COCoFil2 : Un nouveau système de filtrage collaboratif basé sur le modèle des espaces de communautés

An-Te Nguyen

UNIVERSITÉ JOSEPH FOURIER – GRENOBLE I

N^0 attribué par la bibliothèque
/__/__/__/__/__/__/__/__/__/__/

THÈSE

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ JOSEPH FOURIER – GRENOBLE I

Discipline : Informatique

présentée et soutenue publiquement

par

An Te NGUYEN

le 23/11/2006

Titre :

COCoFil2 : Un nouveau système de filtrage collaboratif basé sur le modèle des espaces de communautés

Composition du jury

Professeur	Jean-Pierre	GIRAUDIN	*Président*
Docteur	Daniel	ROCACHER	*Rapporteur*
Professeur	Chantal	SOULE-DUPUY	*Rapporteur*
Professeur	Catherine	BERRUT	*Codirectrice de thèse*
Docteur	Nathalie	DENOS	*Codirectrice de thèse*
Professeur	Bich-Thuy	DONG-THI	*Codirectrice de thèse*

Thèse préparée au sein du laboratoire de Communication Langagière et Interaction Personne-Système,
Fédération IMAG – Université Joseph Fourier – Grenoble I

Remerciements

Je tiens tout d'abord à adresser mes remerciements les plus chaleureux aux membres du jury qui ont bien voulu s'intéresser à ma thèse.

A **Jean-Pierre GIRAUDIN** qui m'a fait l'honneur de présider le jury.

Aux rapporteurs **Daniel ROCACHER** et **Chantal SOULE-DUPUY** pour leurs commentaires pertinents qui ont permis d'améliorer ce manuscrit.

A **Catherine BERRUT** pour avoir encadré mes travaux tout au long de ces années et pour avoir été présente à chaque fois que le besoin s'en faisait sentir.

A **Nathalie DENOS** pour avoir co-encadré cette thèse et pour sa présence constante qui a permis de mener à bien cette étude.

A **Bich-Thuy DONG-THI** non seulement pour avoir co-encadré cette thèse mais aussi pour tout ce qu'elle m'a apporté depuis toujours.

Je tiens à remercier les participants du projet APMD pour le temps passé à travailler ensemble et dont j'ai profité pour cette thèse. Je regrette de ne pouvoir continuer à collaborer avec eux en raison de mon retour définitif au Vietnam, et je leur souhaite une bonne continuation avec beaucoup de succès.

Je souhaite également remercier les membres du laboratoire CLIPS, en particulier l'équipe MRIM, pour m'avoir accueilli avec sympathie. Je citerai sans un ordre particulier : Annie, Caroline, Georges, Helga, Jean, Leila, Lizbeth, Loïc, Marie-Christine, Marie-France, Mbarek, Mohamed, Philippe et les deux Stéphane. Je tiens à remercier en particulier Delphine qui m'a aidé avec gentillesse dans la correction de ce manuscrit.

Je tiens à remercier Bao-Quoc, Dan-Thu, Ngoc-Hoa, Nhien-An, Tien-Huy, Trung-Hung, Viet-Bac, Xuan-Hung et plusieurs autres pour leur amitié et leur soutien sans faille.

Je ne terminerai pas sans témoigner ma reconnaissance à ma famille : à mes parents, frères et sœurs, pour tout ce qu'ils m'ont offert pour que je puisse reprendre mes études ; à mes beaux-parents, beaux-frères et belles-sœurs pour leur dévouement ; et finalement, à ma femme Minh-Hong et mes enfants Vinh-Phuc et Thuc-Doan, pour leur appui indéfectible, leur patience et pour leur courage. Je les en remercie du fond du cœur.

Résumé

Face au problème de la surcharge d'information, le *filtrage collaboratif* a pour principe d'exploiter les évaluations que des utilisateurs ont faites de certains documents, afin de recommander ces mêmes documents à d'autres utilisateurs proches de lui, et sans qu'il soit nécessaire d'analyser le contenu des documents. C'est ainsi qu'émerge la notion de *communauté*, définie comme un groupe de personnes qui partagent en général les mêmes centres d'intérêt. De notre point de vue, la problématique du filtrage collaboratif consiste à gérer de façon intelligente des communautés puisque, selon son principe de base, la qualité des recommandations dépend fondamentalement de la qualité des communautés formées par le système.

Le premier aspect de la gestion des communautés à étudier est la capacité des utilisateurs à percevoir des communautés. D'une part, la perception des communautés permet d'améliorer la confiance des utilisateurs dans les recommandations générées à partir de ces communautés, et par conséquent de les motiver à fournir des évaluations sur lesquelles appuyer la formation des communautés pour le filtrage collaboratif. D'autre part, cette capacité autorise les utilisateurs à explorer d'autres communautés potentiellement intéressantes.

Le second aspect à prendre en compte est les informations sur lesquelles appuyer la formation des communautés. On voit dans la réalité qu'une personne reçoit souvent toutes sortes de recommandations intéressantes de ses proches, de ses collègues de travail, etc. Nous émettons donc l'hypothèse que la multiplicité des critères pour former des communautés, incluant profession, centres d'intérêt, historique des évaluations, etc., peut être exploitée pour enrichir les recommandations générées pour un utilisateur.

Enfin, les communautés d'un utilisateur évoluent au cours du temps. En raison de la multiplicité des critères, la qualité du positionnement des utilisateurs au sein des communautés est conditionnée par la qualité des valeurs données pour chaque utilisateur à chaque critère. Certains critères demandent beaucoup d'efforts de la part des utilisateurs, et peuvent être coûteux également pour le système, d'où des difficultés à positionner les utilisateurs dans des communautés.

Ainsi, pour la gestion des communautés dans un système de filtrage collaboratif, nous proposons le *modèle des espaces de communautés* qui présente les caractéristiques suivantes : gestion des communautés explicites, formation multiple des communautés selon divers critères et stratégie de positionnement des utilisateurs au sein des communautés.

L'intégration de notre modèle des espaces de communautés dans un système de filtrage collaboratif permet donc d'améliorer l'exploitation des communautés formées à partir des critères disponibles dans les profils des utilisateurs. Nous présentons la plateforme du filtrage collaboratif COCoFil2 comme la mise en œuvre du modèle proposé ainsi que nos travaux de validation sur un jeu de données réelles.

Mots-clés : *filtrage collaboratif, espaces de communautés, polymorphisme, induction de communautés*.

Table des matières

Liste des figures

Liste des tableaux

Liste des notations

A	ensemble de critères (attributs)
C	ensemble d'attributs de condition
$\gamma(P, D)$	dépendance d'attributs
D	ensemble d'attributs de décision
G_a	communauté du critère a
$\mu(D)$	consistance approximative
Ω_a	espace de communautés du critère a
P	sous-ensemble d'attributs
$\mathcal{P}$	vecteur de positionnement
$POS_P(D)$	région positive
ζ	classificateur
$R_D^{(\theta)}$	réduction approximative avec le seuil θ
$\mathcal{R}$	relation d'équivalence
$\mathcal{R}_P$	relation d'indiscernabilité
$\sigma(P, D)$	signification d'attributs
T_{mxn}	table (matrice) de communautés
$[u]_P$	classe d'équivalence par $\mathcal{R}_P$ de l'élément u
U	ensemble des utilisateurs
$U/\mathcal{R}$	ensemble quotient
V_a	ensemble des valeurs possibles de l'attribut a (domaine)
X	concept

Partie 0.

Introduction

Aujourd'hui, chacun est confronté au problème de la surcharge d'information. Une approche pour pallier ce problème consiste à personnaliser l'accès aux informations, en utilisant des *profils* représentant des intérêts relativement stables des utilisateurs. Le « filtrage d'information » (*Information Filtering*) s'appuie sur de tels profils pour filtrer un flux de documents en vue de ne conserver que les documents pertinents pour l'utilisateur. Le « filtrage collaboratif » (*Collaborative Filtering*) est un cas particulier de filtrage, qui a pour principe d'exploiter les évaluations que des utilisateurs ont faites de certains documents, afin de recommander ces mêmes documents à d'autres utilisateurs proches de lui, et sans qu'il soit nécessaire d'analyser le contenu des documents. C'est ainsi qu'émerge la notion de *communauté*, définie comme un groupe de personnes qui partagent en général les mêmes centres d'intérêt.

Exploiter intelligemment les communautés pour produire de meilleures recommandations est donc un des objectifs des systèmes de filtrage collaboratif. Pour cela, la gestion des communautés joue un rôle très important puisque, selon le principe de base du filtrage collaboratif, la qualité des recommandations envoyées aux utilisateurs dépend fondamentalement de la qualité des communautés formées par le système. Cette thèse vise à répondre à la question : « *Comment peut-on gérer au mieux les communautés dans un système de filtrage collaboratif, afin de fournir de meilleures recommandations aux utilisateurs* ? ». Cette problématique comporte trois aspects essentiels : la perception des communautés, leur formation et le positionnement des utilisateurs au sein des communautés.

Perception des communautés. La perception des communautés est la capacité des utilisateurs à obtenir une vue globale sur les autres participants pour comprendre ce qui se cache derrière les recommandations envoyées au cours du temps par le système. Cette possibilité de perception est très utile dans les activités du système de filtrage collaboratif. Par exemple, elle permet d'améliorer la confiance des utilisateurs dans les recommandations, et par conséquent de les motiver à fournir des évaluations sur lesquelles fonder la formation des communautés. Par ailleurs, la perception des communautés *explicites* autorise les utilisateurs à explorer d'autres communautés potentiellement intéressantes.

Formation des communautés. En général, le processus de formation des communautés dans un système de filtrage collaboratif traditionnel se réalise par la comparaison de l'historique des évaluations des utilisateurs. Pourtant, on voit dans la réalité qu'une personne reçoit souvent toutes sortes de recommandations intéressantes de ses proches, de ses collègues de travail, des membres de son club de loisirs, etc. Nous émettons ici l'hypothèse que la *multiplicité des critères* permettant de former des communautés, incluant âge, profession, ville de résidence, centres d'intérêt, historique des évaluations, etc., peut être exploitée pour enrichir et diversifier les recommandations générées pour un utilisateur.

Positionnement des utilisateurs au sein des communautés. Les communautés d'un utilisateur évoluent au cours du temps grâce aux interactions entre l'utilisateur et le système, notamment grâce aux évaluations produites par cet utilisateur. Lorsqu'il s'inscrit et commence à utiliser le système, le problème du *démarrage à froid* se pose puisque ses communautés sont encore inconnues. Par conséquent, le système ne peut pas lui fournir de recommandations pertinentes. Plus généralement, dans le contexte d'une multiplicité de critères, la qualité du positionnement des utilisateurs au sein des communautés dépend fondamentalement de la qualité des valeurs associées à chaque utilisateur pour chaque critère. Certaines de ces valeurs requièrent *beaucoup d'efforts* de la part des utilisateurs, et cela peut être coûteux également pour le système, d'où des difficultés à positionner les utilisateurs dans les espaces de communautés. Par exemple, un nouvel utilisateur peine à définir son genre de film préféré, qui est par ailleurs susceptible d'évoluer. De même, évaluer un grand nombre de films est une tâche lourde pour l'utilisateur, et c'est pourtant ainsi que les communautés sont formées selon le critère de l'historique des évaluations.

Donc, notre objectif est de concevoir un modèle efficace pour la gestion des communautés dans un système de filtrage collaboratif. En nous basant sur la théorie des ensembles d'approximation, nous proposons dans cette thèse le *modèle des espaces de communautés* qui présente les caractéristiques suivantes : gestion des communautés explicites, formation multiple des communautés selon divers critères et stratégie de positionnement des utilisateurs au sein des communautés.

Un espace de communautés dans notre modèle est un ensemble de communautés formées selon un critère particulier comme par exemple l'âge, la profession, la ville de résidence, les centres d'intérêt, ou l'historique des évaluations.

a) Communautés explicites. Nous proposons une méthode de formation et de visualisation des espaces de communautés afin de les rendre complètement explicites. Cette méthode combine l'algorithme des fourmis artificielles et l'algorithme des K-moyennes afin de créer des « cartes de communautés » en 2D. Grâce à ces cartes, l'utilisateur obtient la possibilité de percevoir toutes les communautés d'un espace relatif à un critère donné, sous réserve de sa capacité cognitive à les appréhender.

b) Formation multiple des communautés. Nous proposons la formalisation des espaces de communautés selon des critères variés : les communautés ne sont plus seulement formées sur la base de l'historique des évaluations des utilisateurs mais aussi

sur tous les autres critères de rapprochement entre utilisateurs, qui sont disponibles dans le système. Par conséquent, un utilisateur appartient à une communauté dans chaque espace. Au niveau formel, nous utilisons une relation d'équivalence pour la représentation de chaque espace de communautés relatif à un critère donné. Ainsi, les communautés sont des classes d'équivalence dans l'ensemble quotient de la relation concernée.

c) Induction des communautés. Nous proposons de plus la possibilité de rattacher un utilisateur à une communauté dans un espace relatif à un certain critère à partir de ses communautés déjà connues dans d'autres espaces. La réponse positive à cette question permet au modèle de diminuer l'effort des utilisateurs notamment en situation de démarrage à froid, et à terme cela permet aux utilisateurs de découvrir des communautés potentiellement intéressantes qu'ils ne connaissent pas encore. Pour ce faire, nous nous appuyons sur la théorie des ensembles d'approximation en raison de ses moyens efficaces pour analyser la dépendance entre un critère clé a priori fixé et les critères restants. Nous proposons une extension de cette théorie pour définir les mesures destinées à comparer les critères problématiques sans fixer aucun critère clé a priori. Nous définissons alors des moyens pour analyser les relations des critères entre eux. Lorsque plusieurs communautés multicritères d'un utilisateur sont manquantes ou douteuses, ces mesures aident le système à déterminer, parmi les critères problématiques, ceux qui peuvent être traités pour induire les communautés, ainsi qu'à établir un ordre de traitement des critères ainsi déterminés, dans le but d'élaborer des stratégies pour améliorer le positionnement de l'utilisateur dans les divers espaces de communautés.

L'intégration de notre modèle dans un système de filtrage collaboratif permet donc d'améliorer l'exploitation des communautés formées à partir des critères disponibles dans les profils des utilisateurs.

Partie I.

Contexte de recherche et objectifs

Cette première partie a pour but d'introduire le sujet de notre thèse.

Dans le Chapitre 1, nous parlons du contexte de notre étude. Il s'agit des systèmes de filtrage collaboratif, qui envoient automatiquement des recommandations aux utilisateurs en fonction des « communautés » auxquelles ils appartiennent, ces communautés étant des groupes de personnes qui partagent les mêmes centres d'intérêt.

La problématique de la gestion des communautés, qui est au cœur des systèmes de filtrage collaboratif, est présentée dans le Chapitre 2. Elle comporte trois aspects essentiels des communautés dans un système de filtrage collaboratif : la perception, la formation des communautés et le positionnement des utilisateurs au sein des communautés.

Nous présentons dans le Chapitre 3 les objectifs de la thèse ainsi qu'une vue d'ensemble de notre démarche et notre proposition afin d'atteindre ces objectifs. De façon générale, notre objectif est de concevoir un modèle efficace pour la gestion des communautés dans un système de filtrage collaboratif. Nous proposons dans cette thèse le « modèle des espaces de communautés » qui présente les caractéristiques suivantes : gestion des communautés explicites, formation multiple des communautés selon divers critères et efficacité du positionnement des utilisateurs au sein des communautés.

Chapitre 1

Système de filtrage collaboratif

La recherche d'information (*Information Retrieval*) [vRi79] dans le contexte du développement des ressources sur Internet demeure un défi à relever. Les utilisateurs doivent souvent faire face à un très grand nombre de choix, et ils rencontrent beaucoup de difficultés pour prendre une décision appropriée. Par exemple, dans les moteurs de recherche comme Google, AltaVista, Yahoo, etc., un utilisateur formule son besoin par une requête, en utilisant des mots-clés qui seront comparés avec des documents indexés dans les bases de données. Le résultat retourné à l'utilisateur contient souvent un grand nombre de documents non pertinents. Il doit sélectionner manuellement les documents pertinents. Il s'agit d'une tâche pénible et ennuyeuse pour l'utilisateur.

Le problème de la surcharge d'information peut être pallié par la personnalisation de l'accès aux informations, en utilisant des *profils* représentant des intérêts relativement stables des utilisateurs. En d'autres termes, les profils des utilisateurs sont utilisés comme des critères persistants dans la recherche d'information. Le filtrage d'information (*Information Filtering*) [BC92] s'appuie également sur de tels profils pour filtrer un flux de documents en vue de ne conserver que les documents pertinents pour l'utilisateur.

Dans ce chapitre, dédié au contexte de notre étude, nous présentons d'abord le filtrage d'information de façon générale. Ensuite, nous décrivons les techniques existantes les plus répandues : le *filtrage basé sur le contenu* et le *filtrage collaboratif*. Enfin, nous focalisons sur les systèmes de filtrage collaboratif en discutant les processus et les facteurs principaux.

1.1 Filtrage d'information

L'objectif principal d'un système de filtrage d'information, ou système de recommandation (*Recommender System*), est de filtrer un flux entrant d'informations de façon personnalisée pour chaque utilisateur, tout en s'adaptant en permanence à son besoin d'information. Autrement dit, dans le but de personnaliser la recherche d'information dans un domaine d'application particulier, un système de filtrage collecte, sélectionne, classifie et suggère à l'utilisateur les informations qui répondent vraisemblablement à ses intérêts à long terme. Il existe à présent de nombreux systèmes de recommandation utilisés dans divers domaines comme la recherche documentaire, le commerce électronique, les loisirs, etc. On peut citer à titre d'exemple quelques sites Web populaires comme CiteSeer[1], Amazon[2], MovieLens[3], etc.

Pour réaliser le filtrage, le moteur du système de recommandation gère les profils des utilisateurs, et les exploite pour sélectionner les documents à transmettre à chacun. Le moteur adapte ces profils au cours du temps en exploitant au mieux le retour de pertinence que les utilisateurs fournissent sur les informations (documents) reçues. Par exemple, dans la Figure 1.1, la fonction de décision du système traite le flux entrant de documents pour suggérer à l'utilisateur, en consultant son profil, les documents que vraisemblablement il préfère. A son tour, l'utilisateur doit fournir ses évaluations, c'est-à-dire évaluer fréquemment les recommandations, pour que le système comprenne mieux ses besoins en information, et lui fournisse par conséquent de meilleures nouvelles recommandations.

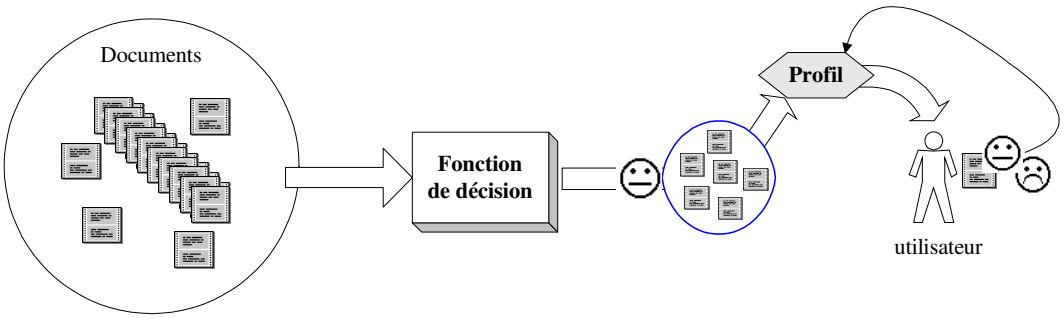


Figure 1.1 – *Schéma général du filtrage d'information.*

1.2 Techniques de filtrage

Actuellement, il existe trois grandes approches de filtrage : basé sur le contenu, collaboratif et hybride. Le filtrage basé sur le contenu compare les nouveaux documents au profil de chaque utilisateur, et recommande ceux qui sont le plus proche. Le filtrage collaboratif compare les

[1] http://citeseer.ist.psu.edu
[2] http://www.amazon.com
[3] http://movielens.umn.edu

utilisateurs entre eux sur la base de leurs jugements passés pour créer des communautés, et chaque utilisateur reçoit les documents jugés pertinents par sa communauté. Le filtrage hybride combine le filtrage basé sur le contenu et le filtrage collaboratif pour exploiter au mieux les avantages de chacun.

Dans la suite, nous présentons plus en détails ces approches de filtrage, en particulier le filtrage collaboratif qui est au centre de la problématique de cette thèse.

1.2.1 Filtrage basé sur le contenu

Le filtrage basé sur le contenu (*Content-based Filtering*) [Lan95, Lie95, PB97], qui est une évolution générale des études sur le filtrage d'information, s'appuie sur le contenu des documents (thèmes abordés) pour les comparer à un profil lui-même constitué de thèmes. Chaque utilisateur du système possède alors un profil qui décrit ses propres centres d'intérêt. Par exemple, le profil peut contenir une liste des thèmes que l'utilisateur aime bien ou qu'il n'aime pas [SB88]. Lors de l'arrivée d'un nouveau document, le système compare la représentation du document avec le profil pour prédire la satisfaction de l'utilisateur sur ce document.

Le premier avantage du filtrage basé sur le contenu est qu'il peut répondre aux intérêts à long terme des utilisateurs, en employant des techniques efficaces dans le domaine de l'intelligence artificielle pour la mise à jour des profils et le recoupement entre profils et documents [MLD03]. En outre, l'utilisateur dans un tel système ne dépend absolument pas des autres. Ainsi, il recevra des recommandations du système même s'il est le seul inscrit, pour peu qu'il ait décrit son profil en donnant un ensemble de thèmes qui l'intéressent.

En revanche, cette technique de filtrage est soumise à l'effet « entonnoir », car le profil évolue naturellement par restriction progressive sur les thèmes recherchés. Ainsi, l'utilisateur ne reçoit que les recommandations relatives aux thèmes présentés dans son profil, une fois devenu stable. Par conséquent, il ne peut pas découvrir de nouveaux domaines potentiellement intéressants pour lui.

Le filtrage basé sur le contenu est également victime d'un effet de masse, car ne bénéficiant pas des jugements de qualité que d'autres utilisateurs ont pu faire sur les documents qu'il reçoit, c'est l'utilisateur lui-même qui devra procéder à l'écrémage des documents reçus, écrémage qui fait intervenir d'autres critères que celui de la thématique.

Enfin, le filtrage basé sur le contenu doit également faire face au problème du démarrage à froid. Par exemple, un nouvel utilisateur rencontre plus ou moins de difficultés à définir les thèmes qu'il préfère, afin que le système instancie son profil, malgré certaines techniques d'apprentissage permettant d'en inférer à partir des textes « exemples » fournis par l'utilisateur.

1.2.2 Filtrage collaboratif

A l'opposé du filtrage basé sur le contenu, le filtrage collaboratif (*Collaborative Filtering*) a pour principe d'exploiter les évaluations que des utilisateurs ont faites de certains documents, afin de recommander ces mêmes documents à d'autres utilisateurs, et sans qu'il soit nécessaire d'analyser le

contenu des documents [BHK98, GON+92, HKJ+99, RIS+94]. Par exemple, dans la Figure 1.2, supposons que l'on a des communautés formées par la proximité des évaluations des utilisateurs. Le document *d* sera recommandé à l'utilisateur *u*, car ce document est apprécié de la communauté G où se trouve l'utilisateur.

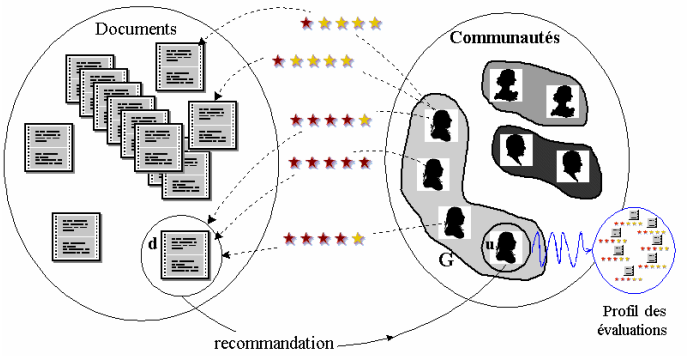


Figure 1.2 – *Principe général du filtrage collaboratif.*

Tous les utilisateurs du système de filtrage collaboratif peuvent tirer profit des évaluations des autres en recevant des recommandations pour lesquelles les utilisateurs les plus proches ont émis un jugement de valeur favorable, et cela sans que le système dispose d'un processus d'extraction du contenu des documents. Grâce à son indépendance vis-à-vis de la représentation des données, cette technique peut s'appliquer dans les contextes où le contenu est soit indisponible, soit difficile à analyser, et en particulier elle peut s'utiliser pour tout type de données : texte, image, audio et vidéo.

De plus, l'utilisateur est capable de découvrir divers domaines intéressants, car le principe du filtrage collaboratif ne se fonde absolument pas sur la dimension thématique des profils, et n'est pas soumis à l'effet « entonnoir ».

Un autre avantage du filtrage collaboratif est que les jugements de valeur des utilisateurs intègrent non seulement la dimension thématique mais aussi d'autres facteurs relatifs à la qualité des documents tels que la diversité, la nouveauté, l'adéquation du public visé, etc.

De nombreux systèmes de recommandation s'appuient partiellement ou totalement sur le filtrage collaboratif [Bur02, MLD03], en raison des avantages importants ci-dessus. On constate néanmoins certains inconvénients de cette technique, incluant le démarrage à froid, la masse critique, le rapport coût-bénéfice et l'expression limitée du besoin, que nous voulons prendre en compte dans cette thèse afin d'améliorer la performance des systèmes de recommandation.

Démarrage à froid. Ce phénomène se produit en début d'utilisation du système, dans des situations critiques où le système manque de données pour procéder à un filtrage personnalisé de bonne qualité. En général, la communauté d'un utilisateur évolue au cours du temps grâce aux évaluations produites par l'utilisateur lui-même. Lorsqu'il s'inscrit pour utiliser le système, sa communauté est encore inconnue, ce qui conduit à l'impossibilité de fournir des recommandations pertinentes.

Masse critique. Afin de former de meilleures communautés, le système exige un nombre suffisant d'évaluations en commun entre les utilisateurs pour les comparer entre eux. Par exemple, on ne peut pas conclure que deux personnes sont dans une même communauté si elles n'ont qu'une seule évaluation en commun. Et pourtant, vu la taille énorme de l'ensemble des documents, achats, etc. dans les systèmes, le nombre des évaluations en commun entre utilisateurs risque d'être faible.

Rapport coût-bénéfice. Le filtrage collaboratif est un processus qui implique fortement les utilisateurs, puisque ses performances dépendent de la bonne utilisation du système : les utilisateurs doivent chacun émettre suffisamment d'évaluations pour dépasser le problème du démarrage à froid ; ils doivent être en suffisamment grand nombre pour atteindre une certaine masse critique au-delà de laquelle les calculs de prédiction prennent toute leur valeur ; les évaluations doivent concerner des ensembles de documents qui se recoupent au maximum, afin de permettre au système de comparer les profils ; etc. Mais ce vœu pieux d'une bonne utilisation du système se heurte au problème du rapport coût-bénéfice dont les utilisateurs tiennent compte de façon consciente ou inconsciente. En effet, l'utilisateur ne perçoit pas toujours favorablement le rapport coût-bénéfice que ce type de système apporte. Lorsqu'il évalue des documents, l'utilisateur se demande si ses efforts seront payés en retour à court ou à moyen terme. Les coûts assumés par l'utilisateur recouvrent l'effort d'évaluation des documents stricto sensu, mais aussi l'effort de compréhension de la tâche à accomplir, l'effort de prise en main de l'outil qui ne s'intègre pas toujours harmonieusement avec les outils auxquels l'utilisateur est habitué.

Expression limitée du besoin. En général, les utilisateurs ne peuvent exprimer l'évolution de leur besoin d'information que sous la forme d'une succession d'évaluations des documents reçus. Ainsi, un changement dans le besoin d'information sera potentiellement mal traduit par un utilisateur qui ne reçoit pas du système les documents lui permettant d'exprimer ce changement.

Pour terminer cette présentation du filtrage collaboratif, nous précisons que l'on trouve dans la littérature une autre technique nommée « filtrage démographique » (*Demographic Filtering*) dont l'idée principale est de catégoriser les utilisateurs selon leurs informations démographiques ; les recommandations sont ensuite générées en fonction des catégories démographiques des utilisateurs [Kru97, Paz99, Ric79]. Nous partageons l'opinion de Burke que cette technique a le même principe de base que le filtrage collaboratif, mais qu'elle s'appuie sur d'autres types de données pour créer des communautés [Bur02].

1.2.3 Filtrage hybride

Constatant les avantages et inconvénients de chacune des deux approches ci-dessus, on comprend que de nombreux systèmes reposent sur leur combinaison, ce qui en fait des systèmes de filtrage dits « hybrides ». En général, l'hybridation s'effectue en deux phases : (i) appliquer séparément le filtrage collaboratif et autres techniques de filtrage pour générer des recommandations candidates, et (ii) combiner ces ensembles de recommandations préliminaires selon certaines méthodes telles que la pondération, la mixtion, la cascade, la commutation, etc. [Bur02], afin de produire les recommandations finales pour les utilisateurs.

Plus généralement, les systèmes hybrides gèrent des profils d'utilisateurs orientés contenu, et la comparaison entre ces profils donne lieu à la formation de communautés d'utilisateurs permettant le filtrage collaboratif.

1.3 Systèmes de filtrage collaboratif

Dans cette thèse, nous nous intéressons aux systèmes de recommandation qui se basent *partiellement* ou *totalement* sur le filtrage collaboratif. Par la suite et pour simplifier, le terme « système de filtrage collaboratif » est utilisé pour désigner, sauf précision autre, un tel système qui génère les recommandations en tenant compte de l'opinion des communautés.

Nous donnons dans cette section une description générale des systèmes de filtrage collaboratif, en décrivant les principaux processus ainsi que les facteurs clés.

1.3.1 Processus du filtrage collaboratif

Comme le montre la Figure 1.3, il y a trois processus principaux dans un système de filtrage collaboratif : évaluation des recommandations, formation des communautés et production des recommandations [Her00, HKJ+99]. Parmi ces trois processus, la formation des communautés, ou plus généralement la *gestion des communautés*, est l'objectif scientifique principal de cette thèse.

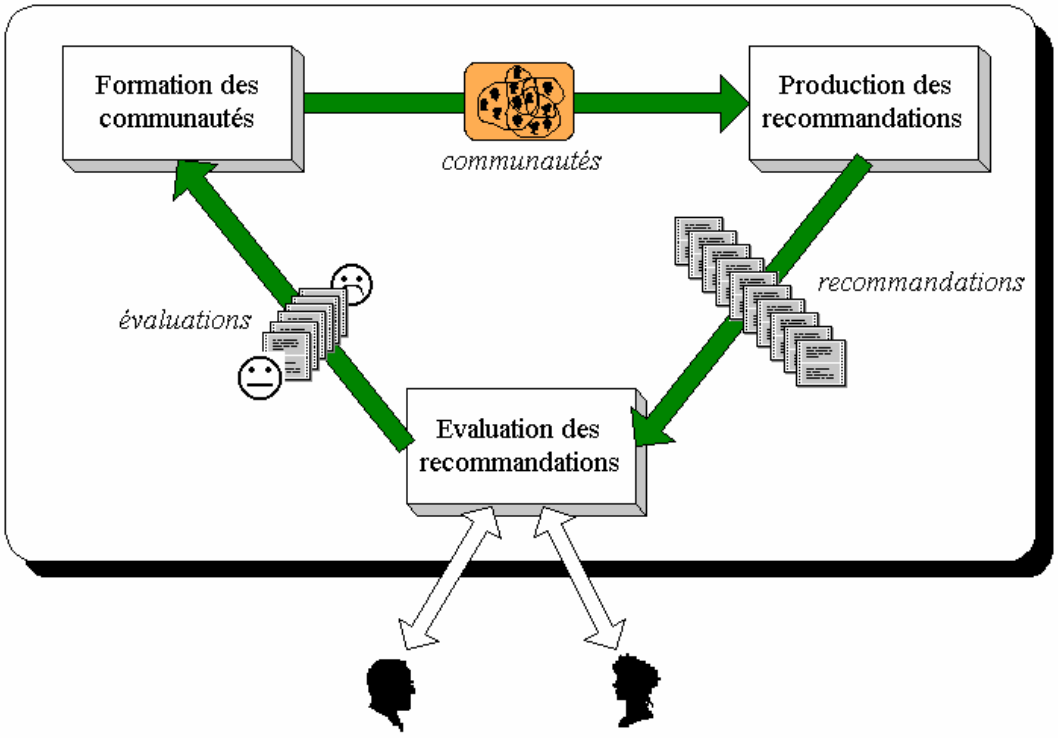


Figure 1.3 – *Trois processus principaux d'un système de filtrage collaboratif.*

1.3.1.1 Evaluation des recommandations

Selon le principe de base du filtrage collaboratif, les utilisateurs doivent fournir leurs évaluations sur des documents afin que le système forme les communautés. Evaluer une recommandation peut se faire de façon explicite ou implicite, comme suit.

– **Explicite** : L'utilisateur donne une valeur numérique sur une échelle donnée ou une valeur qualitative de satisfaction, par exemple, mauvaise, moyenne, bonne et excellente.

– **Implicite** : Le système induit la satisfaction de l'utilisateur à travers ses actions [CLW+01, Nic97]. Par exemple, le système estimera qu'une recommandation supprimée correspond à une évaluation très mauvaise, alors qu'une recommandation imprimée ou sauvegardée peut être interprétée comme une bonne évaluation.

Il faut par ailleurs noter que les recommandations qu'évalue un utilisateur peuvent être générées par le système et/ou choisies par l'utilisateur lui-même [MovieLens].

1.3.1.2 Formation des communautés

Le processus de formation des communautés est le noyau d'un système de filtrage collaboratif. Pour chaque utilisateur, le système doit calculer sa communauté, généralement cela se fait par la proximité des évaluations des utilisateurs. Pour ce faire, on peut calculer dans un premier temps la proximité entre un utilisateur donné et tous les autres. Enfin, afin de créer concrètement la communauté de l'utilisateur, on applique souvent la méthode des voisins les plus proches en utilisant un seuil pour le niveau de proximité ou un seuil pour la taille maximale de la communauté, en raison de sa performance et sa précision [Her00, HKJ+99].

D'autres approches pour la formation des communautés telles que l'approche probabiliste et l'approche des réseaux, sont présentées en détails dans le Chapitre 5.

1.3.1.3 Production des recommandations

Dans ce dernier processus, une fois la communauté de l'utilisateur créée, le système prédit l'intérêt qu'un document particulier peut présenter pour l'utilisateur en s'appuyant sur les évaluations que les membres de la communauté ont faites de ce même document. Lorsque l'intérêt prédit dépasse un certain seuil, le système recommande le document à l'utilisateur [BHK98].

1.3.2 Profils et communautés

Ici, nous discutons les profils basés sur l'historique des évaluations des utilisateurs, ainsi que les communautés, qui sont les deux facteurs clés d'un système de filtrage collaboratif.

1.3.2.1 Profil utilisateur

Le filtrage collaboratif est réalisé à l'aide d'un profil constitué de l'historique des évaluations de chaque utilisateur. Bien que le filtrage collaboratif puisse s'appliquer sur des évaluations explicites

aussi bien que sur des évaluations implicites, dans cette thèse nous prenons en compte essentiellement les évaluations explicites. Ainsi, un profil est constitué de paires (document, évaluation), et il s'enrichit progressivement au fur et à mesure que l'utilisateur évalue des documents reçus [MLD03].

Il est important de noter que dans le principe du filtrage « purement collaboratif », le profil de l'historique des évaluations ne comporte pas d'informations sur le contenu des documents évalués : seul l'identificateur du document est conservé dans le profil.

Par contre, dans un système de recommandation hybride combinant le filtrage collaboratif avec d'autres techniques, chaque utilisateur possède un profil « multidimensionnel » qui comprend, outre l'historique des évaluations, d'autres données telles que les informations personnelles, les centres d'intérêt, etc.

1.3.2.2 Communautés

La notion de communauté dans un système de filtrage collaboratif est définie comme le regroupement des utilisateurs en fonction de l'historique de leurs évaluations, afin que le système calcule des recommandations. Selon cette optique, les profils sont un facteur interactif, alors que les communautés sont considérées comme un facteur interne du système.

Les aspects importants de la notion de communauté sont développés dans les chapitres qui suivent.

Chapitre 2

Problématique

Dans un système de filtrage collaboratif, la *gestion des communautés* joue un rôle très important puisque la qualité des recommandations envoyées aux utilisateurs dépend fondamentalement de la qualité des communautés formées par le système selon le principe de base du filtrage collaboratif. De ce fait, nous présentons dans ce chapitre la problématique de la gestion des communautés dans un système de filtrage collaboratif, qui motive les travaux de cette thèse.

A travers le présent chapitre, nous identifions les problèmes fondamentaux relatifs à la gestion des communautés.

2.1 Trois aspects problématiques de la gestion des communautés

En vue de répondre à la question : « Comment peut-on gérer au mieux les communautés afin de fournir de meilleures recommandations aux utilisateurs ? », et comme le montre la Figure 2.1, trois aspects problématiques sont à aborder : perception des communautés pour les utilisateurs, formation des communautés par le système et positionnement des utilisateurs dans les communautés.

Dans ce chapitre, nous abordons d'abord la *perception des communautés* pour les utilisateurs (section 2.2). Nous détaillons d'une part l'impact de cet aspect sur la motivation des utilisateurs à fournir leur évaluation sur des recommandations (flèche (1) dans la Figure 2.1), qui alimente la formation des communautés, et d'autre part la capacité d'exploration des communautés potentiellement intéressantes (flèche (2) dans la même figure).

Ensuite, du côté du système, nous discutons, dans la section 2.3, la *formation des communautés* pour générer des recommandations collaboratives (flèches (3) et (4)), notamment sur quels critères appuyer ce processus. Enfin, les difficultés émergeantes dans le *positionnement des utilisateurs* au sein des communautés (flèche (5)) sont présentées dans la section 2.4. Nous visons en particulier l'intégration de nouveaux utilisateurs pour lesquels on dispose de peu d'information.

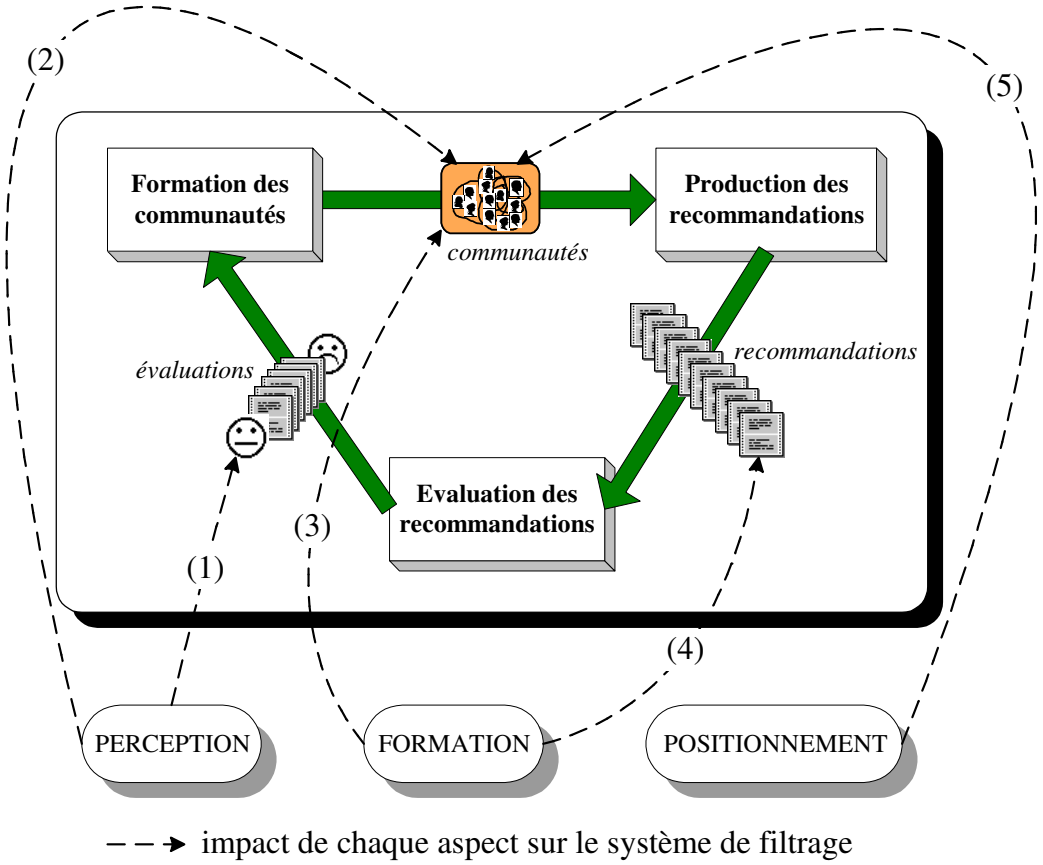


Figure 2.1 – *Trois aspects problématiques de la gestion des communautés.*

2.2 Perception des communautés

La perception des communautés est la capacité des utilisateurs à obtenir une vision globale sur les autres participants pour comprendre ce qui se cache derrière les recommandations envoyées au cours du temps par le système. Cette possibilité de perception est très utile dans les activités du système de filtrage collaboratif. Elle permet par exemple au système de renforcer la performance de la formation des communautés.

Il existe dans la littérature des études qui mettent en évidence l'impact de la perception des communautés sur l'acceptation des recommandations par les utilisateurs, et par conséquent, sur la

motivation des utilisateurs à fournir des évaluations afin que le système crée des communautés [Her00, HKR00]. En réalité, il semble naturel que les utilisateurs aient beaucoup plus confiance en ce qu'ils peuvent percevoir.

La perception des communautés autorise par ailleurs les utilisateurs à explorer d'autres communautés potentiellement intéressantes. On peut envisager que l'utilisateur souhaite connaître les recommandations récentes que le système procure à une communauté particulière ou les évaluations typiques de cette communauté sous réserve du respect de la confidentialité, en vue de changer de communauté. Il est toutefois à remarquer que la réalisation de cette possibilité nécessite une étude complémentaire sur la capacité cognitive des utilisateurs à interagir avec ces communautés.

Cependant, l'utilisateur ne perçoit que les évaluations des recommandations qui lui sont faites, et les communautés formées dans la majorité des systèmes courants lui sont souvent implicites et invisibles. En général, les communautés sont considérées comme des résultats intermédiaires dans la production des recommandations. Cette invisibilité des communautés limite véritablement la capacité d'explication du système quant aux communautés qui sont à l'origine des recommandations envoyées aux utilisateurs, et en particulier empêche les utilisateurs d'explorer les communautés existantes dans le système.

2.3 Formation des communautés

La formation des communautés est une base importante pour la production de recommandations dans le filtrage collaboratif. Selon le principe classique de ce filtrage, les communautés sont en général formées par la proximité des évaluations passées des utilisateurs. Pour construire la communauté d'un utilisateur donné, la méthode des voisins les plus proches est l'approche la plus populaire [BHK98, HKJ+99]. Ainsi, l'utilisateur reçoit des recommandations à partir de sa communauté par l'historique des évaluations.

Par ailleurs, on constate qu'une personne peut recevoir des informations intéressantes par l'intermédiaire de ses proches, de ses collègues de travail, des membres de son club de loisirs, etc. Il peut alors sembler dommage qu'une personne utilisatrice d'un système reçoive uniquement des recommandations calculées à partir de ses évaluations. De fait, on peut se poser la question de proposer à cette personne de former autour d'elle autant de communautés qu'elle le souhaite : la communauté de ses proches, celle de ses collègues de travail ou plus généralement toute communauté de personnes avec lesquelles elle partage un centre d'intérêt. Ainsi, contrairement à l'approche classique des systèmes de filtrage collaboratif, qui associent une et une seule communauté à chaque utilisateur, tout utilisateur peut appartenir à des communautés très diverses, mais complémentaires pour lui.

Prenons l'exemple d'un système de recommandation de films dans lequel on connaît pour chaque utilisateur, à travers son profil, sa profession, la ville où il habite, son genre de film préféré et ses évaluations sur certains films. On peut envisager un regroupement des utilisateurs proches les uns des autres relativement à chacune de ces informations. La Figure 2.2 nous montre un exemple de positionnement multiple (ou polymorphisme), où les utilisateurs sont associés très différemment les

uns aux autres selon le critère choisi. L'utilisateur 3 ayant pour profil « Chercheur » pour le critère Profession, « Paris » pour le critère Ville et « Documentaire » pour le critère Genre préféré et une liste de ses évaluations dans son profil, appartient à quatre communautés différentes. Ainsi, il se peut que cet utilisateur puisse recevoir les films envoyés via les communautés Chercheur, Paris et Documentaire ainsi que ceux de la communauté Groupe 2 basée sur le critère Evaluation. Il est naturel que ces recommandations diversifiées puissent être exploitées de façon sélective dans une stratégie adaptée à la situation rencontrée.

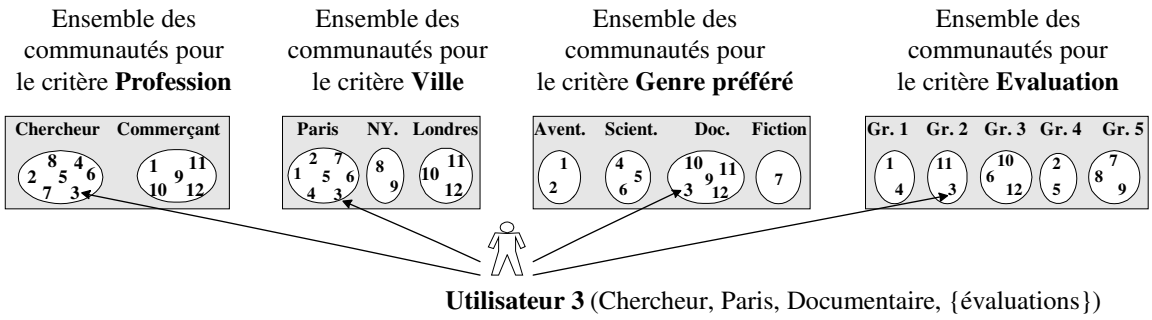


Figure 2.2 – *Polymorphisme de positionnement d'utilisateurs*.

En résumé, la formation monocritère des communautés par l'historique des évaluations dans les systèmes de filtrage collaboratif classiques limite inutilement l'enrichissement et la diversification des recommandations pour les utilisateurs. Il est souhaitable que la formation multicritère des communautés et le polymorphisme du positionnement des utilisateurs soient réalisés afin de rendre plus performants qualitativement les systèmes de filtrage collaboratif.

2.4 Positionnement des utilisateurs au sein des communautés

Dans un système de filtrage collaboratif, les utilisateurs reçoivent des documents que leur recommande le système sur la base de leurs communautés. Les communautés d'un utilisateur évoluent au cours du temps grâce aux interactions entre l'utilisateur et le système, notamment grâce aux évaluations produites par cet utilisateur. Lorsqu'il s'inscrit et commence à utiliser le système, le problème du « démarrage à froid » se pose puisque ses communautés sont encore inconnues. Par conséquent, le système ne peut pas lui fournir de recommandations pertinentes [ME95].

Plus généralement, la qualité du positionnement des utilisateurs dans les communautés dépend fondamentalement de la qualité des valeurs données pour chaque utilisateur à chaque critère. Certains critères demandent beaucoup d'efforts de la part des utilisateurs, et peuvent être coûteux également pour le système [MLD03]. Par exemple, un nouvel utilisateur peine à définir son genre de film préféré, qui est par ailleurs susceptible d'évoluer. De même, évaluer un grand nombre de films est une tâche lourde pour l'utilisateur [RAC+02], et c'est pourtant ainsi que les communautés sont formées pour le critère Evaluation.

Cette difficulté conduit à l'absence de valeur pour un ou plusieurs critères, par exemple on ne connaît pas le genre préféré, et à l'existence de valeurs douteuses ou périmées, par exemple les évaluations sont en très petit nombre, d'où des difficultés à positionner les utilisateurs dans les communautés.

En constatant les problèmes qui émergent dans ce contexte de gestion des communautés, les objectifs scientifiques ainsi que la proposition de la thèse sont présentés dans le Chapitre 3 qui suit.

Chapitre 3

Objectifs et proposition

3.1 Objectifs

La gestion des communautés conditionne la qualité des recommandations du filtrage collaboratif. De ce fait, l'objectif général de cette thèse est de concevoir un modèle qui permet de gérer efficacement les communautés en leur conférant les trois caractéristiques suivantes : gestion des communautés explicites, formation multiple des communautés selon divers critères et efficacité du positionnement des utilisateurs au sein des communautés.

3.1.1 Gestion des communautés explicites

Le modèle doit être capable de gérer explicitement les communautés pour qu'elles ne soient plus considérées comme des résultats intermédiaires dans la production de recommandations. De ce fait, les communautés du système peuvent participer à tous les processus importants du filtrage collaboratif. En outre, nous croyons que l'existence permanente des communautés renforce la confiance des utilisateurs dans le système. Et, ceci est une garantie pour un rapport coût-bénéfice favorable dans l'utilisation d'un système de filtrage collaboratif.

3.1.2 Formation multiple de communautés

Dans l'objectif de s'adapter au contexte de multiplicité des critères issus des profils d'utilisateurs, nous souhaitons un modèle autorisant la formation multiple des communautés selon tous les critères disponibles dans les profils d'utilisateurs. Nous appelons *espace de communautés* l'ensemble de communautés formées à partir d'un critère donné.

3.1.3 Efficacité du positionnement des utilisateurs dans les communautés

Il existe à la fois des critères simples et complexes pour former les communautés. Par exemple, pour les critères démographiques, les utilisateurs fournissent sans aucun effort les données telles que leur âge, profession, etc., et les communautés formées sur ces critères sont ensuite créées par comparaison directe des valeurs. Quant au système, une simple explication sur cette comparaison et/ou sur le regroupement plus significatif de valeurs rassure suffisamment les utilisateurs sur la qualité de leurs communautés actuelles.

Cependant, les critères plus complexes comme l'historique des évaluations ou les centres d'intérêts exigent plus d'effort de la part des utilisateurs (fournir les données afin de construire de meilleures communautés) et pour le système (expliquer à un individu pourquoi il est rattaché à une communauté particulière).

Nous préférons donc un modèle qui offre des moyens efficaces d'une part pour soulager les utilisateurs en diminuant leur effort à fournir des données nécessaires à la création des espaces de communautés, et d'autre part pour faciliter la tâche d'explication du système afin de renforcer leur confiance dans les communautés actuelles.

Pour conclure, nous voulons souligner ici que la validation du modèle proposé pour la gestion des communautés sur un jeu de données réelles est indispensable.

3.2 Proposition

Nous constatons en passant en revue les travaux existants sur les communautés dans un système de recommandation qu'il n'existe aucun modèle conceptuel applicable pour les communautés. Nous proposons de concevoir un nouveau modèle pour la gestion des communautés.

Afin de prendre en compte les besoins en terme de perception des communautés par les utilisateurs, de formation multiple des communautés, et de positionnement des utilisateurs au sein des communautés, nous proposons, comme la Figure 3.1 les montre, d'utiliser la multiplicité des informations issues des profils d'utilisateurs afin de :

- former des espaces de communautés (A), c'est-à-dire de former autant de communautés que nécessaire à l'expression plurielle du besoin en information de chaque utilisateur, et
- modéliser ces espaces de communautés afin de les représenter (B1) et de les induire (B2).

Nous montrons comment chacun de ces points est un apport à la perception, la formation et au positionnement des utilisateurs dans les communautés.

Nous donnons par la suite une description générale de notre proposition avant de la détailler dans les parties III et IV.

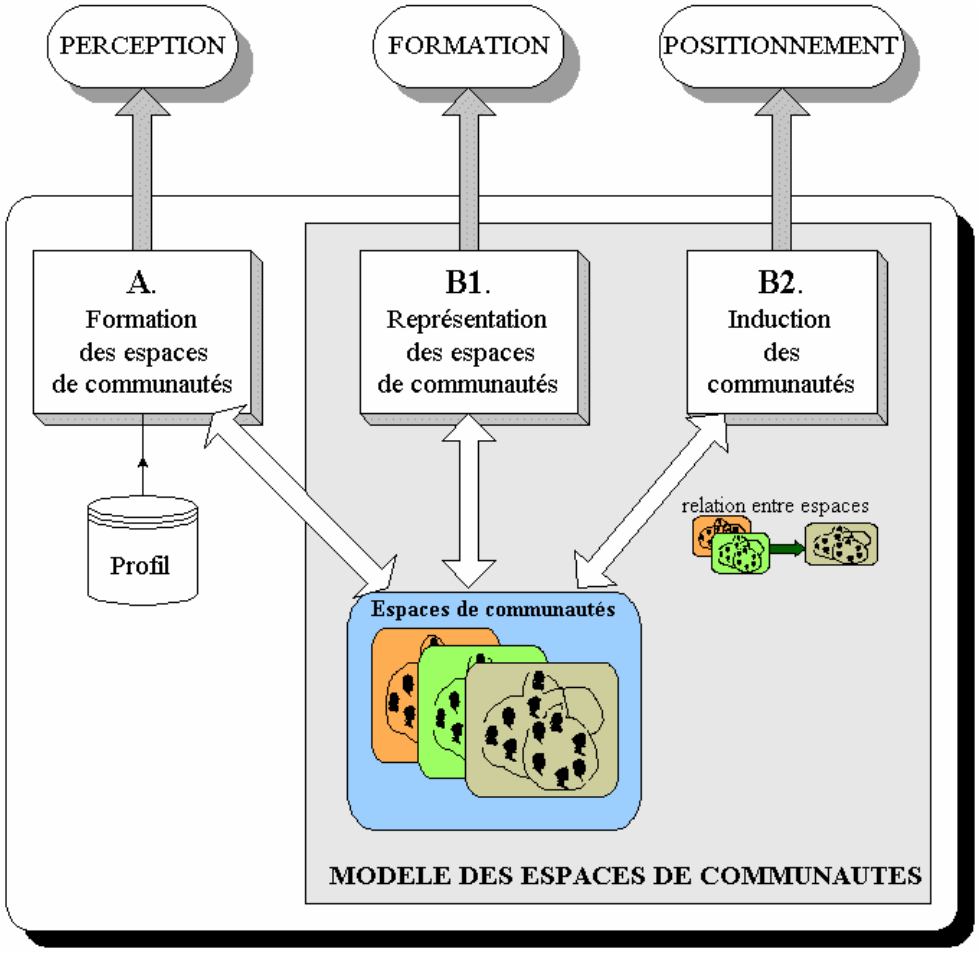


Figure 3.1 – *Notre proposition pour la perception, la formation et le positionnement des communautés d'utilisateurs des systèmes de filtrage collaboratif.*

3.2.1 Formation des espaces de communautés

La formation des espaces de communautés a pour but de construire de façon « directe » les communautés par un critère donné afin que le système génère des recommandations collaboratives pour les utilisateurs. Cette tâche de formation est très importante dans notre approche puisqu'elle alimente aussi notre modèle des espaces de communautés. Après avoir été formés, les espaces de communautés existent, évoluent et apportent une contribution significative dans tous les processus du système de filtrage collaboratif. Par exemple, outre la production des recommandations pour les

utilisateurs, les espaces de communautés peuvent être utilisés dans d'autres activités telles que l'explication des recommandations, l'interaction entre utilisateurs et système, et en particulier dans l'induction mutuelle entre communautés.

De façon générale, pour un critère particulier, les utilisateurs peuvent être regroupés par une comparaison de leurs informations ou éventuellement par un processus de formation de plusieurs étapes plus élaboré. Une analyse de ces étapes est présentée dans le Chapitre 9. En outre, afin de rendre complètement perceptibles les communautés, et plus généralement qu'elles deviennent une modalité d'interaction entre utilisateurs et système, nous proposons l'approche des « cartes de communautés » en 2D. Avec la création de telles cartes, l'utilisateur obtient une possibilité de percevoir toutes les communautés dans le système.

3.2.2 Représentation des espaces de communautés

Nous proposons la formalisation des « espaces de communautés » selon des critères variés : les communautés ne sont plus seulement formées sur la base de l'historique des évaluations des utilisateurs mais aussi sur tous les autres critères de rapprochement entre utilisateurs, qui sont disponibles dans le système.

Ainsi, nous utilisons une « table de communautés » pour représenter tous les espaces de communautés existants dans le système. Du côté des utilisateurs, le polymorphisme de positionnement dans notre approche est caractérisé par le fait que chaque utilisateur est associé à la liste des communautés auxquelles il appartient.

L'intégration de la multiplicité des critères dans notre modèle permet d'enrichir et de diversifier les recommandations générées pour les utilisateurs. Un utilisateur peut donc recevoir les recommandations de chacune des communautés auxquelles il appartient.

3.2.3 Induction des communautés

Au cours du temps, les communautés d'un utilisateur sont souvent manquantes, périmées ou douteuses que le système essaie de corriger, d'une façon ou d'une autre, afin de le positionner au mieux dans les espaces de communautés. Ce phénomène des listes de communautés imparfaites des utilisateurs reflète les difficultés dans le positionnement des utilisateurs au sein des espaces de communautés. La cause de ces difficultés peut être la complexité dans le calcul du système, dans le contexte de la multiplicité de critères, et l'imperfection ou l'indisponibilité des informations dans les profils des utilisateurs.

Ainsi, nous proposons de rattacher un utilisateur à une communauté dans un espace relatif à un certain critère à partir de ses communautés déjà connues dans d'autres espaces. En fait, nous tentons de répondre par exemple à la question : « Est-il possible de positionner un nouvel utilisateur dans l'espace du critère de l'historique des évaluations en utilisant simplement ses informations démographiques disponibles, sans même lui demander de fournir des évaluations ? ». La réponse positive à cette question permet au modèle de diminuer de beaucoup l'effort des utilisateurs dans le

positionnement, notamment en situation de démarrage à froid, et à terme cela permet aux utilisateurs de découvrir des communautés potentiellement intéressantes qu'ils ne connaissent pas encore.

Il est par ailleurs à noter que nous prenons en compte en particulier l'existence de plusieurs communautés imparfaites dans une même liste de communautés d'un utilisateur, en visant la possibilité d'exploiter la relation entre les critères ou espaces de communautés.

3.3 Plan du manuscrit

Dans la Partie II, nous passons en revue les études sur la notion de communauté. D'abord, nous discutons de la perception des communautés dans le Chapitre 4, et ensuite de la formation des communautés dans le Chapitre 5. Nous présentons dans le Chapitre 6 les approches courantes pour le problème du démarrage à froid. Enfin, le Chapitre 7 propose un bilan de l'état de l'art.

La partie III est la proposition principale de la thèse, où nous présentons notre modèle des espaces de communautés. Le Chapitre 8 comporte la représentation des espaces de communautés et l'exploitation des relations entre ces espaces pour l'induction mutuelle des communautés. Le Chapitre 9 présente un processus général de formation des communautés afin d'alimenter notre modèle des espaces de communautés. Un bilan de cette partie est présenté dans le Chapitre 10.

Dans la Partie IV, nous décrivons la plateforme COCoFil2 comme la mise en œuvre de notre modèle proposé dans la partie précédente. Nous proposons dans le Chapitre 11 l'approche des cartes de communautés en 2D pour le module de formation des communautés afin de faciliter la perception des communautés dans un système de filtrage collaboratif. Ensuite, le Chapitre 12 présente le module d'induction des communautés, qui est utile au positionnement des utilisateurs au sein des communautés en particulier dans les situations délicates comme le démarrage à froid. Le module de filtrage d'information, se basant sur notre méthode de filtrage par niveau d'accord, est présenté dans le Chapitre 13. Enfin, le Chapitre 14 donne un bilan de la mise en œuvre.

La validation de notre proposition sur le jeu de données réelles MovieLens est décrite dans la Partie V. D'abord, nous montrons en détails dans le Chapitre 15 la préparation des données pour l'expérimentation. Puis, le Chapitre 16 présente les travaux d'analyse de la performance de notre approche des cartes de communautés pour la formation des communautés. Ensuite, les travaux présentés dans le Chapitre 17 visent à analyser les relations entre les espaces de communautés formés grâce à la multiplicité des critères issus des profils, qui servent de base à notre approche d'induction des communautés. La motivation des travaux présentés dans le Chapitre 18 est de valider notre méthode basée sur le modèle des espaces de communautés pour le problème d'intégration de nouveaux utilisateurs dans un système de filtrage collaboratif. Le Chapitre 19 présente un bilan de la validation.

Dans la Partie VI, nous donnons la conclusion de cette thèse. Le Chapitre 20 présente les apports théoriques et pratiques. Enfin, les travaux à court et à moyen terme sont décrits dans le Chapitre 21.

Partie II.

Communautés : Etat de l'art

Depuis des siècles, la notion de communauté a été étudiée dans les domaines des sciences humaines et sociales [JG04]. Pourtant, on a observé en 1915 la première définition sociologique de Galpin pour les communautés rurales au niveau du commerce et des services autour d'un village [Gal15, HD59]. Depuis, il a émergé un nombre important de définitions des communautés selon tous les domaines et les points de vue. En 1972, dans leur ouvrage, Bell et Newby ont dit :

> *"… the concept of community has been the concern of sociologists for more than two hundred years, yet a satisfactory definition of it in sociological terms appears as remote as ever"* [BN72: p. 21].

Dans une analyse il y a 50 ans, Hillery a déjà recensé 94 définitions de « communauté » selon les critères : espace, relation sociale, intérêt, langue, religion, sentiment, etc. [Ham97, Hil55, TM88]. Il a trouvé que le seul point sur lequel s'accordent ces définitions est qu'une communauté est un groupe de personnes ! Hillery a au moins donné une conclusion :

> *"Most ... are in basic agreement that community consists of persons in social interaction within a geographic area and having one or more additional ties."* [Hil55: p.111, TM88].

Dans le contexte d'un système de filtrage collaboratif, les « communautés » des utilisateurs sont généralement formées par la proximité des évaluations qu'ils ont faites des informations qui leur ont été présentées. Ensuite, les communautés sont utilisées pour la production des recommandations.

Dans cette partie, nous passons en revue les principales études sur la notion de communauté dans le contexte ci-dessus. L'état de l'art est présenté en suivant naturellement les aspects développés dans le chapitre de la problématique.

Nous présentons dans le Chapitre 4 les études existantes sur la perception des communautés dans la perspective d'utiliser les communautés comme une modalité d'interaction entre utilisateurs et système. On trouve ensuite les approches de formation des communautés dans le Chapitre 5, et enfin, le Chapitre 6 est dédié au problème du démarrage à froid.

Chapitre 4

Perception des communautés

Avant de présenter les études relatives à la perception des communautés, nous expliquons notre intérêt pour ces études, dans l'optique d'une interaction entre utilisateurs et système. Nous nous appuyons sur l'hypothèse que certaines limitations de l'approche collaborative peuvent être dépassées en amplifiant la connaissance réciproque des utilisateurs, et plus globalement leur conscience de la communauté à laquelle ils appartiennent.

A titre d'exemple, le démarrage à froid peut être compensé par une période de « filtrage collaboratif actif » [ME95] où les utilisateurs qui se connaissent s'échangent des recommandations. Ou encore, concernant le problème du rapport coût-bénéfice défavorablement perçu par l'utilisateur, il se demande souvent, en évaluant des recommandations, si ses efforts seront payés en retour à court ou à moyen terme. Le faible bénéfice ressenti tient entre autres au fait que les recommandations ne véhiculent pas les informations permettant de comprendre les raisons de la prédiction, y compris celles permettant de découvrir les membres des communautés les plus proches. De plus, lorsqu'un utilisateur n'est pas satisfait des recommandations que lui fournit le système, les approches classiques limitent les actions de l'utilisateur en présentant le processus de filtrage comme une boîte noire. Si le système lève le voile sur la formation des communautés d'utilisateurs, cela permet d'offrir à l'utilisateur des moyens pour rectifier son profil de façon plus radicale et éclairée.

En d'autres termes, nous partageons l'opinion de Herlocker sur l'importance de l'explication des recommandations du système pour que les utilisateurs acceptent ces recommandations, et par conséquent les évaluent afin de construire des communautés [Her00 : Chapter 5]. Il est à noter que

l'explication du système est particulièrement indispensable pour les recommandations dans les domaines risqués comme l'assurance, l'investissement, et même les voyages. Par expérience, Herlocker montre que la capacité d'explication du système contribue significativement à la confiance des utilisateurs dans le système en général et dans les recommandations en particulier. Dans son approche d'explication des recommandations, Herlocker met en évidence le rôle de la perception des communautés :

> *« … It is in performing step (2) (positionner des utilisateurs au sein des communautés) that ACF (Automated Collaborative Filtering) systems show their true value over normal human word-of-mouth recommendations, with ACF systems being able to examine thousands of potential soulmates, and choose the most similar of the bunch. What do we have to do to help the user determine if the ACF system has identified the correct set of neighbors for the user's current context of need? The process that is used to locate other people with similar profiles is one key to the success of the collaborative filtering technology… An explanation could give the user the ability to examine the ratings of the chosen neighbors and when the user discovers the offending neighbor, he can disregard the prediction, or perhaps the system will allow him to manually remove that neighbor from consideration…»* [Her00 : p. 82-83].

Donc, nous nous intéressons aux études consistant à tirer profit de la présence des utilisateurs dans le système en s'intéressant à leur organisation en communautés, et nous présentons dans ce chapitre les principales études existantes sur deux niveaux de perception des communautés dans un système de filtrage collaboratif : les communautés invisibles comme un facteur interne de recommandation et la perception partielle des communautés.

4.1 Communautés invisibles comme un facteur interne de recommandation

4.1.1 Communautés pour le calcul de prédiction des recommandations

Les communautés dans la plupart des systèmes de filtrage collaboratif sont considérées comme un facteur interne pour le calcul de la prédiction des recommandations [BHK98, GON+92, HKJ+99, RIS+94]. Elles n'interviennent que rarement dans d'autres processus du filtrage collaboratif. En général, les communautés sont complètement invisibles pour les utilisateurs. Le fait que les communautés ne sont pas exploitées en tant que telles, conduit à limiter les performances du système de filtrage collaboratif.

4.1.2 Communautés dans l'explication des recommandations

D'après Herlocker, le problème de la masse critique, qui génère éventuellement de mauvaises recommandations, est vraisemblablement la cause la plus importante de la faible acceptation des

utilisateurs vis-à-vis des systèmes de recommandation dans les domaines risqués [Her00]. Afin de pallier ce problème lié au nombre d'évaluations, on trouve dans la littérature les travaux dédiés à la construction d'interfaces conviviales encourageant les utilisateurs à fournir de plus en plus d'évaluations [CSP03, SKR02, SS02]. De l'autre côté, Herlocker focalise sur la capacité d'explication des recommandations aux utilisateurs, afin de renforcer leur confiance et par conséquent les encourager à évaluer les recommandations reçues [Her00, HKR00]. En général, les travaux de Herlocker montrent que les utilisateurs font facilement confiance à leurs communautés pour accepter les recommandations.

En effet, Herlocker et ses collègues à l'université de Minnesota ont réalisé deux expériences [Her00, HKR00], sur le jeu de données réelles du système MovieLens [MovieLens] pour analyser l'acceptation des utilisateurs quant aux recommandations. D'abord, afin de motiver leurs études, ils ont invité 210 utilisateurs volontaires à évaluer des films indépendants de leurs goûts. Ils ont, à la fin, constaté que 86% des personnes invitées souhaitent des explications sur les recommandations pour donner leurs évaluations.

Nous faisons de plus un autre constat en observant les données de MovieLens fournies par le groupe de recherche de Herlocker à l'université de Minnesota : les utilisateurs n'évaluent que les recommandations qui leurs plaisent bien. Cela confirme que la capacité d'explication des recommandations conditionne la motivation des utilisateurs à évaluer des recommandations, ce qui est au cœur du problème de masse critique.

Partant de ce résultat, les auteurs veulent étudier la meilleure façon d'expliquer les recommandations pour que les utilisateurs les acceptent mieux. Herlocker propose dans sa thèse le modèle de boîte blanche[4] pour la conception d'explication des recommandations. Brièvement, ce modèle comporte des guides pour l'explication des trois processus, y compris les données impliquées, du système de filtrage collaboratif. Afin d'éviter une surcharge supplémentaire d'information via le modèle, Herlocker et ses collègues ont fait une expérience sur les informations explicatives à présenter aux utilisateurs pour qu'ils acceptent et évaluent des recommandations. Ils ont utilisé 21 types d'explication en combinant les deux dimensions suivantes : a) les données, par exemple évaluations des communautés, performance passée du système, informations thématiques, etc., et b) les formes, par exemple types de graphique.

Il y a eu 78 utilisateurs qui ont participé à cette expérience. Les résultats finaux ont montré que les explications relatives aux communautés sont très appréciées par les utilisateurs.

En résumé, nous trouvons que la présence des communautés a été prise en compte dans l'approche d'explication de Herlocker afin de répondre au problème de la masse critique. Dans cette approche, la notion de communautés devient plus concrète pour les utilisateurs. On remarque toutefois que les communautés dans les travaux de Herlocker restent toujours invisibles, et ne sont utilisées que de façon indirecte.

---

[4] Dans sa thèse, Herlocker propose également le modèle de boîte noire. Mais, dans le contexte de la perception des communautés, nous ne nous intéressons pas à ce modèle.

4.2 Perception partielle des communautés

Dans cette section, nous décrivons les approches qui offrent aux utilisateurs une perception partielle de leurs communautés actuelles. Nous commençons par une extension du filtrage collaboratif classique permettant aux utilisateurs d'échanger des informations intéressantes. Enfin, nous présentons la plateforme de filtrage collaboratif orientée vers communautés COCoFil, de laquelle partent nos études dans cette thèse.

4.2.1 Filtrage collaboratif actif

En 1995, Maltz et al. ont proposé le « filtrage collaboratif actif » (*Active Collaborative Filtering*) [ME95] qui permet la collaboration explicite entre utilisateurs.

D'après les auteurs, les systèmes comme Tapestry [GON+92], GroupLens [RIS+94], etc. sont des systèmes de filtrage collaboratif « passif » du fait qu'il n'y a aucune communication directe entre les personnes qui évaluent un document donné et les lecteurs potentiels. Partant des principes originaires de Tapestry [GON+92], les auteurs ont conçu un système de filtrage collaboratif actif permettant à l'utilisateur d'envoyer lui-même à sa propre communauté de collègues ou amis, les *indicateurs* des documents intéressants.

Normalement, un indicateur contient un lien hypertexte vers le document original, les informations contextuelles comme le titre, en particulier l'expéditeur et les commentaires facultatifs pour aider l'utilisateur à juger préalablement l'intérêt et la pertinence potentiels du document avant d'y accéder.

Ainsi, le filtrage collaboratif actif se fonde sur l'intention de partager les informations entre les personnes qui se connaissent assez bien. Chacune doit connaître les centres d'intérêt des membres de sa « petite » communauté.

Le filtrage collaboratif actif ne souffre pas des problèmes du démarrage à froid et de la masse critique, car les communautés sont construites par les utilisateurs eux-mêmes. Quant au rapport coût-bénéfice, l'impact du filtrage collaboratif actif est un peu délicat à évaluer. D'une part, grâce à la possibilité pour chacun de fournir des informations pertinentes, le fardeau de trouver de nouveaux documents intéressants devient une tâche partagée dans la communauté. D'autre part, il risque d'y avoir des « profiteurs » qui découragent les autres. Il faut par ailleurs noter le rôle passif des destinataires dans la réception des informations ennuyeuses.

En conclusion, on constate une perception partielle des communautés dans le filtrage collaboratif actif. En pratique, l'application de ce filtrage se limite aux petites communautés des proches, des amis, des collègues, etc., où les membres connaissent bien les intérêts courants de chacun. D'ailleurs, sans aucune aide du système dans la formation des communautés, l'utilisateur n'est pas capable de faire connaissance avec d'autres personnes.

4.2.2 Plateforme COCoFil

Les travaux sur le filtrage collaboratif se focalisent généralement sur les fonctions internes, par exemple le calcul des prédictions, tant pour les efforts d'amélioration des techniques existantes, que pour l'évaluation des performances des systèmes [BHK98, FHH+00, HKJ+99, RIS+94]. Il émerge récemment des travaux alternatifs pour améliorer et mesurer les performances [Can02, CS02, JSZ+03, HKT+04]. Cependant, Berrut, Denos et d'autres chercheurs du laboratoire CLIPS à Grenoble ont suivi une autre direction pour rendre plus actif le rôle des utilisateurs, et en particulier rendre plus contributives les communautés dans un système de filtrage collaboratif. Ils ont développé, dans le cadre du projet européen TIPS[5], la plateforme de filtrage collaboratif COCoFil tournée vers l'innovation en matière de fonctionnalités interactives visant à améliorer le rapport coût-bénéfice. Cela est fait notamment au travers de fonctionnalités orientées vers la notion de communauté et la possibilité de paramétrer personnellement la perception des communautés.

Les travaux de cette thèse partent du point de vue de la plateforme COCoFil sur l'enrichissement de l'exploitation des communautés dans un système de filtrage collaboratif. Ainsi, nous présentons dans cette section son architecture et certaines de ses caractéristiques afin de fournir une première vision de nos objectifs dans cette thèse.

4.2.2.1 Architecture de COCoFil

Comme le nom l'indique, COCoFil (*Community-Oriented Collaborative Filtering*) est une plateforme de filtrage collaboratif orientée vers la communauté [DBG+04]. D'après les auteurs, certaines limitations du filtrage collaboratif peuvent être dépassées en développant de nouveaux principes d'interaction, comme par exemple le principe du filtrage collaboratif actif intégré dans la plateforme, qui contribue à limiter l'effet du démarrage à froid, en permettant aux utilisateurs qui se connaissent de se recommander directement des documents, comme ils le feraient de manière naturelle entre collègues ou amis.

Ainsi, l'objectif principal de cette plateforme est d'intégrer toutes les idées alternatives existantes ou nouvelles, s'articulant autour de la notion de communauté, afin d'en valider le principe et d'en évaluer l'utilité. Elle offre des fonctionnalités innovantes orientées vers les communautés pour tirer un meilleur bénéfice des possibles relations entre utilisateurs du système.

La plateforme COCoFil comporte trois modules : Filtrage collaboratif, Paramétrage et Gestion de contact. Dans ce chapitre dédié à la perception des communautés, la présentation ci-après des caractéristiques de cette plateforme se limite au dernier module « Gestion de contact » relatif à ce sujet (voir Figure 4.1). Ce module assure l'identification dans la plateforme COCoFil, c'est-à-dire qu'il permet d'une part aux utilisateurs de saisir leurs informations personnelles, et d'autre part au système d'identifier l'utilisateur lors de son accès. Grâce à ce module, l'utilisateur peut en outre organiser son carnet d'adresses et échanger des recommandations avec d'autres utilisateurs dans le cadre du filtrage collaboratif actif.

---

[5] http://tips.sissa.it

Normalement, les fonctionnalités de COCoFil sont structurées selon les grandes familles d'activités que l'on peut y pratiquer. De plus, des passerelles sont établies entre les activités pour améliorer l'accessibilité des fonctionnalités.

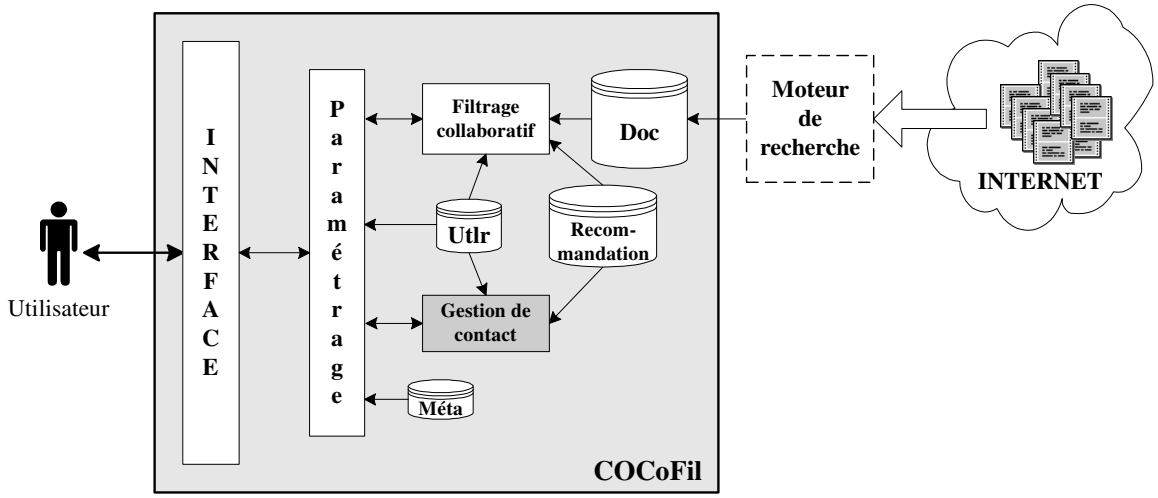


Figure 4.1 – *Architecture de la plateforme COCoFil.*

4.2.2.2 Identification et confidentialité

Afin d'obtenir un compte d'accès, l'utilisateur doit fournir certaines informations personnelles telles que nom, prénom, adresse électronique, au moment de l'inscription volontaire. Toutes ces informations sont utiles non seulement pour l'identification mais aussi pour la crédibilité des recommandations dans le filtrage collaboratif actif. En effet, l'identification ou au moins la pseudonymie enrichissent la perception des recommandations circulant dans la communauté. Par exemple, il semble naturel qu'un utilisateur s'intéresse plus aux suggestions de personnes identifiées qui lui ont souvent adressé des recommandations pertinentes par le passé ; de même, il aura sans doute tendance à négliger les recommandations anonymes.

Cela ne veut pourtant pas dire que la confidentialité n'est pas assurée dans COCoFil. L'utilisateur a toujours la possibilité d'indiquer qu'il souhaite l'anonymat : ses informations personnelles ne seront alors pas divulguées. Aussi, il peut choisir de conserver l'anonymat pour ses évaluations (voir Figure 4.2), sachant qu'il pourra très simplement, lors d'une évaluation particulière, choisir l'option inverse.

Ces possibilités de configuration de la confidentialité sont d'autant plus importantes que la plateforme intègre des fonctionnalités destinées à rendre plus perceptible la communauté. Par ailleurs, l'utilisateur peut décider des informations qu'il souhaite divulguer ou non, qu'il s'agisse d'informations relatives à son identité, ou même des évaluations qu'il fait pour chaque document. Le

système propage ces contraintes de confidentialité, limitant ainsi certaines fonctionnalités de nature à faire se connaître les utilisateurs entre eux.

Ces fonctionnalités particulièrement orientées vers la communauté permettent notamment de connaître les utilisateurs qui ont attribué une note semblable à celle donnée par l'utilisateur pour un document donné, ou ceux qui d'une manière plus globale ont un profil similaire au sien.

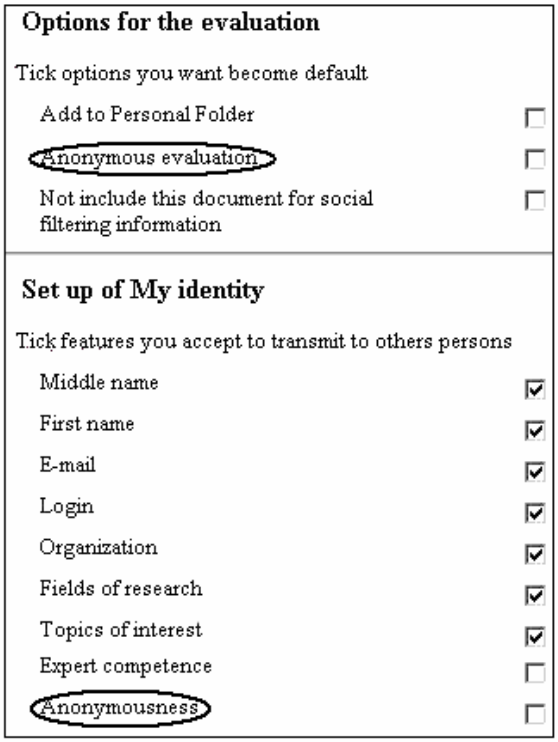


Figure 4.2 – *Paramètres ajustables pour chaque évaluation de document.*

4.2.2.3 Carnet d'adresses

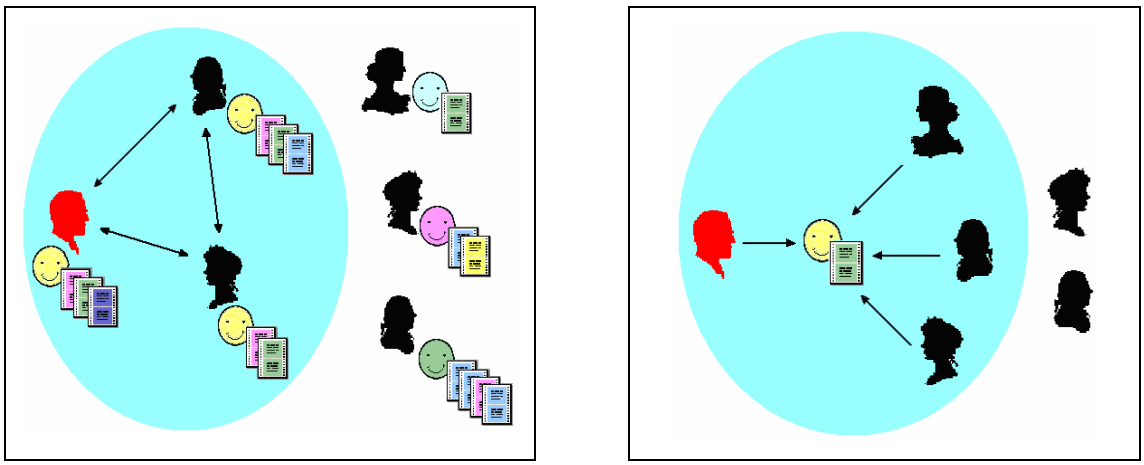
Selon le principe du filtrage collaboratif actif, lorsqu'un utilisateur trouve des documents plus ou moins intéressants pour certains autres utilisateurs qu'il connaît, il peut les leur recommander. La plateforme COCoFil offre des outils permettant aux utilisateurs de faire du filtrage actif dans les meilleures conditions : effort limité via un groupe de destinataires par défaut, possibilité d'enrichir son carnet d'adresses grâce aux passerelles entre le carnet d'adresses et les fonctionnalités de perception des autres.

Chaque utilisateur de la plateforme COCoFil possède un *carnet d'adresses* qui permet de répertorier des informations utiles (nom, prénom, surnom, adresse électronique, etc.) des personnes « intéressantes », et évidemment « connues », avec lesquelles il souhaite échanger régulièrement des recommandations.

En particulier, un carnet d'adresses dans COCoFil peut *être organisé en groupes* par son propriétaire afin de faciliter le filtrage collaboratif actif. Cette possibilité soulage considérablement la tâche d'envoi des recommandations du fait que l'utilisateur peut envoyer des recommandations à un groupe de personnes au lieu de le faire individuellement. Il existe en outre un groupe spécial dans chaque carnet d'adresses, nommé « Default group » qui contient des destinataires par défaut permettant une recommandation en un seul clic : l'effort à produire pour le filtrage actif est ainsi réduit au minimum pour ce groupe.

4.2.2.4 Perception des autres

Dans la plateforme COCoFil, l'utilisateur a la possibilité de percevoir ses communautés. En effet, il peut consulter la liste des personnes qui ont des profils similaires au sien, et ainsi connaître de nouvelles personnes ayant les mêmes centres d'intérêt que lui. Par exemple, les flèches bidirectionnelles dans la Figure 4.3.a illustrent la similarité de profils. On trouve aussi la fonctionnalité de type « passerelle », qui permet à partir de cette liste, de compléter simplement son carnet d'adresses. On peut espérer que cette fonctionnalité permet à l'utilisateur de mieux cibler ses recommandations directes dans le filtrage actif. Bien sûr, cette liste est soumise au filtre des choix d'anonymat faits par les utilisateurs ! Une personne ayant demandé l'anonymat n'apparaîtra dans aucune liste. Pourtant, son profil sera toujours pris en compte dans le calcul de prédiction du moteur.



a) Personnes ayant les profils similaires

b) Personnes ayant la même évaluation sur un document particulier

Figure 4.3 – *Perception partielle de communautés dans la plateforme COCoFil.*

Dans le même esprit de perception de la communauté, l'utilisateur peut connaître la liste des personnes ayant fait la même évaluation que lui pour un document particulier, comme illustré dans la Figure 4.3.b. Cette fonctionnalité contribue également à de meilleures recommandations par filtrage actif. En effet, si un utilisateur constate qu'un nouveau document est très proche d'un autre qu'il a déjà jugé pertinent, il peut décider, au lieu de le recommander à sa liste de destinataires par défaut, de

s'inspirer de la liste des personnes ayant fait la même évaluation que lui sur l'ancien document. A nouveau, cette fonctionnalité est soumise aux règles d'anonymat, non seulement cette fois pour ce qui est de l'identité des personnes, mais aussi par rapport à l'anonymat des évaluations : les évaluations anonymes n'apparaîtront pas.

Lorsqu'on consulte les utilisateurs ayant fait une évaluation similaire ou ayant un profil similaire, est offerte la possibilité d'ajouter directement les utilisateurs en question au carnet d'adresses. Cela constitue un exemple de passerelle entre fonctionnalités, qui contribue à concrétiser la notion de communauté.

En résumé, COCoFil est une plateforme de filtrage collaboratif particulièrement orientée vers la communauté. En effet, elle intègre des fonctionnalités destinées à mieux exploiter la notion de communauté d'utilisateurs. Toutes ces fonctionnalités ont pour but de permettre aux utilisateurs d'intervenir dans le processus de recommandation directement et de se découvrir entre membres d'une même communauté. Ainsi, les utilisateurs ont une perception partielle de leur propre communauté, et ils ignorent les autres communautés. Par conséquent, ils sont incapables d'explorer des communautés potentiellement pertinentes pour eux.

4.3 Conclusion

Pour conclure le présent chapitre, les communautés dans les systèmes de filtrage collaboratif restent souvent invisibles à cause de la restriction inutile à l'exploitation pour le calcul de prédiction. La Figure 4.4 nous montre que certaines études s'en remettent à la perception partielle des communautés afin de répondre également aux problèmes particuliers du filtrage collaboratif tels que l'explication pour la confiance d'utilisateurs, la masse critique, le démarrage à froid, le rapport coût-bénéfice, etc. Nous pensons que le fait de rendre complètement perceptibles les communautés, étant un facteur clé du filtrage collaboratif, pourrait fournir un élément de base pour une solution plus radicale aux problèmes de cette technique de filtrage.

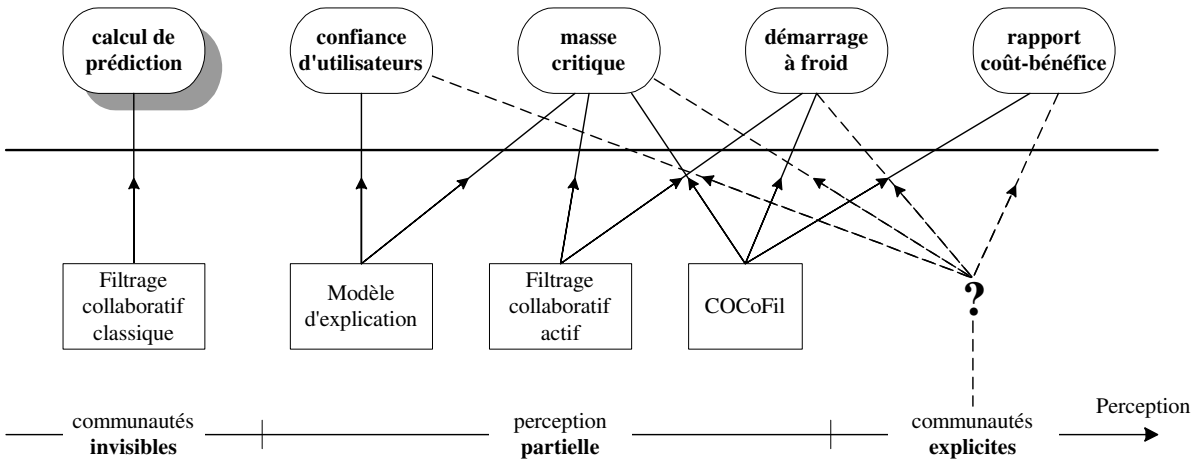


Figure 4.4 – *Niveau de perception des communautés*.

Chapitre 5

Formation des communautés

L'objectif principal de la formation des communautés est de regrouper les utilisateurs en fonction de leur proximité des évaluations afin de générer des recommandations collaboratives. Selon Perugini et al. [PGF03], la dimension de connexion des utilisateurs peut être considérée comme une mesure de qualité d'un système de filtrage collaboratif, car la qualité des recommandations collaboratives dans le système est conditionnée par la qualité de la formation des communautés.

Dans ce chapitre, nous passons en revue les trois approches les plus répandues pour former des communautés par le critère de proximité des évaluations : des voisins les plus proches, probabiliste et des réseaux.

5.1 Approche des voisins les plus proches

Pour la formation de communautés, l'approche des voisins les plus proches (*Neighborhood-based Method*) est actuellement la plus populaire [BHK98, HKJ+99, JCS04, RIS+94]. Dans cette approche, le système essaie d'identifier les meilleurs voisins pour un utilisateur donné, en traitant la matrice des évaluations V_{mxn} (voir Figure 5.1) dont chaque ligne correspond à l'historique des évaluations d'un utilisateur.

Plus précisément, étant donné l'utilisateur *u*, l'approche des voisins les plus proches se réalise en général en deux étapes :

E1. Mesurer la (dis)similarité entre l'utilisateur *u* et les autres, et

E2. Sélectionner les meilleurs voisins en fonction de la (dis)similarité entre l'utilisateur *u* et les autres calculée dans l'étape précédente.

	d_1	…	d_j	…	d_n
u_1	$v_{1,1}$		$v_{1,j}$		$v_{1,n}$
…					
u_i	$v_{i,1}$		$v_{i,j}$		$v_{i,n}$
…					
u_m	$v_{m,1}$		$v_{m,j}$		$v_{m,n}$

$v_{i,j}$: évaluation de u_i sur d_j

Figure 5.1 – *Matrice des évaluations* V_{mxn}.

Pour la première étape E1, on constate dans la littérature l'utilisation des mesures de (dis)similarité variées telles que la corrélation de Pearson, la corrélation de Spearman, le cosinus, et autres possibilités [BHK98, HKJ+99]. Ainsi, la matrice V_{mxn} est traitée par paires de lignes (V_u, V_i), par exemple avec la corrélation de Pearson :

$$corrélation(V_u, V_i) = \frac{\sum_j (v_{u,j} - \bar{v}_u).(v_{i,j} - \bar{v}_i)}{\sqrt{\sum_j (v_{u,j} - \bar{v}_u)^2 . \sum_j (v_{i,j} - \bar{v}_i)^2}} \quad (5.1)$$

où, $\bar{v}_u$ et $\bar{v}_i$ sont les scores moyens des utilisateurs *u* et *i* respectifs.

Alors, pour l'utilisateur *u*, on obtient une table ordonnée par (dis)similarité comme illustré dans la Figure 5.2.

Parmi les mesures citées, la corrélation de Pearson est la plus utilisée en raison de sa performance dans le calcul de prédiction [Her00].

Enfin, afin de sélectionner les meilleurs voisins pour l'utilisateur *u* dans l'étape E2, il y a deux stratégies possibles. D'abord, on peut utiliser un seuil δ pour la proximité entre utilisateurs (voir Figure 5.2 et Figure 5.3) comme dans le système Ringo [SM95]. Cette méthode permet de contrôler la qualité des communautés, mais la taille des communautés risque d'être très faible pour calculer la prédiction, si le seuil de proximité δ est trop fort.

Une autre alternative est d'utiliser un seuil *K* pour fixer la taille maximale de l'ensemble des voisins (K voisins les plus proches) [RIS+94]. Selon ses travaux expérimentaux, Herlocker propose un seuil *K* qui varie de 20 à 50 en raison de la précision des prédictions [Her00].

	Utilisateur	s_i = *dissimilarité*(V_u, V_i)	
	u_1	s_1	
	…	…	
	u_d	s_d	$(s_d \leq \delta)$
	…	…	
K voisins les plus proches	u_K	s_K	
	…	…	
	…	…	
	u_t	s_t	

Figure 5.2 – *Table ordonnée des (dis)similarités entre l'utilisateur u et tous les autres* $(s_i \leq s_j,\ i \leq j)$.

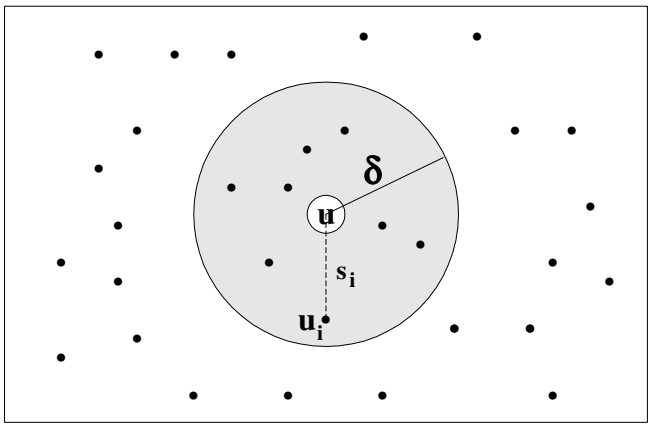


Figure 5.3 – *Illustration de sélection des voisins les plus proches par le seuil* δ *(en 2D).*

En résumé, l'approche des voisins les plus proches exploite des relations explicitement disponibles dans les profils des utilisateurs. Elle est simple à implémenter, et efficace dans la plupart des cas. Mais, cette approche souffre du coût de calcul, car le système doit traiter à chaque fois la matrice entière des évaluations afin de former des communautés.

5.2 Approche probabiliste

Afin de créer des communautés pour le filtrage collaboratif, on trouve également l'approche probabiliste, moins utilisée [BHK98]. Dans cette approche, la prédiction de satisfaction de l'utilisateur

pour un document, en tenant compte des évaluations de sa communauté, est considéré comme une tâche de classification traditionnelle. En principe, le système essaie de construire par apprentissage un modèle probabiliste à partir des évaluations des utilisateurs. Ensuite, le système applique ce modèle afin de prédire la satisfaction, par exemple *like* ou *dislike*, de l'utilisateur sur un document qu'il n'a pas encore évalué.

En général, on implique la formation des communautés dans la construction du modèle de prédiction. Par exemple, Miyahara et Pazzani ont proposé d'utiliser la classification Bayésienne naïve, qui est une méthode de classification supervisée simple et efficace à la fois, pour construire le modèle de prédiction, en considérant les utilisateurs comme des attributs ou caractéristiques de données [MP00]. Dans ce sens, la formation des communautés pour le calcul de la prédiction correspond à la sélection des caractéristiques les plus discriminantes (*Feature Selection*) comme on voit souvent dans le domaine de classification supervisée.

Plus précisément, les auteurs utilisent une matrice des évaluations binaires V_{mxn} (voir exemple dans la Figure 5.4). Cette matrice est transformée en une autre matrice V' de la façon suivante : chaque ligne V_i dans la matrice originale est divisée en deux $V'_{i,like}$ et $V'_{i,dislike}$ comme illustré dans la Figure 5.5, sauf la ligne V_4 de l'utilisateur dont on veut prédire la satisfaction sur le document d_5. En d'autres termes, les profils sont divisés en deux parties : évaluations positives et évaluations négatives.

	d_1	d_2	d_3	d_4	$\boldsymbol{d_5}$
u_1	*like*	*dislike*	*dislike*		*like*
u_2	*dislike*			*dislike*	*dislike*
u_3		*like*	*like*		*like*
$\boldsymbol{u_4}$	*like*	*dislike*	*like*	*like*	?

Figure 5.4 – *Matrice des évaluations binaires V_{mxn}.*

	d_1	d_2	d_3	d_4	$\boldsymbol{d_5}$
$f_1 = (u_1, \text{like})$	1	0	0	0	1
$f_2 = (u_1, \text{dislike})$	0	1	1	0	0
$f_3 = (u_2, \text{like})$	0	0	0	0	0
$f_4 = (u_2, \text{dislike})$	1	0	0	1	1
$f_5 = (u_3, \text{like}))$	0	1	1	0	1
$f_6 = (u_3, \text{dislike})$	0	0	0	0	0
$\boldsymbol{u_4}$	*like*	*dislike*	*like*	*like*	?

Figure 5.5 – *Matrice de transformation V'.*

En se basant sur l'hypothèse de l'indépendance des attributs, ou caractéristiques, dans la classification Bayésienne naïve, la valeur de $\Pr(C|f_1, f_2, ..., f_t)$ est proportionnelle à :

$$\Pr(C)\prod_{s=1}^{t}\Pr(C|f_s)$$

où, C est la classe à prédire (*like, dislike*), et f_s est une caractéristique (voir Figure 5.5).

Ensuite, les auteurs réalisent la sélection des caractéristiques f_s les plus discriminantes pour le modèle de prédiction, correspondant à la sélection des utilisateurs dans l'approche des voisins les plus proches, en fonction du gain d'information [Qui86].

En résumé, les avantages de l'approche probabiliste sont la compacité du modèle, et donc la rapidité du calcul de prédiction. En revanche, ces techniques probabilistes sont très compliquées, et le processus d'apprentissage est souvent long. De plus, les soi-disant communautés dans de telles techniques dépendent du document en considération. Ainsi, cette notion de communauté dans l'approche probabiliste est un peu différente des autres approches où les communautés sont relativement indépendantes des documents.

5.3 Approche des réseaux

Au contraire des relations explicitement disponibles dans les profils, d'autres relations peuvent être découvertes à partir de données véhiculant de façon implicite un réseau social, comme par exemple les affiliations de personnes que l'on peut trouver sur le Web (qui est affilié à quelle institution). En général, le processus de découverte des relations implicites entre utilisateurs se compose de trois phases :

E1. Collecter et fouiller des données transactionnelles, par exemple communication, messages, favoris, évaluations, etc.,

E2. Reconnaître et modéliser des intérêts souvent implicites, et induire les communautés existantes, et

E3. Explorer et exploiter des communautés.

Nous présentons dans la suite les travaux représentatifs de l'approche des réseaux.

5.3.1 Réseaux sociaux

Un *réseau social* est défini comme un graphe non orienté dont les nœuds appartiennent à une seule classe d'objets ou personnes, et les arcs ont le même type de relation, par exemple « être ami ». On essaie d'identifier des communautés en faisant émerger des relations sociales existant dans le graphe. Par exemple, le système Hidden Web-Referral Web [KSS97a, b] a pour but de chercher des

ressources sur le Web comme les experts, les documents, etc. en explorant des réseaux sociaux. Dans cette approche, on construit un réseau social en reliant deux personnes dont les noms apparaissent à proximité dans une page Web.

L'objectif de ce système est d'aider l'utilisateur à explorer de façon interactive le réseau social afin de trouver :

(a) une chaîne de références vers un expert particulier,

(b) des experts sur un sujet donné, et

(c) des documents concernant un certain sujet attribué à un expert relativement proche.

En général, les études des réseaux sociaux visent à détecter les relations sociales existantes dans les données plutôt qu'à modéliser explicitement les intérêts des utilisateurs [PGF03].

5.3.2 Fouille et exploration de structures

Dans l'approche de fouille et d'exploration de structures (*Mining and Exploiting Structure*) [MKR03], le système transforme un réseau biparti R (voir Figure 5.6), qui représente la matrice des évaluations V_{mxn}, ayant 2 classes de nœuds {personne p_i} et {document d_j}, en un réseau social uni parti G_s généralement en 3 étapes :

E1. Fouiller le réseau d'affiliation,

E2. Identifier, modéliser/extraire le réseau social G_s,

E3. Rattacher les deux réseaux en G_r pour l'exploration et l'exploitation dans la production de recommandations, par exemple déterminer la distance moyenne entre personne et objet.

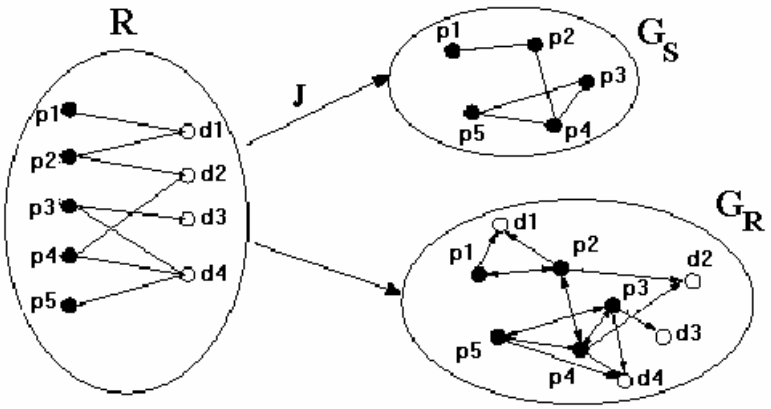


Figure 5.6 – *Jumping connection.*

Mirza et al. [MKR03] ont proposé la technique « hammock jump » *J*, qui relie deux personnes ayant un certain nombre d'évaluations communes *w*, pour induire le réseau social. Cette approche permet d'une part de reconnaître et d'explorer des structures dans l'ensemble des évaluations, et d'autre part de calibrer et d'évaluer la performance du système en termes de connexion des utilisateurs avec des documents, en analysant les caractéristiques structurelles des graphes G_s et G_r : par exemple, le nombre de personnes liées par la technique appliquée dans le système, le rapport entre le paramètre *w* et la taille de l'ensemble d'apprentissage qu'un utilisateur doit fournir au système pour recevoir des recommandations, etc.

En résumé, les travaux relevant de l'approche des réseaux permettent de former des communautés par transitivité sans avoir besoin des évaluations en commun entre utilisateurs, ce qui est une solution importante pour le problème de la masse critique. Par contre, la notion de communautés dans les réseaux est beaucoup moins forte que dans l'approche des voisins les plus proches.

5.4 Conclusion

Pour conclure, ce que nous retenons du présent chapitre est que dans la plupart des systèmes de filtrage collaboratif, l'historique des évaluations est le seul critère utilisé pour former des communautés, quelque soit la méthode.

Chapitre 6

Démarrage à froid

En général, on distingue trois types de démarrage à froid pour un système de filtrage collaboratif :

– le démarrage à froid pour un nouveau système (« new system »), où les performances des systèmes sont très mauvaises en raison de l'absence d'informations sur lesquelles fonder le processus de filtrage personnalisé. Ce problème est généralement traité en exploitant des données externes, données dont on ne dispose pas toujours, selon le cadre applicatif [MSR04].

– le démarrage à froid pour un nouveau document (« new item ») : c'est un problème spécifique à l'approche collaborative, pour laquelle les objets à recommander ne sont décrits que par les évaluations fournies par les utilisateurs. Ce problème est généralement traité en combinant une approche de filtrage basé sur le contenu avec le filtrage collaboratif (approche hybride), par exemple en utilisant la similarité, au niveau du contenu, entre documents pour estimer la satisfaction des utilisateurs sur le nouveau document en fonction de leurs évaluations sur certains documents assez proches [SPU01] ; ou en introduisant des agents intelligents qui évaluent les documents automatiquement [GSK+99].

– le démarrage à froid pour un nouvel utilisateur (« new user ») : du fait qu'il n'a pas encore donné d'évaluations, sa communauté par l'historique des évaluations est toujours inconnue, ce qui conduit à l'impossibilité de calculer des recommandations pour lui.

Dans cette thèse, nous nous intéressons essentiellement au dernier type de démarrage à froid, pour un nouvel utilisateur[6], et nous présentons dans le présent chapitre les approches courantes qui répondent à la question : « Comment le système peut-il positionner un nouvel utilisateur dans une communauté formée sur la base de l'historique des évaluations alors que cette personne n'a pas encore fourni d'évaluations ? ».

6.1 Filtrage collaboratif actif

En donnant aux utilisateurs les possibilités de former eux-mêmes des communautés par la connaissance de personnes, collègues ou amis, les systèmes s'appuyant sur le filtrage collaboratif actif [ME95] ne souffrent absolument pas du démarrage à froid. Néanmoins, cette approche ne peut s'appliquer qu'aux petites communautés où chaque personne connaît parfaitement les centres d'intérêt des autres.

6.2 Approche des recommandations exploratoires

L'approche classique pour le démarrage à froid consiste à demander au nouvel utilisateur d'évaluer un ensemble des recommandations exploratoires afin d'obtenir une communauté initiale et de premières recommandations. Dans cette approche, la construction de l'ensemble exploratoire joue un rôle capital dans le succès du système.

Rashid et ses collègues à l'université de Minnesota ont identifié deux critères importants pour évaluer un ensemble exploratoire : l'effort demandé aux utilisateurs pour évaluer, et la précision du positionnement ou des recommandations générées [RAC+02]. Ils ont également présenté certaines méthodes de base pour sélectionner des recommandations exploratoires.

a) *Au hasard* : Les recommandations sont choisies au hasard par le système ou par l'utilisateur lui-même [MovieLens]. Cette méthode permet de capturer la préférence globale de l'utilisateur sur l'ensemble des documents en raison du choix sans biais. Mais, elle demande beaucoup d'effort à l'utilisateur en cas de documents étrangers.

b) *Choix personnel* : Dans le système MovieFinder, les recommandations sont choisies par l'utilisateur lui-même [NH98]. A l'inscription, le nouvel utilisateur peut citer les films qu'il aime en particulier et/ou qu'il n'aime absolument pas. Cette méthode est similaire à l'application du filtrage basé sur le contenu dans 6.3.1. Au contraire de la première méthode, le choix personnel ne demande pas trop d'effort à l'utilisateur. Néanmoins, on ne peut capturer qu'une préférence limitée de l'utilisateur.

---

[6] Ainsi, dans le reste du manuscrit, le terme « démarrage à froid » est utilisé pour désigner, sauf précision autre, le problème d'intégration d'un nouvel utilisateur dans le système.

c) *Popularité* : Le système sélectionne les documents les plus récents ou les plus évalués dans le passé [MovieLens]. Cette méthode demande peu d'effort à l'utilisateur, car il les a peut-être déjà lus. En revanche, il faut noter qu'elle fournit parfois peu d'information. Par exemple, il est difficile de positionner l'utilisateur s'il ne donne que des scores maximums pour les films que tout le monde adore aussi.

d) *Entropie* : Le système préfère les documents « informatifs » qui permettent de « séparer » les utilisateurs (certains les aiment bien et certains autres les détestent), plutôt que ceux appréciés par la plupart des gens [KM01]. L'avantage de la dernière méthode est qu'elle fournit beaucoup d'informations. Par contre, elle demande beaucoup d'effort à l'utilisateur en cas de documents étrangers.

Il est naturel que l'on combine plusieurs méthodes dans un système afin d'obtenir une bonne performance.

En résumé, l'inconvénient important de cette approche classique d'évaluation des recommandations exploratoires, est que les documents que l'utilisateur doit évaluer afin d'obtenir sa communauté initiale n'ont a priori aucun rapport privilégié avec ses centres d'intérêt potentiels. La tâche d'évaluation des recommandations exploratoires est relativement simple mais fastidieuse. Par exemple, le système de recommandation de films MovieLens [MovieLens] exige de l'utilisateur au moins 15 évaluations avant de fournir des recommandations, mais les films proposés n'étant pas ciblés pour cet utilisateur, ce dernier peut ne rencontrer que des films qu'il n'aime pas, ou des films qu'il ne sait pas évaluer, comme par exemple des films dont il n'a jamais entendu parler. Ainsi l'utilisateur devra parcourir une liste parfois très longue avant d'atteindre ce nombre de 15 évaluations. Les travaux se consacrant à cette approche ont donc pour objectif de trouver les meilleurs documents à présenter aux nouveaux utilisateurs, et cela indépendamment de l'utilisateur considéré.

6.3 Approche par hybridation

Une alternative pour le démarrage à froid est de combiner le filtrage collaboratif avec d'autres techniques de filtrage, basé sur le contenu et/ou démographique, dans un système hybride. L'idée principale de l'approche par hybridation est de proposer au nouvel utilisateur d'adopter un profil prédéfini selon le cas, afin de recevoir des recommandations exploratoires.

6.3.1 Combinaison avec le filtrage basé sur le contenu

Pour la méthode de combinaison du filtrage collaboratif avec le filtrage basé sur le contenu dans un système hybride [Bur02], la réponse la plus courante consiste à demander à l'utilisateur de définir ses centres d'intérêt, en termes de contenu, à partir d'une liste de termes et/ou d'exemples décrivant au mieux ses centres d'intérêt [CGM+99, MMN02, MR00].

Melville et ses collègues à l'université de Texas ont proposé une méthode hybride où les valeurs manquantes dans la matrice des évaluations V (voir Figure 5.1) sont d'abord remplacées par les

prédictions par le filtrage basé sur le contenu [MMN02, MR00]. Enfin, les recommandations collaboratives sont générées à partir de ces « pseudo évaluations » des utilisateurs.

Dans leur approche, Claypool et al. ont proposé la méthode de pondération des prédictions des filtrages qui se réalisent séparément [CGM+99]. Ainsi, afin de générer des recommandations pour un nouvel utilisateur, on élimine simplement la prédiction collaborative dans le calcul de la formule (13.30).

Dans les méthodes précitées, l'effort demandé au nouvel utilisateur est important : il doit procéder à une réflexion pour synthétiser ses centres d'intérêt sous la forme de termes, ou bien rechercher des exemples pertinents de documents. Cette dernière tâche peut être automatisée lorsque le système dispose de données externes sur les utilisateurs, par exemple pour des chercheurs académiques qui sont aussi auteurs de publications [MAS+02, MSR04], mais en règle générale, on ne dispose pas de telles données. Dans la plupart des cas, le profil résultant est incomplet et bruité.

6.3.2 Combinaison avec d'autres techniques

Une approche alternative pour le démarrage à froid consiste à associer le nouvel utilisateur à un profil-type (« stéréotype ») parmi ceux prédéfinis. Le processus de construction de ces profils-type nécessite un ensemble de données d'apprentissage, puis le processus de démarrage à froid doit confronter à ces profils-type certaines informations relatives au nouvel utilisateur.

Le système obtient généralement les informations nécessaires en interagissant avec l'utilisateur, par exemple en lui posant une série de questions [Kru97]. Cette méthode demande souvent des experts dans le domaine applicatif.

Par contre, on peut exploiter une source d'informations « démographiques » externe comme les pages Web personnelles des utilisateurs [Paz99]. Pazzani a étudié entre autres les performances de cette approche en la prenant pour unique base d'un système de recommandation « démographique », et nous retenons de son travail que les performances de recommandation via les données démographiques sont certes moins bonnes que celles de systèmes basés sur le contenu ou collaboratifs, mais toutefois acceptables.

De la même façon, dans son système ProfBuilder, Wasfi a proposé d'explorer implicitement l'historique des pages Web visitées pour construire les profils [Was99]. Mais, cette méthode pose le problème de la confidentialité.

6.4 Conclusion

Pour conclure, selon le type de filtrage, collaboratif ou encore hybride, les réponses au problème du démarrage à froid varient. A l'exception des contextes applicatifs où des sources externes d'informations sur les nouveaux utilisateurs existent, l'utilisateur est toujours mis à contribution. Cela veut dire que les processus de démarrage à froid ont tous un coût important pour l'utilisateur. D'une

part, ils requièrent, selon les modalités de la contribution, un effort en termes de temps ou d'accomplissement d'une tâche plus ou moins difficile, longue et fastidieuse. D'autre part, le profil et les communautés résultant de ce processus ne peuvent pas donner lieu à des recommandations de qualité. Ils conduisent donc à un rapport coût-bénéfice déficitaire pour les utilisateurs.

Chapitre 7

Bilan

Il ressort des travaux existants que les communautés dans les systèmes ne sont pas exploitées de façon efficace alors même qu'il agit d'un facteur clé des systèmes de filtrage collaboratif.

Pratiquement, les communautés restent toujours un facteur interne du système pour le calcul et/ou l'explication des recommandations. Certains systèmes offrent aux utilisateurs la possibilité de percevoir partiellement leurs communautés. Mais, en réalité, les utilisateurs pourraient en faire un usage beaucoup plus important afin d'exploiter au mieux le système. Les communautés devraient être une modalité d'interaction entre utilisateurs et système.

Par ailleurs, le fait de ne s'en remettre qu'au critère de l'historique des évaluations pour former des communautés limite inutilement la production des recommandations. Du côté des utilisateurs, le rapport coût-bénéfice leur paraît apparemment déficitaire dès le premier contact avec le système, car ils ne reçoivent pas de recommandations pertinentes après avoir fourni beaucoup d'efforts pour évaluer les recommandations exploratoires.

Partant des limites des études actuelles, nous proposons dans cette thèse le modèle des espaces de communautés afin d'améliorer la performance de la gestion des communautés dans un système de filtrage collaboratif.

Partie III.

Gestion des communautés dans un système de filtrage collaboratif

Cette partie présente notre proposition d'un modèle pour la gestion des communautés multicritères et explicites dans un système de filtrage collaboratif. L'intégration de notre modèle des espaces de communautés dans un tel système permet d'améliorer l'exploitation des communautés formées à partir des critères disponibles dans les profils d'utilisateurs.

D'abord, grâce à la multiplicité de critères de formation des communautés, notre modèle favorise une diversification de recommandations provenant des communautés concernées. De plus, ce modèle permet d'enrichir les fonctionnalités du système et de rendre plus performant le rôle des communautés et des utilisateurs. En effet, les communautés peuvent être exploitées non seulement dans la phase de génération de recommandations mais aussi dans d'autres activités importantes du système telles que la perception et l'exploration des communautés, l'explication des recommandations, l'analyse des relations entre communautés et à terme l'adaptation des profils.

Cette partie commence par la présentation de notre modèle des espaces de communautés (Chapitre 8). L'objectif principal de ce modèle est de représenter les espaces de communautés et d'exploiter les relations entre ces espaces pour diverses activités du filtrage collaboratif, incluant l'induction des communautés.

Notre modèle est conçu en partant de l'hypothèse que les espaces de communautés sont déjà créés par un processus général de formation des communautés (Chapitre 9).

Chapitre 8

Modèle des espaces de communautés

Nous constatons que dans la plupart des systèmes de filtrage collaboratif existants, les communautés sont monocritères, et qu'il n'existe qu'un seul ensemble, ou espace, de communautés créé généralement par la proximité des évaluations explicites ou implicites des utilisateurs. Pourtant, on trouve une multiplicité de critères sur lesquels appuyer la construction des communautés, tels que : informations personnelles, centres d'intérêt, préférences de livraison et de sécurité, etc. [AS99, BK05]. Il s'agit de l'existence de plusieurs espaces de communautés à la fois, et nous avons donc besoin d'un modèle pour les gérer de façon intelligente.

8.1 Introduction

Dans ce chapitre, nous présentons notre *modèle des espaces de communautés*, qui a pour objectif de représenter les espaces de communautés et d'induire des communautés en exploitant les relations entre ces espaces.

D'abord, comme le montre la partie B1 de la Figure 8.1, nous utilisons une table de communautés pour représenter les espaces de communautés dans le système de filtrage collaboratif (section 8.2). Chaque espace de communautés est une représentation des utilisateurs en communautés selon un critère particulier. Les vecteurs de positionnement sont également définis pour représenter le polymorphisme du positionnement des utilisateurs au sein des communautés multicritères.

Ensuite, nous abordons l'induction de communautés par règles (partie B2 de la Figure 8.1), qui permet de déterminer la communauté d'un utilisateur dans un espace particulier à partir de ses

communautés dans d'autres espaces. Cette méthode d'induction basée sur la théorie des ensembles d'approximation proposée par Pawlak (section 8.2.3) fournit les moyens d'analyser la relation entre critères ou espaces (section 8.4). Cela rend plus performante la formation des communautés multicritères dans les cas où le système souffre de la complexité du calcul requis pour créer les communautés, ou encore dans les cas où les informations nécessaires au positionnement dans l'espace en question sont indisponibles ou bruitées.

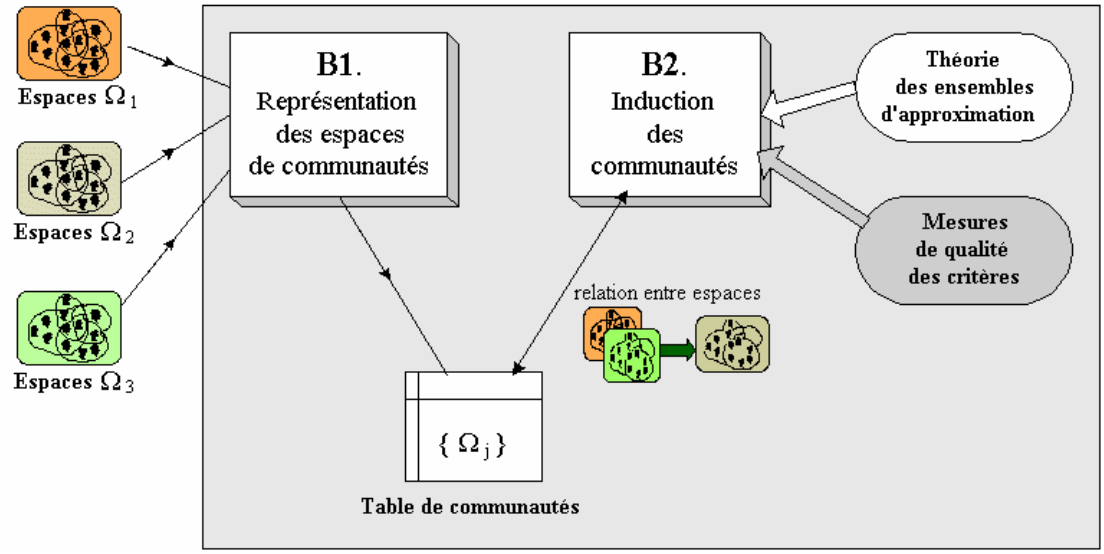


Figure 8.1 – *Modèle des espaces de communautés*.

Enfin, nous complétons notre méthode d'induction de communautés par les mesures de comparaison entre critères en vue de préserver la qualité de l'induction (section 8.5). Lorsque plusieurs communautés multicritères d'un utilisateur sont manquantes ou douteuses, ces mesures aident le système à déterminer, parmi les critères problématiques, ceux qui peuvent être traités pour induire les communautés, ainsi qu'à établir un ordre de traitement des critères ainsi déterminés.

8.2 Modélisation de communautés multicritères

Sous l'angle fonctionnel de la production des recommandations aux utilisateurs dans un système de filtrage collaboratif, une communauté dans notre approche est définie comme un ensemble d'utilisateurs qui sont proches les uns des autres selon un critère donné de comparaison. La dimension purement sociale n'est pas prise en considération, comme c'est le cas dans le domaine des sciences humaines et sociales. Pourtant, notre approche n'exclut pas de telles communautés. En général, les gens dans une même communauté ne doivent forcément pas se connaître, sauf les « vraies » communautés comme dans le filtrage collaboratif actif.

En utilisant chacun des critères disponibles dans les profils, par exemple informations personnelles, centres d'intérêts, historique des évaluations, etc., comme critère de formation, le

système peut créer autant d'espaces de communautés, et autoriser ainsi un utilisateur à appartenir à autant de communautés qu'il y a de critères pour les former.

Grâce à ce polymorphisme de positionnement, la production des recommandations sera enrichie sachant que l'utilisateur peut recevoir des recommandations par le biais de chacune des communautés auxquelles il appartient. On voit dans la réalité qu'une personne reçoit souvent toutes sortes de recommandations intéressantes de ses proches, de ses collègues de travail, des membres de son club de loisirs, etc. Alors, cette approche de communautés multicritères peut conduire à des recommandations plus diversifiées qui peuvent être exploitées de façon sélective dans une stratégie adaptée à la situation des utilisateurs.

Cette section présente les notions principales pour la modélisation de ces communautés multicritères dans un système de filtrage collaboratif : espace de communautés, vecteur de positionnement et table de communautés.

8.2.1 Espace de communautés

Nous définissons d'abord deux ensembles non vides : l'ensemble U des *utilisateurs* ($|U| = m$), et l'ensemble A des *critères* de formation des communautés disponibles dans le système ($|A| = n$).

Pour tout critère $a \in A$, nous définissons, conformément à la théorie des ensembles d'approximation sur laquelle s'appuie notre modèle, la *relation d'indiscernabilité* $\mathcal{R}_a$ qui est une relation d'équivalence sur U, où les utilisateurs sont en relation s'ils ont les mêmes valeurs, c'est-à-dire qu'ils considèrent la même chose, pour le critère a :

$$\mathcal{R}_a \subseteq U \times U : \qquad \forall u, u' \in U, \quad u \,\mathcal{R}_a\, u' \;\Leftrightarrow\; a(u) = a(u') \tag{8.1}$$

Exemple : $u \,\mathcal{R}_{\text{Ville}}\, u' \Leftrightarrow Ville(u) = Ville(u')$: u et u' habitent la même ville.

Les *communautés* relatives au critère a, notées Ga_k avec $k \in [1, n]$, sont les classes d'équivalence de U suivant $\mathcal{R}_a$. La partition de l'ensemble des utilisateurs U par le critère a, notée Ω_a, est appelée *espace de communautés*, et correspond à l'ensemble quotient de U par $\mathcal{R}_a$:

$$\Omega_a = U/\mathcal{R}_a = \{Ga_1, \ldots, Ga_r\} \tag{8.2}$$

Nous rappelons qu'une relation d'équivalence $\mathcal{R}$ sur un ensemble X vérifie les trois propriétés suivantes :

i) réflexivité : $x \,\mathcal{R}\, x, \qquad \forall x \in X$

ii) symétrie : $x \,\mathcal{R}\, y \Rightarrow y \,\mathcal{R}\, x, \qquad \forall x, y \in X$

iii) transitivité : $(x \,\mathcal{R}\, y) \wedge (y \,\mathcal{R}\, z) \Rightarrow (x \,\mathcal{R}\, z) \qquad \forall x, y, z \in X$

La deuxième propriété de symétrie implique un rôle identique pour tous les éléments dans une classe d'équivalence, c'est-à-dire que dans une communauté, il n'y a pas de rôle particulier tel que représentant ou modérateur.

Nous définissons la notion de *critère composé* comme un ensemble d'au moins deux critères $P \subseteq A$. L'*espace de communautés composé* Ω_P est simplement l'ensemble quotient $U/\mathcal{R}_P$ par la relation d'équivalence $\mathcal{R}_P$, et Ω_P est donc constitué des groupes d'utilisateurs ayant les mêmes valeurs pour tous les critères de P.

$$\forall u, u' \in U, \qquad u \,\mathcal{R}_P\, u' \;\Leftrightarrow\; (\forall a \in P,\ a(u) = a(u')) \tag{8.3}$$

Si l'on définit par exemple le critère composé P = {Ville, Genre}, l'espace composé Ω_P comporte six communautés suivantes (voir Tableau 8.1) :

$G_1 = [u_1]_P = \{u_1, u_2\}$, (Ville = « Paris », Genre = « Aventure »)

$G_2 = [u_3]_P = \{u_3\}$, (Ville = « Paris », Genre = « Documentaire »)

$G_3 = [u_4]_P = \{u_4, u_5, u_6\}$, (Ville = « Paris », Genre = « Scientifique »)

$G_4 = [u_7]_P = \{u_7\}$, (Ville = « Paris », Genre = « Fiction »)

$G_5 = [u_8]_P = \{u_8, u_9\}$, (Ville = « New York », Genre = « Documentaire »)

$G_6 = [u_{10}]_P = \{u_{10}, u_{11}, u_{12}\}$, (Ville = « Londres », Genre = « Documentaire »)

Critère / Utilisateur	Profession	**Ville**	**Genre préféré**	Evaluation	
u_1	Commerçant	*Paris*	*Aventure*	Groupe 1	$\boldsymbol{G_1}$
u_2	Chercheur	*Paris*	*Aventure*	Groupe 4	
u_3	Chercheur	*Paris*	*Documentaire*	Groupe 2	$\boldsymbol{G_2}$
u_4	Chercheur	*Paris*	*Scientifique*	Groupe 1	$\boldsymbol{G_3}$
u_5	Chercheur	*Paris*	*Scientifique*	Groupe 4	
u_6	Chercheur	*Paris*	*Scientifique*	Groupe 3	
u_7	Chercheur	*Paris*	*Fiction*	Groupe 5	$\boldsymbol{G_4}$
u_8	Chercheur	*New York*	*Documentaire*	Groupe 5	$\boldsymbol{G_5}$
u_9	Commerçant	*New York*	*Documentaire*	Groupe 5	
u_{10}	Commerçant	*Londres*	*Documentaire*	Groupe 3	$\boldsymbol{G_6}$
u_{11}	Commerçant	*Londres*	*Documentaire*	Groupe 2	
u_{12}	Commerçant	*Londres*	*Documentaire*	Groupe 3	

Tableau 8.1 – *Exemple de critère composé P = {Ville, Genre}.*

Ici, le terme « critère composé » veut simplement dire la combinaison conjonctive des espaces de communautés « simples » Ωa_j. Une approche générale de formation des communautés par un critère composé, par exemple {Genre, Evaluation}, peut être beaucoup plus compliquée mais elle n'est pas l'objectif principal de la thèse. Dans le reste du manuscrit, les termes communauté, critère et espace de communautés désignent, sauf précision autre, les notions simples.

8.2.2 Vecteur de positionnement

Du côté des utilisateurs, chaque personne est rattachée à une communauté dans chacun des espaces Ω_a, $a \in A$. Nous appelons *vecteur de positionnement* de l'utilisateur u, noté $\mathcal{P}_u$, la liste des étiquettes de ses propres communautés selon chacun des critères.

Formellement, nous avons besoin d'une fonction de positionnement $\mathcal{P}$ suivante :

$$\mathcal{P} : U \longrightarrow Va_1 \text{ x } \ldots \text{ x } Va_n$$

$$\forall u \in U : \quad \mathcal{P}(u) = (pu_1, \ldots, pu_n)$$

où Va_j est l'ensemble des étiquettes de communauté par le critère a_j.

Etant donnés un critère $a_j \in A$ et l'espace de communautés correspondant : $\Omega a_j = \{Ga_{j1}, \ldots, Ga_{jr}\}$. L'utilisateur u appartient à une communauté particulière :

$$\exists k \in \{1, \ldots, r\} : \quad u \in Ga_{jk}$$

En outre, il existe pour chaque critère a_j une correspondance entre l'ensemble Va_j et l'espace de communautés Ωa_j, car à chaque valeur va_{jk} dans Va_j est associée une et une seule communauté Ga_{jk}. On peut définir sans perte de généralité la bijection $\mathcal{F}a_j$ entre Ωa_j et Va_j comme suit.

$$\mathcal{F}a_j : \Omega a_j \longrightarrow Va_j$$

$$\forall k = 1,\ldots, r : \quad \mathcal{F}a_j(Ga_{jk}) = va_{jk} \tag{8.4}$$

Alors, on obtient à partir de la formule (8.4) :

$$pu_j = \mathcal{F}a_j(Ga_{jk}) = va_{jk} \qquad \square$$

Il est à remarquer que le profil et le vecteur de positionnement d'un utilisateur dans notre modèle sont en général différents. Le vecteur de positionnement est une structure qui ne fait absolument pas d'hypothèse sur la représentation du profil, mais qui s'y ajoute en exploitant les

informations de profil existantes. Techniquement, le profil alimente la construction du vecteur de positionnement, mais nous montrons plus loin que le vecteur de positionnement peut être par ailleurs adapté par un processus d'induction des communautés par règles, ce qui cause la différence soit temporaire, soit permanente entre le vecteur de positionnement et le profil.

8.2.3 Table de communautés

Dans notre modèle, une *table de communautés* T_{mxn} représentant les espaces de communautés est caractérisée par une paire de deux ensembles non vides :

$$T_{mxn} = < U , A >$$

où U est l'ensemble des utilisateurs, et A contient les critères de formation des communautés (voir Tableau 8.2).

	Critère / Utilisateur	a_1	…	a_j	…	a_n
	u_1					
	…					
$\boldsymbol{T[u_i]}$	u_i			$\boldsymbol{T[u_i, a_j]}$		
	…					
	u_m					
				$\boldsymbol{Ta_j}$		

Tableau 8.2 – *Table de communautés*.

Pour tout critère $a_j \in A$, nous notons Va_j l'ensemble des valeurs présentes dans la colonne Ta_j, qui sont les étiquettes des communautés formées à partir du critère a_j. Alors, on a : $T[u_i, a_j] = a_j(u_i)$

Le Tableau 8.1 est un exemple de table de communautés d'un système de recommandation de films.

En général, on peut lire la table de communautés selon une vue opérationnelle ou une vue structurelle. La vue opérationnelle est orientée selon les lignes : elle reflète les relations évolutives entre les utilisateurs (lignes de la table) que le système peut exploiter pour générer les recommandations étant donné un critère. La vision structurelle est orientée selon les colonnes : elle traduit la dépendance relativement durable entre les critères (colonnes de la table), dépendances plus durables que les relations entre utilisateurs.

On peut noter que le vecteur de positionnement $\mathcal{P}_u$ de l'utilisateur u correspond à la ligne $T[u]$ dans la table de communautés T (voir Tableau 8.2). On peut ainsi voir la table T comme l'ensemble des vecteurs de positionnement des utilisateurs dans les espaces de communautés.

Pour construire cette table de communautés, le système peut s'en remettre à des méthodes de regroupement d'utilisateurs selon tous les critères de l'ensemble *A*, et la table de communautés *T* est remplie, colonne par colonne, par les étiquettes des communautés résultant de ce regroupement. Nous montrons dans le Chapitre 9 sa construction. La tâche de formation des communautés multicritères alimentant la table des communautés sera également discutée dans le Chapitre 11 de mise en œuvre du modèle proposé.

8.3 Motivation du choix de la formalisation par la théorie des ensembles d'approximation

Avant de décrire dans 8.4 et 8.5 notre approche basée sur la théorie des ensembles d'approximation pour induire des communautés imparfaites dans un vecteur de positionnement, nous expliquons par la suite les raisons de notre choix de cette théorie mathématique.

8.3.1 Besoin d'induction des communautés pour le positionnement des utilisateurs

En général, le vecteur de positionnement d'un utilisateur contient souvent des communautés manquantes, périmées ou douteuses que le système essaie de corriger, d'une façon ou d'une autre, afin de le positionner au mieux dans les espaces de communautés. Ce phénomène des communautés imparfaites reflète les difficultés dans le positionnement des utilisateurs au sein des espaces de communautés.

La cause des difficultés dans le positionnement est diverse. D'abord, c'est la complexité du calcul dans le contexte de la multiplicité de critères. En effet, pour créer les espaces de communautés, pour certains critères, la classification des utilisateurs se réalise facilement par la comparaison directe des valeurs. Par exemple, le système peut diviser les utilisateurs en deux classes selon le sexe masculin ou féminin. Parfois, le système doit appliquer une combinaison des valeurs pour regrouper les utilisateurs, par exemple pour former des communautés par tranches d'âge ou par catégories de profession. La tâche de formation des communautés pour des critères comme les centres d'intérêt ou l'historique des évaluations est plus difficile, car elle requiert les étapes suivantes : choisir la mesure de similarité entre utilisateurs appropriée, faire appel à une méthode de classification, calibrer plusieurs paramètres, etc.

De plus, les difficultés du positionnement des utilisateurs dans les espaces concernés proviennent éventuellement de l'insuffisance des données dans les profils. A titre d'exemple, le système de filtrage collaboratif doit souvent faire face au problème du démarrage à froid rencontré pour le positionnement d'un nouvel utilisateur dans l'espace du critère Evaluation : en effet, cet utilisateur n'a encore évalué aucune recommandation. Ce problème s'aggrave encore dans le contexte des communautés multicritères, où le nouvel utilisateur peine à définir ses propres centres d'intérêt pour que le système puisse lui proposer une communauté correcte dans l'espace du critère concerné.

Une autre situation difficile éventuelle est qu'au cours de l'exploitation, la communauté de l'utilisateur dans l'espace du critère Evaluation est susceptible d'évoluer. Mais, la seule façon pour lui

d'exprimer cette évolution est d'évaluer les recommandations reçues, alors même qu'il ne reçoit pas du système les recommandations lui permettant d'exprimer ce changement. Supposons que le système détecte une accumulation de retours de pertinence négatifs susceptibles de décourager l'utilisateur. Ce dernier est en situation difficile de changement de communauté mais le système ne dispose d'aucune autre ressource que les évaluations pour repositionner cet utilisateur. L'utilisateur se trouvant dans l'impossibilité d'adapter sa communauté, finira par abandonner le système, et cela quelle que soit la qualité du moteur de filtrage [Gal05, GBD03].

La performance de formation des communautés peut être améliorée de façon significative par une méthode permettant d'induire pour un utilisateur la communauté dans un espace problématique à partir de ses communautés fiables dans d'autres espaces. Par exemple, si l'utilisateur ne satisfait plus les recommandations issues de sa communauté du critère Evaluation qui est déjà périmée, nous cherchons à exploiter d'autres ressources que ses évaluations pour lui proposer une nouvelle communauté appropriée. De la même façon, nous tentons de rattacher un nouvel utilisateur à une communauté dans l'espace de communautés relatif au critère Evaluation sans même demander d'évaluations à l'utilisateur.

8.3.2 Choix du formalisme

Pour améliorer le positionnement des utilisateurs, nous proposons d'utiliser l'approche d'induction des communautés par règles, basée sur la théorie des ensembles d'approximation proposée par Z. Pawlak au début des années 80 [Paw82]. L'idée principale de cette théorie est de diviser d'abord les attributs des données, ou pour nous les critères de formation des communautés, en deux parties : un attribut de décision prédéfini ($D = \{d\}$) selon le domaine d'application et les attributs de condition (C) qui restent. Cette théorie offre des moyens efficaces pour analyser la dépendance de l'attribut de décision par rapport aux attributs de condition en utilisant la définition approximative des concepts dans l'ensemble quotient $U/\mathcal{R}_D$ par les classes d'équivalence de la relation $\mathcal{R}_C$.

La théorie des ensembles d'approximation est choisie comme base formelle de notre approche en raison de la nature des données à traiter, mais aussi parce qu'elle se base sur les relations d'équivalence, parce qu'elle offre des possibilités d'explication sur les communautés, et peut aussi se combiner avec d'autres approches.

Nature des données. Nous rappelons que la problématique dans l'induction de communautés requiert le traitement de données symboliques que sont les étiquettes de la table de communautés (voir Tableau 8.1), et la prise en compte et l'analyse des imperfections de ces données dans les vecteurs de positionnement. Pour répondre à ces problèmes, on peut s'en remettre aux techniques de fouille de données. Ces techniques sont développées pour découvrir les connaissances potentiellement intéressantes dans les gigantesques bases de données existantes [CHM+01]. Néanmoins, d'après Mazlack et Coppock [CM03, Maz99], ces techniques ne sont efficaces que pour traiter des données quantitatives puisqu'elles font l'hypothèse que les données sont quantitatives et peuvent être mesurées par des métriques précises. Pour traiter des données nominales (symboliques) comme celles des tables de communautés, cela nous conduirait à l'obligation de quantifier les données, ce qui n'est pas souhaitable.

En revanche, la théorie de l'évidence (*Dempster-Shafer Theory of Evidence*) et la théorie des ensembles floues (*Fuzzy Sets*) sont deux approches classiques pour traiter les données incertaines nominales sans avoir besoin de les quantifier, mais elles requièrent certaines informations a priori sur ces données.

La théorie de l'évidence nécessite des informations a priori sur les données afin de définir les fonctions de croyance [Sha76]. En effet, soit un espace ou ensemble fini des hypothèses possibles Ω. On définit une fonction de masse m sur les sous-ensembles de Ω, $P(\Omega)$, telle que :

$$m : P(\Omega) \rightarrow [0,1]$$
$$m(\varnothing) = 0 \text{ et } \sum_{X \in P(\Omega)} m(X) = 1$$

La valeur $m(X)$ exprime le degré de croyance sur l'hypothèse X. Alors, la fonction de croyance est définie par :

$$Bel : P(\Omega) \rightarrow [0,1]$$
$$Bel(X) = \sum_{Y \subseteq X} m(Y)$$

Dans la théorie des ensembles flous proposée par Zadeh [Zad65], le degré d'appartenance d'un élément à un ensemble particulier est une valeur dans l'intervalle [0,1] plutôt qu'une valeur binaire comme dans la théorie des ensembles classique. Pourtant elle requiert aussi des informations a priori, comme la théorie précédente, pour définir la fonction d'appartenance μ. Soient deux ensembles X et Y tels que : $Y \subseteq X$. La fonction d'appartenance μ_Y est définie comme :

$$\mu_Y : P(\Omega) \rightarrow [0,1]$$
$$\mu_Y(x) = \text{degré d'appartenance de l'élément } x \text{ à l'ensemble } Y$$

A l'opposé de ces deux approches subjectives, la théorie des ensembles d'approximation est un outil mathématique efficace pour traiter des données autant qualitatives que quantitatives, et ne requiert a priori *aucune information* ou *paramètre additionnels* [CM03, JS03, Maz99, Paw82, Paw04]. L'idée principale de cette théorie est que l'incertitude des données peut être mesurée par des bornes inférieure et supérieure. Elle permet d'identifier et d'exploiter les dépendances entre les attributs des données. Si nous considérons que les divers critères de formation des communautés sont les attributs des utilisateurs, nous pouvons envisager, de compenser l'absence de certaines valeurs d'attributs en exploitant au mieux les attributs où les valeurs sont disponibles.

Relations d'équivalence. La théorie des ensembles d'approximation se fonde sur les relations d'équivalence dans le but d'identifier et d'exploiter la dépendance entre les attributs des données. Dans notre approche, nous adoptons aussi les relations d'équivalence pour formaliser les

communautés multicritères dans un système de filtrage collaboratif. Cette compatibilité du modèle formel facilite notre exploitation des moyens efficaces fournis par la théorie des ensembles d'approximation pour analyser la relation entre les classes d'équivalence représentant les communautés multicritères. Dans le cadre de cette théorie, les objets étudiés sont les utilisateurs, leurs attributs sont les divers critères de formation des espaces de communautés, et les données sont les étiquettes des communautés auxquelles ils appartiennent. Nous souhaitons compenser l'absence ou l'incertitude de certaines valeurs d'attributs dans le positionnement d'utilisateurs en exploitant au mieux les attributs où les valeurs sont disponibles. Par exemple connaissant Ville, Profession et Genre, nous aimerions pouvoir proposer une communauté dans l'espace relatif au critère classique Evaluation sans même demander d'évaluations à l'utilisateur.

Possibilité d'explication sur les communautés. Les études expérimentales de Herlocker et al. [Her00, HKR00] ont montré que l'explication des recommandations envoyées par le système améliore significativement la confiance des utilisateurs sur ces recommandations, et en particulier les motive pour fournir des évaluations, ce sans quoi le système ne peut pas former de meilleures communautés. En réalité, l'utilisateur a besoin des explications compréhensibles et convaincantes afin de prendre la décision, par exemple d'acheter un produit ou d'aller voir un film au cinéma. Les informations relatives à sa communauté lui permettent de découvrir ce qui se cache dans les phases de production des recommandations dans un système de filtrage collaboratif. Les auteurs ci-dessus ont également montré que l'histogramme des évaluations de leur communauté est particulièrement apprécié par les utilisateurs.

Dans ce sens, nous pensons qu'une explication telle que « la plupart des chercheurs parisiens aiment bien les films scientifiques », en se basant sur la règle d'induction : (Profession = « Chercheur », Ville = « Paris ») → (Genre = « Scientifique »), pour un professeur à l'université à Paris est plus ou moins convaincante même si son genre de film préféré actuel n'est pas le même.

Possibilité de combinaison avec d'autres approches. Pour répondre au problème de traiter des données nominales incertaines, on a plusieurs alternatives d'approches. A titre d'exemple, on peut citer la théorie de l'évidence, la théorie des ensembles flous, etc. Il existe actuellement dans la littérature des études permettant la combinaison de la théorie des ensembles d'approximation avec ces approches [PS94, Yao98a, Yao98b]. Ainsi, la théorie des ensembles d'approximation est préférée en envisageant la possibilité d'intégrer plusieurs approches dans le futur afin d'améliorer la performance du modèle proposé.

Nous présentons dans la suite les principes basés sur la théorie des ensembles d'approximation pour l'induction de communautés par règles, discutons également ses limites, et abordons par conséquent la nécessité d'une extension de cette théorie en vue de l'adapter au contexte des communautés multicritères.

8.4 Induction de communautés

Le premier objectif de notre approche d'induction des communautés, outre celui de la comparaison de critères présenté dans 8.5, est d'exploiter certaines notions de la théorie des ensembles

d'approximation, concernant notamment la classification par règles, pour la tâche de positionnement des utilisateurs dans les espaces de communautés. Les notions nécessaires à la classification par règles sont définies dans la section 8.4.1. Ensuite, nous montrons dans 8.4.2 une brève illustration d'utilisation de ces notions pour induire une communauté inconnue dans le vecteur de positionnement d'un utilisateur. Enfin, les moyens d'analyse de dépendance entre attributs, reflétant la performance de la classification par règles, sont aussi présentés dans la section 8.4.3.

8.4.1 Règles de décision

L'ensemble des attributs ou critères de formation des communautés A est divisé en deux : $A = C \cup D$, où D contient un seul attribut d dit *décision* et C contient les attributs restants dits *condition*. La table de communautés T est dite *table de décision* (voir exemple dans la Figure 8.2), et chaque ligne de la table, correspondant à un utilisateur particulier u, est considérée comme une *règle de décision*, ou « règle » tout court.

Par exemple, on a dans la Figure 8.2, où Evaluation est pris comme attribut de décision, la règle u_7 :

(Profession = « Chercheur », Ville = « Paris », Genre = « Fiction ») → (Evaluation = « Groupe 5 »)

qui dit : « Un chercheur parisien qui aime les films Fiction appartient à la communauté libellée Groupe 5 dans l'espace Evaluation ».

U \ A	Profession	Ville	Genre	D = {Evaluation}	
u_1	Commerçant	Paris	Aventure	Groupe 1	
u_2	Chercheur	Paris	Aventure	Groupe 4	
u_3	Chercheur	Paris	Documentaire	Groupe 2	
u_4	Chercheur	Paris	Scientifique	Groupe 1	
u_5	Chercheur	Paris	Scientifique	Groupe 4	
u_6	Chercheur	Paris	Scientifique	Groupe 3	
u_7	Chercheur	Paris	Fiction	Groupe 5	← X_5
u_8	Chercheur	New York	Documentaire	Groupe 5	
u_9	Commerçant	New York	Documentaire	Groupe 5	
u_{10}	Commerçant	Londres	Documentaire	Groupe 3	
u_{11}	Commerçant	Londres	Documentaire	Groupe 2	
u_{12}	Commerçant	Londres	Documentaire	Groupe 3	

Figure 8.2 – *Exemple de table de décision avec D = {Evaluation} et P = {Ville, Genre}.*

Dans une règle, la prémisse est constituée des *conditions*, par exemple (Ville = « Paris ») et Genre = « Fiction », et la conclusion telle que (Evaluation = « Groupe 5 ») est aussi qualifiée de *décision*.

Dans le reste du manuscrit, nous simplifions parfois les termes *attribut de condition* et *attribut de décision* en *condition* et *décision* respectivement, si cette simplification ne donne lieu à aucune paradoxe. En assimilant $\{d\}$ à d, une règle peut donc être notée comme :

$$C \to D \quad \text{ou} \quad C \to d$$

Nous remarquons ici que dans notre modèle, les critères de l'ensemble A jouent des rôles symétriques, c'est-à-dire qu'il n'y a aucune préférence a priori parmi ces critères. Pourtant, la signification des critères peut être prise en compte selon la situation rencontrée au cours de l'exploitation. Par exemple, un système de recommandation peut créer les communautés via le critère de proximité des évaluations fournies dans le passé par les utilisateurs, afin de leur produire de nouvelles suggestions alors que les premières recommandations pour un nouvel utilisateur peuvent être générées à partir de ses communautés démographiques puisque sa communauté relative au critère Evaluation n'est pas encore connue. La discussion sur l'importance des critères en cours d'exploitation se trouve plus tard dans la section 8.5 du présent chapitre.

Dans la théorie des ensembles d'approximation, la prémisse des règles est souvent réduite à un sous-ensemble d'attributs de condition $P \subseteq C$: $P \to D$

Soit la relation d'indiscernabilité $\mathcal{R}_D$. Une classe d'équivalence $[u]_D \in U/\mathcal{R}_D$ est appelée *concept*. Dans l'exemple précédent où D = {Evaluation}, on a cinq concepts (voir Figure 8.2) :

$X_1 = \{u_1, u_4\}$ (Evaluation = « Groupe 1 »)

$X_2 = \{u_3, u_{11}\}$ (Evaluation = « Groupe 2 »)

$X_3 = \{u_6, u_{10}, u_{12}\}$ (Evaluation = « Groupe 3 »)

$X_4 = \{u_2, u_5\}$ (Evaluation = « Groupe 4 »)

$X_5 = \{u_7, u_8, u_9\}$ (Evaluation = « Groupe 5 »)

Supposons que P = {Ville, Genre}. Les règles u_1 et u_2 sont contradictoires, car elles ont la même prémisse : (Ville = « Paris ») et (Genre = « Aventure ») alors qu'elles donnent deux conclusions différentes : « Groupe 1 » et « Groupe 4 ».

Dans ce cas, on dit que la règle $P \to D$ n'est pas une règle certaine. En effet, on définit une *règle certaine* $u \in U$ comme une règle dont la classe d'équivalence par $\mathcal{R}_P$ est incluse dans celle par $\mathcal{R}_D$:

$$[u]_P \subseteq [u]_D \tag{8.5}$$

Remarque :

Si $[u]_P$ est un singleton, alors u est une règle certaine, car $[u]_P$ est absolument incluse dans une certaine classe d'équivalence par $\mathcal{R}_D$.

$$(|[u]_P| = 1) \Rightarrow (u \in POS_P(D)) \tag{8.6}$$

Lorsqu'une règle est certaine, dans toute la table *T*, si les prémisses sont les mêmes que celles de la règle en question, les conclusions sont aussi les mêmes. Dans l'exemple ci-dessus, u_3, u_7, u_8 et u_9 sont des règles certaines puisque :

$[u_3]_P \subseteq [u_3]_D = X_2$

$[u_7]_P \subseteq [u_7]_D = X_5$

$[u_8]_P = [u_9]_P \subseteq [u_8]_D = [u_9]_D = X_5$

Dans notre approche, on peut utiliser l'ensemble des règles certaines pour induire, pour un utilisateur, la communauté relative au critère choisi comme attribut de décision *d,* en se basant sur les données des attributs de condition.

La *région positive* de $U/\mathcal{R}_P$, notée $POS_P(D)$, est l'union de toutes les règles certaines de la table *T*. Elle contient donc les règles donnant toujours la même valeur pour l'attribut de décision étant données les valeurs pour les attributs de décision.

Dans la Figure 8.3, nous retrouvons les règles certaines u_3, u_7, u_8 et u_9 dans la région positive $POS_P(Evaluation)$.

U	Profession	Ville	Genre	D = {Evaluation}	POS_P(Evaluation)
u_1	Commerçant	Paris	Aventure	Groupe 1	
u_2	Chercheur	Paris	Aventure	Groupe 4	
u_3	Chercheur	Paris	Documentaire	Groupe 2	x
u_4	Chercheur	Paris	Scientifique	Groupe 1	
u_5	Chercheur	Paris	Scientifique	Groupe 4	
u_6	Chercheur	Paris	Scientifique	Groupe 3	
u_7	Chercheur	Paris	Fiction	Groupe 5	x
u_8	Chercheur	New York	Documentaire	Groupe 5	x
u_9	Commerçant	New York	Documentaire	Groupe 5	x
u_{10}	Commerçant	Londres	Documentaire	Groupe 3	
u_{11}	Commerçant	Londres	Documentaire	Groupe 2	
u_{12}	Commerçant	Londres	Documentaire	Groupe 3	

Figure 8.3 – *Exemple de région positive avec D = {Evaluation} et P = {Ville, Genre}.*

8.4.2 Illustration d'un processus de correction des vecteurs de positionnement par règles

Considérons maintenant un vecteur de positionnement où il ne manque qu'une valeur de communauté. En utilisant la théorie des ensembles d'approximation, on prend comme attribut de décision *d,* avec $D = \{d\}$, le critère correspondant à la valeur manquante, et on cherche une règle certaine applicable dans la région positive avec $P = C$, où $C = A \setminus D$, afin d'inférer la valeur pour *d*.

Supposons que le nouvel utilisateur est un chercheur à Paris, qui aime les films documentaires et que sa communauté dans l'espace Evaluation n'est pas encore connue car il n'a encore évalué aucun film. La Figure 8.3 montre que l'on a la règle certaine u_3 suivante :

(Profession = « Chercheur», Ville = « Paris », Genre = « Documentaire ») → (Evaluation = « Groupe 2 »)

Le système peut donc rattacher cet utilisateur à la communauté libellée « Groupe 2 » dans l'espace Evaluation.

En pratique, les règles applicables pour compléter la valeur pour $d \in D$ ne sont pas toujours existantes, et on doit alors faire appel à certaines techniques de remplissage de valeurs manquantes [GH00, Grz97, KPS98].

8.4.3 Dépendance d'un attribut de décision, consistance d'une table et signification des attributs de condition

La *dépendance* de l'attribut de décision d telle que $D = \{d\}$ par rapport à l'ensemble $P \subseteq C$ peut être mesurée par la formule :

$$\gamma(P, D) = \frac{\left|POS_P(D)\right|}{\left|U\right|} \tag{8.7}$$

Ce coefficient traduit la part des utilisateurs donnant lieu à des règles certaines. Si $\gamma(P, D)$ est égal à 1, on obtient une dépendance totale de D par rapport aux critères dans l'ensemble P, c'est-à-dire que tous les utilisateurs peuvent être regroupés dans les communautés du critère D en utilisant les critères de P. Sinon, on a une dépendance partielle de D par rapport à P.

Pour l'ensemble des attributs de condition $C = A \setminus D$, $\gamma(C, D)$ exprime la *consistance* ou la qualité d'induction de la table T vis-à-vis de l'attribut de décision d par les attributs de condition de C.

Par souci de performance de l'induction par règles, la théorie des ensembles d'approximation cherche à réduire la taille (nombre d'attributs) des prémisses dans les règles. Alors, on définit la *réduction* de C comme le sous-ensemble P de C comprenant l'ensemble minimal d'attributs de condition permettant de préserver la région positive :

$$POS_P(D) = POS_C(D)$$

Une même table de décision peut donner lieu à plusieurs réductions, et le calcul des réductions est un problème NP-difficile [SR92]. De ce fait, il existe dans la littérature des méthodes heuristiques de calcul des réductions afin de rendre réaliste la théorie des ensembles d'approximation [NN96].

De plus, on peut mesurer l'importance d'un sous-ensemble d'attributs de condition $P \subset C$ sur l'intervalle [0,1] par sa *signification*, qui mesure l'impact de sa suppression sur la consistance de la table T :

$$\sigma_{C,D}(P) = 1 - \frac{\gamma(C \setminus P, D)}{\gamma(C, D)} \tag{8.8}$$

Si P est une réduction de C, on a $\sigma_{C,D}(P) = 1$, aussi noté $\sigma(P)$ si C et D sont fixés.

Pour conclure, la structure « statique » de la table de décision ne dépend en principe que du domaine applicatif, et non pas des situations rencontrées, car la désignation de l'attribut de décision se fait normalement dès la conception, et les facteurs de consistance et de signification reflètent la qualité des données existantes au niveau de performance d'induction à un moment précis dans l'exploitation. Par contre, dans notre nouvelle approche d'induction des communautés détaillée dans 8.5, tout critère peut être considéré comme attribut de décision selon la situation rencontrée, afin que le système puisse prédire la valeur. Ainsi, chaque critère de formation des communautés peut, en tant qu'attribut de décision, être caractérisé par les deux facteurs de qualité d'induction ci-dessus.

Notre nouveau principe donne une autre vue structurelle sur la table de décision ou table de communautés. Il permet de répondre à certains problèmes principaux du filtrage collaboratif : le démarrage à froid, le rapport coût-bénéfice et la masse critique.

8.5 Qualité de critère dans l'induction de communautés

Nous rappelons que dans le contexte de multiplicité des critères, le vecteur de positionnement d'un utilisateur peut contenir plusieurs communautés manquantes ou douteuses. La cause de cette imperfection est que certains critères demandent beaucoup d'efforts de la part des utilisateurs pour fournir des valeurs, et que certains autres sont coûteux pour le système. Donc, nous avons besoin d'un mécanisme pour induire plusieurs communautés imparfaites dans un même vecteur de positionnement.

Cependant, dans la théorie des ensembles d'approximation, l'attribut de décision est prédéterminé. En principe, elle ne permet d'induire que la valeur du critère choisi a priori comme attribut de décision, et nous ne pouvons par conséquent induire que les communautés d'utilisateurs dans l'espace de ce critère. Au cas où plusieurs communautés sont manquantes ou douteuses dans un vecteur de positionnement, le système peut corriger toutes ces valeurs en considérant successivement chaque critère problématique comme attribut de décision, et cela dans un ordre arbitraire. Mais, cette méthode naïve n'est pas toujours efficace, et il faut donc élaborer une stratégie pour déterminer les espaces auxquels appliquer l'induction, ainsi que l'ordre dans lequel les considérer, afin de commencer par celles dont le résultat est le plus sûr.

Notre objectif est donc de déterminer un ordre sur les attributs de décision à prendre en compte successivement. C'est pourquoi nous proposons dans la suite une extension de la théorie des ensembles d'approximation avec des mesures de qualité de l'attribut de décision considéré dans

l'induction, en vue de comparer les critères et déterminer l'ordre dans lequel appliquer un processus de correction des vecteurs de positionnement par règles.

Nous partons du principe qu'une table de bonne consistance donnera lieu à de bonnes corrections. Les trois mesures proposées dans cette section s'appuient sur la consistance de la table et la signification des attributs de condition qui caractérisent les critères considérés comme attributs de décision.

8.5.1 Mesure basée sur la consistance

Etant donnés l'attribut de décision $d \in D$ et les attributs de condition $C = A \setminus D$, la valeur $\gamma(C, D)$ reflète la consistance de la table T par rapport à l'attribut de décision d. Notre première mesure de qualité d'attribut de décision est basée sur cette notion de qualité d'induction.

Soient deux attributs de décision D_1 et D_2. On dit que D_2 est meilleur que D_1 si ce premier donne lieu à une table de meilleure consistance. Autrement dit, cette mesure favorise l'attribut de décision qui fournit les règles certaines les plus nombreuses.

$$D_1 \lhd D_2 \Leftrightarrow | POS_{A \setminus D1}(D_1) | \leq | POS_{A \setminus D2}(D_2) | \tag{8.9}$$

Nous proposons cette mesure de qualité, orientée système, en nous appuyant sur l'hypothèse que plus le nombre de règles certaines est élevé, mieux les communautés sont induites.

Prenons l'exemple d'un nouvel utilisateur u dont les communautés des critères Evaluation et Genre manquent dans son vecteur de positionnement :

$$\mathcal{P}_u = (\text{Commerçant, Londres, \_, \_})$$

Alors, le Tableau 8.3 montre que le critère Genre est meilleur en tant qu'attribut de décision du fait de la plus grande taille de sa région positive, 10 par rapport à 6. Les colonnes POS_C(Genre) et POS_C(Evaluation) dans ce tableau indiquent les règles certaines incluses dans les régions positives lorsque les critères Genre et Evaluation sont pris comme attribut de décision, respectivement.

Pour compléter le vecteur initial $\mathcal{P}_u$, le système choisit d'abord le critère Genre comme attribut de décision. Ensuite, les règles certaines u_{10}, u_{11} et u_{12} permettent d'inférer « Documentaire » comme la communauté du critère Genre. Alors, on obtient :

$$\mathcal{P}_u = (\text{Commerçant, Londres, Documentaire, \_})$$

Enfin, puisque les règles certaines u_1, u_2, u_3, u_7, u_8 et u_9 dans la région positive POS_C(Evaluation) ne peuvent pas être utilisées pour le nouveau vecteur $\mathcal{P}_u$, le critère Evaluation pourrait être instancié par « Groupe 3 » qui est la valeur dominante dans les trois règles applicables u_{10}, u_{11} et u_{12}.

Utilisateur	*Profession*	*Ville*	*Genre préféré*	*Evaluation*	POS_C*(Genre)*	POS_C*(Evaluation)*
u_1	Commerçant	Paris	Aventure	Groupe 1	x	x
u_2	Chercheur	Paris	Aventure	Groupe 4		x
u_3	Chercheur	Paris	Documentaire	Groupe 2	x	x
u_4	Chercheur	Paris	Scientifique	Groupe 1	x	
u_5	Chercheur	Paris	Scientifique	Groupe 4		
u_6	Chercheur	Paris	Scientifique	Groupe 3	x	
u_7	Chercheur	Paris	Fiction	Groupe 5	x	x
u_8	Chercheur	New York	Documentaire	Groupe 5	x	x
u_9	Commerçant	New York	Documentaire	Groupe 5	x	x
u_{10}	Commerçant	Londres	Documentaire	Groupe 3	x	
u_{11}	Commerçant	Londres	Documentaire	Groupe 2	x	
u_{12}	Commerçant	Londres	Documentaire	Groupe 3	x	

Tableau 8.3 – *Comparaison de critères par la taille de la région positive en résultant*.

Il est à noter que le vecteur initial $\mathcal{P}_u$ ne serait pas complètement corrigé si l'on commençait par le critère Evaluation puisque POS_C(Evaluation) ne contient pas les trois règles u_{10}, u_{11} et u_{12}, et que l'ordre des critères ne dépend que de la situation et pas des cas particuliers d'utilisateurs.

L'exemple précédent montre que l'induction de communauté n'est parfois pas achevée à cause de la petite taille de la région positive contenant les règles certaines. Il serait donc utile d'intégrer d'autres méthodes compatibles de classification par règles. Nous voulons souligner ici que notre mesure de consistance de table est aussi applicable aux méthodes de classification par règles.

En effet, dans ces approches de classification, on utilise souvent les facteurs *support* et *confiance* pour mesurer la qualité des règles présentes dans le classificateur (voir Tableau 8.4). Le support d'une règle est la proportion d'occurrences de cette règle dans l'ensemble des exemples, et la confiance est la proportion d'occurrences de cette règle par rapport aux occurrences de la prémisse [GPS02]. Nous montrons ensuite que les deux facteurs de qualité de règle sont déjà impliqués dans la mesure proposée.

Règle	*Prémisse* $\rightarrow$ *Décision*	*Support*	*Confiance*
r_1	$p_1 \rightarrow d_1$	s_1	c_1
r_2	$p_2 \rightarrow d_2$	s_2	c_2
…	…	…	…
r_λ	$p_\lambda \rightarrow d_\lambda$	s_λ	c_λ

Tableau 8.4 – *Structure d'un classificateur par règles*.

D'abord, on peut considérer la région positive $POS_C(D)$ comme un classificateur spécial, ou plus précisément, on peut générer un classificateur ζ_D à partir de cet ensemble des règles certaines, en sélectionnant les règles représentantes de chaque classe d'équivalence suivant $\mathcal{R}_C$. Alors, chaque règle dans le classificateur généré a toujours une confiance maximale, égale à 1 car elle est une règle certaine, et son support correspond au nombre d'occurrences de cette règle dans la région positive, ou correspond à la taille de sa classe d'équivalence par $\mathcal{R}_C$. La Figure 8.4 illustre un exemple de génération d'un classificateur ζ_D à partir de la région positive $POS_C(D)$. Le support de r_3 est égal à 2/12 puisqu'elle correspond à u_8 et u_9.

Utilisateur	*C*		*D*	$POS_C(D)$
	Ville	*Genre*	*Evaluation*	
u_1	Paris	Aventure	Groupe 1	
u_2	Paris	Aventure	Groupe 4	
u_3	Paris	Documentaire	Groupe 2	x
u_4	Paris	Scientifique	Groupe 1	
u_5	Paris	Scientifique	Groupe 4	
u_6	Paris	Scientifique	Groupe 3	
u_7	Paris	Fiction	Groupe 5	x
u_8	New York	Documentaire	Groupe 5	x
u_9	New York	Documentaire	Groupe 5	x
u_{10}	Londres	Documentaire	Groupe 3	
u_{11}	Londres	Documentaire	Groupe 2	
u_{12}	Londres	Documentaire	Groupe 3	

Règle $\in \zeta_D$	*Prémisse* $\rightarrow$ *Décision*		*Support*	*Confiance*
r_1	(Paris, Documentaire)	$\rightarrow$ Groupe 2	1/12	1
r_2	(Paris, Fiction)	$\rightarrow$ Groupe 5	1/12	1
r_3	(New York, Documentaire)	$\rightarrow$ Groupe 5	2/12	1

Figure 8.4 – *Génération d'un classificateur ζ_D à partir de la région positive $POS_C(D)$.*

Afin d'intégrer d'autres méthodes de classification par règles dans l'induction de communautés, nous proposons une généralisation de la mesure basée sur la consistance comme suit.

Soit le classificateur ζ_D. La qualité d'une règle $r : P \rightarrow D \in \zeta_D$ est mesurée par :

$$\rho(r) = Confiance(r).Support(r) \tag{8.10}$$

où

$$Confiance(r) = \frac{nbre\_occurences(r)}{nbre\_occurences(P)} \tag{8.11}$$

et,

$$Support(r) = \frac{nbre\_occurences(r)}{|U|} \tag{8.12}$$

De plus, on définit la qualité du classificateur par :

$$\varphi(\zeta_D) \equiv \varphi(D) = \sum_{r \in \zeta_D} \rho(r) \tag{8.13}$$

Enfin, nous définissons une autre mesure de qualité de critère en tant qu'attribut de décision en fonction de la qualité du classificateur :

$$D_1 \lhd D_2 \Leftrightarrow |\varphi(D_1)| \leq |\varphi(D_2)| \tag{8.14}$$

Cette mesure est une généralisation de celle de la formule (8.9) puisque dans le contexte de la théorie des ensembles d'approximation reposant sur les régions positives, on a :

$$\varphi(D) = \frac{|POS_C(D)|}{|U|} \tag{8.15}$$

En effet, puisque la confiance des règles est égale à 1, alors :

$$\varphi(D) = \sum_{r \in \zeta_D} Confiance(r).Support(r) = \frac{1}{|U|} \sum_{r \in \zeta_D} nbre\_occurences(r) = \frac{|POS_C(D)|}{|U|} \quad \square$$

8.5.2 Mesures basées sur les réductions approximatives

Dans la théorie des ensembles d'approximation, les réductions jouent un rôle très important pour l'induction par règles. Néanmoins, la définition originale de cette notion est trop forte, et les réductions sont parfois proches de l'ensemble *C*. Dans le cadre d'un système de filtrage collaboratif, les données fournies par les nouveaux utilisateurs concernent souvent un faible nombre de critères. Par conséquent, le système ne peut pas réaliser la tâche d'induction des communautés inconnues si ces critères ne couvrent aucune réduction.

Donc, nous proposons les autres mesures de qualité d'attribut de décision définies à partir des réductions approximatives afin de tenir compte de l'adaptation au contexte applicatif : on cherche les attributs de décision permettant de considérer un sous-ensemble *P* d'attributs de condition aussi réduit que possible tout en conservant une signification supérieure à un seuil θ [Paw04].

Etant donnés l'attribut de décision $d \in D$, les attributs de condition *C et* le seuil θ , on définit d'abord les *réductions approximatives* $R_D^{(\theta)}$:

$$R_D^{(\theta)} = \{P \subseteq C \mid \sigma(P) \geq \theta\} \tag{8.16}$$

Ce sont les sous-ensembles de critères « acceptables » et inclus dans *C*. Ils sont utiles dans le cas où on préfère les règles de petite prémisse plutôt qu'une précision absolue. Par exemple, si θ est égal à 0,8, $R_{\text{Evaluation}}^{(\theta)}$ dans l'exemple précédent contient deux réductions approximatives :

P_1 = {Ville, Genre}

P_2 = {Profession, Ville}

alors qu'avec θ = 1, comme dans la définition originale, on n'a aucune réduction inférieure à l'ensemble *C*.

A partir des ensembles $R_D^{(\theta)}$ on peut définir diverses relations d'ordre sur les attributs de décision, soit en fixant un nombre maximum α d'attributs de condition à conserver, soit en fixant l'ensemble C_0 des attributs de condition jugées utiles dans un contexte donné :

$$D_1 \lhd D_2 \;\Leftrightarrow\; |\{P \in R_{D1}^{(\theta)} \mid |P| \le \alpha\}| \;\le\; |\{Q \in R_{D2}^{(\theta)} \mid |Q| \le \alpha\}| \tag{8.17}$$

$$D_1 \lhd D_2 \;\Leftrightarrow\; |\{P \in R_{D1}^{(\theta)} \mid C_0 \subseteq P\}| \;\le\; |\{Q \in R_{D2}^{(\theta)} \mid C_0 \subseteq Q\}| \tag{8.18}$$

La formule (8.17) favorise l'attribut de décision qui donne de petites réductions, ce qui signifie que l'on a besoin de connaître peu de choses sur l'utilisateur. Plutôt que la taille des réductions, la formule (8.18) prend en compte leur contenu, préférant par exemple les attributs de condition demandant peu d'effort à produire par l'utilisateur (son code postal, par exemple). En tout cas, ce sont deux mesures orientées utilisateurs opposées à la mesure orientée système basée sur la consistance de la table de communautés.

En pratique, puisque la tâche de déterminer toutes les réductions relatives à un attribut de décision $d \in D$, nécessaire pour analyser sa qualité par (8.17) ou (8.18), est un problème NP-difficile, on peut limiter a priori la taille de $R_D^{(\theta)}$ pour diminuer la complexité du calcul, ou calibrer le seuil θ pour que les réductions dans $R_D^{(\theta)}$ contiennent dans la plupart des cas les critères de C_0.

8.5.3 Mesure basée sur la consistance approximative

Nous proposons finalement une mesure complémentaire permettant de départager les attributs de décision que les mesures définies dans (8.17) ou (8.18) ne peuvent le faire. Elle repose sur la consistance $\mu(D)$ de la table de communautés par rapport aux réductions approximatives.

$$\mu(D) = \frac{1}{\left|R_D^{(\theta)}\right|} \sum_{P \in R_D^{(\theta)}} \sigma(P) \tag{8.19}$$

$$D_1 \lhd D_2 \Leftrightarrow \left|\mu(D_1)\right| \le \left|\mu(D_2)\right| \tag{8.20}$$

En résumé, nous venons de présenter dans ce chapitre notre modèle des espaces de communautés pour la gestion des communautés multicritères et explicites dans un système de filtrage collaboratif, en partant de l'hypothèse que les espaces de communautés sont préalablement formés. Dans le suivant chapitre, nous discutons les principaux aspects du processus de former des communautés par divers critères afin de remplir la table de communautés.

Chapitre 9

Formation des communautés

Comme le montre la Figure 9.1, la formation des communautés réalise la construction « directe » des espaces de communautés. La tâche de formation des communautés est très importante pour notre modèle des espaces de communautés puisqu'il se base fondamentalement sur la table de communautés, remplie par ces espaces. Ainsi, le présent chapitre décrit notamment les étapes pour former des communautés.

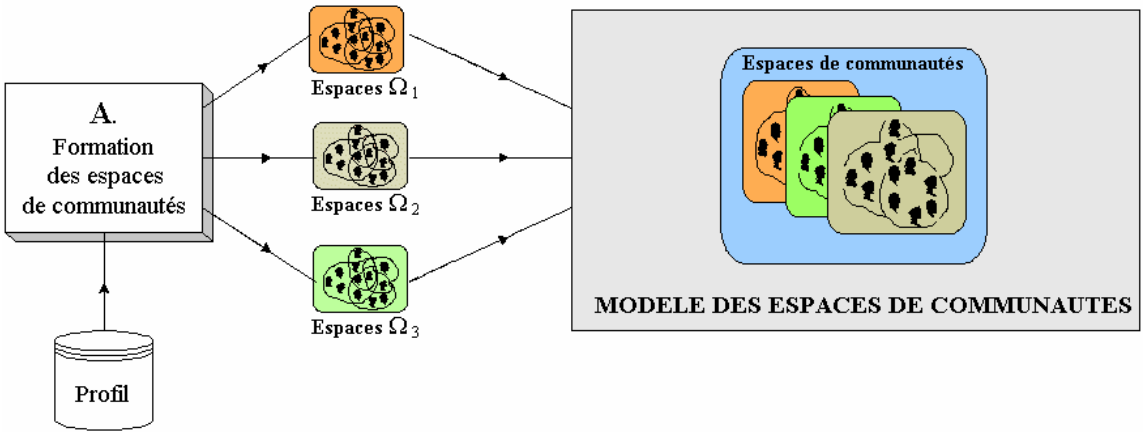


Figure 9.1 – *Formation des communautés.*

Dans notre modèle, les espaces représentés dans la table de communautés *T* (voir Tableau 8.1) sont construits à partir des données disponibles dans les profils des utilisateurs. Pour certains critères, les communautés sont créées de façon directe (sexe, ville), et pour certains autres, on doit regrouper plusieurs valeurs afin de former des communautés plus significatives, par exemple tranche d'âge, catégorie de profession, etc.

Par contre, pour les critères plus complexes comme les centres d'intérêt ou l'historique des évaluations, on doit suivre les étapes décrites ci-après afin d'obtenir les communautés pour le critère concerné. Nous voulons souligner ici que notre approche des cartes de communautés en 2D présentée dans le Chapitre 11, qui facilite la perception des communautés, s'appuie également sur ces étapes.

Comme le montre la Figure 9.2, la formation des communautés consiste à :

- définir les critères issus des profils pour former des communautés,
- extraire les valeurs des critères pour les utilisateurs, en les considérant comme les points dans un espace en *d* dimensions, selon le critère concerné,
- calculer la proximité ou les distances entre utilisateurs par critère,
- appliquer les méthodes de classification sur les distances des utilisateurs afin de générer les espaces de communautés, et
- mesurer la qualité des espaces de communautés.

Nous présentons par la suite une brève analyse sur les éléments de la littérature concernant les étapes de la formation des communautés.

9.1 Définition des critères de formation des communautés

En principe, la formation des communautés doit reposer sur un attribut ou critère particulier dans les profils d'utilisateurs. Nous adoptons la taxonomie des critères en fonction de la nature de données proposée dans [AS99, BK05]. D'après ces auteurs, les composants d'un profil peuvent être regroupés selon six dimensions : informations personnelles, centres d'intérêt, préférences de qualité, de livraison et de sécurité, et l'historique de l'interaction.

a) Informations personnelles. Cette dimension de critère comporte toutes les informations personnelles propres à l'utilisateur telles que son identité (nom, prénom, numéro de carte d'identité), ses données démographiques (sexe, âge, situation familiale, coordonnées, langues, profession), etc. Ce sont des informations plus ou moins statiques et pour lesquelles il est facile d'obtenir, sous réserve de confidentialité. Par conséquent, les communautés relatives à des critères issus de cette dimension sont assez fiables pour être exploitées en début d'utilisation. En revanche, l'utilisation de ces critères ne semble pas convenir à la situation où l'utilisateur est en plein l'évolution du besoin en information. Pourtant, nous montrons dans le Chapitre 12 que notre approche d'induction des communautés permet

d'utiliser ces critères personnels pour des situations délicates. Supposons que l'utilisateur est un professeur et qu'il n'est plus satisfait par les recommandations récentes tirées directement de sa communauté de profession. Alors, le système peut déclencher le processus d'induction des communautés en utilisant sa profession pour lui proposer de nouvelles communautés relatives aux autres critères, et ceci lui permet par conséquent de découvrir d'autres domaines potentiellement intéressants.

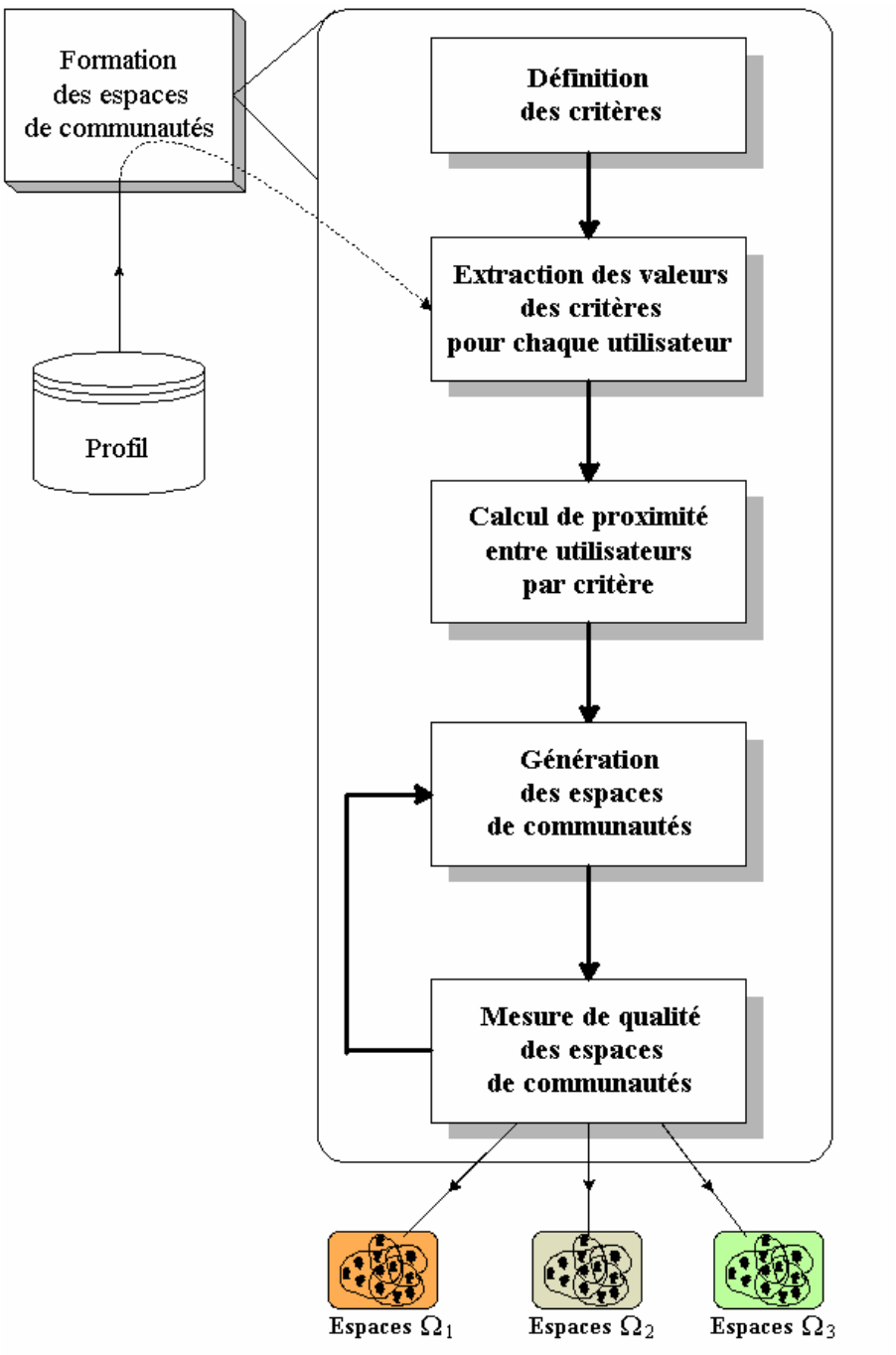


Figure 9.2 – *Etapes de formation des communautés.*

b) Centres d'intérêt. Ce sont les domaines auxquels s'intéresse l'utilisateur. Cette dimension traduit ce dont il a besoin en termes de contenu d'information, et de ce fait il est souvent qualifié de critère « Contenu ». Dans notre approche, ces critères peuvent également être utilisés pour construire des communautés d'utilisateurs. Bien que l'effet « entonnoir » soit un inconvénient inévitable du filtrage basé sur le contenu, notre approche apporte un élément de solution à ce problème. En effet, un utilisateur dans une communauté fondée sur les centres d'intérêt peut non seulement profiter des informations liées aux centres d'intérêt des membres de sa communauté mais aussi découvrir d'autres informations intéressantes par ses voisins grâce aux positions de ces personnes dans les autres espaces de communautés.

c) Qualité. Cette dimension contient tous les facteurs reflétant les préférences relatives à la qualité de l'information, comme la disponibilité de données, le niveau de confiance, la réputation des auteurs, la fraîcheur, la concision, le style et la structure du document, etc. Dans cette dimension, nous nous intéressons en particulier à la diversité de l'information.

d) Livraison. On peut citer à titre d'exemple la modalité de livraison des informations, comme le format, le standard, le volume et le mode de visualisation, le délai et le prix, etc.

e) Sécurité. La dimension de sécurité dans le contexte du filtrage collaboratif, est le niveau de confidentialité concernant tous les autres critères. Par exemple, la plateforme de filtrage collaboratif COCoFil [DBG+04] permet à l'utilisateur de définir le niveau de confidentialité sur les diverses parties de son profil dans les activités d'échange d'informations entre utilisateurs. L'utilisateur dans cette plateforme a normalement la possibilité d'indiquer qu'il souhaite l'anonymat : ses informations personnelles ne seront alors pas divulguées. Aussi, il peut choisir de conserver l'anonymat pour ses évaluations, sachant qu'il pourra très simplement, lors d'une évaluation particulière, choisir l'option inverse. Il faut néanmoins remarquer que l'anonymat des utilisateurs diminue la crédibilité des recommandations dans le filtrage collaboratif. En effet, l'identification et au moins la pseudonymie enrichissent la perception des recommandations circulant dans la communauté. Par exemple, il semble naturel qu'un utilisateur s'intéresse plus aux suggestions de personnes identifiées qui lui ont souvent adressé des recommandations pertinentes par le passé ; de même, il aura sans doute tendance à accorder moins d'importance aux recommandations anonymes.

f) Historique de l'interaction. Cette dernière dimension comprend tous les retours de pertinence que l'utilisateur a donnés sur les recommandations. Ces retours de pertinence prennent la forme de scores ou autres valeurs qualitatives explicites telles que excellent, bon, moyen, mauvais, etc., ou bien de préférences implicites que le système peut déduire à travers le comportement de l'utilisateur, par exemple lire, sauvegarder, supprimer, imprimer, envoyer à ses amis les recommandations. L'historique de l'interaction est le critère classique de formation des communautés dans les systèmes de filtrage collaboratif [BHK98, MLD03, PGF03, RIS+94]. Ce critère exprime en général l'historique des évaluations d'utilisateurs, et nous utilisons souvent le nom « Evaluation » pour ce critère. Un autre cas de cette dimension est le critère « Motivation » qui reflète la tendance des utilisateurs à fournir régulièrement des évaluations sur les recommandations reçues [Gal05].

Nous proposons en plus une autre vision sur ces critères sous l'angle de l'effort de la part des utilisateurs pour les renseigner, ainsi que de la part du système pour former les communautés s'y

rapportant. Dans la Figure 9.3, on retrouve les critères simples tels qu'informations personnelles ainsi que les critères plus complexes demandant plus d'efforts aux utilisateurs pour fournir les données correspondantes.

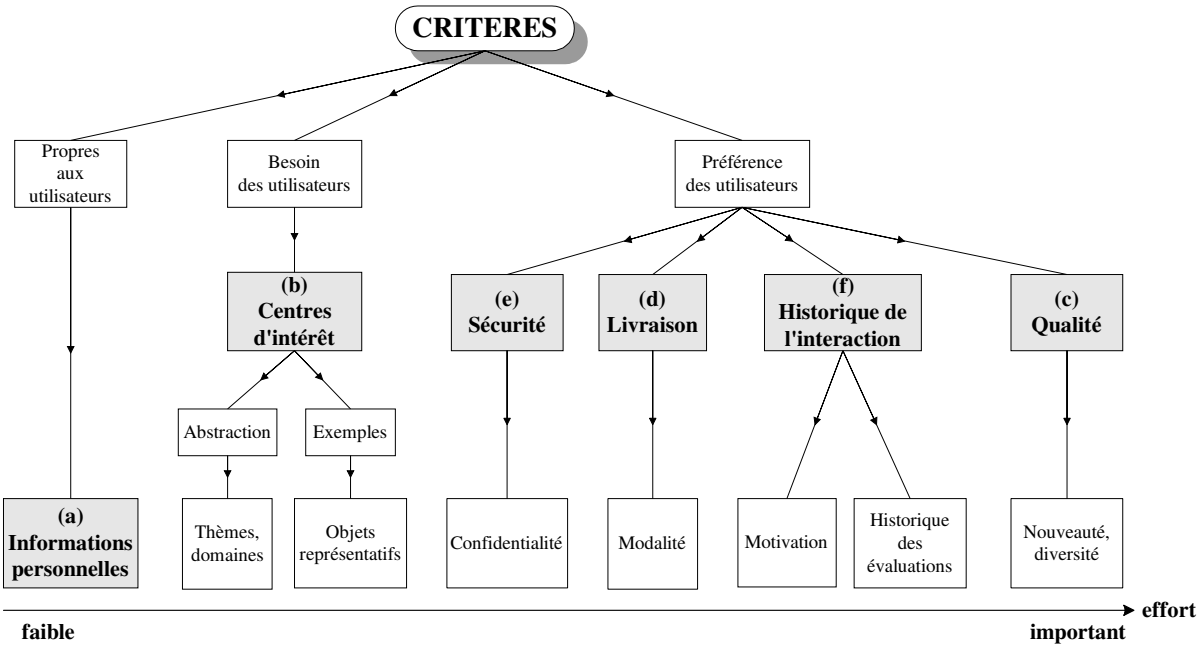


Figure 9.3 – *Taxonomie des critères de formation des communautés*.

9.2 Extraction des valeurs des critères

Une fois les critères choisis, on doit extraire des valeurs des critères pour chaque utilisateur[7]. Pour les critères simples comme dans la dimension des informations personnelles, on peut réutiliser les valeurs ou en regrouper pour obtenir des valeurs plus significatives, par exemple tranche d'âge, catégorie de profession, etc.

Pour le critère Evaluation, chaque utilisateur peut être considéré comme un point dans un espace en d dimensions ($d \geq 2$), et caractérisé par la liste, ou le vecteur, de ses propres évaluations, éventuellement raffinée. Quant au critère Contenu, dans la plupart des systèmes existants, un utilisateur est caractérisé par un vecteur des termes ou sujets préférés, sans ou avec pondération. On peut trouver dans [MLD03] une revue récente des travaux de représentation, construction et adaptation de la partie de contenu dans les profils.

[7] Dans certains domaines de recherche, on utilise aussi le terme « extraction des caractéristiques » (*Feature Extraction*) pour cette étape.

9.3 Calcul de proximité entre utilisateurs

Pour chaque critère de formation des communautés, il faut définir une mesure de proximité entre les utilisateurs. Pour les critères simples, la formation des communautés est simplement le regroupement des utilisateurs par comparaison des valeurs des données dans les profils. Bien que les données des critères simples puissent être autant quantitatives que qualitatives, la distance entre utilisateurs dans une communauté d'un tel critère n'est guère significative. Par exemple, on peut regrouper deux utilisateurs de 12 et 15 ans dans une communauté des adolescents, mais l'écart d'âge entre eux nous donne peu de signification.

Par contre, pour les critères plus complexes comme Contenu ou Evaluation, on a besoin de mesures pour calculer la proximité entre les utilisateurs. Les mesures de proximité entre utilisateurs sont nécessaires aux méthodes de classification non supervisée, pour les critères dont les connaissances a priori sur le modèle des communautés sont inconnues. On peut trouver de nombreuses études sur l'utilisation des distances dans différents domaines d'application comme : la segmentation des documents audio [Cam97], le traitement d'images [RPT+01, Vel01], la recherche d'information [TvR04], etc. Les mesures de proximité les plus utilisées sont les distances de Minkowski, euclidienne, de Manhattan, de Chebychev, de Mahalanobis, du $\chi 2$, de Kullback-Leibler, le cosinus, la corrélation de Pearson, etc. [JMF99]. Il existe cependant peu d'études concernant la formation des communautés dans les systèmes de filtrage collaboratif. Dans ce contexte, on constate que la corrélation de Pearson et le cosinus sont les mesures les plus utilisées pour le critère Evaluation [BHK98, Her00, HKJ+99].

9.4 Génération des espaces de communautés

Afin de générer l'espace de communautés d'un critère donné, on doit appliquer des méthodes de classification. Il existe deux grandes familles de méthodes de classification : *supervisée* et *non supervisée*, aussi connue sous le nom de regroupement (*clustering*) ou partitionnement.

Dans une méthode de classification supervisée, le modèle des classes doit être connu préalablement à l'étape de classification. En général, ce modèle est construit par apprentissage sur un ensemble d'échantillons, et à l'étape de classification, le système prédit une des classes pour la nouvelle observation. On peut citer des méthodes supervisées connues comme la classification Bayésienne, les arbres de décision, les réseaux neuronaux [Heb49, PMc43], les machines à vecteurs de supports (*Support Vector Machine – SVM*) [Vap82, 95], les *K* plus proches voisins, etc. [JMF99, Mit97].

Les étapes décrites dans ce chapitre visent notamment à la formation des espaces de communautés pour les critères complexes comme Contenu ou Evaluation. Pour ces critères, les méthodes supervisées ne sont pas efficaces car les modèles préalables de communautés sont souvent difficiles à obtenir. A titre d'exemple, dans l'application de la classification Bayésienne pour former des communautés par le critère Evaluation [BHK98], la définition des paramètres du modèle de communautés tels que le nombre de communautés et la probabilité a priori d'appartenance à une communauté, est loin d'être une tâche évidente.

De ce fait, nous nous en remettons à la deuxième famille de méthodes de classification, non supervisée. Parmi de nombreux algorithmes de classification non supervisée existants, on peut citer trois algorithmes parmi les plus populaires : la classification ascendante hiérarchique, l'algorithme des K-moyennes, et l'algorithme des C-moyennes floues, qui sont respectivement les représentants des trois approches de classification hiérarchique, de partitionnement, et de classification floue [Ber02, JMF99]. Dans le reste du manuscrit, nous utilisons le terme « méthode de classification » pour désigner une des méthodes non supervisées.

Outre ces méthodes classiques, l'approche des fourmis artificielles (*Ant Colony Optimization* – ACO) que nous utilisons dans notre approche, permet de répartir une population sur un plan, en 2D, tout en reflétant la proximité entre les individus [APG+04, DGF+90].

9.5 Qualité des espaces de communautés

Pour estimer la qualité des espaces de communautés, on peut utiliser les critères populaires comme les inerties d'une classe, intra-classe et inter-classes [HKD03, JMF99, Gué03], ou d'autres critères comme l'entropie spatiale [Sha48] dans le cas d'un nombre important d'utilisateurs. En principe, on essaie de minimiser l'inertie intra-classe et de maximiser l'inertie inter-classes.

$$I_{classe}(C_j) = \sum_{x,y \in Cj} d(x,y) \tag{9.1}$$

$$I_{classe\_gravité}(C_j) = \sum_{x \in Cj} d^2(x,g_j) \tag{9.2}$$

$$I_{\text{intra}}(\{C_1,...,C_k\}) = \sum_{j=1}^{k} \sum_{x \in C_j} d^2(x,g_j) \tag{9.3}$$

$$I_{\text{inter}}(C_1,...,C_k) = \sum_{j=1}^{k} |C_j| . d^2(g_j, g) \tag{9.4}$$

où $d(x, y)$: distance entre x et y
g_j : centre de gravité de la classe C_j
g : centre de gravité de l'ensemble $\{C_1,...,C_k\}$

$$Entropie_S(C_1,...C_k) = -\sum_{j=1}^{k} p(C_j).\log(p(C_j)) \tag{9.5}$$

$$\text{où } p(C_j) = \frac{|C_j|}{nombre\ \ d'objets}$$

En résumé, en appliquant les étapes de formation des communautés sur les critères issus des profils, on obtient à la fin les espaces de communautés, comme illustré dans la Figure 9.2. Nous rappelons ici que ces espaces de communautés ont pour but d'alimenter les tables de communautés

dans notre modèle proposé pour la gestion des communautés dans un système de filtrage collaboratif. Par ailleurs, suite aux étapes ci-dessus, nous présentons dans le Chapitre 11 l'approche des cartes de communautés en 2D pour la formation des communautés en facilitant la perception des communautés.

Chapitre 10

Bilan

Avec la représentation formelle des communautés multicritères par une table de communautés, des espaces de communautés et des vecteurs de positionnement en utilisant les relations d'équivalence, nous répondons au problème de la formation monocritère des communautés dans les systèmes de filtrage (voir l'introduction). De plus, cette formalisation rend complètement explicites les communautés, et elles ne sont plus considérées simplement comme des résultats intermédiaires. Ceci nous offre une base prometteuse pour exploiter au mieux les espaces de communautés. Il s'agit d'une méthode d'induction des communautés pour améliorer la performance du positionnement des utilisateurs au sein des communautés multicritères. En outre, la formation multiple des communautés et les éventuels contrastes qu'elles sont susceptibles de faire surgir, peuvent servir de point de départ à l'adaptation du profil de l'utilisateur. En bref, avec notre modèle des espaces de communautés, les communautés deviennent une modalité efficace pour la personnalisation au-delà des évaluations d'utilisateurs.

Par ailleurs, notre approche d'induction des communautés répond également au problème d'intégration de nouveaux utilisateurs. Effectivement, on peut exploiter les données disponibles à froid comme les informations personnelles pour offrir à tout nouvel utilisateur un premier vecteur de positionnement complet sans qu'il ait d'effort à fournir.

Nous soulignons ici une différence importante entre notre modèle et d'autres applications de la théorie des ensembles d'approximation : la théorie fait l'hypothèse que l'attribut de décision est fixé dès la conception du système, et que l'on essaie de sélectionner à partir d'un ensemble d'apprentissage des réductions d'attributs de condition en conservant la qualité de la table de décision. On peut dire que cette théorie focalise notamment sur la qualité des attributs de condition plutôt que des attributs de décision. Au contraire, dans notre modèle aucun attribut de décision n'est prédéfini. Au départ, tous les critères sont sur le même plan, et pendant le temps d'exploitation, le système va choisir parmi les critères le meilleur attribut de décision en analysant les données de référence, voire toutes les données disponibles, afin de réaliser une certaine tâche de filtrage d'information dans une situation particulière.

Enfin, nous voyons que le modèle proposé ne vise pas à remplacer les systèmes de filtrage existants, mais offre des moyens supplémentaires prenant en compte un cadre plus large (communautés multicritères), et offre des moyens pour positionner les utilisateurs se trouvant dans des situations problématiques.

Partie IV.

Plateforme COCoFil2 : Mise en œuvre du modèle des espaces de communautés

Après avoir présenté le modèle des espaces de communautés dans la partie précédente, nous décrivons sa mise en œuvre pour une plateforme de filtrage collaboratif. C'est la plateforme COCoFil2, successeur de la plateforme COCoFil (*Community-Oriented Collaborative Filtering*) [DBG+04] décrite dans l'état de l'art. Notre nouvelle plateforme est orientée vers les espaces de communautés, et se compose de trois modules principaux qui reposent tous trois sur les espaces de communautés : formation des communautés, induction des communautés et filtrage d'information. Le schéma fonctionnel de notre mise en œuvre, incluant la formation et l'induction des communautés et le filtrage d'information, est illustré dans la Figure IV.1.

a) Le module de formation des communautés construit les espaces de communautés selon les critères disponibles dans les profils d'utilisateurs pour alimenter la table de communautés.

b) Le module d'induction des communautés complète le premier module pour améliorer la table de communautés. En utilisant les mesures de qualité de critère dans l'induction des communautés par règles, ce module est utile au positionnement des utilisateurs au sein des communautés en particulier dans les situations délicates comme le démarrage à froid, l'évolution importante du besoin en information, etc.

c) Le module de filtrage d'information produit les recommandations en prenant en compte l'opinion des communautés multicritères présentes dans la table de communautés. Les recommandations générées pour un utilisateur à partir de ses propres communautés peuvent être exploitées séparément pour chaque critère de formation ou de façon combinée selon une stratégie d'hybridation particulière.

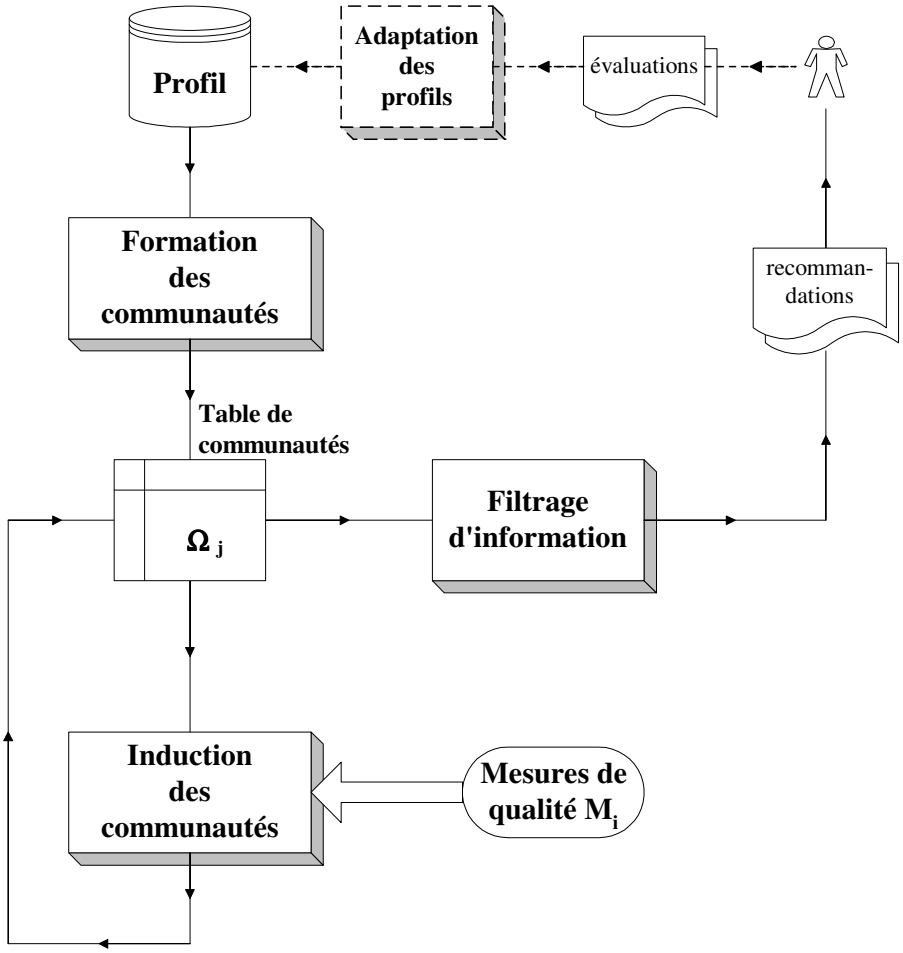


Figure IV.1 – *Schéma fonctionnel de la plateforme COCoFil2.*

Finalement, il faut noter la présence du module d'adaptation des profils dans le schéma de fonctionnel, qui n'est pas l'ultime objectif mais plutôt une des perspectives de la thèse. La contribution potentielle de notre approche d'espaces de communautés à l'adaptation des profils d'utilisateurs sera discutée dans la conclusion.

Chapitre 11

Formation des communautés

Le module de formation des communautés a pour but, comme le nom l'indique, de regrouper les utilisateurs en communautés, selon un critère donné. Les résultats de ce module sont des espaces de communautés qui alimentent la table de communautés dans notre modèle des espaces de communautés. Dans le Chapitre 9, les étapes pour former des communautés sont déjà discutées. Et, dans le présent chapitre, nous présentons notre approche des cartes en 2D pour la formation des communautés en suivant les étapes précitées.

Comme le montre la Figure 11.1, notre approche est une formation multiple de communautés selon tous les critères disponibles dans les profils des utilisateurs, en combinant l'algorithme des fourmis artificielles et l'algorithme classique des K-moyennes afin de faciliter la perception des communautés. Avant de décrire par la suite notre approche des cartes de communautés, nous voulons préciser ici que nous orientons nos choix dans les étapes par analogie avec les usages habituels.

11.1 Choix des critères

Dans l'optique d'une diversification des recommandations et d'une compensation inductive maximale des espaces de communautés, nous adoptons en principe tous les critères disponibles (voir Tableau 11.1) pour créer autant d'espaces de communautés dans la plateforme COCoFil2. De plus, nous précisons ici que dans notre modèle tous les critères présents dans le système sont généralement

sur le même plan et qu'aucun critère n'est prédéfini comme critère clé ; c'est pendant le temps de l'exploitation que le système choisit le critère clé en fonction de la situation rencontrée.

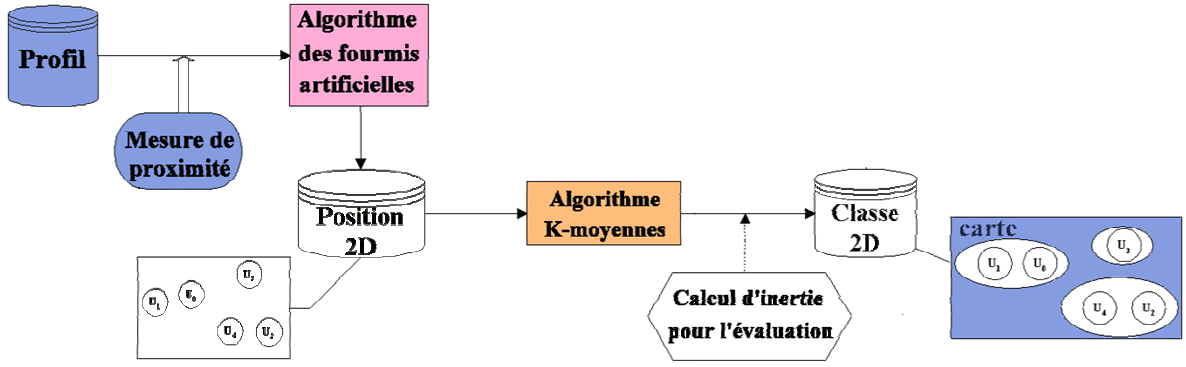


Figure 11.1 – *Schéma de création des cartes de communautés*.

Etape	Mise en œuvre
1. Choix des critères	• informations personnelles • centres d'intérêt • préférence de qualité • préférence de livraison • préférence de sécurité • historique de l'interaction ou des évaluations
2. Extraction des valeurs des critères	• modèles vectoriels
3. Mesures de proximité	• corrélation de Pearson • cosinus • distances de Minkowski, euclidienne
4. Génération des espaces	• combinaison de l'algorithme des fourmis artificielles et l'algorithme des K-moyennes
5. Qualité des espaces	• inerties

Tableau 11.1 – *Table des choix pour la mise en œuvre de la plateforme COCoFil2*.

11.2 Extraction des valeurs des critères

Dans cette étape, nous choisissons les modèles vectoriels, dans lesquels les utilisateurs sont en général caractérisés par des vecteurs, et considérés comme des points dans un espace en *d* dimensions. Pour certains critères assez simples comme informations personnelles, la dimension *d* est égale à 1, et

chaque utilisateur prend comme caractéristique une valeur particulière. Pour d'autres plus complexes comme les centres d'intérêt ou l'historique des évaluations, la valeur de dimension d est peut-être égale ou souvent supérieure à 2. Par exemple, chaque utilisateur dans un système de recommandation de films est représenté par le vecteur traduisant ses genres de film préférés, pour le critère des centres d'intérêt, et par le vecteur de ses propres évaluations, pour le critère de l'historique des évaluations.

11.3 Mesures de proximité entre utilisateurs

En général, nous nous appuyons sur les mesures de proximité telles que la corrélation de Pearson, les distances de Minkowski, euclidienne, le cosinus, etc. selon le critère concerné, par analogie avec les usages habituels dans la mise en œuvre de la plateforme COCoFil2 (voir Tableau 11.1).

11.4 Génération des espaces de communautés

Dans notre approche, les communautés peuvent être utilisées non seulement dans la production des recommandations mais aussi dans d'autres activités du filtrage collaboratif, à condition d'une perception totale. Autrement dit, les espaces de communautés doivent être aisément représentables et perceptibles par les utilisateurs. Pour faciliter la visualisation, nous proposons l'approche des cartes de communautés en 2D, à partir d'une combinaison de l'algorithme des fourmis artificielles et l'algorithme des K-moyennes, comme méthode de classification dans la formation des communautés (voir Figure 11.1).

Avant de décrire dans la section 11.4.3 notre méthode de création des cartes en 2D facilitant la perception des communautés, nous présentons dans 11.4.1 et 11.4.2 les deux algorithmes de base que nous utilisons dans notre proposition.

11.4.1 Algorithme des fourmis artificielles

L'algorithme des fourmis artificielles [APG+04, DGF+90, MSV01] est proposé pour la première fois par Denebourg pour le problème du tri d'objets. L'auteur imite le comportement des fourmis réelles dans la nature pour construire son algorithme. L'idée principale de cet algorithme est que les objets correspondant à des points dans un espace à d dimensions ($d > 2$), sont plongés dans un espace en 2D, c'est à dire une grille dont chaque cellule peut contenir un objet (voir Figure 11.2 et Figure 11.3) :

– Un agent (fourmi) n'a qu'une perception locale des objets et n'est pas capable de communiquer avec les autres ;

– Lorsqu'un agent libre rencontre un objet dans une cellule, il le ramasse avec une probabilité de $(k_1 /(k_1 + f))^2$, où la fonction de densité f représente la proportion des objets qu'il a récemment rencontrés par rapport au nombre maximal d'objets qu'il aurait pu rencontrer, et k_1 est une constante ;

– Une fois qu'un objet a été ramassé, l'agent chargé se déplace au hasard dans la grille et dépose l'objet avec une probabilité de $(f/(k_2 + f))^2$, où k_2 est une constante. Il ne le dépose que si cette probabilité dépasse un certain seuil.

– Le test d'arrêt se base notamment sur un seuil (t_{max}) pour le nombre d'itérations.

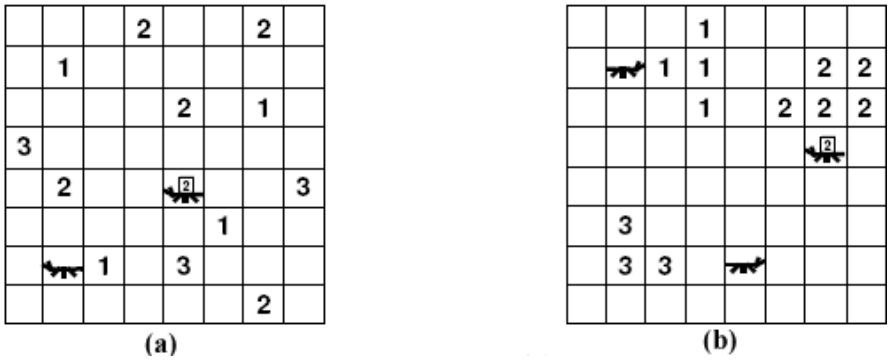


Figure 11.2 – *Exemple pour illustrer l'algorithme des fourmis artificielles* [APG+04].

```
LF_AntBasedClustering (Objets = {x1,…, xn}, Agents = {a1,…, aA}, grille G)
{  S0. INITIALISATION : Placer aléatoirement les objets et les agents sur G.
   S1. BOUCLE PRINCIPALE
      for (t = 1; t <= tmax ; t++)
         for (ag = 1; ag <= nombre d'agents; ag++)
         {
            if (l'agent ag ne transporte pas d'objet et trouve un objet xi)
               if (la probabilité de ramasser probPick(ag, xi) >= seuil)
                  pickItem(ag, xi);
            else if (l'agent ag transporte un objet xi et la cellule est vide)
               if (la probabilité de déposer probDrop(ag, xi) >= seuil)
                  dropItem(ag, xi);
            L'agent ag se déplace vers une cellule non occupée
         }
   S2. return (l'emplacement des objets {x1,…, xn} sur la grille G)
}
```

Figure 11.3 – *Algorithme des fourmis artificielles.*

En 1994, Lumer et al. ont apporté des modifications à cet algorithme [LF94], notamment en remplaçant la fonction de densité *f* par une moyenne des similarités entre l'objet en question x_i et les objets situés dans son voisinage (cf. (11.6)). L'algorithme d'origine de Denebourg est donc devenu un véritable algorithme de classification, et on trouve actuellement de nombreuses études et applications sur cette classification biomimétique [APG+04, DBT00]. Les avantages de cet algorithme sont la possibilité de l'appliquer à plusieurs types de données et la possibilité de visualiser le résultat aisément.

$$f(x_i) = max\left(\frac{1}{s^2}\sum_{x \in V(x_i)}\left[1 - \frac{d(x, x_i)}{\alpha}\right], 0\right) \tag{11.6}$$

où s^2 : taille du voisinage *V*, et
α : constante

On voit que plus l'objet x_i est similaire aux objets dans son voisinage, plus il a de chances d'être déposé, et moins il a de chances d'être ramassé.

En ce qui concerne la performance de l'algorithme, Handl et al. ont présenté leur comparaison de l'algorithme des fourmis artificielles modifié avec trois autres algorithmes : K-moyennes, classification ascendante hiérarchique et cartes de Kohonen (*One-dimensional Self-Organizing Maps - 1D-SOM*) sur 6 jeux de données dont 3 jeux artificiels et 3 jeux réels provenant de [UCI Knowledge Discovery in Databases Archive] [HKD03]. En utilisant les quatre critères F_Measure, Rand Index, Inner Cluster Variance et Dunn Index, les auteurs ont observé que l'algorithme des fourmis artificielles donne d'excellents résultats.

En général, l'algorithme des fourmis artificielles est une méthode de projection comme les cartes de Kohonen [Koh97]. Nous préférons cet algorithme biomimétique en raison de sa simplicité et sa bonne performance par rapport à ces dernières [HKD03].

Il existe dans la littérature plusieurs extensions pour rendre plus performant l'algorithme des fourmis artificielles. D'abord, les agents dans [LF94] ont des vitesses différentes et de courtes mémoires sur les objets qu'ils ont récemment déposés pour améliorer le déplacement des agents, et ils peuvent également détruire les accumulations d'objets selon le cas. Montmarché et al. proposent la possibilité d'empiler des objets sur une même cellule [MSV99, MSV01] ; Labroche et al. exploitent les groupes d'agents plutôt que les individus [LMV02], etc. Pourtant, en ce moment, nous appliquons simplement l'algorithme des fourmis artificielles de base, et ses extensions seront étudiées dans l'optique d'améliorer la performance de notre méthode.

11.4.2 Algorithme des K-moyennes

L'idée principale de l'algorithme des K-moyennes, aussi utilisé dans notre méthode de création des cartes de communautés, est de classifier des objets en *K* classes en minimisant la variance intra-classe et en maximisant l'écartement inter-classes [JMF99, McQ67]. Cet algorithme commence par choisir au hasard *K* centres de gravité (voir Figure 11.4 et Figure 11.5). Puis, on construit les classes initiales autour de ces centres : chaque objet appartient à la classe dont le centre est le plus proche. A

chaque itération, on recalcule les centres en fonction de la variance intra-classe, et on forme les nouvelles classes jusqu'à ce que l'on n'obtienne plus de changement de partition.

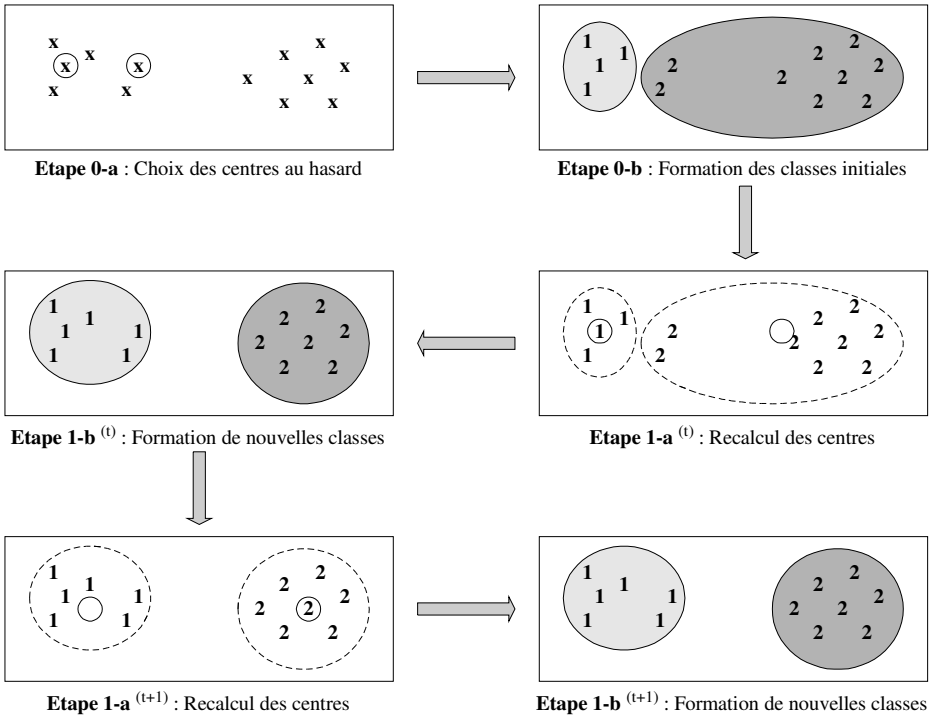


Figure 11.4 – *Exemple pour illustrer l'algorithme des K-moyennes.*

K_moyennes (Objets = {x_1,…, x_n}, k)

{ S0. **INITIALISATION :**

a) Choisir au hasard k centres de gravité : $G^{(0)} = \{g_1,\ldots,g_k\}$

b) Construire la partition de k classes : $C^{(0)} = \{C_1,\ldots, C_k\}$

où $C_i = \{ x \in Objects / \forall i \neq j, d(x,g_i) < d(x,g_j) \}, \forall j \in [1,k]$ }

S1. **BOUCLE PRINCIPALE**

a) Recalculer les centres de gravité : $G^{(t)} = \{g_1,\ldots,g_k\}$

où $g_j = \frac{\sum_{x \in C_j} x}{|C_j|}, \forall j \in [1,k]$

b) et former les nouvelles classes : $C^{(t)} = \{C_1, \ldots, C_k\}$

Test d'arrêt : ($C^{(t+1)} \cong C^{(t)}$) ou (nombre d'itération > seuil)

S2. **return** (la partition $C^{(t)} = \{C_1, \ldots, C_k\}$)

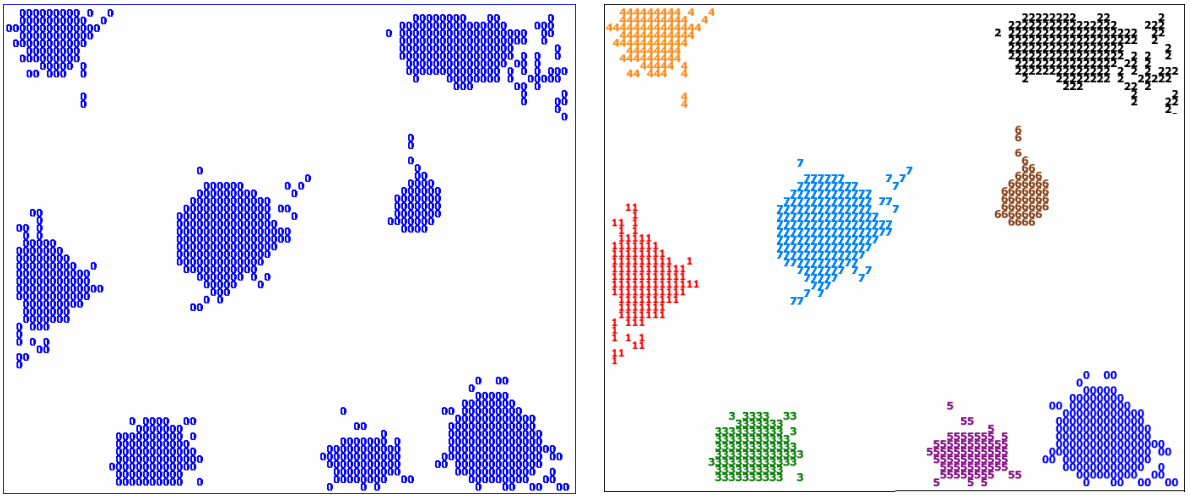
}

Figure 11.5 – *Algorithme des K-moyennes.*

L'algorithme des K-moyennes est une des méthodes de classification les plus populaires en raison de sa simplicité et son efficacité dans la plupart des cas [JMF99].

11.4.3 Méthode de création des cartes en 2D

Supposons que le système doit créer la carte des communautés par le critère des évaluations des utilisateurs. D'abord, chaque utilisateur est caractérisé par la liste, ou le vecteur, de ses évaluations dans le passé, et il est considéré comme un point dans l'espace $\mathbb{R}^d$, où d est le nombre de documents. Puis, tous les utilisateurs sont aléatoirement placés dans une grille pour que l'algorithme des fourmis artificielles puisse s'appliquer sur les positions initiales des utilisateurs afin d'obtenir leur emplacement dans l'espace en 2D (voir Figure 11.6.a). Jusqu'à ce moment, les étiquettes des communautés ne sont pas encore désignées aux utilisateurs. Finalement, l'algorithme des K-moyennes s'applique sur ces emplacements pour étiqueter les communautés, et on obtient la carte illustrée dans la Figure 11.6.b.



(a) (b)

Figure 11.6 – *Création des cartes de communautés.*

11.5 Qualité des espaces de communautés

Suite aux algorithmes impliqués dans notre méthode de création des cartes en 2D, la qualité des espaces de communautés dans la plateforme COCoFil2 est mesurée par les inerties sous-jacentes.

En résumé, notre approche, consistant à créer des cartes ou espace de communautés pour alimenter la table de communautés, fournit en même temps un moyen d'interaction entre le système et

les utilisateurs. Cette possibilité a pour but de dépasser le problème de la perception limitée des communautés dans les systèmes de filtrage collaboratif, et offre également une bonne base pour l'adaptation interactive des profils des utilisateurs. En effet, notre approche rend complètement explicites les communautés dans les systèmes de filtrage collaboratif. Elles ne sont plus considérées comme les résultats intermédiaires pour la production de recommandations, et elles offrent la perspective d'une nouvelle modalité d'interaction dans les systèmes de filtrage collaboratif.

Par exemple, à titre d'illustration, si un utilisateur est en plein l'évolution du besoin en information, et si ses évaluations ne sont donc plus utiles à l'adaptation incrémentale de son profil, le système lui fournit une possibilité d'adaptation plus radicale. En se basant sur une base de connaissances sur les situations rencontrées et la capacité cognitive, l'interaction à travers les cartes de communautés lui permettra de choisir lui-même ses nouvelles communautés, et par conséquent d'hériter de profils typiques de ces communautés.

Chapitre 12

Induction des communautés

La fonctionnalité principale de ce module est de prédire pour un utilisateur sa communauté dans un espace de communautés problématique à partir de ses communautés fiables dans d'autres espaces. Les espaces problématiques sont souvent ceux qui sont relatifs à des critères complexes tels que les centres d'intérêt ou l'historique des évaluations. Cette possibilité permet d'une part d'aider le système à pallier les situations difficiles telles que le démarrage à froid ou l'évolution du besoin en information des utilisateurs, et d'autre part de diminuer l'effort demandé aux utilisateurs pour fournir les informations nécessaires à leur positionnement au sein des espaces de communautés.

Nous rappelons que dans notre modèle des espaces de communautés, un utilisateur u est positionné dans divers espaces de communautés, chaque espace Ω_a étant associé à un critère $a \in A$ (Age, Profession, Contenu, Evaluation, etc.), et reflétant un facteur possible de rapprochement entre les utilisateurs (même tranche d'âge, même catégorie socioprofessionnelle, proximité des thèmes correspondant aux centres d'intérêt, similarité dans l'évaluation des documents, etc.). Dans chacun de ces espaces de communautés Ω_a, l'utilisateur appartient à une communauté particulière Ga_k.

Typiquement, le système de filtrage peut s'appuyer sur chacune de ces communautés pour produire un flux de recommandations selon l'approche collaborative. Etant donnée la table de communautés T (voir Tableau 8.1) construite à partir des espaces de communautés, le vecteur de

positionnement $\mathcal{P}_u$ de l'utilisateur *u* est composé des positions de cet utilisateur dans les divers espaces de communautés Ω_a, et chacune de ces positions constitue une source de recommandation potentielle.

Pendant son cycle de vie, le vecteur de positionnement $\mathcal{P}_u$ d'un utilisateur *u* est souvent imparfait à cause des difficultés rencontrées dans la formation « directe » des espaces de communautés : manque de données dans les profils et/ou complexité dans les calculs. Le système doit alors corriger ce vecteur en introduisant des valeurs estimées là où les valeurs sont manquantes ou douteuses. Ces valeurs sont estimées grâce à un processus de classification par règles. Comme le montre la Figure 12.1, cette tâche de correction consiste à construire les classificateurs à partir de la table de communautés (section 12.1), identifier les communautés ou critères problématiques selon la situation rencontrée (section 12.2), et corriger les communautés dans le vecteur $\mathcal{P}_u$ (section 12.3).

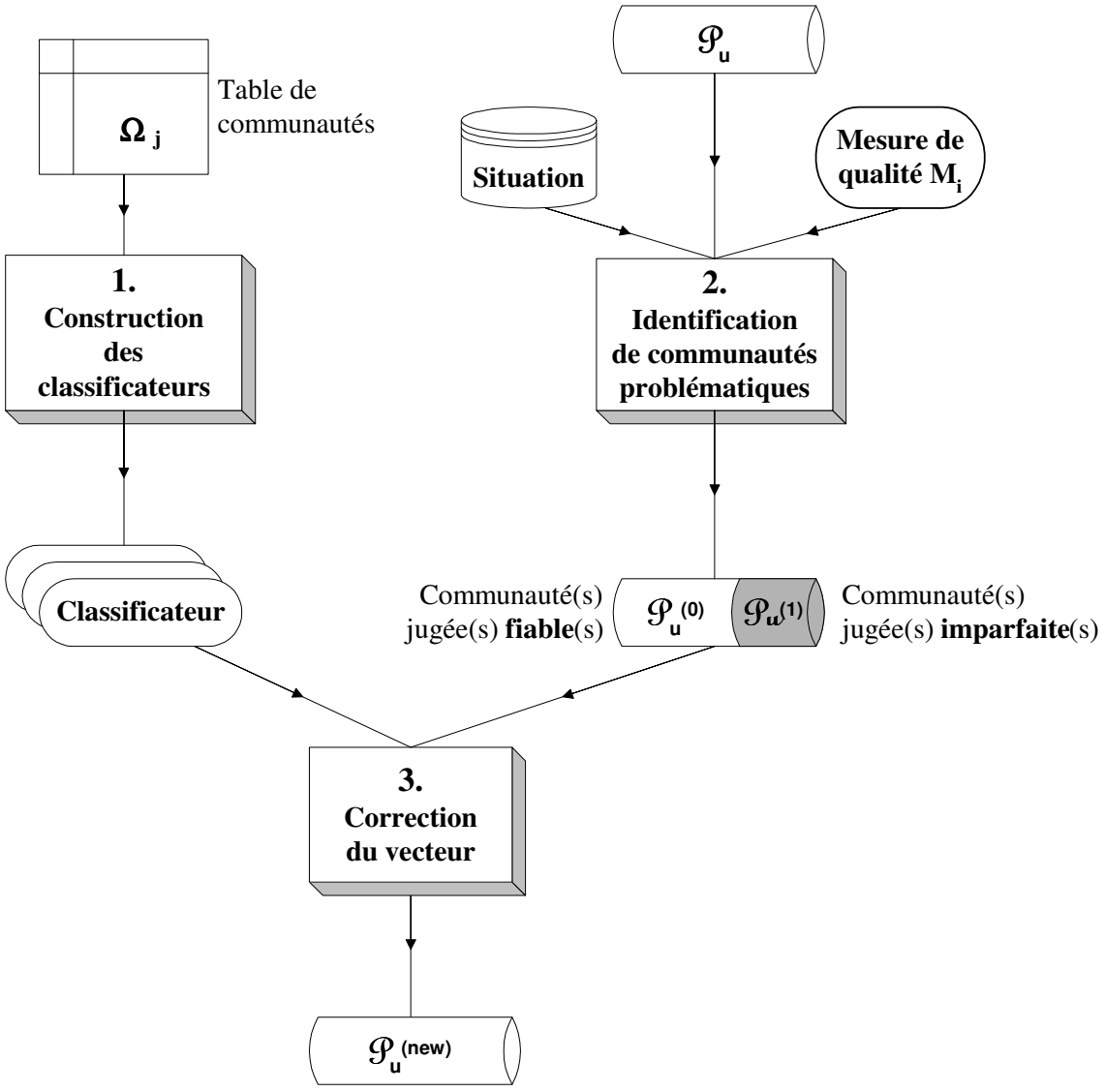


Figure 12.1 – *Schéma d'induction des communautés*.

12.1 Construction de classificateurs

Puisque toute communauté dans un vecteur de positionnement peut être mise en cause pendant son cycle de vie, nous avons en principe besoin d'autant de classificateurs qu'il y a de critères de formation des communautés. Afin de construire les classificateurs nécessaires, nous nous intéressons aux méthodes de classification par règles, en raison de la compatibilité avec notre modèle des espaces de communautés basé sur la théorie des ensembles d'approximation.

En pratique, la seule utilisation des classificateurs issus des règles certaines en se basant sur la théorie des ensembles d'approximation, limite la performance du système en particulier dans le cas où les tables de communautés n'ont pas de hautes consistantes. Dans ce cas, les classificateurs comportent peu de règles certaines, et par conséquent les communautés imparfaites dans un vecteur de positionnement ne sont éventuellement pas complètement corrigées. Ainsi, l'intégration de plusieurs méthodes de classification par règles compatibles rend plus efficace le module d'induction des communautés. Cette possibilité d'intégration est montrée dans 8.5.1.

Pour la construction des classificateurs, nous présentons donc l'utilisation de la théorie des ensembles d'approximation ainsi qu'une approche alternative très connue : les arbres de décision [Mit97]. Nous évitons les approches compliquées telles que réseaux neuronaux [Heb49, PMc43], machines à vecteurs de supports (*Support Vector Machine – SVM*) [Vap82, 95], etc. dans l'optique d'une interaction avec les utilisateurs incluant l'explication des recommandations, qui joue un rôle très important dans la qualité des évaluations des utilisateurs [Her00, HKR00].

12.1.1 Classificateurs par réductions et des ensembles d'approximation

Le principe de la classification supervisée est de construire un classificateur à partir d'un ensemble d'apprentissage afin de prédire la meilleure classe pour une nouvelle observation. Suite à ce principe, on peut construire un classificateur qui contient toutes les règles applicables extraites d'une table de communautés étant donnés l'attribut de décision D et les attributs de condition C.

La théorie des ensembles d'approximation vise à découvrir la dépendance entre les attributs et identifier les attributs de condition indispensables dans l'induction. On exploite les réductions, qui sont les ensembles minimaux d'attributs de condition indispensables, pour préserver les régions positives, dans le but de réduire la table de décision, et par conséquent la taille des prémisses des règles de classification. Alors, la construction du classificateur commence par le calcul des réductions et se termine par la génération des règles à partir de la table réduite.

Pour calculer les réductions, le traitement consécutif de $(2^{|C|} - 1)$ sous-ensembles non vides de l'ensemble des critères C est irréaliste. Théoriquement, les réductions peuvent être calculées en utilisant la matrice d'indiscernabilité [SR92] dont la complexité est en $O(|U|.|C|^2)$. De plus, le calcul des réductions de taille k et de taille minimale sont respectivement des problèmes NP-complet et NP-difficile [NN96]. On peut ainsi trouver dans la littérature plusieurs méthodes heuristiques efficaces pour le calcul des réductions et les méthodes de sélection de caractéristiques (*Feature Selection*). Pourtant, nous ne nous intéressons pas à cette dernière famille de méthodes qui est préférée dans les domaines où il y a un grand nombre d'attributs, de plusieurs dizaines à plusieurs milliers, comme dans

la fouille de données, la reconnaissance des formes, le traitement du signal, etc. En principe, les méthodes heuristiques remplacent le traitement exhaustif de tous les sous-ensembles de *C* par une stratégie aléatoire ou heuristique [JS03, ZY04].

Une fois les réductions créées, les règles formant le classificateur sont identifiées dans la région positive concernée, et elles sont prêtes à servir pour l'induction des communautés.

Il se peut que la table de communautés d'apprentissage ne soit pas complète lors de la construction du classificateur. Il existe plusieurs solutions pour régler ce problème de valeurs manquantes dans l'ensemble d'apprentissage :

i) enlever tous les exemples contenant au moins une valeur manquante,

ii) considérer la valeur manquante dans l'exemple comme une valeur spéciale de l'attribut concerné,

iii) remplacer la valeur manquante dans l'exemple par la valeur la plus fréquente de l'attribut [CN89],

iv) remplacer la valeur manquante par la valeur dominante dans le concept de l'exemple en question [KBR84],

v) remplacer l'exemple contenant la valeur manquante par un ensemble de nouveaux exemples « virtuels » prenant toutes les valeurs possibles dans le domaine de l'attribut [Grz97],

vi) remplacer l'exemple contenant la valeur manquante par un ensemble de nouveaux exemples « virtuels » prenant toutes les valeurs présentes dans les exemples de même concept que l'exemple en question.

Une autre approche que nous mettons en perspective est d'utiliser des *relations caractéristiques*, qui sont une généralisation des relations d'indiscernabilité, afin de définir des ensembles d'approximation à partir de tables de décision incomplètes [Grz05].

12.1.2 Arbre de décision

Un arbre de décision, qui est une méthode de classification supervisée, a pour but de classifier les objets par une division hiérarchique en sous-ensembles [Mit97]. Il y a deux types de nœuds dans un tel arbre :

- les feuilles sont les étiquettes des classes, et
- les nœuds non feuilles y compris la racine sont les attributs de test.

Par ailleurs, chaque branche partant d'un nœud correspond à une valeur possible du nœud (voir Figure 12.2).

Pour construire un arbre de décision par apprentissage à partir d'un ensemble d'exemples, on cherche à chaque pas l'attribut le plus discriminant pour les exemples en utilisant une fonction de

qualité. Une fois l'attribut le plus discriminant choisi, si les exemples concernés sont dans une même classe, le nœud contextuel devient une feuille. Sinon, l'algorithme itère jusqu'à ce qu'un des tests d'arrêt suivants soit satisfait :

i) tous les exemples restants sont dans une même classe, ou

ii) il n'y a plus d'attribut, ou

iii) il n'y a plus d'exemple dans l'ensemble d'apprentissage.

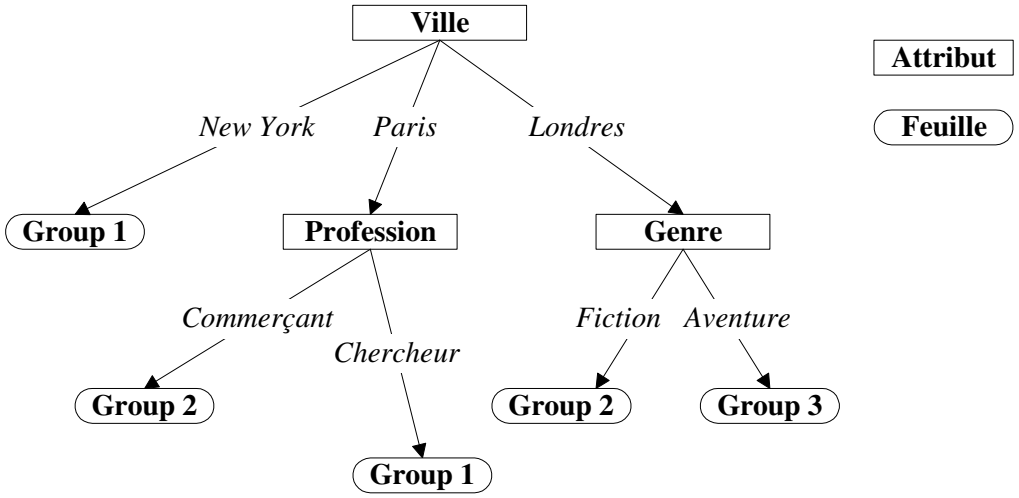


Figure 12.2 – *Exemple d'un arbre de décision (D = {Evaluation}).*

ID3 est l'algorithme le plus connu pour construire un arbre de décision [Qui86]. Dans cet algorithme, le choix de l'attribut le plus discriminant à chaque pas s'appuie sur l'entropie d'information [Sha48].

Soient l'attribut a avec le domaine $V_a = \{va_i\}$ et l'ensemble des exemples X. Alors, l'entropie de l'ensemble X par rapport à l'attribut a est mesurée par :

$$Entropie(X, a) = -\sum_{i \in I} |X_i| . \log(|X_i|)$$

où $\{X_i , i \in I\}$ est l'ensemble des exemples de X qui prennent va_i pour valeur pour l'attribut a.

Ensuite, on définit le gain d'information de X par rapport à l'attribut a comme suit.

$$Gain(X, a) = Entropie(X, a) - \sum_{i \in I} \frac{|X_i|}{|X|} . Entropie(X_i, a)$$

L'algorithme C4.5 [Qui93] est une amélioration de ID3 qui introduit une phase d'élagage. En utilisant une heuristique basée sur les taux d'erreurs, qui sont la proportion des exemples mal classés, cet algorithme essaie de remplacer un sous arbre par une feuille particulière ou de fusionner deux nœuds en un seul nœud. Néanmoins, John et al. ont montré que la performance de la méthode C4.5 est rigoureusement dégradée en présence des attributs dispensables [JKP94], ce que la théorie des ensembles d'approximation peut éviter grâce à des réductions de l'ensemble des attributs de condition *C*.

Nous retenons que la construction des arbres de décision peut se réaliser « off line », et la période de reconstruction des arbres dépend des systèmes applicatifs.

Le principe de classification d'une nouvelle observation par les arbres de décision est que l'on commence les tests par la racine, et que l'on suit ensuite le chemin, jusqu'à ce que l'on atteigne une feuille particulière, et cette feuille est désignée comme la classe prédite pour la nouvelle observation.

Il faut noter que l'on peut considérer les arbres de décision comme une méthode de classification par règles, car on peut facilement transformer un arbre de décision en un ensemble de règles. En effet, on prend chaque chemin reliant la racine et une feuille en combinant les tests des attributs dans le chemin pour générer une règle de classification. Par exemple, l'arbre de décision dans la Figure 12.2 nous donne les règles suivantes :

(Ville = « New York ») → (Evaluation = « Groupe 1 »)

(Ville = « Paris », Profession = « Commerçant ») → (Evaluation = « Groupe 2 »)

(Ville = « Paris », Profession = « Chercheur ») → (Evaluation = « Groupe 1 »)

(Ville = « Londres », Genre = « Fiction ») → (Evaluation = « Groupe 2 »)

(Ville = « Londres », Genre = « Aventure ») → (Evaluation = « Groupe 3 »)

Pour conclure, nous citons les différences importantes entre la théorie des ensembles d'approximation et d'autres approches de classification par règles pour un guidage du choix des méthodes pour construire des classificateurs.

(i) Au niveau de la qualité du classificateur généré par une méthode de classification par règles autre que la théorie des ensembles d'approximation, la compacité, ou le nombre de règles recouvrant l'ensemble des exemples, est le critère plus important que le nombre des attributs présents dans les règles, bien qu'un nombre faible d'attributs soit toujours souhaitable.

(ii) Les méthodes de classification par règles, autres que la théorie des ensembles d'approximation, ne focalisent pas sur la structure de la table de décision, les relations entre attributs et entre règles, à l'exception du support de règle, mais qui ne le fait qu'au niveau des données. Elles prennent en compte essentiellement les conditions et les décisions, par exemple (Profession = « Chercheur »), (Genre = « Documentaire »), etc., plutôt que les attributs de condition et de décision, par exemple Profession, Genre. Au contraire, la théorie des ensembles d'approximation met en évidence les relations entre les attributs et les classes d'équivalence, et les relations entre les règles.

12.2 Identification des communautés problématiques

Cette étape est déclenchée chaque fois que le système ou un utilisateur rencontre une situation difficile. On peut déjà citer deux situations difficiles courantes que nous abordons tout au long du manuscrit : le démarrage à froid et l'évolution du besoin en information.

Les données d'entrée de ce module sont généralement le profil de l'utilisateur $\mathcal{P}_u$, la base de connaissances sur les situations et les mesures de qualité des critères dans l'induction des communautés (voir Figure 12.1). Le résultat est donc l'ensemble des critères ou communautés problématiques dans le profil de l'utilisateur, ordonné conformément à l'ordre d'induction souhaitable.

D'abord, la base de connaissances est utilisée pour présélectionner les communautés qui causent des difficultés. Ce sont par exemple les communautés manquantes dans le profil initial d'un nouvel utilisateur. Au cas où le système détecte une accumulation d'évaluations négatives de l'utilisateur existant, qui traduit apparemment son insatisfaction quant aux recommandations reçues, la base de connaissances peut aider le système à identifier les communautés de l'utilisateur qui ont généré ces mauvaises recommandations.

Ensuite, le système utilise les mesures de qualité des critères devant servir d'attribut de décision dans l'induction pour raffiner l'ensemble des critères présélectionnés en éliminant les communautés des critères dont la qualité d'induction par les classificateurs n'est pas garantie. Par exemple, le système ne prend en compte que les communautés relatives aux critères a_j dont la valeur $\varphi(D = \{a_j\})$ dépasse un seuil donné (cf. (8.15)).

En général, le choix des mesures pour l'identification des communautés dépend de la situation rencontrée.

Finalement, les communautés restantes sont triées selon ces mesures, et on obtient l'ensemble ordonné des communautés à corriger dans $\mathcal{P}_u$. Il est à remarquer que l'ordre des critères ne dépend que de la situation et pas des cas particuliers d'utilisateurs.

12.3 Correction du vecteur de positionnement

Une fois créé l'ensemble ordonné des communautés problématiques, les classificateurs correspondants sont utilisés pour la correction du vecteur de positionnement imparfait de l'utilisateur.

La correction d'une communauté s'appuie normalement sur les principes de base de la classification par règles [Grz97, KPS98] :

i) Le système cherche une règle applicable dans le classificateur.

ii) S'il n'existe aucune règle applicable, la communauté est remplacée par une valeur par défaut. Certaines approches présentées dans 12.1.1 peuvent être utilisées pour définir les valeurs par défaut.

iii) S'il existe plusieurs règles applicables, il faut résoudre le conflit éventuel entre la décision de ces règles. En général, le choix de la règle se réalise en fonction de la qualité de chaque règle dans le classificateur.

Nous voulons remarquer ici la différence éventuelle entre la table de communautés et les profils d'utilisateurs. Au début, la table de communautés est construite à partir des profils d'utilisateurs. Pourtant, au fur et à mesure que le système met à jour les vecteurs de positionnement d'utilisateurs par le module d'induction des communautés, la distance entre les profils originaux et la table de communautés augmente. Par exemple, un utilisateur déclare à l'inscription qu'il aime bien les films documentaires, et il est par conséquent positionné au début dans la communauté correspondant à ce genre de film. Après un certain temps, cet utilisateur pourrait être rattaché à la communauté des amateurs de films scientifiques par induction à partir de ses communautés dans les autres espaces quoique son profil enregistre toujours le genre de film documentaire.

Dans les perspectives de cette thèse, nous envisageons d'étudier l'adaptation du profil d'un utilisateur en exploitant la possibilité d'hériter les profils typiques de ses communautés.

Chapitre 13

Filtrage d'information

Le module de filtrage d'information a pour objectif de générer des recommandations pour les utilisateurs en fonction de leurs communautés multicritères. La prédiction de la satisfaction d'un utilisateur pour un document particulier dans un système de filtrage collaboratif classique est calculée par la proximité des évaluations en utilisant la corrélation de Pearson. Les difficultés de cette tâche apparaissent dans le démarrage à froid où le nouvel utilisateur n'a pas encore d'évaluations, ou dans le cas d'un faible nombre d'évaluations en commun entre les utilisateurs. De plus, en raison de la précision et la performance, le système traite notamment les voisins les plus proches de l'utilisateur, en fonction d'un nombre fixe (*K* meilleurs) [RIS+94] ou d'un seuil de diamètre [BHK+98, SP90]. Toutes ces deux approches s'appuient sur une distance, et elles ne sont pas applicables pour les critères où la distance entre les utilisateurs dans une communauté est identique, par exemple communautés des chercheurs, des Parisiens, etc.

De ce fait, nous proposons une nouvelle méthode supplémentaire pour générer, pour un utilisateur, des recommandations par niveau d'accord au sein de ses communautés. Ces recommandations peuvent constituer directement les suggestions finales, ou être considérées comme la matière alimentant un processus de filtrage hybride afin d'obtenir un ensemble de recommandations plus affiné.

13.1 Filtrage collaboratif par « niveau d'accord »

Notre proposition est une méthode ad hoc pour produire des recommandations en fonction du « niveau d'accord » au sein des communautés concernées. Le principe de cette méthode de production de recommandations repose sur une forme de « quasi-unanimité » de la communauté sur la qualité d'un document. En général, la quasi-unanimité, ou *unanimité* plus court, de la communauté peut être représentée par deux seuils : S_{accord} pour le nombre d'évaluations et S_{score} pour le score moyen de la communauté sur un document particulier.

Considérons un utilisateur u_0 appartenant à la communauté G dans un espace de communautés donné. Alors, cet utilisateur reçoit tous les documents jugés intéressants de façon *unanime* par la communauté G. Le niveau d'accord au sein de la communauté est alors maximal, puisqu'il y a *unanimité*. Pour chaque critère, c'est-à-dire pour chaque espace de communautés, la méthode doit sélectionner les documents à recommander parmi ceux qui ont été évalués par la communauté G dans laquelle l'utilisateur u_0 se situe. Alors, le document d est suggéré à l'utilisateur u_0 s'il vérifie les deux conditions suivantes :

$$\frac{1}{|G|}\sum_{u\in G} v_{u,d} \geq S_{score} \tag{13.1}$$

$$|\{\, v_{u,d} \mid (u \in G) \text{ et } (v_{u,d} \neq null)\}| \geq S_{accord} \tag{13.2}$$

où $v_{u,d}$: score donné par l'utilisateur u sur le document d

Selon la condition (13.1), le système procède à un premier filtre en ne considérant que les documents évalués par la communauté G avec un score moyen supérieur ou égal au seuil S_{score}. Enfin, le système réalise un deuxième filtre, suite à la condition (13.2), où on ne conserve que les documents qui ont été ainsi évalués par une part suffisante des membres de la communauté G, dont la limite est fixée par un autre seuil S_{accord}.

Notre méthode ad hoc de filtrage par niveau d'accord permet d'une part la diversité de recommandations par des communautés multicritères et d'autre part de pallier les situations délicates comme le démarrage à froid. Le principe général de la méthode peut être décliné de diverses manières, par exemple en limitant le « niveau d'accord » à un sous-ensemble de la communauté comme par exemple à ses membres les plus représentatifs, ou aux voisins les plus proches comme dans l'approche classique.

Les recommandations issues des communautés multicritères peuvent être combinées par des méthodes d'hybridation de filtrage.

13.2 Hybridation de filtrage

En général, l'hybridation dans un système de filtrage se réalise en deux phases : *préliminaire* et *couplage*, comme illustré dans la Figure 13.1. D'abord, le système applique à part une ou plusieurs

techniques de filtrage de base telles que filtrage collaboratif, filtrage basé sur le contenu et filtrage démographique. Cette phase préliminaire génère les recommandations candidates, et qui sont combinées par certaines méthodes d'hybridation dans la phase de couplage, afin de produire les recommandations finales destinées à l'utilisateur.

Pour la phase de couplage, Burke a passé en revue sept méthodes habituelles d'hybridation : pondération, mixtion, cascade, commutation, augmentation de traits, combinaison de traits et méta modèle [Bur02]. Suite à ces méthodes, l'hybridation de filtrage peut être une combinaison des résultats des filtrages réalisés dans la phase préliminaire ou une application séquentielle de filtrages de plus en plus spécifiques. Nous constatons, à travers cette revue, que dans la majorité des systèmes hybrides, les travaux essaient de combiner le filtrage collaboratif avec une autre technique de filtrage.

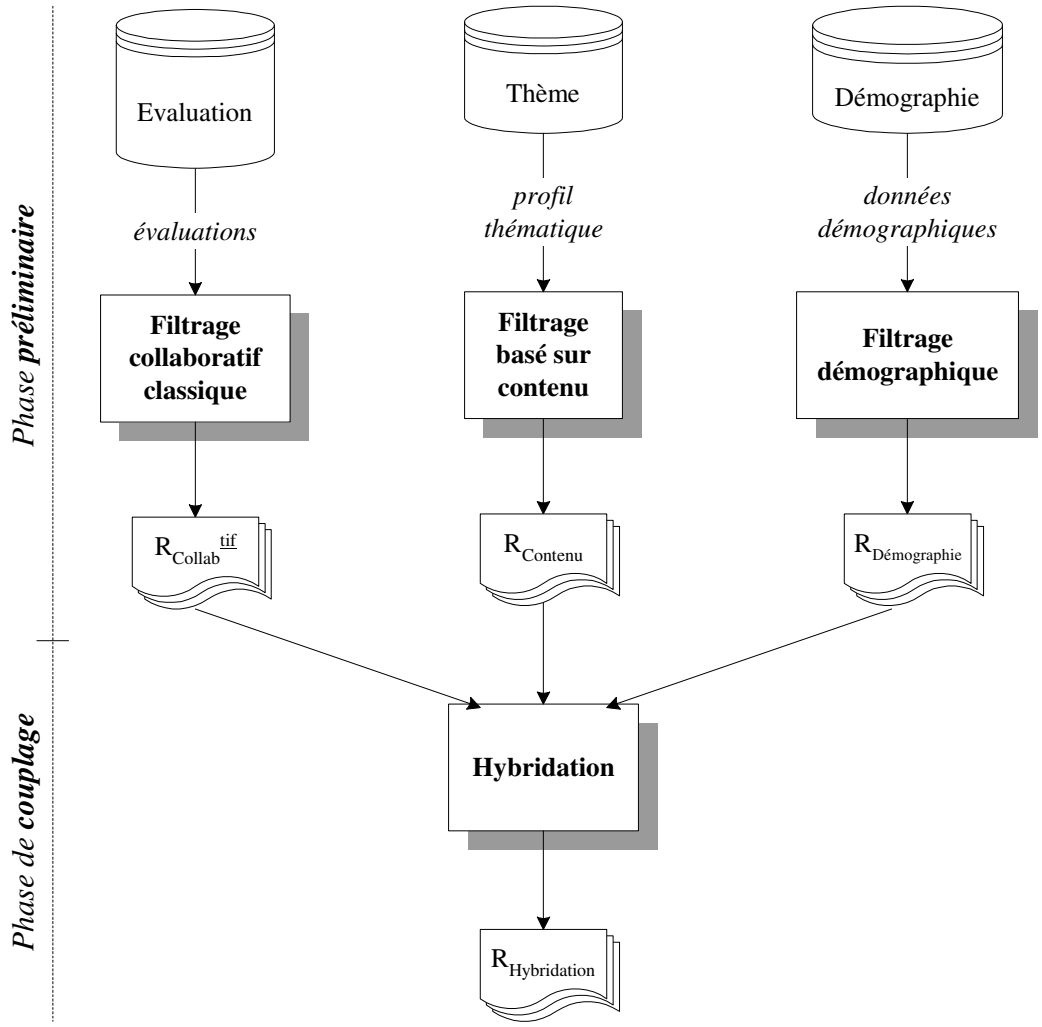


Figure 13.1 – *Hybridation classique de filtrages*.

Dans notre plateforme COCoFil2, nous améliorons le schéma d'hybridation classique en introduisant la multiplicité des critères et le filtrage collaboratif par niveau d'accord dans la phase

préliminaire, et nous choisissons la méthode de pondération pour l'hybridation. La prédiction finale pour un utilisateur *u* sur le document *d* est une pondération des prédictions produites par les techniques effectuées dans la phase précédente :

$$prédiction_{finale}(u,d) = \sum_i w_i . prédiction_i(u,d) \tag{13.30}$$

Les prédictions composante $prédiction_i(u,d)$ sont éventuellement calculées par le filtrage basé sur le contenu, le filtrage collaboratif classique sur le critère Evaluation, ou le filtrage collaboratif par niveau d'accord sur les autres critères disponibles. Pour calibrer les paramètres w_i, nous adoptons l'apprentissage proposé par Claypool et al. [CGM+99]. Au début, les poids w_i ci-dessus sont égaux, et le processus d'apprentissage les ajuste en pénalisant la mauvaise composante de la prédiction au fur et à mesure que l'utilisateur évalue les recommandations. Cette technique de pondération est très simple et insensible à l'ordre d'application des filtrages. Par ailleurs, elle permet d'ajuster rapidement l'effet des filtrages à travers l'utilisation des valeurs spécifiques des poids w_i selon la situation rencontrée. Par exemple, le système peut utiliser en même temps une autre stratégie des poids dédiée au démarrage à froid, dont le poids invariant pour le filtrage collaboratif est plus faible que celui pour le filtrage basé sur le contenu en raison de faibles nombre et confiance des premières évaluations de nouveaux utilisateurs.

En général, les sept méthodes d'hybridation mentionnées plus haut peuvent être divisées en deux catégories en fonction du niveau de couplage : fort et faible. La catégorie de couplage fort des filtrages comprend les deux dernières méthodes : combinaison de traits et méta modèle. Dans ces méthodes, le filtrage utilisé dans la seconde phase d'hybridation doit être capable de comprendre le fonctionnement interne de celui (ou ceux) appliqué(s) dans la phase préliminaire afin de réaliser une intégration solide dans un système hybride. Par contre, dans les méthodes de couplage faible, incluant la pondération, on limite l'interface entre les techniques préliminaires via une exploitation simple de leurs résultats. Chacune n'a donc pas besoin de comprendre le fonctionnement des autres. Ceci assure la simplicité de l'implémentation d'un système ouvert, grâce à la facilité d'intégration de nouvelles techniques de filtrage.

13.3 Diversification de recommandations

Vu le contexte du positionnement multiple des utilisateurs dans notre plateforme COCoFil2, la diversité des recommandations doit être prise en considération afin d'une part d'améliorer la qualité de la liste des recommandations pour un utilisateur, et d'autre part de lui permettre de découvrir de nouveaux domaines potentiellement intéressants.

Pour la diversité d'une liste de recommandations en termes de contenu, Ziegler et al. reposent sur la taxonomie de thèmes proposée par Amazon [ZMK+05]. Il faut remarquer que l'on ne dispose pas toujours de telles taxonomies. Ainsi, nous proposons une définition de la diversité en termes de contenu d'une liste de recommandations.

Soient la liste de recommandations L et l'utilisateur u dont le vecteur de positionnement est :

$$\mathcal{P}_u = (G_1, G_2, ..., G_n)$$

où $A = \{Evaluation, a_2, \ldots, a_n\}$

$G_1 \in \Omega_{\text{Evaluation}}$

$G_2 \in \Omega a_2$

…

$G_n \in \Omega a_n$

La diversité de la liste L vis-à-vis de l'utilisateur u est liée aux conditions suivantes.

i) Les recommandations dans la liste L sont *pertinentes* pour l'utilisateur u. Cela se traduit en une petite distance entre cette liste et le profil d'évaluations de sa communauté G_1 dans l'espace $\Omega_{Evaluation}$:

$$\text{distance}_{Evaluation}(L, profil_{Evaluation}(G_1)) \leq \varepsilon \tag{13.4}$$

Nous nous basons sur l'hypothèse d'une forte similarité entre l'historique des évaluations, ou profil Evaluation, de l'utilisateur u et celui des membres dans sa communauté G_1.

ii) Les recommandations dans la liste L sont relativement *éloignées du profil de contenu* de l'utilisateur :

$$\text{distance}_{Contenu}(L, profil_{Conenu}(u)) > \delta \tag{13.5}$$

Afin de définir les paramètres ε et δ, nous envisageons trois possibilités. Premièrement, ces deux paramètres sont communs à tous les utilisateurs du système. Cette possibilité nécessite un processus d'apprentissage pour calibrer les paramètres ε et δ. Une alternative est que le système attribue des valeurs différentes de paramètres pour chaque communauté. Et, la dernière possibilité la plus compliquée est la personnalisation de ces paramètres, c'est-à-dire que chaque utilisateur possède ses propres valeurs de paramètres, ε_u et δ_u.

Il est à noter que notre approche de diversité de recommandations peut en principe être étendue aux autres critères que le contenu en remplaçant ce dernier par le critère concerné dans la formule (13.5).

Chapitre 14

Bilan

Dans cette partie, nous avons présenté une nouvelle plateforme de filtrage collaboratif COCoFIl2 basée sur le modèle des espaces de communautés. Dans cette plateforme, les problèmes de la formation monocritère des communautés, de la perception limitée des communautés et du démarrage à froid sont traités.

Notre approche des cartes de communautés, qui exploite un algorithme de positionnement en 2 dimensions et l'algorithme classique de K-moyennes afin d'obtenir des cartes de communautés, rend complètement explicites les communautés dans le système. Ces cartes s'appuient sur divers critères différents de formation des communautés. Grâce au positionnement multiple où un utilisateur peut appartenir à la fois à plusieurs communautés, lorsqu'il rencontre des difficultés dans une communauté particulière, l'utilisateur peut abandonner temporairement cette communauté problématique, et s'en remettre à ses autres communautés. De plus, nous pensons que les diverses formations de communautés, et les éventuels contrastes qu'elles sont susceptibles de faire surgir, peuvent servir de point de départ à l'adaptation du profil de l'utilisateur.

Par ailleurs, le module d'induction vise à dépasser le démarrage à froid à l'inscription ainsi que l'évolution du besoin en information pendant le temps de l'exploitation grâce au mécanisme d'induction des communautés qui se base sur le modèle des espaces de communautés. Nous cherchons à découvrir la relation entre les communautés de l'utilisateur. Lors de l'apparition de difficultés dans la détermination d'une des communautés pour l'utilisateur, le système peut utiliser ses autres communautés pour induire la communauté problématique. Ainsi, notre plateforme offre des moyens supplémentaires prenant en compte un cadre plus large (communautés multicritères) et des possibilités de positionner des utilisateurs se trouvant dans des situations problématiques.

Enfin, dans notre plateforme COCoFil2, nous proposons une nouvelle méthode de filtrage collaboratif par niveau d'accord pour les communautés multicritères, qui enrichit l'hybridation des méthodes de filtrage. Grâce à notre nouvelle méthode, l'hybridation se réalise non seulement au niveau de la combinaison du filtrage collaboratif avec d'autres techniques dans la phase de couplage [Bur02], mais également au niveau de communautés multicritères du filtrage collaboratif dans la phase préliminaire. Ceci permet de rendre plus diversifiées les recommandations envoyées aux utilisateurs.

Partie V.

Validation

Après avoir présenté dans les parties précédentes le modèle des espaces de communautés pour la gestion des communautés, ainsi que sa mise en œuvre avec la plateforme de filtrage collaboratif COCoFil2, cette partie est consacrée notamment à la validation de ses aspects essentiels sur un jeu de données réelles.

La Figure V.1 illustre notre schéma d'expérimentation, incluant la préparation de données et les trois aspects de validation : construction et comparaison des cartes de communautés, validation des mesures de qualité de critère et démarrage à froid.

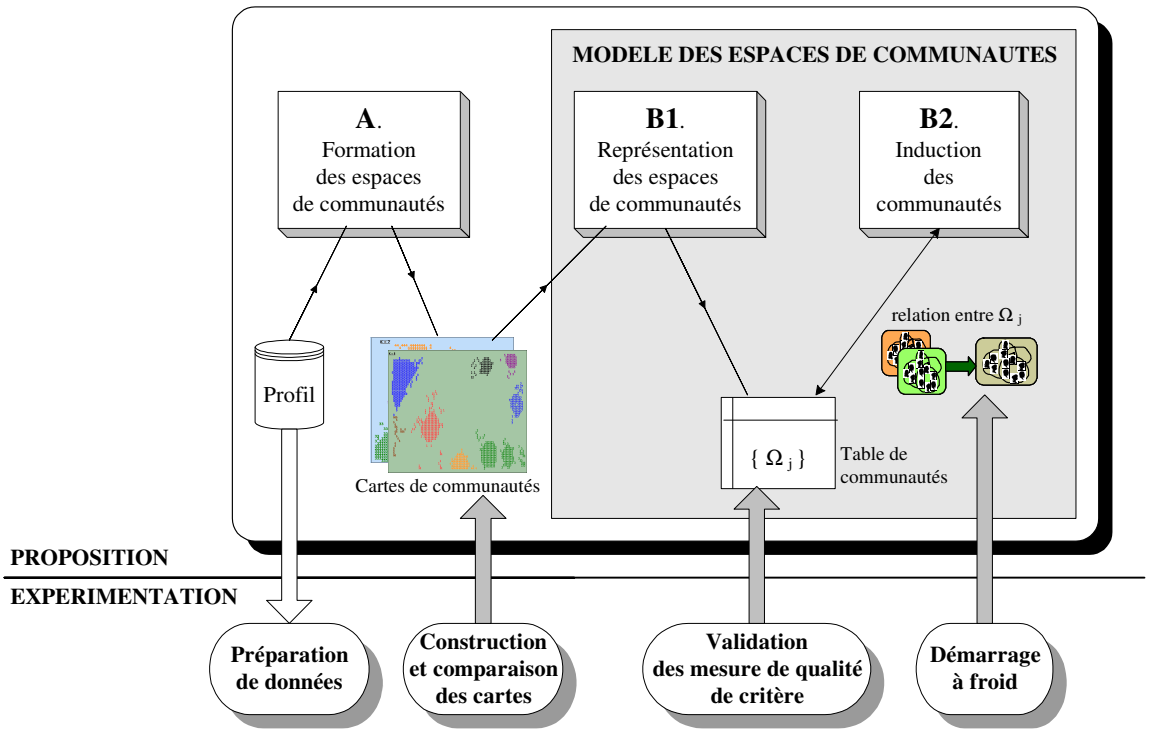


Figure V.1 – *Schéma d'expérimentation.*

D'abord, la préparation des données pour l'expérimentation est décrite en détails dans le Chapitre 15.

Concernant la formation des espace de communautés, le Chapitre 16 décrit les travaux pour analyser la performance de notre méthode de construction des divers espaces de communautés en vue de remplir la table de communautés, en particulier pour les deux critères Contenu et Evaluation, qui sont à la fois complexes et capitaux dans les systèmes de filtrage. Ces travaux répondent à la question : « Est-ce que les espaces de communautés des critères sont véritablement différents ? », en espérant que leur différence pourrait être utile dans certaines activités du système de filtrage

Les travaux présentés dans le Chapitre 17 visent à valider les mesures de qualité de critère en tant qu'attribut de décision dans l'induction des communautés. A travers les résultats de cette validation, nous voulons montrer que l'on peut utiliser les mesures proposées dans le modèle pour sélectionner et trier les critères problématiques dans un vecteur de positionnement imparfait pour la tâche d'induction des communautés. Ces travaux sont indispensables si l'on souhaite une qualité de base de la correction de tels vecteurs.

Enfin, la motivation des travaux présentés dans le Chapitre 18 est de valider notre méthode basée sur le modèle des espaces de communautés pour le problème d'intégration de nouveaux utilisateurs dans un système de filtrage collaboratif. Cette méthode d'induction des communautés peut en principe s'effectuer également pour l'évolution éventuelle du besoin en information d'utilisateurs au cours du temps de l'exploitation. Pourtant, nous faisons figurer la validation de cette possibilité dans les perspectives, car en raison que la performance de ce processus dépend non seulement de notre méthode d'induction mais aussi de la méthode de détection du changement de l'intérêt des utilisateurs. Cette dernière repose à son tour sur une basse de connaissances sur les situations problématiques possibles, qui ne constitue pas l'objectif premier de la thèse.

Chapitre 15

Préparation des données

Dans ce chapitre, nous décrivons essentiellement la préparation des données que nous utilisons dans notre expérimentation. D'abord, nous donnons quelques chiffres décrivant ces données, puis nous montrons comment définir les critères de formation des communautés. Finalement, l'extraction des valeurs de critères des utilisateurs à partir des données est également décrite.

15.1 Jeu de données MovieLens

Pour expérimenter divers aspects du modèle des espaces de communautés proposé dans cette thèse, nous utilisons un des deux jeux de données réelles du système de recommandation de films MovieLens[8]. Ces jeux de données sont très populaires et utilisés dans beaucoup d'études du domaine de filtrage collaboratif [HKR00, RAC+02, SPU01, etc.]. Le site Web MovieLens est développé par le groupe de recherche GroupLens[9] à l'université de Minnesota, Etats-Unis. On dispose sur ce site de deux jeux de données d'évaluations de films de tailles différentes. Le premier jeu, le plus grand[10], comprend 1 000 000 évaluations, de 1 à 5 étoiles, faites par environ 6000 utilisateurs, et le second que nous utilisons dans l'expérimentation comprend 100 000 évaluations[11] fournies par 943 utilisateurs pendant la période du 09/1997 au 04/1998 sur 1 682 films. Sauf précision autre, le nom MovieLens désigne dans la suite ce second jeu de données.

[8] http://movielens.umn.edu/
[9] http://www.grouplens.org/
[10] http://www.grouplens.org/data/million/
[11] http://www.grouplens.org/data/

Nous donnons ici quelques statistiques simples sur le jeu MovieLens pour avoir un premier éclairage sur ces données :

- le nombre d'évaluations faites par chaque personne varie de 20 à 737 ;
- le nombre de films jugés en commun entre utilisateurs est généralement bas, il n'y a que 1,48% d'utilisateurs ont fait en moyen de 31 à 33% d'évaluations en commun avec les autres, et la plupart d'entre eux (41,36%) ont de 16 à 20% d'évaluations en commun avec les autres (voir Figure 15.1) ; et
- nous constatons une proportion faible de scores défavorables (6,11% et 11,37% pour 1 et 2 étoiles respectivement) qui montre la tendance des utilisateurs à n'évaluer que les films qu'ils aiment.

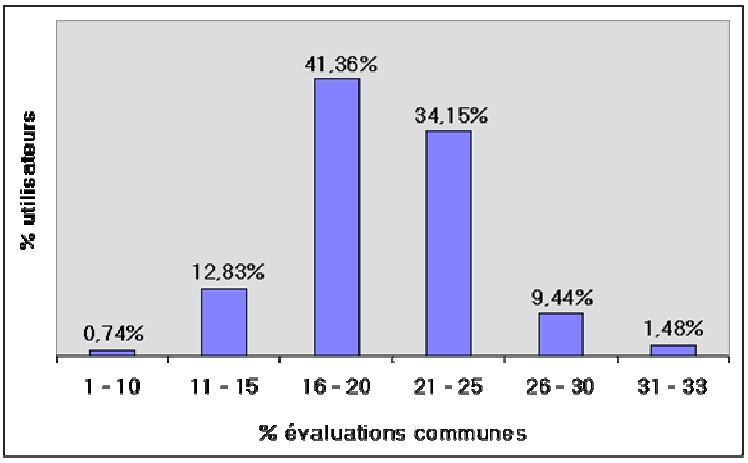


Figure 15.1 – *Histogramme des nombres d'évaluations en commun.*

15.2 Disponibilité des critères de formation des communautés

On peut voir dans la partie de modélisation que la multiplicité des attributs ou critères dans les profils d'utilisateurs joue un rôle fondamental dans notre modèle puisqu'ils permettent de construire différents espaces de communautés. Suivant la catégorisation des critères présentée dans 9.1, nous exploitons trois catégories de critères tirés des données MovieLens pour former les espaces de communautés : critères d'informations personnelles, critère de centres d'intérêts et critères relatifs à l'historique de l'interaction.

15.2.1 Critères d'informations personnelles

Chaque utilisateur du système MovieLens doit déclarer à l'inscription ses informations personnelles : âge, profession et ville de résidence (zip code) aux Etats-Unis. Ces données nous

permettent de définir respectivement les trois critères démographiques suivants : *Age, Profession* et *Géographie*.

15.2.2 Critère de centres d'intérêt

Parmi les informations descriptives de films dans le système MovieLens telles que le titre, l'année de sortie, etc. nous nous intéressons en particulier aux genres auxquels appartiennent les films. Dans ce jeu, un film peut en général appartenir à plusieurs genres, et nous considérons donc ces genres comme décrivant l'aspect de *contenu* du film. Par conséquent, nous pouvons construire pour chaque utilisateur un profil thématique contenant les genres de film qu'il préfère.

15.2.3 Critères relatifs à l'historique de l'interaction

Le premier critère dans cette catégorie est le critère classique Evaluation comme dans tous les systèmes de filtrage collaboratif courants. De plus, grâce aux évaluations d'utilisateurs, nous pouvons définir le critère *Motivation* [Gal05] qui traduit la volonté des utilisateurs à fournir régulièrement des évaluations sur les recommandations reçues, afin que le système puisse réaliser le filtrage collaboratif pour les communautés.

En résumé, chaque utilisateur dans le système de recommandation de films MovieLens est caractérisé par six attributs ou critères que nous pouvons utiliser pour former autant d'espaces de communautés : Age, Profession, Géographie, Contenu, Evaluation et Motivation. Dans cette thèse, nous ne prenons pas en compte les critères composés. Le prétraitement des données MovieLens pour l'extraction des valeurs de critères des utilisateurs est décrit dans la suite.

15.3 Extraction des valeurs de critères

Avant de réaliser l'expérimentation, nous manipulons les données de MovieLens afin d'extraire pour chaque utilisateur les valeurs relatives aux six critères précités.

15.3.1 Critère Age

Les âges des utilisateurs s'échelonnent de 7 à 73 ans, et les utilisateurs sont caractérisés par une des 5 tranches d'âge : moins de 16 ans, 16-25 ans, 26-45 ans, 46-60 ans et plus de 60 ans (voir le pourcentage d'utilisateurs par tranche d'âge dans la Figure 15.2).

15.3.2 Critère Profession

Il y a dans le jeu MovieLens 21 professions et nous regroupons les professions assez proches, comme enseignant, chercheur et étudiant, et formons 7 catégories de profession suivantes :

- enseignant - chercheur - étudiant
- commerçant
- ingénieur, technicien
- artiste, professionnel des loisirs
- santé publique
- retraité, personne au foyer,
- autres professions.

Ainsi, nous attribuons à chaque utilisateur une des 7 catégories ci-dessus selon sa profession déclarée à l'inscription. La Figure 15.3 nous montre le pourcentage d'utilisateurs par profession.

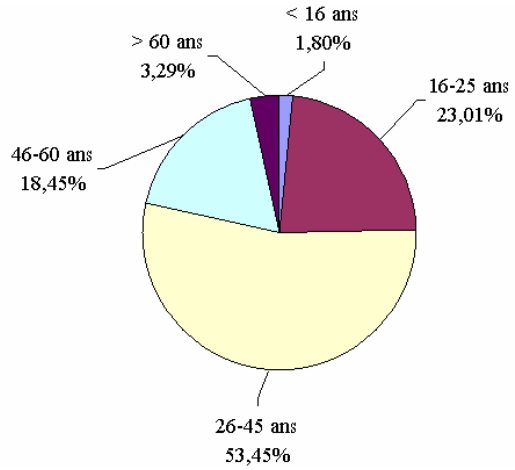


Figure 15.2 – *Pourcentage d'utilisateurs par tranche d'âge*.

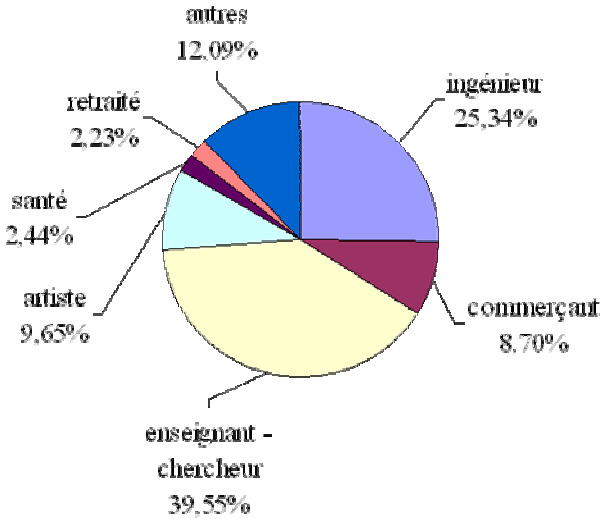


Figure 15.3 – *Pourcentage d'utilisateurs par profession*.

15.3.3 Critère Géographie

Les utilisateurs ont aussi donné leurs codes postaux (zip code) aux Etats-Unis, ce que nous utilisons pour retrouver les états comme étant leur caractéristique géographique. En réalité, 44 états sont présents dans les données.

15.3.4 Critère Contenu

Pour le critère Contenu, la base propose au total 19 genres de film, et chaque film peut être associé à plusieurs genres à la fois. Par exemple, certains films sont associés à 6 genres. Pour chaque utilisateur, nous construisons la partie thématique de son profil, aussi appelée « profil Contenu », comme un vecteur de 19 valeurs (voir Tableau 15.1) qui reflètent le niveau d'intérêt qu'il a manifesté pour les genres de film, en se basant sur le nombre, la moyenne et la variance de ses évaluations concernant chacun de ces genres. Le poids du genre g_j dans le profil de l'utilisateur u est calculé par :

$$poids(u, g_j) = a.\frac{n_{u,j}}{N_u} + b.\frac{\mu_{u,j}}{5} + c.\frac{\upsilon_{u,j}}{5} \tag{15.1}$$

où a, b, c : paramètres

$n_{u,j}$: nombre d'évaluations de l'utilisateur u sur les films du genre g_j

N_u : nombre total d'évaluations de l'utilisateur u

$\mu_{u,j}$: score moyen des évaluations de l'utilisateur u sur les films du genre g_j

$\upsilon_{u,j}$: variance des évaluations de l'utilisateur u sur les films du genre g_j

Les poids des genres dans le profil Contenu de l'utilisateur u doivent être normalisés pour que :

$$\sum_{j=1}^{19} poids(u, g_j) = 1 \tag{15.2}$$

En constant le faible taux de scores défavorables parmi les évaluations des utilisateurs, nous concluons que les utilisateurs n'évaluent que les films qu'ils aiment bien, et que le nombre des évaluations relatives au genre en question joue un rôle important dans la traduction du niveau d'intérêt de l'utilisateur pour ce genre. De ce fait, nous choisissons des valeurs inégales pour les paramètres dans (15.1) en favorisant le paramètre a relatif au nombre d'évaluations : $a = 0{,}5$, $b = 0{,}3$ et $c = 0{,}2$.

15.3.5 Critère Evaluation

Le critère Evaluation est un critère classique s'appuyant sur les évaluations fournies par les utilisateurs comme dans les systèmes de filtrage collaboratif existants. Nous utilisons également le terme « profil Evaluation » pour désigner l'ensemble de toutes les évaluations d'un utilisateur donné.

genre	action	adventure	animation	children	comedy	crime	…	war	western
u_1	9,60	6,00	3,16	3,83	11,32	4,56	…	4,69	2,74
u_2	7,47	4,42	3,21	4,00	10,40	6,97	…	3,96	0,00
u_3	9,58	5,35	0,00	0,00	8,40	7,82	…	5,17	0,00
u_4	11,33	6,78	0,00	0,00	7,84	7,67	…	5,35	0,00
u_5	10,95	7,43	4,84	6,25	14,92	4,13	…	4,44	2,07
u_6	5,76	5,37	3,56	4,83	11,94	4,34	…	5,44	2,94
u_7	9,05	6,43	3,41	4,86	8,52	4,54	…	5,46	3,49
u_8	18,29	9,52	0,00	2,18	5,72	6,62	…	7,59	4,45
u_9	8,87	7,64	0,00	0,00	10,92	0,00	…	9,30	0,00
u_{10}	6,46	4,63	3,33	3,56	10,02	5,05	…	5,67	3,23

Tableau 15.1 – *Extrait des profils Contenu (poids en %).*

15.3.6 Critères Motivation

Nous attribuons à chaque utilisateur une motivation égale à la moyenne mensuelle du nombre d'évaluations depuis son inscription dans le système, traduisant ainsi la tendance des utilisateurs à fournir régulièrement des évaluations sur les recommandations reçues. Nous définissons aussi une échelle de 5 niveaux de motivation : très faible, faible, moyenne, bonne et excellente. Le pourcentage d'utilisateurs par motivation est illustré dans la Figure 15.4.

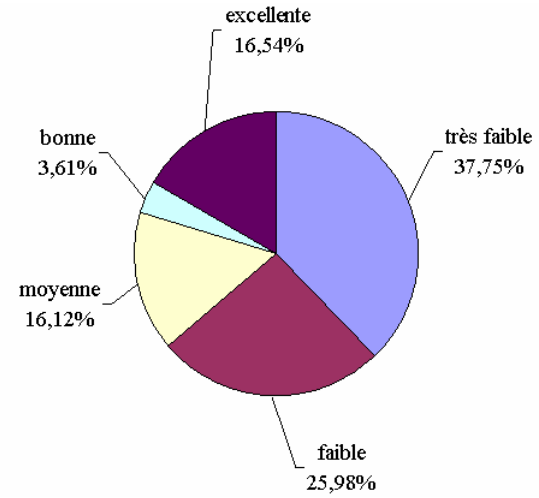


Figure 15.4 – *Pourcentage d'utilisateurs par motivation.*

Les travaux d'expérimentation du modèle proposé sur les données de MovieLens sont présentés dans les chapitres qui suivent.

Chapitre 16

Formation des communautés

16.1 Objectifs

L'objectif principal de cette partie de l'expérimentation est de valider notre approche d'utilisation des cartes de communautés en 2D créées par la combinaison de l'algorithme des fourmis artificielles avec l'algorithme des K-moyennes sur divers critères disponibles dans les profils des utilisateurs, afin de pallier les problèmes de la formation monocritère et de la perception limitée de communautés.

Plus précisément, nous avons d'abord envie de montrer la performance de la création de cartes en vérifiant la faisabilité sur de grandes quantités de données en des temps raisonnables. Ensuite, nous voulons montrer concrètement via les cartes en 2D, les espaces de communautés et en particulier les espaces relatifs aux deux critères complexes Contenu et Evaluation, que le système peut fournir aux utilisateurs comme un moyen de perception des communautés.

Il faut néanmoins noter que nous ne visons pas à valider la dimension interactive. Cela veut dire que nous mettons aux travaux futurs les questions telles que : « Comment les cartes de communautés sont-elles présentées à l'utilisateur ? », « De quelles manières l'utilisateur interagit-il avec le système via les cartes ? », etc.

Enfin, cette première partie d'expérimentation montre également qu'il existe une différence permanente entre les deux espaces $\Omega_{Contenu}$ et $\Omega_{Evaluation}$ dans le système MovieLens. Ceci nous donne par ailleurs une réponse pertinente à la question : « Faut-il mettre en œuvre différentes catégories de

communautés dans un système de filtrage collaboratif ? », puisque cette différence pourrait être utile dans certaines activités du système de filtrage.

16.2 Protocole

Dans la présente section, nous évoquons d'abord les données d'entrées que nous utilisons dans l'expérimentation. Ensuite, nous décrivons comment nous appliquons la méthode proposée dans la section 11.4.3 pour créer les cartes $\Omega_{Contenu}$[12] et $\Omega_{Evaluation}$, et citons les mesures utilisées pour comparer ces deux cartes. Finalement, nous montrons la performance des algorithmes appliqués ainsi que les résultats d'analyse des cartes générées.

16.2.1 Données d'entrée et méthode

Afin de créer les cartes de communautés $\Omega_{Contenu}$ et $\Omega_{Evaluation}$, nous utilisons les profils Contenu et Evaluation des utilisateurs, comme illustré dans la Figure 16.1. La préparation des données pour cette partie d'expérimentation est assez simple. Effectivement, les profils Contenu sont les vecteurs pondérés de genres préférés préalablement bâtis dans l'étape d'extraction des valeurs des critères alors que les profils Evaluation sont les ensembles des évaluations des utilisateurs.

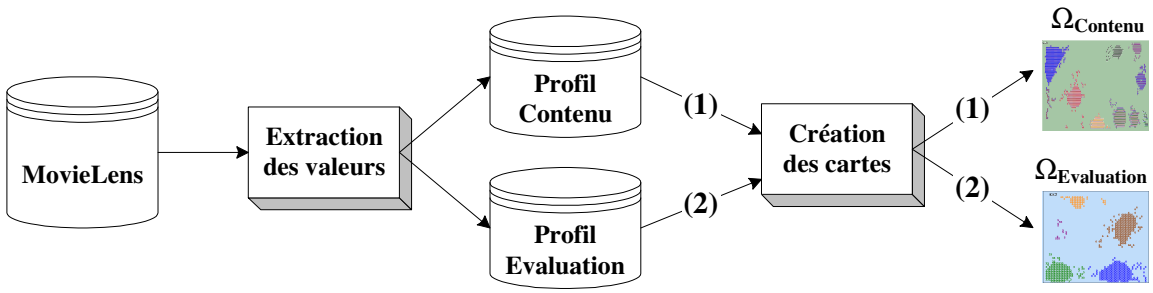


Figure 16.1 – *Expérimentation sur la formation des communautés.*

Nous commençons la création d'une des deux cartes en plaçant aléatoirement les utilisateurs sur une grille de taille 100 x 100. En nous basant sur les étapes illustrées dans la Figure 9.2, nous avons d'abord besoin d'une mesure de proximité entre les profils pour le processus de positionnement des utilisateurs dans l'algorithme des fourmis artificielles. Cette mesure est utilisée pour le calcul de la fonction de densité *f* dans la formule (11.6). Sachant que nos objectifs sont d'abord de créer, et ensuite de comparer les deux cartes $\Omega_{Contenu}$ et $\Omega_{Evaluation}$, nous utilisons deux mesures de proximité :

[12] Dans le reste du manuscrit, nous utilisons le symbole Ω_a pour désigner sans distinction l'espace et la carte de communautés du critère $a \in A$.

- la corrélation de Pearson pour le calcul sur le critère Evaluation, comme c'est l'usage courant dans les systèmes de filtrage collaboratif traditionnels, et

- pour le critère Contenu, nous choisissons la distance euclidienne en raison de sa haute performance dans les espaces euclidiens en 2D. Ceci a été montré par des travaux expérimentaux [JMF99].

Pour calibrer les paramètres de l'algorithme des fourmis artificielles, nous exploitons les expériences décrites dans [DBT00, HKD03] avec quelques modifications afin d'obtenir de bons résultats (voir Tableau 16.1).

Paramètre ($m = \|U\| = 943$ personnes)	Valeur
k_1	0,10
k_2	0,15
α pour critère Contenu	0,50
α pour critère Evaluation	1,40
Taille de la grille ($\sim\sqrt{m.10}$)	100
Nombre d'agents (fourmis)	10
Taille de voisinage	3 x 3
Seuil pour ramasser et déposer	0,05

Tableau 16.1 – *Valeurs de paramètre pour l'algorithme des fourmis artificielles*.

Puisque le nombre d'agents est très petit par rapport au nombre d'utilisateurs m et au nombre d'itérations t, la complexité de cet algorithme est en $O(m^2.t)$ [SP04]. En nous basant sur la formule (16.6), nous constatons que l'algorithme des fourmis artificielles donne de bonnes classifications à partir de 1,5 millions d'itérations (voir Tableau 16.2), et nous choisissons 2 millions d'itérations, comme Handl et al. l'ont proposé, compte tenu du temps d'exécution raisonnable (moins de 3mn, sans compter le temps de calcul des distances entre utilisateurs).

En ce qui concerne l'algorithme des K-moyennes, le nombre de classes est limité de 10 à 19 afin d'obtenir les communautés dont la taille moyenne varie de 50 à 100 personnes. En pratique, il est certain que le nombre de communautés dans les deux cartes Ω_{Contenu} et $\Omega_{\text{Evaluation}}$ n'est pas toujours égal. Pourtant, nous rappelons qu'un des objectifs de cette expérimentation est de comparer la dispersion des utilisateurs dans ces cartes. Ainsi, nous utilisons la même valeur du paramètre K pour les deux cartes afin de faciliter la comparaison par paires de communautés de chacune des deux cartes sans perte de précision quant au résultat de la comparaison.

Nous appliquons 50 essais consécutifs de la chaîne des processus présentée dans la Figure 16.1, aux profils Contenu ainsi qu'aux profils Evaluation, et nous donnons plus tard les résultats moyens de ces observations.

16.2.2 Mesures

Afin de mesurer le niveau de dissimilarité des cartes $\Omega_{Contenu}$ et $\Omega_{Evaluation}$, nous utilisons les deux indicateurs Rand Index et F_Mesure comme dans les expériences de Handl et al. [HKD03].

16.2.2.1 Rand Index

L'indicateur Rand Index [Ran71] est une des mesures courantes de (dis)similarité par paires de classes entre deux classifications. La formule (16.1) prend valeur dans l'intervalle [0, 1]. Si cette valeur est égale à 1, les classifications sont identiques. Par contre, si elle est égale à 0, on a deux classifications complètement différentes.

$$R = \frac{a + d}{a + b + c + d} \tag{16.1}$$

où $a = |\{x_i, x_j / (\text{classe}_{Contenu}(x_i) = \text{classe}_{Contenu}(x_j)) \text{ et } (\text{classe}_{Evaluation}(x_i) = \text{classe}_{Evaluation}(x_j))\}|$

$b = |\{x_i, x_j / (\text{classe}_{Contenu}(x_i) = \text{classe}_{Contenu}(x_j)) \text{ et } (\text{classe}_{Evaluation}(x_i) \neq \text{classe}_{Evaluation}(x_j))\}|$

$c = |\{x_i, x_j / (\text{classe}_{Contenu}(x_i) \neq \text{classe}_{Contenu}(x_j)) \text{ et } (\text{classe}_{Evaluation}(x_i) = \text{classe}_{Evaluation}(x_j))\}|$

$d = |\{x_i, x_j / (\text{classe}_{Contenu}(x_i) \neq \text{classe}_{Contenu}(x_j)) \text{ et } (\text{classe}_{Evaluation}(x_i) \neq \text{classe}_{Evaluation}(x_j))\}|$

16.2.2.2 F_Mesure

L'indicateur F_Mesure a été proposé par van Rijsbergen pour estimer la qualité globale de recherche d'information [vRi79]. Elle est calculée à partir de la précision et du rappel. La *précision* du résultat d'une requête est la proportion des documents pertinents parmi ceux retrouvés (cf. (16.2)), et le *rappel* est la proportion des documents pertinents par rapport à ceux qui sont pertinents dans le corpus (cf. (16.3)), comme illustré dans la Figure 16.2.

$$précision = \frac{|P \cap R|}{|R|} \tag{16.2}$$

$$rappel = \frac{|P \cap R|}{|P|} \tag{16.3}$$

Alors, l'indicateur F_Mesure est calculé par :

$$F = 2.\frac{précision\ .\ rappel}{(précision + rappel)} \tag{16.4}$$

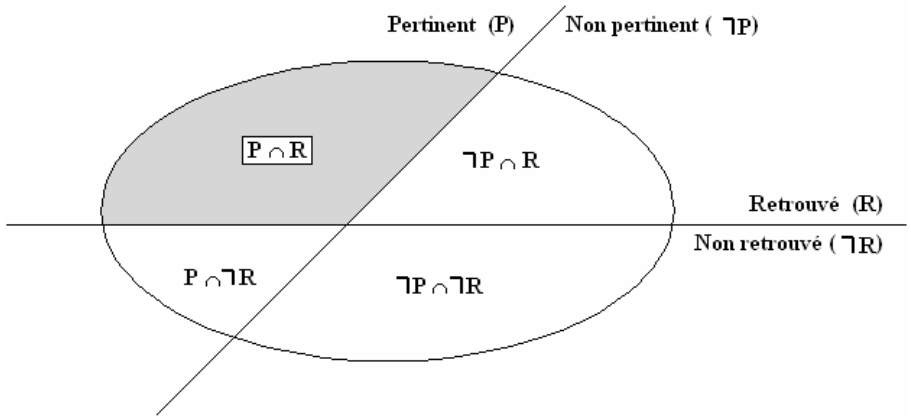


Figure 16.2 – *Précision et rappel en Recherche d'information*.

Si l'on considère par exemple $\Omega_{Contenu}$ comme la classification de référence et $\Omega_{Evaluation}$ comme la classification à comparer, on peut définir une adaptation de la formule (16.4) :

$$F' = 2.\frac{p_{ij}.r_{ij}}{(p_{ij} + r_{ij})} \tag{16.5}$$

où n_i : nombre d'éléments de la classe C_i dans la carte $\Omega_{Contenu}$
n_j : nombre d'éléments de la classe C_j dans la carte $\Omega_{Evaluation}$
n_{ij} : nombre d'éléments de la classe C_i apparaissant dans la classe C_j
$p_{ij} = n_{ij} / n_j$: précision de la classification $\Omega_{Evaluation}$
$r_{ij} = n_{ij} / n_i$: rappel de la classification $\Omega_{Evaluation}$

16.3 Analyse

Après avoir présenté le protocole de l'expérimentation dans la section précédente, nous donnons ici ses performances (temps d'exécution) et les résultats de l'analyse comparative des cartes $\Omega_{Contenu}$ et $\Omega_{Evaluation}$.

16.3.1 Performance des algorithmes

Les résultats que nous présentons ici sont obtenus à partir de 50 essais indépendants réalisés sur un PC de 2,40 Ghz de processeur et 1 Go de mémoire.

La performance de l'algorithme des fourmis artificielles est montrée dans le Tableau 16.2 où on voit des temps d'exécution raisonnables. De plus, grâce à la densité de carte (cf. (16.6)), nous pouvons choisir le seuil de 2 000 000 d'itérations (t_{max} ~ 2000 x m) comme dans les expériences de Handl et al. [HKD03] (voir Figure 11.3).

$$densité(\{C_1,...,C_k\}) = \sum_{j=1}^{k} \sum_{x,y \in C_j} d^2(x,y) \qquad (16.6)$$

où $d(x, y)$: distance entre deux éléments x et y dans une classe C_j.

Itération	5.10^5	10.10^5	15.10^5	20.10^5	25.10^5	30.10^5
Densité (%)	21,84	20,29	16,85	16,66	16,68	16,27
Temps (s)	48,53	88,94	129,53	172,72	217,69	256,34

Tableau 16.2 – *Densité de carte et temps d'exécution en fonction du nombre d'itérations (K = 10).*

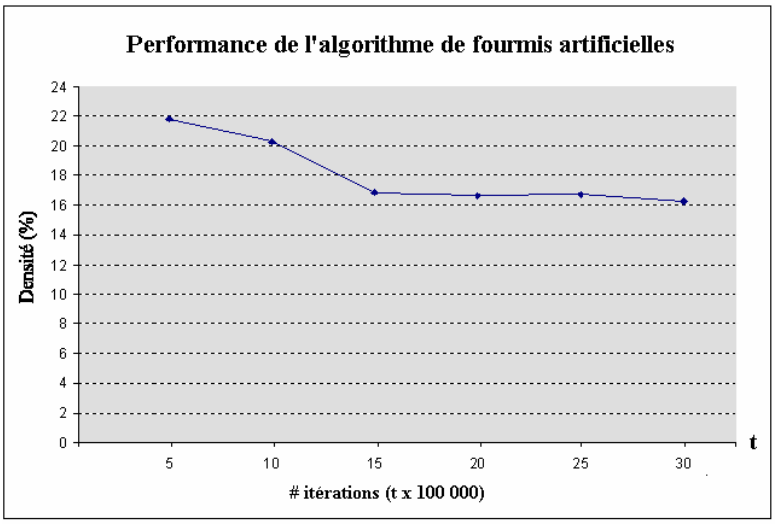


Figure 16.3 – *Densité de carte et temps d'exécution en fonction du nombre d'itérations (K = 10).*

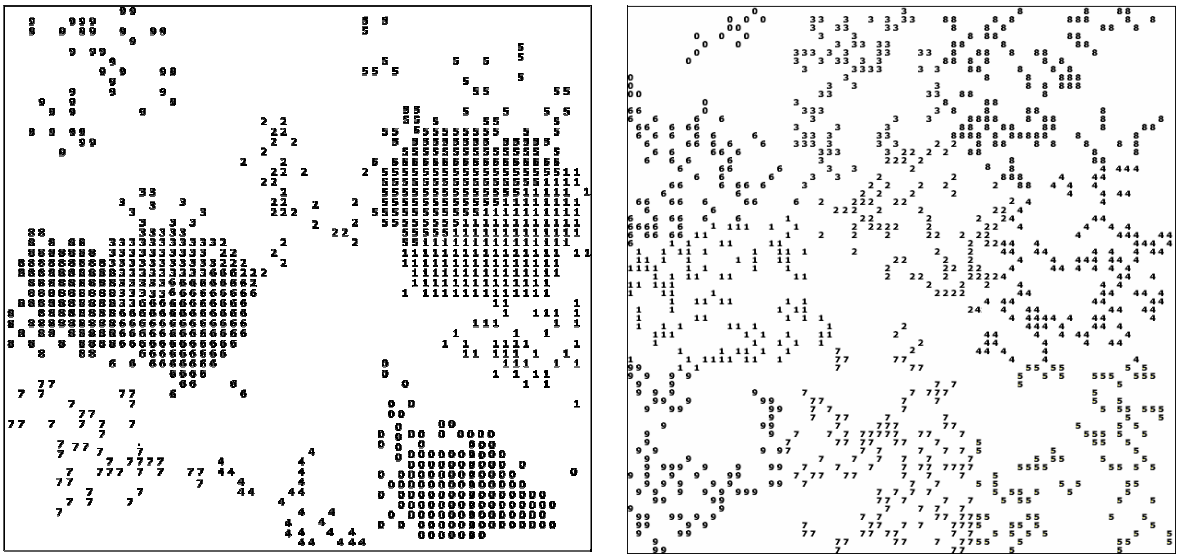
En appliquant l'algorithme des K-moyennes dont la complexité est en O(|*U*|) puisque la valeur de *K* est souvent très petite par rapport à |*U*|, sur le résultat du premier processus, la vitesse de convergence est assez rapide du fait que les utilisateurs sont déjà bien placés par l'algorithme des fourmis artificielles. On peut obtenir des classifications stables après une moyenne de vingtaine d'itérations et moins de 10 secondes (voir Tableau 16.3).

	K = 10	*K* = 15	*K* = 19
Itération moyenne	18,16	19,84	20,92
Temps (s)	7,90	8,60	9,56

Tableau 16.3 – *Performance de l'algorithme des K-moyennes.*

16.3.2 Comparaison

La Figure 16.4 nous donne une comparaison visuelle « à l'œil nu » des deux cartes $\Omega_{Contenu}$ et $\Omega_{Evaluation}$ pour une première impression. On peut aisément constater un écartement significatif entre ces cartes de communautés : les personnes dans la carte $\Omega_{Contenu}$ sont bien regroupées par rapport à la carte $\Omega_{Evaluation}$. A propos de la dispersion des utilisateurs dans la carte $\Omega_{Evaluation}$, Breese et al. ont évoqué le problème du nombre faible d'objets jugés en commun entre deux utilisateurs dans le calcul de la corrélation de Pearson [BHK98] que nous utilisons par analogie avec les systèmes de filtrage collaboratif. De fait, nous constatons que la majorité des utilisateurs dans le jeu MovieLens ont fait au maximum 25% d'évaluations en commun (avec des scores différents) avec les autres (voir Figure 15.1).



(a) Carte $\Omega_{Contenu}$ (b) Carte $\Omega_{Evaluation}$

Figure 16.4 – *Visualisation des cartes de communautés (K = 10).*

L'écartement entre ces cartes est aussi reflété de façon statistique par les indicateurs Rand Index et F_Mesure (voir Tableau 16.4 et Figure 16.5). Ce sont les résultats moyens de 50 essais réalisés en fonction d'un nombre de communautés identique *K* dans les deux cartes. Les valeurs faibles (< 0,25) de ces deux indicateurs dans le Tableau 16.4 montrent la différence nette entre les deux cartes $\Omega_{Contenu}$ et $\Omega_{Evaluation}$.

Par ailleurs, les données utilisées dans l'expérimentation sont la base de données totale qui contient toutes les évaluations des utilisateurs. Les résultats du Tableau 16.4 expriment ainsi l'écartement des cartes à la fin avril 1998. Afin de prouver la permanence de cet écartement, nous réalisons en plus une analyse temporelle sur les indicateurs, et nous obtenons les mêmes résultats (voir Tableau 16.5 et Figure 16.6). Dans cette analyse, nous laissons de côté les données des deux premiers mois en raison du nombre faible d'utilisateurs, 80 en 09/97 et 189 en 10/97.

Indication	$K = 10$	$K = 15$	$K = 19$
Rand Index (%)	21,3784	15,0734	11,9748
F_Mesure (%)	16,3473	13,3683	12,1261

Tableau 16.4 – *Faible similarité des cartes $\Omega_{Contenu}$ et $\Omega_{Evaluation}$.*

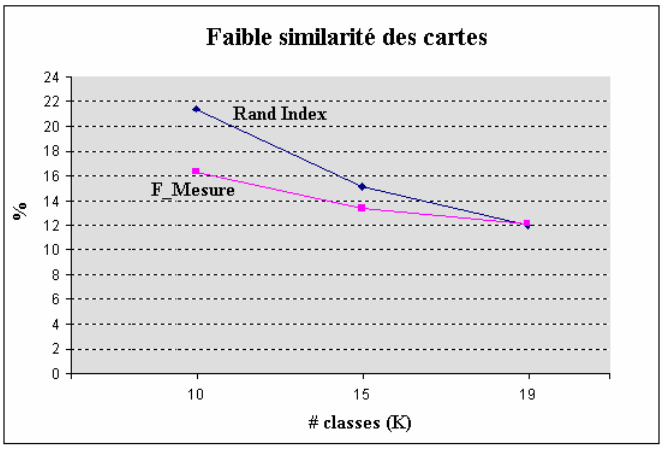


Figure 16.5 – *Faible similarité des cartes $\Omega_{Contenu}$ et $\Omega_{Evaluation}$.*

	11/97	12/97	01/98	02/98	03/98	04/98
(# utilisateurs)	(426)	(529)	(647)	(736)	(866)	(943)
Rand Index (%)	6,4818	9,0343	12,0032	12,0032	19,4652	21,3784
F_Mesure (%)	18,7740	18,5326	18,0546	17,7703	17,4561	16,3473

Tableau 16.5 – *Analyse mensuelle des indications Rand Index et F_Mesure : permanence de la faible similarité entre les cartes $\Omega_{Contenu}$ et $\Omega_{Evaluation}$ ($K = 10$).*

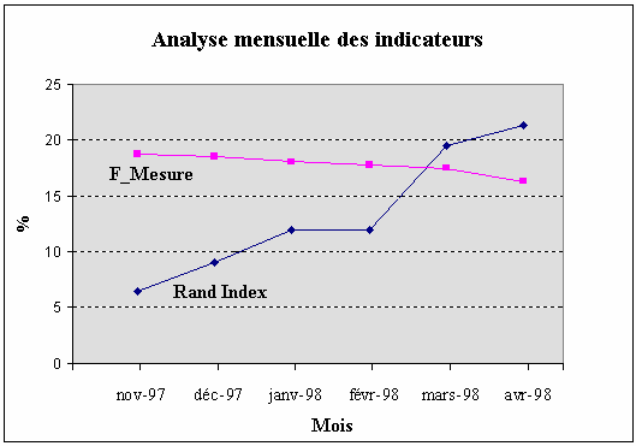


Figure 16.6 – *Analyse mensuelle des indications Rand Index et F_Mesure : permanence de la faible similarité entre les cartes $\Omega_{Contenu}$ et $\Omega_{Evaluation}$ ($K = 10$).*

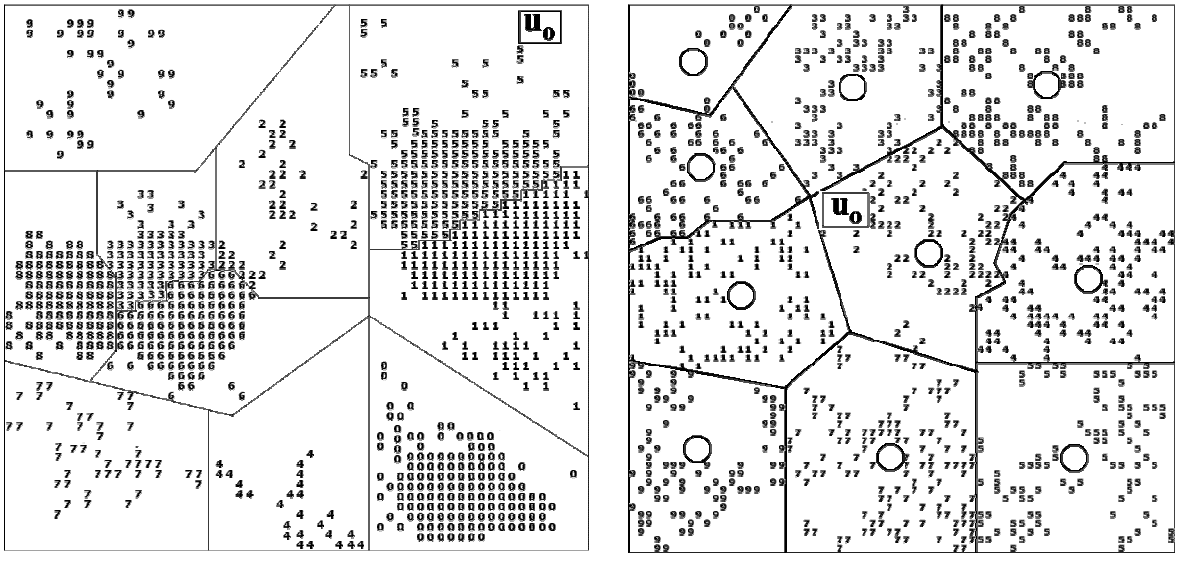
16.4 Conclusion

En résumé, les résultats d'expérimentation permettent de valider notre approche d'utilisation des cartes de communautés dans un système de filtrage collaboratif.

Effectivement, on voit la faisabilité de créer les cartes des critères complexes tels que Contenu et Evaluation sur une base de données de taille importante en des temps raisonnables.

De plus, avec de telles cartes, le système de filtrage collaboratif offre aux utilisateurs la possibilité de percevoir non seulement leurs propres communautés comme dans la plateforme COCoFil [DBG+04] mais aussi d'autres communautés existantes. Par exemple, en respectant certains niveaux de confidentialité, les utilisateurs peuvent éventuellement consulter les profils des personnes dans toutes les communautés d'une carte donnée en vue de se (re)positionner dans une communauté appropriée. On peut envisager la construction des profils représentatifs des communautés pour faciliter leur perception, ce que nous mettons en perspectives de cette thèse.

Finalement, on constate sur le jeu de données MovieLens des contrastes permanents entre les cartes traduisant les dimensions Contenu et Evaluation dans les profils des utilisateurs, contrastes que l'on espère pouvoir exploiter comme point de départ à un diagnostic de la situation de l'utilisateur, ou au moins à l'adaptation du profil de l'utilisateur. Ainsi, prenons le scénario où l'utilisateur u_0 est en situation difficile, reflétée par son envoi de retours de pertinence négatifs accumulés. On peut voir dans la carte $\Omega_{Contenu}$ que cet utilisateur est très isolé (voir Figure 16.7.a) tandis qu'il se positionne à l'intersection de plusieurs communautés dans la carte $\Omega_{Evaluation}$ (voir Figure 16.7.b). La Figure 16.7 montre les cartes que l'on pourrait proposer à l'utilisateur u_0 pour lui proposer de se rattacher à une des classes représentées par des utilisateurs typiques symbolisés par des cercles sur la carte $\Omega_{Evaluation}$. Cette fonctionnalité nécessite une étude de la capacité cognitive des utilisateurs à utiliser de telles cartes dans ce but. Nous en discutons dans la partie Conclusion.



(a) Carte $\Omega_{Contenu}$ (b) Carte $\Omega_{Evaluation}$

Figure 16.7 – *Proposition de positionnement dans des communautés.*

Chapitre 17

Mesures de qualité de critères

17.1 Objectifs

Nos travaux d'expérimentation dans ce chapitre ne visent pas à calibrer des paramètres ou à montrer la performance d'un système particulier s'appuyant sur notre modèle des espaces de communautés, mais à montrer comment analyser les critères disponibles dans les profils dans l'optique qu'ils se compensent entre eux afin de surmonter les situations délicates telles que le démarrage à froid. L'objectif principal du présent chapitre est donc de valider les mesures de qualité de critères proposées dans le modèle en les utilisant pour comparer les critères en tant qu'attribut de décision dans l'induction des communautés.

17.2 Protocole

Les données nécessaires à la validation des mesures sont la table de communautés de 943 vecteurs de positionnement d'utilisateurs. Le processus de remplissage de cette table par les espaces de communautés relatifs aux 6 critères Age, Profession, Géographie, Motivation, Contenu et Evaluation, est discuté par la suite.

Utilisateur	Age	Profession	Géographie	Motivation	Contenu	Evaluation
u_1	16-25	Ingénieur	LA	Excellente	$G_{Contenu\_8}$	$G_{Evaluation\_34}$
u_2	46-60	Autres	SC	Bonne	$G_{Contenu\_8}$	$G_{Evaluation\_12}$
u_3	16-25	Loisir	MD	Moyenne	$G_{Contenu\_4}$	$G_{Evaluation\_58}$
u_4	16-25	Ingénieur	SD	Excellente	$G_{Contenu\_2}$	$G_{Evaluation\_28}$
u_5	26-45	Autres	WA	Faible	$G_{Contenu\_5}$	$G_{Evaluation\_27}$
u_6	26-45	Commerce	WA	Faible	$G_{Contenu\_4}$	$G_{Evaluation\_20}$
u_7	46-60	Ingénieur	ID	Excellente	$G_{Contenu\_5}$	$G_{Evaluation\_81}$
u_8	26-45	Ingénieur	WA	Très faible	$G_{Contenu\_6}$	$G_{Evaluation\_2}$
u_9	26-45	Chercheur	WA	Faible	$G_{Contenu\_1}$	$G_{Evaluation\_25}$
u_{10}	46-60	Commerce	AK	Faible	$G_{Contenu\_6}$	$G_{Evaluation\_8}$

Tableau 17.1 – *Extrait de la table de communautés.*

Pour les 4 premiers critères, la création des espaces est assez simple, avec un nombre fixe de communautés dans chaque espace de communautés :

– Espace Ω_{Age} : Les utilisateurs sont regroupés par tranche d'âge, et nous obtenons 5 communautés dans l'espace Ω_{Age} ;

– Espace $\Omega_{Profession}$: Nous construisons 7 communautés en fonction des catégories de profession ;

– Espace $\Omega_{Géographie}$: En utilisant les états déduits des zip codes donnés par les utilisateurs, nous formons également 44 communautés dans l'espace $\Omega_{Géographie}$;

– Espace $\Omega_{Motivation}$: La caractéristique de « tendance à fournir des évaluations » nous permet de construire 5 communautés conformément à 5 niveaux de motivation : très faible, faible, moyenne, bonne et excellente.

Pour simplifier les notations, nous réutilisons les valeurs des critères comme les étiquettes des communautés dans les espaces Ω_{Age}, $\Omega_{Profession}$, $\Omega_{Géographie}$ et $\Omega_{Motivation}$.

Pour créer l'espace $\Omega_{Contenu,}$ le système regroupe les utilisateurs partageant les mêmes intérêts quant aux genres de film alors que dans l'espace $\Omega_{Evaluation}$, les utilisateurs sont regroupés selon leur façon de juger les films. La construction des espaces $\Omega_{Contenu}$ et $\Omega_{Evaluation}$ est donc plus élaborée que pour les 4 critères déjà évoqués.

En pratique, les utilisateurs du système MovieLens sont souvent bien regroupés dans $\Omega_{Contenu}$, et le nombre de communautés est stable tandis que le critère Evaluation conduit à une dispersion des utilisateurs en raison du faible nombre de films jugés en commun entre utilisateurs. De ce fait, afin d'utiliser le nombre de communautés comme paramètre variable, nous modifions notre méthode décrite dans 11.4.3 où nous appliquions d'abord la méthode des fourmis artificielles pour placer les utilisateurs dans une grille en 2D, puis la méthode des K-moyennes pour former les communautés d'utilisateurs concrètes dans le but de comparer des catégories de communautés. Nous nous en

remettons à la classification ascendante hiérarchique dans laquelle on commence par créer les classes de singleton, et dans la boucle principale, on fusionne les deux classes les plus proches jusqu'à ce que le test d'arrêt sur l'inertie ou sur le nombre de classes soit satisfait (voir Figure 17.1).

Pour estimer la distance entre deux classes C_1 et C_2, les approches les plus populaires sont [JMF99] :

a) Distance minimale : $$\text{distance}(C_1, C_2) = \min_{\substack{c_1 \in C_1 \\ c_2 \in C_2}} [d(c_1, c_2)] \quad (17.1)$$

b) Distance maximale : $$\text{distance}(C_1, C_2) = \max_{\substack{c_1 \in C_1 \\ c_2 \in C_2}} [d(c_1, c_2)] \quad (17.2)$$

c) Distance moyenne : $$\text{distance}(C_1, C_2) = \frac{1}{|C_1| \cdot |C_2|} \sum_{\substack{c_1 \in C_1 \\ c_2 \in C_2}} d(c_1, c_2) \quad (17.3)$$

d) Distance de centres de gravité : $$\text{distance}(C_1, C_2) = d(cg_1, cg_2) \quad (17.4)$$

où cg_1 et cg_2 sont respectivement les centres de gravité de C_1 et C_2

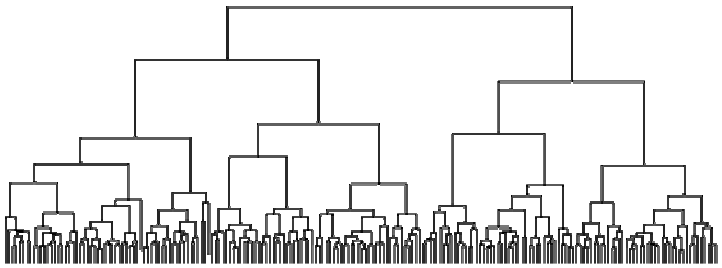


Figure 17.1 – *Algorithme de classification ascendante hiérarchique*.

Après avoir testé ces alternatives de distance, nous choisissons pour notre expérimentation la distance minimale en raison de sa meilleure performance de classification.

Normalement, si l'on commence par 943 classes de singleton, la classification hiérarchique est inefficace pour ne pas dire irréaliste. Ainsi, nous appliquons d'abord l'algorithme des K-moyennes avec K = 100 sur l'emplacement calculé préalablement par l'algorithme des fourmis artificielles pour créer 100 communautés initiales, et nous effectuons la classification ascendante hiérarchique sur ces classes pour former les communautés finales. Notre méthode de création des espaces ci-dessus est illustrée dans la Figure 17.2.

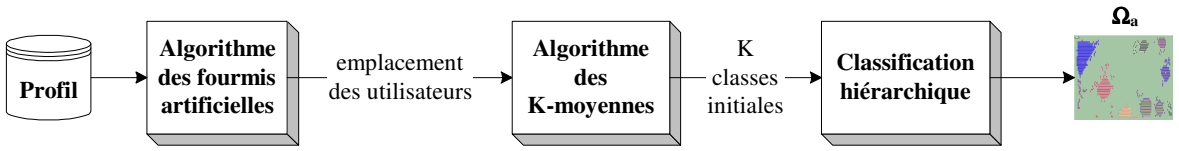


Figure 17.2 – *Méthode de création des cartes $\Omega_{Contenu}$ et $\Omega_{Evaluation}$.*

Cette modification nous permet d'une part d'améliorer la performance de la classification hiérarchique, et d'autre part d'obtenir un nombre de communautés plus flexible. Dans l'espace $\Omega_{Contenu}$, le nombre de communautés obtenu par la classification hiérarchique est relativement stable quand on fait varier le seuil d'entropie, alors que celui de $\Omega_{Evaluation}$ varie fortement. Toutes les mesures impliquées dans l'expérimentation seront donc paramétrées par le nombre de communautés dans $\Omega_{Evaluation}$.

Notre but est d'établir un ordre entre les critères afin de savoir lequel utiliser comme critère clé pour prédire au mieux une valeur manquante ou douteuse dans le vecteur de positionnement d'un utilisateur. Pour chacune des mesures proposées, nous calculons donc les valeurs pour les 6 critères, et nous comparons ces valeurs afin de trier les critères. Bien que le calcul des ensembles d'approximation soit en général un problème NP-difficile, nous n'utilisons aucune méthode heuristique [ZY04] en raison du nombre faible de critères dans le jeu de test et du temps de calcul raisonnable, une dizaine de minutes par analyse.

17.3 Analyse

Nous rappelons tout d'abord que l'objectif concret de cette expérimentation est d'établir un ordre entre les critères selon une mesure considérée, pour savoir lequel utiliser comme critère clé afin de prédire au mieux une valeur manquante ou douteuse dans le vecteur imparfait de positionnement d'un utilisateur.

Dans le Tableau 17.2 relatif à la mesure basée sur la consistance (cf. (8.9)), nous constatons que le critère Evaluation est la pire décision puisque la consistance du résultat est toujours la plus faible, que Géographie et Age sont les meilleurs critères, et que Profession et Motivation viennent ensuite, avant le critère Contenu. Nous remarquons aussi que plus le nombre de communautés de $\Omega_{Evaluation}$ diminue, de 93 à 10, plus la consistance de la table se dégrade excepté celle relative au critère Evaluation qui montre la tendance inverse.

Ce phénomène est logique puisque si Evaluation est pris comme attribut de décision et le nombre de communautés dans $\Omega_{Evaluation}$ est élevé, la taille moyenne d'une classe d'équivalence $[u]_{Evaluation}$ dans $U/\mathcal{R}_{Evaluation}$ devient petite. Par conséquent, le nombre de règles certaines (cf. (8.5)) dans la région positive concernée POS_C(Evaluation) est considérablement diminué.

Par contre, si Evaluation fait partie de l'ensemble des attributs de condition C, la taille d'une classe $[u]_C$ a tendance à se réduire, et il est fortement probable qu'elle soit incluse dans une certaine

classe d'équivalence $[u]_D$ (cf. (8.5)). Par exemple, si l'on crée 93 communautés dans $\Omega_{\text{Evaluation}}$, la majorité des classes d'équivalence $[u]_C$, où Evaluation $\in C$, sont des singletons. Alors, selon la remarque (8.6) dans la section 8.4.1, toutes ces classes sont certainement incluses dans des classes d'équivalence dans $U/\mathcal{R}_D$, où D est un des critères restants. Par conséquent, la taille des régions positives $POS_C(D \neq \{Evaluation\})$ augmente considérablement, et on voit ainsi les grandes valeurs de consistance, de 86,74% à 95,97%, de 5 premières lignes dans la dernière colonne du Tableau 17.2.

	Nombre de communautés dans l'espace $\Omega_{\text{Evaluation}}$					
	10	21	31	51	79	93
Géographie	77,62	85,68	89,82	93,64	95,55	95,97
Age	76,25	83,78	86,74	91,41	93,85	94,91
Profession	61,40	71,79	78,90	87,27	92,26	93,21
Motivation	60,13	69,99	76,67	86,21	91,20	92,90
Contenu	50,27	60,45	67,76	76,88	84,94	86,74
Evaluation	46,98	46,55	46,02	45,71	45,28	45,28

Tableau 17.2 – *Analyse de la consistance de la table de communautés (%).*

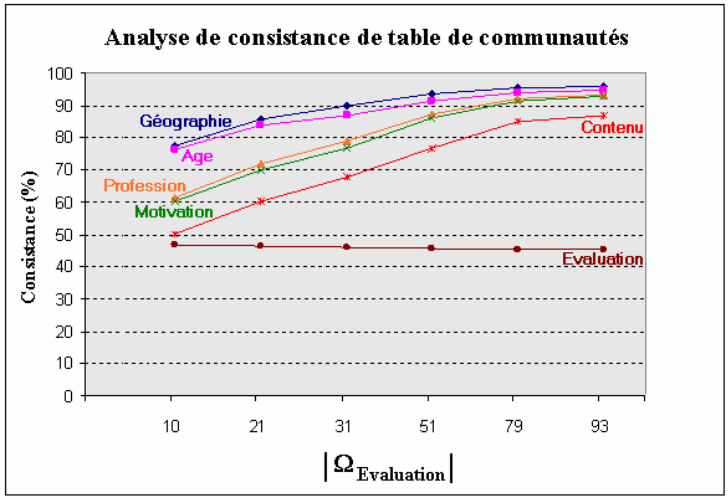


Figure 17.3 – *Analyse de la consistance de la table de communautés (%).*

Pourtant, nous soulignons ici un autre phénomène très intéressant sur les deux critères Géographie et Contenu, et qui montre l'utilité de la mesure dans ce contexte applicatif. En principe, si le critère choisi comme décision forme un nombre élevé de petites communautés pour faciliter par exemple la perception des communautés, la taille de la région positive, et donc la consistance de la table de communautés, risque d'être faible. Au contraire, si l'on prend comme décision un critère formant un petit nombre de grandes communautés, on a plus de chance d'obtenir une meilleure consistance. Pourtant, notre mesure a montré que selon les données ce n'est pas toujours le cas. En effet, le critère Géographie domine tous les autres quoi qu'il donne lieu à 44 communautés et une

vingtaine de personnes par communauté, alors que le critère Contenu qui ne crée que 8 communautés n'est pas bien classé (5e position). La mesure proposée permet donc d'ordonner les critères en tenant compte des données elles-mêmes via la consistance de la table, ce qui garantit un meilleur choix que si l'on se contente du simple choix heuristique favorisant le plus faible nombre de communautés.

Pour la première mesure basée sur les réductions approximatives (cf. (8.17)), le paramètre α peut varier de 1 à 5 selon le nombre de critères que l'on souhaite dans les conditions. Pour l'expérience nous choisissons la valeur 1 pour nous placer dans une situation où l'utilisateur interagit avec les communautés : il comprendra mieux un critère simple qu'un critère composé. La valeur 0,7 de θ a été choisie en se basant sur le Tableau 17.2. Pour la formule (8.18), nous choisissons Géographie pour C_0 à titre d'exemple.

Dans le Tableau 17.3 et le Tableau 17.4 qui donnent la taille de $R_D^{(\theta)}$, le critère Evaluation dont les communautés sont coûteuses à calculer, est déjà « éliminé » ; les deux critères Géographie et Age sont toujours dominants mais il y a un changement de priorité entre Motivation et Profession par rapport à la première mesure. La différence est assez négligeable pour que l'on puisse l'ignorer si l'on souhaite favoriser Profession en raison de la simplicité du calcul de $\Omega_{\text{Profession}}$ comparé à $\Omega_{\text{Motivation}}$.

	Nombre de communautés dans l'espace $\Omega_{\text{Evaluation}}$					
	10	21	31	51	79	93
Géographie	-	1	3	4	4	4
Age	-	1	1	3	4	4
Motivation	-	-	-	2	3	3
Profession	-	-	-	1	3	3
Contenu	-	-	-	-	2	2
Evaluation	-	-	-	-	-	-

Tableau 17.3 – *Analyse du nombre de réductions approximatives $R_D^{(\theta)}$ ($\theta = 0,7$ et $\alpha = 1$).*

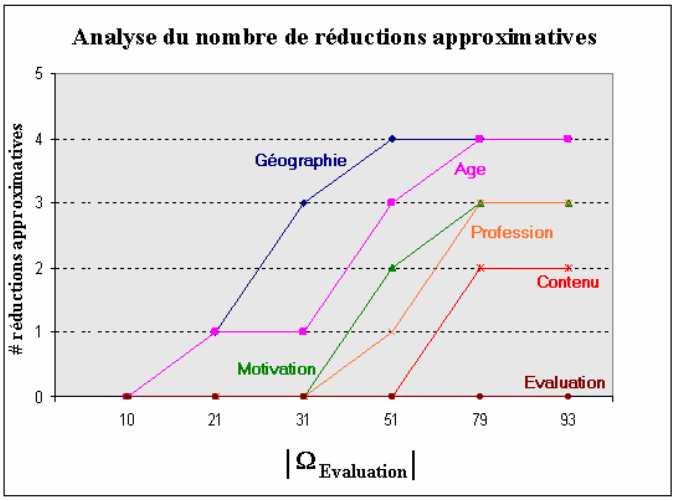


Figure 17.4 – *Analyse du nombre de réductions approximatives $R_D^{(\theta)}$ ($\theta = 0,7$ et $\alpha = 1$).*

	Nombre de communautés dans l'espace $\Omega_{Evaluation}$					
	10	21	31	51	79	93
Age	-	1	1	1	2	3
Motivation	-	-	-	1	1	1
Profession	-	-	-	1	1	1
Contenu	-	-	-	-	1	1
Evaluation	-	-	-	-	-	-

Tableau 17.4 – *Analyse du nombre de réductions approximatives* $R_D^{(\theta)}$ *(*$\theta = 0,7$ *et* $C_0 = \{Géographie\}$*).*

Sur le jeu de données MovieLens, la mesure basée sur la consistance approximative (cf. (8.20)) dont les résultats sont présentés dans le Tableau 17.5 permet de départager les critères indistingués par les autres mesures, et cela sans conduire à un conflit, car l'ordre est compatible avec celui des autres mesures.

	Nombre de communautés dans l'espace $\Omega_{Evaluation}$					
	10	21	31	51	79	93
Géographie	-	72,75	75,19	80,62	81,04	82,38
Age	-	71,79	78,05	78,15	80,51	81,07
Profession	-	-	-	74,23	79,53	81,51
Motivation	-	-	-	73,49	79,39	81,48
Contenu	-	-	-	-	72,48	75,66
Evaluation	-	-	-	-	-	-

Tableau 17.5 – *Analyse de la consistance approximative (%) avec* $\theta = 0,7$*.*

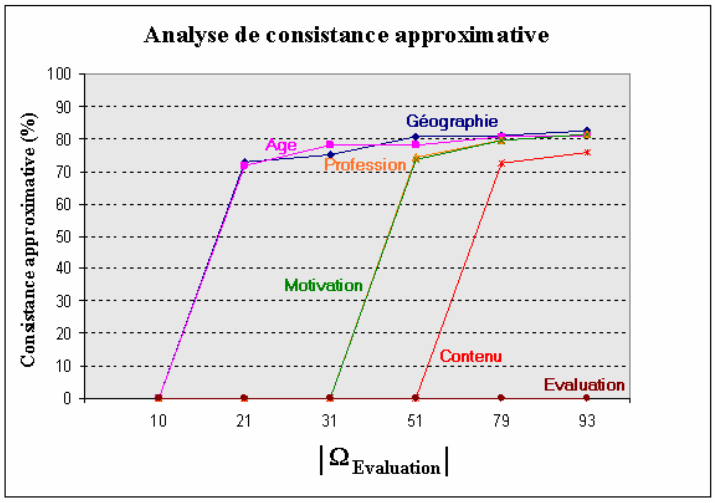


Figure 17.5 – *Analyse de la consistance approximative (%) avec* $\theta = 0,7$*.*

17.4 Conclusion

Les résultats expérimentaux présentés dans ce chapitre montrent que, dans le cas des données de MovieLens, les critères Evaluation et Contenu sont ceux qui obtiennent la priorité la plus faible comme critère clé, alors qu'ils correspondent aux communautés les plus coûteuses à former. Cela indique que l'on peut limiter ce coût en positionnant progressivement les nouveaux utilisateurs dans ces deux espaces via leurs positions dans les autres espaces. Cette approche améliore donc la performance de la formation multicritère de communautés dans un système de filtrage.

Pour conclure, outre les éclairages qu'elles procurent sur les données elles-mêmes, nous constatons que ces mesures de qualité de critère permettent, sur ce jeu de données réelles, de proposer un ordre dans lequel traiter les critères comme attribut de décision dans le processus de correction des vecteurs de positionnement tout en tenant compte des caractéristiques des données considérées.

Chapitre 18

Démarrage à froid

18.1 Objectifs

Dans les systèmes de filtrage collaboratif, les utilisateurs reçoivent des documents que leur recommande le système sur la base de leurs communautés, mais le problème du démarrage à froid conduit à des performances très pauvres pour les nouveaux utilisateurs.

L'objectif de cette partie d'expérimentation est de valider sur le jeu de données MovieLens notre méthode d'intégration de nouveaux utilisateurs en appliquant l'induction des communautés à partir des données « disponibles à froid ». Le principe est d'associer automatiquement les meilleures communautés initiales à un nouvel utilisateur, communautés à partir desquelles seront générées ses premières recommandations. Ces données disponibles à froid sont les informations sur l'utilisateur que l'on peut recueillir dès son inscription, et que l'utilisateur peut fournir avec une grande fiabilité, et sans effort particulier : par exemple, son âge, sa profession et son lieu de résidence.

18.2 Protocole

18.2.1 Données d'entrée et méthodes

Le schéma de notre expérimentation sur le démarrage à froid, comprenant le prétraitement des données et les trois méthodes de générer des recommandations, est illustré dans la Figure 18.1.

En ce qui concerne les données pour notre expérimentation, nous exploitons 5 critères Age, Profession, Géographie, Contenu et Evaluation, permettant en principe de former 5 espaces de communautés respectivement. Parmi ces critères, il est assez facile à fournir la valeur des trois premiers par un nouvel utilisateur u dès son inscription. Le système peut alors le positionner dans les espaces de communautés concernés, et doit prédire ensuite ses communautés dans les espaces Ω_{Contenu} et $\Omega_{\text{Evaluation}}$. Donc, son vecteur de positionnement $\mathcal{P}_u$ contient ses propres communautés G_{Age}, $G_{\text{Profession}}$ et $G_{\text{Géographie}}$ dans trois espaces Ω_{Age}, $\Omega_{\text{Profession}}$ et $\Omega_{\text{Géographie}}$ respectivement et deux communautés manquantes dans les espaces Ω_{Contenu} et $\Omega_{\text{Evaluation}}$ (voir $\mathcal{P}_u$ en haut dans la Figure 18.1).

Tout d'abord, nous divisons l'ensemble des utilisateurs de MovieLens en deux catégories :

(i) l'ensemble des 77 « nouveaux » utilisateurs (U_N) qui se sont inscrits au cours du dernier mois (04/1998), et

(ii) l'ensemble des 866 utilisateurs « existants » (U_E) inscrits avant le dernier mois.

Trois méthodes pour le démarrage à froid y compris la méthode proposée dans la thèse seront appliquées aux vecteurs initiaux incomplets des utilisateurs dans l'ensemble U_N pour prédire leurs communautés manquantes et pour générer ensuite les premières recommandations pour eux-mêmes. Ainsi, la performance de ces méthodes de démarrage à froid sera mesurée de façon indirecte à travers les recommandations qu'elles génèrent.

La première méthode impliquée dans cette expérimentation est l'application du filtrage collaboratif classique par analogie avec le système MovieLens pour lequel un nouvel utilisateur doit fournir, outre des informations personnelles, des évaluations sur au moins 15 des films proposés avant de recevoir des recommandations. Pour nous rapprocher de ces conditions, nous extrayons les 15 premières évaluations dans le profil Evaluation d'un utilisateur u de U_N (voir Figure 18.1). Nous générons ensuite les recommandations pour cet utilisateur par le processus classique de filtrage collaboratif selon la corrélation de Pearson [BHK98]. Nous obtenons alors l'ensemble des premières recommandations R_{Pearson} (voir METHODE 1 dans la Figure 18.1).

Nous voulons mettre en évidence ici l'ensemble $E_{\text{Réelles}}$ des évaluations de l'utilisateur u, non utilisées pour générer des recommandations, qui sera considéré comme l'ensemble des données de référence. Cela veut dire que cet ensemble nous sert de référence dans la comparaison de qualité entre les ensembles de premières recommandations générées par plusieurs méthodes de démarrage à froid.

La deuxième méthode est la production directe des recommandations à partir des communautés démographiques G_{Age}, $G_{\text{Profession}}$ et $G_{\text{Géographie}}$ sans passer par l'étape de classification, ici inutile puisque

les valeurs du vecteur $\mathcal{P}_u$ sont déjà connues pour ces attributs. Plus précisément, nous générons trois ensembles de recommandations R_{Age}, $R_{\text{Profession}}$ et $R_{\text{Géographie}}$ par niveau d'accord dans chacun des trois espaces de communautés correspondants, et puis nous les combinons linéairement avec des pondérations égales pour ces trois sources de recommandation pour créer l'ensemble $R_{\text{Démographie}}$ (voir METHODE 2 dans la Figure 18.1). Pour le processus de recommandation par niveau d'accord, nous utilisons les seuils $S_{\text{score}} = 4$ étoiles et $S_{\text{accord}} = 25\%$.

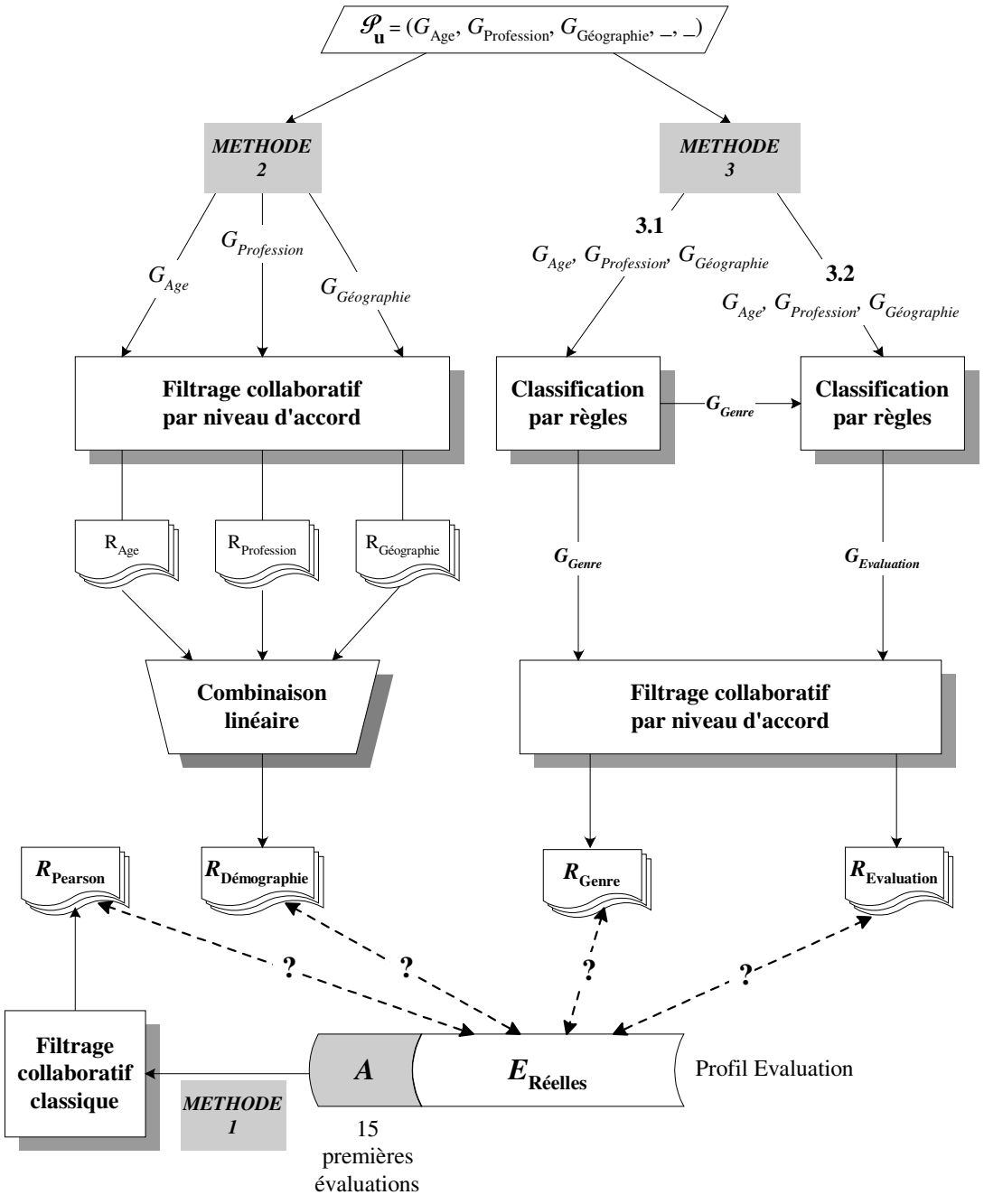


Figure 18.1 – *Expérimentation sur le démarrage à froid par comparaison de divers ensembles de premières recommandations.*

La dernière méthode de démarrage à froid à analyser est notre approche d'induction des communautés basée sur le modèle des espaces de communautés. Pour un nouvel utilisateur u faisant partie de l'ensemble U_N, nous initialisons d'abord son vecteur de positionnement avec ses données disponibles à froid. Par exemple, pour 5 critères Age, Profession, Ville, Contenu et Evaluation, il verra son vecteur initialisé ainsi : $\mathcal{P}_u$ = (26-45, Chercheur, Paris, _, _). A partir du vecteur de positionnement incomplet et des données relatives aux autres utilisateurs, nous complétons les valeurs manquantes de ce vecteur en vue de limiter l'effort de l'utilisateur, et produisons un ensemble de premières recommandations pour passer le cap du démarrage à froid.

Ce type de méthode commence par une phase d'apprentissage s'appuyant sur les données disponibles pour construire un classificateur qui est un ensemble de règles, $X \rightarrow d$ où X est un sous-ensemble d'attributs et d est l'attribut choisi comme attribut de décision, c'est-à-dire celui pour lequel on veut instancier la valeur manquante. Nous exploitons d'abord l'ensemble des vecteurs de positionnement dont le système dispose : ce sont les vecteurs des utilisateurs dans U_E arrivés plus tôt dans le système. Nous extrayons de ces données des connaissances qui permettent d'estimer la valeur manquante. Pour cela, nous choisissons d'utiliser l'algorithme C5.0, successeur de l'algorithme très connu C4.5 [Qui93], afin de construire deux classificateurs nécessaires pour les critères Contenu et Evaluation. Ce choix se justifie par la bonne performance de cet algorithme et sa capacité d'explication, qui est nécessaire pour une future interaction avec l'utilisateur.

La méthode se poursuit par une phase de classification, où on cherche une règle applicable $X \rightarrow d$ dont la prémisse X est satisfaite par les données disponibles de l'utilisateur u considéré. La valeur d de la règle trouvée remplace alors la valeur manquante dans le vecteur de positionnement de l'utilisateur u.

En pratique, voici comment nous réalisons la classification et la recommandation (voir METHODE 3 dans la Figure 18.1). En se basant sur les résultats d'analyse de qualité des critères qui montrent que le critère Contenu est meilleur que le critère Evaluation dans l'induction des communautés, les attributs Age, Profession et Géographie sont utilisés en entrée de la classification par règles C5.0 à la première étape, pour associer à l'utilisateur u une communauté G_{Genre} dans l'espace Ω_{Contenu}.

A l'étape suivante, le processus de recommandation par niveau d'accord génère pour u l'ensemble de recommandations R_{Genre} en rapport avec la position de l'utilisateur u dans la communauté G_{Genre} dans laquelle il se situe. C'est-à-dire que nous sélectionnons les films à recommander parmi ceux qui ont été évalués par cette communauté. Nous procédons à un premier filtre en ne considérant que les films évalués avec un score supérieur ou égal au seuil S_{score}, puis à un deuxième filtre où on ne conserve que les films qui ont été ainsi évalués par une part suffisante des membres de la communauté, dont la limite est fixée par l'autre seuil S_{accord}.

Finalement, nous appliquons la même méthode pour le critère Evaluation afin de produire l'ensemble de premières recommandations $R_{\text{Evaluation}}$.

Chacun des quatre ensembles de premières recommandations R_{Pearson}, $R_{\text{Démographie}}$, R_{Contenu} et $R_{\text{Evaluation}}$ sera comparé avec l'ensemble d'évaluations de référence $E_{\text{Réelles}}$.

Avant de décrire les mesures de comparaison et les résultats d'analyse, nous voulons souligner ici que l'objectif principal de notre approche est de diminuer l'effort des utilisateurs dans le démarrage à froid, et que nous nous se contentons de faire aussi bien, et pas forcément mieux, que la première méthode qui génère l'ensemble R_{Pearson},. En ce cas, nous gagnerons effectivement le rapport coût-bénéfice.

18.2.2 Mesures

Pour estimer la qualité des ensembles de premières recommandations pour un nouvel utilisateur u dans l'ensemble U_{N}, nous utilisons les mesures suivantes.

– Erreur moyenne absolue (*MAE – Mean Absolute Error*) [RAC+02] : cette mesure calcule la différence moyenne entre les prédictions de recommandations p_j du système et les scores e_j donnés réellement par l'utilisateur dans les évaluations. Dans cette analyse de méthodes de démarrage à froid, nous utilisons les deux formules suivantes :

$$M_1 = \frac{1}{|E|} \sum_j |p_j - e_j| \tag{18.1}$$

$$M_2 = \frac{1}{|R|} \sum_j |p_j - e_j| \tag{18.2}$$

où E : ensemble des évaluations de référence,
R : ensemble des recommandations à analyser.

La mesure (18.1) prend pour référence de normalisation l'ensemble E des évaluations réellement faites par les utilisateurs pour lesquels on produit des recommandations. Mais dans le cas du jeu de données MovieLens, qui provient de l'utilisation réelle d'un système de filtrage collaboratif, rien ne garantit que $E_{\text{Réelles}}$ couvre l'ensemble des informations que les utilisateurs auraient évaluées positivement si l'intégralité des documents disponibles leur avait été présentée. Nous complétons donc cette mesure « absolue » par la mesure (18.2), qui normalise le nombre d'erreurs par le nombre de recommandations produites. Cette mesure diminue l'impact d'un mauvais rappel possible de $E_{\text{Réelles}}$, et offre un bon complément, se prêtant bien à une évaluation comparative entre les différentes approches.

– Corrélation de Pearson : cette mesure est souvent utilisée dans le domaine du filtrage collaboratif [BHK98] pour le processus de prédiction de recommandations, mais on peut aussi l'utiliser pour estimer la corrélation entre les recommandations produites et les évaluations réelles :

$$M_3 = \frac{\sum_j (p_j - \overline{p})(e_j - \overline{e})}{\sqrt{\sum_j (p_j - \overline{p})^2 \sum_j (e_j - \overline{e})^2}} \tag{18.3}$$

où $\overline{e}$: moyenne des scores des évaluations faites par l'utilisateur,
$\overline{p}$: moyenne des scores prédits dans les recommandations.

Plus les taux d'erreurs M_1 et M_2 diminuent, plus la qualité de l'ensemble de recommandations R augmente. Au contraire, plus la valeur de M_3 est grande, meilleure est la qualité des recommandations.

18.3 Analyse

Les résultats des mesures M_1, M_2 et M_3 en fonction du nombre de meilleures recommandations (Top_N) à présenter aux utilisateurs sont respectivement illustrés dans Tableau 18.1, Tableau 18.2 et Tableau 18.3 où nous comparons chaque liste de recommandations aux données de référence $E_{\text{Réelles}}$. L'approche servant de base de comparaison (R_{Pearson}) est sur la première ligne, et la meilleure approche est en italique.

De façon générale, la liste de recommandations R_{Genre} générée par notre méthode domine les autres. Pourtant, bien que les taux d'erreurs de cet ensemble soient légèrement meilleurs que l'ensemble R_{Pearson} (voir les deux premières lignes dans le Tableau 18.1 et le Tableau 18.2), il importe que nous gagnions le rapport coût-bénéfice, car les utilisateurs n'ont pas besoin d'évaluer les recommandations exploratoires.

Approche \ |R|	Top_5	Top_10	Top_15	Top_20	**Moyenne**
R_{Pearson}	1,76	2,84	4,09	5,30	**3,50**
R_{Genre}	*1,38*	*2,48*	*4,09*	*5,91*	***3,47***
$R_{\text{Evaluation}}$	3,32	4,30	4,95	4,99	**4,39**
$R_{\text{Démographie}}$	2,59	4,15	4,79	6,51	**4,51**

Tableau 18.1 – *Résultat de la mesure d'erreur moyenne absolue (M_1).*

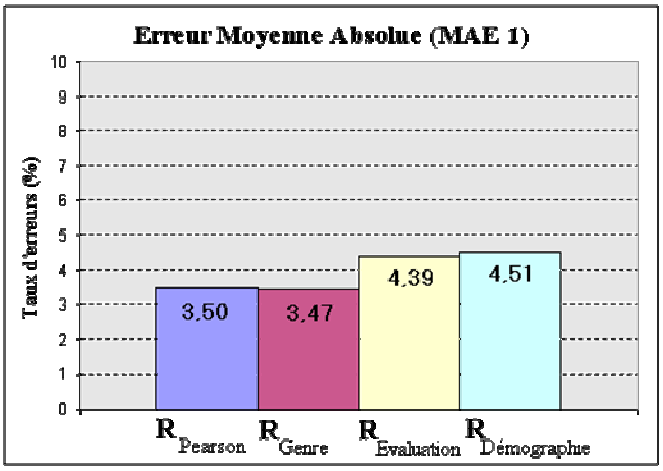


Figure 18.2 – *Résultat de la mesure d'erreur moyenne absolue (M_1).*

Logiquement, le Tableau 18.2 donne les mêmes résultats de comparaison entre les ensembles de recommandations à un facteur de $|E_{Réelles}| / |R|$ près selon l'ensemble R en question. La comparaison des taux d'erreurs entre R_{Genre} et les autres ensembles de recommandations montre l'utilité de notre approche comme une solution personnalisée pour le problème du démarrage à froid dans un système de filtrage. Néanmoins, ces bons résultats ne permettent pas de déterminer quelle part de ces performances sont dues à la production des premières recommandations, puisqu'elles dépendent de plusieurs facteurs comme la performance de la méthode de classification, la qualité du calibrage des paramètres utilisés dans le filtrage collaboratif par niveau d'accord, et l'hybridation du filtrage par des attributs démographiques.

Approche \ |R|	Top_5	Top_10	Top_15	Top_20	**Moyenne**
$R_{Pearson}$	27,97	24,02	22,83	22,45	**24,32**
R_{Genre}	*26,59*	*24,96*	*23,18*	*22,44*	***24,29***
$R_{Evaluation}$	43,45	29,54	27,59	27,45	**32,01**
$R_{Démographie}$	83,32	83,44	79,74	79,83	**81,58**

Tableau 18.2 – *Résultat de la mesure d'erreur moyenne absolue (M_2).*

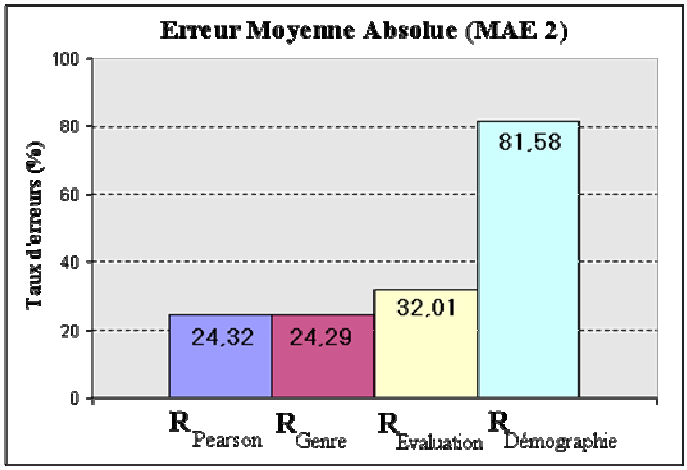


Figure 18.3 – *Résultat de la mesure d'erreur moyenne absolue (M_2).*

Les résultats du Tableau 18.3 confirment la qualité de l'ensemble R_{Genre} par rapport à l'ensemble $R_{Pearson}$ et montrent toujours la mauvaise qualité de $R_{Démographie}$. Ils rendent également compte d'une qualité relativement stable en fonction de la taille des ensembles de recommandations.

Par ailleurs, nous montrons dans 17.3, en analysant ces mêmes données, que l'attribut Contenu peut être mieux prédit par les autres attributs, que l'attribut Evaluation. Ce résultat est ici confirmé en pratique.

Approche \ |R|	Top_5	Top_10	Top_15	Top_20	**Moyenne**
$R_{Pearson}$	28,95	25,90	24,02	24,97	**25,96**
R_{Genre}	*29,92*	*30,99*	*31,53*	*32,23*	***31,17***
R_{Eval}	31,39	30,40	32,13	30,52	**31,11**
$R_{Démographie}$	18,08	19,45	20,34	19,20	**19,27**

Tableau 18.3 – *Résultat de la mesure de corrélation de Pearson (M_3).*

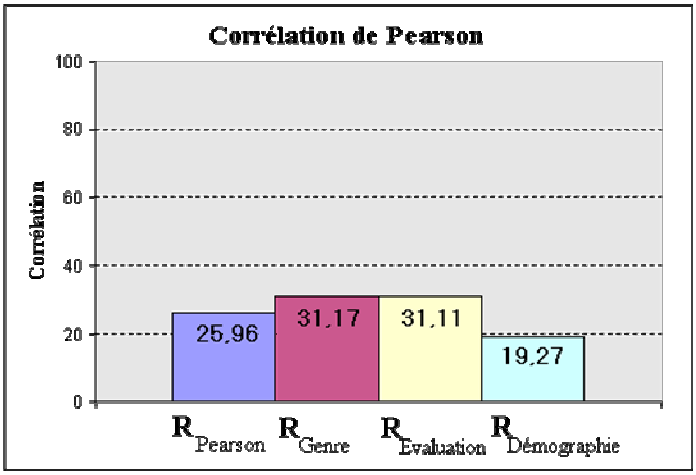


Figure 18.4 – *Résultat de la mesure de corrélation de Pearson (M_3).*

18.4 Conclusion

Pour conclure, notre méthode d'intégration de nouveaux utilisateurs donne des résultats à peine meilleurs que la technique classique prise pour base de comparaison, mais du fait de l'absence d'intervention de l'utilisateur dans notre approche, nous concluons à un rapport coût-bénéfice bien meilleur. Les performances de cette approche sont évaluées selon leur capacité à fournir, sans effort de la part de l'utilisateur, des recommandations équivalentes à celles obtenues suite à un processus de démarrage à froid classique, qui lui, requiert un effort de l'utilisateur.

Chapitre 19

Bilan

Les travaux expérimentaux présentés dans cette partie valident notre approche d'utilisation des cartes de communautés. Ils montrent que la multiplicité des critères de formation des communautés permet d'enrichir les fonctionnalités d'un système de filtrage collaboratif.

D'abord, on voit que dans un tel système, les cartes rendent complètement explicites les communautés multicritères en renforçant la perception des communautés. Ceci fournit une bonne base pour améliorer l'interaction entre utilisateurs et système et en particulier la capacité d'explication du système aux utilisateurs sur leurs communautés ainsi que sur les recommandations qu'ils reçoivent.

De plus, l'écart éventuel entre des cartes provenant de la diversité des ressources de filtrage, est potentiellement utile dans plusieurs activités du filtrage. Il pourrait être utilisé dans le positionnement interactif des utilisateurs au sein des espaces de communautés, sous réserve d'un processus tenant compte des capacités cognitives des utilisateurs.

On peut aller un peu plus loin sur cette voie en étudiant la possibilité d'adaptation du profil thématique d'un utilisateur par l'héritage des profils des communautés dont il est membre.

Par ailleurs, la multiplicité des critères nous fournit la possibilité d'une compensation entre critères. Nous montrons par l'expérience sur les données MovieLens que la relation entre les espaces de communautés des critères donne une solution efficace pour le démarrage à froid en diminuant l'effort demandé aux utilisateurs pour fournir les données nécessaires au positionnement. En particulier, les travaux expérimentaux de cette partie valident également la capacité de notre approche d'induction des communautés à personnaliser des recommandations exploratoires, en vue de proposer aux utilisateurs de meilleures communautés initiales, ce que l'on ne trouve pas dans les systèmes de filtrage courants.

Partie VI.

Conclusion

Dans cette thèse, nous nous intéressons aux systèmes de filtrage collaboratif qui, en exploitant des communautés pour envoyer des recommandations aux individus, présentent plusieurs avantages importants tels que la possibilité de découvrir de nouveaux domaines intéressants, la prise en compte de plusieurs autres dimensions de qualité d'informations que le contenu, etc. On voit de tels systèmes qui se développent quotidiennement. Parallèlement, il existe dans la littérature de nombreuses études visant à pallier certains inconvénients du filtrage collaboratif, par exemple le démarrage à froid, la masse critique et le rapport coût-bénéfice. Dans ces études, on voit que les communautés restent souvent invisibles, et elles sont considérées simplement comme des résultats intermédiaires dans le calcul de la prédiction. En d'autres termes, les communautés ne sont pas exploitées en tant que telles au sein du filtrage collaboratif.

Donc, l'objectif scientifique principal de cette thèse est d'améliorer la gestion des communautés afin de les exploiter au mieux, et par conséquent de générer de meilleures recommandations aux individus.

L'état de l'art nous montre que dans la littérature, il n'existe aucun modèle pour la gestion globale des communautés dans un système de filtrage collaboratif. Ainsi, afin d'atteindre l'objectif ci-dessus, nous proposons dans cette thèse le modèle des espaces de communautés, qui, en général, permet de gérer de façon efficace les communautés dans un système de filtrage collaboratif :

− en les rendant complètement *explicites* dans l'optique d'une interaction avec les utilisateurs,

− en exploitant la *multiplicité des critères* disponibles dans les profils des utilisateurs, pour la formation diversifiée des communautés,

− en *diminuant l'effort* des utilisateurs à fournir des informations nécessaires pour le positionnement au sein des communautés, en particulier au démarrage à froid où les utilisateurs ressentent un rapport coût-bénéfice déficitaire.

Les contributions théoriques ainsi que pratiques de cette thèse sont synthétisées dans le Chapitre 20. Finalement, les travaux futurs à court et à moyen terme sont décrits dans le Chapitre 21.

Chapitre 20

Bilan

20.1 Apports théoriques

Nous proposons dans cette thèse le *modèle des espaces de communautés* fondé sur la théorie des ensembles d'approximation, ainsi que certaines extensions de cette théorie qui conduisent à des mesures de qualité des critères dans l'induction des communautés. Les apports théoriques de notre modèle concernent en général les trois aspects évoqués dans le chapitre de la problématique, et développés tout au long de ce manuscrit : la perception des communautés, la formation des communautés et le positionnement des utilisateurs au sein des communautés (Tableau 20.1).

20.1.1 Perception des communautés

La perception des communautés est une des questions importantes dans la gestion de communautés. Elle permet non seulement de renforcer la capacité d'explication des recommandations, mais également de contribuer à résoudre le problème du rapport coût-bénéfice. De plus, la capacité de percevoir toutes les communautés sert de base à l'adaptation interactive des profils des utilisateurs.

Dimension	Apports
1. Perception des communautés	• une approche pour former et visualiser des espaces de communautés par la combinaison d'une méthode de projection et d'une méthode classique de classification, • la proposition de rendre explicites toutes les communautés afin de faciliter la perception de communautés.
2. Formation des communautés multicritères	• une analyse des étapes de formation des communautés, en particulier pour les critères complexes, • une approche intégrant une multiplicité de critères pour former des communautés, • un modèle d'espaces de communautés multicritères basé la théorie des ensembles d'approximation permettant la diversité des recommandations et la compensation des communautés pour les profils défaillants, • une représentation du polymorphisme du positionnement des utilisateurs au sein des espaces de communautés, • une méthode de filtrage collaboratif par niveau d'accord.
3. Positionnement des utilisateurs	• une méthode d'induction des communautés inconnues ou périmées d'un utilisateur afin de répondre aux problèmes du démarrage à froid, du rapport coût-bénéfice et de la masse critique, • une extension de la théorie des ensembles d'approximation comprenant les mesures de qualité de critère afin d'améliorer la performance de l'induction des communautés, • une base pour l'adaptation de profils par héritage des caractéristiques des communautés.

Tableau 20.1 – *Table récapitulative des apports théoriques de la thèse*.

Nous proposons donc dans cette thèse de rendre complètement explicites les communautés dans un système de filtrage collaboratif. Pour ce faire, nous proposons une combinaison de l'algorithme des fourmis artificielles et l'algorithme des K-moyennes.

20.1.2 Formation des espace de communautés

En ce qui concerne la formation des communautés, nous proposons une approche de formation multiple selon les critères disponibles dans les profils des utilisateurs. Nous donnons également une analyse des étapes de formation des communautés selon un critère donné.

Au niveau formel, nous proposons une formalisation des « espaces de communautés » en utilisant des relations d'équivalence. En particulier, nous donnons une formalisation du polymorphisme de positionnement des utilisateurs à travers les « vecteurs de positionnement ».

Enfin, dans le contexte de la multiplicité des critères pour la formation des communautés, nous proposons le filtrage collaboratif par niveau d'accord pour la diversification des recommandations.

20.1.3 Positionnement des utilisateurs au sein des espaces de communautés

Notre approche de positionnement des utilisateurs par l'induction, en exploitant les relations entre espaces de communautés permet de diminuer l'effort des utilisateurs et de soulager les difficultés du système dans le positionnement. Par conséquent, elle donne une réponse pertinente aux problèmes du démarrage à froid, du rapport coût-bénéfice et de la masse critique.

Par exemple, nous pensons qu'utiliser les données disponibles à froid peut permettre de compenser partiellement les données non fournies initialement par les utilisateurs, qui sont utiles dans le positionnement des utilisateurs au sein des communautés. Notre approche de démarrage à froid s'apparente donc à celle des profils-type, avec l'optique de limiter au maximum le nombre et la complexité des informations à demander aux utilisateurs, sachant qu'à terme cette approche est destinée à être combinée avec l'une ou l'autre des approches classiques de façon à leur conférer un meilleur rapport coût/bénéfice. Les données disponibles à froid nous permettent d'attribuer au nouvel utilisateur une communauté pour elles-mêmes (communautés d'âge, de profession etc.) mais également dans l'espace des communautés du critère Contenu, ou du critère Evaluation.

Par ailleurs, la méthode proposée peut s'appliquer dans d'autres cadres que celui du démarrage à froid. En effet, au cours de l'utilisation d'un système de recommandation, les utilisateurs rencontrent souvent des situations où leurs communautés sont périmées ou leur profil ne reflète plus correctement leur besoin, par exemple lors d'un changement important des centres d'intérêt. On peut alors considérer que le profil, ou le « vecteur de positionnement », n'est que partiellement fiable. L'approche présentée ici peut alors être appliquée pour corriger automatiquement les valeurs douteuses de ce vecteur afin de servir de point de départ à une évolution plus radicale du profil de l'utilisateur.

20.2 Apport global pour les systèmes de filtrage collaboratif

Plus généralement, les travaux de cette thèse contribue à une solution pour les quatre problèmes suivants du filtrage collaboratif : le démarrage à froid, la masse critique, le rapport coût-bénéfice et l'expression limitée du besoin.

Démarrage à froid. Pour ce phénomène se produisant en début d'utilisation du système, notre approche d'induction des communautés permet au système d'intégrer de nouveaux utilisateurs en exploitant simplement les informations faciles à fournir.

Masse critique. Vu que notre approche d'induction permet de positionner un utilisateur dans l'espace des communautés par le critère de l'historique des évaluations sans même lui demander de fournir des évaluations, cette thèse donne une réponse au problème lié au nombre d'évaluations en commun entre les utilisateurs afin de les positionner dans l'espace concerné.

Rapport coût-bénéfice. En raison de la capacité de diminuer l'effort des utilisateurs dans les activités du filtrage collaboratif, notre approche soulage considérablement le problème du rapport

coût-bénéfice. Les utilisateurs n'ont plus besoin de se demander si leur effort sera payé en retour à court ou à moyen terme.

Expression limitée du besoin. Finalement, le problème de l'expression limitée du besoin, en particulier dans la situation de changement de besoin, peut être pallié par l'approche d'interaction entre utilisateurs et système via la nouvelle modalité de communautés ainsi que par le support du polymorphisme de positionnement, par exemple le système peut prédire, pour un utilisateur en difficulté, une nouvelle communauté dans l'espace par le critère de l'historique des évaluations non seulement en utilisant ses évaluations passées mais aussi en exploitant ses autres communautés actuelles.

20.3 Apports pratiques

20.3.1 Mise en œuvre du modèle des espace de communautés

Notre modèle des espaces de communautés permet d'étendre les fonctionnalités des systèmes de filtrage collaboratif, en les rendant capables de gérer des communautés multicritères explicites, et de mieux exploiter les différents critères selon la situation rencontrée en mesurant leur qualité en tant que critère clé ou décision. En effet, en se basant sur ce modèle, on peut construire une méthode d'exploitation des données « disponibles à froid » pour améliorer l'intégration de nouveaux utilisateurs au système. De plus, notre modèle combiné avec l'utilisation des cartes de communautés permettra à terme d'enrichir l'interaction avec les utilisateurs afin de surmonter les problèmes classiques d'exploitation des systèmes de recommandation. En outre, la multiplicité des critères pour la formation des communautés et le filtrage collaboratif par niveau d'accord facilitent la diversification des recommandations.

Dans ce sens, nous proposons dans cette thèse la plateforme COCoFil2 comme la mise en œuvre de notre modèle des espaces de communautés dans un système de filtrage collaboratif.

20.3.2 Validation du modèle proposé sur un jeu de données réelles de taille importante

Il est par ailleurs à noter que les aspects principaux de notre modèle ont été validés, grâce aux résultats concrets d'expérimentations sur un jeu de données réelles de taille importante.

Chapitre 21

Perspectives

21.1 Travaux à court terme

Nos travaux à court terme concernent notamment la validation approfondie et diversifiée du modèle des espaces de communautés actuel afin de développer un système de filtrage collaboratif réel qui se base sur la plateforme COCoFil2. Cette validation se réalisera selon les dimensions suivantes : passage à l'échelle, hybridation des filtrages, extension des domaines d'application et étude des capacités cognitives d'utilisateurs.

a) Passage à l'échelle. En général, nous souhaitons expérimenter notre modèle sur des jeux de données réelles de grande taille quant au nombre des utilisateurs, des objets et des évaluations, ainsi que des critères dans les profils. Dans le cadre de cette thèse, nous avons déjà expérimenté notre modèle des espaces de communautés sur le second jeu de données de MovieLens contenant 943 utilisateurs et 100 000 évaluations sur 1 682 films. Ainsi, nous souhaitons valider encore le modèle sur le jeu MovieLens restant de taille plus grande, contenant 6 040 utilisateurs et 1 000 000 évaluations sur 3 883 films. Un autre jeu de données envisagé pour l'expérimentation supplémentaire est le jeu Book-Crossing[13] construit par C.-N. Ziegler à l'université de Freiburg. Ce jeu contient 1 149 780 évaluations explicites/implicites, données par 278 858 utilisateurs sur 271 379 ouvrages. Le nombre

[13] http://www.informatik.uni-freiburg.de/~cziegler/BX

d'évaluations dans ces deux jeux cibles est presque le même, mais le nombre d'utilisateurs et d'objets dans le jeu de données Book-Crossing est beaucoup plus grand par rapport au jeu MovieLens.

b) Hybridation des filtrages. Nous rappelons que l'objectif principal de cette thèse est la gestion des communautés dans un système de filtrage collaboratif, et nous avons déjà expérimenté les principaux aspects de notre modèle pour cette tâche de gestion. Malgré tout, l'ultime objectif des systèmes de filtrage d'information est de générer de meilleures recommandations aux utilisateurs. Ainsi, nous avons envie d'expérimenter la combinaison de notre approche des espaces de communautés, incluant la méthode de filtrage collaboratif par niveau d'accord, avec d'autres techniques dans un système hybride afin d'analyser l'impact de notre approche sur la qualité et la diversité des recommandations.

c) Extension des domaines d'application. Les travaux expérimentaux actuels dans la littérature se trouvent notamment dans le commerce électronique et les loisirs. Dans les travaux futurs, nous envisageons de valider notre modèle dans d'autres domaines d'application. Par exemple, la documentation académique est un domaine traditionnel, où on dispose de nombreux critères pour former des communautés, tels que l'affiliation, la fonction occupée, les domaines de recherche, les co-auteurs, les langues maternelle et étrangères, les préférences de qualité de documents, etc., outre les informations personnelles et l'historique des évaluations. La formation à distance (*eLearning*) est également un domaine intéressant, où la formation de communautés d'étudiants et d'enseignants est une question pertinente pour la construction de cours personnalisés et pour le tutorat [Bar03, Cro98, SSF02, Wen98]. Cette extension nous permettra donc d'expérimenter la multiplicité et la diversité des critères de formation des communautés.

d) Capacité cognitive des utilisateurs. Dans le Chapitre 11, nous présentons notre approche d'utilisation des cartes de communautés en 2D pour la perception des communautés. Afin que les communautés puissent devenir une modalité d'interaction entre utilisateurs et système, nous souhaitons mener une étude pour évaluer la capacité cognitive des utilisateurs à appréhender ces cartes ainsi que la pertinence de l'information qu'elles présentent. Pour cette dernière, nous envisageons d'étudier la capacité des utilisateurs à percevoir les communautés dans un espace à travers des profils d'évaluations typiques de ces communautés. Des pistes pour construire ce profil typique d'évaluations d'une communauté sont d'ores et déjà envisagées : créer le profil du « centre de gravité » ou raffiner le profil d'un représentant de la communauté, appliquer les méthodes de sélection comme pour les recommandations exploratoires au démarrage à froid. Ensuite, nous essayerons de prédire l'intérêt des utilisateurs pour les documents à partir de chacune des communautés et de comparer ces prédictions aux véritables évaluations fournies ultérieurement par les utilisateurs, en utilisant des métriques de type rappel/précision.

21.2 Travaux à moyen terme

Les travaux à moyen terme sont les études d'évolution des communautés et des profils en exploitant les relations entre des espaces de communautés, et en particulier l'amélioration de notre modèle des espaces de communautés.

21.2.1 Evolution des communautés

Les travaux actuels nous conduisent à étudier l'évolution des communautés. Par exemple, nous avons envie d'analyser la situation où les évaluations de l'utilisateur, qu'elles soient positives ou négatives, ne permettent pas de mettre à jour son profil Contenu, alors que le processus d'induction des communautés prédit pour cette personne une nouvelle communautés dans l'espace $\Omega_{Contenu}$ à partir de ses autres communautés. Cette analyse pourra servir à la tâche de détection du changement d'intérêt des utilisateurs.

Par ailleurs, nous espérons que l'analyse des relations entre les espaces $\Omega_{Evaluation}$ et $\Omega_{Contenu}$ nous permettra aussi de déterminer le moment où les évaluations de l'utilisateur n'apportent plus de changement significatif à son profil Contenu. Dans ce cas, pour adapter le profil, il faut s'en remettre à l'interaction à travers des cartes plutôt qu'aux évaluations. Ceci donnerait un élément pour la prévention du découragement des utilisateurs [Gal05].

21.2.2 Adaptation des profils

Dans un système de filtrage hybride, qui combine le filtrage collaboratif et le filtrage basé sur le contenu, les utilisateurs reçoivent des recommandations en fonction de leurs profils Contenu et de leurs communautés multicritères. Pour adapter les profils, l'approche la plus populaire est de les raffiner au fur et à mesure que les utilisateurs fournissent leurs évaluations [MLD03]. Néanmoins, un changement dans leur besoin d'information au niveau des centres d'intérêt n'est pas toujours bien pris en compte. C'est peut-être à cause de la lenteur du processus d'adaptation : il faut que l'utilisateur évalue parfois beaucoup de documents pour que se produise un changement significatif dans son profil. En particulier, des difficultés apparaissent lorsque l'utilisateur ne reçoit pas les recommandations qui lui permettent d'exprimer son changement d'intérêt, ou lorsque les évaluations de l'utilisateur n'apportent aucune nouvelle information pour adapter son profil. Donc, comme le montre la Figure 21.1, nous avons envie d'étudier la possibilité d'adapter les profils, en exploitant des relations entre les espaces dans la table de communautés.

Voici un exemple de scénario : le système détecte une accumulation de retours de pertinence négatifs de l'utilisateur sur les recommandations émanant du profil Contenu, ce qui traduit une situation susceptible de le décourager. Il y aurait deux possibilités pour cette situation délicate.

D'une part, le système peut entamer un dialogue avec l'utilisateur en lui montrant sa position parmi les communautés dans la cartes $\Omega_{Contenu}$, et en lui proposant de se rattacher à une communauté qui lui semble plus proche que celle dans laquelle il se situe actuellement. Cette possibilité pourrait suivre celle sur l'étude de la capacité cognitive d'utilisateurs déjà évoquée. D'autre part, le système peut également appliquer le processus d'induction des communautés pour prédire la nouvelle communauté de l'utilisateur dans l'espace $\Omega_{Contenu}$, à partir de ses autres communautés.

Une fois sa communauté dans $\Omega_{Contenu}$ déterminée, quelque soit la méthode appliquée, l'utilisateur pourrait hériter du profil Contenu typique de cette communauté.

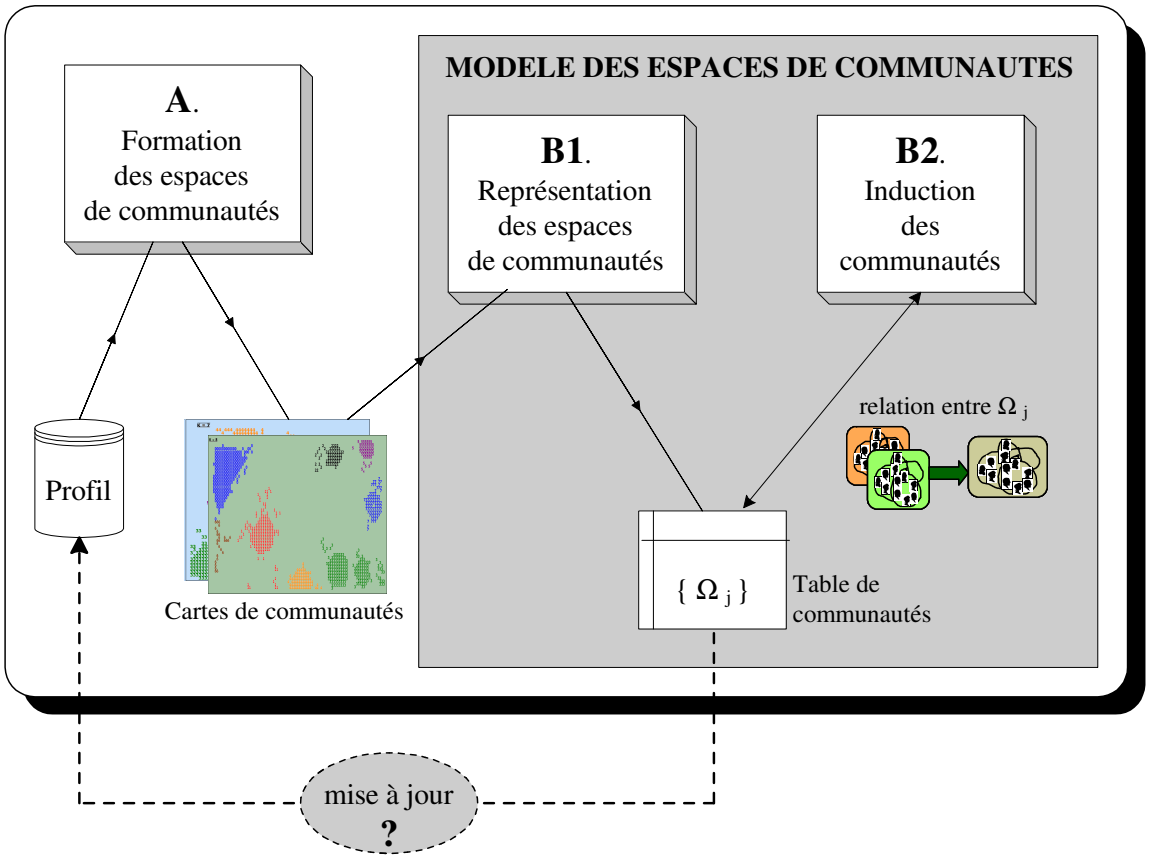


Figure 21.1 – *Approche d'adaptation de profils en exploitant des espaces de communautés.*

Il est à noter que cette méthode d'adaptation des profils nécessite une base de connaissances sur les situations problématiques rencontrées incluant plusieurs dimensions comme la cause, la nature de profil, la qualité ou la consistance de la table de communautés concernée dans l'induction, etc.

21.2.3 Amélioration du modèle des espaces de communautés

21.2.3.1 Enrichissement des mesures de qualité de critère dans l'induction des communautés

Dans les travaux futurs, nous envisageons de faire varier les mesures de qualité de décision tout en restant dans le cadre de la théorie des ensembles d'approximation, notamment en utilisant non seulement les règles certaines dans les régions positives mais aussi les règles possibles dans les bornes supérieures [Paw82, 84, 04] : elles pourraient donner aux utilisateurs la possibilité de découvrir des communautés potentiellement intéressantes dans une optique exploratoire.

De plus, nous aimerions aussi étudier la notion de « borne inférieure paramétrée » [ZY04], qui pourrait être utilisée dans les systèmes dont la majorité des critères ne donnent pas, en tant que décision, des tables de communautés de haute consistance. Le principe est qu'une règle u est « quasi certaine » ssi sa classe d'équivalence par P, $[u]_P$, est « presque » incluse dans sa classe d'équivalence

par D, $[u]_D$ (cf. (8.5)). Cette notion permettrait de contrôler la taille des régions positives dans le but de rendre plus discriminante la consistance des tables.

21.2.3.2 Application étendue de la théorie des ensembles d'approximation

Selon le principe de base de la théorie des ensembles d'approximation, la table de communautés, ou table de décision, utilisée pour induire des communautés dans un vecteur de positionnement imparfait, doit être complète. Dans la réalité, il se peut qu'une telle table contienne elle-même des valeurs manquantes. Pour résoudre ce problème, on peut prétraiter la table avant l'induction, en utilisant des méthodes de remplissage de données [GH00]. Néanmoins, cette approche risque de faire disparaître l'incertitude intrinsèque des données. Il existe en même temps des études qui proposent des extensions de la théorie des ensembles d'approximation pour répondre au problème de données manquantes ci-dessus, en préservant toujours l'incertitude des données.

En ce moment, nous nous intéressons, pour nos travaux à moyen terme, aux études de généralisation des relations d'indiscernabilité pour traiter les tables de décision incomplètes.

A titre d'exemple, Kryszkiewicz propose d'utiliser une valeur spéciale appelée « inconnue » (*unknown*), et symbolisée par *, pour les valeurs manquantes de tous les attributs de condition dans la table de décision [Kry95, Kry99]. Ensuite, l'auteur remplace la relation d'indiscernabilité dans la théorie des ensembles d'approximation par la relation tolérante $\mathcal{R}_P$ comme suit.

Soit un sous-ensemble d'attributs de condition P.

$$\forall u, u' \in U, \quad u \,\mathcal{R}_P\, u' \;\Leftrightarrow\; (T[u, a] = T[u', a]) \vee (T[u, a] = *) \vee (T[u', a] = *), \forall a \in P \qquad (21.1)$$

Cette nouvelle relation ne vérifie que deux propriétés : la réflexivité et la symétrie, et pas la transitivité.

Plus récemment, Grzymala-Busse propose la relation caractéristique (*characteristic relations*) pour remplacer la relation d'indiscernabilité [Grz03, Grz04, Grz05, GS04]. D'abord, il distingue deux types, ou raisons, des valeurs manquantes des attributs de condition dans une table de décision : « inconnue » (*lost*) et « inintéressante » (*do not care*), qui sont symbolisées par * et ?, respectivement. Toute prémisse doit contenir au moins une valeur spécifiée ($\notin \{*, ?\}$).

Alors, la relation caractéristique $\mathcal{R}_P$ est définie comme suit.

$$u \,\mathcal{R}_P\, u' \;\Leftrightarrow\; (T[u, a] = T[u', a]) \vee (T[u, a] = *) \vee (T[u', a] = *), \forall a \in P : T[u, a] \neq ? \qquad (21.2)$$

Si la table ne contient que des valeurs manquantes de type ?, la relation caractéristique est définie par :

$$u \,\mathcal{R}_P\, u' \;\Leftrightarrow\; (T[u, a] = T[u', a]), \forall a \in P : T[u, a] \neq ? \qquad (21.3)$$

De façon similaire, si la table ne contient que des valeurs manquantes de type *, la relation caractéristique est définie comme la relation tolérante.

En résumé, nous souhaitons mener des études pour appliquer ces extensions de la théorie des ensembles d'approximation à notre modèle des espaces de communautés, en particulier dans la définition des mesures de qualité des critères dans l'induction des communautés, afin de l'adapter aux divers contextes applicatifs.

Liste de publications personnelles

Nguyen A.-T., Denos N., Berrut C., Modèle des espaces de communautés orienté vers la diversité de recommandations pour les systèmes de filtrage, *Revue en Sciences du Traitement de l'Information, Information – Interaction – Intelligence (I3)*, 2006 (*à apparaître*).

Nguyen A.-T., Denos N., Berrut C., Exploitation des données « disponibles à froid » pour améliorer le démarrage à froid dans les systèmes de filtrage d'information, *Actes du 24ème Congrès annuel de l'Informatique des Organisations et Systèmes d'Information et de Décision (INFORSID'06)*, Hammamet, Tunisie, 2006, p. 81-95.

Nguyen A.-T., Denos N., Berrut C., Modèle d'espaces de communautés basé sur la théorie des ensembles d'approximation dans un système de filtrage hybride, *Actes de la 3ème COnférence en Recherche Information et Applications (CORIA'06)*, Lyon, France, 2006, p. 303-314.

Nguyen A.-T., Denos N., Berrut C., Cartes de communautés pour l'adaptation interactive de profils dans un système de filtrage d'information, *Actes du 23ème Congrès annuel de l'Informatique des Organisations et Systèmes d'Information et de Décision, INFORSID'05*, Grenoble, France, 2005, p. 253-268.

Denos N., Berrut C., Gallardo-Lopez L., Nguyen A-T., COCoFil : Une plateforme de filtrage collaboratif orientée vers la communauté, *Actes de la 1ère COnférence en Recherche d'Information et Applications (CORIA'04)*, Toulouse, France, 2004, p. 9-26.

Bibliographie

[APG+04] Azzag H., Picarougne F., Guinot C., Venturini G., Un survol des algorithmes biomimétiques pour la classification, *Classification et fouille de données*, *RNTI-C-1*, Cépaduès, 2004.

[AS99] Amato G., Straccia U., User Profile Modeling and Applications to Digital Libraries, *Proceedings of the 3rd European Conference on Research and Advanced Technology for Digital Libraries (ECDL'99), LNCS*, vol. 1696, France, 1999, p. 184-197.

[Bar03] Barab S.-A., An Introduction to the Special Issue, *Designing for Virtual Communities in the Service of Learning*, vol. 19 (3), 2003, p. 197-201.

[BC92] Belkin N.-J., Croft W.-B., Information Filtering and Information Retrieval: Two Sides of the Same Coin?, *Communications of the ACM*, vol. 35 (12), 1992, p. 29-38.

[Ber02] Berkhin P., Survey of Clustering Data Mining Techniques, Technical Report, *Accrue Software*, 2002.

[BHK98] Breese J.-S., Heckerman D., Kadie C., Empirical Analysis of Predictive Algorithms for Collaborative Filtering, *Proceedings of the 14th Conference on Uncertainty In Artificial Intelligence (UAI'98)*, Wisconsin, USA, 1998, p. 43-52.

[BK05] Bouzeghoub M., Kostadinov D., Personnalisation de l'information : Aperçu de l'état de l'art et définition d'un modèle flexible de profils, *Actes de la 2ème COnférence en Recherche d'Information et Applications (CORIA'05)*, France, 2005, p. 201-218.

[BN72] Bell C., Newby H., Theories of Community, *Community Study: An Introduction to the Sociology of the Local Community*, New York, Praeger Publishers, 1972.

[Bur02] Burke R., Hybrid Recommender Systems: Survey and Experiments, *Journal of Personalization Research, User Modeling and User-Adapted Interaction*, vol. 12 (4), 2002, Kluwer Academic Publishers, p. 331-370.

[C5.0] C5.0, Release 2.02, September 2005, http://www.rulequest.com/see5-info.html.

[Cam97] Campbell J.-P., Speaker Recognition: *A Tutorial, Proceedings of the IEEE*, vol. 85 (9), 1997, p. 1437-1462.

[Can02] Canny J., Collaborative filtering with privacy via factor analysis, *Proceedings of the 25th International ACM Conference on Research and Development in Information Retrieval (SIGIR'02)*, Tampere, Finland, 2002.

[CGM+99] Claypool M., Gokhale A., Miranda T., Murnikov P., Netes D., Sartin M., Combining Content-Based and Collaborative Filters in an Online Newspaper, *Proceedings of the 22nd*

International ACM Conference on Research and Development in Information Retrieval (SIGIR'99), USA, 1999.

[CHM+01] Coppock S., He A., Mazlack L., Zhu Y., Experiments in Rough Set Based Data Mining, *Proceedings of the Artificial Neural Networks In Engineering (ANNIE'01)*, USA, 2001, p. 339-344.

[CLW+01] Claypool M., Le P., Waseda M., Brown D., Implicit interest indicators, *Proceedings of the 6th International Conference on Intelligent User Interfaces (IUI'01)*, New Mexico, USA, 2001, p. 33-40.

[CM03] Coppock S., Mazlack L., Rough Sets Used in the Measurement of Similarity of Mixed Mode Data, *Proceedings of the 22th Conference of the North American Fuzzy Information Processing Society*, USA, 2003.

[CN89] Clark P., Niblett T., The CN2 induction algorithm, *Machine Learning*, vol. 3 (4), 1989, p. 261-284.

[Cro98] Cross K. P., Why Learning Communities? Why Now?, *About Campus*, vol. 3 (3), 1998, p. 4-11.

[CS02] Cöster R., Svensson M., Inverted file search algorithms for collaborative filtering, *Proceedings of the 25th International ACM Conference on Research and Development in Information Retrieval (SIGIR'02)*, Tampere, Finland, 2002.

[CSP03] Carenini G., Smith J., Poole D., Towards more Conversational and Collaborative Recommender Systems, *Proceedings of the International Conference of Intelligent User Interfaces (IUI'03)*, Florida, USA, 2003, p. 12-18,

[DBG+04] Denos N., Berrut C., Gallardo-Lopez L., Nguyen A.-T., COCoFil : Une plateforme de filtrage collaboratif orientée vers la communauté, *Actes de la 1ère COnférence en Recherche d'Information et Applications (CORIA'04)*, Toulouse, 2004, France, p. 9-26.

[DBT00] Dorigo M., Bonabeau E., Theraulaz G., Ant algorithms and stigmergy, *Future Generation Computer Systems (FGCS)*, vol. 16, Elsevier, 2000, p. 851-871.

[DGF+90] Deneubourg J.-L., Goss S., Franks N., Sendova-Franks A., Detrain C., Chrétien L., The dynamics of collective sorting: robot-like and ant-like robots, *Proceedings of the 1st Conference on Simulation of Adaptive Behavior (SAB'90)*, USA, 1990, p. 356-365.

[FHH+00] Fisher D., Hildrum K., Hong J., Newman M., Thomas M., Vuduc R., SWAMI: A Framework for Collaborative Filtering Algorithm Developmen and Evaluation, *Proceedings of the 23th International ACM Conference on Research and Development in Information Retrieval (SIGIR'00)*, Athens, Greece, 2000.

[Gal05] Gallardo-López M.-L., Accès à l'Information par un Système de Filtrage Collaboratif Contrôlé, Thèse, Université Joseph Fourier, Grenoble, France, 2005.

[Gal15] Galpin C.-J., The Social Anatomy of an Agricultural Community, *Research Bulletin*, vol. 34, University of Wisconsin Agricultural Experiment Station, 1915.

[GBD03] Gallardo-López M.-L., Berrut C., Denos N., Une approche pour le contrôle de la qualité des Systèmes de Filtrage Collaboratif, *Manifestation de Jeunes Chercheurs STIC (MAJESCTIC'03)*, France, 2003.

[GH00] Grzymala-Busse J.-W., Hu M., A Comparison of Several Approaches to Missing Attribute Values in Data Mining, *Proceedings of the 2nd Conference on RS and Current Trends in Computing (RSCTC'00)*, Canada, 2000.

[GON+92] Goldberg D., Oki B., Nichols D., Terry D.-B., Using Collaborative Filtering to Weave an Information Tapestry, *Communications of the ACM*, vol. 35 (12), 1992, p. 61-70.

[GPS02] Greco S., Pawlak Z., Slowinski R., Generalized Decision Algorithms, Rough Inference Rules, and Flow Graphs, *Proceedings of the 3rd International Conference on Rough Sets and Current Trends in Computing (RSCTC'02)*, PA, USA, 2002, p. 93-104.

[Grz97] Grzymala-Busse J.-W., A new version of the rule induction system LERS, *Fundamenta Informaticae*, vol. 31, 1997, p.27-39.

[Grz03] Grzymala-Busse J.-W., Rough Set Strategies to Data with Missing Attribute Values, *Proceedings of the Workshop on Foundations and New Directions in Data Mining*, *associated with the 3rd International Conference on Data Mining*, USA, 2003, p.56-63.

[Grz04] Grzymala-Busse J.-W., Data with Missing Attribute Values: Generalization of Indiscernibility Relation and Rule Induction, *Transactions on Rough Sets I, LNCS 3100 Springer*, 2004, p. 78-95.

[Grz05] Grzymala-Busse J.-W., Incomplete Data and Generalization of Indiscernibility Relation, Definability, and Approximations, *Proceedings of the 10th International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing (RSFDGrC'05)*, Canada, 2005, p. 244-253.

[GS04] Grzymała-Busse J.-W., Siddhaye S., Rough Set Approaches to Rule Induction from Incomplete Data, *Proceedings of the 10th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU'04)*, Perugia, Italy, 2004, vol. 2, p. 923-930.

[GSK+99] Good N., Schafer J.-B., Konstan J.-A., Borchers A., Sarwar B., Herlocker J., Riedl J., Combining collaborative filtering with personal agents for better recommendations, *Proceedings of the 16th National Conference on Artificial Intelligence*, Orlando, USA, 1999, p. 439- 446.

[Gué03] Guénoche A., Partitions optimisées selon différents critères : évaluation et comparaison, *Revue Mathématiques et sciences humaines/Mathematics and social sciences*, vol. 161, 2003, p.41-58.

[Ham97] Hamman R., Introduction to Virtual Communities Research and Cybersociology Magazine, *Cybersociology Magazine*, Issue 2, Online Virtual Communities, 1997.

[HD59] Harper E.-B., Dunham A., Community Organization in Action, *New York: Association Press*, 1959.

[Heb49] Hebb D.-O., The Organization of Behavior, *John Wiley & Sons New York*, 1949 (Introduction and Chapter 4 reprinted in Anderson & Rosenfeld, 1988, p. 45-56).

[Her00] Herlocker J.-L., *Understanding and Improving Automated Collaborative Filtering Systems*, Ph.D Dissertation, University of Minnesota, 2000.

[Hil55] Hillery G.-Jr., Definitions of Community: Areas of Agreement, *Rural Sociology*, vol. 20, 1955, p. 111-122.

[HKD03] Handl J., Knowles J., Dorigo M., On the performance of ant-based clustering, *Proceedings of the 3rd International Conference on Hybrid Intelligence Systems*, Australia, 2003.

[HKJ+99] Herlocker J.-L., Konstan A.-J., Borchers A., Riedl J., An Algorithmic Framework for Performing Collaborative Filtering, *Proceedings of the 22nd International ACM Conference on Research and Development in Information Retrieval (SIGIR'99)*, USA, 1999, p. 230-237.

[HKR00] Herlocker J.-L., Konstan J.-A., Riedl J., Explaining Collaborative Filtering Recommendations, *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work (CSCW'00)*, Pennsylvania, USA, 2000, p. 241-250.

[HKT+04] Herlocker J.-L., Konstan J.-A., Terveen L., Riedl J., Evaluating Collaborative Filtering Recommender Systems, *ACM Transactions on Information Systems (TOIS)*, vol. 22 (1), 2004, p. 5-53.

[JCS04] Jin R., Chai J.-Y., Si L., An automatic weighting scheme for collaborative filtering, *Proceedings of the 27th International ACM Conference on Research and Development in Information Retrieval (SIGIR'04)*, Sheffield, UK, 2004, p. 337-344.

[JG04] Jones Q., Grandhi S.-A., Supporting Proximate Communities with P3-Systems: Technology for Connecting People-To-People-To-Geographical-Places, *The Interaction Society: Practice, Theories, & Supportive Technologies*, Edited by M. Weiberg, Idea Group, Inc. New York, 2004.

[JKP94] John G.-H., Kohavi R., Pfleger K., Irrelevant features and the subset selection problem, *Proceedings of the 11th International Conference on Machine Learning*, CA, USA, 1994.

[JMF99] Jain A.-K., Murty M.-N., Flynn P.-J., Data Clustering: A Review, *ACM Computing Surveys*, vol. 31 (3), 1999, p. 264-323.

[JS03] Jensen R., Shen. Q., Finding rough set reducts with ant colony optimization, *Proceedings of the UK Workshop on Computational Intelligence,* 2003, p. 15-22.

[JSZ+03] Jin R., Si L., Zhai C., Callan J., Collaborative Filtering with Decoupled Models for Preferences and Ratings, *Proceedings of the 12th International Conference on Information and Knowledge Management (CIKM'03)*, Louisiana, USA, 2003.

[KBR84] Kononenko I., Bratko I., Roskar E., Experiments in automatic learning of medical diagnostic rules, Technical report, *Jozef Stefan Institute,* Ljubljana, 1984.

[KM01] Kohrs A., Merialdo B., Improving Collaborative Filtering for New Users by Smart Object Selection, *Proceedings of International Conference on Media Features (ICMF)*, Italy, 2001.

[Koh97] Kohonen T., Self-Organizing Maps, *Springer Series in Information Sciences*, vol. 30, 1997.

[KPS98] Komorowski J., Polkovski L., Skowron A., Rough Sets: A Tutorial, 1998.

[Kru97] Krulwich B., Lifestyle Finder: Intelligent user profiling using large-scale demographic data, *AI Magazine*, vol. 18 (2), 1997, p. 37-45.

[Kry95] Kryszkiewicz M., Rough set approach to incomplete information systems, *Proceedings of the 2nd Annual Joint Conference on Information Sciences*, NC, USA, 1995, p. 194-197.

[Kry99] Kryszkiewicz M., Rules in incomplete information systems, *Information Sciences*, vol. 113, 1999, p. 271-292.

[KSS97a] Kautz H., Selman B., Shah M., ReferralWeb: Combining Social Networks and Collaborative Filtering, *Communications of the ACM*, vol. 40 (3), 1997, p. 63-65.

[KSS97b] Kautz H., Selman B., Shah M., The Hidden Web, *AI Magazine*, vol. 18 (2), 1997, p. 27-36.

[Lan95] Lang K., NewsWeeder: Learning to Filter Netnews, *Proceedings of the 12th International Conference on Machine Learning (ICML'95)*, CA, USA, 1995, p. 331-339.

[LF94] Lumer E., Faieta B., Diversity and Adaptation in Populations of Clustering Ants, From Animals to Animats 3: *Proceedings of the 3rd International Conference on Simulation of Adaptive Behavior (SAB'94)*, 1994, p. 501-508.

[Lie95] Lieberman H., Letizia: An agent that assists web browsing, *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI-95)*, Canada, 1995, p. 924-929.

[LMV02] Labroche N., Monmarché N., Venturini G., A new clustering algorithm based on the chemical recognition system of ants, *Proceedings of the 15th European Conference on Artificial Intelligence (ECAI '02)*, 2002, p. 345-349.

[MAS+02] Middleton S.-E., Alani H., Shadbolt N.-R., De Roure D.-C., Exploiting Synergy Between Ontologies and Recommender Systems, *Proceedings of the 11th International World Wide Web Conference (WWW'02), International Workshop on the Semantic Web*, Hawaii, USA, 2002.

[Maz99] Mazlack L., Softly Focusing On Data, *Proceedings of the 18th Conference of the North American Fuzzy Information Processing Society (NAFIPS'99)*, USA, 1999.

[McQ67] McQueen J., Some methods for classification and analysis of multivariate observations, *Proceedings of the 5th Berkeley Symposium on Mathematical, Statistics and Probability*, 1967, p. 281-298.

[ME95] Maltz D., Ehrlich E., Pointing the way: Active collaborative filtering, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'95)*, USA, 1995, p. 202-209.

[Mit97] Mitchell T.-M., Machine Learning, *MIT Press and McGraw-Hill*, 1997.

[MKR03] Mirza B. J., Keller B. J., Ramakrishnan N., Studying Recommendation Algorithms by Graph Analysis, *Journal of Intelligent Information Systems*, vol. 20 (2), 2003, p. 131-160.

[MLD03] Montaner M., López B., De La Rosa J.-L., A Taxonomy of Recommender Agents on the Internet, *Artificial Intelligence Review*, vol. 19, 2003, Kluwer Publishers, p. 285-330.

[MMN02] Melville P., Mooney R.-J., Nagarajan R., Content-Boosted Collaborative Filtering for Improved Recommendations, *Proceedings of the 18th National Conference on Artificial Intelligence (AAAI'02)*, Edmonton, Canada, 2002, p. 187-192.

[MovieLens] MovieLens, http://movielens.umn.edu, http://www.grouplens.org.

[MP00] Miyahara K., Pazzani M.-J., Collaborative Filtering with the Simple Bayesian Classifier, *Proceedings of the 6th Pacific Rim International Conference on Artificial Intelligence (PRICAI'00)*, Australia, 2000, p. 679-689.

[MR00] Mooney R.-J., Roy L., Content-based book recommending using learning for text categorization, *Proceedings of the 5th ACM Conference on Digital Libraries (DL'00)*, USA, 2000, p. 195-204.

[MSR04] Middleton S.-E., Shadbolt N.-R., De Roure D.-C., Ontological user profiling in recommender systems, *ACM Transactions on Information Systems (TOIS)*, vol. 22 (1), ACM Press, 2004, p. 54-88.

[MSV99] Monmarché N., Slimane M., Venturini G., On improving clustering in numerical databases with artificial ants, *Proceedings of the 5th European Conference on Advances in Artificial Life*, Switzerland, LNCS vol. 1674, 1999, p. 626-635.

[MSV01] Monmarché N., Slimane M., Venturini G., L'algorithme Antclass : classification non supervisée par une colonie de fourmis artificielles, *Extraction des Connaissances et Apprentissage : Apprentissage et évolution*, vol. 1 (3), 2001, p. 131-166.

[NH98] Nguyen H., Haddawy P., The Decision-Theoretic Video Advisor, *Working Notes of the AAAI-98 Workshop on Recommender Systems*, Wisconsin, USA, 1998, p. 77-80.

[Nic97] Nichols D.-M., Implicit Rating and Filtering, *Proceedings of the 5th DELOS Workshop on Filtering and Collaborative Filtering*, Budapest, Hungary, 1997, p. 31-36.

[NN96] Nguyen S.-H., Nguyen H.-S., Some efficient algorithms for rough set methods, *Proceedings of the Conference of Information Processing and Management of Uncertainty in Knowledge-Based Systems*, Granada, Spain, 1996, pages 1451-1456.

[Paw82] Pawlak Z., Rough Sets, *International Journal of Computer and Information Sciences,* vol. 11 (5), 1982, Plenum Publishing Corporation, p. 341-356.

[Paw84] Pawlak Z., Rough Classification, *International Journal of Man-Machine Studies,* vol. 20, 1984, Academic Press Inc. (London) Limited, p. 469-483.

[Paw04] Pawlak Z., Some Issues on Rough Sets, *Transaction on Rough Sets I, LNCS 3100*, 2004.

[Paz99] Pazzani M., A framework for collaborative, content-based and demographic filtering, *Artificial Intelligence Review*, vol. 13 (5), 1999, p. 393-408.

[PB97] Pazzani M., Billsus D., Learning and Revising User Profiles: The Identification of Interesting Web Sites, *Machine Learning*, vol. 27, 1997, Kluwer Academic Publisher, p. 313-331.

[PGF03] Perugini S., Gonçalves M.-A., Fox E.-A., A Connection-Centric Survey of Recommender Systems Research, *Journal of Intelligent Information Systems*, vol. 23 (1), 2003.

[PMc43] McCulloch W.-S., Pitts W., A Logical Calculus of the Ideas Immanent in Nervous Activity, *Bulletin of Mathematical Biophysics*, vol. 5, 1943, p. 115-133 (reprinted in *Neuroncomputing: Foundations of Research*, 1988, p. 15-27).

[Pol02] Polkowski L., *Rough Sets: Mathematical Foundations*, Physica-Verlag, 2002.

[PS94] Pawlak Z., Skowron A., Rough membership functions, *Advances in the Dempster Shafer Theory of Evidence*, John Wiley & Sons Inc., 1994, p. 251-271.

[Qui86] Quinlan J.-R., Induction of decision trees, *Machine Learning*, vol. 1 (1), 1986, p. 81-106.

[Qui93] Quinlan J.-R., C4.5: Programs for Machine Learning, *Morgan Kaufmann*, San Mateo, USA, 1993.

[RAC+02] Rashid A., Albert I., Cosley D., Lam S.-K., Mcnee S.-M., Konstan J.-A., Riedl J., Getting to Know You: Learning New User Preferences in Recommender Systems, *Proceedings of the 7th International Conference on Intelligent User Interfaces, IUI'02*, California, USA, 2002, p. 127-134.

[Ran71] Rand W.-M., Objective criteria for the evaluation of clustering methods, *Journal of the American Statistical Association*, vol. 66, 1971, p. 846-850.

[Ric79] Rich E., User Modeling via Stereotypes, *Cognitive Science*, vol. 3, 1979, p. 329-354.

[RIS+94] Resnick P., Iacovou N., Suchak M., Bergstrom P., Riedl J., GroupLens: An Open Architecture for Collaborative Filtering of Netnews, *Proceedings of the Conference on Computer Supported Cooperative Work (CSCW'94)*, NC, USA, 1994.

[RPT+01] Rubner Y., Puzicha J., Tomasi C., Buhmann J.-M., Empirical Evaluation of Dissimilarity Measures for Color and Texture, *Computer Vision and Image Understanding*, vol. 84 (1): *Special issue on empirical evaluation of computer vision algorithms*, 2001, p. 25-43.

[SB88] Salton G., Buckley C., Term Weighting Approaches in Automatic Text Retrieval, *Information Processing and Management*, vol. 24 (5), 1988, p.513-523.

[Sha48] Shannon C., A mathematical theory of information, *The Bell System Technical Journal*, 27, 1948.

[Sha76] Shafer G., *A Mathematical Theory of Evidence*, Princeton University Press, 1976.

[SKR02] Schafer J.-B., Konstan J.-A., Riedl J., Meta-recommendation systems: user-controlled integration of diverse recommendations, *Proceedings of the 11th International Conference on Information and Knowledge Management* (CIKM'02), Virginia, USA, 2002, p. 43-51.

[SM95] Shardanand U., Maes P., Social Information Filtering: Algorithms for Automating "Word of Mouth, *Proceedings of The SIGCHI Conference on Human Factors in Computing Systems (CHI'95)*, USA, 1995, p. 210-217.

[SP04] Serban G., Pintea M.-C., Heuristics and learning approaches for solving the Travling Salesman Problem, *Studia Universitatis Babes-Bolyai Informatica*, vol. XLIX (2), 2004, p. 27-36.

[SP90] Shafer G., Pearl J., Readings in Uncertainty Reasoning, *Morgan Kaufmann Publishers*, San Mateo, 1990.

[SPU01] Schein A.-I., Popescul A., Ungar L.-H., Generative Models for Cold-Start Recommendations, *Proceedings of the 2001 SIGIR Workshop on Recommender Systems*, USA, 2001.

[SR92] Skowron A., Rauszer C., The Discernibility Matrices and Functions in Information Systems, *Intelligent Decision Support: Handbook of Applications and Advances of Rough Set Theory, Series: Theory and Decision Library*, vol. 11, Kluwer Academic Publishers, Dordrecht, 1992, p. 331-362.

[SS02] Swearingen K., Sinha R., Interaction Design for Recommender Systems, *Designing Interactive Systems*, London, UK, 2002.

[SSF02] Schneider D., Synteta P., Frété C., Community, Content and Collaboration Management Systems in Education: A New Chance for Socio-Constructivist Scenarios?, *Proceedings of the 3rd Congress on Information and Communication Technologies in Education*, Greece, 2002.

[TM88] Trojanowicz R.-C., Moors M.-H., The Meaning of Community in Community Policing, *National Neighbourhood Foot Patrol Centre*, vol. 137, 1988, p. 1-16.

[TvR04] Tombros A., van Rijsbergen C.-J., Query-Sensitive Similarity Measures for Information Retrieval, *Knowledge and Information Systems*, vol. 6 (5), 2004, Springer London, p. 617-642.

[UCI] UCI Knowledge Discovery in Databases Archive, http://kdd.ics.uci.edu.

[Vap82] Vapnik V.-N., Estimation of Dependences Based on Empirical Data, *Springer Series in Statistics*, Springer-Verlag, 1982.

[Vap95] Vapnik V.-N., The nature of statistical learning theory, *Springer-Verlag*, 1995.

[Vel01] VeltKamp R.-C., Shape Matching: Similarity Measures and Algorithms, *Proceedings of the 17th International Conference on Shape Modeling and Applications*, Italy, 2001, p. 188-197.

[vRi79] van Rijsbergen C.-J., Information Retrieval, *Butterworth Publisher*, 1979.

[Was99] Wasfi A.-M., Collecting User Access Patterns for Building User Profiles and Collaborative Filtering, *Proceedings of the 4th International Conference on Intelligent User Interfaces (IUI'99)*, California, USA, 1999, p. 57-64.

[Wen98] Wenger E., Communities of practice: Learning, meaning, and identity, *New York Cambridge University Press*, 1998.

[Yao98a] Yao Y.-Y., A comparative study of fuzzy sets and rough sets, *Information Sciences*, vol. 109 (1-4), 1998, p. 227-242.

[Yao98b] Yao Y.-Y., Generalized rough set models, *Rough Sets in Knowledge Discovery*, Polkowski L. and Skowron A. (Eds.), Physica-Verlag, Heidelberg, 1998, p. 286-318.

[Zad65] Zadeh L., Fuzzy Sets, *Information and Control*, vol. 8 (3), 1965.

[ZMK+05] Ziegler C.-N., McNee S.-M., Konstan J.-A., Lausen G., Improving recommendation lists through topic diversification, *Proceedings of the 14th International Conference on World Wide Web*, Chiba, Japan, 2005, p. 22-32.

[ZY04] Zhang M., Yao J.-T., A Rough Sets Based Approach to Feature Selection, *Proceedings of the 23rd Conference of the North American Fuzzy Information Processing Society*, Canada, 2004.

Glossaire

Carte de communautés : moyen de visualiser un espace de communautés en 2D	11.4.3, 16.3
Communauté : ensemble des utilisateurs qui sont proches les uns des autres selon un critère de comparaison	1.3.2.2, 8.2
Confiance : la proportion d'occurrences de la règle par rapport aux occurrences de sa prémisse dans la table de décision	8.5.1
Consistance : la proportion des règles certaines par rapport à la taille de la table de décision	8.5.1
Consistance approximative : la consistance relative aux réductions approximatives	8.5.3
Espace de communautés : ensemble des communautés formées par un critère donné	8.2.1
Polymorphisme : capacité d'un utilisateur appartenant à la fois à plusieurs communautés	8.2.1, 8.2.2
Profil : ce qui décrit un utilisateur selon plusieurs dimensions : informations personnelles, centres d'intérêt, etc.	1.3.2.1, 9.1
Réduction : ensemble des attributs de condition préservant la région positive	8.4.3
Réduction approximative : réduction dont la signification dépasse un seuil donné	8.5.2
Région positive : ensemble des règles certaines d'une table de décision, ou table de communautés	8.4.1
Relation d'indiscernabilité : relation d'équivalence définie sur une table d'information ou table de décision	8.2.1
Support : la proportion d'occurrences de la règle dans la table de décision	8.5.1
Table de communautés : table de décision représentant des espaces de communautés	8.2.3
Vecteur de positionnement : vecteur des communautés d'un utilisateur, représentant le polymorphisme de son positionnement au sein des communautés	8.2.2

Partie VII.

Annexe

Annexe A

Relations d'équivalence

Dans l'annexe A, nous présentons les notions élémentaires des relations d'équivalence, qui sont définies à partir des relations binaires. Les relations d'équivalence sont nécessaires à la théorie des ensembles d'approximation ainsi qu'aux formalisations dans ce manuscrit.

A.1 Relation binaire

Une *relation binaire* d'un ensemble de départ A vers un ensemble d'arrivée B, notée $\mathcal{R}_{AB}$ ou simplement $\mathcal{R}$ s'il n'y a pas d'ambiguïté, est définie par un sous-ensemble de A x B : $\mathcal{R} \subseteq A \text{ x } B$. Pour tout $a \in A$ et $b \in B$, si $(a,b) \in \mathcal{R}$, on dit que l'élément a est *en relation* avec l'élément b, et on le note $a\mathcal{R}b$.

A titre d'exemple, soient $A = \{a_1, a_2, a_3, a_4, a_5, a_6\}$ et $B = \{b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8\}$. Nous avons une relation binaire $\mathcal{R} \subseteq A \text{ x } B$ qui contient six éléments suivants : (voir Figure A.1)

$$\mathcal{R} = \{\boldsymbol{r}_1 = (a_1, b_8), \boldsymbol{r}_2 = (a_1, b_2), \boldsymbol{r}_3 = (a_4, b_6), \boldsymbol{r}_4 = (a_3, b_5), \boldsymbol{r}_5 = (a_5, b_3), \boldsymbol{r}_6 = (a_6, b_3)\}$$

	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8
a_1		r_2						r_1
a_2								
a_3					r_4			
a_4						r_3		
a_5			r_5					
a_6			r_6					

Figure A.1 – *Matrice de relation binaire $a\mathcal{R}b$.*

Cette relation peut être représentée par un diagramme comme illustré dans la Figure A.2.

Remarques :

- En général, $a\mathcal{R}b$ n'implique pas $b\mathcal{R}a$ (symétrie), et on n'a pas toujours $a\mathcal{R}a$ (réflexion).
- En cas où $A = B$, on dit que $\mathcal{R}_A$ est une relation définie sur A ou dans A, et son diagramme représentatif devient un graphe orienté comme illustré dans la Figure A.3.

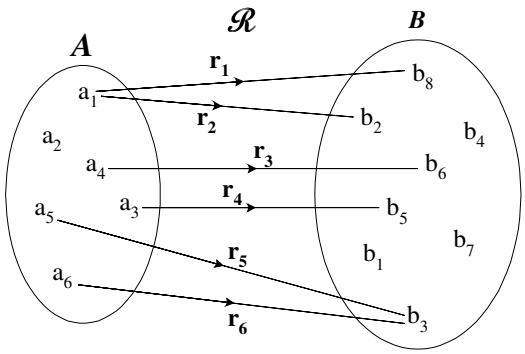


Figure A.2 – *Diagramme de la relation binaire* $\mathcal{R} \subseteq A$ x B .

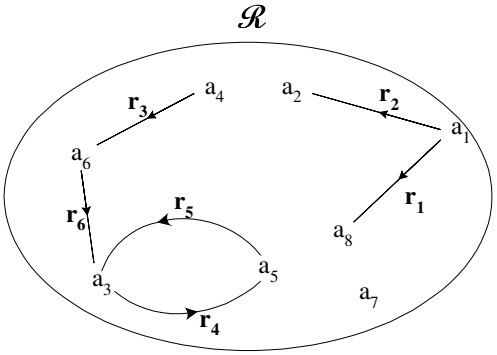


Figure A.3 – *Graphe de la relation binaire* $\mathcal{R}$ *sur l'ensemble A.*

A.2 Relation d'équivalence

A.2.1 Définition

Une *relation d'équivalence* $\mathcal{R}$ sur l'ensemble A est une relation binaire qui vérifie trois propriétés :

i) réflexive : $a\mathcal{R}a$, $\forall a \in A$

ii) symétrique : $a\mathcal{R}b \Rightarrow b\mathcal{R}a$, $\forall a, b \in A$, et

iii) transitive : $(a\mathcal{R}b) \wedge (b\mathcal{R}c) \Rightarrow (a\mathcal{R}c)$, $\forall a, b, c \in A$.

Puisque toute relation d'équivalence est symétrique, les arcs présents dans la Figure A.4 sont non orientés, et la matrice de représentation est également symétrique par rapport à la diagonale.

Exemples :

- « être dans la même classe » est une relation d'équivalence sur l'ensemble des lycéens
- // (parallèle) et = (égalité) sont des relations d'équivalence

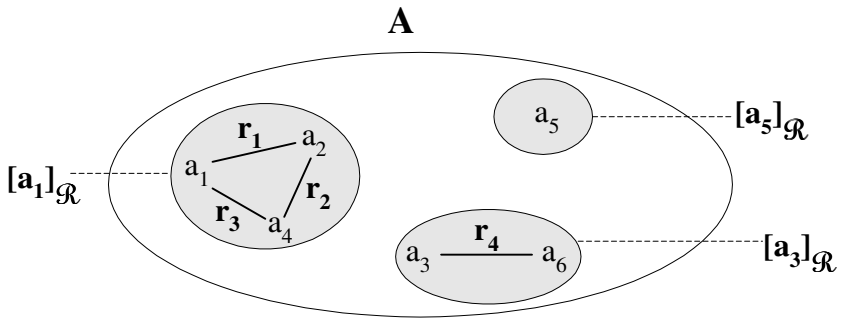


Figure A.4 – *Relation d'équivalence* $\mathcal{R}$.

A.2.2 Classe d'équivalence et ensemble quotient

Soit la relation d'équivalence $\mathcal{R}$ sur l'ensemble *A*. La *classe d'équivalence* d'un élément $a \in A$ suivant $\mathcal{R}$, notée $[a]_{\mathcal{R}}$, contient des éléments qui sont en relation avec *a* :

$$[a]_{\mathcal{R}} = \{b \in A \mid a \mathcal{R} b\}$$

Par exemple, dans la Figure A.4 nous avons les trois classes d'équivalence suivantes :

- $[a_1]_{\mathcal{R}} = \{a_1, a_2, a_4\}$,
- $[a_3]_{\mathcal{R}} = \{a_3, a_6\}$, et
- $[a_5]_{\mathcal{R}} = \{a_5\}$

Remarque :

- le rôle des éléments dans une classe d'équivalence est identique, c'est-à-dire que l'on peut choisir arbitrairement un de ces éléments comme le représentant de la classe : $[a_1]_{\mathcal{R}} = [a_2]_{\mathcal{R}} = [a_4]_{\mathcal{R}}$

L'*ensemble quotient* de A par la relation d'équivalence $\mathcal{R}$, noté $A/\mathcal{R}$, est l'ensemble de toutes les classes d'équivalence suivant $\mathcal{R}$:

$$A/\mathcal{R} = \{[a]_{\mathcal{R}} \mid a \in A\}$$

On dit aussi que $A/\mathcal{R}$ est une *partition*, ou classification, de *A* par la relation $\mathcal{R}$, qui vérifie :

i) $\forall [a]_{\mathcal{R}}, [b]_{\mathcal{R}} \in A/\mathcal{R}, \ ([a]_{\mathcal{R}} \neq [b]_{\mathcal{R}}) \Leftrightarrow ([a]_{\mathcal{R}} \cap [b]_{\mathcal{R}} = \varnothing)$

ii) $\bigcup_{[a]_R \in A/R} [a]_R = A$

Annexe B

Théorie des ensembles d'approximation

Dans la suite, nous ne présentons que les notions de la théorie des ensembles d'approximation, qui sont utiles pour la thèse, et les informations complémentaires se trouvent par exemple dans [Paw82, Paw84, Paw04, Pol02].

B.1 Représentation de données

B.1.1 Table de décision

Dans la théorie des ensembles d'approximation, une *table d'information T* est caractérisée par une paire de deux ensembles non vides : $T = < U, A >$, où U est l'*univers des objets* et A est un ensemble d'au moins deux *attributs*. Le Tableau B.1 montre un exemple de table d'information.

U \ A	Profession	Ville	Genre préféré	Evaluation
u_1	Commerçant	Paris	Aventure	Groupe 1
u_2	Chercheur	Paris	Aventure	Groupe 4
u_3	Chercheur	Paris	Documentaire	Groupe 2
u_4	Chercheur	Paris	Scientifique	Groupe 1
u_5	Chercheur	Paris	Scientifique	Groupe 4
u_6	Chercheur	Paris	Scientifique	Groupe 3
u_7	Chercheur	Paris	Fiction	Groupe 5
u_8	Chercheur	New York	Documentaire	Groupe 5
u_9	Commerçant	New York	Documentaire	Groupe 5
u_{10}	Commerçant	Londres	Documentaire	Groupe 3
u_{11}	Commerçant	Londres	Documentaire	Groupe 2
u_{12}	Commerçant	Londres	Documentaire	Groupe 3

Tableau B.1 – *Table d'information*.

De plus, l'ensemble des attributs A est divisé en deux : A = $C \cup D$ où D contient un seul attribut dit *décision*, et C contient les attributs restants dits *condition*. La table T est dite *table de décision*, et chaque ligne de la table $T[u]$ est considérée comme une *règle de décision*, ou « règle » tout court. Par exemple, dans la Figure B.1 où Evaluation est pris comme attribut de décision, on a la règle u_7 :

(Profession = « Chercheur », Ville = « Paris », Genre = « Fiction ») → (Evaluation = « Groupe 5 »)

qui dit : « Un chercheur parisien qui aime les films Fiction appartient au groupe libellé Groupe 5 de l'attribut Evaluation ».

B.1.2 Relation d'indiscernabilité

Soit le sous-ensemble d'attributs $P \subseteq A$. Pour traduire le fait que les deux objets u et u' ont les mêmes valeurs sur l'ensemble P, on définit la *relation d'indiscernabilité*, notée $\mathcal{R}_P$, comme une relation d'équivalence sur U telle que :

$$\forall u, u' \in U, \qquad u \mathcal{R}_P u' \;\Leftrightarrow\; (\forall a \in P, T[u, a] = T[u', a])$$

A partir de la relation d'indiscernabilité $\mathcal{R}_P$, on obtient l'ensemble quotient $U/\mathcal{R}_P$ qui est la partition de U regroupant les objets de U prenant les mêmes valeurs sur P :

$$U/\mathcal{R}_P = \{[u]_P \mid u \in U\}, \quad \text{avec } [u]_P \equiv [u]_{\mathcal{R}P}$$

A titre d'exemple, pour P = {Ville, Genre}, l'ensemble quotient $U/\mathcal{R}_P$ contient les six classes d'équivalence suivantes (voir Figure B.1) :

$G_1 = [u_1]_P = \{u_1, u_2\}$, (Ville = « Paris », Genre = « Aventure »)

$G_2 = [u_3]_P = \{u_3\}$, (Ville = « Paris », Genre = « Documentaire »)

$G_3 = [u_4]_P = \{u_4, u_5, u_6\}$, (Ville = « Paris », Genre = « Scientifique »)

$G_4 = [u_7]_P = \{u_7\}$, (Ville = « Paris », Genre = « Fiction »)

$G_5 = [u_8]_P = \{u_8, u_9\}$, (Ville = « New York », Genre = « Documentaire »)

$G_6 = [u_{10}]_P = \{u_{10}, u_{11}, u_{12}\}$, (Ville = « Londres », Genre = « Documentaire »)

U \ A	Profession	Ville	Genre	D = {Evaluation}	
u_1	Commerçant	Paris	Aventure	Groupe 1	
u_2	Chercheur	Paris	Aventure	Groupe 4	
u_3	Chercheur	Paris	Documentaire	Groupe 2	
u_4	Chercheur	Paris	Scientifique	Groupe 1	
u_5	Chercheur	Paris	Scientifique	Groupe 4	
u_6	Chercheur	Paris	Scientifique	Groupe 3	
u_7	Chercheur	Paris	Fiction	Groupe 5	← X_5
u_8	Chercheur	New York	Documentaire	Groupe 5	
u_9	Commerçant	New York	Documentaire	Groupe 5	
u_{10}	Commerçant	Londres	Documentaire	Groupe 3	
u_{11}	Commerçant	Londres	Documentaire	Groupe 2	
u_{12}	Commerçant	Londres	Documentaire	Groupe 3	

Figure B.1 – *Exemple de table de décision avec D = {Evaluation} et P = {Ville, Genre}.*

Lorsque P = D, où D contient l'attribut de décision, une classe d'équivalence $[u]_D \in U/\mathscr{R}_P$ est appelée *concept*. Un concept est donc l'ensemble des objets qui prennent la même valeur sur *D*. Dans l'exemple précédent, où D = {Evaluation}, on a 5 concepts (voir Figure B.1) :

$X_1 = \{u_1, u_4\}$ (Evaluation = « Groupe 1 »)

$X_2 = \{u_3, u_{11}\}$ (Evaluation = « Groupe 2 »)

$X_3 = \{u_6, u_{10}, u_{12}\}$ (Evaluation = « Groupe 3 »)

$X_4 = \{u_2, u_5\}$ (Evaluation = « Groupe 4 »)

$X_5 = \{u_7, u_8, u_9\}$ (Evaluation = « Groupe 5 »)

Dans la suite, on va travailler autour des concepts, afin d'étudier dans quelle mesure ils peuvent être définis au moyen des attributs de condition, et permettre d'induire la valeur du critère Evaluation à partir des valeurs de Profession, Genre préféré et Ville.

B.2 Ensembles d'approximation

Rappelons d'abord que *C* est l'ensemble des attributs de condition, composé des attributs restants une fois que l'attribut de décision *D* est choisi.

Un concept *X* est dit *définissable* par $P \subseteq C$, s'il est une union de classes d'équivalence de la relation d'indiscernabilité $\mathscr{R}_P$:

$$X = \bigcup_{[u]_P \in U / R_P} [u]_P \tag{B.4}$$

Cette définition est très contraignante. En effet, dans l'exemple précédent, X_5 est le seul concept définissable par P = {Ville, Genre} puisqu'il est l'union des classes $[u_7]_P$ et $[u_8]_P$, et les autres ne sont pas définissables par *P*.

Si le concept *X* est non définissable par *P*, il sera mesuré de façon approximative par ses deux bornes inférieure et supérieure comme suit.

La *borne P-inférieure* du concept *X* comprend tous les objets dans l'univers *U* dont la classe d'équivalence est incluse dans *X* :

$$\underline{P}(X) = \{u \in U \mid [u]_P \subseteq X\} \subseteq X \tag{B.5}$$

Les objets dans la borne *P*-inférieure d'un concept *X* sont des règles certaines par rapport à *P* puisqu'ils peuvent être certainement classés dans *X* en se basant sur les attributs de *P*.

Reprenons l'exemple plus haut. Nous avons : $\underline{P}(X_2) = \{u_3\}$, puisque $[u_3]_P \subseteq X_2$, mais $[u_{11}]_P = [u_{10}]_P \not\subset X_2$; de façon similaire, nous obtenons : $\underline{P}(X_5) = X_5$, et $\underline{P}(X_1) = \underline{P}(X_3) = \underline{P}(X_4) = \varnothing$.

La *borne P-supérieure* du concept X comprend tous les objets de l'univers U dont la classe d'équivalence recouvre au moins en partie le concept X.

$$\overline{P}(X) = \{u \in U \mid [u]_P \cap X \neq \varnothing\} \tag{B.6}$$

Alors, la borne P-supérieure comporte les règles possibles par rapport à P, c'est-à-dire les objets qui sont possiblement classés dans X en se basant sur les attributs de condition de P.

Dans notre exemple précédent, nous avons les bornes P-inférieures de cinq concepts telles que :

- $\overline{P}(X_1) = \{u_1, u_2, u_4, u_5, u_6\}$
- $\overline{P}(X_2) = \{u_3, u_{10}, u_{11}, u_{12}\}$
- $\overline{P}(X_3) = \{u_4, u_5, u_6, u_{10}, u_{11}, u_{12}\}$
- $\overline{P}(X_4) = \{u_1, u_2, u_4, u_5, u_6\}$
- $\overline{P}(X_5) = \{u_7, u_8, u_9\}$

Pour résumer, nous retenons que $\underline{P}(X) \subseteq X \subseteq \overline{P}(X)$. En plus, si X est définissable par P, nous avons : $\underline{P}(X) = X = \overline{P}(X)$. Sinon, X est qualifié de *P-ensemble d'approximation* (rough set) car il requiert une approximation pour être défini.

B.3 Dépendance d'un attribut de décision

La *région positive* par rapport à l'ensemble $P \subseteq C$, notée $POS_P(D)$, est définie comme l'union des bornes P-inférieures pour tous les concepts X de D :

$$POS_P(D) = \bigcup_{X \in R_D} \underline{P}(X) \tag{B.7}$$

Ainsi, la région positive comprend toutes les règles certaines de la table de décision.

En utilisant la notion de région positive, on peut mesurer le « niveau de dépendance » de l'attribut de décision D par rapport à l'ensemble P par le coefficient suivant :

$$k_{P,D} = \gamma(P, D) = \frac{\left|POS_P(D)\right|}{\left|U\right|} \tag{B.8}$$

Cela traduit la capacité de la région positive à intégrer tous les objets de U. On dit aussi que l'attribut de décision D dépend de P au degré k, noté $P \Rightarrow_k D$. Si $k = 1$, on obtient une « dépendance totale », et tous les objets de l'univers U peuvent être certainement regroupés dans les concepts en utilisant P. Par contre, si $k < 1$, on a une « dépendance partielle ».

En termes de classification, le coefficient k représente la proportion des règles certaines dans l'univers U. Il est également le ratio des objets pouvant être regroupés dans les concepts en utilisant l'ensemble P sur le nombre total d'objets. De ce point de vue, la valeur $\gamma(C, D)$ exprime aussi le « degré de consistance » de la table T, ou le « degré d'approximation » de la classification de U vis-à-vis de l'attribut de décision D ($U/\mathcal{R}_D$) par les attributs de condition de C.

B.4 Signification des attributs de condition

On cherche maintenant à travailler sur l'ensemble des attributs de condition, de façon à savoir desquels on pourrait se passer pour déterminer l'attribut de décision D. Un attribut de condition $c \in P$ est dit « indispensable dans P » si $U/\mathcal{R}_P \neq U/\mathcal{R}_{P\setminus\{c\}}$. Cela signifie que si l'on enlève l'attribut de condition c, la partition sera changée. Sinon, l'attribut de condition c est dit « dispensable dans P ».

On définit aussi la *réduction* de C, notée *reduct*(C), comme le sous-ensemble P de C comprenant l'ensemble minimal des attributs de condition (indispensables) de C suffisant pour préserver la région positive : $POS_P(D) = POS_C(D)$

Une table de décision T peut donner plusieurs réductions. En théorie, la tâche de déterminer l'ensemble de toutes les réductions de C, noté $RED(C)$, est un problème NP-difficile, et il existe dans la littérature des algorithmes heuristiques permettant de déterminer l'ensemble des réductions $RED(C)$.

Dans le même esprit, on peut mesurer l'importance d'un attribut de condition $c \in C$ sur l'intervalle [0,1] plutôt que de l'évaluer de façon binaire {dispensable, indispensable}. En effet, la « signification de l'attribut de condition c » peut être mesurée par l'impact de sa suppression sur le degré de consistance de la région positive :

$$\sigma_{C,D}(c) = 1 - \frac{\gamma(C\setminus\{c\},D)}{\gamma(C,D)} \tag{B.9}$$

De façon similaire, on peut mesurer la signification d'un sous-ensemble des attributs de condition $P \subset C$ par :

$$\sigma_{C,D}(P) = 1 - \frac{\gamma(C\setminus P,D)}{\gamma(C,D)} \tag{B.10}$$

Si P est une réduction de C, on a : $\sigma_{C,D}(P) = 1$. Si C et D sont fixés, on peut simplifier la notation $\sigma_{C,D}(.)$ en $\sigma(.)$.