Un système personnalisé de recommandation à partir de concepts quadratiques dans les folksonomies

Mohamed Nidhal Jelassi

N° d'ordre : D. U : 2693
E D S P I C : **756**

UNIVERSITE BLAISE PASCAL - CLERMONT II

UNIVERSITE TUNIS EL MANAR

ECOLE DOCTORALE

SCIENCES POUR L'INGENIEUR DE CLERMONT-FERRAND

T h è s e

Présentée par

Mohamed Nader Jelassi

(Master en informatique)

pour obtenir le grade de

DOCTEUR D'UNIVERSITÉ

SPECIALITE : Informatique

Titre de la thèse :
Un système personnalisé de recommandation à partir de concepts quadratiques dans les folksonomies

Soutenue publiquement le 11 mai 2016 devant le jury :

M. Mohand Boughanem	Président
M. Osmar Zaiane	Rapporteur et examinateur
M. Mohamed Mohsen Gammoudi	Rapporteur et examinateur
Mme. Amel Borgi	Rapporteur
M. Engelbert Mephu Nguifo	Directeur de thèse
M. Sadok Ben Yahia	Directeur de thèse

Dédicaces

À ma chère Maman Saloua

À mon cher Papa Chedly

À mon fantastique chat Ross

À mon cher frère Nidhal et son épouse Samira

À mon adorable nièce Yasmine

À mes tantes Wafa, Najet, les deux Saida et Saloua, à mes oncles, cousins et cousines

À mes frères Mohamed, Bilel, Khalil et Mehdi

Remerciements

Je tiens tout d'abord à remercier Monsieur **Mohand Boughanem**, pour avoir accepté de présider le jury de la soutenance.

Je tiens à exprimer ma gratitude à Monsieur **Mohamed Mohsen Gammoudi** et Monsieur **Osmar Zaïane**, pour avoir accepté de rapporter ce travail, ainsi qu'à Madame **Amel Borgi** et Monsieur **Nourine Lhouari** et de me faire l'honneur de faire partie du jury de cette thèse.

Je tiens à exprimer mon respect, ma gratitude et mes plus chaleureux remerciements à Monsieur **Sadok Ben Yahia**, Professeur à la Faculté des Sciences de Tunis pour avoir été mon directeur de thèse, mais pas que ! Encadrant récidiviste, puisqu'il a également mené au succès mes travaux de PFE et de Mastère. Je le remercie pour ses conseils, sa prise de recul sur mes travaux (que je n'avais hélas pas) et sa manière délicieuse de donner envie de faire de la recherche, sinon je n'aurai jamais réitéré l'expérience sept années de suite.

Je voudrai aussi remercier Monsieur **Engelbert Mephu Nguifo**, Professeur des Universités à l'Université Blaise Pascal de Clermont Ferrand pour avoir co-dirigé cette thèse. J'ai notamment apprécié la grande confiance qu'il a placée en moi et en mes recherches. J'ai jamais ressenti de pression ou de stress (du moins négatifs) relatifs à ma thèse grâce à lui. Il m'a donné "carte blanche" dès le début, et peu de professeurs eûssent été capables de me faire autant confiance. Il n'a jamais remis en doute ma capacité de mener à bien cette thèse, et c'est agréable de pouvoir mener ses travaux dans une quiétude et un environnement parfaits.

Je tiens à remercier tous ceux qui ont participé de près ou de loin à la réalisation de ce travail, particulièrement Madame Sihem Amer-Yahia pour m'avoir inspiré pour mon évaluation. Merci également à la ville de Clermont-Ferrand pour m'avoir octroyé une bourse d'études afin de terminer ma dernière année de thèse.

Hommages

Cette partie du mémoire ne concerne ni des remerciements -j'ai jamais aimé ce terme-, ni des dédicaces -le terme est faible et peu représentatif-, c'est donc sous forme d'hommages doublée d'un conte que je souhaite m'adresser à certaines personnes, sans qui je n'aurais jamais réussi à faire ma thèse pendant ces trois ou quatre années.

Il était une fois...
Une reine qui s'appelle Saloua, personnage le plus beau et le plus important de ce conte, et comment ! puisque elle est ma Maman. Je réalise son rêve le plus cher grâce à ce doctorat. Ma tâche aura été impossible sans son soutien, sa présence virtuelle et sa grande fierté à mon égard. Son amour infini a transpercé les mondes pour agir en tant que motivation numéro 1 dans ma thèse. Je suis bercé quotidiennement par les milliers de souvenirs qu'elle m'a laissé et qui m'accompagnent jour et nuit.

Ensuite, il y a un roi qui s'appelle Chedly. Personnage indispensable de ce conte et premier supporteur de mes travaux, il a largement contribué à me faciliter la tâche et à me mettre dans les meilleures conditions. Il a été sur tous les fronts, me laissant en manque de rien. Je lui suis infiniment reconnaissant. Indéboulonnable. Magique.

Le roi et la reine avaient un prince charmant, qui est mon frère Nidhal. Sans m'auto-accuser de plagiat, il est mon modèle depuis la nuit des temps. Véritable renard rusé, machine à conseils et grand frère affectueux, je garde notamment en tête les nombreux et inoubliables moments passés ensemble à Clermont et Saint-Etienne entre discussions, jeux, cinémas et balades dans les rues des deux villes. Enfin, c'est la seule personne à me battre régulièrement à FIFA, F1 et QI, ça en jette. Le prince charmant épousa, un an avant le début de ce conte, une charmante épouse, Samira, qui m'a souvent bien accueillie chez elle en servant des petits plats succulents. Elle est la soeur idéale qui manquait à l'histoire de ce conte. Ce couple a donné naissance à une princesse : Yasmine. Ma nièce préférée (c'est facile, j'en ai qu'une) est un rayon de soleil dans ma vie, source intarissable de bonheur et un sourire contagieux. Je peux donner libre cours à ma folie et à l'enfant qui sommeille toujours en moi, avec elle.

Tout bon conte comporte son lot de fées, et celui-ci ne déroge pas à la règle. Les fées qui ont veillé sur moi pendant cette thèse sont mes tantes : l'ange Wafa, personne la plus douce et la plus adorable qui puisse exister au monde, et accessoirement créatrice du gâteau au chocolat le plus délicieux qui soit. Najet, distributeur d'amour et prête à tout pour mon bonheur, Saida K. chez qui j'ai passé plusieurs séjours inoubliables à Sousse quand j'étais en panne d'inspiration ou cherchant l'isolement dans un cadre idyllique, Saida J. qui a toujours été présente pour moi et m'a souvent bien accueillie chez elle et enfin khalti Saloua qui m'a toujours agréablement accueillie chez elle comme son troisième fils, me donnant l'impression d'être en famille. Dans tout conte qui se respecte, la famille joue un rôle prépondérant et mes hommages vont aussi à mes cousins Nabil, féru de chats et de chocolats, Mourad, irremplaçable et drôle, mon coéquipier Mohamed et Sami "Contra" ainsi qu'à mon oncle Nejib, mon alter-ego littéraire et mon inoubliable oncle Radhouane. Ensuite, il y a une autre famille qui a beaucoup compté pour moi, les Moulahi : les parents, Ammar, Radhouane, Moez, et bien d'autres.

Le héros d'un conte, aussi courageux soit-il, a toujours besoin de chevaliers vaillants à ses côtés. Pour éviter des querelles vexatoires, procédons par ordre alphabétique ! Tout d'abord, Bilel, mon fidèle compagnon depuis si longtemps que j'ai l'impression qu'il a toujours été là. La recherche a commencé avec lui il y a 7 ans, et un jour j'en suis certain, elle reprendra avec lui. L'amitié avec lui se résume à une seule âme répartie en deux corps. Ensuite, Khalil, frère dévoué avec qui j'ai partagé des aventures et moments spéciaux et drôles, notamment en découvertes culinaires, à Tunis et aussi à Paris : oui, nous avons vu le métro parisien fermé. Mehdi est un ami à part, sorte de conteneur de toutes mes folies, souffre-douleur et également un véritable défouloir dans des périodes difficiles. Enfin, que dire sur Mohamed ? Il est l'acolyte parfait que j'ai eu à avoir à mes côtés. Le "Bro" a mille cordes à son arc : frère, encadrant, conseiller personnel, initiateur de challenges, coiffeur, chef cuisinier, coach sportif et j'en passe. Les discussions, nuits blanches, aventures rocambolesques passés ensemble chez moi, à Ezzahra, à Clermont-Ferrand, à Montpellier ou quotidiennement par téléphone/Skype m'ont aidé à traverser tranquillement ma thèse. Important. Indispensable. Vital. Définitivement. Plusieurs valets sont également à citer : Mansour, par sa serviabilité, dévouement, gentillesse et son grand coeur, Mostfa, mon autre frère, bon et loyal, Omar, pour les bons moments et Quentin, mon double parisien.

À présent, nous allons remonter un petit peu le temps, retrouver des personnages ayant joué chacun un rôle particulier durant les années Fac *: Imen, malgré son aversion inexplicable pour les chats, Ramzi, Nejeh, Khaoula, Inès, Malek, Hmida, Nejib, Amina, Mouadh, Nidhal. Une pensée également pour mes amis, un temps adversaires de foot, qui ont du supporter mon caractère horrible, ma mauvaise foi et mes tacles assassins lors d'inoubliables parties de foot : Rami, Aymen, Mohamed, Mansour...*

Remontons encore plus loin dans le temps, plusieurs années avant le début de ce conte, pour nous retrouver aux années Lycée. *Une pensée sincère et émue à tout le corps professoral toutes années confondues du lycée qui ont joué un rôle majeur dans ma formation, particulièrement, Madame Fitati, passée du statut de ma prof préférée à amie de la famille, Madame Karoui, certainement la prof la plus douce qui soit au monde et Madame Salem, prof géniale, et plusieurs autres que je revois très souvent. L'aventure est aussi chargée en rencontres magiques et d'amis irremplaçables qui sont toujours à mes côtés aujourd'hui : ma soeur Eya, Mouna, Siwar, Walid, Raghda, Neila,...*

Le temps passe et on finit par passer de l'autre côté de l'estrade pour devenir prof et passer le relais aux nouveaux. Pendant toutes ces années, j'ai fait la rencontre d'élèves magiques et talentueux qui m'ont pas mal appris et sont devenus amis au fil du temps : Ahmed, Ghassen, Wassim, Aness, Raouaa, Rim, Mathieu, Johann, Laura et j'en passe.

À Clermont-Ferrand, certes, je n'ai pas réussi à me faire un cercle d'amis mais, durant ma solitude heureuse, la meilleure compagnie que j'ai eu est Clermont elle-même. Un conte se déroule souvent dans un cadre idyllique et celui-ci n'échappe pas à la règle. J'ai eu la chance d'étudier dans une ville fantastiquement belle et envoûtante. Clermont a tout d'une belle femme sans en avoir les mauvais côtés. Entre le jardin Lecoq, les nombreux parcs, les rues de Jaude, les promenades au fil de l'Allier, l'ascension du Puy-de-dôme, les forêts vertigineuses, les beaux lacs, les vues à couper le souffle, les parcs animaliers, j'avoue avoir passé beaucoup plus de temps dans la nature auvergnate merveilleuse qu'assis dans mon bureau.
L'aveu est post-thèse, il y a prescription.
Il serait injuste de réduire les bons moments passés à Clermont à la ville. Il m'est arrivé plusieurs rencontres géniales avec des personnes superbes : Béatrice du LIMOS, douce et gentille, la maman du labo qui m'a aidé un nombre infini de fois (parfois des après-midi entières !) avec mes soucis administratifs auxquels je comprenais rien. Cécile, responsable de la bibliothèque dans ma résidence, avec qui j'ai eu des discussions enflammées pendant des heures autour de bouquins. Christine, gérante à Jaude, pour tous les fous rires qu'on a eu dans son magasin quand je passais chez elle. Kahina, Diego et Raksmey, qui ont un temps relativement court partagé mon bureau et avec qui j'ai beaucoup sympathisé.

Quittons à présent le monde des humains pour un monde meilleur, afin de rendre un hommage sincère et mérité à mon chat Ross, qui a tout de l'être humain sans en avoir les vices. Co-locataire idéal, il a été présent pour moi plus que n'importe quel humain en m'apportant l'affection nécessaire. Et également un hommage à tous les chats du refuge APA de Clermont, chez qui je passe régulièrement des moments forts. Enfin, pour terminer sur une note d'humour mais non dénuée de vérité, je rends hommage à Mulder et Scully. Les épisodes de X-Files passés en boucle m'ont accompagné bien des nuits.

Un système personnalisé de recommandation à partir de concepts quadratiques dans les folksonomies

Résumé : Les systèmes de recommandation ont acquis une certaine popularité parmi les chercheurs, où de nombreuses approches ont été proposées dans la littérature. Les utilisateurs des folksonomies partagent des items (*e.g.,* livres, films, sites web, etc.) en les annotant avec des tags librement choisis. Avec l'essor du Web 2.0, les utilisateurs sont devenus les principaux acteurs du système étant donné qu'ils sont à la fois les contributeurs et créateurs de l'information. Ainsi, il est important de répondre à leurs besoins en leur proposant une recommandation plus ciblée. Pour ce faire, nous considérons une nouvelle dimension dans une *folksonomie* classiquement composée de trois dimensions <utilisateurs,tags,ressources> et nous proposons une approche afin de regrouper les utilisateurs ayant des intérêts proches à travers des structures appelées concepts quadratiques. Ensuite, nous utilisons ces structures afin de proposer un nouveau système personnalisé de recommandation. Nous évaluons nos approches sur divers jeux de données du monde réel. Ces expérimentations ont démontré de bons résultats en termes de précision et de rappel ainsi qu'une bonne évaluation sociale. De plus, nous étudions quelques unes des métriques utilisées pour évaluer le systèmes de recommandations, comme la couverture, la diversité, l'adaptivité, la sérendipité ou encore la scalabilité. Par ailleurs, nous menons une étude de cas sur quelques utilisateurs comme complément à notre évaluation afin d'avoir l'avis des utilisateurs sur notre système. Enfin, nous proposons un nouvel algorithme qui permet de mettre à jour un ensemble de concepts triadiques sans avoir à re-scanner l'entière *folksonomie*. Les premiers résultats comparant les performances de notre proposition par rapport au redémarrage du processus d'extraction des concepts triadiques sur quatre jeux de données du monde réel a démontré son efficacité.

Mots clés : folksonomie, concepts quadratiques, systèmes de recommandation, personnalization, profil, temps, précision, rappel, métriques, propriétés, couverture, mise à jour.

A personalized recommender system based on quadri-concepts in folksonomies

Abstract : Recommender systems are now popular both commercially as well as within the research community, where many approaches have been suggested for providing recommendations. Folksonomies' users are sharing items (*e.g.,* movies, books, bookmarks, etc.) by annotating them with freely chosen tags. Within the Web 2.0 age, users become the core of the system since they are both the contributors and the creators of the information. In this respect, it is of paramount importance to match their needs for providing a more targeted recommendation. For such purpose, we consider a new dimension in a *folksonomy* classically composed of three dimensions <users,tags,resources> and propose an approach to group users with close interests through quadratic concepts. Then, we use such structures in order to propose our personalized recommendation system of users, tags and resources.

We carried out extensive experiments on two real-life datasets, *i.e.,* MovieLens and BookCrossing which highlight good results in terms of precision and recall as well as a promising social evaluation. Moreover, we study some of the key assessment metrics namely coverage, diversity, adaptivity, serendipity and scalability. In addition, we conduct a user study as a valuable complement to our evaluation in order to get further insights. Finally, we propose a new algorithm that aims to maintain a set of triadic concepts without the re-scan of the whole *folksonomy*. The first results comparing the performances of our proposition and the running from scratch the whole process over four real-life datasets show its efficiency.

Key words : folksonomy, quadratic concepts, recommender systems, personalization, profile, timestamp, precision, recall, metrics, properties, coverage update.

Table des matières

Liste des algorithmes

Liste des figures

Liste des tableaux

Introduction générale

Une *folksonomie* désigne un système de classification collaborative par les internautes. L'idée est de permettre à des utilisateurs de partager et de décrire des objets via des mots-clés (tags) librement choisis. Formellement, une *folksonomie* est composée de trois ensembles : un ensemble d'utilisateurs, un ensemble de tags (ou étiquettes) et un ensemble de ressources (films, livres, sites web, photos, etc.). Les utilisateurs sont les acteurs principaux du système et contribuent au contenu par l'ajout de ressources et l'affectation de tags. Cependant, il s'avère que le choix de tags et de ressources partagées par un utilisateur d'une *folksonomie* varie selon plusieurs critères : le genre, l'âge ou encore la profession de celui qui partage l'information. Cela a incité les chercheurs à proposer des systèmes de recommandation personnalisés afin de suggérer les tags et ressources les plus appropriés aux utilisateurs et de répondre aux besoins de chaque utilisateur. En effet, le domaine de personnalisation tente de fournir des solutions afin d'aider les utilisateurs à partager les bons tags et les bonnes ressources parmi le très grand nombre de données dans les folksonomies. Ainsi, un système de recommandation offre à l'utilisateur une liste de tags ou de ressources recommandés qu'il est susceptible d'aimer et lui permet de trouver plus facilement ses tags et ressources préférés dans la *folksonomie*. De plus, la personnalisation tente d'aider les utilisateurs à aborder le problème de surcharge d'information. Pour atteindre cet objectif, nous considérons une nouvelle dimension dans une *folksonomie*, classiquement composée de trois dimensions (utilisateurs, tags et ressources), et nous proposons une approche de regroupement des utilisateurs aux intérêts équivalents sous forme de structure appelées

concepts quadratiques. Cette quatrième dimension peut recouvrir différents aspects : par exemple le profil (genre, âge, profession, etc.), ou le temps si on veut étudier la dynamique temporelle des folksonomies. L'utilisation des concepts quadratiques est motivée par le fait que si on peut facilement étudier les tags utilisés par un seul utilisateur sur une ressource, il est évident de constater que la tâche devient rapidement intraitable lorsque cela implique plusieurs utilisateurs et plusieurs ressources. Grâce aux concepts quadratiques, nous pouvons résoudre ce problème. De plus, ces concepts sont une représentation réduite de la *folksonomie* qui peut contenir des milliers de quadruplets dans la vraie vie. Une fois extraits, ces concepts quadratiques sont utilisés pour notre algorithme de recommandation personnalisée multi-mode (utilisateurs, tags et ressources).

L'objectif de notre thèse consiste à proposer un algorithme d'extraction des concepts quadratiques et puis à mettre en place un système personnalisé de recommandation. Nous justifions la pertinence des approches proposées par une argumentation théorique, et évaluons leur efficacité à travers une étude expérimentale sur des jeux de données artificielles ou/et réelles. La validation expérimentale regroupe à la fois les temps d'exécution mais aussi la précision et le rappel du système de recommandation ainsi qu'un ensemble de métriques de qualité. Dans ce mémoire, nous allons proposer trois principaux axes complémentaires qui sont les suivants :

1. **un nouvel algorithme pour l'extraction des quadri-concepts à partir des *folksonomies*** : la tâche d'extraction des quadri-concepts à partir des *folksonomies* représente le contexte sous forme de quadruplets maximaux <utilisateurs,tags,ressources,variables>. La première problématique de la thèse est donc d'abord de proposer un algorithme qui permet d'extraire les quadri-concepts. Le principal algorithme concurrent pour l'extraction des quadri-concepts est l'algorithme DATA PEELER qui est générique, *i.e.*, capable d'extraire des concepts n-aires et donc capable d'extraire des quadri-concepts à partir de contextes à quatre dimensions. Toutefois, l'algorithme est très gourmand en mémoire puisqu'il stocke tout le contexte en mémoire tout en générant un nombre exponentiel de candidats. L'idée est donc de proposer un nouvel algorithme appelé QUADRICONS spécialement dédié pour la tâche d'extraction des quadri-concepts. A cet effet, nous avons recouru à la définition d'un nouvel opérateur de fermeture spécialement dédié aux contextes à quatre dimensions. De plus, QUADRICONS propose en premier lieu de localiser les quadri-générateurs afin de faciliter l'extraction des quadri-concepts.

Ces éléments sont considérés comme les plus petits éléments d'une classe d'équivalence dont la fermeture correspond à un quadri-concept. De point de vue performance, le principal point fort de Data Peeler, *i.e.*, sa généricité pour un contexte n-aire, constitue également sa faiblesse. En effet, lorsque le nombre de dimensions est égal à 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 de meilleures performances (en termes de temps d'exécution et mémoire consommée). De plus, QUADRICONS, ne stocke pas le jeu de données en mémoire avant l'extraction des quadri-concepts. Enfin, QUADRICONS génère moins de candidats grâce à l'habile détection des quadri-générateurs qui réduisent considérablement l'espace de recherche. L'étude expérimentale faite sur deux jeux de données du monde réel permettra de comparer les temps d'exécution et la mémoire consommée par les deux algorithmes.

2. **un nouveau système personnalisé de recommandation basé sur le profil des utilisateurs** Les quadri-concepts extraits par QuadriCons sont des quadruplets de la forme <utilisateurs, tags, ressources, variables>. Cette dernière dimension qui est la variable peut se présenter sous différents aspects (temps, profil, etc.). Ainsi, afin de proposer notre système personnalisé de recommandation, nous avons modélisé cette variable par le profil des utilisateurs. Plusieurs travaux se sont attelés à cette tâche dans la littérature. Cependant, la principale lacune de ces approches est qu'elles soient limitées à l'information <utilisateur, tag, ressource> ce qui donne des recommandations qui ne tiennent pas compte du profil de chacun des utilisateurs. C'est donc dans ce sens que nous avons exploité la quatrième variable des quadri-concepts en la modélisant par le profil des utilisateurs. Cela permet notamment d'éviter de proposer les mêmes recommandations à tous les utilisateurs et de personnaliser les réponses faites aux utilisateurs. Par ailleurs, en étudiant les travaux de la littérature, nous remarquons que la plupart des approches ne proposent pas une recommandation aux nouveaux utilisateurs ou ne tiennent pas compte de nouvelles données qui arrivent au fur et à mesure dans les folksonomies. Dans ce sens, nous nous appuyons sur le profil des utilisateurs afin de proposer des recommandations en fonction de ce dernier sans qu'il n'y ait besoin pour les nouveaux utilisateurs d'avoir un historique de tagging. Ainsi, la prise en compte du profil des utilisateurs au moment du processus de recommandation, ainsi que le recours aux quadri-concepts afin de regrouper les utilisateurs

ayant partagé le maximum de tags et de ressources en commun tout en ayant des profils équivalents, a permis d'améliorer les recommandations faites aux utilisateurs. Prenant en entrée l'utilisateur cible ainsi que son profil, notre algorithme proposé, *i.e.,* FolkRec, est donc capable de générer une recommandation personnalisée pour chaque utilisateur selon le mode de recommandation qu'il désire et selon le profil qu'il possède. L'étude expérimentale que nous menons comparera les valeurs de de précision et de rappel, deux métriques de qualité pour les systèmes de recommandations, de notre système avec les différentes approches. Par suite, nous proposons d'évaluer le système de recommandation PersoRec, sur des jeux de données du monde réel tel que le réseau social BookCrossing et le système de recommandation filmographique MovieLens, suivant un certain nombre de propriétés définis dans la littérature.

3. **Un algorithme pour la mise à jour des données dans les *folksonomies*** Enfin, pour la dernière approche de cette thèse, nous nous sommes penchés sur le problème de mise à jour des données dans les *folksonomies*. Nous proposons donc un algorithme appelé FolkIncr qui est capable d'incruster de nouvelles données dans un ensemble de données déjà extrait. Cela a été motivé par la dynamique des *folksonomies* où des milliers de données sont ajoutés sans que cela ne soit pris en compte par les algorithmes dits "statiques" de la littérature. En effet, les utilisateurs partagent des milliers de ressources de manière quotidienne et cela devient irréalisable de relancer à chaque fois le processus d'extraction des quadri-concepts. Cela nous a donc motivé pour concevoir notre algorithme afin de maintenir à jour l'ensemble des données sans devoir à rescanner l'entière *folksonomie*. Cela a pour effet que les recommandations ne s'appuient plus sur un ensemble statique de données mais sur un ensemble régulièrement mis à jour. Cela permettra de prendre en compte notamment les nouveaux tags et ressources qui viennent d'être ajoutés à la *folksonomie* et de proposer aux utilisateurs des recommandations récentes qui prennent en compte les dernières mises à jour du système.

Structure du mémoire

Le présent mémoire, décrivant le travail réalisé au cours de cette thèse, est composé de cinq chapitres :

Le **chapitre 1** présente les principales notions de base ainsi que l'état de l'art sur l'extraction des concepts quadratiques à partir des *folksonomies*. Ce chapitre introduit donc les notions mathématiques relatives à l'Analyse Formelle des Concepts, puis présente le principal algorithme dédié à la tâche d'extraction des concepts quadratiques.

Le **chapitre 2** définit d'abord un système de recommandation et ses différents types, puis, présente l'état de l'art et notamment les différents algorithmes, proposés dans la littérature, pour la recommandation dans les *folksonomies*. Ce chapitre mettra en évidence les différentes approches, techniques et autres stratégies utilisées pour présenter des solutions à la problématique de recommandation ainsi qu'une étude comparative de ces algorithmes. Cette synthèse permet aussi de situer nos contributions par rapport aux travaux antérieurs.

Le **chapitre 3** est consacré à l'introduction de notre nouvelle approche pour l'extraction des quadri-concepts à partir des *folksonomies* ainsi que la présentation de l'algorithme QuadriCons. Une étude comparative sera menée par rapport au principal algorithme de la littérature en termes de temps d'exécution et de mémoire consommée. Nous montrons, par ailleurs, que QuadriCons est valide et complet.

Le **chapitre 4** permet d'introduire un système personnalisé de recommandation dans les *folksonomies*. Ce système repose à la fois sur les quadri-concepts et sur des informations supplémentaires sur les utilisateurs dans le but de personnaliser les recommandations. De plus, une application sur des jeux de données du monde réel est décrite et interprétée pour mettre en exergue l'apport de notre contribution.

Le **chapitre 5** est consacré à l'introduction d'une approche pour la mise à jour des données dans les *folksonomies*. En effet, l'algorithme que nous introduisons va permettre d'incruster les nouvelles données ajoutées aux concepts déjà extraits sans devoir à redémarrer le processus d'extraction des quadri-concepts. L'étude expérimentale que nous présentons mettra en lumière l'apport de notre approche par rapport aux travaux de la littérature.

Le mémoire se termine par une conclusion qui résume l'ensemble de nos travaux et

présente quelques perspectives.

CHAPITRE 1

Extraction des concepts quadratiques dans les *folksonomies*

1.1 Introduction

Dans ce chapitre, nous allons, tout d'abord, introduire les principales notions de base et fondements mathématiques utilisés tout au long de ce mémoire. Nous allons commencer par présenter les définitions relatives au contexte triadique qui épouse la structure tripartite d'une *folksonomie* composée d'utilisateurs, de tags et de ressources. Nous présenterons également les principales notions de l'analyse formelle des concepts (AFC). À cet effet, nous définissons un générateur minimal, un opérateur de fermeture pour la *folksonomie* et les classes d'équivalences qui vont en découler. Par suite, nous évoquons le problème d'extraction des quadri-concepts et le principal algorithme de la littérature qui s'est attelé à cette tâche.

1.2 L'Analyse Formelle des Concepts (AFC)

L'analyse formelle des concepts est une méthode d'analyse de données spécialisée dans l'extraction d'un ensemble ordonné de concepts au sein d'un ensemble de données [Wille, 2009]. Elle est utilisée dans diverses applications comme [Sellami *et al.*, 2014]. Elle a été étendue depuis 1995 au cas triadique afin de traiter les données tri-dimensionnelles [Lehmann et Wille, 1995] [Ganter et Wille, 1999]. Cependant, très peu de travaux se sont intéressés à l'analyse triadique des concepts. Avec l'arrivée des *folksonomies* comme étant la structure centrale des réseaux sociaux [Hotho *et al.*, 2006]

[Fagnan *et al.*, 2014], la focalisation sur l'analyse triadique des concepts s'est considérablement accrue.

Le concept de *folksonomie*, structuré sous la forme d'un contexte triadique, est considéré comme faisant partie intégrante du Web 2.0 (ou web social) [1] [Abnar *et al.*, 2014]. Dans ce qui suit, nous définissons d'abord une *folksonomie*. Ensuite, nous définissons un tri-set et un tri-concept ainsi que l'opérateur de fermeture qui permet de calculer ce dernier. Auparavant, Wille et Ganter ont introduit dans [Ganter et Wille, 1999] une définition d'un contexte triadique qui est l'extension d'un contexte diadique classique par ajout d'une nouvelle colonne appelée *condition* permettant la représentation de données supplémentaires.

Définition 1 (CONTEXTE TRIADIQUE) *[Ganter et Wille, 1999] Un contexte triadique d'extraction (ou un contexte d'extraction) est un quadruplé* $\mathcal{K} = (\mathcal{E}, \mathcal{I}, \mathcal{C}, \mathcal{Y})$*, où* $\mathcal{E}$*,* $\mathcal{I}$ *and* $\mathcal{C}$ *sont des ensembles, et* $\mathcal{Y}$ *est une relation ternaire entre* $\mathcal{E}$*,* $\mathcal{I}$ *et* $\mathcal{C}$*, i.e.,* $\mathcal{Y} \subseteq \mathcal{E} \times \mathcal{I} \times \mathcal{C}$*. Les éléments de* $\mathcal{E}$*,* $\mathcal{I}$ *et* $\mathcal{C}$ *sont respectivement appelés objets, attributs, et conditions et* $(e, i, c) \in \mathcal{Y}$*, sous-entend que l'objet* e *est relatif à l'attribut* i *relativement à la condition* c*.*

1.2.1 Folksonomie

Un contexte triadique représente exactement la structure d'une *folksonomie* dont la définition est la suivante :

Définition 2 (FOLKSONOMIE)*[Hotho* et al.*, 2006] Une folksonomie est un ensemble de tuples* $\mathcal{F} = (\mathcal{U}, \mathcal{T}, \mathcal{R}, \mathcal{Y})$ *où*

- $\mathcal{U}$*,* $\mathcal{T}$ *et* $\mathcal{R}$ *sont des ensembles finis dont les éléments sont respectivement appelés utilisateurs, tags et ressources.*
- $\mathcal{Y} \subseteq \mathcal{U} \times \mathcal{T} \times \mathcal{R}$ *représente une relation triadique dont chaque* $y \subseteq \mathcal{Y}$ *peut être représenté par un triplet :*

$$y = \{(u, t, r) \mid u \in \mathcal{U}, t \in \mathcal{T}, r \in R\}.$$

ce qui signifie que l'utilisateur u *a annoté la ressource* r *par le tag* t*.*

Une *folksonomie* est donc un système de classification qui dérive de la création collaborative de son contenu par les utilisateurs. Cette pratique est appelée *"Social*

1. *https ://fr.wikipedia.org/wiki/Web_2.0*

Tagging", un processus par lequel de nombreux utilisateurs ajoutent des données sous forme de tags pour partager des ressources. C'est un concept qui est différent d'une ontologie [Roussey *et al.*, 2006] où la classification est faite par des experts. En d'autres termes, il s'agit d'un support du Web 2.0 pour la classification de ressources. L'annotation des ressources par les tags facilite ainsi le partage et la recherche d'information **Exemple** : DEL.ICIO.US [2], FLICKR [3], YOUTUBE [4], MOVIELENS [5], LAST.FM [6], etc.

Exemple 1 *Le Tableau 1.1 illustre un exemple d'une folksonomie* $\mathcal{F}$ *avec* $\mathcal{U} = \{u_1, u_2, u_3, u_4\}$, $\mathcal{T} = \{t_1, t_2, t_3, t_4\}$ *et* $\mathcal{R} = \{r_1, r_2, r_3\}$.

Notons que chaque "$\times$*" représente une relation triadique entre un utilisateur appartenant à* $\mathcal{U}$*, un tag appartenant à* $\mathcal{T}$ *et une ressource annotée appartenant à* $\mathcal{R}$*. Par exemple, l'utilisateur* u_1 *a tagué la ressource* r_1 *par le biais des tags* t_2, t_3 *et* t_4.

$\mathcal{F}/\mathcal{R}$	r_1				r_2				r_3			
$\mathcal{U}/\mathcal{T}$	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		×	×	×		×	×	×		×	×	×
u_2		×	×	×	×	×	×	×	×	×	×	×
u_3		×	×	×	×	×	×	×	×	×	×	×
u_4		×	×		×			×	×			×

Tableau 1.1 — Une *folksonomie*.

1.2.2 Tri-set et Tri-concept

Un tri-set est un triplet particulier de la *folksonomie* dont chaque partie peut contenir plus d'un élément. La définition d'un tri-set est donné comme suit :

Définition 3 (UN TRI-SET (FRÉQUENT)) *[Ganter et Wille, 1999] Soit* $\mathcal{F} = (\mathcal{U}, \mathcal{T}, \mathcal{R}, \mathcal{Y})$ *une folksonomie/contexte triadique. Un tri-set de* $\mathcal{F}$ *est un triplet* (A, B, C) *avec* $A \subseteq \mathcal{U}$, $B \subseteq \mathcal{T}$, $C \subseteq \mathcal{R}$ *tel que* $A \times B \times C \subseteq \mathcal{Y}$*. Un tri-set* (A, B, C) *de* $\mathcal{F}$ *est dit fréquent lorsque* $\mid A \mid \geq minsupp_u$, $\mid B \mid \geq minsupp_t$ *et* $\mid C \mid \geq minsupp_r$*, où* $minsupp_u$, $minsupp_t$ *et* $minsupp_r$ *sont des seuils définis par l'utilisateur.*

2. *https ://delicious.com*
3. *https ://www.flickr.com*
4. *https ://www.youtube.com*
5. *https ://www.last.fm/fr/*
6. *https ://www.flickr.com*

Exemple 2 *Dans la folksonomie du Tableau 1.1,* TS_1=($\{u_1, u_2, u_3\}$, $\{t_2, t_3\}$, $\{r_1\}$) *et* TS_2=($\{u_2, u_3, u_4\}$, $\{t_1\}$, $\{r_2\}$) *sont deux exemples de tri-sets de* $\mathcal{F}$. *Ils représentent deux différents groupes d'utilisateurs ayant annoté des ressources avec un certain ensemble de tags.*

Aussi bien que dans le cas diadique classique où l'ensemble des itemsets fréquents est très grand, l'ensemble des tri-sets fréquents est très élevé et redondant. Cependant, cet ensemble peut très bien être condensé sans perte d'information. De ce fait, nous allons appliquer la notion de fermeture au cas triadique, *i.e.*, nous considérons un sous-ensemble de tri-sets contenant la même information, appelé *l'ensemble des tri-concepts fréquents*. Dans ce qui suit, nous présentons donc une adaptation de la notion d'*"itemsets fermés fréquents"* [Agrawal *et al.*, 1993] au cas triadique.

Définition 4 (CONCEPT TRIADIQUE (FRÉQUENT)) *Un concept triadique (ou un tri-concept) d'une folksonomie* $\mathcal{F} = (\mathcal{U}, \mathcal{T}, \mathcal{R}, \mathcal{Y})$ *est un triplet* (U, T, R) *où* $U \subseteq \mathcal{U}$, $T \subseteq \mathcal{T}$, *et* $R \subseteq \mathcal{R}$ *avec* $U \times T \times R \subseteq \mathcal{Y}$ *tel que le triplet* (U, T, R) *est maximal, i.e., il est impossible d'augmenter un ensemble sans faire diminuer l'un des deux autres. Un tri-concept est dit fréquent s'il est un tri-set fréquent. L'ensemble des tri-concepts fréquents de* $\mathcal{F}$ *est représenté par* $\mathcal{TC} = \{TC \mid TC = (U, T, R) \in \mathcal{Y}$ *est un tri-concept}.*

Notons que pour un tri-concept (U, T, R), les ensembles U, R et T sont respectivement appelés ***Extent***, ***Intent***, et `Modus` du tri-concept (U, T, R).

Exemple 3 *D'après la folksonomie du Tableau 1.1,* TS_1=($\{u_1, u_2, u_3\}$, $\{t_2, t_3\}$, $\{r_1\}$) *n'est pas un tri-concept de* $\mathcal{F}$. *Cependant,* TC_1=($\{u_1, u_2, u_3\}$, $\{t_2, t_3, t_4\}$, $\{r_1, r_2, r_3\}$) *est un tri-concept de* $\mathcal{F}$ *: c'est l'ensemble maximal de tags et de ressources partagées par* u_1, u_2 *et* u_3.

Dans ce qui suit, nous présentons la définition d'un nouvel opérateur de fermeture pour un contexte triadique ainsi que celle d'une classe d'équivalence. Ensuite, nous définissons un tri-générateur minimal, qui est l'extension de celle d'un générateur minimal au cas triadique. Mais tout d'abord, nous définissons formellement le problème d'extraction des tri-concepts.

Le problème d'extraction des tri-concepts

Soit $\mathcal{F} = (\mathcal{U}, \mathcal{T}, \mathcal{R}, \mathcal{Y})$ une *folksonomie* et soient $minsupp_u$, $minsupp_t$, $minsupp_r$

trois seuils minimaux de support fixés par l'utilisateur. Le problème d'extraction des tri-concepts fréquents consiste à déterminer tous les tri-concepts (U, T, R) de $\mathcal{F}$ tel que $\mid U \mid \geq minsupp_u$, $\mid T \mid \geq minsupp_t$ et $\mid R \mid \geq minsupp_r$.

Ainsi, l'extraction des tri-concepts revient à découvrir tous les ensembles d'utilisateurs, tags et ressources (U, T, R) tels que pour chaque triplet, tous les utilisateurs de U ont assigné les tags de T aux ressources de R. De plus, la cardinalité de chacun de ces ensembles doit vérifier le seuil minimal de support correspondant [Jäschke *et al.*, 2008].

Ce problème d'extraction des tri-concepts est plus dur que celui de l'extraction des itemsets fermés dans un contexte diadique classique. En effet, nous avons à considérer trois seuils de supports au lieu d'un seul, étant donné le nombre de dimensions du contexte triadique. Cependant, aussi bien que dans le cas diadique, ces trois seuils de supports vérifient les contraintes d'anti-monotonie : Si le tri-set (U_1, T_1, R_1) avec U_1 maximal n'est pas fréquent par rapport au seuil de support $minsupp_u$, alors tous les tri-sets (U_1, T_2, R_2) avec $T_1 \subseteq T_2$ et $R_1 \subseteq R_2$ sont également non fréquents. La même propriété est vérifiée pour les deux autres dimensions. Par exemple, si le tri-set $\{\{u_5, u_6, u_7\}, \{t_2, t_3\}, \{r_1, r_2\}\}$ n'est pas fréquent pour $minsupp_u = 4$ alors le tri-set $\{\{u_5, u_6, u_7\}, \{t_2, t_3, t_4\}, \{r_1, r_2\}\}$ est également infréquent.

1.2.3 Opérateur de fermeture, classe d'équivalence et générateurs minimaux

Lehmann et Wille ont introduit dans [Lehmann et Wille, 1995] deux opérateurs de dérivation pour la construction des concepts triadiques. Toutefois, ces opérateurs ne sont utilisables que sur les contextes diadiques, *i.e.*, la *folksonomie* doit auparavant être transformée en trois contextes diadiques, afin que ces opérateurs puissent être appliqués pour l'extraction des tri-concepts.

Par conséquent, nous présentons, dans ce qui suit, un nouvel opérateur de fermeture pour une *folksonomie*.

Définition 5 (Opérateur de fermeture)*[Trabelsi* et al.*, 2012] Soit T = (A, B, C) un tri-set de $\mathcal{F}$. Un opérateur h est défini comme suit :*

$$h(T) = h(A, B, C) = (U, T, R) \text{ tel que } U = \{u_i \in \mathcal{U} \mid (u_i, t_i, r_i) \in \mathcal{Y} \; \forall \; t_i \in B, \forall \; r_i \in C\}$$

$$\wedge\ T = \{t_i \in \mathcal{T} \mid (u_i, t_i, r_i) \in \mathcal{Y}\ \forall\ u_i \in U, \forall\ r_i \in C\}$$
$$\wedge\ R = \{r_i \in \mathcal{R} \mid (u_i, t_i, r_i) \in \mathcal{Y}\ \forall\ u_i \in U, \forall\ t_i \in T\}$$

Proposition 1 *L'opérateur h est un opérateur de fermeture.*

Preuve 1 *L'opérateur de fermeture* h *doit vérifier les trois propriétés suivantes [Couch et Chiarini, 2008] :*
(1) Extensivité
Soit $T = (A, B, C)$ un tri-set de $\mathcal{F} \Rightarrow h(T) = (U, T, R)$ tel que :
$U = \{u_i \in \mathcal{U} \mid (u_i, t_i, r_i) \in \mathcal{Y}\ \forall\ t_i \in B, \forall\ r_i \in C\} \supseteq A$ étant donné que $(u_i, t_i, r_i) \in \mathcal{Y}\ \forall\ u_i \in A, \forall\ t_i \in B, \forall\ r_i \in C$

De plus, $T = \{t_i \in \mathcal{T} \mid (u_i, t_i, r_i) \in \mathcal{Y}\ \forall\ u_i \in U, \forall\ r_i \in C\} \supseteq B$ puisque $U \supseteq A$. Finalement, $R = \{r_i \in \mathcal{R} \mid (u_i, t_i, r_i) \in \mathcal{Y}\ \forall\ u_i \in U, \forall\ t_i \in T\} \supseteq C$ puisque $U \supseteq A$ et $T \supseteq B$.

Ainsi, $(A, B, C) \subseteq (U, T, R) \Rightarrow T \subseteq h(T)$

(2) Idempotence
Soit $T = (A, B, C)$ un tri-set de $\mathcal{F} \Rightarrow h(T) = (U, T, R) \Rightarrow h(U, T, R) = (U', T', R')$ avec :

$U' = \{u_i \in \mathcal{U} \mid (u_i, t_i, r_i) \in \mathcal{Y}\ \forall\ t_i \in T, \forall\ r_i \in R\} = U$ [Lehmann et Wille, 1995].

De plus, $T' = \{t_i \in \mathcal{T} \mid (u_i, t_i, r_i) \in \mathcal{Y}\ \forall\ u_i \in U, \forall\ r_i \in C\} = T$ [Lehmann et Wille, 1995].

Finalement, $R' = \{r_i \in \mathcal{R} \mid (u_i, t_i, r_i) \in \mathcal{Y}\ \forall\ u_i \in U, \forall\ t_i \in T'\} = R$ [Lehmann et Wille, 1995].

Ainsi, $(U', T', R') = (U, T, R) \Rightarrow h(h(T)) = h(T)$

(3) Isotonicité
Soit $T = (A, B, C)$ et $T' = (A', B', C')$ deux tri-sets de $\mathcal{F}$ avec $T \subseteq T' \Rightarrow h(T) = (U, T, R)$ et $h(T') = (U', T', R')$ tel que :

D'une part, $U' = \{u_i \in \mathcal{U} \mid (u_i, t_i, r_i) \in \mathcal{Y}\ \forall\ t_i \in B', \forall\ r_i \in C'\}$.

et $U = \{u_i \in \mathcal{U} \mid (u_i, t_i, r_i) \in \mathcal{Y}\ \forall\ t_i \in B, \forall\ r_i \in C\}$. $\Rightarrow U' \supseteq U$ puisque $B \subseteq B'$ et $C \subseteq C'$ [Lehmann et Wille, 1995].

D'autre part, $T = \{t_i \in \mathcal{T} \mid (u_i, t_i, r_i) \in \mathcal{Y}\ \forall\ u_i \in U, \forall\ r_i \in C\}$, $R = \{r_i \in \mathcal{R} \mid (u_i, t_i, r_i) \in \mathcal{Y}\ \forall\ u_i \in U, \forall\ t_i \in T\}$, $T' = \{t_i \in \mathcal{T} \mid (u_i, t_i, r_i) \in \mathcal{Y}\ \forall\ u_i \in U, \forall\ r_i \in$

C} et R′ = {r_i ∈ ℛ | (u_i, t_i, r_i) ∈ 𝒴 ∀ u_i ∈ U, ∀ t_i ∈ T′}

⇒ T ⊆ T′ puisque U ⊆ U′ et R ⊆ R′ étant donné que U ⊆ U′ et T ⊆ T′[Lehmann et Wille, 1995].

Ainsi, (U, T, R) ⊆ (U′, T′, R′) ⇒ h(T) ⊆ h(T′)

Selon (1), (2) et (3), h est un opérateur de fermeture pour un contexte triadique / folksonomie.

Formellement, $h(T)$ est le tri-set le plus large d'une *folksonomie* $\mathcal{F}$ qui contient les ensembles maximaux de tags et de ressources partagés par un même groupe d'utilisateurs.

Exemple 4 *Considérons la folksonomie $\mathcal{F}$ donnée par le Tableau 1.1, nous avons $h(\{u_1, u_2, u_3\}, \{t_2, t_3\}, \{r_1\}) = (\{u_1, u_2, u_3\}, \{t_2, t_3, t_4\}, \{r_1, r_2, r_3\})$.*

Ainsi, l'opérateur de fermeture h appliqué à un tri-set génére un concept triadique. De plus, l'opérateur de fermeture partitionne l'espace de recherche en classes d'équivalence, dont la définition est donnée dans ce qui suit.

Définition 6 *[Trabelsi* et al.*, 2012]* (Classe d'équivalence) *Soient $T_1 = (A_1, B_1, C_1)$ et $T_2 = (A_2, B_2, C_2)$ deux tri-sets de $\mathcal{F}$ et $TC \subseteq \mathcal{TC}$. T_1 et T_2 appartiennent à la même classe d'équivalence représentée par le tri-concept TC, i.e., $T_1 \equiv_{TC} T_2$ ssi $h(T_1) = h(T_2) = TC$.*

Les générateurs minimaux (GMs), *aka* clés minimales, jouent un rôle important dans plusieurs problèmes théoriques et pratiques impliquant les systèmes de fermeture qui résident dans les théories de graphes, les bases de données relationnelles, la fouille de données pour ne citer que. Dans ce qui suit, nous introduisons une extension de la définition d'un générateur minimal au contexte triadique.

Définition 7 *[Trabelsi* et al.*, 2012]* (Tri-générateur minimal) *Soit $g = (A, B, C)$ un tri-set tel que $A \subseteq \mathcal{U}$, $B \subseteq \mathcal{T}$ et $C \subseteq \mathcal{R}$ et $TC \subseteq \mathcal{TC}$ un tri-concept. Le triplet g est un tri-générateur minimal (ou tri-générateur) de TC ssi $h(g) = TC$ et $\nexists\, g_1 = (A_1, B_1, C_1)$ tel que $A = A_1$, $(B_1 \subseteq B \wedge C_1 \subset C) \vee (B_1 \subset B \wedge C_1 \subseteq C)$ et $h(g) = h(g_1) = TC$.*

Exemple 5 *Le tri-concept* $TC = (\{u_1, u_2, u_3\}, \{t_2, t_3, t_4\}, \{r_1, r_2, r_3\})$ *a deux tri-générateurs* $g_1 = (\{u_1, u_2, u_3\}, \{t_4\}, \{r_1\})$ *et* $g_2 = (\{u_1, u_2, u_3\}, \{t_3\}, \{r_2\})$. *Cependant,* $g_3 = (\{u_1, u_2, u_3\}, \{t_2, t_3\}, \{r_1, r_2\})$ *n'est pas un tri-générateur de TC puisque* $\exists\ g_2$ *tel que* g_2.extent=g_3.extent, g_2.intent $\subseteq$ g_3.intent *et* g_2.modus $\subset$ g_3.modus.

Reposant sur les définitions ainsi introduites, nous proposons, dans ce qui suit, l'algorithme TriCons pour l'extraction des tri-concepts fréquents à partir d'une *folksonomie*.

1.2.4 TriCons : un algorithme pour l'extraction des tri-concepts

Dans [Trabelsi *et al.*, 2012], nous avons introduit l'algorithme TriCons dédié à l'extraction des tri-concepts fréquents à partir d'une *folksonomie*. TriCons opère en 3 étapes.

1. L'extraction des tri-générateurs.
2. Le calcul du `modus` des tri-concepts.
3. Le calcul de l'`intent` des tri-concepts.

Afin d'extraire les tri-concepts fréquents, TriCons commence par balayer l'espace de recherche dans le but d'extraire les tri-générateurs. Le fait que les tri-générateurs soient les plus petits éléments d'une classe d'équivalence rend leur détection et parcours largement plus facile que le balayage de toute la *folksonomie*. Ces tri-générateurs, étant les représentants des différentes classes d'équivalence, rendent le calcul des tri-concepts moins ardu. De plus, la taille de l'ensemble des tri-générateurs étant souvent largement moins grande que la taille d'une *folksonomie*, TriCons aura beaucoup moins de données à manipuler. Et bien que sa complexité théorique soit exponentielle [Trabelsi *et al.*, 2012], les tests expérimentaux ont démontré que l'algorithme réalise des temps d'exécution acceptables sur des jeux de données du monde réel [7].

1.3 Extraction des quadri-concepts

Dans ce qui suit, nous présentons le principal algorithme dédié à la tâche de l'extraction des quadri-concepts dans les contextes à quatre dimensions.

7. *http ://www.isima.fr/ jelassi/triconsrecap2*

1.3.1 Problématique et motivation sur la quatrième dimension

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, dans les structures des *folksonomies*, plusieurs millions de données sont ajoutées quotidiennement. Par exemple, dans les réseaux sociaux à très large échelle comme Facebook ou encore Twitter, ce sont des centaines de milliers de ressources ou de tweets qui sont partagés tous les jours. Dans les *folksonomies* plus réduites comme MovieLens ou encore BookCrossing, on parle de centaines de partage d'information quotidiens. Ainsi, en raison des très grandes tailles des *folksonomies*, plusieurs travaux se sont focalisées sur l'extraction de représentation concises (sans perte d'information) de motifs intéressants. Ces structures, qui doivent être une représentation réduite du trop grand nombre de données -souvent redondantes- dans les *folksonomies*, doivent également répondre au critère de non perte d'information. En effet, il doit être possible de trouver l'ensemble de données à partir de cet ensemble réduit et concis de concepts représentatifs. Ces structures appelées concepts triadiques ont fait l'objet d'une étude dans la littérature dans [Jäschke *et al.*, 2008] [Ji *et al.*, 2006] [Cerf *et al.*, 2009]. 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 triadiques ou encore quadratiques à partir de 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 s'avère être 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 mémoire, nous traitons la quatrième dimension de manière indifférente pour l'aspect méthodologique, mais afin d'extraire des résultats à partir de jeux de données du monde réel, nous focaliserons plus tard soit sur l'aspect profil soit sur l'aspect temps. La tâche spécifique d'extraction de quadri-concepts vise à extraire un ensemble de quadruplets fréquents, dont chaque quadruplet (U, T, R, V) consiste en un ensemble U d'utilisateurs, un ensemble T de tags, un ensemble R de ressources et un ensemble V de variables. Ces quadruplets, appelés *quadri-concepts fréquents*, vérifient la propriété suivante : chaque utilisateur de U avec une variable

de V a tagué 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. Dans ce qui suit, nous introduisons le principal algorithme de la littérature pour la tâche d'extraction des concepts quadratiques.

1.3.2 DATA PEELER : un algorithme pour l'extraction des concepts n-aires (Cerf *et al.*, 2009)

En 2009, Cerf et *al.* ont proposé l'algorithme DATA PEELER [Cerf, 2010] [Cerf *et al.*, 2009] pour l'extraction des concepts n-aires à partir de contextes n-aires. DATA PEELER utilise une approche en profondeur d'abord et stocke le contexte dans une structure d'arbre binaire afin de dériver l'ensemble des concepts n-aires. En particulier, lorsque n est égal à 4, l'algorithme DATA PEELER est capable d'extraire les quadri-concepts. DATA PEELER commence par stocker le contexte quadratique dans sa totalité dans une structure appelée *"arbre binaire"* ; le recours à cette structure permet de parcourir les données plus rapidement. Ensuite, de manière récursive, les noeuds de l'arbre sont parcourus afin de calculer la fermeture du noeud courant en stockant dans une pile les éléments à ajouter au quadri-concept. À chaque niveau, le noeud courant est partitionné en deux nouveaux noeuds après sélection de l'élément à énumérer ; cet élément est indépendant de l'exécution de l'algorithme. DATA PEELER termine lorsque tous les quadri-concepts sont extraits grâce à ce parcours de l'arbre binaire. Enfin DATA PEELER prend également comme entrée des seuils minimaux de supports dont le nombre est égal au nombre de dimensions du contexte n-aire. Ainsi, DATA PEELER permet, si les supports minimaux sont renseignés, de générer les concepts n-aires fréquents.

1.3.3 Exemple illustratif

Pour illustrer un exemple d'exécution de l'algorithme DATA PEELER, nous avons choisi de prendre un exemple d'une *folksonomie* réduite présentée par le Tableau 1.2 où $\mathcal{U}$=$\{1,2,3\}$, $\mathcal{T}$=$\{a,b,c\}$ et $\mathcal{R}$=$\{\alpha,\beta,\gamma\}$ et d'exécuter une instance de DATA PEELER lorsque le nombre de dimensions n est égal à 3. Nous avons également pris les seuils

$\mathcal{U}/\mathcal{R}$-$\mathcal{T}$	α			β			γ		
	a	b	c	a	b	c	a	b	c
1	×	×	×	×	×		×	×	
2	×	×	×		×	×	×	×	
3	×	×	×		×	×	×	×	×

Tableau 1.2 — Une *folksonomie* réduite.

minimaux de supports suivants : $minsupp_u = 2$, $minsupp_t = 1$ et $minsupp_r = 1$. La Figure 1.1 montre une trace d'exécution de DATA PEELER sur cette *folksonomie*. Chaque noeud de l'arbre représente un couple de tri-sets complémentaires par rapport au noeud père. Le premier noeud est un couple d'un tri-set d'ensembles vides et d'un tri-set d'ensembles maximaux de $\mathcal{U}$, $\mathcal{T}$ et $\mathcal{R}$. Suite à une première décomposition aléatoire et indépendante de l'exécution de DATA PEELER, chaque noeud se décompose pour donner deux noeuds fils après sélection de l'élément à énumérer. Par exemple, le noeud ((1 2,a b,α),(3,c,β γ)) est décomposé suivant l'élément γ donnant un noeud fils (à gauche) contenant cet élément et un noeud fils (à droite) ne contenant pas γ. Le premier tri-concept {{1,2,3}, {a,b,c}, {α}} est ainsi obtenu après décompositions successives suivant les éléments γ, c, 3 et α. Cependant, lorsqu'un tri-concept est extrait, la décomposition ne s'arrête pas car ce dernier peut engendrer un nouveau tri-concept via de nouvelles décompositions (chaque tri-concept est entouré d'un carré sur la Figure 1.1). Par exemple, le tri-concept {{1,2,3}, {a,b}, {α,γ}} permet de créer un nouveau tri-concept suite à une décomposition suivant, respectivement, les éléments β et a.

1.3.4 Critique de l'algorithme et Contributions

Le principal souci de DATA PEELER est la stratégie qu'il adopte pour parcourir le contexte. En effet, DATA PEELER utilise une approche en profondeur d'abord qui comporte le risque de générer des branches infinies. Ainsi, les performances de DATA PEELER se détériorent rapidement lorsque une ou plusieurs dimensions du contexte contient un grand nombre d'éléments différents. Ainsi, DATA PEELER n'affiche de bonnes performances que lorsque les différentes dimensions du contexte ne contiennent pas plus que quelques centaines d'éléments différents ; il est même, selon ses auteurs, meilleur que ses concurrents [Jäschke *et al.*, 2008] [Ji *et al.*, 2006] sur un contexte n-dimensionnel [Cerf *et al.*, 2009]. Il est cependant à noter que cela est rarement le cas

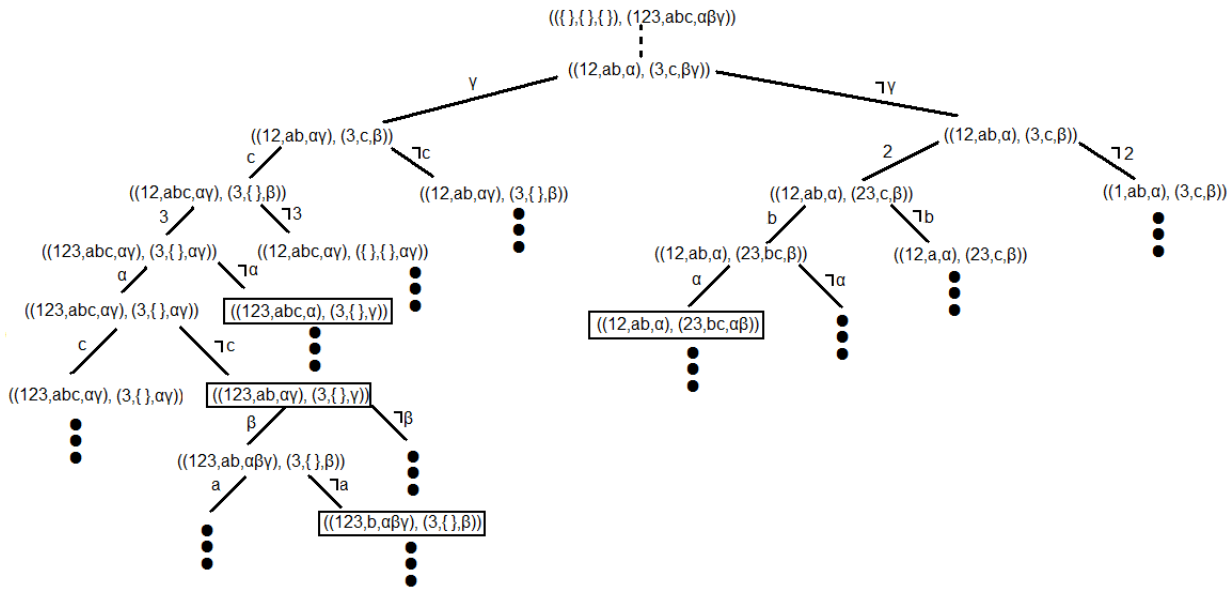


Figure 1.1 — Exemple illustratif de l'Algorithme DATA PEELER sur la *folksonomie* décrite par le Tableau 1.2

dans les *folksonomies* où le nombre d'utilisateurs, tags et ressources dépassent aisément les milliers. Ainsi, la stratégie utilisée par DATA PEELER, qui consiste au parcours de l'arbre binaire pour extraire les n-concepts, devient inefficace et conduit à des branches infinies et un calcul très complexe des n-concepts. De plus, un inconvénient majeur de cette approche réside dans le fait qu'elle stocke tout le jeu de données en mémoire avant de procéder à l'extraction des concepts ce qui ralentit considérablement ce processus. En effet, dans des jeux de données du monde réel, le nombre de données peut atteindre les milliers et le nombre de candidats peut être également très grand. Stocker toutes les données en mémoire s'avère donc être une stratégie peu recommendable pour ce genre de contexte.

1.4 Conclusion

Tout au long de ce chapitre, nous avons, tout d'abord, présenté les principales notions mathématiques de l'analyse formelle des concepts que nous allons utiliser pendant ce mémoire. Nous avons également présenté l'adaptation des notions de générateurs minimaux, opérateur de fermeture et classes d'équivalence au cas triadique représenté par une *folksonomie*. Ensuite, nous avons formellement présenté le problème d'extraction

des tri-concepts qui constituent des conceptualisations cachées au sein des *folksonomies*. Enfin, nous avons présenté le principal algorithme de la littérature consacré au problème d'extraction des quadri-concepts dans les *folksonomies*. Cette étude a mis en lumière la nécessité de proposer un nouvel algorithme qui se consacre à l'extraction des quadri-concepts à partir des *folksonomies* et qui se base sur un opérateur de fermeture spécialement dédié à ce type de contexte. Cette première contribution sera présentée dans le troisième chapitre.

Dans le prochain chapitre, nous allons survoler les principales approches de systèmes de recommandations dans les *folksonomies* qui sont réparties en quatre catégories. Une étude critique de ces travaux sera menée, ainsi qu'une présentation d'un certain ensemble de propriétés comme la couverture, le cold start ou encore la scalabilité pour les systèmes de recommandations.

CHAPITRE 2

Systèmes de recommandations dans les *folksonomies*

2.1 Introduction

Dans ce chapitre, nous définissons les différents types de systèmes de recommandations ainsi que les différentes métriques qui permettent de jauger la qualité des recommandations. Ainsi, les mesures classiques comme la précision et le rappel ou encore de nouvelles propriétés comme la couverture, la nouveauté ou encore la sérendipité seront également définies. Ensuite, nous survolons les principaux travaux de la littérature qui s'y sont consacrés. Ces approches sont réparties en plusieurs catégories : celles qui sont basées sur la popularité, celles qui sont basées sur les mesures de similarité, celles qui combinent l'historique de tagging et les mesures de similarités ou encore celles qui sont basées sur le profil des utilisateurs. Ce chapitre est ponctué par un tableau comparatif qui mettra en lumière les lacunes des approches existantes par rapport aux propriétés des systèmes de recommandations.

2.2 Systèmes de Recommandations

Dans ce qui suit, nous présentons les différents types de systèmes de recommandations, puis nous introduisons les différentes propriétés relatives à ces derniers.

2.2.1 Types de systèmes de recommandations

Les systèmes de recommandation (SR) sont une forme spécifique de filtrage de l'information visant à présenter les éléments d'information (films, musique, livres, news, images, pages Web, etc.) qui sont susceptibles d'intéresser l'utilisateur [Adomavicius et Tuzhilin, 2005]. Généralement, un système de recommandation permet de comparer le profil d'un utilisateur à certaines caractéristiques de référence, et cherche à prédire la "note" ou la "préférence" que donnerait un utilisateur à un item donné [Ricci *et al.*, 2011]. Les systèmes de recommandation sont devenus très populaires ces dernières années et sont appliqués dans de nombreuses applications. Parmi les applications les plus populaires, on cite les films, les musiques, les actualités, les livres, les articles scientifiques et les produits en général. Il existe également des systèmes de recommandations d'experts, de blagues, de restaurants, de services financiers, d'assurance-vie, de personnes (sites de rencontres en ligne) ou encore de followers sur twitter [Felfernig *et al.*, 2007].

Il existe deux méthodes différentes par lesquelles les SR permettent de générer une liste de recommandations : (i) la méthode de filtrage collaboratif (ou *collaborative filtering*) qui permet de construire un modèle à partir du comportement passé de l'utilisateur (les ressources précédemment partagées ou/et les tags (ou notes) données à ces ressources) et des comportements des autres utilisateurs ; ensuite, à partir de ce modèle, le SR est capable de prédire les ressources (ou les tags pour ces ressources) qui sont susceptibles d'intéresser l'utilisateur [Melville et Sindhwani, 2010] ; (ii) la méthode basée sur le contenu (ou *content-based method*) utilise des caractéristiques et propriétés des ressources afin de recommander d'autres ressources qui possèdent des propriétés similaires [Mooney et Roy, 2000]. Ces deux méthodes peuvent également être combinées, on parle alors de systèmes de recommandations hybrides.

Les SR présentent une alternative utile et intéressante aux algorithmes de recherche d'informations étant donné qu'ils aident les utilisateurs à découvrir des ressources qu'ils n'auraient pas trouvé par eux-mêmes. Enfin, les SR présentent depuis plusieurs années un domaine de recherche très prisé par les chercheurs et scientifiques dans la fouille de données notamment.

Les SR basés sur le filtrage collaboratif

Les méthodes basées sur le filtrage collaboratif ou *collaborative filtering* sont basées sur la collection et l'analyse d'informations sur le comportement des utilisateurs, leurs activités et leurs préférences pour ensuite prédire ce que les utilisateurs sont susceptibles d'aimer d'après leurs similarités avec d'autres utilisateurs. Un des avantages de cette méthode est que le système de recommandation n'est pas obligé de "comprendre" la ressource (ce dont il s'agit), ainsi, il est capable de recommander un grand nombre de ressources avec un moindre coût. Ce genre de méthode se base sur l'hypothèse que deux utilisateurs qui ont aimé les mêmes ressources par le passé vont aimer les mêmes ressources dans le futur et vont aussi aimer des ressources similaires à celles qu'ils ont aimé par le passé. Lors de la création d'un modèle à partir des données d'un utilisateur, on distingue deux types de données :
les données explicites comme :

- les tags/notes donnés par les utilisateurs aux ressources ; dans le cas des notes, une échelle (généralement de 0 à 5 ou de 0 à 10) est précisée ;
- les recherches opérées par les utilisateurs ;
- les préférences des utilisateurs (le système peut demander à ses utilisateurs de classer des ressources par ordre décroissant selon leur goût) ;
- les ressources partagées par les utilisateurs.

les données implicites comme :

- l'analyse du nombre de consultations par les utilisateurs sur les ressources ;
- l'obtention de la liste des ressources que les utilisateurs ont écouté ou regardé en ligne ;
- le nombre de clics des utilisateurs sur les ressources.

Un des exemples les plus connus de filtrage collaboratif est un algorithme popularisé par le système de recommandation de *Amazon.com* qui se base sur l'hypothèse *"Les utilisateurs qui achètent x achèteront aussi y"*. Le site de musique *last.fm* permet lui de recommander de la musique à ses utilisateurs en se basant sur celles écoutées par des utilisateurs similaires. Enfin, des réseaux sociaux célèbres comme Facebook, MySpace ou encore LinkedIn utilisent le filtrage collaboratif afin de recommander des amis, groupes ou pages à ses utilisateurs. Cependant, parmi les limites de ce type d'approche, on peut citer trois principaux problèmes auxquelles se heurtent les SR basés sur le filtrage collaboratif :

Démarrage à froid De tels SR exigent que les utilisateurs possèdent un historique et un comportement passé afin de générer des recommandations.

Scalabilité Généralement, dans ce genre de SR, le nombre d'utilisateurs et de ressources dépassent les millions. Ainsi, une grande quantité de mémoire est nécessaire afin de générer les recommandations.

Éparsité des données Le nombre de ressources est extrêmement plus grand que le nombre d'utilisateurs qui auront à partager uniquement une petite portion de l'ensemble total des ressources. Ainsi, même les ressources les plus populaires seront très peu partagées. Cela présente un inconvénient surtout pour les sites de e-commerce.

Les SR basés sur le contenu

Les méthodes de filtrage basées sur le contenu se basent sur une description des ressources ainsi que sur les préférences des utilisateurs. Ainsi, dans ce genre de SR, des mots-clés (ou tags) sont utilisés afin de décrire les ressources ; un profil est ensuite crée afin d'indiquer pour chaque utilisateur le type de ressources qu'il aime. En d'autres termes, de tels SR essaient de recommander des ressources qui sont similaires à ceux que l'utilisateur a aimé par le passé. Pour ce faire, ces SR se basent sur ces mots-clés qui décrivent les ressources (qu'on appelle aussi propriétés des ressources) et les comparent avec ceux que l'utilisateur a déjà partagé par le passé pour ensuite lui recommander les ressources les plus similaires à ceux qu'il a déjà tagué/noté. Ce genre d'approche est très utilisé dans la recherche d'information. Ce genre de méthodes utilisent donc un profil pour chaque ressource qui consiste en un ensemble d'attributs qui le caractérisent. Par exemple, un film est caractérisé par son titre, son genre, le nom de son réalisateur, le nombre d'oscars qu'il a reçu, etc. Ensuite, le SR compare le profil de chaque ressource avec les profils des ressources que l'utilisateur du système a déjà partagé.

Un des inconvénients de ce genre de méthodes est l'incapacité du système à s'adapter aux changements de type des données, *i.e.,* un SR qui construit un profil de ressources à partir des préférences des utilisateurs sur un certain type de contenu ne sera pas capable de l'adapter à d'autres types de contenus. Par exemple, un SR qui recommande des articles d'information à des utilisateurs en se basant sur les articles qu'ils ont déjà lu sera incapable de recommander à ces mêmes utilisateurs d'autres types de contenu comme des vidéos, des produits ou encore de la musique. Cela est dû au fait que les

caractéristiques des ressources changent en fonction de la nature de ces dernières.

Un des SR les plus populaires qui se basent sur ce genre de méthode est PANDORA RADIO qui recommande des morceaux de musique à ses utilisateurs en se basant sur les caractéristiques des chansons déjà écoutées par ces derniers (comme le titre, le genre ou encore la durée des chansons).

Les SR hybrides

Plusieurs études ont démontré que les approches hybrides combinant les méthodes de filtrage collaboratif et de filtrage basé sur le contenu peuvent être plus efficaces dans certains cas. Les approches hybrides peuvent être implémentées selon différentes méthodes : (i) une méthode de filtrage collaboratif puis une méthode de filtrage basé sur le contenu de manière séparée, puis en combinant les deux méthodes ; (ii) en ajoutant les fonctionnalités des méthodes de filtrage collaboratif à celles des méthodes de filtrage basé sur le contenu ; (iii) en ajoutant les fonctionnalités des méthodes de de filtrage basé sur le contenu à celles des méthodes filtrage collaboratif. Les méthodes hybrides peuvent être une solution à des problèmes rencontrés par les SR comme le démarrage à froid ou encore l'éparsité des données. Netflix est un exemple très connu de ce genre de méthodes. En effet, le système de recommandation de ce site propose des recommandations qui se basent à la fois sur les films regardés par des utilisateurs similaires (*i.e.*, filtrage collaboratif) ainsi que sur les films qui ont les mêmes caractéristiques que les films que l'utilisateur a partagé (*i.e.*, filtrage basé sur le contenu).

2.2.2 Métriques de qualité : Précision, Rappel et F1-Score

Afin d'évaluer un système de recommandation, deux métriques sont communément utilisées dans la recherche d'information classique : la Précision et le Rappel [Baeza-Yates et Ribeiro-Neto, 1999]. En effet, évaluer l'efficacité d'un algorithme de recommandation est loin d'être trivial. En premier lieu, parce que différents algorithmes peuvent être meilleurs ou moins bons en fonction du jeu de données sur lequel ils sont appliqués. D'autre part, les objectifs fixés par un système de recommandation peuvent être divers et variés. Un système de recommandation peut être mis en place pour estimer avec exactitude la note que donnerait un utilisateur à un élément, alors que d'autres auront comme objectif principal de ne pas proposer des recommandations

erronées. On peut donc légitimement se demander jusqu'à quel point ces différentes méthodes de recommandation sont réellement efficaces. Néanmoins, pour déterminer l'efficacité d'un système, l'indicateur le plus répandu dans la littérature est la précision (*cf.,* Equation 2.1) qui représente la qualité de la recommandation, c'est-à-dire à quel point les suggestions proposées sont conformes aux intérêts de l'utilisateur.

$$Précision = \frac{nombre\ de\ recommandations\ pertinentes}{nombre\ de\ recommandations} \tag{2.1}$$

La précision détermine donc la probabilité qu'un élément recommandé soit pertinent. Ainsi, la meilleure mesure de l'efficacité d'un algorithme de recommandation et de la pertinence des suggestions est donc d'évaluer la précision de la prédiction effectuée par le système en comparant les prédictions avec les choix qu'aurait fourni l'utilisateur dans le cas réel [Penet *et al.*, 2011].

Il est également possible de faire varier le nombre de recommandations proposées à l'utilisateur : on parle alors de requête top-k. Grâce à ce genre de requête, l'utilisateur peut spécifier le nombre k de réponses (recommandations) les plus pertinentes que le système doit lui retourner. Cela permet surtout d'éviter de submerger l'utilisateur par un grand nombre de réponses en lui retournant que le nombre de réponses les plus pertinentes qu'il souhaite [Dedzoe, 2011].

Le rappel, quant à lui, est le nombre de recommandations qui sont pertinentes divisé par le nombre total de recommandations pertinentes qui existent dans le jeu de données en question (*cf.,* Equation 2.2). Donc, le rappel met en lumière la portion de recommandations pertinentes qui a été retournée à l'utilisateur parmi l'ensemble total de recommandations pertinentes.

$$Rappel = \frac{Nombre\ de\ recommandations\ pertinentes}{Nombre\ total\ de\ recommandation\ pertinentes} \tag{2.2}$$

$$F1\text{-}Score = 2 \times \frac{Rappel \times Precision}{Rappel + Precision} \tag{2.3}$$

Tandis que la précision mesure la proportion de recommandations pertinentes et que le rappel mesure la proportion de recommandations pertinentes qui apparaissent dans les top recommandations, le *F1-Score* considère les deux mesures simultanément afin de calculer le score pondéré [Herlocker *et al.*, 2004]. Ainsi, le *F1-score* peut être interprétée comme une moyenne pondérée de la précision et du rappel, qui a une valeur

comprise entre 0 et 1, et qui indique l'utilité globale de la liste de recommandation (*cf.*, Equation 2.3).

Dans ce qui suit, nous discutons plusieurs propriétés afin d'évaluer les propriétés de notre système de recommandation. Ces propriétés sont définies comme la capacité d'un système de recommandation à suggérer à l'utilisateur des éléments pertinents mais pas forcément populaires.

2.2.3 Propriétés des systèmes de recommandations

Dans cette partie, nous examinons un ensemble de propriétés qui sont généralement prises en considération lorsque nous devons décider quelle approche de recommandation sélectionner. Selon les besoins de l'application souhaitée, le concepteur d'un système de recommandation doit décider sur quelles propriétés il doit miser entraînant ainsi un certain compromis. Par exemple, il peut décider que la précision du système diminue au profit d'autres propriétés plus importantes (*e.g.*, diversité). Ainsi, bien qu'elle soit une tâche cruciale, la recommandation d'utilisateurs, de tags et de ressources s'avère parfois insuffisante pour déployer un bon système de recommandation. Souvent, les utilisateurs peuvent être intéressés par plus qu'une bonne recommandation : la découverte de nouveaux éléments, la diversité des éléments, etc. Ainsi, nous devons identifier l'ensemble de propriétés qui influent sur la réussite d'un système de recommandation ([Ricci *et al.*, 2011]) :

Couverture

Comme la précision d'un système de recommandation, en particulier dans un cas de filtrage collaboratif, croît dans de nombreux cas avec la quantité de données, certains algorithmes peuvent fournir des recommandations de haute qualité (*i.e.*, avec une bonne précision), mais seulement pour une petite portion de données. La couverture peut désigner deux propriétés différentes que nous discutons ci-dessous :

1. Couverture de l'espace utilisateur : la couverture de l'espace utilisateur est définie comme étant la portion d'utilisateurs pour laquelle le système peut recommander des éléments. Les algorithmes capables de fournir des recommandations à la majorité des utilisateurs sont donc particulièrement appréciés. La couverture de l'espace utilisateur peut également être mesurée par la richesse du profil d'utilisateur re-

quis pour faire des recommandations. Par exemple, dans un cas d'un système de recommandation à filtrage collaboratif, cela peut être mesurée par le nombre de ressources qu'un utilisateur doit partager avant de recevoir des recommandations. Cette mesure peut très bien être mesurée dans une évaluation hors-ligne.

2. Couverture de l'Espace Ressource : le terme *couverture* désigne également la portion de ressources que le système peut recommander à ses utilisateurs. Cela est également appelé *catalogue de couverture*. La mesure la plus simple de la couverture est de calculer le pourcentage de ressources qui peuvent être recommandées par l'algorithme de recommandation à partir de l'ensemble de données en entrée.

Démarrage à froid

Un problème récurrent dans le processus de recommandation est celui du "démarrage à froid" ou cold start ([Ricci *et al.*, 2011]), *i.e.,* la performance du système vis-à-vis des nouveaux utilisateurs. Un utilisateur est considéré comme nouveau s'il n'a encore tagué aucune ressource. Le cold start peut être considéré comme un sous-problème de la couverture puisqu'il mesure la couverture du système sur un ensemble spécifique d'utilisateurs.

Sérendipité

La sérendipité mesure, avec quel degré de surprise, les recommandations apparaissent pour les utilisateurs. Un système de recommandation essaie de surprendre ses utilisateurs en leur recommandant des éléments qu'ils ne connaissent pas encore. En effet, certains utilisateurs peuvent également être intéressés par un système capable de répondre à leurs requêtes en leur proposant des ressources qu'ils n'ont pas l'habitude de consulter. Par exemple, si un utilisateur a partagé en masse plusieurs films où apparaît un certain acteur ; recommander à cet utilisateur le dernier film où joue cet acteur peut être considéré comme une recommandation nouvelle (puisque l'utilisateur peut ne pas encore connaître ce film) mais difficilement surprenante. Afin de mesurer ce degré de surprise, nous utilisons une métrique de distance d (*c.f.,* équation 4.2) qui mesure la distance entre l'élément recommandé et un ensemble d'éléments déjà partagés par l'utilisateur. La métrique de distance d est définie comme suit ([Ricci *et al.*, 2011]) :

$$d(b, B) = \frac{1 + C_B - C_B.w(b)}{1 + C_B} \tag{2.4}$$

où b est le livre recommandé, B l'ensemble de livres déjà lus par l'utilisateur ciblé, C_B le nombre maximal de livres écrits par un même auteur dans l'ensemble B et $C_{B.w(b)}$ le nombre de livres écrits par l'auteur de b dans l'ensemble B. À noter que la valeur de d se tient dans l'intervalle unitaire. Cette métrique est définie dans [Ricci *et al.*, 2011] pour le cas particulier où les ressources sont des livres mais peut très bien s'appliquer à d'autres types de données (films, musiques, etc.) en changeant par exemple l'auteur d'un livre par le réalisateur d'un film ou encore le chanteur d'une chanson. Par ailleurs, générer des recommandations aléatoires peut être surprenant mais peu pertinent, et dans ce cas, nous devons chercher un compromis entre sérendipité et précision afin de surprendre les utilisateurs tout en gardant des recommandations avec de bonnes précisions. Ainsi, nous devons combiner la métrique décrite ci-dessus avec la pertinence des ressources (*i.e.,* précision). En effet, la sérendipité est atteinte lorsque la ressource est à la fois surprenante et pertinente.

Adaptativité

Tester l'adaptativité d'un système revient à calculer son degré d'adaptation aux changements dans le profil d'un utilisateur ou aux changements dans ses préférences [Ricci *et al.*, 2011] [Mahmood et Ricci, 2007]. En effet, un utilisateur peut changer de catégorie d'âge ou de ville ou encore il peut changer ses préférences en matières de films, *i.e.*, partager des films d'action alors qu'il avait tendance à préférer des films de comédie. Ainsi, afin de mesurer l'adaptativité d'un système, nous mesurons la différence entre la liste de recommandations avant et après le(s) changement(s) opérés dans le profil d'un utilisateur (*i.e.*, l'ajout ou la modification d'une information dans son profil). Pour ce faire, nous utilisons la mesure *Gini Index* [Fleder et Hosanagar, 2007] dans le but de mesurer la variabilité des recommandations faites à un utilisateur lorsque le profil de ce dernier change. Cette métrique mesure la façon avec laquelle les ressources sont choisies de manière inégale par l'utilisateur. Si chaque ressource i est représentée par une proportion $p(i)$ dans le choix d'un utilisateur, la mesure *Gini Index* est décrite comme suit :

$$G = \frac{1}{n-1} \sum_{j=1}^{n} (2j - n - 1) p(i_j) \tag{2.5}$$

où $i_1, \ldots, i_n$ est la liste des ressources triées par ordre croissant selon leurs $p(i)$. La mesure est égale à 0 si toutes les ressources sont choisies de manière égale et vaut 1 si un même item est toujours choisi.

Diversité

Recommander un ensemble d'éléments qui sont similaires n'est pas aussi utile pour les utilisateurs, qui préfèrent la diversité, *i.e.,* des recommandations qui sont différentes et *distantes*. Par exemple, un utilisateur préférera une recommandation de cinq livres écrits par cinq auteurs différents à une recommandation de cinq livres écrits par un même auteur. Afin de mesurer la diversité de nos recommandations, nous utilisons la métrique de distance d (*cf.,* équation 4.2) qui mesure la distance entre l'élément recommandé et un ensemble d'éléments déjà partagés par l'utilisateur. Nous utilisons cette métrique de la manière suivante : nous calculons d'abord la distance entre chaque livre recommandé et le reste de la liste des livres recommandés et ensuite, nous calculons la moyenne de ces résultats afin d'obtenir le score de diversité. Cette métrique est définie dans [Ricci *et al.*, 2011] pour le cas particulier où les ressources sont des livres mais peut très bien s'appliquer à n'importe quel autre type de ressources (films, musiques, etc.).

Scalabilité

Un des objectifs d'un système de recommandation est le passage à l'échelle afin de permettre à ses utilisateurs de naviguer dans de grands jeux de données et de recevoir des recommandations sans un grand délai de temps de réponse. En effet, plus la quantité de données est importante, plus les algorithmes de recommandations peuvent être ralentis ou nécessitent des ressources supplémentaires (processeur, mémoire, etc.) afin de générer les recommandations. L'approche standard pour évaluer la scalabilité d'un système est d'évaluer la complexité de l'algorithme dédié en termes de temps d'exécution (ou temps de réponse) ou/et de mémoire requise.

2.3 État de l'art sur les systèmes de recommandations dans les *folksonomies*

Dans un souci d'améliorer les recommandations dans les *folksonomies*, plusieurs travaux ont été proposés dans la littérature. La plupart de ces approches représentent Nous pouvons partitionner les principaux travaux en quatre catégories :

2.3.1 Approches basées sur la popularité

Dans ([Jäschke *et al.*, 2007]), Hotho *et al.* ont proposé des recommandations de tags dans les *folksonomies* basées sur les tags les plus utilisés. Les auteurs ont présenté deux algorithmes pour la recommandation de tags dans les *folksonomies* : un algorithme basé sur le filtrage collaboratif et un algorithme adapté de PageRank qui permet de traiter directement les contextes triadiques afin de générer des recommandations. Les deux algorithmes sont évalués et comparés sur trois jeux de données du monde réel, *i.e.,* DEL.ICIO.US, LAST.FM et BIBSONOMY. Cette évaluation a permis de constater que l'approche de Hotho *et al.* permet d'améliorer la qualité des recommandations. Selon les auteurs, cette amélioration des résultats est dû au fait qu'elle exploite pleinement la structure triadique des *folksonomies*. De plus, malgré sa simplicité et son aspect non personnalisé des recommandations, l'approche proposée a atteint de bons résultats de précision et de rappel sur les petits jeux de données. Cependant, ces recommandations ne sont pas personnalisées étant donné que les mêmes tags, *i.e.,* les plus populaires, sont proposés à chaque utilisateur.

Lipczak a proposé dans ([Lipczak, 2008]) un système de recommandation de tags en trois étapes. À partir des tags annotés aux ressources, l'auteur ajoute des tags proposés par un lexique basé sur les co-occurrences de tags sur les mêmes ressources. Ensuite, le système filtre les tags déjà utilisés par l'utilisateur. Toutefois, malgré cette étape de filtrage, la recommandation n'est pas personnalisée étant donné qu'elle cherche des tags co-occurrant sur d'autres ressources. L'approche revient ensuite à enlever les tags précédemment annotés par l'utilisateur de ceux qui sont suggérés.

Dans [De Meo *et al.*, 2010a], De Meo *et al.* ont construit et maintenu un profil pour chaque utilisateur. Ainsi, lorsqu'un utilisateur soumet une requête à la *folksonomie* afin de retrouver un ensemble de ressources, les auteurs proposent de retrouver d'avantage

de tags *autoritaires* afin d'enrichir sa requête et les recommandent ensuite à l'utilisateur. Les tags *autoritaires* sont ceux qui ont été les plus utilisés par d'autres utilisateurs sur les mêmes ressources. Selon les auteurs, la stratégie de recourir aux tags autoritaires pour enrichir les requêtes des utilisateurs -à la demande de ces derniers- permettent de découvrir des ressources nouvelles et intéressantes pour les utilisateurs. De plus, cela permet aux systèmes de recommandations de produire des recommandations plus raffinées. Par ailleurs, les tags autoritaires sont également directement ajoutés au profil de l'utilisateur rendant ce dernier plus riche et contenant des tags représentant les intérêts de l'utilisateur. Ce dernier n'était pas spécialement en mesure de spécifier explicitement ces tags par le passé. Cela a pour conséquence une meilleure précision lors du calcul de similarité entre deux utilisateurs permettant ainsi une meilleure précision lors des recommandations. Un autre avantage de l'approche consiste en la (quasi) non intervention de l'utilisateur lors de la génération des recommandations qui est une phase presque automatique. Cependant, dans cette approche, l'utilisateur doit lui-même ajouter les tags aux ressources ou/et sélectionner les tags les plus pertinents pour les ressources ; en d'autres termes, l'utilisateur doit fournir un feedback explicite. De plus, l'évaluation des tags autoritaires implique une analyse hors-ligne des données.

2.3.2 Approches basées sur les mesures de similarité

Dans ([Diederich et Iofciu, 2006a]), Diederich et Iofciu utilisent la *"personomie"* d'un utilisateur, *i.e.,* les tags qui lui sont relatifs, afin de lui recommander des utilisateurs ayant partagé des tags et ressources similaires. En premier lieu, il s'agit de parcourir les personomies afin de construire le profil pour l'utilisateur concerné. La Figure 2.1 montre un exemple d'un profil pour l'utilisateur *Dana Scully* où chaque tag de sa personomie est caractérisée par son occurence dans sa personomie et par le nombre total de son apparition dans l'ensemble des ressources de la *folksonomie*.

Ensuite, à partir de ce profil, les auteurs sont capables de recommander des utilisateurs (dits *collaborateurs*) en utilisant deux mesures de similarité entre utilisateurs introduites dans [Diederich et Iofciu, 2006b]. Les auteurs procèdent à l'extraction des utilisateurs ayant le profil le plus proche, *i.e.*, les *"voisins"* ayant annoté leurs ressources avec les mêmes tags. Les Tableaux 2.2 et 2.3 montrent les utilisateurs *"recommandables"* à l'utilisateur *Dana Scully* en se basant sur les tags ou les ressources que contiennent leurs personomies, *i.e.*, les utilisateurs ayant des profils similaires à ce dernier.

Tag	Occurrences	Fréquence Globale
XML	1	554
UML	1	302
Web Services	1	193
Ontology	1	158
Adaptation	1	102
Semantic Web	5	190
Peer-to-peer	4	123
Personalization	4	92
Standards	1	61
Query languages	1	63
Hypermedia	1	93
Generalization	1	25
Web search	1	49
E-learning	1	59
Netwrok management	1	49
Diagnosis	1	49
Ranking	1	31
Pagerank	1	38
Web engineering	1	35
Adaptive hypermedia	2	30
Meta-modeling	1	9
XML scheme	1	23
XMI	1	9
Asynchronous collaboration	1	8
Synchronous collaboration	1	5
Adaptive Web	2	5

Tableau 2.1 — Un exemple de profil pour l'utilisateur *Dana Scully*.

.

Dans ([Landia et Anand, 2009]), les auteurs ont proposé une nouvelle approche combinant la similarité à la fois entre ressources et entre utilisateurs afin de recommander des tags personnalisés. En effet, deux utilisateurs sont considérés comme similaires s'ils ont assigné les mêmes tags aux mêmes ressources. Un score est ensuite calculé pour chaque tag potentiel en prenant en compte une combinaison entre la similarité de la

Les collaborateurs recommandés	Score
Jack Bagwell	0,390822
Epiphany Doubtfire	0,311705
Theodore Bagwell	0,299058
Allison Dubois	0,253242
Idhal Genious	0,214939
Melvin Frohike	0,173752
Eya Turki	0,173752

Tableau 2.2 — Un exemple de recommandations de collaborateurs pour l'utilisateur *Dana Scully* s'appuyant sur les tags.

Les collaborateurs recommandés	Score
Mulder Ross	0,411228
Alex Krycek	0,274152
Monica Geller	0,137076
Anna-Lucia Rodriguez	0,137076
Nannie Fran	0,137076

Tableau 2.3 — Un exemple de recommandations de collaborateurs pour l'utilisateur *Dana Scully* s'appuyant sur les ressources.

ressource à laquelle le tag est assigné et la similarité de l'utilisateur qui partage la ressouce. Si le score du tag est en dessous d'un certain seuil, cela veut dire que le tag n'est pas assez concordé à la ressource pour l'utilisateur ciblé. De plus, si le nombre requis de tags recommandés est en dessous d'un certain nombre prédéfini t, de nouveaux tags sont alors générés à partir du vocabulaire du corpus de la ressource ciblée. L'approche a été évaluée uniquement sur un petit jeu de données (BIBSONOMY) et les premiers résultats indiquent que la personnalisation de la recommandation tags a amélioré la précision du système. Toutefois, dans les cas réels de *folksonomie*, le système proposé par Landia *et al.* n'est pas recommandé étant donné qu'il est rare de trouver des situations dans des *folksonomies* où les tags utilisés par deux utilisateurs sur les mêmes ressources sont identiques. Dans ce cas, les nouveaux tags qui seront générés risquent de ne pas intéresser suffisamment l'utilisateur et de réduire la précision du système.

2.3.3 Approches hybrides combinant l'historique de tagging et les mesures de similarités

Dans ([Hu *et al.*, 2011]), les auteurs se basent à la fois sur l'historique de tagging (tags et ressources) des utilisateurs et sur leurs contacts sociaux afin de générer des recommandations personnalisés de tags. Les auteurs se sont basés sur le système de partage de photos Flickr. Les contacts sociaux d'un utilisateur peuvent être utiles afin de générer des recommandations de tags plus personnalisés aux utilisateurs lors de l'annotation des photos. En plus de l'historique de tagging et des contacts sociaux, les auteurs combinent ces deux données avec une mesure de co-occurence de tags dans le but d'améliorer encore plus la performance de leur système de recommandation. Les auteurs ont précisé que leur système de recommandation, basé uniquement sur le réseau social Flickr, peut être étendu à d'autres contextes avec d'autres types de données. Cependant, cela aura pour effet l'adaptation de l'approche ainsi que de la mesure de similarité aux nouveaux contextes, ce qui n'est pas une tâche aisée. Par ailleurs, si une des limites de cette approche est qu'elle requiert qu'un utilisateur doit posséder des contacts sociaux avant d'avoir des recommandations de tags, un avantage est qu'elle permet de proposer une solution au problème de cold start des ressources. En effet, lorsqu'une nouvelle photo est partagée sur Flickr sans aucun tag associé, le système de recommandation de Hu *et al.* est capable recommander des tags à un utilisateur uniquement en se basant sur ses contacts sociaux.

Dans [Basile *et al.*, 2007], Basile *et al.* ont proposé un système intelligent de recommandation de tags qui est capable d'apprendre aussi bien à partir de l'historique de tagging des utilisateurs qu'à partir du contenu des ressources à annoter. Le système est également capable de recommander une liste de nouveaux tags utiles utilisés par d'autres utilisateurs sur les mêmes ressources.

2.3.4 Approches basées sur le profil des utilisateurs

Dans [Liang *et al.*, 2010], Liang *et al.* proposent un système de recommandation personnalisée qui s'appuie aussi bien sur les tags utilisés par les utilisateurs que sur leur profil personnel. Les auteurs ont proposé une nouvelle mesure de similarité qui calcule la similarité entre les utilisateurs en se basant sur les tags qu'ils ont déjà partagé.

Dans [Bellogín *et al.*, 2013], Bellogìn *et al.* ont étendu la pratique classique d'éva-

luation de pertinence (*i.e.,* rappel et précision) à d'autres métriques afin de mesurer la qualité des recommandations (*e.g.,* couverture, diversité, nouveauté, etc.). De plus, selon les auteurs, en combinant diverses stratégies de recommandations (basées sur le contenu, basées sur le filtrage collaboratif et basées sur le social), cela aura pour effet des recommandations plus pertinentes en termes des métriques de performances. La première étape consiste à diviser le jeu de données en entrée en base d'apprentissage et base de test. Ensuite, lors de la phase d'apprentissage, les auteurs construisent deux profils : un pour les utilisateurs basé sur les tags les plus utilisés, et un autre pour les ressources, toujours basé sur les tags les plus utilisés sur ces derniers. Le système de recommandation génére ensuite, dans une deuxième étape, l'ensemble de tags recommandés. La troisième étape consiste à calculer à partir de la base de test les tags relatifs au couple utilisateur-ressource. Enfin, une comparaison est faite entre les résultats des deux précédentes étapes pour générer l'ensemble de tags recommandés.

Dans [Kim *et al.*, 2011], Kim *et al.* ont proposé une procédure de recommandation pour les communautés de livres en ligne. Cette procédure consiste en deux étapes : tout d'abord, elle cherche les voisins utilisant les préférences des utilisateurs pour les livres ainsi que leurs profil personnel (informations démographiques), ensuite, elle leur génére des recommandations personnalisées. La deuxième étape consiste à supprimer les livres non pertinents de la liste de recommandations en se basant sur les mots-clés préférés par chaque utilisateur (une liste entrée au préalable par chaque utilisateur). Une étape préliminaire permet de construire un modèle pour la communauté cible ainsi qu'un modèle individuel pour chacun de ses membres. Ensuite, la première étape va consister à trouver les voisins de cette communauté en se basant sur le profil des utilisateurs de la communauté pour ensuite sélectionner n livres. La deuxième étape permet, selon plusieurs métriques, d'évaluer la pertinence de cette liste de livres (résultat de la première étape) envers la communauté cible. Dans cette étape, seront par exemple supprimés les livres jugés inintéressants pour certains membres, en se basant sur un vecteur contenant des mots-clés préférés pour chaque membre de la communauté. Le but final est de générer une liste définitive de livres à tous les membres de la communauté.

Dans [Qumsiyeh et Ng, 2012], Qumsiyeh et Ng ont proposé un système personnalisé de recommandation qui se base sur diverses informations sur les utilisateurs comme les notes données aux ressources ou encore les reviews donnés à différentes ressources. Les auteurs visent à répondre aux besoins et intérêts de chaque utilisateur pour faire les recommandations. La Figure 2.1 illustre un exemple de prédiction de genre de films

dans le jeu de données MovieLens. Le système de recommandation tente de prédire quel tag (ici, genre de film) sera affecté par un utilisateur u au nouveau film qu'il va partager. Pour ce faire, le système se base sur les précédents films partagés (et notés) par u ainsi que sur deux mesures de similarité : une mesure WS (*i.e.,* $Word\_Sim$) qui calcule la similarité entre deux genres donnés et une mesure de score des genres (*i.e.,* GS qui signifie *Genre Score*). Ce score, qui se base sur l'ensemble de ressources déjà partagés (et notés) par un utilisateur u, calcule la probabilité d'affectation d'un genre donné (de film, de livre, etc.) à une ressource non encore notée par l'utilisateur u. Dans l'exemple illustré par la Figure 2.1, le système calcule d'abord la similarité entre les genres (deux à deux). Nous pouvons notamment voir la similarité quasi-identique entre les genres *Horror* et *Thriller* étant donné que ces deux genres sont assez proches pour décrire un film (tout comme la similarité entre *Action* et *Adventure*). À l'inverse, les genres *Horror* et *Adventure* sont assez distants. Ensuite, se basant sur ces mesures, le système calcule la probabilité de chaque genre pour le nouveau film que va noter l'utilisateur. Par exemple $GSG(1,H)$ permet de calculer la probabilité que l'utilisateur va affecter le genre *Horror* au nouveau film avec la note 1. La probabilité la plus grande calculée par le système concerne le genre *Action* avec la note 5 (juste devant le genre *Horror* avec la note 1). En effet, nous pouvons voir que d'après le tableau de l'exemple, que l'utilisateur cible a eu tendance à partager les précédents films avec les genre *Horror* tout en leur affectant la note 1 ou bien à affecter le genre *Action* à ces films avec la note maximale 5. Cette dernière combinaison apparaît comme la plus probable.

Rating	Horror (H)	Thriller (T)	Action (Ac)	Adventure (Ad)
1	50	40	6	4
2	20	9	8	4
3	4	7	7	7
4	3	8	15	5
5	1	3	25	18

$WS(H, T) = 0.93$
$WS(H, Ac) = 0.35$
$WS(H, Ad) = 0.15$
$WS(T, Ac) = 0.67$
$WS(T, Ad) = 0.33$
$WS(Ac, Ad) = 0.91$

$$GSG(1,H) = \frac{50}{100} + \left(\frac{40}{100} \times WS(H,T) + \frac{6}{100} \times WS(H,Ac) + \frac{4}{100} \times WS(H,Ad)\right) = \mathbf{0.90}$$

$$GSG(5,H) = \frac{1}{47} + \left(\frac{3}{47} \times WS(H,T) + \frac{25}{47} \times WS(H,Ac) + \frac{18}{47} \times WS(H,Ad)\right) = 0.32$$

$$GSG(1,Ac) = \frac{6}{100} + \left(\frac{50}{100} \times WS(Ac,H) + \frac{40}{100} \times WS(Ac,T) + \frac{4}{100} \times WS(Ac,Ad)\right) = 0.54$$

$$GSG(5,Ac) = \frac{25}{47} + \left(\frac{1}{47} \times WS(Ac,H) + \frac{3}{47} \times WS(Ac,T) + \frac{18}{47} \times WS(Ac,Ad)\right) = \mathbf{0.93}$$

Figure 2.1 — Illustration de l'approche de Qumsiyeh *et al.*

2.3.5 Tableau Comparatif des différentes approches

Afin de mettre en lumière les plus-values de notre approche par rapport à ses prédécesseurs, nous surlignons dans le Tableau 4.14 les différents critères qu'un système de recommandation doit vérifier [Ricci *et al.*, 2011] (voir, Chapitre 1) pour les différentes approches de la littérature [Diederich et Iofciu, 2006a] [Basile *et al.*, 2007] [Jäschke *et al.*, 2007] [Landia et Anand, 2009] [De Meo *et al.*, 2010b] [Hu *et al.*, 2011] [Kim *et al.*, 2011] [Qumsiyeh et Ng, 2012] [Bellogín *et al.*, 2013]. La comparaison est établie selon les propriétés suivantes :

1. **Multi-Mode** : indique si l'approche propose une recommandation multi-mode, *i.e.,* une recommandation simultanée d'utilisateurs, de tags et de ressources.
2. **Couverture** : indique si l'approche permet une couverture de l'espace ressource et une couverture de l'espace utilisateur.
3. **Cold Start** : indique si l'approche traite le problème du cold start, *i.e.,* la recommandation pour les nouveaux utilisateurs.
4. **Adaptativité** : indique si l'approche s'adapte aux évolutions du profil de l'utilisateur ou/et du contenu de la *folksonomie*.
5. **Diversité** : indique si l'approche offre des recommandations diverses aux utilisateurs.
6. **Sérendipité** : indique si l'approche surprend ses utilisateurs avec des recommandations auxquelles ils ne s'attendaient pas.
7. **Scalabilité** : indique si l'approche arrive à réussir le passage à l'échelle.

Le point d'interrogation (" ?") dénote qu'une information est manquante dans l'approche et est difficile à vérifier. Nous pouvons remarquer, par exemple, qu'aucune des approches proposées dans la littérature n'offre une recommandations multi-mode (d'utilisateurs, de tags et de ressources en même temps). Cette propriété est très demandée et appréciée par les utilisateurs. Comme il sera démontré dans le quatrième chapitre, notre approche diffère de ses prédécesseurs en tenant en compte des nouveaux utilisateurs (les critères *couverture* et *cold start*) en leur fournissant des recommandations sans qu'ils n'aient déjà tagué par le passé. En effet, presque aucune des approches ne propose une solution à ce problème qui est assez difficile compte tenu de l'absence de comportement passé d'un nouvel utilisateur. De plus, concernant la propriété de couverture, si les ressources peuvent être couvertes dans certaines approches comme

	M-Mode	Couv.	C-Start	Adaptativité	Diversité	Sérendipité	Scalabilité
[Diederich et Iofciu, 2006a]	non	no	non	non	oui	?	non
[Basile *et al.*, 2007]	non	non	non	non	?	?	?
[Jäschke *et al.*, 2007]	non	non	non	non	oui	?	?
[Landia et Anand, 2009]	non	non	non	non	non	?	?
[De Meo *et al.*, 2010b]	non	non	non	non	?	?	?
[Hu *et al.*, 2011]	non	non	oui	non	oui	?	?
[Kim *et al.*, 2011]	non	non	non	non	oui	?	?
[Qumsiyeh et Ng, 2012]	non	non	non	non	oui	?	?
[Bellogín *et al.*, 2013]	non	oui	non	non	oui	?	?

Tableau 2.4 — Comparaison entre les travaux de la littérature par rapport aux propriétés des systèmes de recommandation.

[Jäschke *et al.*, 2007] qui explore toute la *folksonomie*, la couverture de l'espace utilisateur est impossible à atteindre étant donné le problème précédent, *i.e.,* le problème du cold start. Par ailleurs, nous pouvons remarquer que la plupart des approches satisfont le critère de *diversité* étant donné leurs forts taux de précision pour des recommandations comportant plusieurs ressources ou tags. La diversité est une propriété triviale étant donné qu'elle ne prend pas en compte la valeur de précision, *i.e.,* une approche peut proposer des recommandations qui soient diverses sans qu'ils soient pour autant pertinents.

Par contre, les deux propriétés d'adaptativité et de sérendipité sont difficiles à vérifier. En effet, les approches proposées dans la littérature s'appuient sur un ensemble statique de données et ne proposent pas de solution lorsque ces données changent ou lorsque les données d'un utilisateur, *i.e.,* son profil connaît quelques variations. Par ailleurs, la propriété de sérendipité n'est vérifiable que si l'approche en question propose un moyen de la calculer. Cependant, aucune des approches de la littérature n'a proposé une méthode de calcul de la sérendipité des recommandations. Enfin, la *scalabilité* (le passage à l'échelle) est également difficile à vérifier étant donné que leurs auteurs n'ont pas donné suffisamment d'informations sur leurs approches, *i.e.,* la majorité des approches n'a pas calculé le temps de réponse des recommandations ni la mémoire consommée lorsque le nombre de données devient très grand.

2.3.6 Critiques et contributions

Même si certains travaux, *i.e.,* ceux qui se sont basés sur le profil des utilisateurs, s'appuient sur des informations personnelles de ses utilisateurs afin de proposer des recommandations personnalisées, le principal souci de la majorité des approches de la littérature est qu'elles traitent des données gigantesques. En effet, les *folksonomies* contiennent des milliers de données, ce qui a pour effet d'altérer la qualité des recommandations dans certaines situations, *e.g.*, des ressources redondantes, des utilisateurs inactifs ou encore des tags impopulaires. De plus, cela risque de poser un problème lors du passage à l'échelle étant donné la quantité importante de données générées par les *folksonomies*. Comme nous l'avons discuté dans le point précédent, le cold start demeure également un problème majeur pour les systèmes de recommandations qui s'appuient, pour la plupart, sur les données déjà existantes dans les *folksonomies* excluant ainsi la possibilité de générer des recommandations à un nouvel utilisateur qui ne possède pas encore d'historique de tagging. En regardant de plus près le Tableau 4.14, nous pouvons également voir que plusieurs propriétés comme l'adaptativité ou la sérendipité sont ignorées. Par ailleurs, la plupart des travaux proposés sont limités à l'information <utilisateur, tag, ressource>, ce qui a comme inconvénient des recommandations quasi-similaires pour l'ensemble des utilisateurs.

Ainsi, dans ce mémoire, nous allons étendre ce triplet par l'information inclue dans la quatrième dimension dans le but de répondre à un maximum de propriétés. Dans l'approche que nous proposerons (*cf.,* Chapitre 4), nous insistons sur le nécessaire recours à des informations supplémentaires et à les combiner à l'historique de tagging afin d'améliorer les recommandations. Toutes ces informations seront représentées dans des structures quadratiques représentées par les quadri-concepts. Ainsi, dans ces concepts, nous nous focalisons non seulement sur les tags/ressources les plus utilisés, mais également sur ceux qui ont été utilisés en combinaison, obtenant ainsi un résultat plus spécifique. De plus, ces quadri-concepts, qui sont une représentation condensée et réduite sans perte d'information d'une *folksonomie*, permettent de traiter un ensemble réduit de données ce qui aura pour effet d'améliorer le temps de réponse des recommandations. De plus, puisque ces concepts se focalisent sur les utilisateurs, tags et ressources les plus utilisés en combinaison, nous pouvons éliminer certains problèmes de redondance des données ou encore d'utilisateurs inactifs et de tags trop peu utilisés qui n'apparaîtront pas dans les quadri-concepts. Ainsi, avec un ensemble plus réduit

de données, le passage à l'échelle devient une possibilité envisageable puisque la mémoire consommée sera moins grande et le temps de réponse sera meilleur. Concernant le problème de cold start, le recours à des informations supplémentaires et personnelles sur les utilisateurs va permettre de générer des recommandations aux nouveaux utilisateurs en se basant uniquement sur ces données. En effet, notre algorithme va proposer des recommandations à ses nouveaux utilisateurs en cherchant les tags et ressources déjà partagés par des utilisateurs qui ont en commun un certain nombre d'informations personnelles avec ces nouveaux utilisateurs. Enfin, nous proposons de mesurer à l'aide de métriques proposées dans la littérature l'adaptativité, la sérendipité ou encore la diversité de nos recommandations. De plus, une étude de cas menée sur un groupe d'utilisateurs mettra en lumière les avantages et lacunes de notre système personnalisé de recommandations.

2.4 Conclusion

Durant ce chapitre, nous avons d'abord présenté la définition d'un système de recommandation ainsi que les différentes métriques qui jaugent sa qualité : la précision, le rappel ainsi qu'une liste non exhaustive de propriétés. Ensuite, nous avons survolé les principaux travaux de la littérature qui proposent des systèmes de recommandations dans les *folksonomies* ; ces approches sont réparties en quatre catégories. Une étude critique de ces travaux a révélé plusieurs lacunes et la nécessité d'avoir recours à un nouveau système de recommandation personnalisée qui soit capable de vérifier un certain ensemble de propriétés comme la couverture, le cold start ou encore la scalabilité. Dans le prochain chapitre, nous commençons par notre première contribution qui consiste en la propostion d'un nouvel algorithme d'extraction de quadri-concepts à partir des *folksonomies*. Cet algorithme se base sur l'adaptation d'un certain nombre de notions de l'analyse formelle des concepts au contexte quadratique. La comparaison entre notre algorithme d'extraction des quadri-concepts et l'algorithme existant sera menée sur des jeux de données du monde réel et mettra en lumière la différence en temps d'exécution et en mémoire consommée.

CHAPITRE 3

QuadriCons, un algorithme pour l'extraction des quadri-concepts fréquents

3.1 Introduction

Durant le premier chapitre, nous avons étudié le principal algorithme qui a été proposé dans la littérature, *i.e.,* Data Peeler, pour résoudre le problème d'extraction des quadri-concepts dans les *folksonomies*. Cette étude a mis en lumière les principales lacunes de cet algorithme, assez gourmand en mémoire et dont la stratégie d'exploration des candidats est inefficace dans les grands jeux de données. Cela nous a motivé pour proposer un nouvel algorithme qui se consacre à l'extraction des quadri-concepts à partir des *folksonomies* et qui se base sur un opérateur de fermeture spécialement dédié à ce type de contexte et qui partitionnera l'espace de recherche en classes d'équivalences. La principale originalité de notre algorithme, qui est aussi une extension de l'algorithme TriCons (*cf.,* Chapitre 1), est qu'il commence par détecter les générateurs minimaux permettant un balayage intelligent de l'espace de recherche avant de procéder au calcul des quadri-concepts, ce qui réduit considérablement le nombre de candidats. Ensuite, nous étudions la validité de notre algorithme, *i.e.,* la correction, complétude et terminaison. Enfin, une étude expérimentale sur des jeux de données du monde réel mettra l'accent sur les temps d'exécution et la mémoire consommée de notre algorithme par rapport à ceux de Data Peeler.

Motivation : la quatrième dimension

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 ternaires (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, ...) afin de personnaliser les recommandations, ou le temps si on veut étudier la dynamique temporelle et détecter les tendances dans les *folksonomies*. Dans ce papier, nous traitons la quatrième dimension de manière indifférente pour l'aspect méthodologique, mais afin d'extraire des résultats à partir de jeux de données du monde réel, nous focaliserons plus tard soit sur l'aspect profil soit sur l'aspect temps. Par suite, l'algorithme QUADRICONS, que nous introduisons pour cette tâche spécifique, cherche à extraire un ensemble de quadruplets, dont chaque quadruplet (U, T, R, V) consiste en un ensemble U d'utilisateurs, un ensemble T de tags, un ensemble R de ressources et un ensemble V de variables. Ces quadruplets, appelés *quadri-concepts*, vérifie la propriété suivante : chaque utilisateur de U avec une variable de V a tagué 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**.

3.2 Nouvelles notions mathématiques de l'AFC

Dans cette partie, nous introduisons les principales notions qui seront utilisées par notre algorithme et qui seront au cœur de notre système personnalisé de recommandation (*cf.,* Chapitre 4). Nous commençons par une adaptation de la notion de *folksonomie* [Jäschke *et al.*, 2008] au contexte quadratique [**?**].

Définition 8 *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 de jeux de données du monde réel, nous focaliserons, plus tard, soit sur l'aspect profil, soit sur l'aspect temps.*

Exemple 6 *Le Tableau 3.1 montre un exemple d'une v-folksonomie* $\mathcal{F}_v$ *avec* $\mathcal{U} = \{u_1, \ldots, u_4\}$, $\mathcal{T} = \{t_1, \ldots, t_4\}$, $\mathcal{R} = \{r_1, r_2, r_3\}$ *et* $\mathcal{V} = \{v_1, v_2\}$. *Chaque croix désigne une opération de tagging faite par un utilisateur de* $\mathcal{U}$, *avec une variable de* $\mathcal{V}$, *utilisant un tag de* $\mathcal{T}$ *sur une ressource de* $\mathcal{R}$. *Par exemple, l'utilisateur* u_1 *qui possède les informations de profils* v_1 *(étudiant) et* v_2 *(27 ans) a tagué toutes les ressources avec les tags* t_2, t_3 *et* t_4.

$\mathcal{F}_v$	$\mathcal{R}$	r_1				r_2				r_3			
$\mathcal{V}$	$\mathcal{U}/\mathcal{T}$	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		×	×	×		×	×	×		×	×	×
v_1	u_2		×	×	×	×	×	×	×	×	×	×	×
	u_3		×	×	×	×	×	×	×	×	×	×	×
	u_4		×	×		×			×	×			×
	u_1		×	×	×		×	×	×		×	×	×
v_2	u_2		×	×	×	×			×	×	×	×	×
	u_3												
	u_4												

Tableau 3.1 — un exemple d'une *v-folksonomie*

Un quadri-set est l'extension d'un tri-set ([Trabelsi *et al.*, 2012]) à notre cas quadratique. La définition suivante introduit cette notion.

Définition 9 *(*QUADRI-SET (FRÉQUENT)*) Soit* $\mathbb{F}_v = (\mathcal{U}, \mathcal{T}, \mathcal{R}, \mathcal{V}, \mathrm{Y})$ *une v-folksonomie. Un quadri-set de* $\mathbb{F}_v$ *est un quadruplet* (A, B, C, E) *avec* $A \subseteq \mathcal{U}$, $B \subseteq \mathcal{T}$, $C \subseteq \mathcal{R}$ *et* $E \subseteq \mathcal{V}$ *tel que* $A \times B \times C \times E \subseteq \mathrm{Y}$. *Un quadri-set* (A, B, C, E) *de* $\mathcal{F}$ *est dit fréquent si* $| A | \geq minsupp_u$, $| B | \geq minsupp_t$, $| C | \geq minsupp_r$ *et* $| E | \geq minsupp_v$, *où* $minsupp_u$, $minsupp_t$, $minsupp_r$ *et* $minsupp_v$ *sont des seuils minimaux de support définis par l'utilisateur.*

Exemple 7 *Considérons la v-folksonomie $\mathbb{F}_v$ du Tableau 3.1. $qs_1 = (\{u_1, u_2\}, \{t_2, t_3\}, \{r_1, r_3\}, \{v_1\})$ et $qs_2 = (\{u_1, u_2\}, \{t_2, t_3, t_4\}, \{r_1, r_3\}, \{v_1, v_2\})$ sont deux quadri-sets de la v-folksonomie $\mathbb{F}_v$. Le premier quadri-set nous renseigne que les utilisateurs u_1 et u_2, qui ont en commun la variable v_1, ont partagé les ressources r_1 et r_3 avec les tags t_2 et t_3. Quant au deuxième quadri-set, l'information véhiculée est que les mêmes utilisateurs u_1 et u_2, qui ont en commun les variables v_1 et v_2, ont affecté les tags t_2, t_3 et t_4 aux ressources r_1 et r_3. Nous pouvons remarquer qu'un même groupe d'utilisateurs peut partager deux informations différentes, i.e., un ensemble différent de tags et de ressources.*

Étant donné que l'ensemble des quadri-sets fréquents peut-être 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 10 *(*Concept quadratique (fréquent)*) Un concept quadratique (ou quadri-concept) d'une v-folksonomie $\mathcal{F}_v = (\mathcal{U}, \mathcal{T}, \mathcal{R}, \mathcal{V}, \mathrm{Y})$ est un quadruplet (U, T, R, V) avec $U \subseteq \mathcal{U}$, $T \subseteq \mathcal{T}$, $R \subseteq \mathcal{R}$ et $V \subseteq \mathcal{V}$ avec $U \times T \times R \times V \subseteq \mathrm{Y}$ tel que le quadruplet (U, T, R, V) est maximal, i.e., aucun de ces ensembles ne peut être augmenté sans diminuer un des trois autres ensembles. Pour un quadri-concept $QC = (U, T, R, V)$, U, R, T et V sont, respectivement, appelés **Extent**, **Intent**, **Modus** et **Variable**. Afin d'extraire les concepts les plus intéressants, nous définissons pour chaque dimension un seuil minimal de support,* i.e., *$minsupp_u$, $minsupp_t$, $minsupp_r$ et $minsupp_v$. Il en résulte des quadri-concepts qui sont **fréquents**.*

Exemple 8 *Considérons la v-folksonomie $\mathbb{F}_v$ du Tableau 3.1. Soient $qc_1 = (\{u_1, u_2\}, \{t_2, t_3\}, \{r_1, r_3\}, \{v_1\})$ et $qc_2 = (\{u_1, u_2\}, \{t_2, t_3, t_4\}, \{r_1, r_3\}, \{v_1, v_2\})$ deux quadri-sets. qc_1 n'est pas considéré comme un quadri-concept de $\mathbb{F}_v$ étant donné qu'il n'est pas maximal. En effet, le quadri-set qc_2, qui maximise les ensembles T et V de qc_1, est un quadri-concept de $\mathbb{F}_v$. De même, le quadruplet $(\{u_1, u_2\}, \{t_2, t_3, t_4\}, \{r_1, r_2, r_3\}, v_1)$ n'est pas un quadri-concept étant donné que son extent, i.e., la partie utilisateurs, n'est pas maximale. En effet, l'utilisateur u_3 tout comme u_1 et u_2 (qui ont tous la variable v_1 en commun) a également partagé les ressources r_1, r_2 et r_3 via les tags t_2, t_3 et t_4. Donc, si on maximise cet extent, nous aurons le quadruplet $(\{u_1, u_2, u_3\}, \{t_2, t_3, t_4\}, \{r_1, r_2, r_3\}, v_1)$ qui est un quadri-concept, démontrant que les utilisateurs u_1, u_2 et u_3 (avec v_1 en commun) ont tagué les trois ressources r_1, r_2 et r_3 avec les tags t_2, t_3 et t_4.*

Les *v-folksonomies* ont quatre dimensions. 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 $A_1 \times B_1 \times C_1 \times E_1 \subseteq Y$ n'est pas *u-fréquent* (par rapport à la dimension $\mathcal{U}$), alors tous les (A_1, 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-fréquents*. Il en est de même pour les trois autres dimensions. Dans [Lehmann et Wille, 1995], 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 [Biedermann, 1997] induite par un contexte triadique. Dans ce qui suit, nous adaptons ce lemme pour notre cas avec des contextes à quatre dimensions.

Lemme 1 *(Voir aussi [Voutsadakis, 2002], 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 $A_i \times B_i \times C_i \times E_i \subseteq Y$, 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 $A_2 \subseteq A_1$.*

Avant d'introduire notre opérateur de fermeture pour une *v-folksonomie*, nous définissons un opérateur de fermeture pour un contexte n-aire. Dans [Voutsadakis, 2006], Voutsadakis a défini n opérateurs de dérivation pour un contexte n-aire. Chacun des i-opérateur de dérivation permet de calculer la fermeture relative à la dimension i pour un n-set donné $(1 \leq i \leq n)$. Dans ce qui suit, nous introduisons un nouvel opérateur de fermeture h dont le but est le calcul de la fermeture d'un n-set donné. Un n-set est la généralisation d'un quadri-set au cas n-aire et possède n parties relatives aux n dimensions. Contrairement à [Voutsadakis, 2006], nous utilisons un seul opérateur de fermeture qui calcule toutes les parties du concept n-adique résultant.

Définition 11 (Opérateur de fermeture d'un contexte n-aire) *Soit $S = (S_1, S_2, \ldots, S_n)$ un n-set d'un contexte n-aire $\mathbb{K}^n$ à n dimensions, i.e., $(\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_n)$. Un mapping h est défini comme suit :*

$$h(S) = h(S_1, S_2, \ldots, S_n) = (C_1, C_2, \ldots, C_n) \text{ tel que : } C_1 = S_1$$

$$\wedge\ C_2 = \{C_2^i \in \mathcal{D}_2 \mid (c_1^i, C_2^i, c_3^i, \ldots, c_n^i) \in Y\ \forall\ c_1^i \in C_1, \forall\ c_3^i \in S_3, \ldots, \forall\ c_n^i \in S_n\}$$

$$\vdots$$

$$\wedge\ C_n = \{C_n^i \in \mathcal{D}_n \mid (c_1^i, c_2^i, \ldots, c_{n-1}^i, C_n^i) \in Y\ \forall\ c_1^i \in C_1, \ldots \forall\ c_{n-1}^i \in C_{n-1}\}$$

Proposition 2 *h est un opérateur de fermeture.*

Preuve 1 *Afin de prouver que* h *est un opérateur de fermeture, nous devons prouver qu'il vérifie les trois propriétés d'**extensivité**, d'**idempotence** et d'**isotonie** [Couch et Chiarini, 2008].*

*(1) **Extensivité** : Soit* $S = (S_1, S_2, \ldots, S_n)$ *un n-set de* $\mathbb{K}^n \Rightarrow h(S) = (C_1, C_2, \ldots, C_n)$ *tel que :* $C_1 = \{C_1^i \in \mathcal{D}_1 \mid (C_1^i, c_2^i, \ldots, c_n^i) \in \mathrm{Y}\ \forall\ c_2^i \in S_2, \ldots, \forall\ c_n^i \in S_n\} \supseteq S_1$ *puisque nous avons :* $(C_1^i, c_2^i, \ldots, c_n^i) \in \mathrm{Y}\ \forall\ c_1^i \in S_1, \forall\ c_2^i \in S_2, \ldots, \forall\ c_n^i \in S_n\}$, $\ldots$, $C_n = \{C_n^i \in \mathcal{D}_n \mid (c_1^i, c_2^i, \ldots, c_{n-1}^i, C_n^i) \in \mathrm{Y}\ \forall\ c_1^i \in C_1, \forall\ c_2^i \in C_2, \ldots \forall\ c_{n-1}^i \in C_{n-1}\} \supseteq S_n$ *puisque* $C_1 \supseteq S_1$, $C_2 \supseteq S_2, \ldots, C_{n-1} \supseteq S_{n-1}$. *Ainsi,* $(S_1, S_2, \ldots, S_n) \subseteq (C_1, C_2, \ldots, C_n) \Rightarrow S \subseteq h(S)$

*(2) **Idempotence** : Soit* $S = (S_1, S_2, \ldots, S_n)$ *un n-set de* $\mathbb{K}^n \Rightarrow h(S) = (C_1, C_2, \ldots, C_n) \Rightarrow h(C_1, C_2, \ldots, C_n) = (C'_1, C'_2, \ldots, C'_n)$ *tel que :* $C'_1 = \{C_1^{i'} \in \mathcal{D}_1 \mid (C_1^{i'}, c_2^i, \ldots, c_n^i) \in \mathrm{Y}\ \forall\ c_2^i \in S_2, \ldots, \forall\ c_n^i \in S_n\} = C_1, \ldots, C'_n = \{C_n^{i'} \in \mathcal{D}_n \mid (c_1^{i'}, c_2^{i'}, \ldots, c_{n-1}^{i'}, C_n^{i'}) \in \mathrm{Y}\ \forall\ c_1^{i'} \in C'_1, \forall\ c_2^{i'} \in C'_2, \ldots \forall\ c_{n-1}^{i'} \in C'_{n-1}\} = C_n$ *puisque nous avons* $C'_1 = C_1$, $C'_2 = C_2, \ldots, C'_{n-1} = C_{n-1}$. *Ainsi,* $(C'_1, C'_2, \ldots, C'_n) = (C_1, C_2, \ldots, C_n) \Rightarrow h(h(S)) = h(S)$

*(3) **Isotonie** : Soit* $S = (S_1, S_2, \ldots, S_n)$ *et* $S' = (S'_1, S'_2, \ldots, S'_n)$ *deux n-sets de* $\mathbb{K}^n$ *avec* $S \subseteq S' \Rightarrow h(S) = (C_1, C_2, \ldots, C_n)$ *et* $h(S') = (C'_1, C'_2, \ldots, C'_n)$ *tel que :*

- $C_1 = \{C_1^i \in \mathcal{D}_1 \mid (C_1^i, c_2^i, \ldots, c_n^i) \in \mathrm{Y}\ \forall\ c_2^i \in S_2, \ldots, \forall\ c_n^i \in S_n\}$ *et* $C'_1 = \{C_1^{i'} \in \mathcal{D}_1 \mid (C_1^{i'}, c_2^i, \ldots, c_n^i) \in \mathrm{Y}\ \forall\ c_2^i \in S_2, \ldots, \forall\ c_n^i \in S_n\} \Rightarrow C_1 \subseteq C'_1$ *puisque* $S_2 \subseteq S'_2, \ldots, S_n \subseteq S'_n$
  $\vdots$
- $C_n = \{C_n^i \in \mathcal{D}_n \mid (c_1^i, c_2^i, \ldots, c_{n-1}^i, C_n^i) \in \mathrm{Y}\ \forall\ c_1^i \in C_1, \forall\ c_2^i \in C_2, \ldots \forall\ c_{n-1}^i \in C_{n-1}\}$ *et* $C'_n = \{C_n^{i'} \in \mathcal{D}_n \mid (c_1^{i'}, c_2^{i'}, \ldots, c_{n-1}^{i'}, C_n^{i'}) \in \mathrm{Y}\ \forall\ c_1^{i'} \in C'_1, \forall\ c_2^{i'} \in C'_2, \ldots \forall\ c_{n-1}^{i'} \in C'_{n-1}\} \Rightarrow C_n \subseteq C'_n$ *puisque* $C_1 \subseteq C'_1, C_2 \subseteq C'_2, \ldots, C_{n-1} \subseteq C'_{n-1}$

Alors, $(C_1, C_2, \ldots, C_n) \subseteq (C'_1, C'_2, \ldots, C'_n) \Rightarrow h(S) \subseteq h(S')$.
Selon (1), (2) et (3), h est un opérateur de fermeture.

Lorsque n=4, nous instantancions l'opérateur de fermeture pour un contexte quadratique, *i.e.,* une *v-folksonomie* comme suit :

Définition 12 (Opérateur de fermeture d'une *v-folksonomie*) *Soit* $S = (A,$

B, C, E) un quadri-set de $\mathbb{F}_v$ *tel que A est* ***maximal*** *avec* $A \times B \times C \times E \subseteq Y$. *L'opérateur de fermeture h d'une v-folksonomie* $\mathbb{F}_v$ *est défini comme suit :*

$h(S) = h(A, B, C, E) = (U, T, R, V) \mid U = A$

$\wedge\ T = \{t_i \in \mathcal{T} \mid (u_i, t_i, r_i, v_i) \in Y\ \forall\ u_i \in U, \forall\ r_i \in C, \forall\ v_i \in E\}$

$\wedge\ R = \{r_i \in \mathcal{R} \mid (u_i, t_i, r_i, v_i) \in Y\ \forall\ u_i \in U, \forall\ t_i \in T, \forall\ v_i \in E\}$

$\wedge\ V = \{v_i \in \mathcal{V} \mid (u_i, t_i, r_i, v_i) \in Y\ \forall\ u_i \in U, \forall\ t_i \in T, \forall\ r_i \in R\}$

Remarque 1 *h(S) permet de calculer les plus grands quadri-sets d'une v-folksonomie* $\mathbb{F}_v$ *contenant, pour chacun, des ensembles maximaux de tags, de ressources et de variables partagés par un groupe d'utilisateurs. L'application de l'opérateur de fermeture h sur un quadri-set donne un* quadri-concept $QC = (U, T, R, V)$.

Comme cela est le cas pour les cas diadique et triadique, l'opérateur de fermeture partitionne l'espace de recherche en classes d'équivalences, que nous introduisons dans ce qui suit :

Définition 13 (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 maximaux d'utilisateurs, tags, ressources et variables, les *GMs* ne contiennent qu'un seul ensemble maximal (*i.e.,* les utilisateurs) commun à un seul tag, une seule ressource et une seule variable. Dans ce qui suit, nous introduisons l'extension de la définition d'un *GM* à une *v-folksonomie.*

Définition 14 *(*Quadri-générateur minimal*) Soient g =(A, B, C, E) un quadri-set de* $\mathcal{F}_v$ *tel que* $A \subseteq \mathcal{U}$, $B \subseteq \mathcal{T}$, $C \subseteq \mathcal{R}$ *et* $E \subseteq \mathcal{V}$ *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)* $h(g) = h(g_1) = qc$.

Exemple 9 *Considérons la v-folksonomie $\mathbb{F}_v$ du Tableau 3.1, le quadri-concept qc = ({u_1, u_2}, {t_2, t_3, t_4}, {r_1, r_3}, {v_1, v_2}) et la classe d'équivalence représentée par le quadri-concept qc. Soient g_1=({u_1, u_2}, {t_2}, {r_1}, {v_1}) et g_3 = ({u_1, u_2}, {t_2}, {r_1, r_3}, {v_1}) deux quadri-sets. g_1 est un quadri-générateur du quadri-concept qc puisque $h(g_1)$ = qc. Cependant, bien que $h(g_3)$ = qc, g_3 n'est pas un quadri-générateur de qc étant donné que g_1.extent=g_3.extent, g_1.intent $\subset$ g_3.intent, g_1.modus = g_3.modus et g_1.variable = g_3.variable.*

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

3.3 Pseudo code de QUADRICONS

Avant d'introduire le pseudo code de notre algorithme QUADRICONS, nous introduisons, dans ce qui suit, les deux problèmes que nous nous proposons de résoudre. Tout d'abord, nous présentons le problème d'extraction des quadri-sets fréquents à partir d'une *v-folksonomie*.

Problème 1 ***(Problème d'extraction des quadri-sets fréquents)*** *Soit $\mathbb{F}_v$ = ($\mathcal{U}$, $\mathcal{T}$, $\mathcal{R}$, $\mathcal{V}$, Y) une v-folksonomie et $minsupp_u$, $minsupp_t$, $minsupp_r$ et $minsupp_v$ quatre seuils minimaux de support définis par l'utilisateur. La tâche d'extraction de tous les quadri-sets fréquents consiste à déterminer **tous** les quadri-sets (A, B, C, E) de $\mathbb{F}_v$ tels que | A | $\geq$ $minsupp_u$, | B | $\geq$ $minsupp_t$, | C | $\geq$ $minsupp_r$ et | E | $\geq$ $minsupp_v$.*

À présent, nous introduisons le problème d'extraction des quadri-concepts fréquents à partir d'une *v-folksonomie*.

Problème 2 ***(Problème d'extraction des quadri-concepts fréquents)*** *Soit $\mathbb{F}_v$ = ($\mathcal{U}$, $\mathcal{T}$, $\mathcal{R}$, $\mathcal{V}$, Y) une v-folksonomie et $minsupp_u$, $minsupp_t$, $minsupp_r$ et $minsupp_v$ quatre seuils minimaux de support définis par l'utilisateur. La tâche d'extraction de tous les quadri-concepts fréquents consiste à déterminer **tous** les quadri-concepts (U, T, R, V) de $\mathbb{F}_v$ tels que | U | $\geq$ $minsupp_u$, | T | $\geq$ $minsupp_t$, | R | $\geq$ $minsupp_r$ et | V | $\geq$ $minsupp_v$. L'ensemble des quadri-concepts fréquents de $\mathbb{F}_v$ est égal à $\mathcal{QC}$ = {qc | qc = (U, T, R, V) est un quadri-concept fréquent}.*

Remarque 2 *Il est important de noter que la représentation des quadri-concepts fréquents est extraite sans perte d'information. C'est pour cette raison qu'après avoir résolu le Problème 2, nous pouvons facilement résoudre le Problème 1 en énumérant tous les quadri-sets (A, B, C, E) tel qu'il existe un quadri-concept fréquent (U, T, R, V) tel que* $A \subseteq U$, $B \subseteq T$, $C \subseteq R$, $E \subseteq V$ *et* $\mid A \mid \geq minsupp_u$, $\mid B \mid \geq minsupp_t$, $\mid C \mid \geq minsupp_r$ *et* $\mid E \mid \geq minsupp_v$.

Se basant sur les notions introduites précédemment, nous proposons à présent notre nouvel algorithme QUADRICONS dédié à la tâche d'extraction des quadri-concepts fréquents à partir d'une *v-folksonomie* (*cf.,* Problème 2). QUADRICONS opère en quatre étapes, dont l'ordre peut varier selon les dimensions :

1. l'extraction des quadri-générateurs grâce à la procédure FINDMINIMALGENERATORS
2. le calcul du *modus* de chaque quadri-concept
3. le calcul de l'*intent* de chaque quadri-concept
4. le calcul de la *variable* de chaque quadri-concept

Les trois dernières étapes sont réalisées grâce à la procédure CLOSURECOMPUTE.

Le pseudo code de l'algorithme QUADRICONS est donné par l'Algorithme 1. QUADRICONS prend en entrée une *v-folksonomie* $\mathcal{F}_v = (\mathcal{U}, \mathcal{T}, \mathcal{R}, \mathcal{V}, \mathrm{Y})$ ainsi que quatre seuils minimaux de support (un pour chaque dimension) : $minsupp_u$, $minsupp_t$, $minsupp_r$ et $minsupp_v$. 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, v) l'ensemble U_s représentant l'ensemble maximal d'utilisateurs ayant une même variable en commun 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'Al-

Algorithme 1 : QUADRICONS

Données :

1. $\mathcal{F}_v$ ($\mathcal{U}$, $\mathcal{T}$, $\mathcal{R}$, $\mathcal{V}$, Y) : une *v-folksonomie*.
2. $minsupp_u$, $minsupp_t$, $minsupp_r$, $minsupp_v$: seuils minimaux de support.

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

```
1  début
2      QS : un ensemble de quadri-sets ;
3      /*Étape 1 : L'extraction des quadri-générateurs*/
4      MG = FindMinimalGenerators(F_v, minsupp_u) ;
5      /*Étape 2 : le calcul de la partie modus*/
6      pour chaque quadri-gen g ∈ MG faire
7          ClosureCompute(MG, minsupp_u, minsupp_t, minsupp_r, g, QS, 1) ;
8      PruneInfrequentSets(QS,minsupp_t) ;
9      /*Étape 3 : Le calcul de la partie intent*/
10     pour chaque quadri-set s ∈ QS faire
11         ClosureCompute( QS, minsupp_u, minsupp_t, minsupp_r, s, QS, 2) ;
12     PruneInfrequentSets(QS,minsupp_r) ;
13     /*Étape 4 : Le calcul de la partie variable*/
14     pour chaque  quadri-set s ∈ QS faire
15         ClosureCompute( QS, minsupp_u, minsupp_t, minsupp_r, s, QC, 3) ;
16     PruneInfrequentSets(QC,minsupp_v) ;
17 fin
18 retourner QC ;
```

gorithme 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 5) 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 i (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 11). La quatrième et dernière étape

Algorithme 2 : FindMinimalGenerators

Données :

1. $\mathcal{F}_v$ ($\mathcal{U}$, $\mathcal{T}$, $\mathcal{R}$, $\mathcal{V}$, Y) : une *v-folksonomie*.
2. $minsupp_u$: seuil minimal de support.

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

```
1  début
2      g :un quadri-générateur ;
3      pour chaque quadruplet (u, t, r, v) de F_v faire
4          U_s= {u_i ∈ U | (u_i, t, r, v) ∈ Y} ;
5          si | U_s | ≥ minsupp_u alors
6              g.extent = U_s ; g.intent = r ; g.modus = t ; g.variable = v
7              si g ∉ MG alors
8                  AddQuadri(MG, g)
9  fin
10 retourner MG ;
```

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 17) 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 $minsupp_v$. QuadriCons invoque la fonction **PruneInfrequentSets** (Lignes 7, 11 et 15) afin d'élaguer les quadri-sets/concepts infréquents, *i.e.*, ceux dont la cardinalité du *modus/intent/variable* ne vérifie pas les seuils demandés.

3.4 Exemple illustratif

Considérons la *v-folksonomie* représentée par le Tableau dans la Figure 3.1, avec $minsupp_u = 2$, $minsupp_t = 2$, $minsupp_r = 1$ et $minsupp_v = 1$. La Figure 3.1 illustre la trace d'exécution de QuadriCons sur ce contexte. Comme décrit dans la section précédente, QuadriCons opère en quatre étapes :

1. *(Étape 1)* La première étape de QuadriCons consiste en l'extraction des quadri-

Algorithme 3 : CLOSURECOMPUTE

Données :

1. $\mathcal{S}_{IN}$: l'ensemble d'entrée, min_u, min_t, min_r : seuils minimaux de supports.
2. q : un quadri-générateur/quadri-set, i : un indicateur.

Résultats : $\mathcal{S}_{OUT}$: l'ensemble de sortie.

début
  pour chaque *quadri-set* $q' \in \mathcal{S}_{IN}$ **faire**
    si i=1 ***et*** $q.intent = q'.intent$ ***et*** $q.extent \subseteq q'.extent$ **alors**
      $s.intent = q.intent$; $s.extent = q.extent$; $s.variable = q.variable$; $s.modus = q.modus \cup q'.modus$; ADDQUADRI($\mathcal{S}_{OUT}$, s) ;
    sinon si i=1 ***et*** $q.intent = q'.intent$ ***et*** q *et* q' incomparables **alors**
      $g.extent = q.extent \cap q'.extent$; $g.modus = q.modus \cup q'.modus$; $g.intent = q.intent$; $g.variable = q.variable$;
      If g est *u-frequent* **then** ADDQUADRI($\mathcal{MG}$, g) ;
    sinon si i=2 ***et*** $q.extent \subseteq q'.extent$ ***et*** $q.modus \subseteq q'.modus$ ***et*** $q.intent \neq q'.intent$ **alors**
      $qs.extent = q.extent$; $qs.modus = q.modus$; $qs.variable = q.variable$; $qs.intent = q.intent \cup q'.intent$;
      ADDQUADRI($\mathcal{S}_{OUT}$, qs) ;
    sinon si i=2 ***et*** q *et* q' incomparables **alors**
      $s.extent = q.extent \cap q'.extent$; $s.modus = q.modus \cap q'.modus$; $s.variable = q.variable$; $s.intent = q.intent \cup q'.intent$;
      Si s is *u-frequent* et *t-frequent* **alors** ADDQUADRI($\mathcal{S}_{OUT}$, s) ;
    sinon si i=3 ***et*** $q.extent \subseteq q'.extent$ ***et*** $q.modus \subseteq q'.modus$ ***et*** $q.intent \subseteq q'.intent$ ***et*** $q.variable \neq q'.variable$ **alors**
      $qc.extent = q.extent$; $qc.modus = q.modus$; $qc.intent = q.intent$; $qc.variable = q.variable \cup q'.variable$;
      ADDQUADRI($\mathcal{S}_{OUT}$, qc) ;
    sinon si i=3 ***et*** q *et* q' incomparables **alors**
      $s.extent = q.extent \cap q'.extent$; $s.modus = q.modus \cap q'.modus$; $s.intent = q.intent \cap q'.intent$; $s.variable = q.variable \cup q'.variable$;
      Si s est *u,t,r-frequent* **alors** ADDQUADRI($\mathcal{S}_{OUT}$, s) ;
fin
retourner $\mathcal{S}_{OUT}$;

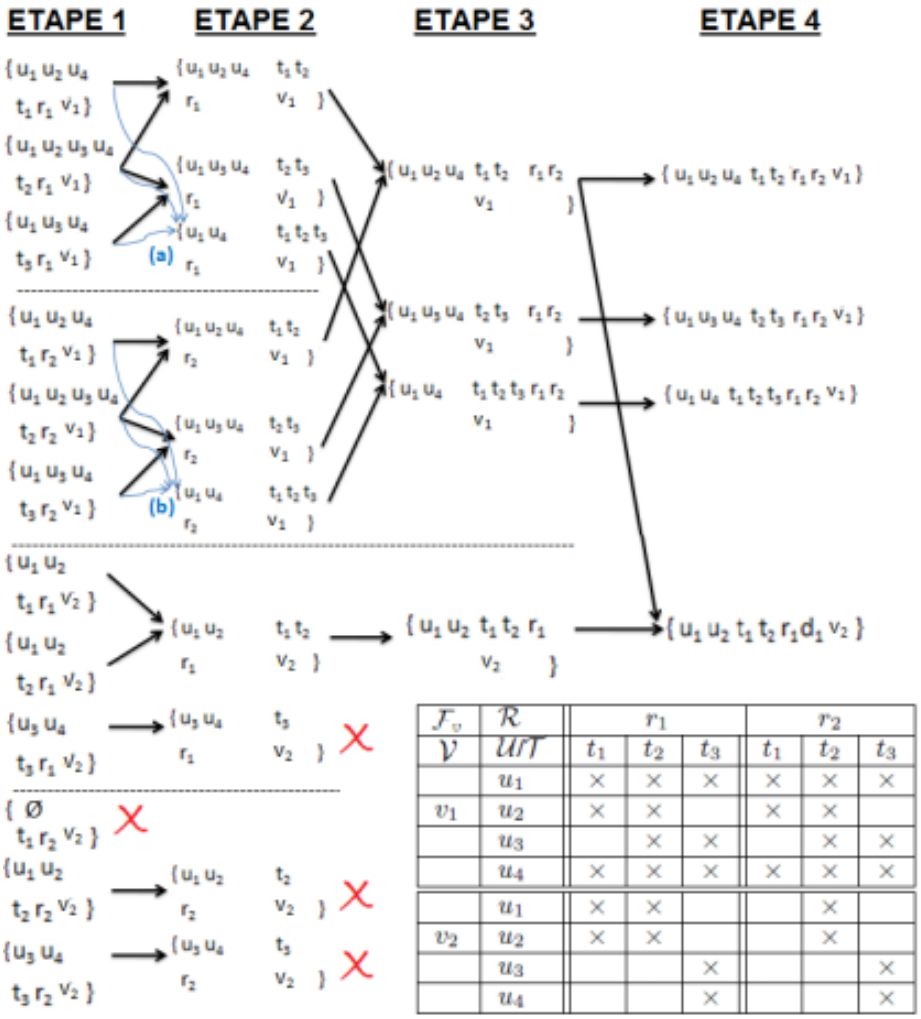
générateurs (*QGs*) à partir de la *v-folksonomie* (Algorithme 1, Ligne 3). Les *QGs* sont des ensembles maximaux d'utilisateurs relatifs à un triplet de tag, de ressource et d'une variable. Ainsi QuadriCons permet d'extraire 12 *QGs* dont 11 qui sont fréquents par rapport au seuil minimal de support $minsupp_u$ (*cf.*, Figure 3.1, Étape 1).

2. *(Étape 2)* Ensuite, QuadriCons invoque la procédure ClosureCompute une première fois pour les quadri-générateurs afin de calculer le *modus* (l'ensemble de tags) de ces candidats (Algorithme 1, Lignes 5-7). Par exemple, étant donné que l'*extent* (l'ensemble d'utilisateurs) du candidat $\{\{u_1, u_2, u_4\}, t_1, r_1, v_1\}$ est inclus dans celui du candidat $\{\{u_1, u_2, u_3, u_4\}, t_2, r_1, v_1\}$, le modus du premier *QG* sera incrémenté et égal à $\{t_1, t_2\}$. Par ailleurs, de nouveaux quadri-générateurs peuvent être crées à partir de l'intersection des premiers (Algorithme 3, Lignes 5-7) : c'est notamment le cas des deux *QGs (a)* et *(b)* (*cf.*, Figure 3.1, Étape 2). Enfin, à ce stade de l'exécution, les candidats qui ne vérifient pas le seuil minimal de support $minsupp_t$ sont élagués (*i.e.*, les trois derniers candidats).
3. *(Étape 3)* Par suite, QuadriCons procède au calcul de l'*intent* (l'ensemble de ressources) de chaque candidat à travers un second appel à la procédure ClosureCompute (Algorithme 1, Lignes 9-11). Par exemple, le candidat $\{\{u_1, u_2, u_4\}, \{t_1, t_2\}, r_1, v_1\}$, dont l'*extent*, *modus* et *variable* sont inclus ou égaux dans ceux du candidat $\{\{u_1, u_2, u_4\}, \{t_1, t_2\}, r_2, v_1\}$, voit son *intent* être incrémenté et devient égal à $\{r_1, r_2\}$. À ce niveau, seuls quatre candidats vérifient encore le seuil minimal de support $minsupp_r$ (*cf.*, Figure 3.1, Étape 3). Ainsi, en plus de calculer la partie *intent* de chaque candidat, cette étape a permis de réduire considérablement le nombre de candidats.
4. *(Étape 4)* Via un ultime appel à la procédure ClosureCompute, QuadriCons procède au calcul de la partie *variable* de chaque candidat avant d'en élaguer les infréquents (Algorithme 1, Lignes 13-15). Par exemple, étant donné que le candidat $\{\{u_1, u_2\}, \{t_1, t_2\}, r_1, v_2\}$ possède un *extent*, un *modus* et un *intent* qui sont inclus dans ceux du candidat $\{\{u_1, u_2, u_4\}, \{t_1, t_2\}, \{r_1, r_2\}, v_1\}$ [1], ainsi, sa partie *variable* devient égal à $\{v_1, v_2\}$ (*cf.*, Figure 3.1, Étape 4).

Après la quatrième étape, QuadriCons termine. Les quatre quadri-concepts fréquents extraits sont :

1. Concrètement, cela veut dire que les utilisateurs u_1 et u_2 qui ont partagé la ressource r_1 avec les tags t_1 et t_2 à travers la variable v_2 l'ont également partagé à travers la variable v_1.

1. $\{\{u_1, u_2, u_4\}, \{t_1, t_2\}, \{r_1, r_2\}, v_1\}$
2. $\{\{u_1, u_3, u_4\}, \{t_2, t_3\}, \{r_1, r_2\}, v_1\}$
3. $\{\{u_1, u_4\}, \{t_1, t_2, t_3\}, \{r_1, r_2\}, v_1\}$
4. $\{\{u_1, u_2\}, \{t_1, t_2\}, r_1, \{v_1, v_2\}\}$



$\mathcal{T}_v$	$\mathcal{R}$	r_1			r_2		
$\mathcal{V}$	$\mathcal{U}\mathcal{T}$	t_1	t_2	t_3	t_1	t_2	t_3
	u_1	×	×	×	×	×	×
v_1	u_2	×	×		×	×	
	u_3		×	×		×	×
	u_4	×	×	×	×	×	×
	u_1	×	×			×	
v_2	u_2	×	×			×	
	u_3			×			×
	u_4			×			×

Figure 3.1 — Trace d'exécution de QUADRICONS sur la *v-folksonomie* présentée dans la Figure.

3.5 Validité et complexité

Dans ce qui suit, nous prouvons la complétude, terminaison et correction de l'algorithme QUADRICONS puis nous calculons sa complexité dans le pire des cas.

3.5.1 Correction

Proposition 3 *L'algorithme* QuadriCons *est correct et complet. Il extrait exactement* ***tous*** *les quadri-concepts fréquents.*

Preuve 2 *La procédure* FindMinimalGenerators *permet d'extraire tous les quadri-générateurs à partir d'une v-folksonomie $\mathcal{F}_v$ étant donné que tous les quadruples du contexte sont énumérés dans le but de regrouper des groupes maximaux d'utilisateurs par rapport à chaque quadruplet (u,t,r,v) (Algorithme 2, Lignes 3-8). Ce parcours de tous les quadruplets permet d'extraire efficacement tous les quadri-générateurs. A partir de ces quadri-générateurs ainsi extraits,* QuadriCons *invoque la procédure* Closure-Compute *trois fois dans le but de calculer le* modus, *l'*intent *et la* variable *de chaque quadri-générateur. À chaque appel, i.e., $i = 1, 2, 3$, pour chaque couple de candidats q et q', deux cas apparaissent :*

1. *(Algorithme 3, lignes 3, 8 et 14) q et q' sont comparables. Ainsi, un nouveau quadri-set (quadri-concept si $i = 3$) est crée à partir de l'union des différentes parties des deux candidats.*
2. *(Algorithme 3, lignes 5, 11 et 17) q et q' sont incomparables. Ainsi, un nouveau quadri-set (quadri-générateur si $i = 1$) est crée à partir de l'intersection des différentes parties de q et q'.*

Ainsi, tous les cas de comparaisons entre les candidats ont été énumérés. Finalement, la procédure PruneInfrequentSets *permet d'élaguer les quadri-concepts infréquents selon les seuils minimaux de supports (Algorithme 1, lignes 7, 11 et 15). Nous concluons que* QuadriCons *permet d'extraire tous les quadri-concepts fréquents. Donc, il est correct.*

3.5.2 Terminaison

Proposition 4 *L'algorithme* QuadriCons *termine.*

Preuve 3 *Le nombre de quadri-générateurs générés par* QuadriCons *est fini. En effet, le nombre de* QGs *candidats générés à partir d'un contexte ($\mathcal{U}$, $\mathcal{T}$, $\mathcal{R}$, $\mathcal{V}$) est au maximum égal à $|\mathcal{T}| \times |\mathcal{R}| \times |\mathcal{V}|$. Puisque l'ensemble $\mathcal{MG}$ de quadri-générateurs est fini, les trois boucles de l'algorithme 1 parcourant cet ensemble sont alors, eux aussi, finis.*

De plus, le nombre total de quadri-concepts fréquents générés par QuadriCons *est, au maximum, égal à* $2^{|\mathcal{T}|+|\mathcal{R}|+|\mathcal{V}|}$. *Donc, l'algorithme* QuadriCons *termine.*

3.5.3 Complexité théorique

Comme pour le cas triadique [Jäschke *et al.*, 2008], 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 $n = |\mathcal{T}| + |\mathcal{R}| + |\mathcal{V}|$. Néanmoins, et comme il sera démontré, dans la partie expérimentale, d'un point de vue pratique, les performances réelles sont loin d'être exponentielles. De ce fait, nous concentrons notre évaluation sur des jeux de données à large échelle.

3.6 Étude expérimentale

Dans cette partie, nous démontrons à travers nos expérimentations sur des jeux de données les performances de QuadriCons *vs.* ceux de l'algorithme Data Peeler en termes de temps d'exécution et de mémoire consommée. De plus, nous donnons quelques exemples de quadri-concepts extraits à partir de quelques jeux de données du monde réel. 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 #d-peeler*. Nous avons utilisé un processeur Intel® Core *i*5 muni d'une mémoire de 8 GB. Les tests ont été menés sur le système d'exploitation Linux (Distribution Ubuntu 12.04 64 bits).

3.6.1 Jeux de données

Pour mener à bien notre évaluation, nous avons utilisé deux jeux de données du monde réel décrits comme suit :

– Le jeu de données de filmographie MovieLens[2] : il s'agit d'un système de recommandation et d'un siteweb de communauté virtuelle qui permet aux utilisateurs de partager des films via des tags. Le jeu de données utilisé pour notre évaluation est téléchargeable gratuitement[3] et contient 15227 tags appliqués à 11272 films

2. *http ://movielens.umn.edu/*
3. *http ://www.grouplens.org/node/73*

par 4010 utilisateurs (*e.g.*, <*Alex*, *X-Files*, *sciencefiction*>).

– Le jeu de données LAST.FM (*http ://last.fm*) est un siteweb de musique en ligne, crée en 2002. Il regroupe plus de 30 millions d'utilisateurs actifs (depuis Mars 2009). Les utilisateurs partagent leurs artistes préférés en les annotant par des tags (*e.g.*, <*Ross*, *MichaelJackson*, *kingofpop*>). Le jeu de données utilisé pour notre évaluation est téléchargeable gratuitement [4]

Afin de modéliser la quatrième dimension, les jeux de données que nous avons utilisé, soit le **profil** (MOVIELENS), soit le **temps** (MOVIELENS, LAST.FM) :

Le temps comme quatrième dimension Le Tableau 3.2 démontre les caractéristiques des deux jeux de données utilisées et qui utilisent le temps (timestamp) comme variable pour la quatrième dimension. Cette variable nous renseigne sur la date à laquelle un utilisateur donné a partagé une ressource donnée avec un tag donné. Les deux jeux de données concernées sont MOVIELENS et LAST.FM. Le Tableau 3.3 montre quelques exemples de quadruplets pour les deux jeux de données.

Le profil comme quatrième dimension Lorsque la variable de la quatrième dimension correspond au profil, plusieurs informations supplémentaires sur les utilisateurs sont données et nous renseignent sur le **genre** de l'utilisateur (homme ou femme), 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. Le Tableau 3.4 montre quelques exemples de quadruplets pour le jeu de données MOVIELENS.

3.6.2 Exemples de quadri-concepts

Dans ce qui suit, nous présentons quelques résultats intéressants de quadri-concepts extraits par QUADRICONS à partir des jeux de données décrits précédemment. Nous présentons d'abord des quadri-concepts dont la quatrième dimension correspond au temps, ensuite ceux avec le profil des utilisateurs comme quatrième varibale.

4. *http ://mtg.upf.edu/static/datasets/last.fm/lastfm-dataset-1K.tar.gz*

	Jeu de données 1	Jeu de données 2
	(MovieLens)	(Last.fm)
# Quadruplets	95580	186479
# Utilisateurs	4010	1892
# Tags	15227	9749
# Ressources	11272 (films)	12523 (artistes)
# Dates (timestamps)	81601	3549
Périodes	22/12/2005 - 20/08/2008	10/04/2008 - 07/05/2011

Tableau 3.2 — Caractéristiques des jeux de données considérés.

Utilisateur	Tag	Ressource	Date
Ross	*cartoon*	*The Lion King*	22/05/2007
Ross	*jungle*	*The Lion King*	22/05/2007
Adele	*cult*	*Mrs. Doubtfire*	16/01/2008
Adele	*robbiewilliams*	*Mrs. Doubtfire*	16/01/2008
Hallo	*thriller*	*Silence of the Lambs*	07/02/2006
⋮	⋮	⋮	⋮

Utilisateur	Tag	Ressource	Date
Regina	*reggae*	*Bob Marley*	18/05/2009
Regina	*legend*	*Bob Marley*	18/05/2009
MrGold	*rap*	*Eminem*	09/02/2010
Snow_white	*legend*	*Michael Jackson*	30/06/2010
Snow_white	*kingofpop*	*Michael Jackson*	30/06/2010
⋮	⋮	⋮	⋮

Tableau 3.3 — Un aperçu des jeux de données MovieLens et Last.fm lorsque la quatrième dimension correspond au temps.

Utilisateur	Tag	Ressource	Profil
Mulder	*action*	*X-Files*	*student*
Mulder	*sciencefiction*	*X-Files*	*25 years old*
Scully	*adventure*	*Jurassic Park*	*professor*
Scully	*bestmovie*	*Jurassic Park*	*woman*
Skinner	*thriller*	*Carrie*	*Canada*
⋮	⋮	⋮	⋮

Tableau 3.4 — Un aperçu du jeu de données MovieLens lorsque la quatrième dimension correspond au profil.

Le temps comme quatrième dimension

Le Tableau 3.5 montre deux exemples de quadri-concepts fréquents extraits à partir des jeux de données MovieLens et Last.fm. Nous avons défini les valeurs de seuils de supports suivants : $minsupp_u = 2$, $minsupp_t = 2$, $minsupp_r = 2$ et $minsupp_v = 1$. Le premier concept illustre le fait que les utilisateurs *krycek* et *maria* aient utilisé les tags *kids*, *fantasy*, *darkness* et *magic* afin d'annoter le film *Harry Potter* et ses suites successivement le 03/12/2005, 16/07/2006 et le 21/02/2008. Un tel concept peut être exploité plus tard pour une recommandation de tags pour ces films ou bien pour étudier l'évolution de tags associés au film *"Harry Potter"* à travers le temps. Le second quadri-concept montre que les utilisateurs *csmdavis*, *franny* et *rossanna* ont tous utilisé les tags *pop*, *concert* et *dance* pour décrire les artistes *Britney Spears* et *Madonna* le 07/05/10 puis le 02/06/11. Nous pourrons par exemple utiliser un tel quadri-concept afin de recommander les utilisateurs *franny* et *rossanna* à *csmdavis* étant donné qu'ils partagent le même intérêt pour les deux artistes. Il serait aussi intéressant d'étudier l'évolution du vocabulaire utilisé par les *fans* de ces artistes à travers le temps afin de voir les tendances qui se dégagent sur ces artistes selon les périodes.

Le profil comme quatrième dimension

Dans ce qui suit, nous présentons quelques résultats intéressants de quadri-concepts fréquents extraits par Quadricons à partir du jeu de données MovieLens lorsque la quatrième dimension correspond au profil. Nous avons défini les valeurs de seuils de supports suivants : $minsupp_u = 2$, $minsupp_t = 2$, $minsupp_r = 2$ et $minsupp_v = 1$, *i.e.,* dans un quadri-concept fréquent, 2 utilisateurs (au moins) avec une information de

Jeux de données	Dates	Utilisateurs	Tags	Ressources
MOVIE LENS	03/12/05 16/07/06 21/02/08	*krycek* *maria*	*kids* *fantasy* *darkness* *magic*	*Harry Potter 1* *The Prisoner of Azkaban* *The Order of the Phoenix*
LAST.FM	07/05/10 02/06/11	*csmdavis* *franny* *rossanna*	*pop* *concert* *dance*	*Britney Spears* *Madonna*

Tableau 3.5 — Examples de quadri-concepts fréquents extraits à partir des jeux de données MovieLens et Last.fm.

profil en commun (*e.g.,* même profession, 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 au moins une information de profil en commun. Ainsi, le Tableau 3.6 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*. Le troisième quadri-concept nous renseigne que trois écrivains âgés entre 36 et 45 ans, *i.e., ross*, *anlucia* et *franela* ont opté pour les tags *quotes*, *classic* et *oldmovie* lorsqu'ils ont partagé les films *Braveheart*, *Magician of OZ* et *Gone with the Wind*.

À présent, nous nous intéressons à des quadri-concepts fréquents de MovieLens dont le profil concerne une seule information. Le but est d'observer la différence de vocabulaire et d'intérêts entre les utilisateurs de différentes professions ou de différentes villes dans le monde :

Profession Le Tableau 3.7, qui décrit des quadri-concepts lorsque le profil concerne la profession des utilisateurs, montre une différence de vocabulaire et d'intérêts entre des utilisateurs issus de professions différentes. Par exemple, les étudiants ont massivement partagé les films d'action via les tags *adventure* et *action*, tandis

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}

Tableau 3.6 — Exemples de quadri-concepts extraits à partir du jeu de données MovieLens.

que les avocats préferent partager des films avec une histoire criminelle comme *The Fugitive* avec des tags comme *detective* et *crime_story*. Notons, par ailleurs, que les bibliothécaires sont intéressés par les films qui sont des adaptations de romans comme *Dead poets society* qui correspondent mieux à leurs centres d'intérêts.

Age Dans cette partie, nous nous sommes intéressés à l'analyse de quadri-concepts avec des utilisateurs issus de différentes catégories d'âge afin d'observer les différents intérêts de chaque catégorie. Chaque quadri-concept du Tableau 3.8 correspond à une catégorie d'âge. Ainsi, nous observons que les utilisateurs âgés de moins de 18 ans préférent les films de science-fiction alors que les utilisateurs âgés entre 25 et 35 ans sont plus intéressés par les films d'action et d'aventure. Par ailleurs, les utilisateurs les plus âgés ont un intérêt tout paticulier pour les vieux films et les grands classiques comme *Casablanca* ou encore *Rear Window*. La catégorie d'âge la plus représentée sur le jeu de données MovieLens est celle des utilisateurs âgés entre 19 et 24 ans qui ont tendance à partager les polars en utilisant les tags *adventure* et *polar*.

Genre Enfin, le Tableau 3.9 montre des exemples de quadri-concepts lorsque le profil correspond au genre des utilisateurs. Il démontre la différence de tagging et d'intérêts entre les hommes et les femmes. Si ces dernières montrent un plus grand intérêt pour les films de romance avec une histoire d'amour comme *Sweet November* ou *Titanic* utilisant des tags comme *romance* ou *lovestory*, les hommes sont plus intéressés par les films d'actions et les thriller comme *Blade Runner*. Grâce aux quadri-concepts qui permettent de regrouper les utilisateurs ayant un même

profil, le Tableau 3.9 a permis d'identifier deux intérêts complètement différents qui peuvent être pris en compte au moment de suggérer aux utilisateurs des films par exemple.

U	{Krycek,Deepthr,Mistx12}
T	{author,based_on_a_book}
R	{The Fugitive,Dead poets society}
P	{librarian}
U	{Fox16,Dana,Cgbspender}
T	{adventure,polar}
R	{Pulp Fiction,The Godfather,The Fugitive}
P	{lawyer}
U	{Mylafi,Jenifer,Nabilawi}
T	{thriller,action}
R	{Braveheart,Magician of OZ}
P	{retired}
U	{Fran,Chandy,Joeytr,Phoeb}
T	{award,oldmovie,classic}
R	{Star Wars,Blade Runner,Monty Python}
P	{engineer}
U	{Ched50,Slioua7,Nina16}
T	{classic,dialog,oscar}
R	{Seven,Appolo 13,Raiders of Lost Ark}
P	{student}

Tableau 3.7 — Exemples de quadri-concepts selon le profil *profession*.

Utilisateurs	Tags	Ressources	Profil
steve,patr,mkl23	sciencefiction,cult	Star Wars,Independance Day,The Rock	7-18 ans
nad16,mehz19,emma	adventure,polar	Toy Story,Seven,Star Wars,Braveheart	19-24 ans
cr7,ines16,sir,lionel	thriller,action	Taxi Driver,Star Wars,Blade Runner	25-35 ans
menez,verra18,pasto	award,oldmovie,classic	Silence Lambs,Casablanca,Usual Suspects	36-45 ans
bernad,bridg,robert	classic,dialog,oscar	Star Wars,M.A.S.H,Rear Window	46-73 ans

Tableau 3.8 — Exemples de quadri-concepts selon the profil *âge*.

Utilisateurs	Tags	Ressources	Profil
lea,chris,david	crime,thriller,action	Silence Lambs,The Rock,Blade Runner	Homme
regina,gillian	passion,romance,lovestory	B.Jones,Titanic,Sound Music	Femme

Tableau 3.9 — Exemples de quadri-concepts selon le profil *genre*.

3.6.3 Compacité des quadri-concepts par rapport aux quadri-sets

Les Figures 3.2 et 3.3 illustrent le nombre de quadri-concepts fréquents versus celui des quadri-sets fréquents sur les jeux de données MovieLens et Last.fm pour différentes valeurs de quadruplets et différents valeurs de seuils minimaux de support. Nous observons que pour les deux jeux de données, le nombre de quadri-sets fréquents augmente rapidement lorsque le nombre de quadruplets augmente. En effet, cela est dû au fait que les quadri-concepts sont larges, *i.e.,* contiennent plusieurs utilisateurs, tags, ressources et variables, ce qui augmente la taille de chaque partie relative à chaque dimension. Cela a pour effet l'augmentation du nombre de quadri-sets qui sont des sous-ensembles des quadri-concepts. Sur les deux jeux de données, le nombre des quadri-concepts fréquents représente seulement 3. 68 % et 28. 99 % du nombre des quadri-sets fréquents. Donc, le calcul des quadri-sets fréquents est une tâche plus difficile et coûteuse que celle des quadri-concepts fréquents alors que l'information véhiculée est la même.

3.6.4 Temps d'exécution

Les Tableaux 3.10 et 3.11 démontrent 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 jeu de données MovieLens et de 40000 à 186479 sur le jeu de données Last.FM, et ce pour différentes valeurs de seuils minimaux de supports. Les deux algorithmes permettent l'extraction de tous les quadri-concepts fréquents. Nous observons que pour toutes les valeurs du nombre de quadruplets, Data Peeler est très loin de QuadriCons en termes de temps d'exécution sur les deux jeux de données. QuadriCons tourne jusqu'à 124 fois plus rapidement que son concurrent sur MovieLens et jusqu'à 332 fois plus rapidement sur Last.FM. Nous expliquons cette différence par le fait que le principal point fort de Data Peeler, *i.e.,* sa généricité pour

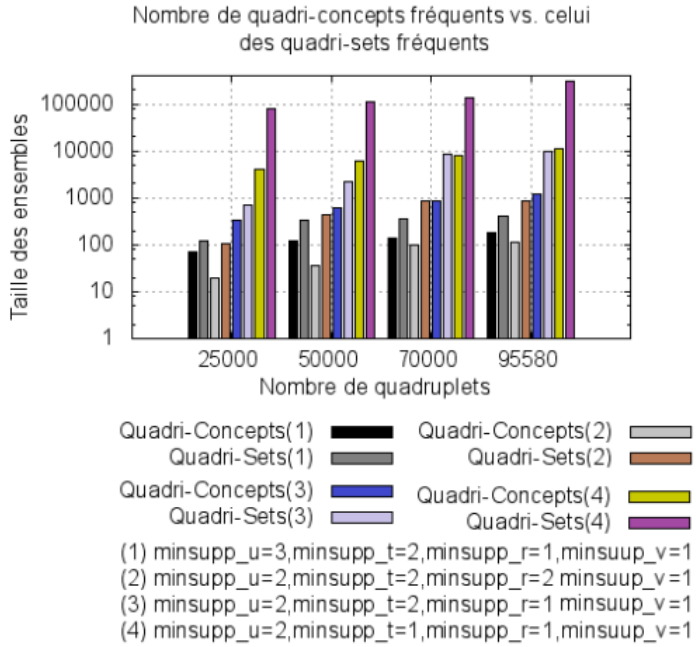


Figure 3.2 — Nombre de quadri-concepts fréquents *vs.* celui des quadri-sets fréquents par rapport au nombre de quadruplets sur le jeu de données MovieLens.

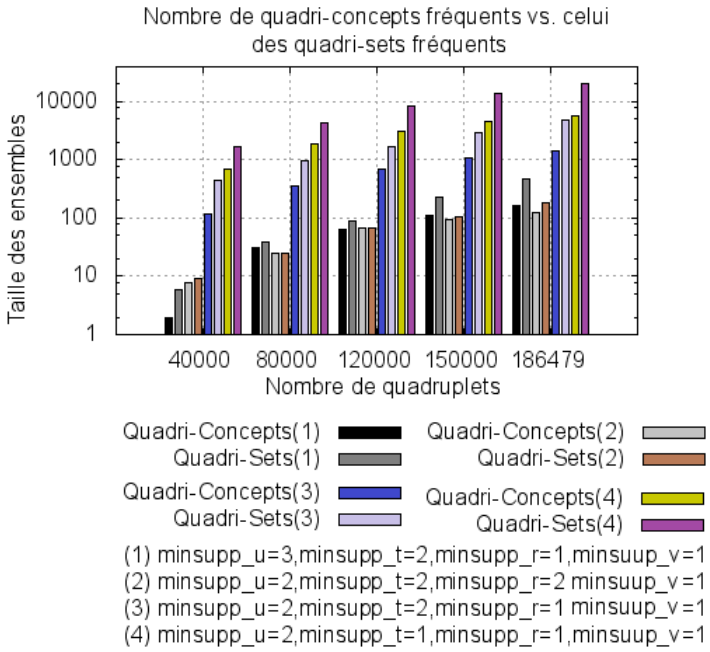


Figure 3.3 — Nombre de quadri-concepts fréquents *vs.* celui des quadri-sets fréquents par rapport au nombre de quadruplets sur le jeu de données Last.FM.

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 et de réduire considérablement le nombre de candidats. Dans des applications pratiques (*e.g.,* recommandations), un utilisateur va préférer un algorithme qui offre un résultat (*e.g.,* des recommandations) rapidement.

3.6.5 Mémoire consommée de QUADRICONS *vs.* DATA PEELER

Les Tableaux 3.10 et 3.11 démontrent la mémoire consommée par les deux algorithmes sur les deux jeux de données MOVIELENS et LAST.FM pour différentes valeurs de quadruplets. QUADRICONS consomme moins de mémoire que son concurrent : moins de 40000 KB et 20000 KB sur les deux jeux de données contre des milliers voire 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 [Trabelsi *et al.*, 2012]. Par exemple, pour extraire les 167 quadri-concepts fréquents à partir de LAST.FM lorsque $minsupp_u = 3$, $minsupp_t = 2$, $minsupp_r = 1$ et $minsupp_v = 1$, QUADRICONS requiert seulement 1754 KB en mémoire pour détecter les 939 quadri-générateurs à partir du jeu de données. Cependant, malgré le peu de quadri-concepts fréquents à extraire dans ce cas, DATA PEELER a requis 788021 KB en mémoire afin de stocker le jeu de données en entier avant de générer les candidats. En conséquence, le fait de détecter les quadri-générateurs avant de procéder à l'extraction des quadri-concepts a permis à QUADRICONS de consommer jusqu'à 54 et 115 fois moins de mémoire que DATA PEELER, respectivement, sur les jeux de données MOVIELENS et LAST.FM. Cependant, 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)
$minsupp_u = 3$, $minsupp_t = 2$, $minsupp_r = 1$, $minsupp_v = 1$				
25000	**0. 86**	**542**	23.10	209843
50000	**2. 05**	**1361**	80.72	378907
70000	**3. 08**	**1760**	128.33	509541
95580	**3. 61**	**2087**	258.00	654761
$minsupp_u = 2$, $minsupp_t = 2$, $minsupp_r = 2$, $minsupp_v = 1$				
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
$minsupp_u = 2$, $minsupp_t = 2$, $minsupp_r = 1$, $minsupp_v = 1$				
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
$minsupp_u = 2$, $minsupp_t = 1$, $minsupp_r = 1$, $minsupp_v = 1$				
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

Tableau 3.10 — Performances de QUADRICONS *vs.* celles de DATA PEELER sur le jeu de données MOVIELENS.

3.7 Conclusion

Dans ce chapitre, nous avons proposé un nouvel algorithme qui se consacre à l'extraction des quadri-concepts à partir des *folksonomies* et qui se base sur un opérateur de fermeture spécialement dédié à ce type de contexte. Cet opérateur permet de partitionner l'espace de recherche en classes d'équivalences dont les plus petits éléments

| Y |	**QUADRI CONS** (sec)	Mémoire Consommée (kilobits)	**DATA PEELER** (sec)	Mémoire Consommée (kilobits)
$minsupp_u = 3$, $minsupp_t = 2$, $minsupp_r = 1$, $minsupp_v = 1$				
40000	**0. 05**	**114**	7.13	309453
80000	**0. 10**	**342**	28.12	445431
120000	**0. 22**	**656**	61.60	550932
150000	**0. 45**	**1241**	119.45	678542
186479	**0. 77**	**1754**	255.71	788021
$minsupp_u = 2$, $minsupp_t = 2$, $minsupp_r = 2$, $minsupp_v = 1$				
40000	**0. 39**	**177**	32.29	456323
80000	**0. 53**	**421**	57.06	590012
120000	**1. 60**	**782**	182.40	698672
150000	**3. 39**	**1025**	354.71	826862
186479	**5. 87**	**1672**	496.55	932871
$minsupp_u = 2$, $minsupp_t = 2$, $minsupp_r = 1$, $minsupp_v = 1$				
40000	**0. 84**	**1876**	51.88	498672
80000	**2. 94**	**3891**	201.58	780762
120000	**8. 71**	**6789**	487.92	1198451
150000	**17. 81**	**11342**	1049.34	1343572
186479	**29. 78**	**14562**	1949.14	1552789
$minsupp_u = 2$, $minsupp_t = 1$, $minsupp_r = 1$, $minsupp_v = 1$				
40000	**2. 91**	**6724**	89.77	1008273
80000	**6. 87**	**11562**	221.93	1336451
120000	**21. 87**	**14345**	724.47	1542006
150000	**46. 52**	**15623**	1524.76	1772919
186479	**88. 16**	**18976**	3118.85	2188452

Tableau 3.11 — Performances de QUADRICONS *vs.* celles de DATA PEELER sur le jeu de données LAST.FM

sont les quadri-générateurs. La principale originalité de notre algorithme est que cette détection des générateurs a permis un balayage intelligent de l'espace de recherche et réduit considérablement le nombre de candidats. L'étude expérimentale que nous avons mené sur des jeux de données du monde réel a mis en lumière notre contribution en améliorant les temps d'exécution tout en réduisant la mémoire consommée de notre algorithme *vs.* celle consommée par DATA PEELER.

Dans le prochain chapitre, nous allons nous appuyer sur les quadri-concepts afin de proposer notre nouveau système personnalisé de recommandation. Ce système s'appuiera sur ces concepts ainsi que sur des informations personnelles des utilisateurs afin de générer des recommandations d'utilisateurs, de tags et de ressources qui répondent au mieux aux besoins des utilisateurs dans les *folksonomies*. Par ailleurs, nous introduisons une mesure de ranking afin d'améliorer la qualité des recommandations. Enfin, une étude expérimentale poussée passera en revue les différentes propriétés de notre système de recommandation, comme la précision, le rappel, la scalabilité, la couverture ou encore la diversité des recommandations proposées. En complément, nous mènerons une étude de cas sur six sujets afin d'avoir le feedback des utilisateurs sur notre système de recommandation.

Contributions

- Mohamed Nader Jelassi : *Une nouvelle approche pour l'extraction efficace des quadri-concepts fréquents*. Rencontre des Jeunes Chercheurs en Intelligence Artificelle (RJCIA) 2014 : 146-151.
- Mohamed Nader Jelassi, Sadok Ben Yahia et Engelbert Mephu Nguifo : *A scalable mining of frequent quadratic concepts in d-folksonomies*. ArXiv e-prints (Dec 2012). *http ://adsabs.harvard.edu/abs/2012arXiv1212.0087N*

CHAPITRE 4

Un système personnalisé de recommandation dans les *folksonomies*

4.1 Introduction

Dans le chapitre précédent, nous avons proposé un nouvel algorithme pour l'extraction des quadri-concepts à partir des *v-folksonomies*. Dans ce chapitre, nous allons nous appuyer sur ces quadri-concepts afin de proposer notre nouveau système personnalisé de recommandation. En effet, les quadri-concepts sont des structures quadratiques mettant en jeu des utilisateurs, tags, ressources et variables supplémentaires. Ainsi, en plus du triplet <utilisateur, tag, ressource>, nous allons utiliser des informations personnelles sur les utilisateurs afin de générer des recommandations qui répondent au mieux aux besoins des utilisateurs dans les *v-folksonomies*. Par ailleurs, nous introduisons une mesure de *ranking* afin de mieux classer et d'améliorer la qualité des recommandations. Enfin, après avoir étudié la précision et le rappel de nos recommandations, nous passerons en revue les différentes propriétés de notre système de recommandation, comme la scalabilité, la couverture, l'adaptivité, la sérendipité ou encore la diversité des recommandations proposées. En complément, nous mènerons une étude de cas sur six utilisateurs afin de recueillir leurs impressions sur notre système de recommandation.

Motivations : le profil comme variable supplémentaire

Les utilisateurs sont les acteurs principaux dans une *folksonomie* étant donné qu'ils contribuent au contenu par l'ajout de ressources et l'affectation de tags : ils sont donc

considérés comme les créateurs de l'information. La participation massive et croissante des utilisateurs dans les *folksonomies* revient au fait que la participation au contenu ne nécessite aucune connaissance spécifique [Jäschke *et al.*, 2008], *i.e.,* chacun est capable de contribuer au contenu sans beaucoup d'efforts donnant le plein pouvoir aux utilisateurs, acteurs principaux de la *folksonomie*. Cependant, il s'avère que le choix de tags et de ressources partagées par un utilisateur d'une *folksonomie* dépendait de son profil : le genre, l'âge, la profession, etc. [Michlmayr et Cayzer, 2007]. Cette diversité culturelle résultante des *folksonomies* est tout à fait fascinante. Si cette diversité peut être considérée comme un point fort des *folksonomies*, cela peut également être vu comme un point faible. Ainsi, les *folksonomies* doivent tenir compte de telles informations lors de la recommandation de tags ou de ressources. Cela a incité les chercheurs à proposer des systèmes de recommandation personnalisés afin de suggérer les tags et ressources les plus appropriés aux utilisateurs et de répondre aux besoins spécifiques de chaque utilisateur. Pour illustrer cela, considérons l'utilisateur *Mike*, un étudiant appartenant à la catégorie d'âge des "18-25 ans", notre objectif est de lui suggérer des tags et ressources les plus partagés par les utilisateurs au même profil (étudiants de la même catégorie d'âge), *i.e.,* utilisant le même vocabulaire (tags) et intéressés par les mêmes ressources dans la *folksonomie*. Par ailleurs, nous pouvons lui proposer des utilisateurs (amis) ayant un profil et des intérêts équivalents, *i.e.,* s'intéressant aux mêmes tags et ressources.

Pour atteindre cet objectif, nous considérons le profil des utilisateurs comme une quatrième dimension d'une *folksonomie*, classiquement composée de trois dimensions (utilisateurs, tags et ressources), et nous proposons une approche de regroupement des utilisateurs au profil et intérêts équivalents sous forme de quadri-concepts. Par suite, nous pouvons proposer un système personnalisé de recommandation de tags et de ressources selon chaque profil d'utilisateur. En outre, nous sommes en mesure de suggérer une liste d'utilisateurs (amis) au même profil et aux intérêts équivalents [Liang, 2010].

Pourquoi personnaliser les recommandations ?

Le domaine de personnalisation tente de fournir des solutions afin d'aider les utilisateurs à partager les bons tags et les bonnes ressources parmi le très grand nombre de données dans les *v-folksonomies*. Ainsi, un système de recommandation offre à l'utilisa-

teur une liste de tags ou de ressources recommandés qu'il est susceptible d'aimer et lui permet de trouver plus facilement ses tags et ressources préférés dans la *v-folksonomie* [Ricci *et al.*, 2011]. De plus, la personnalisation tente d'aider les utilisateurs à aborder le problème de surcharge d'information [Bollen *et al.*, 2010] [Das *et al.*, 2012] en éliminant la redondance des items et en retournant aux utilisateurs un ensemble concis et spécifique répondant à leurs besoins [Kim et Chan, 2003]. Et pour réussir ou tenter de répondre au mieux aux attentes de chaque utilisateur de la *v-folksonomie*, il est utile d'avoir plus d'informations sur lui. En effet, son âge, sa profession ou sa localisation sont des informations qui sont susceptibles de nous aider dans le processus de personnalisation de recommandation.

Pourquoi les quadri-concepts ?

Une autre question se pose alors : pourquoi les quadri-concepts ? D'un côté, si on peut facilement étudier les tags utilisés par un seul utilisateur sur une ressource, il est évident de constater que la tâche devient rapidement intraitable lorsque cela implique plusieurs utilisateurs et plusieurs ressources. D'un autre côté, les tags (ou ressources) recommandés s'avèrent ne pas être très spécifiques [Jäschke *et al.*, 2007], *i.e.,* des tags qui sont des mots "bateau" (au sens vague) ou bien des ressources ne correspondant pas aux besoins spécifiques de l'utilisateur. Grâce aux quadri-concepts, nous pouvons résoudre ces deux problèmes. En effet, d'un côté, les quadri-concepts sont des structures regroupant les tags et ressources en commun à un ensemble maximal d'utilisateurs. D'un autre côté, dans un quadri-concept, les tags et ressources qui ont été utilisés en combinaison seront regroupés d'où un résultat plus spécifique et répondant au besoin de notre système de recommandation. Ces concepts sont une représentation réduite de la *v-folksonomie* qui peut contenir des milliers de quadruplets en pratique. Un exemple de quadri-concept serait : *"Jack et Kate qui sont âgés entre* 18 *et* 25 *ans utilisent les tags 'action' et 'aventure', parmi d'autres, pour annoter des films comme 'Indiana Jones' et 'Star Wars'"*. Une fois extraits, ces quadri-concepts sont utilisés par notre algorithme de recommandation personnalisée multi-mode (utilisateurs, tags et ressources).

4.2 Le Pseudo code de PersoRec

À présent, nous introduisons notre nouveau système personnalisé de recommandation intitulé PersoRec qui s'appuie sur un ensemble de quadri-concepts fréquents extrait à partir d'une *v-folksonomie* donnée. Il est important de noter que, même si l'étape d'extraction des quadri-concepts est une phase qui peut avoir une complexité exponentielle, elle se passe hors-ligne et n'est exécutée qu'une seule fois. En effet, l'algorithme PersoRec parcourt des quadri-concepts déjà extraits. Ainsi, notre système de recommandation ne souffre pas du coût d'extraction des quadri-concepts à chaque recommandation. Cette phase peut donc être vue comme un pré-traitement à la phase de recommandation. PersoRec prend en entrée l'ensemble des quadri-concepts fréquents $\mathcal{QC}$ générés à partir d'une *v-folksonomie*, un utilisateur u avec sa variable v et (optionnellement) une ressource r (à annoter) et donne en sortie trois différents ensembles correspondant à trois tâches de recommandations possibles : un ensemble d'utilisateurs proposés, un ensemble de tags suggérés et un ensemble de ressources recommandées.

PersoRec opère comme suit : dans la Ligne 3, il parcourt l'ensemble des quadri-concepts fréquents en cherchant ceux dont les utilisateurs sont proches de u selon la variable v (*cf.,* Chapitre 3). Le test de la Ligne 4 permet de filtrer les tags et ressources déjà partagés par l'utilisateur u ; cette stratégie est inspirée par celle de [Lipczak, 2008]. Ensuite, pour chaque tâche, PersoRec fonctionne comme suit : pour la tâche de *Proposition d'utilisateurs* (Ligne 6), c'est la partie *utilisateurs* du quadri-concept qc qui est ajoutée à l'ensemble $\mathcal{PU}$ des utilisateurs proposés. Cette tâche aide à connecter les utilisateurs qui ont des intérêts communs et aide également à promouvoir le partage de ressources. Pour la tâche de *Suggestion de tags* (Lignes 8 et 9), le but est de suggérer des tags personnalisés à un utilisateur qui souhaite ajouter une ressource à la *v-folksonomie*. Cette tâche a plusieurs avantages : elle rappelle à l'utilisateur ce dont une ressource s'agit, accroît l'annotation des ressources et permet de consolider le vocabulaire des utilisateurs [Ricci *et al.*, 2011]. Pour cette tâche, nous ajoutons donc les tags affectés à la ressource r par les utilisateurs proches de u à l'ensemble $\mathcal{ST}$. Quant à la tâche de *Recommandation de ressources* (Ligne 12), le but est de proposer une liste personnalisée de ressources conforme aux intérêts de l'utilisateur u ; ces ressources sont ajoutées à l'ensemble $\mathcal{RR}$.

Algorithme 4 : PERSOREC

Données : l'ensemble des quadri-concepts fréquents $\mathcal{QC}$, un utilisateur u avec sa variable v et une ressource r.

Résultats : l'ensemble d'utilisateurs proposés $\mathcal{PU}$, l'ensemble des tags suggérés $\mathcal{ST}$ et l'ensemble des ressources recommandées $\mathcal{RR}$.

```
1  début
2      pour chaque quadri-concept qc ∈ QC faire
3          si v ∈ qc.Variables alors
4              si u ∉ qc.Utilisateurs alors
5                  /*Proposition d'utilisateurs */
6                  PU = PU ∪ qc.extent ;
7                  /*Suggestion de Tags*/
8                  si r ∈ qc.Ressources alors
9                      ST = ST ∪ qc.Tags ;
10                 /*Recommandation de Ressources */
11                 RR = RR ∪ qc.Ressources ;
12     retourner (PU,ST,RR) ;
13 fin
```

Complexité théorique

Comme nous l'avons vu lors du Chapitre 3, la complexité théorique de la phase d'extraction des quadri-concepts fréquents à partir d'un contexte ($\mathcal{U}$, $\mathcal{T}$, $\mathcal{R}$, $\mathcal{V}$) (qui est ici la phase de pré-traitement de PERSOREC) est de l'ordre de $\mathcal{O}(2^n)$ avec $n = |\mathcal{T}| + |\mathcal{R}| + |\mathcal{V}|$. Il est à noter cependant que même si cette étape est coûteuse en temps et mémoire, elle intervient **hors-ligne** et n'est exécutée qu'une **seule** fois (*i.e.*, lorsque le système démarre). En effet, PERSOREC parcourt un ensemble de quadri-concepts fréquents déjà extrait auparavant par un algorithme dédié à cette tâche. Ainsi, notre système de recommandation ne souffre pas du coût d'extraction des quadri-concepts à chaque recommandation. Soit m le nombre de quadri-concepts fréquents. La complexité théorique de notre algorithme PERSOREC est donc de l'ordre de $\mathcal{O}(m)$ puisque PERSOREC parcourt l'ensemble des quadri-concepts fréquents une seule fois.

Dans ce qui suit, afin d'améliorer la qualité des recommandations, nous proposons

un nouvel algorithme de recommandation, *i.e.,* FolkRec.

4.3 Vers une meilleure qualité des recommandations

Dans le but d'améliorer la qualité des recommandations que nous avons proposé, nous introduisons, dans ce qui suit, un nouveau système personnalisé de recommandation qui se base sur une nouvelle mesure de ranking.

4.3.1 Motivations

Une analyse du système personnalisé de recommandation PersoRec introduit précédemment démontre qu'une minorité de nouveaux utilisateurs peuvent ne pas recevoir de recommandations. Ce problème est connu comme "le démarrage à froid" d'un système de recommandation et plus connu sous le nom de *Cold Start* [Ricci *et al.*, 2011]. Il s'agit de mesurer la performance du système vis-à-vis des nouveaux utilisateurs. Un utilisateur est considéré comme nouveau s'il n'a encore tagué aucune ressource. De plus, en prenant en considération les nouveaux utilisateurs dans les *v-folksonomies*, nous tentons d'améliorer la couverture de l'espace utilisateur, *i.e.,* la portion d'utilisateurs à qui le système de recommandation peut fournir des recommandations. Par ailleurs, nous incluons également un filtre sur les tags et les ressources afin d'élaguer les tags et ressources déjà partagés par les utilisateurs. Cela aura pour garantie de proposer aux utilisateurs des nouveaux items qu'il n'a pas encore partagé. Enfin, pour garantir une meilleure personnalisation et qualité des recommandations, nous avons été motivés d'introduire une nouvelle mesure de ranking afin de classer les recommandations. Cette mesure, que nous introduisons ci-après, prend en compte à la fois l'item recommandé ainsi que le profil de l'utilisateur à qui nous souhaitons proposer des recommandations.

4.3.2 La mesure de ranking

Dans le but d'améliorer la précision et le rappel des recommandations proposées dans la littérature, nous proposons un nouveau score de ranking afin de classer les différentes recommandations.

Ainsi, pour un jeu de données donné, les top-k recommandations consistent en une liste d'items classés par valeur de score décroissante [Deshpande et Karypis, 2004].

Dans ce qui suit, la fonction de score est définie pour la recommandation de ressource mais peut très bien être définie pour la recommandation de tags ou d'utilisateurs en changeant les variables de l'équation. Ainsi, pour générer une recommandation de ressource pour un utilisateur donné, nous calculons le ranking comme décrit ci-dessus, et nous restreignons les résultats aux top-k premiers résultats (avec les scores les plus élevés). La mesure de score (notée rec_score) correspondant à un profil v est défini comme suit :

$$rec\_score(r_i, v) = \frac{|u_i|}{|UU|} / \exists t_i \exists r_i \exists v_i, (u_i, t_i, r_i, v_i) \in \mathcal{F}_v \tag{4.1}$$

Donc, le score rec_score d'une ressource r_i correspondant à un profil v est le nombre d'utilisateurs uniques, ayant le même profil v (ou au moins une information de profil $v\_i \in v$), qui ont partagé cette ressource, divisé par le nombre total d'utilisateurs uniques dans l'ensemble des quadri-concepts fréquents (noté UU). Par exemple, si une ressource r_1 a été partagée par 7 différents utilisateurs (au même profil) parmi une liste de 67 utilisateurs uniques, son score sera égal à 0.104 alors qu'une autre ressource r_2 partagée par 16 différents utilisateurs (au même profil) parmi la même liste aura un score égal à 0.238.

Dans ce qui suit, nous présentons le pseudo code de notre algorithme personnalisé de recommandation, *i.e.,* FolkRec.

4.3.3 Pseudo code de FolkRec

FolkRec, dont le pseudo code est donné par l'algorithme 5, prend les mêmes entrées que PersoRec et génére les mêmes ensembles en sortie. Cependant, il diffère dans son mode d'emploi. En effet, FolkRec opère comme suit : il parcourt l'ensemble des quadri-concepts fréquents dont les utilisateurs ont la même variable en commun que u (Lignes 6 et 16). Cependant, le mode d'emploi de FolkRec diffère en fonction du statut de u (ancien ou nouvel utilisateur). Si u est un ancien utilisateur, alors nous calculons l'ensemble de tags et l'ensemble de ressources qu'il a déjà partagés (Lignes 4-5). Ainsi, si un utilisateur, tag ou ressource appartient au quadri-concept qc, alors qc n'est pas pris en compte (Ligne 7) afin de filtrer les informations déjà connues par u. Cette stratégie de filtrage a été inspirée par celle de [Lipczak, 2008]. Par contre, si u est un nouvel utilisateur (lignes 19-30), il n'y a pas besoin de filtrage étant donné

Algorithme 5 : FOLKREC

Données : l'ensemble des quadri-concepts fréquents $\mathcal{QC}$, un utilisateur u avec sa variable v et une ressource r.

Résultats : Les ensembles $\mathcal{PU}$, $\mathcal{ST}$ et $\mathcal{RR}$.

```
1  début
2      pour chaque quadri-concept qc ∈ QC faire
3          si u est un ancien utilisateur alors
4              u.Tags={t ∈ T / ∃ r ∈ R, (u,t,r,v) est un quadri-concept} ;
5              u.Resources={r ∈ R / ∃ t ∈ T, (u,t,r,v) est un quadri-concept} ;
6              si v ∈ qc.Variable alors
7                  si u ∉ qc.Extent alors
8                      /*Proposition d'utilisateurs*/
9                      PU = PU ∪ qc.extent ;
10                     /*Suggestion de Tags*/
11                     si r ∈ qc.Intent alors
12                         ST = ST ∪ qc.modus \ u.Tags ;
13                     /*Recommandation de Ressources*/
14                     RR = RR ∪ qc.Intent \ u.Resources ;
15         si u est un nouvel utilisateur alors
16             si v ∈ qc.Variable alors
17                 /*Proposition d'utilisateurs*/
18                 PU = PU ∪ qc.extent ;
19                 /*Suggestion de Tags*/
20                 si r ∈ qc.Intent alors
21                     ST = ST ∪ qc.modus ;
22                 /*Recommandation de Ressources*/
23                 RR = RR ∪ qc.Intent ;
24     retourner (PU,ST,RR) ;
25 fin
```

que u n'a encore rien partagé. Ainsi, la seule information dont nous avons besoin est contenue dans la variable v relative à cet utilisateur. Donc, FolkRec recommande à u les tags et ressources partagés par des utilisateurs ayant en commun la variable v.

4.4 Étude expérimentale

Dans ce qui suit, nous évaluons notre approche sur deux jeux de données du monde réel : MovieLens et BookCrossing. Nous allons donner quelques exemples intéressants de recommandations puis nous évaluerons la qualité de ces recommandations, *i.e.,* la précision et le rappel de ces dernières. Puis, nous passerons en revue différentes propriétés et métriques de qualité avant de proposer notre étude de cas [Jawaheer *et al.*, 2014].

4.4.1 Jeux de données

Les deux jeux de données du monde réel utilisés pour notre évaluation sont décrits comme suit :

- Le jeu de données de filmographie MovieLens [1] : il s'agit d'un système de recommandation et d'un siteweb de communauté virtuelle qui permet aux utilisateurs de partager des films via des tags. Le jeu de données utilisé pour notre évaluation est téléchargeable gratuitement [2] et contient 15227 tags appliqués à 11272 films par 4010 utilisateurs (*e.g.,* <*Alex*, *X-Files*, *sciencefiction*>).
- Le jeu de données de livres BookCrossing [3] : c'est un "club de lecture" gratuit en ligne crée dans le but d'encourager la pratique de lecture et de faire du "monde entier une bibliothèque". Contrairement à MovieLens, le jeu de données BookCrossing ne s'appuie pas sur des tags pour annoter les ressources (les livres) mais des **notes**. En effet, les utilisateurs sont invités à noter les livres qu'ils partagent en choisissant une note entre 1 to 10, *i.e.,* plus grande est la valeur, plus grande est l'appréciation du livre. Le jeu de données utilisé pour notre évaluation est téléchargeable gratuitement [4] et contient 278858 utilisateurs qui ont produit

1. *http ://movielens.umn.edu/*
2. *http ://www.grouplens.org/node/73*
3. *http ://www.bookcrossing.com/*
4. *http ://www.grouplens.org/node/74*

1149780 notes sur 271379 livres (*e.g.*, <*Spender*, *DaVinciCode*, *9*>).

Nous rappelons que la quatrième dimension peut être considérée de manière indifférente mais pour les besoins de comparaison avec les approches de la littérature et aussi afin d'avoir des résultats concrets sur les jeux de données pris en considérations, nous avons choisi, dans ce qui suit, le **profil des utilisateurs** pour modéliser la variable v. Ainsi, nous considérons à présent que deux utilisateurs sont *proches* s'ils partagent au moins une même information de profil en commun (*e.g.,* un même âge, une même profession, etc.). À cet effet, des informations supplémentaires sur les utilisateurs sont disponibles dans les deux jeux de données et forment le profil des utilisateurs (la quatrième dimension d'une *v-folksonomie*) et qui renseigne, pour MovieLens, sur le **genre** de l'utilisateur (masculin ou féminin) et sa **profession** (au nombre de 21, qui peut être éducateur, écrivain, étudiant, scientifique, etc.). Dans le jeu de données BookCrossing, l'information sur le genre est remplacée par une information sur la ville de résidence de l'utilisateur (*e.g.*, France, Japon, Canada). Par ailleurs, les deux jeux de données fournissent des informations sur l'**âge** des utilisateurs qui est divisé en cinq tranches : (*i*) $7-18$ ans ; (*ii*) $19-24$ ans ; (*iii*) $25-35$ ans ; (*iv*) $36-45$ ans et (*v*) $46-73$ ans. Les Tableaux 4.1 et 4.2 montrent quelques exemples de quadruplets pour les deux jeux de données MovieLens et BookCrossing.

Utilisateur	Tag	Ressource	Profil
Mulder	*action*	*X-Files*	*student*
Mulder	*sciencefiction*	*X-Files*	25 *years old*
Scully	*adventure*	*Jurassic Park*	*professor*
Scully	*bestmovie*	*Jurassic Park*	*female*
Skinner	*thriller*	*Carrie*	*Canada*
⋮	⋮	⋮	⋮

Tableau 4.1 — Un aperçu du jeu de données MovieLens.

Nous avons vu, dans le précédent chapitre, quelques exemples de quadri-concepts fréquents extraits à partir du jeu de données MovieLens (*cf.,* Tableau 4.4). En ce qui concerne le jeu de données BookCrossing, le Tableau 4.3 démontre quelques exemples de quadri-concepts parmi les 18756 vérifiant les seuils minimaux de support définis comme suit : $minsupp_u = 2$, $minsupp_t = 2$, $minsupp_r = 2$ et $minsupp_p = 1$. Par exemple, dans le premier quadri-concept, deux utilisateurs qui viennent de Toronto et qui sont âgés entre 36 et 45 ans, ont partagé les livres *The Chicago Manual of Style* et *The Man Who Counts* avec une note égale à 8. Un autre quadri-concept montre que

Utilisateur	Tag	Ressource	Profil
Jack	*10*	*Mohammed : The Prophet of Islam*	*doctor*
Kate	*8*	*I got you under my skin*	*artist*
Kate	*8*	*I got you under my skin*	38 *years old*
Locke	*9*	*Mouth of Madness*	52 *years old*
Reyes	*9*	*Mouth of Madness*	*Spain*
⋮	⋮	⋮	⋮

Tableau 4.2 — Un aperçu du jeu de données BookCrossing.

deux autres utilisateurs, âgés entre 25 et 35 ans et vivant à Paris, sont intéressés par la franchise *Le Cycle d'Ender* ainsi que par la franchise *Les Amants Maudits*.

4.4.2 Quelles informations de profil améliorent le plus la recommandation ?

Dans ce qui suit, nous nous proposons de comparer les valeurs de précision de notre système de recommandation dans les cas où nous prenons en compte des informations supplémentaires sur les utilisateurs (contenues dans la quatrième dimension) et dans le cas contraire. Ainsi, nous essayons de répondre aux questions suivantes : *(i)* est-ce que la nouvelle dimension introduite représente une importante source d'information qui aide à améliorer les recommandations dans les *folksonomies* ? *(ii)* jusqu'à quel point la nouvelle dimension introduite permet d'améliorer la qualité des recommandations, *i.e.,* combien d'informations de profil sont nécessaires pour avoir la meilleure qualité de recommandation ? *(iii)* et quelles informations de profil sont les plus influentes durant le processus de recommandation. Pour répondre à toutes ces questions, nous proposons dans la définition suivante d'introduire une mesure de proximité entre utilisateurs.

U	{Regina, Rumple}
T	{8}
R	{The Chicago Manual of Style
	The Man Who Counts}
V	{36-45 years, Toronto}
U	{Caphook,Madhat}
T	{10}
R	{Harry Potter and the Prisoner of Azkaban
	Tough Enough (Pokemon Chapter Book, 27)}
V	{7-18 years, Chicago}
U	{shepard,austeen}
T	{10}
R	{Die Weiss Lowin / Contemporary German Lit
	Mittsommermord. Roman.}
V	{36-45 years, Hambourg}
U	{Allison15,Buffay}
T	{9}
R	{The Eye of the World (The Wheel of Time, Book 1)
	Dragons of Autumn Twilight (Vol 1)}
V	{18-25 years, Lisbon}
U	{Maryc,Kingg90}
T	{8}
R	{Le Cycle d'Ender, tome 4 : Les Enfants de l'esprit
	Le Cycle d'Ender, tome 3 : Xénocide
	Les amants maudits (10/18)}
V	{25-35 years, Paris}
U	{Brson,Manblack}
T	{10}
R	{Full Time : the Secret Life of Tony Cascarino
	Special Needs Education : Children...
	The Rasputin File}
V	{18-25 years, Hong Kong}

Tableau 4.3 — Exemples de quadri-concepts extraits à partir du jeu de données BOOKCROSSING.

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}

Tableau 4.4 — Exemples de quadri-concepts extraits à partir du jeu de données MovieLens.

Information de Profil / k	6	7	8	9	10	**Précision Moyenne**	**Précision Maximale**	**Précision Minimale**	**Variance**	**Écart Type**
Degré de proximité = 0										
None	0.56	0.54	0.51	0.48	0.48	0.514	0.48	0.56	0.000922	0.030364
Degré de proximité = 1										
Age	0.60	0.57	0.54	0.51	0.50	0.544	0.50	0.60	0.001447	0.038042
Localisation	0.72	0.73	0.75	0.74	0.71	0.730	0.71	0.75	0.000200	0.014142
Profession	0.55	0.50	0.51	0.50	0.50	0.512	0.50	0.55	0.000260	0.016149
Degré de proximité = 2										
Age + Localisation	0.52	0.52	0.52	0.51	0.51	0.516	0.51	0.52	0.000019	0.004358
Profession + Localisation	0.53	0.51	0.50	0.44	0.42	0.480	0.42	0.53	0.001800	0.042426
Age + Profession	0.63	0.64	0.63	0.64	0.67	0.642	0.63	0.67	0.0002416	0.0155434
Degré de proximité = 3										
Age + Profession + Localisation	0.50	0.42	0.37	0.33	0.30	0.384	0.30	0.50	0.004938	0.070270

Tableau 4.5 — Valeurs de précision des recommandations pour différents degrés de proximité pour le jeu de données MovieLens.

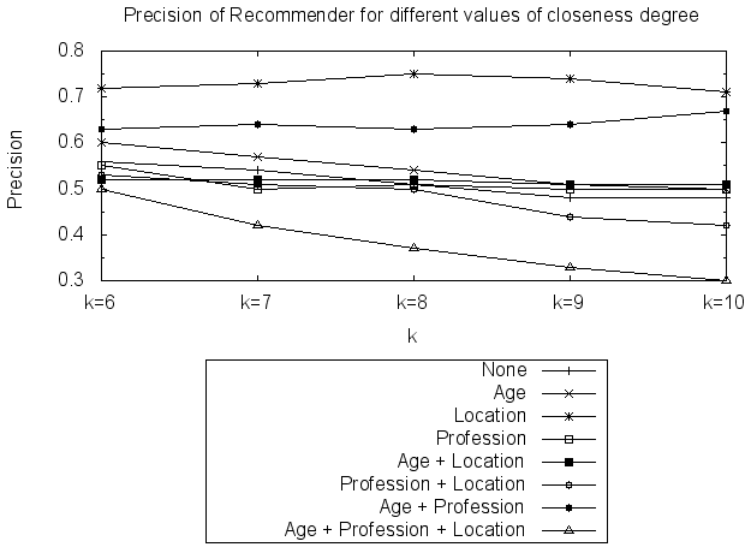


Figure 4.1 — Valeurs de précision des recommandations pour différents degrés de proximité.

Définition 15 *(*Degré de proximité*) Considérons une v-folksonomie* $\mathcal{F}_v = (\mathcal{U}, \mathcal{T}, \mathcal{R}, \mathcal{V}, \mathrm{Y})$*, nous définissons le* ***degré de proximité*** *entre deux utilisateurs de* $\mathcal{U}$ *comme le nombre de leurs variables de* $\mathcal{V}$ *en commun.*

Exemple 10 *Considérons la v-folksonomie représentée par le Tableau 5.1, le quadri-concept* $(\{u_1, u_2, u_3\}, \{t_2, t_3, t_4\}, \{r_1, r_2, r_3\}, v_1)$ *démontre que les utilisateurs* u_1*,* u_2 *et* u_3 *ont un degré de proximité égal à* 1*, i.e., ils partagent la variable* v_1 *en commun. Par ailleurs, le quadri-concept* $(\{u_1, u_2\}, \{t_2, t_3, t_4\}, \{r_1, r_3\}, \{v_1, v_2\})$ *nous renseigne que les utilisateur* u_1 *et* u_2 *partagent deux variables en commun,* i.e., v_1 *et* v_2*. Ainsi, ils ont un degré de proximité égal à* 2*. Par exemple,* u_1 *et* u_2 *peuvent avoir le même age et la même profession.*

Le Tableau 4.5 et la Figure 4.1 montrent les valeurs de précision des recommandations obtenues par notre système de recommandation pour différents degrés de proximité et différentes valeurs de k[5] allant de 6 à 10 sur le jeu de données MovieLens. Tout d'abord, les résultats démontrent l'utilité de la quatrième dimension, *i.e.,* le profil, durant le processus de recommandation. En effet, les autre meilleurs scores de précision sont atteints lorsque notre système de recommandation prend le profil des utilisateurs

5. le nombre de recommandations retournées à l'utilisateur.

comme information supplémentaire. Ainsi, le recours aux informations supplémentaires sur les utilisateurs permet de personnaliser les recommandations et de générer des recommandations plus ciblées. Étonnamment, l'information de profil la plus influente est la **localisation** des utilisateurs. En effet, plus les utilisateurs sont proches géographiquement, plus ils ont tendance à partager les mêmes ressources et à avoir le même comportement social selon les traditions culturelles de leurs pays respectifs. Par exemple, les utilisateurs indiens partagent les films de Bollywood tandis que les utilisateurs japonais ont tendance à partager en masse les mangas. Ensuite, la seconde information de profil la plus influente pour la recommandation est l'**âge** des utilisateurs. En effet, les utilisateurs appartenant à la même catégorie d'âge convergent vers un vocabulaire commun (les jeunes utilisateurs contre les anciens utilisateurs) et partagent le même type de ressources, *e.g.,* les jeunes utilisateurs préfèrent les films d'actions, les plus jeunes partagent les mangas alors que les plus anciens ont tendance à partager les films classiques. Par ailleurs, lorsque nous associons deux informations de profil, l'âge apparaît comme étant l'information de profil la plus importante, notamment lorsqu'elle est associée à la profession ou la localisation. En effet, la précision de notre système de recommandation augmente sensiblement lorsque nous prenons en compte à la fois l'âge et la profession comme informations supplémentaires étant donné que les utilisateurs d'une même catégorie d'âge et exerçant le même métier ont un profil assez proche, *e.g.*, des étudiants de [19-24] ans ou encore des techniciens de [25-35] ans.

Toutefois, la précision de nos recommandations atteint ses moins bons résultats lorsque nous combinons toutes les informations de profil. Si la localisation ou la combinaison âge-profession produit de bons résultats en termes de précision, les combiner réduit considérablement la qualité des recommandation. En effet, le nombre de recommandations décroît étant donné qu'il est rare de trouver des utilisateurs ayant à la fois la même profession, le même âge, la même localisation et partageant les mêmes ressources. Comme il es rare de retrouver des utilisateurs ayant ce même genre de profil, les ressources recommandées ont moins de chance d'être pertinentes. Par ailleurs, si l'âge ou la localisation donnent de bons résultats en termes de précision, cela n'est pas le cas pour la profession qui n'est pas une information influente pour nos recommandations. Les utilisateurs exerçant le même métier ne partagent pas forcément les mêmes intérêts. Enfin, lorsque notre système de recommandation ne prend aucune information supplémentaire sur les utilisateurs, la précision décroît rapidement puisque la liste de recommandation est aléatoire, c'est-à-dire, pas personnalisée. Ainsi, dans le

cas des *v-folksonomies*, plus le nombre de ressources recommandées augmente, moins elles sont pertinentes. En effet, les ressources les plus partagés par le passé ne sont pas nécessairement partagés dans le futur, donc, le score de précision n'est pas élevé lorsqu'aucune information supplémentaire sur les utilisateurs n'est prise en compte.

Nous concluons que prendre en compte des informations supplémentaires sur les utilisateurs permet d'augmenter la précision des recommandations, et pour avoir les meilleurs résultats, il est préférable de s'arrêter à une ou deux informations de profil. Ainsi, si nous prenons une seule information de profil, la localisation et l'âge sont les informations de profil qui donnent les meilleurs scores. Par contre, si nous combinons deux différentes variables, il est conseillé de combiner l'âge avec une autre information de profil.

4.4.3 Exemples de recommandations personnalisées

Les quadri-concepts ont permis de regrouper sous un même concept des utilisateurs au profil équivalent qui partagent des tags et ressources en commun. Les quadri-concepts peuvent être appliqués dans plusieurs domaines dont la suggestion de tags, la recommandation de ressources ou la proposition d'amis grâce à l'algorithme FolkRec.

Suggestion de tags Par exemple, considérons l'utilisateur *rossy* qui souhaite partager le film *Rear Window* : puisque *rossy* est âgé entre 36 et 45 ans, il aurait la possibilité d'utiliser les tags *classic*, *oldmovie* et *quotes*, par contre, s'il était âgé, par exemple, de 60 ans, nous lui aurions proposé les tags *classic*, *dialog* et *oscar*. De même, selon sa profession, les tags suggérés seront différents pour correspondre au mieux à son vocabulaire.

Recommandation de ressources Dans MovieLens, cela semble trivial de dire qu'un jeune utilisateur sera plus intéressé par les comédies que par des films classiques ; de même, il serait plus judicieux de recommander des films romantiques à une femme qu'un film de guerre. Dans notre cas, considérons deux nouveaux utilisateurs de la *v-folksonmie reyes* (51 ans) et *zlatan* (26 ans). Contrairement aux systèmes de recommandation classiques, il sera possible grâce à notre système de recommander des films à ces nouveaux utilisateurs en dépit de l'absence d'historique de tagging. Ainsi, nous proposerons les films *Star Wars*, *M.A.S.H* et *Rear Window* à *reyes* tandis que les films *Usual Suspects*, *Silence of the Lambs* et *Sound of Music*, plus susceptibles de plaire à *zlatan*, seront recommandés à ce dernier.

Le choix inverse aurait été beaucoup moins pertinent. Par ailleurs, dans le jeu de données BookCrossing, les utilisateurs issus de différents pays dans le monde s'intéressent à des livres différents, selon leurs cultures et traditions. Considérons deux nouveaux utilisateurs *Malcolm* (28 ans, de Paris) et *Reese* (17 ans, de Chicago) : notre système recommandera les livres *Le Cycle d'Ender* et *Les amants maudits* pour *Malcolm*, qui sont des livres populaires en France. Par contre, des bouquins comme *Harry Potter* et *Pokemon*, très en vogue chez les jeunes personnes seront recommandés à *Reese*, ce qui correspond mieux à ses intérêts. Puisque ces deux utilisateurs sont nouveaux dans la *v-folksonomie*, nous n'avons pas encore d'information sur leurs historiques de tagging. Ainsi, nos recommandations pour ces deux utilisateurs sont uniquement basées sur leur profil (*i.e.,* leur âges et localisations dans ce cas).

Proposition d'amis Considérons l'utilisateur *krycek* qui est médecin (*cf.,* Tableau 4.4), nous pourrons lui recommander comme amis les utilisateurs *mulder* et *scully* puisqu'ils ont des profils et intérêts équivalents, *i.e.,* tous les trois médecins et appartenant à la même tranche d'âge, en plus d'être intéressés par les mêmes tags et ressources.

4.4.4 Exemple illustratif

Considérons l'ensemble des quadri-concepts fréquents issus des deux jeux de données illustrés par les Tableaux 4.3 et 4.4 et soient u_1=*Jacob* (37 ans, écrivain) et u_2=*Ilana* (Femme, 63 ans) deux nouveaux utilisateurs de MovieLens et u_3=*Sheldon* (Homme, 26 ans, Paris) un nouvel utilisateur de BookCrossing. En premier lieu, supposons que *Jacob* et *Ilana* veulent tous les deux partager le film *Rear Window* (*fenêtre sur cour*). Grâce à notre système personnalisé de recommandation, nous sommes en mesure de fournir deux types de recommandations suivant le profil de chacun de ces deux utilisateurs. Ainsi, *Jacob* aura comme tags suggérés pour ce film : *classic*, *quotes* et *oldmovie* alors que les tags *classic*, *dialog* et *oscar* seront suggérés à *Ilana* pour annoter le même film (*cf.,* Tableau 4.4). Par ailleurs, nous pouvons proposer comme amis les utilisateurs *Ross*, *Anlucia* et *Franela* à *Jacob* étant donné qu'ils ont des profils et intérêts proches, alors qu'il semble plus judicieux de proposer les utilisateurs *Saloua*, *Wafa* et *Yasmine* à *Ilana* car ils ont presque le même âge et partagent les mêmes intérêts. En second lieu, concernant l'utilisateur *Sheldon*, qui est nouveau dans le jeu de données

BookCrossing, FolkRec cherche des quadri-concepts fréquents qui correspondent à son profil, *i.e.,* un homme de 26 ans vivant à Paris. Ainsi, d'après le Tableau 4.3, nous pouvons lui recommander les livres *Le Cycle d'Ender, tome 4 : Les Enfants de l'esprit* et *Le Cycle d'Ender, tome 3 : Xénocide, Les amants maudits (10/18)* car ces deux livres ont été partagés par des utilisateurs ayant un même profil que *Sheldon*, *i.e.,* des parisiens de la même catégorie d'âge.

4.4.5 Précision, Rappel et F1-score

Base d'apprentissage/Base de test et protocole de validation

Pour nos expérimentations, nous avons utilisé le protocole de validation "5-validation croisée" (ou 5-fold cross-validation) [Weiss et Kulikowski, 1991] afin d'évaluer la pertinence de notre approche. Chacun des jeux de données MovieLens et BookCrossing a été partitionné en deux échantillons : un échantillon aléatoire contenant 80% des utilisateurs a été utilisé comme **base d'apprentissage** et un échantillon aléatoire contenant les 20% d'utilisateurs restants, a été utilisé pour la validation de nos tests (*i.e.,* **base de test**). Pour chaque utilisateur du deuxième échantillon (*i.e.,* utilisateur test), 20% aléatoires de ses tags et ressources sont considérées comme ensemble de test/réponse et 80% comme son ensemble d'apprentissage. Nous avons répété cette expérience cinq fois en changeant à chaque fois les 20% représentant la base de test afin de couvrir les 100% de tout l'ensemble. Pour chaque utilisateur test, notre algorithme de recommandation génére une liste d'éléments (utilisateurs, tags ou ressources) en se basant sur son ensemble d'apprentissage. Si un élément de la liste de recommandation se trouve également dans l'ensemble de test de cet utilisateur, alors l'élément est considéré comme **pertinent**. Pour nos expérimentations, nous avons également fait varier le nombre de recommandations fournies à l'utilisateur : il s'agit des top-k recommandations. Grâce à ça, l'utilisateur peut spécifier les k recommandations les plus pertinentes que le système doit lui retourner. Ce sont celles dont les scores sont les plus élevés (*cf.*, Équation 4.1). Cela permet surtout d'éviter de submerger l'utilisateur par un grand nombre de réponses en lui retournant que le nombre de réponses pertinentes qu'il(elle) souhaite.

Approche	Jeux de données	Protocole de Validation	Pertinence	Score	Top-k
Qumsiyeh *et al.* [Qumsiyeh et Ng, 2012]	MovieLens, BookCrossing Yahoo !music et Netflix	5-validation croisée	La note est pertinente si elle est réellement affectée par l'utilisateur à la ressource	Une mesure de score basée sur les notes passées	Non
Kim *et al.* [Kim *et al.*, 2011]	BookCrossing (une représentation réduite)	Pas défini	La recommandation d'un livre est pertinente si le livre est sélectionné par l'utilisateur	Une mesure de similarité entre communautés d'utilisateurs	Oui
Bellogín *et al.* [Bellogín *et al.*, 2013]	MovieLens, Delicious, et LastFM	5-validation croisée	La recommandation est pertinente si elle appartient à la base de test de l'utilisateur cible	Une mesure de préférence et une mesure des items les plus utilisés par les amis	Oui
PersoRec [Jelassi *et al.*, 2013b]	MovieLens et BookCrossing	Holdout	La recommandation est pertinente si elle appartient à la base de test de l'utilisateur cible	Pas utilisé	Yes
FolkRec	MovieLens et BookCrossing	5-validation croisée	La recommandation est pertinente si elle appartient à la base de test de l'utilisateur cible	Une mesure de score basée sur le profil d'un utilisateur (Voir, Équation 4.1)	Oui

Tableau 4.6 — Les différences dans les méthodes abordées entre les différentes approches.

Évaluer l'efficacité d'un algorithme de recommandation est loin d'être trivial. Néanmoins, pour déterminer l'efficacité d'un système, nous pourrions appliquer les métriques classiques de recherche d'informations : la précision, le rappel et le F1-score [Baeza-Yates et Ribeiro-Neto, 1999] (équations **??**, **??** et **??**). Ces mesures représentent la qualité de la recommandation, c'est-à-dire à quel point les suggestions proposées sont conformes aux intérêts de l'utilisateur. Dans ce qui suit, nous nous intéressons à la tâche de recommandation de ressources et nous évaluons la précision de notre approche *vs.* les travaux **pionniers** qui ont un objectif commun avec la notre, *i.e.,* ceux de Bellogin *et al.* [Bellogín *et al.*, 2013], Qumsiyeh *et al.* [Qumsiyeh et Ng, 2012] et Kim *et al.* [Kim *et al.*, 2011]. Nous comparons également les résultats de FolkRec avec PersoRec. Par ailleurs, le Tableau 4.6 illustre quelques comparaisons entre les différentes approches et FolkRec. Il montre pour chaque approche, les jeux de données utilisés pour les expérimentations, le protocole de validation, la manière avec laquelle la pertinence des recommandations est définie par les auteurs, la mesure de ranking utilisée pour les tests et enfin si l'approche utilise ou non les recommandations top-k.

Précision

Les Tableaux 4.7 et 4.8 montrent les différentes valeurs de précision obtenues par notre algorithme de recommandation *vs.* ses concurrents pour différentes valeurs de k variant entre 6 et 10 sur les deux jeux de données. En général, nos recommandations pour les utilisateurs de MovieLens et BookCrossing répondent à leurs attentes. En effet, les recommandations sont pertinentes à 67% et 57%, respectivement, pour MovieLens et BookCrossing, ce qui surpasse, pour la plupart des cas, la précision de ses concurrents. Ainsi, pour le jeu de données MovieLens, notre score de précision est, respectivement, 91%, 168% et 86% meilleur que celui de Bellogin *et al.*, Qumsiyeh *et al.* et celui de PersoRec. Quant au jeu de données BookCrossing, notre précision est plus grande de, respectivement, 11%, 21% et 4% que celles de Kim *et al.*, Qumsiyeh *et al.* et PersoRec. Par ailleurs, les résultats ont montré que FolkRec atteint ses meilleures performances lorsque la valeur de k est égale à 6. Cela est dû au fait que les six premières recommandations correspondent aux besoins des utilisateurs et que lorsque le nombre de recommandations augmente, cela entraîne inévitablement une diminution de la précision étant donné que l'utilisateur choisit moins de ressources que celles qui lui sont recommandées. Quant à la différence entre notre précision et celle

des autres approches, nous l'expliquons par le fait que l'utilisation des quadri-concepts améliore les recommandations en suggérant les tags et ressources les plus proches des besoins des utilisateurs. En effet, alors que les travaux connexes se concentrent sur les éléments les plus items et les plus utilisés (livres, films, tags), les quadri-concepts offrent à nos utilisateurs, les tags et les ressources qui ont été partagés en commun par des utilisateurs aux profils proches. En effet, il s'avère que les utilisateurs ont tendance à partager des tags et des ressources déjà partagés par des utilisateurs ayant un même profil. Par ailleurs, nous avons également réussi à améliorer considérablement les valeurs de précision de PERSOREC pour le jeu de données MOVIELENS en améliorant les recommandations grâce au score de ranking que nous avons introduit ainsi qu'à quelques nouvelles fonctionnalités ajoutées à l'algorithme (considération de nouveaux utilisateurs, filtrage de ressources déjà partagées, etc.).

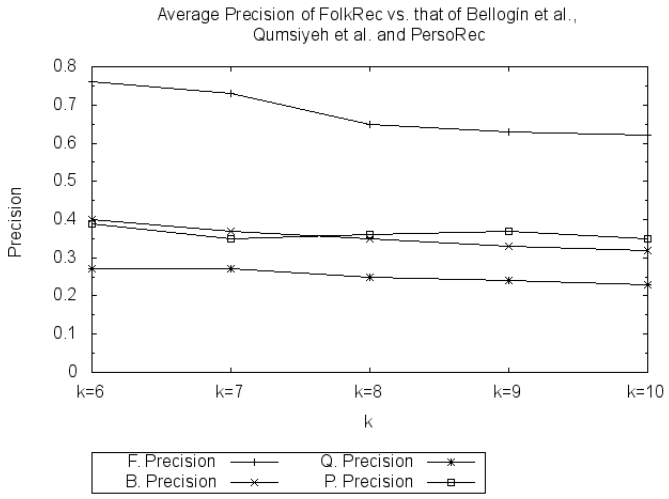


Figure 4.2 — Précision moyenne pour la recommandation de ressources sur MOVIELENS. (F) FOLKREC (B) Bellogìn *et al.* (Q) Qumsiyeh *et al.*(P) PERSOREC (*cf.*, Table 4.7)

k	FOLKREC	Bellogìn *et al.*	Qumsiyeh *et al.*	PERSOREC
6	**0, 76**	0, 40	0, 27	0, 39
7	**0, 73**	0, 37	0, 27	0, 35
8	**0, 65**	0, 35	0, 25	0, 36
9	**0, 63**	0, 33	0, 24	0, 37
10	**0, 62**	0, 32	0, 23	0, 35

Tableau 4.7 — Précision moyenne pour la recommandation de ressources sur MOVIELENS.

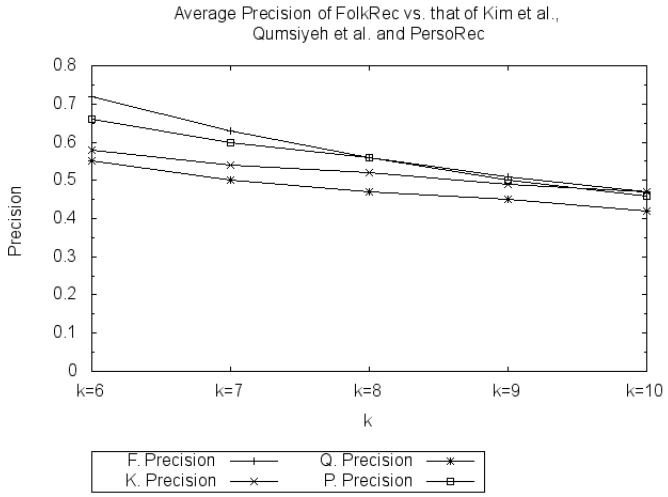


Figure 4.3 — Précision moyenne pour la recommandation de ressources sur BookCrossing. (F) FolkRec (K) Kim *et al.* (Q) Qumsiyeh *et al.*(P) PersoRec (*cf.*, Table 4.8

k	FolkRec	Kim *et al.*	Qumsiyeh *et al.*	PersoRec
6	**0, 72**	0, 58	0, 55	0, 66
7	**0, 63**	0, 54	0, 50	0, 60
8	**0, 56**	0, 52	0, 47	**0, 56**
9	**0, 51**	0, 49	0, 45	0, 50
10	**0, 47**	**0, 47**	0, 42	0, 46

Tableau 4.8 — Précision moyenne pour la recommandation de ressources sur BookCrossing.

Rappel

k	FolkRec	Bellogìn *et al.*	Qumsiyeh *et al.*	PersoRec
6	**0, 54**	0, 20	0, 09	0, 32
7	**0, 51**	0, 16	0, 08	0, 29
8	**0, 48**	0, 14	0, 07	0, 28
9	**0, 46**	0, 11	0, 06	0, 27
10	**0, 40**	0, 10	0, 06	0, 26

Tableau 4.9 — Rappel moyen pour la recommandation de ressources sur MovieLens.

Les Tableaux 4.9 et 4.10 démontrent les différentes valeurs de rappel obtenues par FolkRec *vs.* les autres approches de la littératures pour différentes valeurs de k allant de 6 à 10 sur les jeux de données MovieLens et BookCrossing. Les résultats montrent que notre algorithme surpasse nettement les approches de la littérature. En

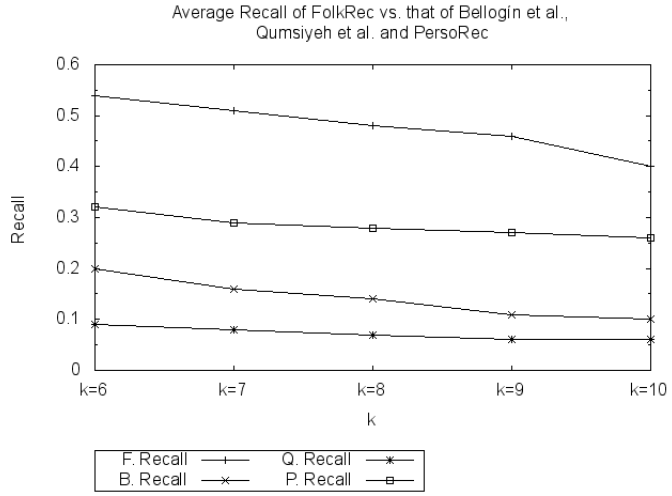


Figure 4.4 — Rappel moyen pour la recommandation de ressources sur MovieLens. (F) FolkRec (B) Bellogìn *et al.* (Q) Qumsiyeh *et al.*(P) PersoRec (*cf.*, Table 4.9)

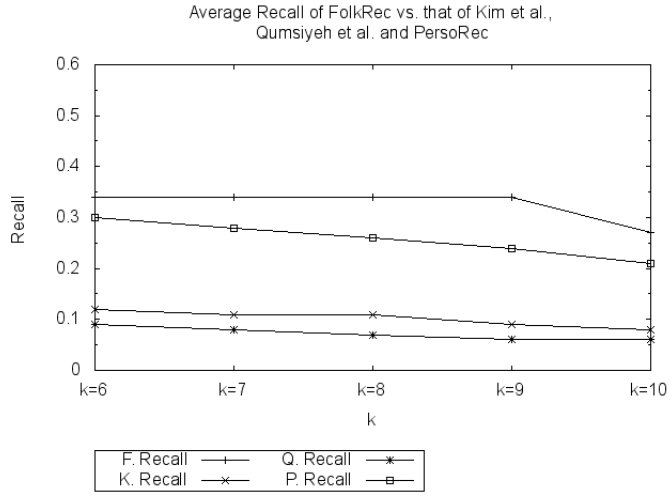


Figure 4.5 — Rappel moyen pour la recommandation de ressources sur BookCrossing. (F) FolkRec (K) Kim *et al.* (Q) Qumsiyeh *et al.*(P) PersoRec (*cf.*, Table 4.10)

k	FolkRec	Kim *et al.*	Qumsiyeh *et al.*	PersoRec
6	**0, 34**	0, 12	0, 09	0, 30
7	**0, 34**	0, 11	0, 08	0, 28
8	**0, 34**	0, 11	0, 07	0, 26
9	**0, 34**	0, 09	0, 06	0, 24
10	**0, 27**	0, 08	0, 06	0, 21

Tableau 4.10 — Rappel moyen pour la recommandation de ressources sur BookCrossing.

effet, FolkRec a une valeur moyenne de rappel égale à 47% sur MovieLens vs. 14% pour Bellogin *et al.*, 7% pour Qumsiyeh *et al.* et 28% pour le système de recommandation PersoRec. Quant au jeu de données BookCrossing, notre rappel est, respectivement, 219%, 352% et 26% meilleur que les valeurs de rappel respectives de Kim *et al.*, Qumsiyeh *et al.* et PersoRec. Cette différence de performances démontre que sur l'ensemble total des éléments pertinents, FolkRec est capable d'en recommander, à ses utilisateurs, une portion plus grande que ses concurrents. Grâce aux quadri-concepts, qui représentent des structures représentatives de la *v-folksonomie* et qui ciblent mieux les besoins des utilisateurs, FolkRec recommande donc des éléments partagés par des utilisateurs qui sont susceptibles d'être ensuite partagés par des utilisateurs avec un profil proche. De plus, la nouvelle stratégie prônée par FolkRec qui consiste à filtrer les tags et ressources déjà partagés par les utilisateurs et à prendre en considération les nouveaux utilisateurs a eu tendance à augmenter son rappel *vs.* les autres approches.

F1-Score

Les Tableaux 4.11 et 4.12 démontrent les différentes valeurs de *F1-Score* obtenues par FolkRec par rapport aux autres approches de la littérature pour différentes valeurs de k allant de 6 à 10 sur les jeux de données MovieLens et BookCrossing. Les résultats montrent très logiquement que FolkRec surpasse ses concurrents sur les deux jeux de données puisque les valeurs de rappel et de précision de FolkRec sont supérieures à celles des autres approches.

k	FolkRec	Bellogìn *et al.*	Qumsiyeh *et al.*	PersoRec
6	**0, 57**	0, 20	0, 09	0, 35
7	**0, 56**	0, 16	0, 08	0, 31
8	**0, 54**	0, 14	0, 07	0, 31
9	**0, 56**	0, 11	0, 06	0, 31
10	**0, 52**	0, 10	0, 06	0, 29

Tableau 4.11 — F1-Score moyen pour la recommandation de ressources sur MovieLens.

k	FolkRec	Kim *et al.*	Qumsiyeh *et al.*	PersoRec
6	**0.48**	0.20	0.15	0,41
7	**0.45**	0.18	0.14	0,38
8	**0.40**	0.18	0.13	0,35
9	**0.41**	0.15	0.12	0,32
10	**0.39**	0.13	0.12	0,28

Tableau 4.12 — F1-Score moyen pour la recommandation de ressources sur BookCrossing.

4.4.6 Évaluation Sociale

Dans ce qui suit, nous étudions l'évaluation sociale de notre système de recommandation. Nous analysons ce qui se passe **après** l'étape de recommandation, *i.e.,* si l'utilisateur cible a vraiment aimé les recommandations et si les utilisateurs (amis) qu'on lui a proposés adoptent le même comportement social. Pour ce faire, nous étudions trois différents cas réels de recommandation avec BookCrossing et un cas réel pour MovieLens. Ainsi, pour le premier jeu de données, nous avons choisi trois utilisateurs avec différents âges et pays : *skinner* (38 ans, New York, USA), *herge* (26 ans, Seixal, Portugal) et *benjamin* (15 ans, Texas, USA). En premier lieu, notre algorithme recommande au premier utilisateur trois différents livres de la franchise *Harry Potter* ainsi que quatre nouveaux amis : *ross* (43 ans, *Illinois*, *USA*), *fran* (54 ans, *California*, *USA*), *emma* (40 ans, *Oregon*, *USA*) et *annalucia* (36 ans, *Teheran*, *Iran*). Il s'est avéré plus tard que ces nouveaux amis ont également partagé tous les livres de la franchise *Harry Potter*. De plus, *skinner*, ainsi que ses amis recommandés, ont attribué aux livres recommandés la note moyenne de 9 ce qui démontre une réelle appréciation des recommandations. En second lieu, nous avons recommandé à *herge* trois différents livres (*Da Vinci Code*, *Wild Animus* et *The Joy Luck Club*) ainsi que quatre amis agés entre 25 et 35 ans : trois d'entre eux venant des *USA* (*Kansas*, *Wisconsin* et *Virginia*) et le quatrième du *Canada* (*Ottawa*). Cependant, bien qu'il ait partagé les trois livres, *herge* ne les a pas notés, et parmi ses "nouveaux amis", l'un d'eux fut vraiment intéressé par les mêmes livres. En dernier lieu, notre algorithme a généré pour *benjamin* deux livres (*Harry Potter and the Prisonnier of Azkaban* et *Harry Potter and Cup of Fire*) ainsi qu'un nouvel ami, *i.e., baefire* (12 ans, *Illinois*, *USA*). Il se trouve, qu'après la recommandation, les deux utilisateurs ont partagé ces livres en leur attribuant la note maximale (10) ce qui indique qu'ils ont vraiment bien apprécié les livres recommandés.

Quant au jeu de données MovieLens, notre utilisateur cible est *Bruce* (47 ans,

Homme, Educateur). FolkRec lui a recommandé quatre films : *Star Wars*, *The Return of the Jedi*, *God Father 1 and 2* ainsi que deux nouveaux utilisateurs : *Slioua* (49 ans, Femme, Educatrice) et *Nina* (49 ans, Homme, Educateur). Tout d'abord, nous pouvons remarquer que *Bruce* a apprécié les films recommandés en leur attribuant la note maximale de 5. Ensuite, nous avons noté que ses amis recommandés ont aussi partagé les mêmes films avec une note moyenne de 4 ce qui démontre que Bruce et ses nouveaux amis ont des intérêts communs pour les mêmes films.

Dans ce qui suit, nous proposons plusieurs propriétés afin d'évaluer les propriétés de notre système de recommandation. Ces métriques sont définies comme la capacité d'un système de recommandation à suggérer à l'utilisateurs des éléments pertinents mais non populaires [Ricci *et al.*, 2011] [Chojnacki et Klopotek, 2010].

Les propriétés de FolkRec

Bien qu'elle soit une tâche cruciale, la recommandation d'utilisateurs, de tags et de ressources s'avère parfois insuffisante pour déployer un bon système de recommandation. Souvent, les utilisateurs peuvent être intéressés par plus qu'une bonne recommandation : découvrir de nouveaux éléments, la diversité des éléments, etc. Ainsi, nous devons identifier l'ensemble de propriétés qui influent sur la réussite d'un système de recommandation [Ricci *et al.*, 2011] :

4.4.7 Couverture de l'espace

Le terme couverture fait référence à la portion de ressources qu'un système de recommandation peut recommander [Ricci *et al.*, 2011]. Une mesure simple consiste à calculer le pourcentage de ressources que le système peut recommander. Pour ce faire, nous avons calculé sur plusieurs *snapshots* des deux jeux de données MovieLens et BookCrossing le nombre de ressources (uniques) qui ont été recommandés par FolkRec, *i.e.,* le nombre de ressources uniques présentes dans tous les quadri-concepts. Puis, nous divisons ce nombre par celui du nombre total des ressources (uniques) présents dans chaque jeu de données. Pour le jeu de données *BookCrossing*, le nombre moyen de ressources uniques est égal à 19662 alors que le nombre de ressources recommandées est égale à 789. Cependant, pour avoir des résultats plus équitables, nous déduisons des 19662 ressources du jeu de données celles qui n'apparaissent qu'une seule

fois dans l'ensemble du jeu de données (13278 au total), *i.e.,* les ressources qui ont été partagés par un seul utilisateur, ce qui donne une couverture moyenne de $13,62\%$ sur BOOKCROSSING. Pour le jeu de données MOVIELENS, qui est plus dense, le nombre de ressources uniques est égal à 955 alors que le nombre de ressources recommandées est égal à 474, ce qui correspond à une couverture de $47,15\%$. De plus, si on déduit les ressources qui n'apparaissent qu'une seule fois dans l'ensemble du jeu de données, la couverture grimpe à $71,62\%$. En général, l'espace ressource est assez bien couvert ce qui démontre que les quadri-concepts sont des structures représentatives de la *v-folksonomie* avec une perte minimale d'information.

4.4.8 Couverture de l'espace Utilisateur

Étant donné que la précision d'un système de recommandation, spécialement dans les réseaux sociaux, est étroitement liée à la croissance de la taille des données, certains algorithmes peuvent fournir des recommandations avec une bonne qualité, mais uniquement pour un petit pourcentage d'utilisateurs. La couverture de l'espace utilisateur est définie comme étant la portion d'utilisateurs pour laquelle le système peut recommander des éléments [Ricci *et al.*, 2011]. Les algorithmes capables de fournir des recommandations à la majorité des utilisateurs sont donc particulièrement appréciés. Ceci dit, FOLKREC est capable de donner des recommandations à (presque) tous les utilisateurs de la *v-folksonomie* indépendamment du fait qu'un utilisateur ait tagué moins d'élements qu'un seuil défini ou le fait qu'il doit avoir un certain nombre d'amis. Dès qu'un utilisateur est ajouté à la *v-folksonomie*, ses informations personnelles sont suffisantes pour obtenir des recommandations de tags, de ressources et d'utilisateurs. Nous avons, par ailleurs, calculé le pourcentage de profils couverts par FOLKREC, ce qui a donné les résultats suivants : 100% de genres (homme et femme), 100% des catégories d'âge, 100% des métiers et 88% des villes [6]. À titre de comparaison, l'approche de Bellogìn *et al.* atteint une couverture égale à $84,6\%$. Notre résultat s'explique par le fait que FOLKREC prend en compte les nouveaux utilisateurs ce qui lui permet de couvrir presque tout l'espace utilisateur.

6. À partir de l'ensemble des 13625 villes représentées dans BOOKCROSSING, nous avons évalué la couverture de FOLKREC sur les villes les plus représentées, *i.e.,* les villes présentes dans, au moins, 500 quadruplets.

4.4.9 Démarrage à froid ou Cold Start

Un problème récurrent dans le processus de recommandation est celui du "démarrage à froid" ou cold start [Schein *et al.*, 2002a] [Weng *et al.*, 2008] [Ricci *et al.*, 2011], *i.e.,* la performance du système vis-à-vis des nouveaux utilisateurs [Schein *et al.*, 2002b]. Jusqu'à présent, nous avons eu recours à la fois à l'historique de tagging et aux informations de profil des utilisateurs pour fournir des recommandations. Cependant, nous ignorons encore les préférences des nouveaux utilisateurs qui viennent tout juste de commencer à utiliser le système. Un utilisateur est considéré comme nouveau s'il n'a encore tagué aucune ressource. Plusieurs systèmes de recommandation tentent de résoudre le problème du cold start en demandant, pour commencer, aux nouveaux utilisateurs d'annoter un certain ensemble de ressources [Ricci *et al.*, 2011]. Cependant, il n'est pas facile de décider quelles ressources un nouvel utilisateur doit annoter. Contrairement à la majorité des approches de la littérature, FolkRec ne requiert pas qu'un utilisateur ait partagé un nombre minimum de ressources avant d'être considéré par le système de recommandation. Cela revient au fait que FolkRec regarde d'abord le profil d'un utilisateur, *i.e.,* ses informations personnelles (âge, profession, etc.) pour lui fournir des recommandations. Après avoir créé de tels stéréotypes comme point de depart (*e.g.,* selon l'âge, le sexe, la profession ou la localisation des utilisateurs), lorsque ce même utilisateur commence à annoter des ressources avec des tags, le système pourra lui fournir de nouvelles recommandations selon ce qu'il a partagé (grâce à un apprentissage incrémental). Avec cette extension, FolkRec est maintenant capable de recommander à ses utilisateurs les nouvelles ressources ajoutées et de prendre en compte les nouveaux tags et utilisateurs sans avoir à redémarrer le processus d'extraction des quadri-concepts. Cependant, la limite de FolkRec est la non incrémentalité des nouveaux items (tags et ressources) qui ne sont pas encore inclus dans l'ensemble de quadri-concepts déjà extraits (*cf.*, Chapitre 5).

4.4.10 Sérendipité

La sérendipité mesure, avec quel degré de surprise, les recommandations apparaissent pour les utilisateurs. Un système de recommandation essaie de surprendre ses utilisateurs en leur recommandant des éléments qu'ils ne connaissent pas (encore). En effet, certains utilisateurs peuvent également être intéressés par un système capable de répondre à leurs requêtes en leur proposant des ressources qu'ils n'ont pas l'habitude de

consulter. Afin de mesurer ce degré de surprise, nous utilisons une métrique de distance d (*c.f.,* équation 4.2) qui mesure la distance entre l'élément recommandé et l'ensemble d'éléments déjà tagués par l'utilisateur. Ainsi, nous évaluons grâce à cette métrique, à quel point nos recommandations sur MovieLens et BookCrossing peuvent surprendre les utilisateurs, *i.e.,* en leur recommandant des livres et films avec lesquels ils ne sont pas familiers. La métrique de distance d est définie comme suit [Ricci *et al.*, 2011] :

$$d(b, B) = \frac{1 + C_B - C_B.w(b)}{1 + C_B} \tag{4.2}$$

À noter que la valeur de d se tient dans l'intervalle unitaire. Puis, afin de mesurer la sérendipité des recommandations, nous avons combiné cette métrique avec la pertinence des items (*i.e.,* precision). En effet, la sérendipité est atteinte si l'item est à la fois surprenant et pertinent. Nous avons testé la métrique d sur 20 utilisateurs aléatoires des deux jeux de données. Pour chacun des 20 utilisateurs, nous avons récupéré ses livres (ou films) partagés, compté le nombre maximal de livres (ou films) d'un seul auteur (ou réalisateur) qu'il/elle a partagé ; puis, nous calculons la distance entre les livres (ou films) recommandées et l'ensemble de livres (ou films) déjà partagés par l'utilisateur. Le score moyen de la métrique de distance d pour les utilisateurs de BookCrossing est égale à $0,86$, alors que la sérendipité (combinaison entre d et la précision) est égale à $0,71$. Pour le jeu de données MovieLens, la métrique de distance atteint la valeur maximale 1 et la sérendipité est égale à $0,83$. Les résultats obtenus sur les deux jeux de données démontrent que les utilisateurs de la *v-folksonomie* ont été très bien surpris par les ressources recommandées. Pour expliquer ces résultats, il est important de rappeler la stratégie prônée par FolkRec qui consiste en l'élagage des tags et ressources déjà partagés par chaque utilisateur avant la génération de la liste de recommandations. Ainsi, les utilisateurs sont plus susceptibles d'avoir des ressources qu'ils n'avaient jusque là pas (encore) partagé. Par exemple, nous avons recommandé pour l'utilisateur *Gold* (18 ans) le célèbre livre *Wild Animus* écrit par *R. Shapero*, un livre massivement partagé par des utilisateurs au même profil, alors qu'il avait tendance, jusqu'à présent, à lire principalement des livres écrits par *Ann M. Martin* (quatre livres partagés).

4.4.11 Diversité

Recommander un ensemble d'items qui sont similaires n'est pas aussi utile pour les utilisateurs, qui préfèrent la diversité, *i.e.,* des recommandations qui sont différentes et *distantes* [Yu *et al.*, 2009] [Badache et Boughanem, 2015] [Adomavicius et Kwon, 2012]. Par exemple, un utilisateur préférera une recommandation de cinq livres écrits par cinq auteurs différents à une recommandation de cinq livres écrits par un même auteur. Afin de mesurer la diversité de nos recommandations sur les jeux de données MovieLens et BookCrossing, nous utilisons la même métrique de distance d (*cf.,* équation 4.2) de la manière suivante : nous calculons la distance entre chaque livre recommandé et le reste de la liste des livres recommandés et ensuite, nous calculons la moyenne de ces résultats afin d'obtenir le score de diversité. Nous avons testé la métrique d sur 20 utilisateurs aléatoires des deux jeux de données et nous avons obtenu un score de diversité moyen égal à $0,84$ pour l'ensemble des utilisateurs de BookCrossing et un score de $0,98$ pour l'ensemble des utilisateurs de MovieLens. De tels scores s'expliquent par le fait que les deux jeux de données contiennent une large gamme de ressources (*i.e.,* 10681 films et 271379 livres) ce qui augmente les chances pour un utilisateur d'avoir des recommandations diverses. Grâce aux quadri-concepts qui sont représentatifs de la *v-folksonomie*, FolkRec vise à recommander des ressources qui sont diverses, *e.g.,* des livres de différents auteurs ou encore des films réalisés par différents réalisateurs. Par example, *Marco22* (étudiant) a reçu une liste de recommandation contenant cinq films réalisés par cinq différents réalisateurs : *Star Wars* (*Lucas*), *Empires Strikes Again* (*Kersh*), *Princess Bride* (*Reiner*), *Return of the Jedi* (*Marquand*) et *Pulp Fiction* (*Tarantino*). À titre de compraisons, le score de diversité atteint par l'approche de Bellogìn *et al.* est égal à $0,35$.

4.4.12 Adaptativité

Tester l'adaptativité du système revient à calculer le taux avec lequel le système arrive à s'adapter aux changements de préférences d'un utilisateur, ou aux changements dans son profil [Ricci *et al.*, 2011] [Mahmood et Ricci, 2007]. Dans ce qui suit, afin d'évaluer l'adaptativité de FolkRec, nous mesurons la différence entre les listes de recommandations fournies par notre système avant et après qu'une information de profil (sur un utilisateur) ait été ajoutée ou modifiée. Pour ce faire, nous utilisons la mesure *Gini Index* [Fleder et Hosanagar, 2007] afin de mesurer la variabilité des

recommandations fournies à un utilisateur lorsque son profil change. Cette métrique mesure avec quelle inégalité différents items sont choisis par les utilisateurs lorsqu'un système de recommandation est utilisé. Si chaque item i compte pour une portion $p(i)$ des choix de l'utilisateur, la mesure *Gini Index* est donnée par :

$$G = \frac{1}{n-1} \sum_{j=1}^{n} (2j - n - 1) p(i_j) \tag{4.3}$$

où i_1, ..., i_n est la liste des ressources ordonnées par $p(i)$ décroissant. L'index est égal à 0 lorsque tous les items sont choisis de manière égale, et 1 lorsque un seul item est toujours choisi. Ainsi, nous évaluons la variabilité des recommandations fournies à différents utilisateurs lorsque leurs profils changent. Pour le jeu de données MovieLens, nous avons imaginé différents scénarios de changements de profils sur différents utilisateurs. La variabilité moyenne obtenue sur ces différents utilisateurs est égale à $0,126$. Considérons, par exemple, l'utilisateur *nicholas* (18 ans, artiste). La valeur de la métrique G des recommandations données à *nicholas* est passé de $0,328$ à $0,492$ lorsque sa profession a changé de *artiste* à *étudiant* ; une telle différence est expliquée par la dissimilarité des recommandations entre les deux différents profils. Cependant, la variabilité décroît de seulement $0,01$ entre utilisateurs des deux catégories d'âge [19-24] ans et [25-35] ans, ce qui semble logique étant donné que les deux profils sont très proches. Concernant le jeu de données BookCrossing, la variabilité est d'environ $0,1$. Considérons l'utilisateur *Skinner* (38 ans, New York, USA) avec le scénario suivant : l'utilisateur a changé de ville en partant vers (Ottawa, Canada). FolkRec prend en compte ce changement de profil et propose à *Skinner* une nouvelle recommandation qui est plus adaptée à son nouveau profil, *i.e.,* nous lui recommandations les livres *The Alchimist*, *The Red Tent* et *Love in the time of cholera*, trois livres partagés en masse par les utilisateurs canadiens vivant à Ottawa offrant ainsi à *Skinner* la possibilité de découvrir les livres qui sont en vogue dans sa nouvelle ville. De plus, nous lui recommandons trois utilisateurs d'Ottawa : *tchang* (27 ans), *archie* (49 ans) et *grump* (40 ans).

Dans cette partie, nous nous sommes focalisés sur les changements dans le profil (changement de profession, ville, etc.) mais nous pouvons également nous focaliser sur les changements dans l'historique de tagging des utilisateurs. Cela semble trivial qu'un utilisateur qui partage de nouvelles ressources, c'est-à-dire, qui devient intéressé par un nouveau genre de films ou de livres (*e.g.,* films d'actions au lieu de films de comédie par

exemple), aura des nouvelles recommandations pouvant correspondre à ses nouvelles préférences.

4.4.13 Scalabilité

L'approche standard pour évaluer la scalabilité d'un système est d'évaluer la complexité de l'algorithme dédié en termes de temps d'exécution ou/et de mémoire requise. Ainsi, nous calculons le temps moyen d'exécution (en millisecondes) de nos recommandations sur les deux jeux de données pour les tâches de recommandation de ressources (dénotée Tâche 1) et de proposition d'utilisateurs (dénotée Tâche 2)[7]. Le Tableau 4.13 affiche tout d'abord le temps d'exécution de FolkRec sur le jeu de données MovieLens qui contient 95580 quadruplets <utilisateur,tag,ressource,profil>. Chaque quadri-concept extrait contient au moins un tag, une ressource et une information de profil, alors que nous faisons varier le support minimum d'utilisateurs (*minsupp_u*), *i.e.,* le nombre minimum d'utilisateurs par quadri-concept. Par exemple, lorsque *minsupp_u* est égal à 6, nous avons 13461 quadri-concepts fréquents où chaque concept contient, au moins, 6 utilisateurs. De plus, nous avons calculé le nombre d'utilisateurs (uniques) sur l'ensemble des quadri-concepts fréquents. Le Tableau 4.13 démontre les bonnes performances de FolkRec pour toutes les valeurs de *minsupp_u*. Tandis que le nombre de quadri-concepts augmente rapidement (de 221 à 13461), le temps d'exécution des recommandations générées par FolkRec est en moyenne de 2 ms et de 8 ms pour, respectivement, la première et deuxième tâche. Le nombre total des recommandations (*i.e.,* le nombre d'utilisateurs uniques) est égal à 865 pour la plus petite valeur de *minsupp_u*.

Le Tableau 4.13 affiche également les performances de FolkRec sur le jeu de données BookCrossing qui contient 762000 quadruplets <utilisateur,tag,ressource,profil>. Chaque quadri-concept fréquent contient, au moins, un tag, une ressource et une information de profil et nous avons fait varier le nombre minimum d'utilisateurs par quadri-concept, *i.e.,* le seuil de support *minsupp_u*, de 30 à 10. Tout d'abord, nous pouvons voir que le nombre maximal de quadri-concepts extraits est égal à 10100 ce qui représente seulement $1,76\%$ de la *v-folksonomie* ; ce qui démontre, une fois de plus, l'utilité des quadri-concepts, qui

7. Nous avons omis la tâche de suggestion de tags étant donné que BookCrossing considère les notes plutôt que les tags.

sont une représentation réduite de la *v-folksonomie*. Contrairement à MovieLens, le nombre d'utilisateurs uniques, *i.e.,* le nombre total de recommandations, augmente considérablement tandis que le nombre de quadri-concepts fréquents croît légèrement (en raison des valeurs élevées de $minsupp\_u$). Cependant, si FolkRec affiche toujours des bonnes performances pour la tâche de recommandation de ressources où le temps moyen d'exécution est de 28 ms, la tâche de proposition d'utilisateurs devient légèrement plus lente étant donné que chaque quadri-concept contient au moins 10 utilisateurs. Cependant, le temps d'exécution demeure raisonnable à hauteur de 384 ms en moyenne.

	$minsupp\_u$	$\|\mathcal{QC}\|$	# Utilisateurs Uniques	Tâche 1 (ms)	Tâche 2 (ms)
(MovieLens)	20	221	526	0,1	2,6
	16	500	605	0,2	3,9
	12	1295	668	0,7	6,1
	8	5123	805	4,0	13,5
	6	13461	865	12,7	23,3
(BookCrossing)	30	553	6789	0,9	149,8
	20	1486	9092	4,9	296,9
	16	2638	10397	13,0	415,5
	12	5698	12239	45,0	542,3
	10	10100	13457	114,7	586,8

Tableau 4.13 — Temps moyen d'exécution des recommandations de FolkRec.

Le Tableau 4.14 reprend le récapitulatif fait dans le Chapitre 2 en mettant, cette fois, en lumière nos contributions par rapport aux travaux de la littérature pour les différentes propriétés pré-définies. Par exemple, concernant le problème de cold start, le recours à des informations supplémentaires et personnelles sur les utilisateurs a permis de générer des recommandations aux nouveaux utilisateurs en se basant uniquement sur ces données. En effet, notre algorithme a permis de proposer des recommandations à ses nouveaux utilisateurs en cherchant les tags et ressources déjà partagés par des utilisateurs qui ont en commun un certain nombre d'informations personnelles avec ces nouveaux utilisateurs.

	M-Mode	Couv.	C-Start	Adaptativité	Diversité	Sérendipité	Scalabilité
[Diederich et Iofciu, 2006a]	non	no	non	non	oui	?	non
[Basile *et al.*, 2007]	non	non	non	non	?	?	?
[Jäschke *et al.*, 2007]	non	non	non	non	oui	?	?
[Landia et Anand, 2009]	non	non	non	non	non	?	?
[De Meo *et al.*, 2010b]	non	non	non	non	?	?	?
[Hu *et al.*, 2011]	non	non	oui	non	oui	?	?
[Kim *et al.*, 2011]	non	non	non	non	oui	?	?
[Qumsiyeh et Ng, 2012]	non	non	non	non	oui	?	?
[Bellogín *et al.*, 2013]	non	oui	non	non	oui	?	?
Notre approche	oui	oui	oui	oui	oui	oui	?

Tableau 4.14 — Comparaison entre les travaux de la littérature par rapport aux propriétés des systèmes de recommandation.

4.5 Étude de cas

Dans ce qui suit, nous menons une étude sur un ensemble d'utilisateurs afin d'avoir leur "feedback" sur notre système personnalisé de recommandation. Ainsi, nous avons choisi six sujets avec des profils différents : (*Nidhal*, Homme, 30 ans, Assistant universitaire, Tunisie), (*Imen*, Femme, 26 ans, Etudiante, Tunisie), (*Roxane*, Femme, 27 ans, Professeur d'arts plastiques, France), (*Raymond*, Homme, 58 ans, Retraité, Belgique), (*Wassim*, Homme, 24 ans, Ingénieur, Canada) and (*Quentin*, Homme, 28 ans, Opticien, France). Nos six sujets sont invités à effectuer un ensemble de tâches à l'aide du système et de répondre à différentes questions. Un tel processus permet de collecter des données sur les utilisateurs qui n'est pas directement observable, comme savoir si le sujet a apprécié l'interface utilisateur ou encore si le sujet a réellement apprécié les recommandations. L'étude est divisée en quatre tâches : (*i*) **Qualité de la recommandation** Nous recommandons à chaque utilisateur une liste de ressources (films puis livres) et nous leur demandons de noter chaque ressource selon l'échelle suivante : 5 :*Très bon*, 4 :*Bon*, 3 :*Assez bon*, 2 :*Mauvais*, 1 :*Très mauvais* ; (*ii*) **Recommandation de ressources** Pour chaque utilisateur de notre étude, nous fournissons une liste élargie de ressources incluant cinq ressources pertinentes [8] et nous lui demandons de sélectionner une liste de ressources qu'il (elle) considère comme intéressante. Ensuite, nous cal-

8. Une ressource (*resp.* tag ou utilisateur) pertinente est une ressource (*resp.* tag ou utilisateur) qui est recommandée par notre système de recommandation.

culons la proportion de ressources pertinentes qui a été sélectionnée. Idéalement, les ressources sélectionnées par l'utilisateurs correspondent aux ressources recommandées par FOLKREC, *i.e.,* les pertinentes ; (*iii*) **Suggestion de tags** Lorsque un utilisateur s'apprête à partager un film ou un livre, nous lui suggérons une liste élargie de tags incluant trois tags pertinents et lui demandons de sélectionner trois tags qu'il considère comme appropriés à la ressource qu'il est entrain de partager. Ensuite, nous comparons les tags sélectionnés par chaque utilisateur avec les tags pertinents afin de savoir si les utilisateurs ont réellement choisi les tags qui ont été recommandés par FOLKREC ; et (*iv*) **Proposition d'utilisateurs** Pour chaque utilisateur de l'étude, nous proposons une liste d'utilisateurs avec leurs profils respectifs et nous lui demandons de sélectionner le ou les utilisateurs qu'il ajouterait comme ami(s). Enfin, les différents utilisateurs de l'étude ont la possibilité de justifier leurs différents choix à travers différentes zones de texte.

Pour la première tâche, les utilisateurs ont noté les films et livres recommandés avec une note moyenne respective de $3,68$ et $3,1$ ce qui démontre une certaine bonne appréciation des recommandations faites par FOLKREC. Parmi les utilisateurs, *Roxane* semble réellement apprécier les films recommandés (deux "très bon" et deux "bon") ; elle a commenté *"J'ai réellement apprécié le fait que le système me recommande mes films préférés : Le Parrain et Star Wars."*. Le même sujet a également apprécié les livres recommandés *"Je ne connaissais pas la plupart de ces livres et j'ai vraiment apprécié ça."* ce qui souligne la **nouveauté** de nos recommandations. La seconde tâche, *i.e.,* la recommandation de ressources, met en évidence que $49,8\%$ des films sélectionnés par les utilisateurs sont pertinentes. Par exemple, sur les six films choisis par *Roxane*, quatre sont pertinents. Quant aux films sélectionnés, la pertinence s'élève également à $49,8\%$: par exemple, 66% des livres sélectionnés par *Raymond* sont pertinents ; le sujet commente à ce propos que "les livres sont très divers, y a du classique comme des livres d'aventures.". Tandis que *Raymond* a apprécié la **diversité** de nos recommandations ; le sujet *Nidhal* souligne la **serendipité** des propositions *"J'étais vraiment surpris par les livres recommandés, je ne connaissais pas la plupart des livres. Ca m'a permis de redécouvrir mes goûts."*. Contrairement aux films et livres préférés qui peuvent varier entre utilisateurs d'un même profil, les résultats de la troisième tâche (*i.e.,* la suggestion de tags) démontrent que des utilisateurs ayant le même profil convergent vers un vocabulaire commun. Ainsi, $66,4\%$ des tags sélectionnés par les sujets correspondent à des tags utilisés par des utilisateurs au même profil. Par exemple, pour annoter le film

Jurassic Park, *Raymond* a préféré les tags *dinosaurs*, *steven_spielberg* et *oscar_winner* aux tags *genetics*, *action* et *thriller*, ce qui correspond parfaitement aux tags choisis par les utilisateurs au même profil que lui. Enfin, la dernière tâche, *i.e.,* la proposition d'utilisateurs, démontre que la moitié des sujets ont choisi un utilisateur ayant le même profil que lui. L'étude démontre, par ailleurs, que si quelques utilisateurs (*Roxane*, *Raymond* et *Quentin*) cherchent des amis avec le même profil, les autres utilisateurs semblent s'intéresser à des utilisateurs avec un profil différent. Par exemple, l'assistant universitaire *Nidhal* ajouterait comme ami *Patrick* qui est bibliothécaire intéressé par des films tirés d'une histoire vraie. Le constat général qui se dégage de notre étude est que les utilisateurs ont été globalement satisfaits par les recommandations de ressources, de tags et d'utilisateurs comme l'a mentionné *Wassim* : *"J'ai trouvé le questionnaire très intéressant. Tout d'abord, parce que cela portait sur des films qui font partie de mes loisirs. J'ai aimé répondre aux questions et cela m'a rappelé quelques vieux films que j'ai adoré regarder : Titanic, Braveheart, Blade Runner, etc."*.

Cependant, la limite de notre étude est qu'elle soit restreinte au profil statique de nos sujets, ce qui représente une information incomplète pour les recommandations. Un de nos sujets, *Imen* a ainsi commenté : *"J'ai pas apprécié les livres recommandés ! Et je n'aime pas spécialement que les films de filles, je préfère aussi les films d'action comme Seven ou Die Hard."*. Ce problème peut être résolu par une évaluation en ligne à travers une analyse temps-réel des tags et ressources partagés par les utilisateurs. Une telle analyse peut, par exemple, nous montrer que *Imen* s'intéresse non seulement aux films romantiques mais aussi aux films d'action. Nous concluons que notre étude est capable de fournir de bonnes recommandations, mais qu'un suivi en ligne des tags et ressources des utilisateurs permettrait d'améliorer les résultats de notre système de recommandation.

4.6 Conclusion

Dans ce chapitre, nous avons utilisé les quadri-concepts extraits par notre algorithme dédié à cette tâche afin de proposer notre nouveau système personnalisé de recommandation. Pour améliorer la qualité des recommandations, nous avons également introduit une mesure de ranking afin de classer les différentes recommandations. Nous avons ensuite mené des expérimentations sur deux jeux de données du monde réel et calculé la précision et le rappel de notre système. Puis, nous avons passé en revue

les différentes propriétés de notre système de recommandation, comme la scalabilité, la couverture, l'adaptivité, la sérendipité ou encore la diversité des recommandations proposées. Enfin, pour encore plus d'interactivité avec les utilisateurs, nous avons mené une étude de cas sur six utilisateurs afin de recueillir leurs impressions sur notre système de recommandation. Une limite de notre approche est que la première recommandation reste toujours dépendante de l'étape de pré-traitement qui consiste en l'extraction des quadri-concepts alors que les recommandations suivantes s'appuient sur un ensemble de quadri-concepts déjà extraits. Une autre limite est que notre système de recommandation compte sur des quadri-concepts qui sont *statiques* qui n'évoluent pas à travers le temps. Ainsi, nous avons besoin d'une méthode incrémentale qui permet de mettre à jour l'ensemble des quadri-concepts afin de proposer les recommandations les plus récentes. C'est ainsi que dans le chapitre suivant, nous allons proposer un nouvel algorithme incrémental dont le but est de mettre à jour l'ensemble des quadri-concepts extraits suite à l'ajout de nouveaux quadruplets dans la base. Cela permettra de prendre en compte notamment les nouveaux tags et ressources qui viennent d'être ajoutés à la *folksonomie* et de proposer aux utilisateurs des recommandations récentes qui prennent en compte les dernières mises à jour du système.

Contributions

- Mohamed Nader Jelassi, Sadok Ben Yahia et Engelbert Mephu Nguifo : *Towards more targeted recommendations in folksonomies*. Social Networks Analysis and Mining journal (SNAM). Springer Ed. (Accepted)
- Mohamed Nader Jelassi, Sadok Ben Yahia et Engelbert Mephu Nguifo : *A personalized recommender system based on users' information in folksonomies*. International Workshop on Web Intelligence & Communities(WI&C 2013) in conjunction with the 22nd International Conference on World Wide Web (WWW 2013). 1215-1224.
- Mohamed Nader Jelassi, Sadok Ben Yahia et Engelbert Mephu Nguifo : *Nouvelle approche de recommandation personnalisée dans les folksonomies basée sur le profil des utilisateurs*. Journées francophones d'Ingénierie des Connaissances (IC) 2013 : 224-226.
- Mohamed Nader Jelassi, Sadok Ben Yahia et Engelbert Mephu Nguifo : *Vers des recommandations plus personnalisées dans les folksonomies*. 25es Journées francophones d'Ingénierie des Connaissances (IC) 2014 : 187-198.
- Mohamed Nader Jelassi, Sadok Ben Yahia et Engelbert Mephu Nguifo : *Towards more targeted recommendations in folksonomies*. 5th Workshop on Context-awareness in Retrieval and Recommendation (CaRR 2015) in conjunction with the 37th European Conference on Information Retrieval (ECIR 2015).
- Mohamed Nader Jelassi, Sadok Ben Yahia et Engelbert Mephu Nguifo : *PERSO-REC : un système personnalisé de recommandations pour les folksonomies basé sur les concepts quadratiques* (IC) 2015.

CHAPITRE 5

FolkIncr, un algorithme pour la mise à jour des tri-concepts

5.1 Introduction

Dans le chapitre précédent, nous avons utilisé les quadri-concepts dans le but de proposer notre nouveau système personnalisé de recommandation. Nous avons également introduit une mesure de ranking afin de classer les différentes recommandations pour en améliorer la qualité. Les expérimentations que nous avons mené sur deux jeux de données du monde réel ont donné des résultats satisfaisants en termes de précision et de rappel. De plus, nous avons passé en revue les différentes propriétés de notre système de recommandation, comme la scalabilité, la couverture, l'adaptivité, la sérendipité ou encore la diversité des recommandations proposées. Enfin, nous avons mené une étude de cas sur six utilisateurs dans l'objectif d'avoir leur feedback sur notre système. Cette étude de cas ainsi que l'évaluation expérimentale nous ont éclairé sur certaines limites de notre système comme la nécessité de mettre à jour l'ensemble des données dans le but d'avoir des recommandations qui prendront en compte les dernières modifications du contexte en entrée.

Dans ce chapitre, pour atteindre cet objectif, nous allons proposer un nouvel algorithme dont le but est de mettre à jour les données extraites suite à l'ajout de nouvelles données en entrée. En effet, les utilisateurs partagent des milliers de ressources de manière quotidienne et cela devient irréalisable de relancer à chaque fois le processus d'extraction de concepts fréquents. Cela nous a donc motivé pour concevoir notre algorithme afin de maintenir à jour l'ensemble des concepts sans devoir

rescanner le contexte en entier. Dans ce qui suit, nous allons traiter la problématique de mise à jour de concepts pour le cas tri-dimensionnel représenté par les *folksonomies*. En effet, le problème d'extraction des tri-concepts est beaucoup plus souvent traité dans la littérature [Ji *et al.*, 2006] [Jäschke *et al.*, 2008] [Trabelsi *et al.*, 2012] [Gnatyshak *et al.*, 2012] [Cerf *et al.*, 2013] que celui de l'extraction des quadri-concepts que nous avons introduit dans ce mémoire. En effet, il s'agit du cas général permettant d'extraire des concepts correspondant à la structure classique d'une *folksonomie* sans ajout de variables supplémentaires. L'évaluation expérimentale que nous allons donc mener comparera les performances de notre algorithme à ceux de la littérature dédiés à la tâche d'extraction des tri-concepts. Cela permettra de prendre en compte notamment les nouveaux tags et ressources qui viennent d'être ajoutés à la *folksonomie* et de proposer aux utilisateurs des recommandations récentes qui prennent en compte les dernières mises à jour du système.

5.2 Problématique

Dans les *folksonomies*, les utilisateurs partagent quotidiennement des milliers de ressources en les annotant avec un ensemble de tags. Afin d'exploiter cette gigantesque quantité d'information pouvant être extraite à partir des *folksonomies*, des travaux ont été dédiés à la tâche d'extraction de représentations concises sans perte d'information de patterns intéressants appelés *tri-concepts* [Cerf *et al.*, 2013] [Jäschke *et al.*, 2008] [Ji *et al.*, 2006] [Trabelsi *et al.*, 2012] (*cf.,* Chapitre 3). La tâche consistant à mettre à jour cet ensemble de tri-concepts est particulièrement motivant étant donné que de tels concepts sont très utilisés dans plusieurs domaines comme l'analyse des expressions de gênes [Zhao et Zaki, 2005], l'analyse des réseaux sociaux [Gnatyshak *et al.*, 2012] ou encore les systèmes de recommandations [Jelassi *et al.*, 2013b] [Ren *et al.*, 2013] vu qu'ils représentent une représentation concise et sans perte d'information des *folksonomies* ainsi qu'une précieuse source d'information. Ainsi, avoir un ensemble de tri-concepts mis à jour permet, par exemple, d'offrir les recommandations les plus récentes aux utilisateurs des *folksonomies*. Cependant, les algorithmes actuels dédiés à la tâche d'extraction des tri-concepts ne prennent pas en compte le problème de mise à jour des données, *i.e.,* des changements mineurs dans la *folksonomie* implique le recalcul des tri-concepts à partir de toute la *folksonomie* ainsi mise à jour. En effet, avec le développement rapide des moyens technologiques, plusieurs applications récentes, spécialement

les réseaux sociaux, générent quotidiennement de larges quantités de données de manière continue. Ainsi, le principal avantage des *folksonomies* réside dans leur dynamique et continuel développement. Leurs structures évoluent continuellement étant donné que les utilisateurs participent à la mise à jour permanente du système. Si une telle dynamique peut être vue comme un point fort, elle peut également être considérée comme une faiblesse. En effet, la plupart des travaux traitant les *folksonomies* ne prennent pas en compte une telle particularité et ne considèrent qu'un *instantané* statique des données à un point particulier dans le temps. Par exemple, les travaux mentionnés ci-dessus [Cerf *et al.*, 2013] [Jäschke *et al.*, 2008] [Ji *et al.*, 2006] [Trabelsi *et al.*, 2012] qui extraient les tri-concepts à partir des *folksonomies* proposent de fouiller uniquement un instantané statique d'une *folksonomie* donnée sans prendre en compte les futures mises à jour afin de mettre à jour les concepts déjà extraits. Ainsi, la mise à jour des données dans les *folksonomies* est le problème majeur lorsque nous voulons extraire les tri-concepts tout en prenant en compte les nouvelles données qui arrivent. Par ailleurs, la quantité de partages quotidiens varient d'une *folksonomie* à une autre. Par exemple, dans MovieLens [1], quelques 567 films sont partagés quotidiennement par les utilisateurs alors qu'avec un milliard de liens partagés tous les jours, les utilisateurs de Delicious [2] organisent massivement les liens qu'ils trouvent intéressants sur la toile. Afin de manipuler les données les plus récentes, une solution consiste à redémarrer un algorithme qui permet d'extraire les tri-concepts et ainsi mettre à jour l'ensemble de sortie. Cependant, cette tâche s'avère être coûteuse étant donné que cela revient à re-scanner tout le contexte afin de modifier, ajouter ou supprimer des tri-concepts. Ainsi, le souci majeur avec les grands contextes dynamiques est d'arriver à ré-arranger l'ensemble de sortie suite à quelques changements mineurs dans l'ensemble d'entrée. En effet, avec les nouvelles données qui arrivent, il n'y a pas assez de temps pour re-scanner l'entière *folksonomie*, lorsqu'une mise à jour est nécessaire. Donc, un unique scan de la *folksonomie* ainsi qu'une utilisation minimale de la mémoire sont nécessaires. Dans ce chapitre, nous proposons un nouvel algorithme qui permet de considérer les nouvelles données dans les *folksonomies* puis de mettre à jour l'ensemble de tri-concepts. À notre connaissance, aucun algorithme n'a été proposé dans la littérature pour traiter le problème de mise à jour de données dans les *folksonomies*. Cependant, quelques travaux ont traité cette problématique dans les contextes à deux

1. https ://movielens.org
2. *http ://delicious.com*

dimensions [Valtchev *et al.*, 2002] [Merwe *et al.*, 2004] [Obiedkov et Duquenne, 2007]. Notre algorithme considère un ensemble de tri-concepts extraits à partir d'une *folksonomie* ainsi qu'un ensemble de nouveaux triplets (au moins un triplet). Le but est d'arriver à mettre jour l'ensemble de tri-concepts sans devoir les re-calculer, *i.e.,* redémarrer le processus d'extraction des tri-concepts. Notre méthode est une technique qui garantit une bonne adaptation aux contextes dynamiques sans un deuxième parcours coûteux de la *folksonomie*.

Exemple motivant

Considérons l'utilisateur *John*, souhaitant soumettre un nouveau papier à la conférence *IDA*, qui cherche les papiers les plus récents dans le domaine du *Data Mining*. Dans un système de recommandation classique, *John* aura une liste de papiers recommandés qui ont été partagés par quelques utilisateurs à travers des tags relatifs au domaine du *Data Mining*. Une telle recommandation est basée sur des donnés statiques d'une *folksonomie* donnée (*e.g.,* BIBSONOMY). Cependant, des papiers récents ajoutés par les mêmes utilisateurs à travers les mêmes tags seront *zappés* étant donné que de tels papiers n'appartiennent pas encore aux données considérées par le système de recommandation. Ainsi, nous avons besoin de mettre à jour ces données afin de prendre en compte les papiers les plus récents et les recommander à *John*. Grâce à une telle mise à jour, *John* aura une recommandation sans perte d'information, *i.e.,* tous les papiers des plus anciens aux plus récents traitant du domaine du *Data Mining*. La maintenance du système à travers des mises à jour a deux avantages : *(i)* le coût de l'opération est beaucoup moins grand que le redémarrage du processus d'extraction des tri-concepts pour l'ensemble des données, *(ii)* la non perte d'information du système qui considère l'ensemble des données (anciennes et nouvelles) avant l'étape de recommandation.

5.3 État de l'art sur les algorithmes

Le problème de mise à jour de données (aussi appelé problème d'incrémentalité) a été largement étudiée dans le domaine de la fouille de données dans le but de maintenir un ensemble d'itemsets fréquents ou/et son treillis correspondant [Valtchev *et al.*, 2002] [Merwe *et al.*, 2004] [Obiedkov et Duquenne, 2007]. La problématique consiste à ce qu'à travers de petits changements dans le contexte d'entrée, correspondent des pe-

tits réarrangements dans le résultat en sortie. Cette problématique concerne surtout les grand jeux de données à large échelle qui ont la caractéristique d'être dynamiques. Par exemple, un jeu de données d'un magasin va évoluer quotidiennement au fur et à mesure que les clients achètent des articles. Chaque article acheté correspondra à une nouvelle ligne dans le jeu de données. L'objectif est donc de mettre à jour l'ensemble d'itemsets fréquents extraits sans devoir relancer à nouveau un algorithme dédié à cette tâche. Dans un cas du monde réel, supposons que nous disposons de l'itemset fréquent suivant extrait à partir d'un large jeu de données d'un magasin d'aliments généraux : (*beurre*, *lait*, *farine*) avec le support égal à 3. Maintenant, supposons qu'un nouvel article *sucre* a été ajouté au jeu de données avec un support égal à 3 et correspondant au mêmes clients qui ont acheté précédemment le *beurre*, le *lait* et la *farine*. Un algorithme incrémental aura la tâche d'incruster ce nouvel article dans l'itemset déjà extrait sans avoir à regénérer l'ensemble d'itemsets fréquents à partir du très grand nombre de données du jeu de données. Ainsi mis à jour, l'itemset précédent sera égal à : (*beurre*, *lait*, *farine*, *sucre*). Cette intéressante perspective a éveillé la curiosité de bon nombre de chercheurs et suscité de nombreux travaux [Valtchev *et al.*, 2002] [Merwe *et al.*, 2004] [Obiedkov et Duquenne, 2007]. Dans ce dernier papier, Valtchev *et al.* proposent une nouvelle méthode pour l'incrémentalité d'un treillis construit à partir d'un contexte. Les auteurs ont présenté un algorithme incrémental qui transforme le treillis extrait suite à l'ajout d'une ligne ou/et colonne dans le contexte d'entrée. En effet, lorsqu'un nouvel objet est ajouté, l'algorithme incrémental proposé permet de mettre à jour le treillis extrait à partir de comparaisons avec les itemsets extraits. À notre connaissance, aucun travail n'a été proposé dans la littérature pour traiter le problème de mise à jour de données dans les *folksonomies*. Ainsi, dans ce qui suit, pour détailler et expliquer le processus de mise à jour de l'ensemble des itemsets extraits à partir d'un contexte, nous allons expliciter la méthode de Valtchev *et al.* pour l'incrémentalité dans les contextes classiques à deux dimensions.

L'algorithme proposé par Valtchev *et al.* montre le processus de mise à jour du treillis extrait suite à l'ajout d'un nouvel objet o. Dans cet algorithme,il y a deux principaux cas selon l'*intent* (*i.e.,* l'ensemble d'attributs correspondant à un concept donné) d'un itemset du treillis. En effet, si l'intent d'un itemset c est inclus dans celui de o (noté o'), alors c est ajouté à un ensemble noté $\mathbf{M(o)}$, sinon il est ajouté à un ensemble noté $\mathbf{G(o)}$. À la fin de ce processus, $\mathbf{M(o)}$ renferme l'ensemble des concepts **M**odifiés, *i.e.,* dont l'extent sera augmenté alors que $\mathbf{G(o)}$ renferme l'ensemble des **G**énérateurs

qui donneront plus tard naissance à de nouveaux concepts. Par ailleurs, implicitement, dans l'algorithme COMPUTE-LATTICE-INC, il existe un troisième ensemble appelé ensemble de concepts inchangés qui garderont intacts leur ensembles d'extent et d'intent au sein du treillis. Par suite, pour tous les itemsets ajoutés à l'ensemble **M(o)**, l'algorithme met à jour leurs extents (*i.e.,* l'ensemble d'objets correspondant à un concept donné) en l'augmentant par o. En effet, si l'intent de c est inclus dans celui de o, cela veut dire que l'objet o peut être ajouté à l'extent de c sans pour autant changer l'intent de c. Parallèlement, pour les itemsets ajoutés à l'ensemble **G(o)** dont l'intent n'est pas inclus dans celui de o, ils donneront naissance à un nouveau concept $\hat{c}$ dont l'extent est égal à l'union entre celui de c et o alors que l'intent sera égal à l'intersection entre celui de c et celui de o. En effet, si l'extent de c sera augmenté par o, l'intent sera diminué suite à l'intersection avec celui de o. Ensuite, l'algorithme procède à un processus de mise à jour des liens au sein du treillis. En effet, un nouveau concept créé sera directement lié à l'itemset c qui lui a donné naissance alors que le lien entre c et le concept qui le contient sera rompu. L'algorithme termine lorsque tous les concepts du treillis ont été énumérés et retourne ainsi le treillis mis à jour.

	a	b	c	d	e	f	g	h
1		X	X	X			X	X
2	X		X					
3				X	X	X	X	X
4							X	
5					X	X		X
6	X	X	X	X				
7		X	X	X				
8				X				
9			X	X		X	X	X

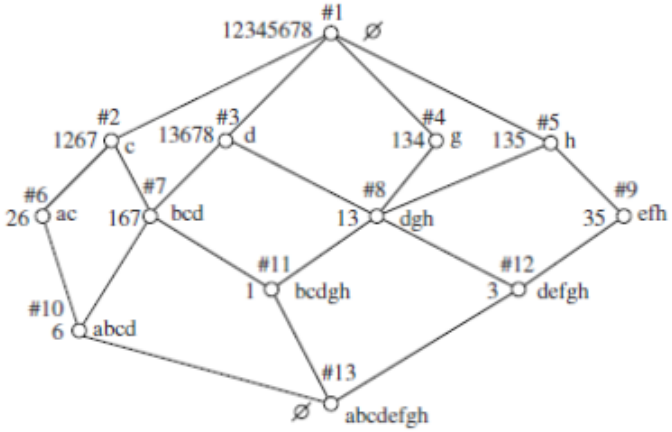


Figure 5.1 — (**Gauche**) Le contexte d'entrée $\mathcal{K} = (O = \{1, 2, \ldots, 8\}, A = \{a, b, \ldots, h\}, R)$ et le nouvel objet 9. (**Droite**) Le treillis extrait à partir de K.

Exemple 11 *(Insertion de l'objet 9)*

La Figure 5.1 décrit un exemple d'un contexte binaire $\mathcal{K}$ contenant 8 objets et 8 colonnes ainsi que le treillis correspondant extrait à partir de ce contexte et renfermant l'ensemble des concepts. Considérons le nouvel objet o, i.e., l'objet o=9 à insérer dans le treillis. Nous avons d'abord un ensemble de concepts qui seront inchangés {c#6, c#10}

tandis qu'à partir de l'intent de chaque concept, nous diviserons l'ensemble des concepts du treillis en deux ensembles : $\boldsymbol{M(o)}$ *et* $\boldsymbol{G(o)}$*. Par exemple, nous avons intent(c#4)=g* $\subseteq o'$*=cdfgh, alors le concept c#4 sera ajouté à l'ensemble* $\boldsymbol{M(o)}$*. Par contre, nous avons intent (c#9)=efh* $\not\subseteq o'$*=cdfgh, alors le concept c#9 sera ajouté à l'ensemble* $\boldsymbol{G(o)}$*. Ce processus se répète pour tous les concepts du treillis et donnera les résultats suivants : l'ensemble des concepts modifiés est égal à* $\boldsymbol{M(o)}$ *= {c#1, c#2, c#3, c#4, c#5, c#8} et l'ensemble des générateurs est égal à* $\boldsymbol{G(o)}$ *= {c#7, c#9, c#11, c#12, c#13}. Ce dernier ensemble donnera naissance à cinq nouveaux concepts {c#14, c#15, c#16, c#17, c#18} dont l'intent respectif est égal à {cd, fh, cdgh, dfgh, cdfgh}. La Figure 5.2 décrit le treillis ainsi mis à jour suite à l'insertion du nouvel objet* 9*.*

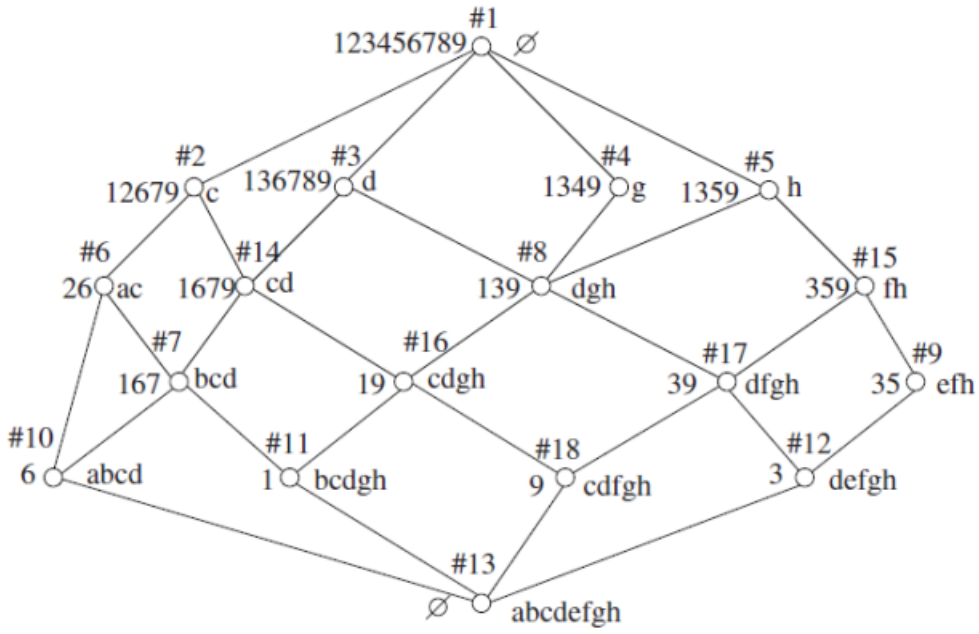


Figure 5.2 — Le treillis extrait à partir de K suite à l'insertion du nouvel objet 9.

Dans cette partie, nous avons décrit le processus de mise à jour dans un contexte binaire classique. Cependant, à notre connaissance, aucun travail n'a été proposé dans la littérature pour répondre à ce problème dans les contextes triadiques représentées par les *folksonomies*. Ainsi, nous nous sommes inspirés par les travaux courants (décrits ci-dessus) afin d'introduire notre nouvel algorithme pour la mise à jour d'un ensemble de tri-concepts à partir d'une *folksonomie*. Notre algorithme garantit une adaptation à la dynamique des *folksonomies* suite à l'insertion de nouveaux triplets sans avoir à

recalculer l'ensemble de tri-concepts à partir d'une *folksonomie* qui est une étape assez coûteuse.

5.4 Pseudo code

Dans ce qui suit, nous présentons d'abord, quelques notions de base avant d'introduire notre nouvel algorithme pour la mise à jour des tri-concepts dans les *folksonomies*.

5.4.1 Notions de base

La définition suivante rappelle celle introduite dans [Jäschke *et al.*, 2008] démontrant les comparaisons entre tri-concepts (triplets en général).

Définition 16 *Pour deux tri-concepts (triplets en général)* (A_1,A_2,A_3) *et* (B_1,B_2,B_3)*, et à partir des trois dimensions* $\mathcal{U}$*,* $\mathcal{T}$ *et* $\mathcal{R}$*, nous pouvons obtenir un quasi-ordre* $\lesssim_{\mathcal{U}}$*,* $\lesssim_{\mathcal{T}}$ *et* $\lesssim_{\mathcal{R}}$ *sur l'ensemble de tous les tri-concepts (triplets en général). Pour* $i = 1, 2, 3$*,* $(A_1,A_2,A_3) \lesssim_i (B_1,B_2,B_3)$ *ssi* $A_i \subseteq B_i$*.*

Exemple 12 *Considérons la* folksonomie *décrite par le Tableau 5.1 et les deux tri-concepts :* TC_2=($\{u_2, u_3, u_4\}$, $\{t_1, t_4\}$, $\{r_2, r_3\}$) *et* TC_3=($\{u_2, u_3\}$, $\{t_1, t_2, t_3, t_4\}$, $\{r_2, r_3\}$)*. Nous avons* $TC_2 \lesssim_{\mathcal{T},\mathcal{R}} TC_3$ *puisque* $\{t_1, t_4\} \subseteq \{t_1, t_2, t_3, t_4\}$ *et* $\{r_2, r_3\} \subseteq \{r_2, r_3\}$*. Cependant,* $TC_2 \lesssim_{\mathcal{U}} TC_3$ *étant donné que* $\{u_2, u_3, u_4\} \nsubseteq \{u_2, u_3\}$

$\mathcal{F}/\mathcal{R}$	r_1				r_2				r_3			
$\mathcal{U}/\mathcal{T}$	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		×	×	×		×	×	×		×	×	×
u_2		×	×	×	×	×	×	×	×	×	×	×
u_3		×	×	×	×	×	×	×	×	×	×	×
u_4		×	×		×			×	×			×

Tableau 5.1 — Un exemple d'une *folksonomie*.

À partir de la *folksonomie* décrite par le Tableau 5.1 et les seuils de supports suivants : $minsupp_u$=2, $minsupp_t$=2 et $minsupp_r$=1, les tri-concepts fréquents extraits sont :

TC_1=($\{u_1, u_2, u_3\}$, $\{t_2, t_3, t_4\}$, $\{r_1, r_2, r_3\}$)	TC_2=($\{u_2, u_3, u_4\}$, $\{t_1, t_4\}$, $\{r_2, r_3\}$)
TC_3=($\{u_2, u_3\}$, $\{t_1, t_2, t_3, t_4\}$, $\{r_2, r_3\}$)	TC_4=($\{u_1, u_2, u_3, u_4\}$, $\{t_2, t_3\}$, $\{r_1\}$)

5.4.2 Intuition derrière l'algorithme

L'objectif de notre algorithme est de mettre à jour un ensemble de tri-concepts déjà extraits en passant un minimum de fois par la *folksonomie*. Pendant la conception de notre algorithme, nous avons pu distinguer quatre cas différents :

1. la mise à jour de tri-concepts déjà existants : cela revient à vérifier si les anciens tri-concepts sont inclus dans les nouveaux tri-concepts par rapport à exactement deux des trois dimensions (*cf.,* Algorithme 6, Lignes 17-19).
2. la découverte de nouveaux tri-concepts : ce cas est vérifié lorsqu'un nouveau triplet est incomparable avec tous les tri-concepts (*cf.,* Algorithme 6, Lignes 3-9). Un nouveau triplet est dit incomparable avec un tri-concept lorsqu'il est inclus dans ce tri-concept par rapport à deux des trois dimensions.
3. la maintenance de tri-concepts inchangés : qui ne sont pas mis à jour mais qui vérifient toujours la propriété de fermeture.
4. la suppression de tri-concepts : qui ne vérifient plus la propriété de fermeture (*cf.,* Algorithme 6, Lignes 14-16).

5.4.3 Pseudo code

Le pseudo code de notre algorithme FolkIncr est décrit par l'Algorithme 6. FolkIncr prend en entrée un ensemble non vide $\mathcal{TC}$ de tri-concepts fréquents, un ensemble $\mathcal{N}$ de nouveaux triplets, une *folksonomie* $\mathcal{F}$ et les seuils de supports minimaux $minsupp_u$, $minsupp_t$ et $minsupp_r$. FolkIncr donne en sortie un ensemble mis à jour $\mathcal{TC}$ de tri-concepts fréquents. FolkIncr opère comme suit : la première étape consiste à lier les nouveaux triplets avec les tri-concepts fréquents. Ainsi, pour chaque nouveau triplet t, FolkIncr parcourt l'ensemble des tri-concepts fréquents $\mathcal{TC}$ (lignes 3-9) : si t est inclus dans un tri-concept déjà existant TC par rapport à **seulement** deux des trois dimensions (*i.e.,* $t \lesssim_{j,k} TC$) (*cf.,* définition 16), alors FolkIncr crée un nouveau triplet g qui est égal, au début, à t (ligne 6). Par suite, l'ensemble de g correspondant à la dimension i est augmentée par l'ensemble de TC relatif à la même dimension (*cf.,* la fonction Augmenter_dim_i, ligne 7). Par exemple, lier le triplet (u_1, t_1, r_1) avec le tri-concept ($\{u_1, u_2, u_3\}$, $\{t_2, t_3, t_4\}$, $\{r_1, r_2, r_3\}$) résulte en un nouveau triplet ($\{u_1\}$, $\{t_1, t_2, t_3, t_4\}$, $\{r_1\}$) puisque le triplet considéré est inclus dans le tri-concept par rapport aux dimensions $\mathcal{U}$ et $\mathcal{R}$ (voir d'autres exemples dans les exemples illustratifs

ci-après). Ensuite, si le nouveau triplet crée g n'appartient pas encore à l'ensemble des nouveaux tri-concepts $\mathcal{TC}'$ et qu'il n'existe pas de triplet g' tel que g soit inclus dans g' [3], alors g est ajouté à l'ensemble $\mathcal{TC}'$ (ligne 9). Ce test est réalisé afin d'éviter d'avoir des triplets redondants.

3. g est inclus dans g' veut dire que tous les ensembles de g (utilisateurs, tags et ressources) sont inclus dans ceux de g'.

Algorithme 6 : FolkIncr

Données :

1. un ensemble $\mathcal{TC}$ de tri-concepts fréquents
2. un ensemble $\mathcal{N}$ de nouveaux triplets
3. une *folksonomie* $\mathcal{F}$
4. les seuils de supports minimaux $minsupp_u$, $minsupp_t$ et $minsupp_r$

Résultats : un ensemble mis à jour $\mathcal{TC}$ de tri-concepts fréquents

```
1  début
2      g : un triplet
3      pour chaque nouveau triplet t de 𝒩 faire
4          pour chaque tri-concept tc de 𝒯𝒞 faire
5              si t ≲_i tc et t ≲_{j,k} tc alors
6                  g=t ;
7                  Augmenter_dim_i(g,i) ; //i,j,k ∈ {𝒰,𝒯,ℛ}
8                  si g ∉ 𝒯𝒞' et ∄ g' / g ⊆ g' alors
9                      mettre g dans 𝒯𝒞' ;
10     pour chaque triplet g de 𝒯𝒞' faire
11         Calcul_Fermeture(g) ;
12         si non fréquent (g, minsupp_u, minsupp_t, minsupp_r) ou ∃ g' ∈ 𝒯𝒞' /
           g = g' alors
13             élaguer(g) ;
14     pour chaque tri-concept tc de 𝒯𝒞 faire
15         si not fermé(tc) alors
16             élaguer (tc) ;
17         pour chaque nouveau tri-concept tc' de 𝒯𝒞' faire
18             si tc ≲_i tc' et tc ≲_{j,k} tc' alors
19                 Augmenter_dim_i(tc,i) ;
20     𝒯𝒞 = 𝒯𝒞 ∪ 𝒯𝒞' ;
21     retourner 𝒯𝒞 ;
22 fin
```

Une fois les nouveaux triplets créés, la fonction CALCUL_FERMETURE est appelée afin de calculer leurs fermetures respectives (ligne 11). À cette fin, nous utilisons l'opérateur de fermeture dédié à un contexte triadique introduit dans [Trabelsi *et al.*, 2012]. À ce niveau, nous avons besoin de passer par la *folksonomie* pour pouvoir calculer les fermetures respectives. Ensuite, nous devons vérifier si chaque nouveau triplet g est fréquent par rapport aux trois seuils minimaux de support. Si g est infréquent ou appartient déjà à l'ensemble $\mathcal{TC}'$, alors il est élagué (lignes 12-13). Après avoir calculé la fermeture des nouveaux triplets, FOLKINCR vérifie si les anciens tri-concepts sont toujours fermés. Ainsi, un test de fermeture est réalisé (lignes 14-16) afin d'élaguer les tri-concepts qui ne sont plus fermés, *i.e.,* qui deviennent inclus dans d'autres tri-concepts (*cf.,* la fonction NON FERMÉ). Enfin, quelques anciens tri-concepts gardent la propriété de fermeture mais ont besoin d'être augmentés : ainsi, si un ancien tri-concept tc est inclus dans un nouveau tri-concept tc' par rapport à deux dimensions j et k, alors, l'ensemble de tc correspondant à la dimension i est augmenté par l'ensemble correspondant de tc' relatif à la même dimension (lignes 18-19). FOLKINCR arrive à terme lorsque tous les anciens tri-concepts sont vérifiés et notre algorithme retourne l'ensemble mis à jour des tri-concepts fréquents $\mathcal{TC}$.

5.5 Exemples illustratifs de mise à jour

Considérons la *folksonomie* décrite par le Tableau 5.1 et les seuils minimaux de supports suivants : $minsupp_u$=2, $minsupp_t$=2 et $minsupp_r$=1. Les tri-concepts fréquents extraits sont :

1. $(\{u_1, u_2, u_3\}, \{t_2, t_3, t_4\}, \{r_1, r_2, r_3\})$;
2. $(\{u_2, u_3, u_4\}, \{t_1, t_4\}, \{r_2, r_3\})$;
3. $(\{u_2, u_3\}, \{t_1, t_2, t_3, t_4\}, \{r_2, r_3\})$;
4. $(\{u_1, u_2, u_3, u_4\}, \{t_2, t_3\}, \{r_1\})$

Considérons à présent le premier scénario suivant : les utilisateurs u_1, u_2, u_3 et u_4 ont tagué la ressource r_1 avec le tag t_1. Ainsi, nous avons quatre nouveaux triplets :

triplet 1 (u_1, t_1, r_1)

triplet 2 (u_2, t_1, r_1)

triplet 3 (u_3, t_1, r_1)

triplet 4 (u_4, t_1, r_1)

La première étape de l'algorithme FolkIncr est de lier les nouveaux triplets avec les tri-concepts fréquents. Par exemple, lier le premier triplet avec l'ensemble des tri-concepts fréquents résulte en deux nouveaux triplets : $(\{u_1\}, \{t_1, t_2, t_3, t_4\}$ et $(\{u_1\}, \{t_1, t_2, t_3\}, \{r_1\})$. Le premier nouveau triplet a un ensemble de tags qui a été augmenté par l'ensemble $\{t_2, t_3, t_4\}$ étant donné que nous avons (tripler 1) $\lesssim_j TC_1$ et (triplet 1) $\lesssim_{i,k} TC_1$ tandis que le second nouveau triplet est élagué puisqu'il est inclus dans le premier nouveau triplet. Ainsi, lier les quatre nouveaux triplets avec l'ensemble des tri-concepts résulte en 7 nouveaux triplets :

1. $(\{u_1\}, \{t_1, t_2, t_3, t_4\}, \{r_1\})$
2. $(\{u_2\}, \{t_1, t_2, t_3, t_4\}, \{r_1\})$
3. $(\{u_2\}, \{t_1\}, \{r_1, r_2, r_3\})$
4. $(\{u_3\}, \{t_1, t_2, t_3, t_4\}, \{r_1\})$
5. $(\{u_3\}, \{t_1\}, \{r_1, r_2, r_3\})$
6. $(\{u_4\}, \{t_1\}, \{r_1, r_2, r_3\})$
7. $(\{u_4\}, \{t_1, t_2, t_3\}, \{r_1\})$

Ensuite, la seconde étape consiste à calculer la fermeture de chacun des nouveaux triplets :

- $(\{u_1\}, \{t_1, t_2, t_3, t_4\}, \{r_1\})$ --→ $(\{u_1, u_2, u_3\}, \{t_1, t_2, t_3, t_4\}, \{r_1\})$
- $(\{u_2\}, \{t_1, t_2, t_3, t_4\}, \{r_1\})$ --→ $(\{u_1, u_2, u_3\}, \{t_1, t_2, t_3, t_4\}, \{r_1\})$ (**élagué**, déjà calculé)
- $(\{u_2\}, \{t_1\}, \{r_1, r_2, \{r_3\})$ --→ $(\{u_2, u_3, u_4\}, \{t_1\}, \{r_1, r_2, r_3\})$ (**élagué**, infréquent par rapport à l'ensemble de tags)
- $(\{u_3\}, \{t_1, t_2, t_3, t_4\}, \{r_1\})$ --→ $(\{u_1, u_2, u_3\}, \{t_1, t_2, t_3, t_4\}, \{r_1\})$ (**élagué**, déjà calculé)
- $(\{u_3\}, \{t_1\}, \{r_1, r_2, r_3\})$ --→ $(\{u_2, u_3, u_4\}, \{t_1\}, \{r_1, r_2, r_3\})$ (**élagué**, infréquent par rapport à l'ensemble de tags)
- $(\{u_4\}, \{t_1\}, \{r_1, r_2, r_3\})$ --→ $(\{u_2, u_3, u_4\}, \{t_1\}, \{r_1, r_2, r_3\})$ (**élagué**, infréquent par rapport à l'ensemble de tags)
- $(\{u_4\}, \{t_1, t_2, t_3\}, \{r_1\})$ --→ $(\{u_1, u_2, u_3, u_4\}, \{t_1, t_2, t_3\}, \{r_1\})$

Ainsi, à la fin de cette étape, nous obtenons deux triplets fréquents : $(\{u_1, u_2, u_3\}, \{t_1, t_2, t_3, t_4\}, \{r_1\})$ et $(\{u_1, u_2, u_3, u_4\}, \{t_1, t_2, t_3\}, \{r_1\})$. L'étape suivante consiste à

élaguer les anciens tri-concepts fréquents qui ne vérifient plus la propriété de fermeture. C'est le cas du tri-concept ($\{u_1, u_2, u_3, u_4\}$, $\{t_2, t_3\}$, $\{r_1\}$) qui perd sa propriété de fermeture. Il est donc élagué puisqu'il est inclus dans le nouveau tri-concept ($\{u_1, u_2, u_3, u_4\}$, $\{t_1, t_2, t_3\}$,$\{ r_1\}$). Enfin, nous vérifions si d'anciens tri-concepts ont besoin d'être augmentés. C'est le cas du tri-concept ($\{u_2, u_3\}$, $\{t_1, t_2, t_3, t_4\}$, $\{r_2, r_3\}$). En effet, il est inclus dans le nouveau tri-concept ($\{u_1, u_2, u_3, u_4\}$, $\{t_1, t_2, t_3\}$, $\{r_1\}$) par rapport aux deux dimensions "utilisateurs" et "tags". Ainsi, la ressource r_1 est ajoutée à l'ensemble de ressources du troisième tri-concept. L'ensemble ainsi mis à jour des tri-concepts fréquents est comme suit :

1. TC_1=($\{u_1, u_2, u_3\}$, $\{t_2, t_3, t_4\}$, $\{r_1, r_2, r_3\}$)
2. TC_2=($\{u_2, u_3, u_4\}$, $\{t_1, t_4\}$, $\{r_1, r_2, r_3\}$)
3. TC_3=($\{u_2, u_3\}$, $\{t_1, t_2, t_3, t_4\}$, $\{r_1, r_2, r_3\}$)
4. TC_4=($\{u_1, u_2, u_3\}$, $\{t_1, t_2, t_3, t_4\}$, $\{r_1\}$)
5. TC_5=($\{u_1, u_2, u_3, u_4\}$, $\{t_1, t_2, t_3\}$, $\{r_1\}$)

Considérons à présent le deuxième scénario suivant : les utilisateurs u_2 et u_3 ont tagué la ressource r_1 avec le tag t_1 tandis que l'utilisateur u_4 a tagué la même ressource avec les tags t_1 et t_4. Ainsi, nous obtenons quatre nouveaux triplets :

triplet 1 (u_2, t_1, r_1)

triplet 2 (u_3, t_1, r_1)

triplet 3 (u_4, t_1, r_1)

triplet 4 (u_4, t_4, r_1)

La première étape de l'algorithme FOLKINCR consiste à lier ces quatre nouveaux triplets avec les tri-concepts fréquents déjà extraits. Après élagage des triplets résultants qui sont redondants, nous obtenons 9 nouveaux triplets :

- ($\{u_2\}$, $\{t_1, t_2, t_3, t_4\}$, $\{r_1\}$)
- ($\{u_2\}$, $\{t_1\}$, $\{r_1, r_2, r_3\}$)
- ($\{u_3\}$, $\{t_1, t_2, t_3, t_4\}$, $\{r_1\}$)
- ($\{u_3\}$, $\{t_1\}$, $\{r_1, r_2, r_3\}$)
- ($\{u_4\}$, $\{t_1\}$, $\{r_1, r_2, r_3\}$)
- ($\{u_4\}$, $\{t_1, t_2, t_3\}$, $\{r_1\}$)
- ($\{u_1, u_2, u_3, u_4\}$, $\{t_4\}$, $\{r_1\}$)

- $(\{u_4\}, \{t_4\}, \{r_1, r_2, r_3\})$
- $(\{u_4\}, \{t_2, t_3, t_4\}, \{r_1\})$

Par suite, nous calculons pour chaque nouveau triplet sa fermeture correspondante, ce qui donne les résultats suivants :

- $(\{u_2\}, \{t_1, t_2, t_3, t_4\}, \{r_1\})$ --→ $(\{u_2, u_3, u_4\}, \{t_1, t_2, t_3, t_4\}, \{r_1\})$
- $(\{u_2\}, \{t_1\}, \{r_1, r_2, r_3\})$ --→ $(\{u_2, u_3, u_4\}, \{t_1, t_4\}, \{r_1, r_2, r_3\})$
- $(\{u_3\}, \{t_1\}, t_2, t_3, t_4\}, \{r_1\})$ --→ $(\{u_2, u_3, u_4\}, \{t_1, t_2, t_3, t_4\}, \{r_1\})$ (**élagué**, déjà calculé)
- $(\{u_3\}, \{t_1\}, \{r_1, r_2, r_3\})$ --→ $(\{u_2, u_3, u_4\}, \{t_1, t_4\}, \{r_1, r_2, r_3\})$ (**élagué**, déjà calculé)
- $(\{u_4\}, \{t_1\}, \{r_1, r_2, r_3\})$ --→ $(\{u_2, u_3, u_4\}, \{t_1, t_4\}, \{r_1, r_2, r_3\})$ (**élagué**, déjà calculé)
- $(\{u_4\}, \{t_1, t_2, t_3\}, \{r_1\})$ --→ $(\{u_2, u_3, u_4\}, \{t_1, t_2, t_3, t_4\}, \{r_1\})$ (**élagué**, déjà calculé)
- $(\{u_1, u_2, u_3, u_4\}, \{t_4\}, \{r_1\})$ --→ $(\{u_1, u_2, u_3, u_4\}, \{t_2, t_3, t_4\}, \{r_1\})$
- $(\{u_4\}, \{t_4\}, \{r_1, r_2, r_3\})$ --→ $(\{u_1, u_2, u_3, u_4\}, \{t_4\}, \{r_1, r_2, r_3\})$ (**élagué**, infréquent par rapport à l'ensemble de tags)
- $(\{u_4\}, \{t_2, t_2, t_4\}, \{r_1\})$ --→ $(\{u_1, u_2, u_3, u_4\}, \{t_2, t_3, t_4\}, \{r_1\})$ (**élagué**, déjà calculé)

Ainsi, nous obtenons trois nouveaux tri-concepts fréquents à la fin de cette étape : $(\{u_2, u_3, u_4\}, \{t_1, t_2, t_3, t_4\}, \{r_1\})$, $(\{u_2, u_3, u_4\}, \{t_1, t_4\}, \{r_1, r_2, r_3\})$ et $(\{u_1, u_2, u_3, u_4\}, \{t_2, t_3, t_4\}, \{r_1\})$. Ensuite, nous élaguons les anciens tri-concepts fréquents qui ne vérifient plus la propriété de fermeture. C'est le cas des tri-concepts $(\{u_1, u_2, u_3\}, \{t_2, t_3, t_4\}, \{r_1, r_2, r_3\})$ et $(\{u_1, u_2, u_3, u_4\}, \{t_2, t_3\}, \{r_1\})$ qui sont élagués. Enfin, le troisième tri-concept $(\{u_2, u_3\}, \{t_1, t_2, t_3, t_4\}, \{r_2, r_3\})$ est augmenté par rapport à la dimension "ressource" donnant naissance au tri-concept suivant : $(\{u_2, u_3\}, \{t_1, t_2, t_3, t_4\}, \{r_1, r_2, r_3\})$. L'ensemble des tri-concepts fréquents mis à jour est donc égal à :

1. $(\{u_1, u_2, u_3\}, \{t_2, t_3, t_4\}, \{r_1, r_2, r_3\})$;
2. $(\{u_2, u_3, u_4\}, \{t_1, t_4\}, \{r_1, r_2, r_3\})$;
3. $(\{u_2, u_3\}, \{t_1, t_2, t_3, t_4\}, \{r_1, r_2, r_3\})$;
4. $(\{u_1, u_2, u_3, u_4\}, \{t_2, t_3, t_4\}, \{r_1\})$;
5. $(\{u_2, u_3, u_4\}, \{t_1, t_2, t_3, t_4\}, \{r_1\})$;

Dans ce qui suit, nous étudions les propriétés de l'algorithme FolkIncr, *i.e.,* sa correction, complétude et terminaison.

5.6 Propriétés de FolkIncr

Proposition 5 *L'algorithme* FolkIncr *est correct et complet. Il met à jour correctement tous les tri-concepts fréquents.*

Preuve 2 ***Création de nouveaux tri-concepts :*** *(Algorithme 6, Lignes* 3-9*) À ce niveau, tous les nouveaux triplets sont comparés avec chaque élément de l'ensemble des tri-concepts fréquents (lignes* 3 *et* 4*). Cela donne naissance à de nouveaux triplets qui vérifient la condition de la ligne* 5*. Ensuite, la seconde boucle (lignes* 14-19*) permet à la fois de calculer la fermeture de chaque triplet et de vérifier leurs fréquences. Ainsi, ces nouveaux triplets sont fermés et fréquents, i.e., ce sont donc des tri-concepts.*

Élagage d'anciens tri-concepts : *(Algorithme 6, Lignes* 14-16*) Cette étape permet d'élaguer les anciens tri-concepts qui ne vérifient plus la propriété de fermeture. Cela assure la correction de notre algorithme, i.e., les tri-concepts qui ne sont plus fermés doivent être élagués.*

Mise à jour d'anciens tri-concepts : *(Algorithme 6, Lignes* 17-19*) Cette boucle assure la mise à jour d'anciens tri-concepts fréquents qui doivent être augmentés (selon une dimension, i.e., Ligne* 25*) suite à l'apparition des nouveaux tri-concepts. Cette étape permet d'assurer la correction de l'ensemble de sortie puisque tous les tri-concepts doivent toujours vérifier la propriété de fermeture.*

Maintenance des anciens tri-concepts : *Implicitement,* FolkIncr *assure que les anciens tri-concepts fréquents qui n'ont pas été élagués ou modifiés appartiennent toujours à l'ensemble des tri-concepts fréquents $\mathcal{TC}$ puisqu'ils sont toujours fermés et fréquents. Cela assure la complétude de notre algorithme.*

Ainsi, grâce à ces quatre cas, FolkIncr *permet de mettre à jour l'ensemble des tri-concepts fréquents $\mathcal{TC}$. Nous concluons que* FolkIncr *met à jour fidèlement tous les tri-concepts fréquents. Donc, il est correct et complet.*

Proposition 6 *L'algorithme* FolkIncr *termine.*

Preuve 3 *Le nombre de tri-concepts générés par* FolkIncr *est fini. En effet, le*

nombre de tri-concepts candidats générés à partir d'un ensemble de m tri-concepts et d'un ensemble de p nouveaux triplets est égal au maximum à $m \times p$. De plus, le nombre d'opérations pour traiter chaque tri-concept candidat est fini. Donc, FolkIncr *termine.*

Complexité Théorique

Soit m le nombre des tri-concepts fréquents déjà extraits et n le nombre de triplets de la *folksonomie*. FolkIncr parcourt quatre fois l'ensemble de tri-concepts afin de les lier avec les nouveaux triplets, calculer la fermeture des nouveaux triplets, vérifier la fermeture des anciens tri-concepts ou encore mettre à jour les anciens tri-concepts. Cependant, lorsque FolkIncr calcule la fermeture des nouveaux triplets, il a besoin de passer par la *folksonomie* m fois, *i.e.,* pour chaque nouveau triplet (Lignes 10 - 13). Ainsi, la complexité théorique de notre algorithme FolkIncr est de $\mathcal{O}(m(4+n))$. Cependant, et comme il sera démontré dans ce qui suit, d'un point de vue pratique, les performances pratiques de notre algorithme sont loin des prédictions théoriques. Pour cette raison, nous nous focaliserons sur une évaluation qui sera menée sur des jeux de données du monde réel à large échelle.

5.7 Évaluation de l'approche

Dans ce qui suit, nous allons comparer les temps d'exécution de notre algorithme FolkIncr avec le redémarrage des trois algorithmes de la littérature dédiés à la tâche d'extraction des tri-concepts, *i.e.,* TriCons, Trias et Data Peeler. Nous avons implémenté notre algorithme en *C++* (compilé avec *GCC* 4.1.2). Nous avons utilisé un processeur Intel® Core *i*5 muni d'une mémoire de 8 GB. Les tests ont été menés sur le système d'exploitation Linux (Distribution Ubuntu 13.10 64 bits).

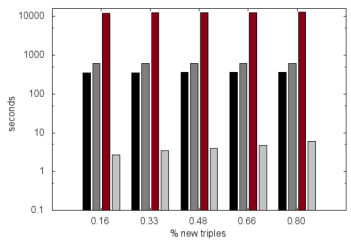
5.7.1 Jeux de données

Nous avons mené nos expérimentations sur différents *instantanés* issus de quatre jeux de données du monde réel. De plus, afin d'étudier le comportement de FolkIncr et de le comparer à ses concurrents dans différentes situations, nous avons également

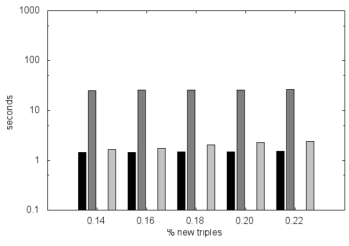
généré deux jeux de données synthétiques avec des densités prédéfinies (*cf.*, Table 5.2) [4]. Pour chaque instantané des jeux de données considérés, nous avons calculé sa densité en utilisant la formule de densité suivante : $\frac{\#triples}{\#users \times \#tags \times \#resources}$. Le Tableau 5.2 résume les caractéristiques des jeux de données utilisés. Les quatre jeux de données du monde réel utilisés pour notre évaluation sont décrits comme suit :

- Le jeu de données de filmographie MovieLens [5] dont le jeu de données utilisé pour notre évaluation est téléchargeable gratuitement [6].
- Le jeu de données de livres BookCrossing [7] dont le jeu de données utilisé est téléchargeable gratuitement [8].
- Le jeu de données Last.fm (*http ://last.fm*) est un siteweb de musique en ligne, crée en 2002. Il regroupe plus de 30 millions d'utilisateurs actifs (depuis Mars 2009). Les utilisateurs partagent leurs artistes préférés en les annotant par des tags (*e.g.*, <*Ross*, *MichaelJackson*, *kingofpop*>). Le jeu de données utilisé pour notre évaluation est téléchargeable gratuitement [9].
- Delicious (*http ://delicious.com*) est un web service social de marque page pour partager, enregistrer et découvrir des sites web ou marque page. Les triples sont des ensembles d'utilisateurs partageant des sites web à travers des tags (*e.g.*, <*Jennifer*, *www.springer.com*, *publisher*>). Le jeu de données utilisé pour notre évaluation est téléchargeable gratuitement [10].

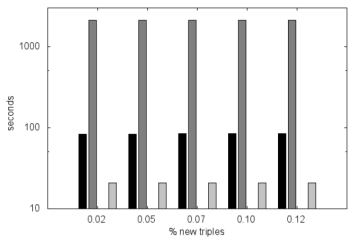
5.7.2 Temps d'exécution



MovieLens 1



MovieLens 2

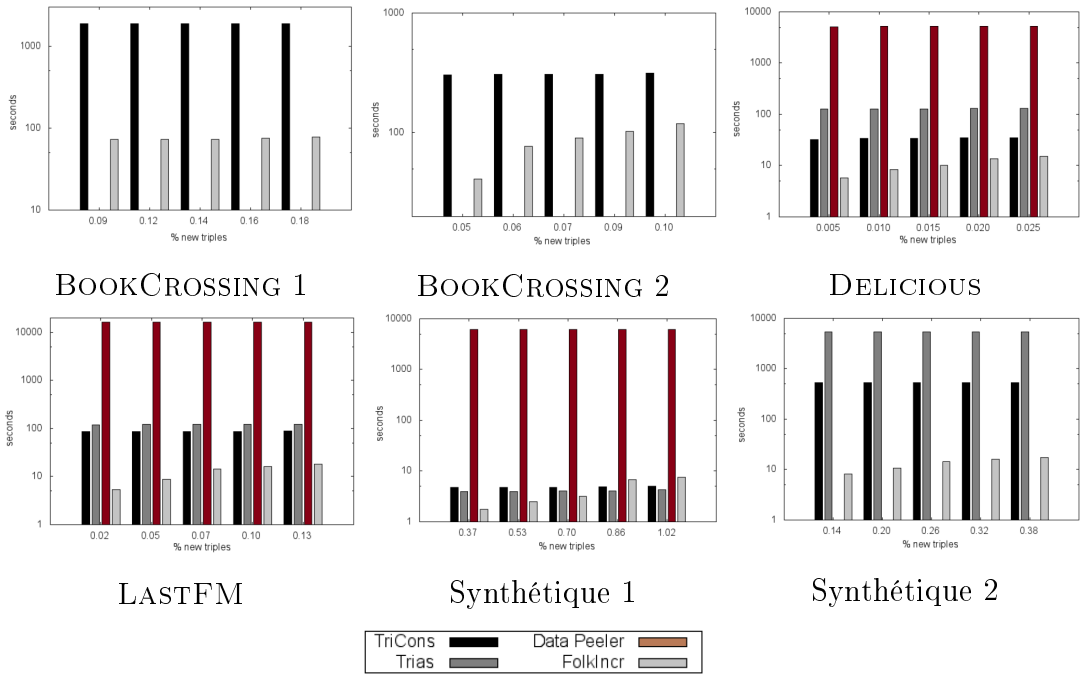


MovieLens 3

4. Nous avons utilisé un générateur de jeux de données synthétiques téléchargeable à partir de ce lien *http ://www.isima.fr/ jelassi/SyntheticGenerator*
5. *http ://movielens.umn.edu/*
6. *http ://www.grouplens.org/node/73*
7. *http ://www.bookcrossing.com/*
8. *http ://www.grouplens.org/node/74*
9. *http ://mtg.upf.edu/static/datasets/last.fm/lastfm-dataset-1K.tar.gz*
10. *http ://files.grouplens.org/datasets/hetrec2011/hetrec2011-delicious-2k.zip*

Jeu de données	Densité	# Triplets	min_u	min_t	min_r	# Tri-Concepts
MovieLens 1	0,0003%	15.000	2	1	1	2123
MovieLens 2	0,00004%	47.957	2	1	1	925
MovieLens 3	1%	100.000	12	1	2	2775
BookCrossing 1	0,001%	102.430	3	1	1	9872
BookCrossing 2	0,001%	200.000	8	1	1	4738
LastFM	0,00008%	186.474	2	1	1	1987
Delicious	0,000008%	437.593	4	1	1	3261
Synthétique 1	2%	18.560	1	1	1	175
Synthétique 2	50%	50.000	2	1	1	504

Tableau 5.2 — Caractéristiques des différents jeux de données.



BookCrossing 1 BookCrossing 2 Delicious

LastFM Synthétique 1 Synthétique 2

La Figure ci-dessus démontre les temps d'exécution de FolkIncr *vs.* le redémarrage des algorithmes TriCons, Trias et Data Peeler. Pour ces expérimentations, nous avons exécuté tous les algorithmes sur différents jeux de données. De plus, nous définissons, pour chaque jeu de données, les valeurs de supports minimaux de supports ($minsupp_u$, $minsupp_t$ and $minsupp_r$), calculons le nombre de tri-concepts extraits (*cf.*, Table 5.2) et varions le nombre de nouveaux triplets considérés.

Nous avons d'abord redémarré les algorithmes TriCons, Trias et Data Peeler sur les ensembles d'entrée mis à jour (le jeu de données de départ + l'ensemble de nouveaux triplets), puis nous avons exécuté l'algorithme FolkIncr qui prend comme

entrée l'ensemble de tri-concepts fréquents extraits par l'un des trois algorithmes dédiés à cette tâche ainsi que l'ensemble de nouveaux triplets et le jeu de données de départ. L'objectif en commun est d'extraire l'ensemble mis à jour des tri-concepts fréquents et de conclure s'il est plus judicieux d'exécuter FOLKINCR afin de mettre à jour cet ensemble ou s'il faut opter pour l'un des trois algorithmes de la littérature pour générer le même ensemble. Ainsi, la Figure démontre que FOLKINCR fait mieux que les trois algorithmes de la littérature pour la plupart des jeux de données et paramètres considérés.

Pour le jeu de données MOVIELENS (premier instantané), FOLKINCR fait mieux que les autres algorithmes pour toutes les valeurs de nouveaux triplets. En effet, par exemple, FOLKINCR prend 103,51 moins de temps que TRICONS afin de mettre à jour l'ensemble des 2123 tri-concepts fréquents, *i.e.,* un gain de temps égal à 99%. Concernant les autres instantanés du jeu de données MOVIELENS, si TRICONS fait légèrement mieux que FOLKINCR dans le second instantané (une différence d'à peine 0, 78 secondes au maximum), notre algorithme demeure meilleur que tous les autres algorithmes pour toutes les valeurs de nouveaux triplets (jusqu'à 83% de gain de temps). Pour les deux instantanés du jeu de données BOOKCROSSING, ni TRIAS ni DATA PEELER n'a été capable de régénérer l'ensemble de tri-concepts fréquents. Par ailleurs, la différence entre FOLKINCR et TRICONS augmente considérablement étant donné que le nombre de triplets du jeu de données est très élevé (entre 100.000 et 200.000) et, par conséquence, le nombre de tri-concepts fréquents est aussi élevé (respectivement 9872 et 4738). Cela a pour effet de ralentir TRICONS qui prend, par exemple, 305, 33 secondes afin de mettre à jour un ensemble de 4738 tri-concepts et 1894, 28 secondes afin de mettre à jour un ensemble de 9872 tri-concepts. En comparaison, FOLKINCR met à jour les deux mêmes ensembles en, respectivement, $7,43$ et $24,50$ moins de temps avec un gain de temps de, respectivement, 86% et 95%. En d'autres termes, nous avons amélioré entre 86% et 95% le temps pour mettre à jour l'ensemble de tri-concepts fréquents en utilisant FOLKINCR au lieu de redémarrer TRICONS. Cela revient au fait que le parcours de l'ensemble de tri-concepts fréquents (opéré par FOLKINCR) est plus rapide que le parcours de l'ensemble du jeu de données (opéré par TRICONS) pour régénérer l'ensemble de tri-concepts fréquents. Alors que TRICONS a besoin de re-scanner l'entière *folksonomyie* et de calculer la fermeture de chaque triplet, ce qui prend un temps considérable, la tâche de FOLKINCR revient seulement à lier l'ensemble de nouveaux triplets avec l'ensemble de tri-concepts fréquents puis de calculer la fermeture seulement pour l'en-

semble résultant. Par exemple, pour le premier instantané de BookCrossing, tandis que TriCons a besoin de régénérer et de calculer la fermeture de chacun des 9772 tri-concepts fréquents, FolkIncr calcule uniquement la fermeture de 24 triplets qui sont le résultats de l'opération de liaison entre les tri-concepts fréquents et les nouveaux triplets. Concernant le jeu de données LastFM, FolkIncr fait mieux que tous les autres algorithmes et s'exécute entre $5,40$ et $17,99$ secondes pour toutes valeurs de nouveaux triplets, ce qui demeure un temps d'exécution raisonnable. Par exemple, pour mettre à jour 3261 tri-concepts, FolkIncr est 15, 87 plus rapide que son principal concurrent, *i.e.,* TriCons, avec un gain de temps égal à 93% par rapport à ce dernier. Les mêmes statistiques peuvent être observées sur le jeu données Delicious où FolkIncr est meilleur que TriCons, Trias et Data Peeler, *i.e.,* un gain de temps moyen égal à 88%. Enfin, si pour le premier jeu de données synthétique, les performances de FolkIncr, TriCons et Trias sont assez proches, l'avantage tourne court pour FolkIncr concernant le second jeu de données synthétique. De tels résultats sont expliqués par le fait que les deux jeux de données son plus denses que les autres jeux de données du monde réel, donc, cela prend plus de temps à FolkIncr pour traiter le nombre impressionnant de candidats générés avant de calculer les tri-concepts fréquents. Ainsi, si le nombre de triplets du jeu de données est assez élevé, cela revient moins cher d'opter pour FolkIncr afin de mettre à jour l'ensemble de tri-concepts fréquents que de redémarrer l'un des algorithmes de la littérature dédiés à cette tâche. Cependant, l'une des limites de notre approche est lorsque les deux ensembles de tri-concepts fréquents et de nouveaux triplets sont très grands, le nombre de comparaisons et de liaisons devient très grand et cela prend beaucoup plus de temps à FolkIncr pour générer et calculer la fermeture de chaque candidat. Par exemple, pour BookCrossing, lorsque le nombre de tri-concepts fréquents est égal à 9872 et le nombre de nouveaux triplets est égal à 200, cela résulte en 1554600 comparaisons ce qui résulte ensuite en le calcul de la fermeture de 30 nouveaux triplets, ce qui a pour effet de ralentir FolkIncr malgré le fait qu'il demeure plus rapide que les autres algorithmes.

5.7.3 Expérimentations qualitatives

Les tri-concepts fréquents que nous cherchons à extraire ou/et à mettre à jour peuvent être intéressants et utiles dans de nombreux domaines, *e.g.,* la recommandation

de ressources, la suggestion de tags ou encore la proposition d'amis. Par exemple, si nous nous intéressons à la proposition d'amis, les meilleurs tri-concepts, d'un point de vue qualitatif, sont ceux ayant les ensembles de tags et de ressources les plus grands. En effet, de tels concepts permettent de regrouper les utilisateurs ayant le maximum de tags et de ressources en commun. Ainsi, il sera recommandé aux utilisateurs d'une *folksonomie* donnée une liste d'amis ayant partagé un maximum de tags et de ressources en commun. Cependant, si nous traitons le domaine de recommandation de ressources, les concepts les plus intéressants seraient ceux qui regroupent un ensemble maximal d'utilisateurs. En effet, il est plus intéressant de recommander des ressources qui ont été partagés par un très grand nombre d'utilisateurs. Ainsi, les tri-concepts les plus intéressants, d'un point de vue qualité, dépendent du domaine dans lequel nous voulons les appliquer.

5.8 Conclusion

Avec l'évolution constante des *folksonomies*, cela revient de plus en plus cher de redémarrer le processus d'extraction de tri-concepts pour manager chaque mise à jour. Pour cette raison, nous avons proposé un nouvel algorithme dont le but est de mettre à jour l'ensemble des tri-concepts extraits suite à l'ajout de nouveaux triplets. Par ailleurs, il sera possible d'étendre notre solution au cas quadri-dimensionnel afin de mettre à jour un ensemble de quadri-concepts. L'évaluation expérimentale que avons mené a comparé les performances de notre algorithme à ceux de la littérature dédiés à la tâche d'extraction des tri-concepts. Ces expérimentations ont démontré qu'il revient moins cher d'opter pour notre algorithme de mise à jour de concepts que de relancer un algorithme de la littérature pour mettre à jour le même ensemble. Cette mise à jour permettra de prendre en compte notamment les nouveaux tags et ressources qui viennent d'être ajoutés à la *folksonomie* et de proposer aux utilisateurs des recommandations récentes qui prennent en compte les dernières mises à jour du système. Une des limites de notre approche est qu'elle a besoin de passer par la *folksonomie* (le jeu de données d'entrée) afin de vérifier la fermeture d'un tri-concept candidat. Ainsi, nous aspirons à améliorer notre algorithme afin de prendre en compte plus de nouveaux triplets et d'éviter d'utiliser la folksonomie pour vérifier la fermeture des candidats.

Conclusion générale et perspectives

Conclusion

La tâche d'extraction des quadri-concepts à partir des folksonomies se propose de représenter le contexte sous forme de quadruplets maximaux <utilisateurs, tags, ressources, variables>. Cette dernière dimension nouvellement introduite dans cette thèse peut être représentée sous différents aspects : le temps si on veut étudier la dynamique temporelle des folksonomies ou encore le profil dans un cas de recommandations personnalisés. Dans cette thèse, nous avons d'abord proposé un algorithme qui permet d'extraire les quadri-concepts et qui soit plus performant que son prédécesseur de la littérature. Ensuite, il s'agit de considérer la quatrième dimension en fonction du besoin (profil, temps, etc.) pour proposer un système personnalisé de recommandation aux utilisateurs. Concernant la première approche, le but était de proposer un algorithme appelé QuadriCons dédié pour la tâche d'extraction des quadri-concepts. A cet effet, nous avons proposé un nouvel opérateur de fermeture et propose de localiser les quadri-générateurs en premier lieu pour faciliter l'extraction des quadri-concepts ensuite. Ainsi, le principal point fort de Data Peeler, le principal concurrent de la littérature, *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. De plus, QuadriCons, contrairement à Data Peeler, ne stocke pas le jeu de données en mémoire avant l'extraction des quadri-concepts. Enfin, QuadriCons génère moins de candidats grâce à l'habile détection des quadri-générateurs qui réduisent considérablement

l'espace de recherche. L'étude expérimentale faite sur deux jeux de données réelles a confirmé l'intérêt de cette approche, qui a donné lieu à une publication à RJCIA'14 [Jelassi, 2014].

Comme deuxième approche dans cette thèse, nous nous sommes appuyés sur l'algorithme QuadriCons afin de proposer un nouveau système personnalisé de recommandation appelé PersoRec. Ainsi, grâce aux quadri-concepts extraits, nous proposons une nouvelle approche afin de personnaliser les recommandations faites aux utilisateurs des folksonomies. La personnalisation des recommandations est réalisée en prenant en compte le profil des utilisateurs. Cette nouvelle donnée permet de proposer aux utilisateurs des tags ou/et ressources plus adaptées à ses besoins. Cela permet notamment d'éviter de proposer les mêmes recommandations à tous les utilisateurs et de personnaliser les réponses faites aux utilisateurs. Une étude critique des approches proposées dans la littérature montre que la plupart des travaux propose aux utilisateurs les tags et ressources les plus utilisés (dits populaires) dans les folksonomies. Si certaines approches se différencient des autres en tenant compte le profil des utilisateurs (principalement son historique de tagging), ils échouent à l'étape de cold start ou de couverture de l'espace. En effet, la plupart des approches sont incapables de proposer une recommandation aux nouveaux utilisateurs ou de tenir compte de nouvelles données qui arrivent au fur et à mesure dans les folksonomies. Ainsi, la prise en compte du profil des utilisateurs au moment du processus de recommandation, ainsi que le recours aux quadri-concepts afin de regrouper les utilisateurs ayant partagé le maximum de tags et de ressources en commun tout en ayant des profils équivalents, a permis d'améliorer les recommandations faites aux utilisateurs. Ces recommandations concernent à la fois les ressources mais également la suggestion de tags affectés à ces dernières et enfin la proposition d'utilisateurs susceptibles d'être amis avec un utilisateur cible. Prenant en entrée l'utilisateur cible ainsi que son profil, notre algorithme proposé, *i.e.,* PersoRec dont l'amélioration par l'introduction d'une mesure de ranking a donné naissance à l'algorithme FolkRec, est donc capable de générer une recommandation personnalisée pour chaque utilisateur selon le mode de recommandation qu'il désire et selon le profil qu'il possède. Ce travail a donné lieu à deux publications à WI&C@WWW'13 [**?**], IC'13 [Jelassi *et al.*, 2013a] et une démonstration à PFIA'15 et EGC'16. Nous avons

ensuite proposé d'étudier les propriétés du nouveau système personnalisé de recommandation proposé FolkRec. En effet, des expérimentations ont été menées sur des

jeux de données du monde réel tel que le réseau social BookCrossing et le système de recommandation filmographique MovieLens ont données des résultats satisfaisants tout d'abord en terme de précision et de rappel où il affiche de meilleurs scores que ses prédécesseurs sur les mêmes jeux de données. Ensuite, nous avons évalué le système de recommandation FolkRec suivant un certain nombre de propriétés définis dans la littérature. Ainsi, FolkRec, contrairement à ses concurrents, réussit à trouver une solution au problème de Cold Star en proposant aux nouveaux utilisateurs des recommandations en prenant en compte comme information uniquement le profil démographique qu'ils entrent en arrivant sur le réseau social. De plus, cela permet une bonne couverture de l'espace puisqu'il propose des recommandations à l'ensemble des utilisateurs. D'autres propriétés ont été démontrées à l'aide de métriques de qualité : tout d'abord, la sérendipité où il a été démontré que le système est capable de surprendre l'utilisateur par des recommandations auxquelles il ne s'attendait pas. Ensuite, la même métrique démontre la nouveauté de FolkRec qui est capable de proposer des recommandations nouvelles que l'utilisateur ne connaît pas encore et qu'il n'a jamais partagées. D'autres métriques ont également été testées afin de vérifier l'efficacité du système : la couverture de l'espace, l'adaptivité qui mesure à quel point un système s'adapte aux changements dans le profil des utilisateurs ou encore la diversité qui mesure à quel point les recommandations proposées peuvent être diverses. En effet, par exemple, un utilisateur va préférer cinq livres différents de cinq auteurs différents plutôt que cinq livres d'un même auteur. Enfin, la scalabilité (passage à l'échelle) a été étudiée afin de mesurer la rapidité de réponse des recommandations. Ce travail a donné lieu à deux publications dans les conférences IC'14 [Jelassi *et al.*, 2014] et CaRR@ECIR'15 [Jelassi *et al.*, 2015a] ainsi que dans la revue internationale *Social Networks Analysis and Mining* [Jelassi *et al.*, 2015b]. Enfin, pour la dernière approche de cette thèse, nous nous sommes penchés sur le problème de mise à jour des données dans les *folksonomies*. Nous avons donc proposé un algorithme appelé FolkIncr qui est capable d'incruster de nouvelles données dans un ensemble de concepts déjà extraits. Cela a été motivé par la dynamique des folksonomies où quotidiennement des milliers de données sont ajoutés sans que cela ne soit pris en compte par les algorithmes dits "statiques" de la littérature. Cela a pour effet que les recommandations de FolkRec ne s'appuient plus sur un ensemble statique de données mais sur un ensemble régulièrement mis à jour offrant ainsi aux utilisateurs les recommandations les plus récentes.

Perspectives

Si notre travail nous a permis d'obtenir des résultats encourageants, plusieurs perspectives et améliorations sont encore envisagées :

- la première des perspectives consiste à proposer un algorithme incrémental et ainsi se passer de la *folksonomie* lors du processus de mise à jour des données. Cet algorithme pourra agir directement sur les tri-concepts pour les mettre à jour sans devoir repasser par la folksonomie pour calculer la fermeture des nouveaux tri-concepts. Cela permettra notamment d'accélérer ce processus et donc de prendre en compte plus de nouveaux triplets, comme c'est le cas dans les grand jeux de données.
- la deuxième perspective est d'approfondir la modélisation de la variable de la quatrième dimension par le temps. Le nouveau système de recommandation qui en découlera s'appuiera sur les règles d'association afin de prédire les recommandations. Une étude expérimentale sur des jeux de données du monde réel est nécessaire pour prouver l'efficacité de cette nouvelle approche.
- toujours dans l'optique de valoriser nos contributions, de nouvelles expérimentations sur des structures différentes des folksonomies mais très utilisées dans le monde (*e.g.,* Amazon), et donc nettement plus volumineux que ceux pris en compte dans l'étude expérimentale présentée dans ce manuscrit, sont nécessaires afin de jauger l'impact de nos recommandations personnalisées.
- améliorer la complexité exponentielle de l'algorithme d'extraction des quadri-concepts en réduisant le nombre de calculs nécessaire pour y parvenir.
- améliorer la plateforme fonctionnelle de notre système de recommandation pour prendre en compte les nouvelles données lors du processus de recommandation aux utilisateurs
- proposer une hiérarchisation des quadri-concepts, c'est-à-dire, des liens entre les quadri-concepts selon l'inclusion des différents ensembles. Cela aura pour effet de réduire le nombre de quadri-concepts et d'en retrouver d'autres grâce à une navigation habile entre les différents ensembles.

Annexe

Afin d'implémenter notre système personnalisé de recommandation et de générer une interface interactive pour proposer des recommandations aux utilisateurs, nous proposons de créer un site web qui gère les recommandations destinées aux utilisateurs basées sur leur profil. Pour ce faire, nous utilisons le framework php Symfony. Symfony a pour objectif d'accélérer la création et la maintenance d'applications web et de remplacer les tâches de codage répétitives.

La démonstration de notre système personnalisé de recommandation [11] démontre le processus de recommandation orienté vers un utilisateur à travers deux jeux de données du monde réel, *i.e.,* MovieLens [12] et BookCrossing [13]. Le processus de recommandation est basé à la fois sur le profil de l'utilisateur et sur les quadri-concepts, *i.e.,* les tags et ressources d'utilisateurs aux profils et intérêts proches. Comme exemple, considérons l'utilisateur *Yasmine* qui utilise le système. La Figure 5.3 illustre un aperçu de notre site web. Sur la gauche, notre système affiche le profile de *Yasmine* ainsi que les films qu'elle a partagé et la liste de ses amis, tandis qu'au centre, notre système affiche les recommandations de films pour *Yasmine*. Sur la droite de l'interface, notre système affiche la liste des amis proposés pour *Yasmine*. Ainsi, un utilisateur de notre site web reçoit des recommandations basés à la fois sur son profil et sur les tags et ressources partagés par des utilisateurs ayant le même profil. De plus, il/elle recevra une liste d'amis proposés qu'il/elle pourra ou pas ajouter à sa liste d'amis. Chaque utilisateur est capable de partager des ressources (*e.g.,* films ou livres), d'ajouter des amis ou encore d'annoter des ressources avec des tags (librement choisis ou ajoutés à partir de la liste de tags suggérés). Ainsi, par exemple, lorsque *Yasmine* souhaite partager le film *Raiders of Lost Ark*, notre système personnalisé de recommandation lui suggère des

11. disponible à cette adresse *http ://www.isima.fr/ jelassi/Demo.*
12. http ://movielens.umn.edu/
13. http ://www.bookcrossing.com/

tags qui ont été utilisés par des utilisateurs ayant le même profil qu'elle, *i.e.*, *indiana jones*, *archeology* et *nazis* (*c.f.*, Figure 5.4). Enfin, avant d'ajouter un utilisateur à sa liste d'amis, *Yasmine* a la possibilité de consulter son profil, *i.e.*, ses informations personnelles ainsi que les ressources qu'il a partagés comme cela est démontré par la Figure 5.5.

Figure 5.3 — Un aperçu de notre système personnalisé de recommandation pour le jeu de données MovieLens. (**gauche**) Le profil de *Yasmine*, les films qu'elle a partagé et la liste de ses amis (**centre**) les recommandations de films pour l'utilisateur *Yasmine* (**droite**) la liste d'amis proposés pour *Yasmine*.

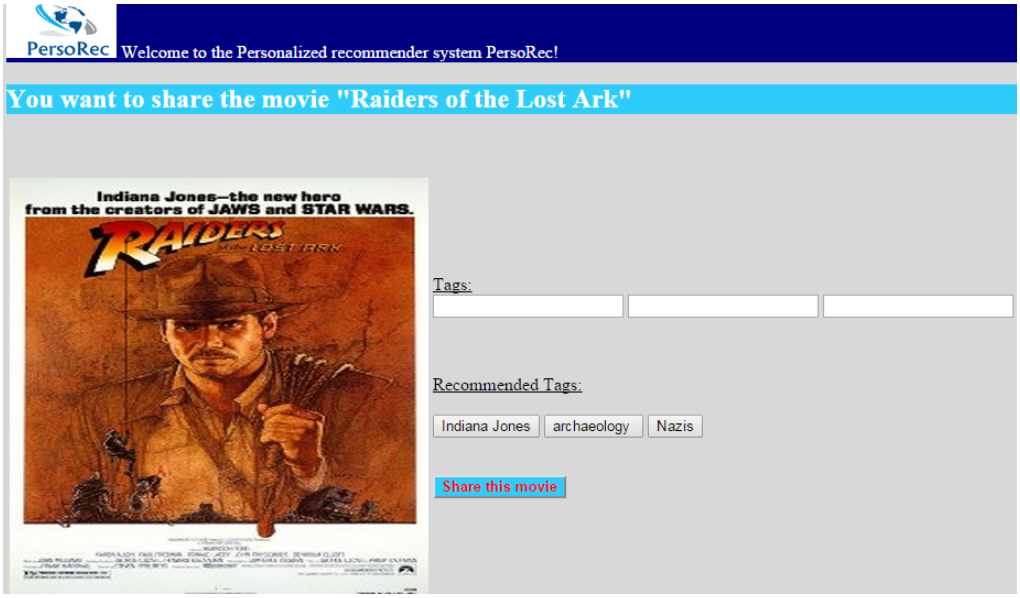


Figure 5.4 — La recommandation de tags pour *Yasmine* qui souhaite partager le film *Raiders of Lost Ark*.

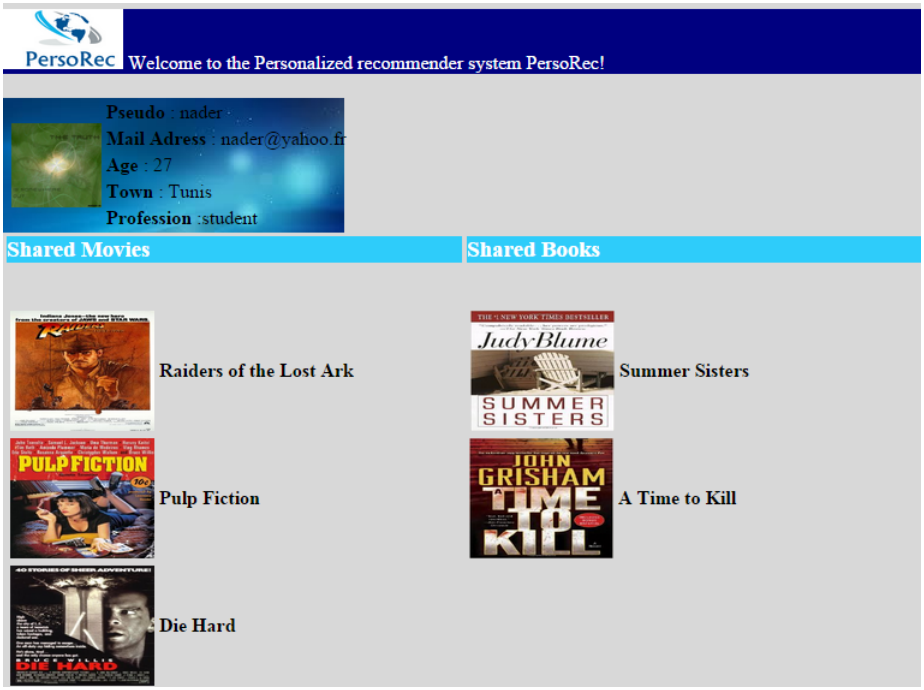


Figure 5.5 — Consultation du profil de l'utilisateur *nader* avant l'ajout à la liste des amis.

Publications scientifiques acceptées

Journal

- Mohamed Nader Jelassi, Sadok Ben Yahia et Engelbert Mephu Nguifo : *Towards more targeted recommendations in folksonomies*. Social Networks Analysis and Mining journal (SNAM). Springer Ed. 5(1) : 68 :1-68 :18 (2015). Springer Ed.
  Chiraz Trabelsi, Mohamed Nader Jelassi et Sadok Ben Yahia : *BGRT : une nouvelle base générique de règles d'association triadiques*. Application à l'auto-complétion de requêtes dans les folksonomies. Document Numérique 15(1) 2012 : 101-124. Editions Lavoisier.

Conférences et Workshops

- Mohamed Nader Jelassi : *A Quadratic Approach for Trend Detection in Folksonomies*. International Conference on Web Reasoning and Rule Systems (RR 2012) : 278-283.
- Mohamed Nader Jelassi, Sadok Ben Yahia et Engelbert Mephu Nguifo : *A personalized recommender system based on users' information in folksonomies*. International Workshop on Web Intelligence & Communities(WI&C 2013) in conjunction with the 22nd International Conference on World Wide Web (WWW 2013). 1215-1224.
- Mohamed Nader Jelassi, Sadok Ben Yahia et Engelbert Mephu Nguifo : *Nouvelle approche de recommandation personnalisée dans les folksonomies basée sur le profil des utilisateurs*. Journées francophones d'Ingénierie des Connaissances (IC 2013) : 224-226.
- Mohamed Nader Jelassi, Sadok Ben Yahia et Engelbert Mephu Nguifo : *Vers des recommandations plus personnalisées dans les folksonomies*. 25es Journées francophones d'Ingénierie des Connaissances (IC 2014) : 187-198.
- Mohamed Nader Jelassi : *Une nouvelle approche pour l'extraction efficace des quadri-concepts fréquents*. Rencontre des Jeunes Chercheurs en Intelligence Artificelle (RJCIA 2014) : 146-151.
- Mohamed Nader Jelassi, Sadok Ben Yahia et Engelbert Mephu Nguifo : *Towards more targeted recommendations in folksonomies*. 5th Workshop on Context-awareness in Retrieval and Recommendation (CaRR 2015) in conjunction with

the 37th European Conference on Information Retrieval (ECIR 2015).

– Mohamed Nader Jelassi, Sadok Ben Yahia et Engelbert Mephu Nguifo : *PERSO-REC : un système personnalisé de recommandations pour les folksonomies basé sur les concepts quadratiques* (IC 2015).

Chiraz Trabelsi, Mohamed Nader Jelassi et Sadok Ben Yahia : *Auto-complétion de requêtes par une base générique de règles d'association triadiques.* Conférence en Recherche d'Information et Applications (CORIA) 2011 : 9-24.

Chiraz Trabelsi, Mohamed Nader Jelassi et Sadok Ben Yahia : *Scalable Mining of Frequent Tri-concepts from Folksonomies.* Pacific Asia Conference on Knowledge Discovery and Data Mining (PAKDD) 2012 : 231-242.

Bibliographie

[Abnar *et al.*, 2014] Abnar, A., Takaffoli, M., Rabbany, R. et Zaïane, O. R. (2014). SSRM : structural social role mining for dynamic social networks. *In 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2014, Beijing, China, August 17-20, 2014*, pages 289–296.

[Adomavicius et Kwon, 2012] Adomavicius, G. et Kwon, Y. (2012). Improving aggregate recommendation diversity using ranking-based techniques. *IEEE Transactions on Knowledge and Data Engineering*, 24(5):896–911.

[Adomavicius et Tuzhilin, 2005] Adomavicius, G. et Tuzhilin, A. (2005). Toward the next generation of recommender systems : A survey of the state-of-the-art and possible extensions. *IEEE Trans. on Knowl. and Data Eng.*, 17(6):734–749.

[Agrawal *et al.*, 1993] Agrawal, R., Imielinski, T. et Swami, A. (1993). Mining association rules between sets of items in large databases. *In ACM-SIGMOD International Conference on Management of Data*, pages 207–216, Washington D. C., USA.

[Badache et Boughanem, 2015] Badache, I. et Boughanem, M. (2015). A priori relevance based on quality and diversity of social signals. *In Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, pages 731–734, New York, NY, USA. ACM.

[Baeza-Yates et Ribeiro-Neto, 1999] Baeza-Yates, R. A. et Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

[Basile *et al.*, 2007] Basile, P., Gendarmi, D., Lanubile, F. et Semeraro, G. (2007). Recommending smart tags in a social bookmarking system. *In Bridging the Gap between Semantic Web and Web 2.0*, pages 22–29.

[Bellogín *et al.*, 2013] Bellogín, A., Cantador, I. et Castells, P. (2013). A comparative study of heterogeneous item recommendations in social systems. *Information Sciences*, 221:142–169.

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

[Bollen *et al.*, 2010] Bollen, D., Knijnenburg, B. P., Willemsen, M. C. et Graus, M. (2010). Understanding choice overload in recommender systems. *In Proceedings of the Fourth ACM Conference on Recommender Systems*, RecSys '10, pages 63–70, New York, NY, USA. ACM.

[Cerf, 2010] Cerf, L. (2010). *Constraint-Based Mining of Closed Patterns in Noisy n-ary Relations*. Thèse de doctorat, Institut National des Sciences Appliquées (INSA), Lyon, France, 2010.

[Cerf *et al.*, 2013] Cerf, L., Besson, J., Nguyen, K. et Boulicaut, J. (2013). Closed and noise-tolerant patterns in n-ary relations. *Data Min. Knowl. Discov.*, 26(3):574–619.

[Cerf *et al.*, 2009] Cerf, L., Besson, J., Robardet, C. et Boulicaut, J.-F. (2009). Closed patterns meet n-ary relations. *ACM Transactions on Knowledge Discovery from Data*, 3:3 :1–3 :36.

[Chojnacki et Klopotek, 2010] Chojnacki, S. et Klopotek, M. A. (2010). Random graphs for performance evaluation of recommender systems. *CoRR*, abs/1010.5954.

[Couch et Chiarini, 2008] Couch, A. et Chiarini, M. (2008). A theory of closure operators. *In Proc. of the 2nd Conference on Dynamical Systems, Differential Equations and Applications*, pages 162–174, Berlin, Heidelberg. Springer-Verlag.

[Das *et al.*, 2012] Das, M., Thirumuruganathan, S., Amer-Yahia, S., Das, G. et Yu, C. (2012). Who tags what ? an analysis framework. *In Proceedings of Very Large Data Bases*, 5(11):1567–1578.

[De Meo *et al.*, 2010a] De Meo, P., Quattrone, G. et Ursino, D. (2010a). A query expansion and user profile enrichment approach to improve the performance of recommender systems operating on a folksonomy. *User Modeling and User-Adapted Interaction*, 20(1):41–86.

[De Meo *et al.*, 2010b] De Meo, P., Quattrone, G. et Ursino, D. (2010b). A query expansion and user profile enrichment approach to improve the performance of re-

commender systems operating on a folksonomy. *User Modeling and User-Adapted Interaction*, 20(1):41–86.

[Dedzoe, 2011] Dedzoe, W. (2011). *Traitement de Requêtes Top-k dans les Communautés Virtuelles P2P de Partage de Données*. ThÈse, Université de Nantes.

[Deshpande et Karypis, 2004] Deshpande, M. et Karypis, G. (2004). Item-based top-n recommendation algorithms. *ACM Trans. Inf. Syst.*, 22(1):143–177.

[Diederich et Iofciu, 2006a] Diederich, J. et Iofciu, T. (2006a). Finding communities of practice from user profiles based on folksonomies. *In Proceedings of the 1st International Workshop on TEL-CoPs, Crete, Greece*, pages 288–297.

[Diederich et Iofciu, 2006b] Diederich, J. et Iofciu, T. (2006b). Finding communities of practice from user profiles based on folksonomies. *In Proc. of the EC-TEL Workshops, Crete, Greece.*

[Fagnan *et al.*, 2014] Fagnan, J., Rabbany, R., Takaffoli, M., Verbeek, E. et Zaïane, O. R. (2014). Community dynamics : Event and role analysis in social network analysis. *In Advanced Data Mining and Applications - 10th International Conference, ADMA 2014, Guilin, China, December 19-21, 2014. Proceedings*, pages 85–97.

[Felfernig *et al.*, 2007] Felfernig, A., Isak, K., Szabo, K. et Zachar, P. (2007). The VITA financial services sales support environment. *In Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, July 22-26, 2007, Vancouver, British Columbia, Canada*, pages 1692–1699.

[Fleder et Hosanagar, 2007] Fleder, D. M. et Hosanagar, K. (2007). Recommender systems and their impact on sales diversity. *In Proc. of the 8th ACM Conference on Electronic Commerce*, EC '07, pages 192–199, New York, NY, USA. ACM.

[Ganter et Wille, 1999] Ganter, B. et Wille, R. (1999). *Formal Concept Analysis.* Springer, Heidelberg.

[Gnatyshak *et al.*, 2012] Gnatyshak, D., Ignatov, D. I., Semenov, A. et Poelmans, J. (2012). Gaining insight in social networks with biclustering and triclustering. *In* Aseeva, N., Babkin, E. et Kozyrev, O., éditeurs : *BIR*, volume 128 de *Lecture Notes in Business Information Processing*, pages 162–171. Springer.

[Herlocker *et al.*, 2004] Herlocker, J. L., Konstan, J. A., Terveen, L. G. et Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, pages 5–53.

[Hotho *et al.*, 2006] Hotho, A., Jäschke, R., Schmitz, C. et Stumme, G. (2006). Information retrieval in folksonomies : Search and ranking. *In Proc. of ESWC, Budva, Montenegro*, volume 4011 de *LNCS*, pages 411–426. Springer, Heidelberg.

[Hu *et al.*, 2011] Hu, J., Wang, B. et Tao, Z. (2011). Personalized tag recommendation using social contacts. *In Proc. of Workshop SRS'11, in conjunction with CSCW*.

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

[Jäschke *et al.*, 2007] Jäschke, R., Marinho, L., A. Hotho, A., Lars, S.-T. et Stum, G. (2007). Tag recommendations in folksonomies. *In Proc. of the 11th ECML PKDD, Warsaw, Poland*, pages 506–514.

[Jawaheer *et al.*, 2014] Jawaheer, G., Weller, P. et Kostkova, P. (2014). Modeling user preferences in recommender systems : A classification framework for explicit and implicit user feedback. *ACM Trans. Interact. Intell. Syst.*, 4(2):8 :1–8 :26.

[Jelassi, 2014] Jelassi, M. N. (2014). Une nouvelle approche pour l'extraction efficace des quadri-concepts fréquents. *In 12èmes Rencontres de Jeunes Chercheurs en Intelligence Artificielle (RJCIA 2014)*, Rouen, France.

[Jelassi *et al.*, 2013a] Jelassi, M. N., Ben Yahia, S. et Mephu Nguifo, E. (2013a). Nouvelle approche de recommandation personnalisée dans les folksonomies basée sur le profil des utilisateurs. *In Conférence Francophone Ingénierie des Connaissances (IC 2013)*, Lille, France.

[Jelassi *et al.*, 2013b] Jelassi, M. N., Ben Yahia, S. et Mephu Nguifo, E. (2013b). A personalized recommender system based on users' information in folksonomies. *In 5th International Workshop on Web Intelligence & Communities collocated with the WWW 2016 conference*, pages 1215–1224.

[Jelassi *et al.*, 2014] Jelassi, M. N., Ben Yahia, S. et Mephu Nguifo, E. (2014). Vers des recommandations plus personnalisées dans les folksonomies. *In Conférence Francophone Ingénierie des Connaissances (IC 2014)*, Clermont-Ferrand, France.

[Jelassi *et al.*, 2015a] Jelassi, M. N., Ben Yahia, S. et Mephu Nguifo, E. (2015a). Towards more targeted recommendations in folksonomies. *In 5th Workshop on Context-awareness in Retrieval and Recommendation (CaRR 2015) in conjunction with the 37th European Conference on Information Retrieval (ECIR 2015).*

[Jelassi *et al.*, 2015b] Jelassi, M. N., Ben Yahia, S. et Mephu Nguifo, E. (2015b). Towards more targeted recommendations in folksonomies. *Journal of Social Network Analysis and Mining (SNAM)*. Accepted.

[Ji *et al.*, 2006] Ji, L., Tan, K.-L. et Tung, A. K. H. (2006). Mining frequent closed cubes in 3d datasets. *In Proc. of Very Large Data Bases 2006*, pages 811–822, Seoul, Korea.

[Kim *et al.*, 2011] Kim, H. K., Oh, H. Y., Gu, J. C. et Kim, J. K. (2011). Commenders : A recommendation procedure for online book communities. *Electron. Commer. Rec. Appl.*, 10(5):501–509.

[Kim et Chan, 2003] Kim, H. R. et Chan, P. K. (2003). Learning implicit user interest hierarchy for context in personalization. *In Proceedings of the 8th International Conference on Intelligent User Interfaces*, IUI '03, pages 101–108, New York, NY, USA. ACM.

[Landia et Anand, 2009] Landia, N. et Anand, S. (2009). Personalised tag recommendation. *Recommender Systems & the Social Web, New York, NY, USA*, pages 83–86.

[Lehmann et Wille, 1995] Lehmann, F. et Wille, R. (1995). A triadic approach to formal concept analysis. *In Proc. of the 3rd International Conference on Computational Science*, pages 32–43, Santa Cruz, California, USA. Springer-Verlag.

[Liang, 2010] Liang, H. (2010). *User profiling based on folksonomy information in Web 2.0 for personalized recommender systems.* Thèse de doctorat, Queensland University of Technology.

[Liang *et al.*, 2010] Liang, H., Xu, Y., Li, Y. et Nayak, R. (2010). Personalized recommender system based on item taxonomy and folksonomy. *In Proceedings of the 19th ACM International Conference on Information and Knowledge Management '10*, pages 1641–1644, New York, NY, USA. ACM.

[Lipczak, 2008] Lipczak, M. (2008). Tag recommendation for folksonomies oriented towards individual users. *In Proc. of the ECML/PKDD Discovery Challenge, Antwerp, Belgium*, pages 84–95.

[Mahmood et Ricci, 2007] Mahmood, T. et Ricci, F. (2007). Learning and adaptivity in interactive recommender systems. *In Proc. of the 9th Intl. Conf. on Electronic Commerce*, ICEC '07, pages 75–84.

[Melville et Sindhwani, 2010] Melville, P. et Sindhwani, V. (2010). Recommender systems. *In* Sammut, C. et Webb, G., éditeurs : *Encyclopedia of Machine Learning*, pages 829–838. Springer US.

[Merwe *et al.*, 2004] Merwe, D. V. D., Obiedkov, S. et Kourie, D. (2004). Addintent : A new incremental algorithm for constructing concept lattices. *In Lecture Notes in Computer Science Vol 2961*, pages 372–385.

[Michlmayr et Cayzer, 2007] Michlmayr, E. et Cayzer, S. (2007). Learning user profiles from tagging data and leveraging them for personal(ized) information access. *In Proc. of the Workshop on Tagging and Metadata for Social Information Organization in the 16th WWW, Banff, Alberta, Canada.*

[Mooney et Roy, 2000] Mooney, R. J. et Roy, L. (2000). Content-based book recommending using learning for text categorization. *In Proceedings of the Fifth ACM Conference on Digital Libraries*, DL '00, pages 195–204, New York, NY, USA. ACM.

[Obiedkov et Duquenne, 2007] Obiedkov, S. et Duquenne, V. (2007). Attribute-incremental construction of the canonical implication basis. *Annals of Mathematics and Artificial Intelligence*, 49(1-4):77–99.

[Penet *et al.*, 2011] Penet, C., Demarty, C.-H., Gravier, G. et Gros, P. (2011). De la détection d'évènements sonores violents par SVM dans les films. *In Proc. of the 13th ORASIS - Congrès des jeunes chercheurs en vision par ordinateur*, Praz-sur-Arly, France.

[Qumsiyeh et Ng, 2012] Qumsiyeh, R. et Ng, Y.-K. (2012). Predicting the ratings of multimedia items for making personalized recommendations. *In SIGIR'12*, pages 475–484, New York, NY, USA. ACM.

[Ren *et al.*, 2013] Ren, Y., Zhu, T., Li, G. et Zhou, W. (2013). Top-n recommendations by learning user preference dynamics. *In Pacific Asia Knowledge Discovery and Data Mining*, pages 390–401.

[Ricci *et al.*, 2011] Ricci, F., Rokach, L., Shapira, B. et Kantor, P. B., éditeurs (2011). *Recommender Systems Handbook*. Springer.

[Roussey *et al.*, 2006] Roussey, C., Calabretto, S., Harrathi, F. et Gammoudi, M. M. (2006). Multilingual indexing based on ontologies. *In Leading the Web in Concurrent Engineering. Next Generation Concurrent Engineering, Proceedings of the 13th ISPE International Conference on Concurrent Engineering (ISPE CE 2006), September 18-22, 2006, Antibes, France.*, pages 418–425.

[Schein *et al.*, 2002a] Schein, A. I., Popescul, A., Ungar, L. H. et Pennock, D. M. (2002a). Methods and metrics for cold-start recommendations. *In Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '02, pages 253–260, New York, NY, USA. ACM.

[Schein *et al.*, 2002b] Schein, A. I., Popescul, A., Ungar, L. H. et Pennock, D. M. (2002b). Methods and metrics for cold-start recommendations. *In Proc. of the 25th Annual Intl. ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '02, pages 253–260, New York, NY, USA. ACM.

[Sellami *et al.*, 2014] Sellami, M., Gammoudi, M. M. et Hacid, M. (2014). Secure data integration : A formal concept analysis based approach. *In Database and Expert Systems Applications - 25th International Conference, DEXA 2014, Munich, Germany, September 1-4, 2014. Proceedings, Part II*, pages 326–333.

[Trabelsi *et al.*, 2012] Trabelsi, C., Jelassi, N. et Ben Yahia, S. (2012). Scalable mining of frequent tri-concepts. *In Proc. of the 15th Pacific Asia Knowledge Discovery and Data Mining, Kuala Lampur, Malaysia*, pages 231–242.

[Valtchev *et al.*, 2002] Valtchev, P., Missaoui, R. et Godin, R. (2002). A framework for incremental generation of frequent closed itemsets. *In Workshop on Discrete Mathematics & Data Mining, 2nd SIAM Conf. on Data Mining*, pages 75–86.

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

[Voutsadakis, 2006] Voutsadakis, G. (2006). n-closure systems and n-closure operators. *Algebra Universalis*, 55(2):369–386.

[Weiss et Kulikowski, 1991] Weiss, S. M. et Kulikowski, C. A. (1991). *Computer Systems That Learn : Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning, and Expert Systems.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[Weng *et al.*, 2008] Weng, L.-T., Xu, Y., Li, Y. et Nayak, R. (2008). Exploiting item taxonomy for solving cold-start problem in recommendation making. *In ICTAI (2)*, pages 113–120. IEEE Computer Society. 978-0-7695-3440-4.

[Wille, 2009] Wille, R. (2009). Restructuring lattice theory : an approach based on hierarchies of concepts. *In Proceedings of the 7th International Conference on Formal Concept Analysis*, ICFCA '09, pages 314–339, Berlin, Heidelberg. Springer-Verlag.

[Yu *et al.*, 2009] Yu, C., Lakshmanan, L. et Amer-Yahia, S. (2009). It takes variety to make a world : Diversification in recommender systems. *In Proceedings of the 12th International Conference on Extending Database Technology : Advances in Database Technology*, EDBT '09, pages 368–378, New York, NY, USA. ACM.

[Zhao et Zaki, 2005] Zhao, L. et Zaki, M. J. (2005). Tricluster : An effective algorithm for mining coherent clusters in 3d microarray data. *In Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*, SIGMOD '05, pages 694–705, New York, NY, USA.