h(g_1) = qc$.*

Dans ce qui suit, nous introduisons l'algorithme QUADRICONS pour l'extraction des quadri-Concepts fréquents.

3 L'algorithme QUADRICONS

Se basant sur les notions introduites précédemment, nous proposons à présent notre nouvel algorithme QUADRICONS algorithm dédié à la tâche d'extraction des quadri-Concepts fréquents à partir d'une *p-folksonomie*. Puis, nous donnons un exemple illustratif de notre algorithme.

Le pseudo code. Dans ce qui suit, nous introduisons notre algorithme générer-et-tester, appelé QUADRICONS, pour

l'extraction des quadri-Concepts fréquents à partir d'une *p-folksonomie*. QUADRICONS opère en quatre étapes comme suit : tout d'abord, il invoque la procédure FINDMINIMALGENERATORS pour l'extraction des quadri-générateurs. Puis, la procédure CLOSURECOMPUTE est invoquée pour les trois prochaines étapes afin de calculer respectivement le *modus*, l'*intent* et la *variable* des quadri-concepts. Le pseudo code de l'algorithme QUADRICONS est donné par l'Algorithme 1. QUADRICONS prend en entrée une *p-folksonomie* $\mathcal{F}_p$ = ($\mathcal{U}$, $\mathcal{T}$, $\mathcal{R}$, $\mathcal{P}$, Y) ainsi que quatre seuils minimaux de support (un pour chaque dimension) : $minsupp_u$, $minsupp_t$, $minsupp_r$ et $minsupp_p$. La sortie de QUADRICONS est l'ensemble de tous les quadri-concepts fréquents vérifiant ces seuils de supports. QUADRICONS opère comme suit : il commence par invoquer la procédure FINDMINIMALGENERATORS (Étape 1), dont le pseudo code est donné par l'Algorithme 2, afin d'extraire et de stocker les quadri-générateurs dans l'ensemble $\mathcal{MG}$ (Ligne 3). Pour une telle extraction, FINDMINIMALGENERATORS calcule pour chaque quadruplet (u, t, r, p) l'ensemble U_s représentant l'ensemble maximal d'utilisateurs ayant une même information de profil et partageant le tag t sur la ressource r (Algorithme 2, Ligne 3). Si $|U_s|$ est fréquent par rapport à $minsupp_u$ (Ligne 4), un quadri-générateur est alors crée (s'il n'existe pas encore) avec ses champs appropriés (Ligne 5). L'Algorithme 2 invoque la fonction **ADDQUADRI** dont le rôle est d'ajouter le quadri-générateur g à l'ensemble $\mathcal{MG}$ (Ligne 7).

Ensuite, QUADRICONS invoque la procédure CLOSURECOMPUTE (Étape 2) pour chaque quadri-générateur de $\mathcal{MG}$ (Lignes 5-7), dont le pseudo code est donné par l'Algorithme 3 : le but étant de calculer le modus de chaque quadri-concept. À ce stade, les deux premiers cas de l'Algorithme 3 (Lignes 3 et 6) doivent être considérés selon l'*extent* de chaque quadri-générateur. La procédure CLOSURECOMPUTE retourne l'ensemble $\mathcal{QS}$ formé par des quadri-sets. L'indicateur *flag* (ici égal à 1) marqué par QUADRICONS indique si le quadri-set, considéré par la procédure CLOSURECOMPUTE, est un quadri-générateur. Lors de la troisième étape, QUADRICONS invoque une seconde fois la procédure CLOSURECOMPUTE pour chaque quadri-set de $\mathcal{QS}$ (Lignes 9-11), afin de calculer la partie *intent*. CLOSURECOMPUTE se concentre sur les quadri-sets de $\mathcal{QS}$ ayant des *intent* différents (Algorithme 3, Ligne 10). La quatrième et dernière étape de QUADRICONS invoque une dernière fois la procédure CLOSURECOMPUTE avec un indicateur égal à 3. Cela permet de localiser les quadri-sets ayant des parties *variable* différentes (Algorithme 3, Ligne 18) avant la génération des quadri-concepts. QUADRICONS arrive à terme après cette étape et retourne l'ensemble des quadri-concepts fréquents vérifiant les quatre seuils minimaux de support $minsupp_u$, $minsupp_t$, $minsupp_r$ et $minsupp_p$. QUADRICONS invoque la fonction **PRUNEINFREQUENTSETS** (Lignes 8, 13 et 18) afin d'élaguer les quadri-sets/concepts infréquents, *i.e.*, ceux dont la cardinalité du *modus/intent/variable* ne vérifie pas les seuils demandés.

Proposition 1. L'algorithme QUADRICONS est correct et complet. Il extrait exactement **tous** les quadri-concepts fréquents.

Proposition 2. L'algorithme QUADRICONS termine.

Remarque 2. En raison de contraintes d'espace, nous n'avons pas été en mesure de mettre la preuve de terminaison et de correction de QUADRICONS. Toutefois, ces preuves sont disponibles dans [4].

Algorithme 2: FINDMINIMALGENERATORS

Data :

1. $\mathcal{MG}$: L'ensemble des quadri-générateurs fréquents.
2. $\mathcal{F}_p$ ($\mathcal{U}$, $\mathcal{T}$, $\mathcal{R}$, $\mathcal{P}$, Y) : Une *p-folksonomie*.
3. $minsupp_u$: Seuil minimal de support.

Result : $\mathcal{MG}$: {quadri-générateurs fréquents}.

```
1 begin
2     g :un quadri-générateur ;
3     foreach quadruplet (u, t, r, p) de F_p do
4         U_s= {u_i ∈ U | (u_i, t, r, p) ∈ Y} ;
5         if | U_s | ≥ minsupp_u then
6             g.extent = U_s ; g.intent = r ; g.modus = t ;
              g.variable = p
7             if g ∉ MG then
8                 AddQuadri(MG, g)
9 return MG ;
```

Complexité théorique : Comme pour le cas triadique [3], le nombre des quadri-concepts fréquents peut augmenter exponentiellement dans le pire des cas. Ainsi, la complexité théorique de notre algorithme est de l'ordre de $\mathcal{O}(2^n)$ avec $n = |\mathcal{T}| + |\mathcal{R}| + |\mathcal{P}|$. Néanmoins, et comme il sera démontré dans la section suivante, d'un point de vue pratique, les performances réelles sont loin d'être exponentielles. De ce fait, nous concentrons notre évaluation sur un jeu de données à large échelle.

4 Évaluation et Discussion

Dans ce qui suit, nous démontrons à travers nos expérimentations les performances de QUADRICONS *vs.* l'algorithme DATA PEELER en termes de temps d'exécution et mémoire consommée. Nous avons implémenté notre algorithme en langage *C++* (compilé avec *GCC* 4.1.2) tandis que l'exécutable de DATA PEELER a été téléchargé à partir de ce lien : *http ://homepages.dcc.ufmg.br/ lcerf/fr/prototypes.html#dpeeler*. Nous avons utilisé un processeur Intel Core $i5$ muni d'une mémoire de 8 GB. Les tests ont été menés sur le système d'exploitation Linux (Distribution UBUNTU 12.04 64 bits). Nous avons également mis l'accent sur les différences de mémoire consommée entre les deux algorithmes. Nous avons mené nos différentes expérimentations sur le jeu de données du monde réel MOVIELENS. MOVIELENS

Algorithme 3: CLOSURECOMPUTE

Data :

1. $\mathcal{S}_{IN}$: L'ensemble d'entrée.
2. min_u, min_t, min_r : Seuils minimaux de supports.
3. q : Un quadri-générateur/quadri-set.
4. $\mathcal{S}_{OUT}$: L'ensemble de sortie.
5. i : Un indicateur.

Result : $\mathcal{S}_{OUT}$: L'ensemble de sortie.

begin
  foreach *quadri-set* $q' \in \mathcal{S}_{IN}$ **do**
    if i=1 ***et*** $q.intent = q'.intent$ ***et*** $q.extent \subseteq q'.extent$ **then**
      $s.intent = q.intent$; $s.extent = q.extent$; $s.variable = q.variable$; $s.modus = q.modus \cup q'.modus$; ADDQUADRI($\mathcal{S}_{OUT}$, s) ;
    else if i=1 ***et*** $q.intent = q'.intent$ ***et*** q *et* q' incomparables **then**
      $g.extent = q.extent \cap q'.extent$; $g.modus = q.modus \cup q'.modus$; $g.intent = q.intent$; $g.variable = q.variable$;
      If g *u-frequent* **then** ADDQUADRI($\mathcal{MG}$, g) ;
    else if i=2 ***et*** $q.extent \subseteq q'.extent$ ***et*** $q.modus \subseteq q'.modus$ ***et*** $q.intent \neq q'.intent$ **then**
      $qs.extent = q.extent$; $qs.modus = q.modus$; $qs.variable = q.variable$; $qs.intent = q.intent \cup q'.intent$;
      ADDQUADRI($\mathcal{S}_{OUT}$, qs) ;
    else if i=2 ***et*** q *et* q' incomparables **then**
      $s.extent = q.extent \cap q'.extent$; $s.modus = q.modus \cap q'.modus$; $s.variable = q.variable$; $s.intent = q.intent \cup q'.intent$;
      If s is *u-frequent* et *t-frequent* **then** ADDQUADRI($\mathcal{S}_{OUT}$, s) ;
    else if i=3 ***et*** $q.extent \subseteq q'.extent$ ***et*** $q.modus \subseteq q'.modus$ ***et*** $q.intent \subseteq q'.intent$ ***et*** $q.variable \neq q'.variable$ **then**
      $qc.extent = q.extent$; $qc.modus = q.modus$; $qc.intent = q.intent$; $qc.variable = q.variable \cup q'.variable$;
      ADDQUADRI($\mathcal{S}_{OUT}$, qc) ;
    else if i=3 ***et*** q *et* q' incomparables **then**
      $s.extent = q.extent \cap q'.extent$; $s.modus = q.modus \cap q'.modus$; $s.intent = q.intent \cap q'.intent$; $s.variable = q.variable \cup q'.variable$;
      If s est *u-frequent*, *t-frequent* et *r-frequent* **then** ADDQUADRI($\mathcal{S}_{OUT}$, s) ;
return $\mathcal{S}_{OUT}$;

est un système de recommandation qui permet aux utilisateurs d'annoter des films. Le jeu de données utilisé (http ://www.grouplens.org/node/73) contient 95580 tags affectés à 10681 films par 71567 utilisateurs. Des informations supplémentaires sur les utilisateurs sont disponibles dans le jeu de données et forment son profil (la quatrième dimension d'une *p-folksonomie*) et qui renseigne sur le **genre** de l'utilisateur (masculin ou féminin), sa **profession** (au nombre de 21, qui peut être éducateur, écrivain, étudiant, scientifique, etc.) ainsi que sur son **âge** (5 tranches d'âge) : (*i*) $7-18$ ans ; (*ii*) $19-24$ ans ; (*iii*) $25-35$ ans ; (*iv*) $36-45$ ans et (*v*) $46-73$ ans.

Exemples de quadri-concepts extraits. Dans ce qui suit, nous présentons quelques résultats intéressants de quadri-concepts extraits par QUADRICONS à partir du jeu de données MOVIELENS. Nous avons défini les valeurs de seuils de supports suivants : $minsupp_u = 2$, $minsupp_t = 2$, $minsupp_r = 2$ et $minsupp_p = 2$, *i.e.,* dans un quadri-concept fréquent, 2 utilisateurs (au moins) avec deux informations de profils en commun (*e.g.,* même profession et même age) ont assigné les mêmes tags (2 au moins) aux mêmes ressources (2 au moins). De toute évidence, il est plus intéressant de fixer chaque seuil de support à 2 dans le but d'avoir des quadri-concepts avec une valeur ajoutée illustrant les tags et ressources partagés en commun par un groupe de deux utilisateurs (au moins) ayant deux informations de profil en commun. Ainsi, le Tableau 2 illustre quelques exemples (des plus intéressants) de quadri-concepts parmi les 10627 quadri-concepts fréquents vérifiant les seuils de supports décrits ci-dessus. Par exemple, le premier quadri-concept montre que les utilisateurs *saloua*, *wafa* et *yasmine*, trois femmes retraitées, ont partagé les films *Star Wars*, *M.A.S.H* et *Rear Window* via les tags *classic*, *dialog* et *oscar*. Dans ce qui suit, afin d'évaluer les performances de QUADRICONS *vs.* celles de l'instantiation de DATA PEELER au cas quadratique, nous avons tourné les deux algorithmes sur le même dataset et nous avons fait varier les valeurs de seuils de supports.

Utilisateurs	Tags	Ressources	Profil
{saloua, wafa, yasmine}	{classic, dialog, oscar}	{Star Wars, M.A.S.H, Rear Window}	{Femme, 46-73 ans, retraité}
{mulder, scully, krycek}	{bestmovie, cult	{Usual Suspects, Silence of Lambs, Sound of Music}	25-35 ans, Homme, domaine santé}
{rossy, anlucia, franela}	{classic, oldmovie, quotes}	{Rear Window, Magician OZ, Gone with Wind}	36-45 ans, Homme, écrivain}

TABLE 2 – Exemples de quadri-concepts extraits à partir du jeu de données MOVIELENS.

Temps d'exécution de QUADRICONS vs. celui de DATA PEELER. Le Tableau 3 démontre les différents temps d'exécution de QUADRICONS *vs.* ceux de DATA PEELER pour différents nombres de quadruplets, qui croissent de 20000 à 95580 sur le dataset MOVIELENS, et ce pour dif-

férentes valeurs de seuils minimaux de supports. Les deux algorithmes permettent l'extraction de tous les quadri-concepts fréquents, qui sont au nombre d'un peu plus de $10,000$. Nous observons que pour toutes les valeurs du nombre de quadruplets, DATA PEELER est très loin de QUADRICONS en termes de temps d'exécution. QUADRICONS tourne jusqu'à 124 fois plus rapidement que son concurrent. Nous expliquons cette différence par le fait que le principal point fort de DATA PEELER, *i.e.,* sa généricité pour un contexte n-aire, constitue aussi sa faiblesse. En effet, pour n=4, *i.e.,* une instance particulière du problème général traité par DATA PEELER, QUADRICONS, spécialement dédié à la tâche d'extraction des quadri-concepts, est plus apte à mieux les extraire avec un laps de temps largement inférieur. Ainsi, afin d'améliorer le travail existant, notre stratégie de localiser les *QGs* a l'avantage d'extraire les quadri-concepts plus rapidement que son concurrent. Dans des applications pratiques (*e.g.,* recommandations), un utilisateur va préférer un algorithme qui offre un résultat (*e.g.,* des recommandations) en quelques millisecondes.

Mémoire consommée par QUADRICONS vs. celle par DATA PEELER. Le Tableau 3 démontre la mémoire consommée par les deux algorithmes sur le jeu de données MOVIELENS pour différentes quadruplets. QUADRICONS consomme moins de mémoire que son concurrent : jusqu'à 40000 KB contre des millions de KB pour DATA PEELER. Une telle différence s'explique par le fait que QUADRICONS, contrairement à DATA PEELER, ne stocke pas le jeu de données en mémoire avant l'extraction des quadri-concepts. De plus, QUADRICONS génère moins de candidats grâce à l'habile détection des quadri-générateurs qui réduisent considérablement l'espace de recherche. Cette stratégie est adoptée par QUADRICONS afin d'améliorer les performances d'extraction des quadri-concepts tandis que DATA PEELER paie le prix de sa généricité, ce qui était déjà le cas pour le cas triadique [8]. En conséquence, le fait de détecter les quadri-générateurs avant l'extraction des quadri-concepts permet à QUADRICONS de consommer jusqu'à 54 fois moins de mémoire que DATA PEELER sur le jeu de données MOVIELENS. La limite de notre algorithme est que lorsque les seuils de supports ont des valeurs trop basses, les candidats deviennent très nombreux et il devient, en conséquence, très difficile de les parcourir.

| Y |	**QUADRI CONS** (sec)	Mémoire Consommée (kilobits)	**DATA PEELER** (sec)	Mémoire Consommée (kilobits)
$minsupp_u = 2, minsupp_t = 2,$ $minsupp_r = 2, minsupp_d = 1$				
25000	**0. 36**	**198**	39.98	399672
50000	**0. 97**	**431**	107.71	508943
70000	**1 .96**	**567**	227.65	667006
95580	**3. 79**	**1182**	472.87	842551
$minsupp_u = 2, minsupp_t = 2,$ $minsupp_r = 1, minsupp_d = 1$				
25000	**5.76**	**2491**	421.44	769822
50000	**15.92**	**5246**	1269.70	976200
70000	**29.22**	**9845**	2037.73	1153401
95580	**43.92**	**16556**	3478.98	1446242
$minsupp_u = 2, minsupp_t = 1,$ $minsupp_r = 1, minsupp_d = 1$				
25000	**97. 56**	**10982**	1022.12	1272988
50000	**188. 61**	**14671**	1987.06	1561992
70000	**263. 63**	**19548**	2876.02	1751258
95580	**528. 58**	**38762**	5965.94	2098452

TABLE 3 – Performances de QUADRICONS *vs*. celles de DATA PEELER sur le jeu de données MOVIELENS.

5 Conclusion et Perspectives

Nous avons considéré une nouvelle dimension dans les *folksonomies* et proposé l'algorithme QUADRICONS pour extraire les quadri-concepts fréquents. L'étude expérimentale a démontré que QUADRICONS offre une meilleure méthode pour l'extraction des quadri-concepts que l'algorithme DATA PEELER. Parmi les perspectives de nos travaux, nous pouvons citer la généralisation du problème au cas n-aire afin de prendre en compte des informations supplémentaires dans les *folksonomies* comme le timestamp ou encore la définition des règles d'association quadratiques relatives aux quadri-concepts.

Remerciements. Ce travail est partiellement financé par le projet franco-tunisien PHC Utique 11G141. Nous remercions les relecteurs pour leurs remarques constructives.

Références

[1] Biedermann, K. : Triadic Galois connections. In : General algebra and applications in discrete mathematics. pp. 23–33 (1997)

[2] Cerf, L., Besson, J., Robardet, C., Boulicaut, J.F. : Closed patterns meet n-ary relations. ACM TKDD 3, 3 :1–3 :36 (March 2009)

[3] Jäschke, R., Hotho, A., Schmitz, C., Ganter, B., Stumme, G. : Discovering shared conceptualizations in folksonomies. Journal of Web Semantics. 6, 38–53 (2008)

[4] Jelassi, M.N., Ben Yahia, S., Mephu Nguifo, E. : A scalable mining of frequent quadratic concepts in d-folksonomies. ArXiv e-prints (Dec 2012)

[5] Jelassi, M.N., Ben Yahia, S., Mephu Nguifo, E. : A personalized recommender system based on users' information in folksonomies. In : Proc. of the 5th Intl. Workshop on Web Intelligence & Communities WI&C at 22nd Intl. WWW conf, Rio de Janeiro, May'13 (2013)

[6] Lehmann, F., Wille, R. : A triadic approach to formal concept analysis. In : Proc. of the 3rd ICCS. pp. 32–43. Springer-Verlag, California, USA (1995)

[7] Mika, P. : Ontologies are us : A unified model of social networks and semantics. Journal of Web Semantics. 5(1), 5–15 (2007)

[8] Trabelsi, C., Jelassi, N., Ben Yahia, S. : Scalable mining of frequent tri-concepts. In : Proc. of the 15th PAKDD, Kuala Lampur, Malaysia. pp. 231–242 (2012)

[9] Voutsadakis, G. : Polyadic concept analysis. Order 19(3), 295–304 (2002)