



HAL
open science

Les systèmes de recommandation à base de confiance

Charif Alchiekh Haydar

► **To cite this version:**

Charif Alchiekh Haydar. Les systèmes de recommandation à base de confiance. Autre [cs.OH]. Université de Lorraine, 2014. Français. NNT : 2014LORR0203 . tel-01751172

HAL Id: tel-01751172

<https://hal.univ-lorraine.fr/tel-01751172>

Submitted on 29 Mar 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : ddoc-theses-contact@univ-lorraine.fr

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4

Code de la Propriété Intellectuelle. articles L 335.2- L 335.10

http://www.cfcopies.com/V2/leg/leg_droi.php

<http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm>



Les systèmes de recommandation à base de confiance

THÈSE

présentée et soutenue publiquement le 3 septembre 2014

pour l'obtention du

Doctorat de l'Université de Lorraine
(mention informatique)

par

Charif ALCHIEKH HAYDAR

Composition du jury

<i>Rapporteurs :</i>	Isabelle Tellier	Professeur à l'université Paris 6.
	Patrick Gallinari	Professeur à l'université Paris 3.
<i>Examineurs :</i>	Nathalie Denos	Maitre de conférence à l'université Pierre Mendès France.
	Kamel Smaili	Professeur à l'université de Lorraine.
<i>Directrice :</i>	Anne Boyer	Professeur à l'université de Lorraine.
<i>Co-encadrant :</i>	Azim Roussanaly	Maitre de conférence à l'université de Lorraine.

Mis en page avec la classe thesul.

Remerciements

Les remerciements.

Voilà, ma thèse est terminée. Une période assez particulière de ma vie vient de s'achever. Une période riche d'expériences et d'apprentissages scientifiques, mais aussi d'échanges humains instructifs. Si cette période s'est bien finie, c'est d'une part grâce au soutien des agréables personnes que j'ai pu côtoyer. Au regard de leurs qualités, mes remerciements semblent modestes. J'espère leur rendre ici une part de ce qu'ils méritent, tout en souhaitant que la vie me donne encore l'occasion d'affirmer ma gratitude envers eux.

En premier lieu, mes remerciements vont à mes deux directeurs Anne Boyer et Azim Rousanally, pour leur soutien, leur patience, et leur optimisme même durant les périodes les plus critiques de cette thèse. Je serai toujours reconnaissant de leur bienveillance au plan scientifique, comme au plan humain. Je remercie aussi Armelle BRUN pour son précieux temps qu'elle m'a souvent accordé, pour ses conseils et son inestimable engagement.

Je remercie madame Isabelle Tellier et monsieur Patrick Gallinari d'avoir accepté d'être les rapporteurs de ma thèse, et d'avoir pris le temps de lire ce manuscrit et d'évaluer ce travail. J'adresse également mes remerciements à monsieur Kamel Smaili et madame Natalie Denos d'avoir accepté de faire partie de mon jury de thèse. Je remercie chaleureusement ma famille pour leurs constants soutiens et encouragements menés depuis la Syrie. Je remercie aussi profondément ma tante Toham Krempf et son mari André Krempf pour leur présence constante près de moi.

Je tiens à remercier Bernard Jullien pour son aide inestimable en langue française, et pour les précieuses discussions et échanges que j'ai eus avec lui, mais aussi avec sa famille.

Je remercie aussi mes amis en France, et en Syrie pour tous les bons moments qui éclairaient mon esprit, et m'aidaient à me ressourcer pour aller plus loin dans mon travail. Merci à Ghassan Watfa, Jalil ELghalabzouri, Main Kafa, Ahmad Hamad, Zienab Darwich, Mohammad Ali, Inès Souilème, Mayssa Jaouadi, Nadim El-Chaghouri, Hazem Hassan, Hassan Alomar.

Je remercie aussi les membres de l'équipe Kiwi qui étaient plus qu'une famille pour moi, tout au long de cette thèse. Je remercie Geoffray Bonnin et Nicolas Jones pour m'avoir aidé à m'orienter au début de ma thèse et pour être plus que des amis adorables. Je remercie Sylvain Castagnos de ses conseils, et de son assistance. Je remercie Sahbi Sidhom pour ses conseils mais aussi pour les pauses café très reposantes après de longues journées de travail. Je remercie aussi les collègues de bureau qui sont devenus des amis inoubliables : Rita Aleksandrova, Lina Fahed, Thomas Largillier, Manel Sorba, Sonia Ben Ticha, Ilham Esslimani, Laura Infante Blanco. Je remercie Samuel Nowakowski et Audrey Knauf pour leur patience à me lire et à me corriger. Je remercie Brahim Batouche pour son engagement et ses remarques intéressantes durant la préparation de soutenance et le jour même de la soutenance. Je remercie Ngoc-Chan Nguyen, Cédric Bernier, Benjamin Gras et Amaury L'Huillier pour avoir été des collègues remarquables.

Finalement, je remercie l'équipe de l'entreprise Womup pour sa confiance, et pour avoir tout tenté afin que cette thèse se poursuive.

*Je dédie cette thèse
à ma famille.
En espérant la voir
le plus tôt possible.*

Table des matières

Introduction générale

ix

Partie I Introduction et état de l'art

Cadre général	1
1 Constat de départ	1
2 Problématiques scientifiques	2
3 Contribution	4
Chapitre 1	
Les systèmes de recommandation	
1.1 Les approches de la recommandation	8
1.1.1 Systèmes de recommandation à base d'items	10
1.1.2 Systèmes de recommandation à base d'utilisateurs	11
1.1.3 Synthèse sur les approches de recommandation	14
1.1.4 Les systèmes de recommandation hybrides	18
1.2 Évaluation des systèmes de recommandation	19
1.2.1 Évaluation par tests hors ligne	20
1.3 Résumé	22
Chapitre 2	
La confiance computationnelle	
2.1 Introduction	25
2.2 La confiance sociale	26
2.2.1 Les zones de confiance	27
2.2.2 Le contexte de la confiance	28
2.2.3 Les deux paradigmes de confiance	28
2.2.4 Mise à jour de la confiance	31

2.3	La modélisation de la confiance sociale	32
2.4	Tour d’horizon des algorithmes de confiance	33
2.4.1	Advogato	33
2.4.2	AppleSeed	34
2.4.3	EigenTrust	35
2.4.4	MoleTrust	37
2.4.5	TidalTrust	39
2.4.6	Le modèle d’Abdul-Rahman	40
2.4.7	Le modèle de Richardson	41
2.4.8	Le modèle d’O’Donovan	42
2.4.9	Bitreillis	43
2.4.10	La logique subjective et les réseaux de confiance	44
2.5	Résumé	49

Chapitre 1**Hybridation du filtrage collaboratif avec la confiance**

1.1	Introduction	55
1.2	Données et contexte	56
1.3	Motivation	57
1.4	Hybridation	58
1.4.1	Stratégies d'hybridation	59
1.4.2	Méthode d'évaluation	60
1.4.3	Analyse des résultats	61
1.5	Analyse du comportement des utilisateurs	65
1.5.1	Les caractéristiques du comportement de l'utilisateur	65
1.5.2	Classification d'utilisateurs	72
1.6	Conclusion	79

Chapitre 2**Modèles de confiance à base de la logique subjective**

2.1	Motivation	81
2.2	Données et contexte	82
2.3	Confiance locale	84
2.4	Confiance collective	86
2.5	Confiance globale	88
2.5.1	Le traitement des réponses non-acceptées	90
2.6	Mesures de qualité	93
2.7	Résultats et discussions	93
2.7.1	Précision	93
2.7.2	Complexité	94
2.7.3	Résistance face aux attaques malveillantes	95
2.8	Conclusion	96

Partie III Conclusion 99

Chapitre 1 Conclusion et perspectives
--

1.1 Bilan	101
1.2 Perspectives	104

Bibliographie 107

Introduction générale

Première partie

Introduction et état de l'art

Table des figures

1.1	La catégorisation des systèmes de recommandation	9
2.1	Les zones de confiance	27
2.2	Le réseau de confiance	30
2.3	La transitivité de confiance selon Moletrust avec $d=4$	38
2.4	MoleTrust avec plusieurs chemins	38
2.5	tidalTrust	40
2.6	Opinion	46
1.1	Un commentaire rédigé dans Epinions	56
1.2	Histogramme des notes aux données Epinion	57
1.3	MAE	63
1.4	HMAE	64
1.5	UMAE	65
1.6	Nombre de notes des utilisateurs (NNU) et UMAE	66
1.7	La satisfaction des utilisateurs par catégorie	67
1.8	La popularité des items (PIN)	69
1.9	codev et UMAE	70
1.10	CODEV et le nombre de notes par utilisateur	70
1.11	Les utilisateurs de nature confiante/réservée	71
1.12	Réputation des amis crédibles (RAC) et UMAE	72
1.13	La structure de la classe C0	75
1.14	La structure de la classe C1	76
1.15	La structure de la classe C2	76
1.16	La structure de la classe C3	77
1.17	La structure de la classe C4	77
1.18	La structure de la classe C5	78
1.19	La structure de la classe C6	78
1.20	La structure de la classe C7	79
2.1	Une question dans Stackoverflow	83
2.2	Graphe de confiance	87
2.3	évaluation des trois modèles de confiance à base de la logique subjective	96

Cadre général

1 Constat de départ

Notre quotidien nous expose fréquemment à des situations qui nous conduisent à prendre des décisions et à opérer des choix. Nourriture, vêtements, voiture, lycée pour nos enfants, lieu de vacances etc, qu'ils soient cruciaux ou secondaires, nos choix sont dictés par nos connaissances, préférences, et par les informations dont nous disposons.

Les sociétés qui nous ont précédés ont souffert d'un manque d'informations car même lorsque celles-ci existaient, les moyens d'y accéder étaient peu développés.

Aujourd'hui, avec Internet, nous possédons le plus grand entrepôt de données que l'humanité ait pu connaître. En mars 2009, la capacité numérique d'Internet équivalait à une pile de livres égale à 10 fois la distance entre la Terre et Pluton [Wray2009]). En 2008, Google annonçait cesser de dénombrer les pages indexées par son moteur de recherche, après avoir dépassé les 10^{12} pages [Alpert2008]. Le site "World Wide Web"¹ développé par [de Kunder2006], qui a pris la relève pour compter les pages, déclare qu'en juillet 2013, Google référençait environ $3,47 \times 10^{12}$ pages. Ces chiffres montrent non seulement la taille du web, mais aussi la vitesse à laquelle il s'accroît. On parle d'"explosion de données" pour décrire ce phénomène.

Le Web 1.0 [Berners-Lee1992] a mis à la disposition du grand public, de manière quasi immédiate, un énorme flux d'informations entraînant une multiplicité des choix qui ne facilite pas toujours l'accès aux données pertinentes pour les utilisateurs. Ainsi, le domaine de la recherche d'informations [Baeza-Yates *et al.*1999, Croft *et al.*2009] est né pour rendre Internet plus facilement exploitable. Son but principal était d'ordonner les informations disponibles selon leur pertinence relativement à une requête.

Le Web 2.0 [Graham2005, O'Reilly2005] a fait son apparition, accompagné d'outils plus interactifs comme les forums et les blogs. Le grand public qui était jusque là passif dans le Web 1.0, est devenu actif dans le Web 2.0. Il est désormais devenu aisé pour tout le monde d'ajouter une information ou de partager un avis [Birdsall2007].

Pour s'adapter à ce changement, les techniques de recherche d'informations ont dû intégrer des aspects sociaux et focalisés sur l'utilisateur et son comportement. Les systèmes de recommandation (SR) [Resnick and Varian1997] en font partie. Contrairement aux systèmes de recherche d'informations qui attendent la requête de l'utilisateur pour agir, les SR tentent d'anticiper les besoins des utilisateurs et leur proposent des items qu'ils sont susceptibles d'apprécier. En effet, l'objectif d'un SR est de recommander à ses utilisateurs des items (articles, ressources, produits, pages web, etc.) sélectionnés parmi un large choix et censés être appréciés par eux. Il tente de prédire si un utilisateur donné appréciera ou non un item. Pour parvenir à un tel but, un SR a besoin d'accumuler des données sur les utilisateurs et/ou les items disponibles. Il sauvegarde les traces de ses utilisateurs, avant d'y appliquer des méthodes statistiques pour prédire leur futur

1. <http://www.worldwidewebsize.com/>

comportement à l'égard d'items.

De leur côté, les utilisateurs du SR s'en servent pour des objectifs différents, comme trouver des items convenables, s'exprimer, améliorer son profil, tester la crédibilité du système, ou simplement naviguer [Herlocker *et al.*2004]. Ayant différents caractères et habitudes, ils se comportent de manières très variées pour réaliser leurs objectifs. Par exemple, un utilisateur peut attribuer une bonne note à un film parce qu'il aime son héros, ou parce qu'il apprécie l'histoire du film, ou encore pour des raisons liées à son contexte : il l'a vu à la suite d'un film particulièrement ennuyeux, il est de particulièrement bonne/mauvaise humeur, le film est réputé et apprécié par de nombreuses personnes... De même, deux utilisateurs qui apprécient le même film, peuvent exprimer leur appréciation différemment ; comme par exemple : donner une note élevée, écrire un commentaire à son sujet, ou encore le recommander à des amis. Ainsi, non seulement les utilisateurs ont différents goûts et préférences, mais ils développent aussi des habitudes et des pratiques particulières, qui peuvent modifier leurs profils. Cette diversité comportementale affecte la performance du SR par un phénomène de fluctuation, car ce dernier est sensible aux valeurs des notes des utilisateurs mais pas à leurs habitudes d'usage.

Par la suite, le web social a émergé au sein du web 2.0, changeant la manière dont les internautes s'expriment sur le web. Le web social est un ensemble de relations qui lient les gens sur le Web [Appelquist *et al.*2010]. Internet devient un espace de socialisation [Bouquillion and Matthews2010]. Aujourd'hui, les réseaux sociaux envahissent tous les domaines du Web ; les communautés virtuelles se forment, et se dissolvent sans cesse, et socialisent l'écrasante majorité des activités du Web [Gruber2008]. Le web social, en pratique, permet à ses utilisateurs de partager et de propager l'information, d'interagir les uns avec les autres, et d'établir mutuellement des relations d'amitié et de confiance. Ceci engendre un nouveau type de données exploitables par les SR. Les relations de confiance sont l'objet de plusieurs études pour faire de la recommandation à base de confiance [Massa and Avesani2004, Golbeck and Hendler2006]. Bien que leur exploitation se montre souvent efficace pour améliorer la performance des SR, on peut néanmoins noter deux problèmes. Le premier concerne la fluctuation des SR, car les nouvelles données ouvrent la porte à de nouveaux types de comportements et de pratiques. Certains utilisateurs par exemple sont très méfiants à l'égard des inconnus sur le web, certains font confiance plus rapidement, d'autres jamais.

Le deuxième problème concerne la modélisation de la confiance. Fondamentalement, la notion de confiance est du ressort des sciences humaines. Pourtant, ces dernières ne procurent pas une définition consensuelle. Cette dissension se transpose aux modèles de confiance dans le domaine de la recommandation. Il est évident que l'interprétation de la confiance joue un rôle crucial dans l'élaboration de ces modèles.

2 Problématiques scientifiques

La fluctuation de performance du SR Les SR exploitent plusieurs types de données pour prédire le lien entre l'utilisateur et l'item. Dans la littérature, on identifie deux grandes familles de SR [Perugini and Gonçalves2002, Adomavicius and Tuzhilin2004].

La première famille correspond aux SR à base d'utilisateurs. Elle s'appuie sur l'idée que les utilisateurs "similaires" l'un à l'autre apprécient les mêmes items. Alors, quand un utilisateur apprécie un item, le système prédit que les personnes qui lui sont similaires apprécieront aussi cet item. Le terme "similaire" ici est générique et définit l'approche :

- Si les similaires d'un utilisateur sont ceux qui apprécient fréquemment les mêmes items que lui, on parle de l'approche par filtrage collaboratif à base d'utilisateurs (FC) [Resnick

*et al.*1994], et on les appelle "voisins".

- Lorsque les similaires sont des personnes que l'utilisateur juge fiables, il s'agit de l'approche à base de confiance (SRC) [Massa and Avesani2004], et on les appelle "amis".

La différence principale entre les deux approches est qu'en FC c'est le SR qui choisit les voisins en utilisant une mesure de similarité. Alors que, dans le SRC, c'est l'utilisateur qui choisit, lui-même, ses amis.

La deuxième famille correspond aux SR à base d'items. Les approches de cette famille considèrent que lorsqu'un utilisateur apprécie un item, il appréciera les items similaires à cet item. Le terme "similaire" est ici aussi générique.

- Si la similarité est basée sur la proximité des valeurs des attributs communs entre les items, on parle de l'approche à base de contenu [Van Meteren and Van Someren2000], et on parle d'items assimilés.
- Lorsque la similarité est déduite de l'appréciation commune des utilisateurs à l'égard de deux items, il s'agit du filtrage collaboratif (FC) à base d'items [Sarwar *et al.*2001, Deshpande and Karypis2004], et on parle d'items voisins.

Le filtrage collaboratif revêt une importance particulière dans le domaine des SR ; surtout l'approche à base d'utilisateurs. C'est l'approche la plus répandue. Elle demande des connaissances minimales sur les utilisateurs et les items, et donne souvent de bons résultats [Amatriain *et al.*2009a]. Malgré sa bonne performance, elle souffre de plusieurs difficultés comme le démarrage à froid [Maltz and Ehrlich1995], le manque de données [Miller *et al.*2004], le passage à l'échelle, la fragilité face aux attaques malveillantes [Mobasher *et al.*2007], et la fluctuation de la performance selon les utilisateurs [Massa and Bhattacharjee2004].

Grâce au web social, les SR à base de confiance sont apparus. Non seulement ils fournissent de meilleures recommandations que le FC, mais ils sont aussi plus sécurisés, et permettent de réduire l'impact du démarrage à froid et le manque de données [Massa and Avesani2004, Golbeck and Hendler2006].

Toutefois, cette amélioration ne bénéficie pas de manière uniforme à tous les utilisateurs. Certains utilisateurs ont tendance à se montrer méfiants vis-à-vis d'autres utilisateurs, leurs activités se limitent donc à noter des items. D'autres font plus facilement confiance aux utilisateurs non-réputés, ce qui limite la qualité des recommandations qu'ils reçoivent.

Deux mesures sont essentiellement prises en compte lors de l'évaluation de la performance d'un SR. La précision de prédiction des préférences de ses utilisateurs, et la couverture (quantité) de ses prédictions. Un bon système de recommandation doit faire un compromis entre ces deux mesures.

La première problématique consiste alors à trouver une solution pour la fluctuation du SR afin de lui donner une adaptation aux caractéristiques et aux habitudes de ses divers utilisateurs.

La modélisation de confiance Dans [Mcknight and Chervany1996], les auteurs montrent qu'il y a une tendance à se mettre d'accord sur l'effet positif de la confiance [Kramer and Tyler], mais pas sur sa définition. En vérité, des définitions significativement différentes continuent à être utilisées dans diverses disciplines, allant d'un trait personnel [Rotter1980], à un phénomène structurel [Shapiro1987].

Cette divergence se transpose aussi aux SR à base de confiance. Ces systèmes présentent des interprétations dissemblables (modèles) de la confiance. C'est le cas sur plusieurs points :

Certains modèles considèrent la confiance comme une relation entre deux individus (nommée aussi confiance locale) comme [Massa and Avesani2004]. D'autres la représentent sous forme d'un

score universel associé à l'utilisateur pour mesurer sa fiabilité ou utilité (nommée aussi confiance globale ou réputation) [Ziegler and Lausen2004].

La constitution des valeurs de confiance fait aussi l'objet de plusieurs propositions. Dans certains modèles, elle est une valeur déclarative [Massa and Avesani2004]. Dans d'autres, elle est une valeur cumulative inférée de plusieurs interactions [Victor *et al.*2009], ou une variable aléatoire discrète (probabiliste) [Jøsang *et al.*2006].

La transitivité de la confiance sert à propager la confiance. Dans le cas d'absence d'une relation de confiance entre deux utilisateurs, les amis communs peuvent jouer un rôle d'intermédiaire pour créer cette relation grâce à la transitivité de la confiance. Cette dernière est aussi un point de divergence des modèles. La transitivité de la confiance n'a pas été formellement démontrée. Pourtant, peu de modèles affirment qu'elle n'est pas transitive [Abdul-Rahman and Hailes2000]. Les autres modèles imposent différentes contraintes sur la transitivité [Massa and Avesani2004, Jøsang *et al.*2006, Golbeck and Hendler2006, Ziegler and Lausen2004].

D'autres points de convergence concernent l'agrégation des informations venant de plusieurs amis à différents degrés de fiabilité, et la contextualisation de la confiance.

La deuxième problématique consiste à trouver un cadre pour modéliser la confiance pour les systèmes de recommandation.

3 Contribution

La première contribution de cette thèse consiste en l'étude de la fluctuation des performances des SR. Nous considérons que cette fluctuation provient de comportements hétérogènes des utilisateurs. Notre étude inclut deux approches à base d'utilisateurs (FC et SRC). Notre observation indique la présence d'utilisateurs dont le comportement est conforme à seulement une des deux approches, alors ils sont privés de recommandations dans l'autre. Nous proposons un SR hybride de ces approches. L'hybridation sert à maximiser le nombre d'utilisateurs obtenant des recommandations, sans perte sensible en précision. Notre démarche consiste à comparer plusieurs stratégies d'hybridation afin de trouver le meilleur compromis entre la précision et la couverture.

Par la suite, nous nous intéressons à expliquer certains phénomènes comportementaux en faisant l'exploration des erreurs de recommandation. Au départ, nous proposons plusieurs caractéristiques comportementales, dont l'ensemble constitue un vecteur du comportement de l'utilisateur. Ensuite nous utilisons un algorithme de classification pour isoler des sous-ensembles d'utilisateurs en fonction de leurs comportements communs, et nous étudions la variation de performance de chaque approche pour chacun d'entre eux. Cette partie de contribution comprend trois phases qui ont fait l'objet de trois articles publiés dans des conférences internationales :

- [Haydar *et al.*2012a] :
"Hybridising collaborative filtering and trust-aware recommender systems", Charif Haydar, Azim Roussanaly, Anne Boyer, In *8th International Conference on Web Information Systems and Technologies -WEBIST* Porto, Portugal, 2012.
- [Haydar *et al.*2012b] :
"Analyzing Recommender System's Performance Fluctuations across Users", Charif Haydar, Azim Roussanaly, Anne Boyer, In *International Cross Domain Conference and Workshop -CD-ARES* Prague, Czech republic 2012.
- [Haydar *et al.*2012c] :
"Clustering Users to Explain Recommender Systems' Performance Fluctuation", Charif Haydar, Azim Roussanaly, Anne Boyer, In *The 20th International Symposium on Methodologies for Intelligent Systems -ISMIS12* Macau, China, 2012.

Dans la deuxième partie de la contribution, nous discutons de plusieurs modèles de confiance dans les SR. En effet, l'objectif principal de l'utilisateur du SR est de trouver des items appréciables. Le SRC lui offre la possibilité de coopérer avec d'autres utilisateurs (amis), en acceptant leur assistance pour réaliser son objectif. Pour qu'il coopère, l'utilisateur a besoin de s'assurer de la capacité et de la volonté de ses amis à l'assister. C'est à travers sa confiance en eux qu'il va estimer ces deux indices, et prendre la décision de coopérer. Notre regard sur la confiance rejoint celui de [Arrow1974, Deutsch1977, Gambetta1988], qui considèrent la confiance comme un moyen d'incitation à la coopération.

Dans cette démarche, nous proposons des modèles de confiance à base de logique subjective (LS) [Audun2013]. Celle-ci offre une plateforme probabiliste conçue pour modéliser la confiance. La LS représente les relations de confiance sous forme d'opinions probabilistes. L'opinion est une accumulation des résultats de plusieurs interactions entre l'utilisateur et l'objet de l'opinion (un item ou un autre utilisateur). La SL est équipée de plusieurs opérateurs pour propager et pour agréger les opinions.

Nous nous appuyons sur la flexibilité de la LS pour proposer et comparer plusieurs modèles de confiance. Notre objectif est de savoir quel est le facteur qui pèse le plus sur la décision d'un utilisateur qui fait confiance à un autre. Est-ce la réputation de ce dernier? Ou bien les interactions précédentes entre les deux? Et quels rôles les amis communs jouent-ils dans ce contexte? Ainsi, nous proposons trois modèles de confiance :

- à base de confiance locale, où l'utilisateur s'appuie sur ses propres opinions, *ou* sur celles de ses amis, pour coopérer ou pas avec un autre utilisateur.
- à base de confiance collective, où l'utilisateur s'appuie sur ses propres opinions, *et* sur celles de ses amis, pour coopérer ou pas avec un autre utilisateur.
- à base de confiance globale, où l'utilisateur s'appuie sur le score de la réputation de l'autre utilisateur pour coopérer ou pas avec lui.

Nous comparons les trois modèles en terme de précision, complexité, et résistance face aux attaques malveillantes. Nos conclusions montrent que le modèle de confiance locale est résistante et moins complexe, mais le moins précis. Le modèle de confiance collective est précis et résistante, mais complexe. Le modèle de confiance globale est précis et moins complexe, mais fragile face aux attaques malveillantes.

Cette contribution a fait l'objet de deux publications dans deux conférences internationales, et une publication acceptée dans une revue internationale :

- [Haydar *et al.*2013b]
"Local trust versus global trust networks in subjective logic", Charif Haydar, Azim Roussanaly, Anne Boyer, In *the 2013 IEEE/WIC/ACM International Conference on Web Intelligence (WI13)* Atlanta, USA, 2013.
- [Haydar *et al.*2013a]
"Individual Opinions Versus Collective Opinions in Trust Modelling", Charif Haydar, Azim Roussanaly, Anne Boyer, In *The Third International Conference on Social Eco-Informatics SOTICS 2013* Lisbonne, Portugal, 2013 [Best paper award].
- "Comparing Local, Collective, and Global Trust Models", Charif Haydar, Azim Roussanaly, Anne Boyer, In *International Journal On Advances in Life Sciences*, 2014.

La démarche adoptée dans les deux parties de contribution est délibérément empirique. Autrement dit, nous nous appuyons sur des données réelles pour évaluer les différentes stratégies appliquées.

Finalement, ce document est structuré comme suit. Nous commençons par présenter un état de l'art en abordant deux sujets.

Dans le premier chapitre, nous présentons un état de l'art sur les SR, les approches dispo-

nibles, les problèmes qui affectent leurs performances, notamment la fluctuation et la couverture.

Le deuxième chapitre est dédié à la modélisation de la confiance. Nous discutons des modèles existants, de leurs qualités et de leurs limites.

Dans la deuxième partie de la thèse, nous présentons deux chapitres sur nos contributions. Le premier concerne l'exploration des erreurs de recommandation dans les approches à base d'utilisateurs (confiance et FC), et fait le lien entre la fluctuation des performances et les aspects qualitatifs du comportement d'utilisateurs.

Le deuxième présente nos propositions de modèles de confiance dans un contexte de la recommandation. Nous y discutons des qualités et des limites des modèles de confiance locale, globale et collective. Ce manuscrit se termine par un chapitre de conclusion et présente quelques perspectives à nos travaux.

Les systèmes de recommandation

Sommaire

1.1	Les approches de la recommandation	8
1.1.1	Systèmes de recommandation à base d'items	10
1.1.2	Systèmes de recommandation à base d'utilisateurs	11
1.1.3	Synthèse sur les approches de recommandation	14
1.1.4	Les systèmes de recommandation hybrides	18
1.2	Évaluation des systèmes de recommandation	19
1.2.1	Évaluation par tests hors ligne	20
1.3	Résumé	22

Les systèmes de recommandation (SR) [Resnick and Varian1997, Zuva *et al.*2012, Mahmood and Ricci2009] sont des outils dont l'objectif est de proposer des items pertinents à l'utilisateur. En d'autres termes, ils tentent de prédire pour un utilisateur l'intérêt d'un item. L'item dans ce contexte peut être un produit à acheter, un morceau de musique à écouter, un film à regarder, un livre à lire, une page web à consulter, ou bien autre chose.

Afin de pouvoir fournir des recommandations personnalisées, le SR doit connaître les préférences de chaque utilisateur. Il tente, alors, d'acquérir les informations nécessaires pour construire des profils d'utilisateurs. En particulier, il exploite les traces laissées par les utilisateurs eux-mêmes. Il collecte les traces laissées explicitement ou implicitement [Oard and Kim1998, De Gemmis *et al.*2009]. Les traces explicites sont fournies volontairement par l'utilisateur [Ricci *et al.*2011]. Les traces implicites sont collectées en traçant les actions spontanées de l'utilisateur pendant la navigation [Oard and Kim1998, Castagnos2008, Brun *et al.*2011].

Plusieurs travaux ont été menés au sujet des méthodes de collecte des traces et de leur utilité [Amatriain *et al.*2009b, O'Donovan and Smyth2005]. Ils tentent souvent de distinguer les utilisateurs les plus utiles pour le SR. Ces utilisateurs, nommés experts, peuvent être embauchés et payés pour noter des items [Amatriain *et al.*2009a], ou être les utilisateurs qui notent beaucoup d'items [Jones2010], ou être extraits en utilisant des méthodes d'apprentissage automatique [Esslimani2010, Esslimani *et al.*2010, Brun *et al.*2011]. Dans [Jones2010], l'auteur montre que les notes explicites conviennent mieux aux experts, alors que les notes implicites conviennent mieux aux autres profils. Dans [Amatriain *et al.*2009a, Esslimani2010], les auteurs essaient d'utiliser seulement les données des experts, et considèrent que les données de non-experts contiennent trop de bruit.

Ces travaux montrent que la qualité des traces des experts est supérieure à celle des non-experts. Seulement [O'Donovan and Smyth2005] n'a pas intégré la notion d'experts. À la place, il

associe un score de confiance (*utilité pour le système*) à chaque utilisateur, considérant seulement qu'il y a des utilisateurs plus utiles que d'autres pour le SR. Ce modèle apporte un jugement sur la qualité des traces basée sur l'usage, sans analyse ou explication de ses jugements.

En général, la qualité des traces d'utilisateurs n'est pas la même. Les utilisateurs varient par leurs goûts et préférences, par leurs manières de s'exprimer et d'utiliser le SR, et par leurs objectifs d'utilisation [Ricci *et al.*2011, Herlocker *et al.*2004]. Cette diversité comportementale crée du bruit dans les données acquises, et impacte la précision du SR par un phénomène de fluctuation selon les utilisateurs, où des utilisateurs actifs reçoivent des recommandations de mauvaise qualité. Le SR doit perfectionner sa performance avec les divers types de comportements. Pour cela, il doit analyser et comprendre les traces, afin d'optimiser leur exploitation.

Le problème de fluctuation se manifeste dans différentes approches de recommandation, en affectant différents groupes d'utilisateurs. Ce qui fait que le comportement d'un utilisateur lui permet d'être plus satisfait par une approche de recommandation que par une autre.

Par conséquent, nous menons dans la partie contribution (le chapitre 1) une analyse qualitative du comportement des utilisateurs, pour expliquer le phénomène de fluctuation, avant de regrouper les utilisateurs dans des groupes comportementaux, pour trouver l'approche de recommandation optimale par groupe, dans l'objectif est de rendre le SR plus flexible et adaptable et ainsi de satisfaire le plus d'utilisateurs possible.

Dans ce chapitre, nous menons une investigation sur les SR et le rôle de l'utilisateur dans ces systèmes. Nous commençons par un tour d'horizon sur les approches de recommandation (section 1.1), qui se termine par un tableau de comparaison illustrant les points forts et les points faibles de chaque approche. Ensuite, nous examinons les stratégies d'hybridation des approches précédentes. Finalement, nous discutons les méthodes d'évaluation de SR (section 1.2) pour savoir comment évaluer la performance d'un SR.

1.1 Les approches de la recommandation

La première apparition des SR date de l'année 1992, avec le système *tapestry* [Goldberg *et al.*1992], qui pouvait calculer et envoyer des recommandations aux utilisateurs par email. Les auteurs étaient les premiers à utiliser le terme "filtrage collaboratif". Cinq ans plus tard, Resnick et Varian publiaient leur papier [Resnick and Varian1997] nommé "Recommender systems" dans lequel ils soutiennent que le filtrage collaboratif n'est pas la seule approche pour faire de la recommandation, et ils citent et comparent plusieurs SR de l'époque comme *Fab* [Balabanović and Shoham1997] et *PHOAKS* [Terveen *et al.*1997]. Depuis, de multiples synthèses s'enchaînent pour décrire l'évolution du domaine de la recommandation [Maes *et al.*1999, Tintarev and Masthoff2007, Wei *et al.*2007, Almazro *et al.*2010, Prasad2012, Lü *et al.*2012]. La plupart des chercheurs précédents ont proposé des classifications des approches de la recommandation. Ces classifications sont généralement semblables. La figure 1.1 illustre notre version de la classification.

Essentiellement, il y a deux paradigmes de recommandation, la recommandation à base de connaissances et celle à base de statistiques [Towle and Quinn2000]. Un SR à base de connaissances essaie de s'orienter en fonction des besoins et des préférences de l'utilisateur, afin de trouver l'item qui lui correspond [Burke1999a, Trewin2000]. Des exemples de cette approche existent dans [Smyth B. and Cotter P.2000, Ricci *et al.*2006, Martínez *et al.*2008, Burke1999b]. Plusieurs techniques d'aide à la décision sont disponibles dans cette catégorie de SR, comme le SR à base de cas [Bridge *et al.*2005], celui à base de contraintes [Towle and Quinn2000], et celui à base d'arbre de décisions [Cho *et al.*2002]. Le SR à base de connaissances suit des règles strictes de connaissances, et n'est pas sensible aux changements de comportement à long terme,

ses recommandations ne sont pas personnalisées et il n'évolue pas ou peu avec le temps. Dans le cadre de ce travail, nous nous sommes intéressés uniquement aux SR statistiques qui peuvent s'ajuster aux habitudes des utilisateurs de manière dynamique et évolutive. Dans la suite de ce manuscrit, quand nous parlons des SR, nous nous limitons aux SR statistiques.

En général, l'entrée d'un SR se compose d'un ensemble d'utilisateurs \mathcal{U} , d'un ensemble d'items ou d'articles \mathcal{A} , et d'une base d'informations (notes, annotations, fichiers logs, ...) selon l'approche. Chaque approche dispose d'une fonction de prédiction p qui permet d'extraire les valeurs d'appréciation sous forme de relations entre un utilisateur et un item, $p(u, a)$ est la valeur d'appréciation prédite. Ce qui change d'une approche à une autre est simplement la base d'informations qu'elle utilise, et la fonction de prédiction. Nous faisons un tour d'horizon des principales approches de la recommandation afin de découvrir les interprétations de la fonction de prédiction, leurs qualités et leurs inconvénients.

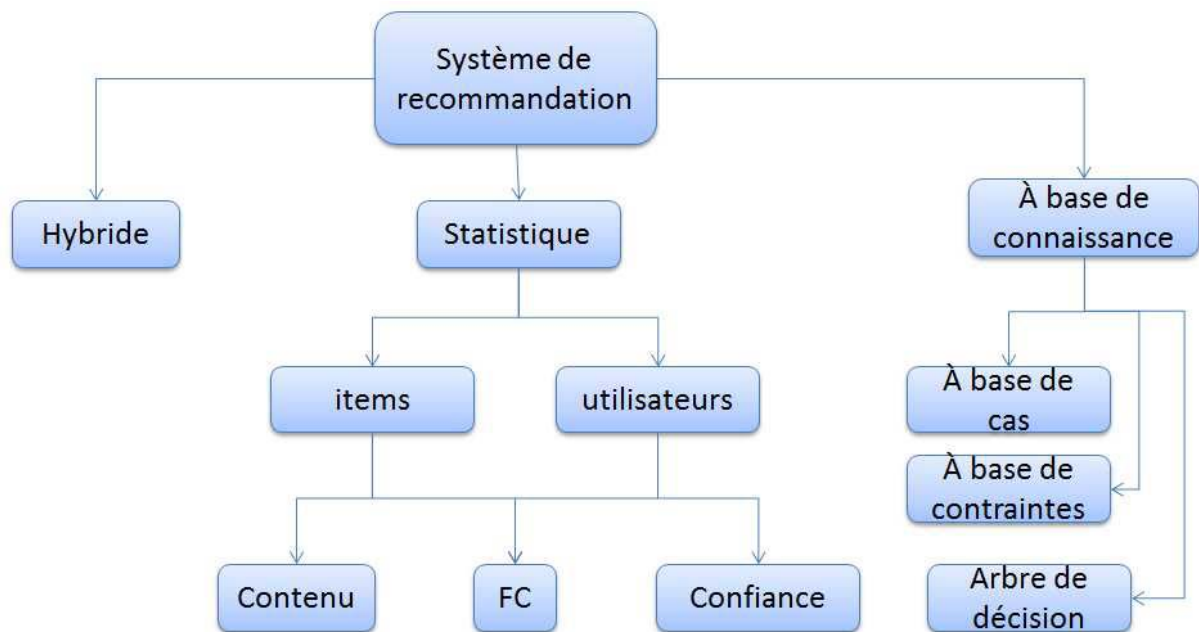


FIGURE 1.1 – La catégorisation des systèmes de recommandation

Bien que notre travail se focalise sur le problème de fluctuation et la satisfaction des utilisateurs divers, nous sommes convaincus que la performance d'un SR est dépendante de plusieurs facteurs. Avant de procéder à notre tour d'horizon, nous allons pointer les principaux facteurs, classés par l'ordre d'importance dans le cadre de cette thèse :

1. La fluctuation : comment le SR gère la diversité des comportements d'utilisateurs, et sa capacité à satisfaire le maximum de ses utilisateurs.
2. Le démarrage à froid : comment le SR traite les nouveaux utilisateurs et les nouveaux items. A priori le SR n'a pas assez d'informations sur ces nouveaux arrivants, alors comment peut-il leur trouver des recommandations pertinentes afin de les attirer pour qu'ils continuent à utiliser le système [Ricci *et al.*2011, Maltz and Ehrlich1995, Schein *et al.*2001, Massa and Avesani2004, Amatriain *et al.*2009a, Bambini *et al.*2011] ?
3. La résistance aux attaques : les SR sont exposés aux attaques malveillantes [Mobasher *et al.*2007]. Le but de ces attaques est varié comme contrôler les recommandations des

utilisateurs afin d'y passer des publicités indésirables, ou détériorer fortement le SR [Lam and Riedl2004, O'Mahony *et al.*2004, Williams *et al.*2007, Burke *et al.*2005, Shi *et al.*2006].

4. Le manque de données : ce problème est dû au fait que les utilisateurs ne fournissent pas assez de traces, surtout explicites [Castagnos2008]. C'est le cas souvent avec des grands catalogues d'items.
5. Le traitement des grosses masses de données et le passage à l'échelle.
6. La gestion de la relation avec l'utilisateur : cela comprend plusieurs caractéristiques qui servent à attirer les utilisateurs :
 - La prise en compte du jugement de qualité d'utilisateurs.
 - La capacité du SR à expliquer ses recommandations à l'utilisateur.
 - La détection des relations entre les items sur la base d'usage (comme le problème du croissant et du café, où deux items n'ont pas de caractéristiques communes sauf qu'ils sont souvent consommés ensemble).
 - La notion d'aubaine (sérendipité²) qui mesure la capacité du SR à être innovant, et à créer des surprises favorables pour l'utilisateur.
7. La difficulté de conception de mise en œuvre du SR.

Les SR appliquent des méthodes statistiques sur les utilisateurs ou sur les items pour calculer les recommandations [Ricci *et al.*2011]. Les approches sont divisées en deux classes : SR à base d'items, et SR à base d'utilisateurs. Dans la section suivante, nous allons expliquer chacune de ces deux classes, et les approches principales.

1.1.1 Systèmes de recommandation à base d'items

Les approches dans cette classe appliquent les calculs statistiques sur les items. L'idée fondamentale est de proposer à l'utilisateur des items similaires à ceux qu'il a aimés. Deux approches sont principalement connues dans cette classe : la recommandation à base de contenu [Basu *et al.*1998] et le filtrage collaboratif à base d'items [Sarwar *et al.*2001].

1.1.1.1 Systèmes de recommandation à base de contenu

Dans cette approche, chaque item est défini par un ensemble d'attributs et leur valeur. Les items ayant des valeurs proches pour leurs attributs sont considérés approchants (similaires). Quand un utilisateur attribue une bonne note à un item, les items approchants sont considérés comme des recommandations potentielles [Van Meteren and Van Someren2000, Basu *et al.*1998, Pazzani1999, Pazzani and Billsus2007]. Des applications de cette approche existent aussi dans [Loh *et al.*2008, Shin *et al.*2008, Elmisery and Botvich2012].

L'approche construit le profil d'un utilisateur en examinant les centres d'intérêt de ce dernier. Pour ce faire, il s'appuie sur les traits des items que l'utilisateur a notés.

Par exemple, dans un contexte de la recommandation de livres, si un utilisateur attribue de bonnes notes à "Les Hauts de Hurlevent" et à "Anna Karenine", le système conclut que cet utilisateur aime les livres romantiques du dix-neuvième siècle. Par conséquent, il va lui recommander des livres du même style et de la même époque (comme "La lettre écarlate" par exemple).

2. résultats fructueux et inattendus, ce terme est attesté en français dès 1954, sans être reconnu par l'Académie Française, il est la traduction du mot anglais "serendipity" apparu en 1754, et formé de l'histoire "Voyages et aventures des trois princes de Serendip", qui découvraient souvent accidentellement, des choses qu'ils ne cherchaient pas. Serendip vient de l'ancien nom que les Arabes ont utilisé pour l'actuel Sri Lanka, qui était à son tour une adaptation de "Simhaladvipa" en Sanskrit qui veut dire l'île habitée par les lions.

1.1.1.2 Filtrage collaboratif à base d'items

La similarité entre items ici est basée sur le jugement des utilisateurs. Pour que deux items soient considérés voisins (similaires), ils doivent être appréciés par les mêmes utilisateurs [Sarwar *et al.*2001, Linden *et al.*2003]. Pour le FC à base d'items, la représentation d'un item peut se limiter à son identifiant. Si on reprend l'exemple de la recommandation de livres, le SR ne sait pas quel genre de livres sont "Les Hauts de Hurlevent" et "Anna Karenine". Il sait juste que les utilisateurs qui ont apprécié ces deux livres, ont aussi apprécié "la lettre écarlate", ce qui permet d'inférer que les trois livres sont voisins.

Diverses mesures de similarité peuvent être utilisées dans cette approche, mais d'après plusieurs études, la similarité cosinus ajustée est le plus souvent efficace [Schafer *et al.*2007, Zhang *et al.*2009].

Le FC existe en deux versions : le FC à base d'items et le FC à base d'utilisateurs [Desrosiers and Karypis2011, Lousame and Sánchez2009, Su and Khoshgoftaar2009, Wang *et al.*2006]. Le FC à base d'utilisateurs est le plus fréquent [Zhang *et al.*2009].

1.1.2 Systèmes de recommandation à base d'utilisateurs

Les approches à base d'utilisateurs considèrent qu'un utilisateur a tendance à privilégier les items appréciés par les utilisateurs qui lui sont similaires. Chaque approche interprète la similarité différemment, le FC considère que les utilisateurs sont voisins s'ils ont un comportement similaire à l'égard des items (ils attribuent des notes proches aux items) [Resnick and Varian1997]. En revanche, le SRC considère qu'un utilisateur a une tendance à imiter le comportement et suivre les conseils des utilisateurs auxquels il fait confiance [Massa and Avesani2004]. Le travail réalisé dans cette thèse s'appuie sur ces deux approches. Les sous-sections suivantes présentent le fonctionnement, les qualités et les inconvénients de ces deux approches.

1.1.2.1 Filtrage collaboratif (FC)

Le FC est l'approche de recommandation la plus répandue [Resnick *et al.*1994, Shardanand and Maes1995, Konstan *et al.*1997, Schafer *et al.*2007, Hill *et al.*1995]. Il donne souvent de bons résultats, et ne nécessite pas beaucoup de préparation de données pour démarrer [Ricci *et al.*2011]. Il faut juste avoir une quantité suffisante de notes utilisateurs-items pour que l'algorithme du FC puisse faire son apprentissage.

Les systèmes de recommandations sont arrivés sur le marché au moment où beaucoup de prestataires de services avaient des quantités de données non-exploitées (les données d'achats, les fichiers logs et même des notations). Les approches à base de connaissance et à base de contenu demandaient des efforts supplémentaires de la part des prestataires afin de préparer le modèle de connaissance ou bien les descriptions des items, alors que le FC était prêt à être appliqué sans cette phase d'ajustement de données [Desrosiers and Karypis2011].

L'entrée du FC est une simple matrice V de notes de taille $n \times m$; où n est le nombre total d'utilisateurs \mathcal{U} , et m est le nombre total d'items \mathcal{A} . L'élément v_{ij} de la matrice représente la note attribuée par l'utilisateur u_i à l'item a_j . S'il se trouve que u_i n'a pas attribué une note à a_j , la valeur v_{ij} est considérée comme valeur manquante ($v_{ij} = nul$). L'objectif du FC est donc réduit à simplement estimer les valeurs manquantes de la matrice V .

Le FC utilise les valeurs existantes dans la matrice pour calculer les valeurs manquantes. Pour calculer une valeur v_{ij} , il se sert des voisins de u_i qui ont déjà noté a_j . Pour les extraire, il mesure la similarité entre u_i et chacun des utilisateurs ayant noté l'item a_j . Plusieurs mesures de similarité sont utilisables [Breese *et al.*1998, Resnick *et al.*1994, Bank and Cole2008, Lü and

Zhou2011], mais la plus fréquente est le coefficient de corrélation de Pearson [Resnick *et al.*1994]. La valeur de cette mesure varie entre -1 (les deux utilisateurs sont complètement opposés) et $+1$ (les deux utilisateurs sont complètement corrélés). La formule de Pearson pour calculer la corrélation entre deux utilisateurs u_i et u_j est [Shardanand and Maes1995] :

$$pearson(u_i, u_j) = \frac{\sum_{a \in A_{i,j}} (v(u_i, a) - \bar{v}_{u_i})(v(u_j, a) - \bar{v}_{u_j})}{\sqrt{\sum_{a \in A_{i,j}} (v(u_i, a) - \bar{v}_{u_i})^2 \sum_{a \in A_{i,j}} (v(u_j, a) - \bar{v}_{u_j})^2}} \quad (1.1)$$

où : $A_{i,j}$ représente le sous-ensemble d'items notés par u_i et u_j à la fois.

$v(u_i, a)$ est la note attribuée par u_i à l'item a .

\bar{v}_{u_i} est la moyenne des notes de u_i .

Ensuite, le FC utilise la différence entre la note attribuée à a_j par chaque voisin de u_i , et la moyenne de notes du voisin. Cette différence est multiplié par la similarité entre u_i et le voisin. La moyenne de valeurs obtenues est par tous les voisins est ajoutée à la note moyenne de u_i pour obtenir la note prédite $v(u_i, a_j)$. La fonction de prédiction est donnée par l'équation suivante [Breese *et al.*1998] :

$$v(u_i, a_j) = \bar{v}_{u_i} + \frac{\sum_{u_k \in U_{a_j}} pearson(u_i, u_k) \times (v(u_k, a_j) - \bar{v}_{u_k})}{card(U_{a_j})} \quad (1.2)$$

où : U_{a_j} est le sous-ensemble d'utilisateurs ayant noté a_j .

Exemple (FC) : Ayant l'ensemble d'utilisateurs $\mathcal{U} = \{u_0, u_1, u_2, u_3, u_4\}$, et l'ensemble d'items $\mathcal{A} = \{a_0, a_1, a_2, a_3, a_4\}$, dans un système où la notation est numérique entre 1 et 5, et la matrice de notes suivante :

	a_0	a_1	a_2	a_3	a_4
u_0	-	5	1	-	3
u_1	3	3	1	-	-
u_2	-	5	-	5	5
u_3	-	-	-	4	-
u_4	1	3	4	2	5

TABLE 1.1 – Matrice de notes

Supposons qu'on veuille prédire la valeur $v_{0,0}$, on commence par sélectionner les voisins de u_0 qui ont noté a_0 , $A_{a_0} = \{u_1, u_4\}$. Par la suite, on calcule la similarité entre u_0 et chaque membre de A_{a_0} en utilisant l'équation 1.1.

La moyenne des notes pour chaque utilisateur est :

$$\bar{u}_0 = 3 \quad \bar{u}_1 = 2,33 \quad \bar{u}_2 = 5 \quad \bar{u}_3 = 4 \quad \bar{u}_4 = 3$$

$$pearson(u_0, u_1) = \frac{(5 - 3) \times (3 - 2,33) + (1 - 3) \times (1 - 2,33)}{\sqrt{((5 - 3)^2 + (1 - 3)^2) \times ((3 - 2,33)^2 + (1 - 2,33)^2)}} = 0,94$$

La similarité presque complète entre u_0 et u_1 peut paraître incohérente, puisque les deux utilisateurs n'ont pas toujours attribué de notes identiques. Il est vrai que u_0 a attribué 5 à a_1 alors que u_1 a attribué 3, mais il faut bien voir que u_1 n'a jamais attribué de valeur supérieure à 3, cela signifie que pour lui 3 représente vraisemblablement une appréciation maximale. En

conclusion, chacun des deux utilisateurs a exprimé son appréciation totale à l'égard de a_1 , et sa dépréciation totale à l'égard de a_2 , ce qui explique le fait que la mesure de Pearson les a considérés similaires l'un à l'autre.

$$\text{pearson}(u_0, u_4) = \frac{(5-3) \times (3-3) + (1-3) \times (4-3) + (3-3) \times (5-3)}{\sqrt{((5-3)^2 + (1-3)^2 + (3-3)^2) \times ((3-3)^2 + (4-3)^2 + (5-3)^2)}} = -0,32$$

Ensuite, on applique l'équation 1.2 pour estimer la note $v_{0,0}$:

$$v_{0,0} = 3 + \frac{0,94 \times (3-2,33) + (-0,32) \times (1-3)}{2} = 3,63$$

La mesure de Pearson est insensible à la taille de la liste. Supposons que les utilisateurs u_a et u_b ont exactement 5 notes communes et identiques, et que u_a et u_c ont 50 notes en commun dont 49 identiques. Pour la mesure de Pearson, la corrélation entre u_a et u_b est supérieure à celle entre u_a et u_c .

FC à base de mémoire et à base de modèle L'exemple de FC ci-dessus montre le FC dans sa version initiale. On l'appelle FC à base de mémoire [Delgado and Ishii1999, Konstan *et al.*1997, Linden *et al.*2003, Nakamura and Abe1998, Yu *et al.*2004]. Les études montrent qu'on peut améliorer la précision en détectant et en éliminant certains voisins qui sont à l'origine de bruit dans la formule 1.2. Il est possible de garder seulement les N voisins les plus similaires [Herlocker *et al.*1999, Resnick *et al.*1994], ou les voisins dont la similarité surpasse un seuil donné [Shardanand and Maes1995]. Cependant des études récentes montrent qu'engager les utilisateurs dissimilaires dans la prédiction peut aussi améliorer la performance [Zeng *et al.*2010, Zeng *et al.*2011].

Le FC à base de mémoire souffre essentiellement de deux problèmes : le passage à l'échelle et le manque de données [Xue *et al.*2005]. Le passage à l'échelle fait que le temps de calcul est considérablement long lors de son application sur des données massives. Le manque de données vient du fait qu'il n'est pas toujours facile de trouver suffisamment de notes communes entre les utilisateurs, particulièrement sur des grands catalogues d'items. Sachant que le FC utilise des mesures statistiques pour calculer la similarité, et que ces mesures demandent un nombre minimal de valeurs pour fonctionner, le FC est déstabilisé par le manque de données [Koren2008, Takács *et al.*2007].

L'objectif principal du FC à base de modèle [Koren and Bell2011] est de réduire l'impact de ce problème de passage à l'échelle et de celui du manque de données. Il s'appuie sur des patrons d'apprentissage plus complexes pour choisir les voisins, comme : la classification Bayésienne [Breese *et al.*1998], l'analyse sémantique latente [Hofmann2003], l'entropie maximisée [Zitnick and Kanade2012], la machine de Boltzmann [Salakhutdinov *et al.*2007], la machine à vecteurs de support [Grčar *et al.*2006], et la factorisation de matrice [Bell *et al.*2007, Koren2008, Paterek2007, Takacs *et al.*2008]. Le FC à base de modèle consomme normalement du temps pour construire le modèle et le mettre à jour. Il est aussi incapable de couvrir autant de diversité d'utilisateurs que la méthode à base de mémoire [Xue *et al.*2005], mais il est en général plus précis [Koren *et al.*, Takács *et al.*2007].

1.1.2.2 Systèmes de recommandation à base de confiance (SRC)

Ils sont appelés aussi recommandation sociale³. Sur le principe, la recommandation à base de confiance ressemble au FC à base d'utilisateurs, avec deux différences majeures. Premièrement, la notion de voisinage est remplacée par celle de confiance. Ainsi, c'est l'utilisateur lui-même qui choisit ses voisins nommés désormais les amis. Deuxièmement, le système est équipé d'une fonction de propagation de confiance qui permet à l'utilisateur de profiter de l'expérience des amis de ses amis [Massa and Avesani2004, Golbeck and Hendler2006, Walter *et al.*2008].

La modélisation de confiance est devenue incontournable dans plusieurs branches du web social, comme la sécurité [Kagal *et al.*2001], le partage des fichiers [Kamvar *et al.*2003], le web sémantique [Artz and Gil2007], les systèmes multi-agents [Ramchurn *et al.*2004, Huynh *et al.*2006], le cloud computing [Li and Ping2009] etc. Le domaine de la recommandation n'a pas été épargné par cette émergence de la notion de confiance. Plusieurs modèles ont été proposés pour faire de la recommandation à base de confiance, comme dans [Massa and Avesani2004, Golbeck and Hendler2006, Walter *et al.*2008, O'Donovan and Smyth2005, Avesani *et al.*2005, Victor *et al.*2009]. Plusieurs approches qui étaient à la base proposées pour d'autres domaines sont utilisables dans le domaine de la recommandation.

Comme dans tous les domaines du web, la confiance dans les systèmes de recommandation a été appliquée selon deux axes principaux : la confiance comme une relation entre deux individus (confiance locale), la confiance comme une synthèse de l'avis de la communauté à l'égard d'un individu (confiance globale ou réputation) [Ziegler and Lausen2004]. La première forme (confiance locale) est la plus fréquente dans les systèmes de recommandation.

Les SRC diffèrent entre eux en plusieurs points comme : (1) la construction et la représentation du lien de confiance, (2) la propagation et l'agrégation des liens de confiance, (3) l'intégration du contexte de confiance. Les modèles de confiance seront détaillés dans le chapitre suivant de l'état de l'art.

1.1.3 Synthèse sur les approches de recommandation

Le tableau 1.2 résume les qualités et les défauts des approches de recommandation présentées dans ce chapitre.

Nous utilisons ce tableau pour donner une vision globale des approches de recommandation et leurs catégories. Nous résumons sur plusieurs critères les approches présentées ici.

3. Avant l'apparition de cette catégorie, ce terme a été utilisé dans certains cas pour le FC. Il est vrai que le FC est un modèle implicitement social, mais nous préférons réserver ce terme pour les modèles à base de confiance, car ils sont explicitement sociaux

		contenu	FC items	FC utilisateur	Confiance
Fluctuation		⊖ oui	⊖ oui	⊖ oui	⊖ oui
Démarrage à froid	côté utilisateur	⊖ oui	⊖ oui	⊖ oui	⊕ moyenne
	côté item	⊕ non	⊖ oui	⊖ oui	⊖ oui
Résistance		⊕ oui	⊖ non	⊖ non	⊕ oui
Manque de données		⊖ oui	⊖ oui	⊖ oui	moyenne
Gestion de relation avec l'utilisateur	Prendre en compte le jugement de qualité des utilisateurs	⊖ non	⊕ oui	⊕ oui	⊕ oui
	Explication de recommandation	⊕ oui	⊖ non	⊖ non	⊕ oui
	Détection des relations entre items à base de l'utilisation commune	⊖ non	⊕ oui	⊕ oui	⊕ oui
	Aubaine	⊖ non	⊕ oui	⊕ oui	⊕ oui
Passage à l'échelle		⊕ non	⊖ oui	⊖ oui	⊕ non
Difficulté de conception		⊖ oui	⊕ non	⊕ non	⊕ non

TABLE 1.2 – Comparaison des approches de recommandation

Fluctuation

Dans l'approche à base de contenu, la précision de recommandation baisse fortement si l'utilisateur note d'avance la plupart des items d'une catégorie donnée [Van Meteren and Van Someren2000]. Par exemple, si l'utilisateur note la plupart des livres romantiques du dix-neuvième siècle, le système aura du mal à lui générer des recommandations pertinentes.

Dans le FC, la fluctuation vient de l'aspect qualitatif des notes. Le FC favorise les utilisateurs qui suivent l'orientation de la communauté. Par exemple, le problème des moutons noirs apparaît chez les utilisateurs ayant un goût atypique [Vozalis and Margaritis2003, Su and Khoshgoftaar2009, Vekariya and Kulkarni2012, Zuva *et al.*2012], car ils notent des items qui ne sont pas fréquemment notés, ou bien ils notent à l'opposé de l'orientation de la communauté. La performance du FC est dépendante de la qualité des notes fournies par l'utilisateur et non seulement de leur quantité. La difficulté de ce problème réside dans la diversité des objectifs des utilisateurs qui utilisent le SR, ainsi que dans la variété des comportements observés pour réaliser leurs objectifs.

Un phénomène de fluctuation peut être aussi identifié dans les SRC. Cela est dû au fait que certains utilisateurs n'ont pas l'intention de se faire des amis, ou de noter d'autres utilisateurs. Ils exploitent le SR en vue seulement de noter des items. Bien qu'ils soient actifs, le SRC ne peut pas leur recommander des items. D'autres causes, comme la qualité des amis (réputés, non-réputés), impactent aussi la qualité de recommandation et génèrent de la fluctuation.

La fluctuation de performance est un problème général dans les SR. Il est peu évoqué, et

quand c'est le cas, il est résumé par le problème des moutons noirs.

Démarrage à froid

L'approche à base de contenu souffre du démarrage à froid du côté des utilisateurs, mais pas du côté des items [Ricci *et al.*2011], car les nouveaux items peuvent être directement associés à leurs approchants grâce à leurs valeurs d'attributs.

Le FC souffre du démarrage à froid du côté des utilisateurs et du côté des items. Il ne peut pas trouver de voisins à un nouvel utilisateur qui n'a pas encore attribué de notes, ou a attribué peu de notes. Il ne peut pas non plus recommander de nouveaux items qui n'ont pas encore été notés par les utilisateurs [Bambini *et al.*2011].

Dans l'approche à base de confiance, l'impact du démarrage à froid est réduit par rapport au FC [Massa and Bhattacharjee2004, Jamali and Ester2009]. Il suffit pour un utilisateur du SRC de noter un seul utilisateur, pour créer un lien entre les deux utilisateurs. Ceci est, dans certains cas, suffisant pour commencer à recevoir des recommandations. Le FC a besoin que les deux utilisateurs aient plusieurs notes en commun pour qu'il puisse établir ce lien entre eux. Il est clair que la tâche demandée à l'utilisateur par le SRC est largement plus facile.

Pour résumer, le FC est l'approche la plus touchée par le problème du démarrage à froid. Toutefois, ce problème impacte moins les SR à base de contenu et à base de confiance. Le FC n'est pas très approprié aux applications où le nombre d'utilisateurs augmente très rapidement, où l'historique d'un utilisateur est court.

Résistance

Les SR à base de contenu ne sont pas vulnérables aux attaques malveillantes parce que les retours d'un utilisateur n'influencent que les recommandations qui lui sont adressées et non celles faites aux autres utilisateurs. En revanche, les SR collaboratifs (FC et SRC) sont, à différents degrés, fragiles aux attaques.

Le FC permet à ses utilisateurs de participer au calcul de recommandation de leurs voisins, sans l'autorisation explicite de ces derniers. Cela encourage certains individus à tenter de prendre le contrôle du système, et réorienter la recommandation à des fins diverses [Mobasher *et al.*2007], comme contourner le système de recommandation pour qu'il recommande les items (produits, sources) fournis par l'attaquant [Mobasher *et al.*2006], transmettre des virus ou des chevaux de Troie dans des liens ou des fichiers partagés, ou même bloquer le système dans sa totalité. Plusieurs formes d'attaques existent dans [Lam and Riedl2004, O'Mahony *et al.*2004, Williams *et al.*2007, Burke *et al.*2005, Shi *et al.*2006].

Dans les SRC, c'est l'utilisateur lui-même qui choisit ses amis, ce qui rend une attaque malveillante par injection de profils pratiquement irréalisable, à moins que l'attaquant puisse trouver une ruse pour convaincre sa victime de l'ajouter volontairement à ses amis, et même quand cela arrive, la victime peut à tout moment bloquer l'attaquant [Ma *et al.*2009].

Les SR à base de confiance globale sont plus fragiles aux attaques de groupe que ceux à base de confiance locale. Un groupe d'attaquants ou de profils peut coopérer pour améliorer la réputation d'un membre du groupe, qui piège ensuite les recommandations [Hoffman *et al.*2009].

Manque de données

L'approche à base de contenu exige une description complète et détaillée des items; les descriptions peuvent être données manuellement au SR, ou extraites automatiquement [Pazzani and Billsus2007]. Cette approche est sensible au manque de données de côté des descriptions.

Dans le FC, quand le catalogue d'items est grand, même un utilisateur hyper actif n'arrive pas à noter 1% du catalogue, et la matrice des notes sera quasi-vide. Il sera donc difficile de trouver des notes communes entre les utilisateurs. Le manque de données est un problème assez fréquent en FC [Konstan *et al.*1997, Bergholz2003, Adomavicius and Tuzhilin2004, Grčar *et al.*2006, Chen *et al.*2010, Pitsilis and Knapskog2012].

L'approche à base de confiance s'appuie sur les notes mutuelles entre utilisateurs, cela est plus efficace pour remplir la matrice de similarité [Jamali and Ester2009, Lee and Brusilovsky2009]. De plus, la propagation de confiance sert aussi le même but. Par conséquent, trouver des amis en SRC est plus aisé que de trouver des voisins au FC [Pitsilis and Knapskog2012]. Cela réduit le manque de données et augmente l'efficacité du système.

Gestion de la relation avec l'utilisateur

En fait, certains items sont reliés les uns avec les autres par leur usage commun, sans avoir de descriptions communes. Beaucoup de gens, par exemple, prennent du café avec un croissant : sur le plan descriptif, il n'y a pas de lien fort de similarité entre une boisson chaude et un dessert, c'est simplement l'usage qui fortifie, voire crée, ce lien. L'approche à base de contenu n'est pas capable de découvrir ce genre de lien, pourtant assez fréquent en pratique.

Le FC et le SRC prennent en compte le jugement d'utilisateurs. Ils sont donc capables de détecter les relations entre les items à base d'utilisation commune. Cela les rend plus efficaces pour la recommandation des sous-ensembles intégraux [Ricci *et al.*2011], et plus adaptatifs aux comportements d'utilisateurs.

Dans l'approche à base de contenu, où les relations de similarité n'évoluent pas [Iaquinta *et al.*], l'aubaine n'est pas atteinte. En revanche, le FC et le SRC peuvent assurer l'aubaine grâce aux liens évolutifs à base d'utilisation commune.

Les SR à base de contenu et SRC sont plus transparents que le FC et plus capables d'expliquer leurs résultats.

Passage à l'échelle

Le FC est le plus affecté par le problème du passage à l'échelle, à cause de la phase de calcul des similarités. Cependant, plusieurs techniques ont été développées pour diminuer l'impact du passage à l'échelle surtout dans les FC à base de modèle comme dans [Grčar *et al.*2006, Takács *et al.*2007, Paterek2007, Koren and Bell2011, Ben Ticha *et al.*2013].

Difficulté de conception

L'approche à base de contenu exige une description complète et détaillée du contenu ; cette opération est souvent faite manuellement par le prestataire de service, ce qui rend le système coûteux. Même les méthodes automatiques ne peuvent capturer que certains aspects du contenu, ce qui génère des représentations partielles de similarité [Pazzani and Billsus2007].

Le FC est une approche simple à appliquer (surtout à base de mémoire), elle peut être appliquée sans beaucoup de connaissances sur le domaine, ou ajustement de données ; elle donne souvent de bons résultats [Su and Khoshgoftaar2009, Desrosiers and Karypis2011]. L'avantage principal de cette approche est qu'elle n'a pas besoin de connaissance préalable ni des items ni des utilisateurs.

Le tableau 1.2 résume les principaux facteurs de performance des systèmes de recommandation. Les approches s'apposent à des termes de qualités et de défauts. Cela a encouragé l'appa-

rition des SR hybrides. L'hybridation sert à combiner plusieurs approches de recommandation pour que chacune d'entre elles pallie les défauts de l'autre [Burke2002].

Dans notre étude, nous nous focalisons sur l'adaptation aux divers comportements des utilisateurs. Dans cette optique, les approches à base d'utilisateurs sont les meilleures. Cependant, elles souffrent toutes du problème de fluctuation vis-à-vis de différents sous-ensembles d'utilisateurs. Un système hybride peut réduire cette fluctuation. Dans la section suivante nous examinons les SR hybrides et leurs techniques.

1.1.4 Les systèmes de recommandation hybrides

Il ne s'agit pas d'une approche indépendante, mais plutôt de combiner plusieurs approches de recommandation [Burke2002]. Pour prédire une note, dans un SR hybride, chaque approche calcule une valeur de similarité entre les deux utilisateurs (score local). Le SR hybride combine les scores pour calculer un score final. La différence entre les techniques d'hybridation réside dans la stratégie choisie pour combiner les scores locaux. L'hybridation peut être appliquée au niveau des notes, ou bien au niveau de la fonction de prédiction. Robin Burke résumait les techniques d'hybridation en sept techniques [Burke2007]. Parmi les sept techniques, trois exigent l'existence d'une approche à base d'items parmi les approches hybridées. Cette thèse approfondit les approches de recommandation à base d'utilisateurs, ce qui fait que ces techniques ne sont pas utilisables pour nos travaux. Les quatre autres fonctionnent indépendamment des catégories des approches ; nous en ajoutons une cinquième (probabiliste). Ici nous donnons les formules des cinq techniques utilisables pour nos travaux :

1. pondérée (weighted) : cette technique combine numériquement les scores venant des deux techniques de recommandation. L'équation suivante peut être utilisée :

$$scoreFinal = \alpha \times score_1 + (1 - \alpha) \times score_2 \quad 0 < \alpha < 1$$

Cette stratégie est flexible, il faut ajuster le paramètre α pour quantifier la contribution de chaque approche. Un exemple pour la recommandation de films est disponible dans [Mobasher *et al.*2004].

2. technique d'alternance (switch) : le système choisit à chaque fois une technique parmi d'autres et applique son résultat [Billsus and Pazzani2000]. On peut considérer que le système tente de prédire la note par la première approche et utilise la deuxième en cas d'échec.

$$scoreFinal = \begin{cases} score_1 & \text{si } score_1 \neq nul \\ score_2 & \text{si } score_1 = nul \end{cases}$$

3. mixte (mixed) : les deux techniques effectuent leurs calculs indépendamment, seuls les meilleurs scores sont retenus ; pour appliquer cela, on peut choisir par exemple une liste qui rassemble les cinq meilleurs items selon chaque technique, les agréger dans une seule liste, et présenter la liste à l'utilisateur [Smyth B. and Cotter P.2000], ou bien chaque approche calcule les scores de tous les items ; chaque item a le score le plus élevé parmi les deux, les items sont ensuite listés selon leurs scores, et les meilleurs sont recommandés.

$$scoreFinal = max(score_1, score_2)$$

4. cascade : cette hybridation est strictement hiérarchique ; une technique inférieure dans la hiérarchie ne peut renverser la décision de celle qui lui est supérieure mais plutôt l'affiner.

Un exemple est donné dans [Burke2002]. Cette technique exige que l’item ait un score dans toutes les approches :

$$scoreFinal = \begin{cases} score_2 & \text{si } score_1 > 0 \\ 0 & \text{sinon} \end{cases}$$

5. probabiliste : l’idée ici est de favoriser les items ayant des scores avec les deux approches au regard de ceux qui en ont seulement une seule. Le score final est donné par l’équation :

$$scoreFinal = 1 - (1 - score_1) * (1 - score_2)$$

1.2 Évaluation des systèmes de recommandation

Nous allons examiner comment on peut évaluer la performance du SR pour s’assurer de sa capacité à satisfaire les besoins qui ont conduit à sa création. Le choix d’une mesure, doit être dépendant du type de données à traiter, et des intérêts des utilisateurs [Herlocker *et al.*2004].

Comme le domaine de la recommandation dérive du domaine de la recherche d’information, il est donc souvent normal d’utiliser des mesures d’évaluation de la recherche d’information [Voorhees2002]. Certaines de ces mesures ont été ajustées aux besoins du domaine de la recommandation.

L’évaluation de performance en SR dans la littérature est souvent limitée au calcul de la précision de prédiction [Herlocker *et al.*2004]. La précision mesure, en général, la différence entre les valeurs des notes prédites par le système de recommandation et les valeurs réellement fournies par les utilisateurs.

L’évaluation de systèmes de recommandation suit, en général, une des trois méthodes : hors ligne, études sur un échantillon d’utilisateurs ou évaluation en ligne [Shani and Gunawardana2011].

L’évaluation hors ligne est la plus simple à réaliser et la moins risquée. Il s’agit globalement de diviser les données disponibles en deux parties, la partie d’apprentissage et la partie test, avant d’utiliser la partie d’apprentissage pour prédire la partie test. Ce type d’évaluation ne pose pas de problème de fuite d’utilisateurs, alors on peut prendre le risque de tester même des approches très fluctuantes. Elle permet d’intégrer facilement une grande masse d’utilisateurs. Elle n’est pas très sensible aux potentiels changements dans le comportement de l’utilisateur. Elle essaie de recopier un comportement que l’utilisateur a eu alors que le système de recommandation n’intervenait pas pour conseiller l’utilisateur, mais elle n’est pas capable de mesurer l’impact d’une telle intervention [Shani and Gunawardana2011].

La deuxième méthode (études sur un échantillon) consiste à recruter un groupe de volontaires, auxquels on demande d’exécuter des tâches bien précises en utilisant le SR, de surveiller et d’enregistrer leurs comportements durant l’expérimentation. Ensuite, on peut aussi poser des questions aux participants concernant leurs impressions sur l’expérimentation et le SR.

Ce type de test peut répondre à un large éventail de questions. Il permet de suivre le comportement d’un utilisateur au cours de son interaction avec le SR, et d’observer si ce dernier a influencé le comportement de l’utilisateur. Le questionnaire permet aussi de collecter des données qualitatives pour expliquer les résultats quantitatifs.

Ce type d’évaluation est coûteux. Il n’est pas toujours facile de recruter un nombre suffisant d’utilisateurs, parfois il faut les motiver par des récompenses ou des dédommagements. Le nombre de participants est souvent limité, et on ne peut pas tirer de conclusions concernant une masse d’utilisateurs. De plus, à cause des contraintes de temps des participants, on ne peut pas leur

demander de faire des tests excessivement longs. Néanmoins, chaque scénario doit être répété plusieurs fois afin d'assurer la fiabilité du résultat [Shani and Gunawardana2011].

Le dernier type est l'évaluation en ligne. Elle est appliquée sur les vrais utilisateurs du système en temps réel [Kohavi *et al.*2009]. Ce test peut être une simple comparaison des chiffres d'affaires avant et après l'application du SR. On l'applique sur un échantillon d'utilisateurs (tirés au hasard), on observe leurs réactions, et on les compare avec ceux du reste de la population. Ce type d'évaluation comporte des risques. En effet, on peut perdre un utilisateur si le système recommande des items non pertinents. Pour cette raison, on recommande de procéder préalablement à une évaluation hors risque afin de garantir un minimum de qualité de recommandation [Shani and Gunawardana2011].

Les caractéristiques des tests font que le premier type d'évaluation (hors ligne) est le plus approprié à notre travail, car ce travail se focalise sur des phénomènes comportementaux chez un grand nombre d'utilisateurs. Afin que les résultats émergent, il faut intégrer le comportement d'un nombre considérable d'utilisateurs. Un test à base d'échantillonnage ne sera pas apte à conclure par ce genre de résultats. Dans la section suivante, nous détaillerons les tests hors ligne en montrant les mesures de qualité utilisées pour valider nos résultats.

1.2.1 Évaluation par tests hors ligne

Il est nécessaire de commencer par distinguer deux stratégies selon lesquelles les SR communiquent leurs résultats aux utilisateurs.

Dans la première, le système répond à la question : *est-ce que l'utilisateur va apprécier cet item ?* Ces systèmes cherchent à prédire toutes les valeurs manquantes de la matrice de notes, et affichent leur valeur prédite à côté de l'item lors de sa consultation par l'utilisateur, MovieLens est un exemple de cette stratégie.

Dans la deuxième, le SR répond à la question suivante : *Quels sont les items que l'utilisateur va apprécier ?* Les systèmes qui suivent cette stratégie donnent en sortie une liste ordonnée des meilleurs items que l'utilisateur va apprécier, la valeur numérique de la note prédite n'est pas une priorité, l'essentiel est que la liste contienne des items pertinents.

La mesure d'évaluation doit s'accorder avec la stratégie suivie par le système. Deux classes de mesures d'évaluation sont utilisées : la classe des mesures de précision prédictive, et la classe des mesures de précision du classement [Herlocker *et al.*2004].

Mesures de précision prédictive

Le but de ces mesures est d'évaluer la précision de la prédiction. La mesure la plus connue dans cette catégorie est l'erreur moyenne absolue (MAE) [Shardanand and Maes1995], c'est une mesure statistique qui s'appuie sur la moyenne des différences entre chaque valeur prédite et sa valeur réelle :

$$MAE = \frac{\sum_{i=1}^N |p_i - a_i|}{N} \quad (1.3)$$

Cette mesure est bonne pour évaluer globalement la capacité du SR à prédire les notes des utilisateurs [Herlocker *et al.*1999, Melville *et al.*2002, Vozalis and Margaritis2003, Massa and Avesani2004, Xue *et al.*2005, Amatriain *et al.*2009a]. Un grand nombre de variantes de cette mesure ont été proposées pour évaluer de manière plus pointue la précision [Andersen and Fenandez-Luna2012] :

	L'item est sélectionné	L'item n'est pas sélectionné	Somme
Pertinent	Nb_{ps}	Nb_{pr}	Nb_p
Impertinent	Nb_{ms}	Nb_{mr}	Nb_m
	Nb_s	Nb_r	Nb

TABLE 1.3 – F-Mesure

- Erreur moyenne absolue normalisée (NMAE) : c'est la version normalisée de MAE. Elle sert principalement à comparer deux modèles où les échelles de notation ne sont pas égales.

$$NMAE = \frac{MAE}{v_{max} - v_{min}} \quad (1.4)$$

où v_{max} et v_{min} sont respectivement la valeur maximale et minimale de notation.

- Erreur moyenne absolue par utilisateur (UMAE) : cette mesure est très importante pour estimer la satisfaction par utilisateur. On calcule une moyenne locale des valeurs prédites par utilisateur, après on calcule la moyenne générale. Supposons qu'un SR prédit 10 notes dont 8 de bonne qualité. Il est possible que les 8 valeurs ne concernent qu'un seul utilisateur (qui note beaucoup), et les deux autres valeurs sont pour deux autres utilisateurs. Dans ce cas, ce système aura un bon score MAE alors qu'il ne satisfait qu'un utilisateur sur trois. UMAE sert à détecter la fluctuation du SR, et mesure sa capacité à satisfaire le maximum de ses utilisateurs.

$$UMAE = \frac{\sum_{j=1}^N \frac{\sum_{i=1}^N |p_{ij} - a_{ij}|}{N_j}}{N} \quad (1.5)$$

où N est le nombre total d'utilisateurs, N_j est le nombre des notes prédites pour l'utilisateur j .

- Erreur moyenne absolue pour les valeurs hautes (HMAE) : les versions examinées pour l'instant mesurent la capacité du système à prédire les valeurs. HMAE considère les erreurs de prédiction en fonction de leurs impacts sur la recommandation. Elle tolère les erreurs dans les notes basses, mais pas dans les notes élevées. L'équation de HMAE est identique à celle de MAE, sauf qu'elle ne s'applique que quand les notes réelles sont élevées (pour nous une note élevée est 4 ou 5 dans une échelle de notation entre 1 et 5).

Mesures de précision de classement

Ces mesures ne prennent pas en compte la valeur que le SR prédit, mais elles considèrent sa décision de recommander ou non un item. Le SR est récompensé pour les bonnes décisions (l'intégration des items pertinents dans sa liste de recommandations), et pénalisé pour les mauvaises (l'intégration des items non pertinents dans la liste, ou l'absence d'un item pertinent de la liste).

L'objectif est de mesurer la fréquence des bons et mauvais jugements portés par le système de recommandation à l'égard des items.

La F-mesure est la mesure la plus utilisée de cette catégorie [Herlocker *et al.*2002]. Elle a été proposée la première fois par Cleverdon en 1968 pour les systèmes de recherche d'information. Elle était appliquée la première fois pour les SR par [Basu *et al.*1998]. Elle se compose de deux valeurs : le rappel et la précision. Nous allons expliquer cette mesure avec l'aide du tableau 1.3.

Le rappel est le rapport entre le nombre d'items pertinents sélectionnés et le nombre total d'items pertinents. Formellement :

$$R = Nb_{ps}/Nb_p \quad (1.6)$$

La précision est le rapport entre le nombre d'items pertinents sélectionnés et le nombre total d'items sélectionnés. Formellement :

$$P = Nb_{ps}/Nb_s \quad (1.7)$$

La F-mesure est un compromis entre le rappel et la précision :

$$F = \frac{2 * R * P}{R + P} \quad (1.8)$$

Plusieurs autres approximations du rappel et de la précision existent [Herlocker *et al.*2004, Sarwar *et al.*2000]. Dans notre travail, nous présentons une approximation du rappel en considérant qu'il est le rapport entre le nombre de valeurs que le SR a prédites et le nombre de valeurs qu'on lui a demandé de prédire. Nous présentons aussi une approximation de la précision en considérant qu'elle est le complément à 1 de l'erreur moyenne absolue normalisée. Notre objectif est de profiter de la puissance de MAE pour mesurer la précision, et d'insérer la notion de couverture (représentée par le rappel) dans cette mesure.

Mesures de précision du classement

Ces mesures sont utiles dans le cas où le SR doit prédire l'ordre de la liste d'items recommandés. La pertinence d'un item n'est pas une valeur binaire mais conditionnée par sa position dans la liste de recommandations. Le système peut être pénalisé par ces mesures même si la liste ne contient que des items pertinents, si ces items sont mal ordonnés.

Dans cette catégorie, nous nous sommes intéressés à la Moyenne du rang réciproque (MRR). Elle est une mesure de qualité utilisée pour évaluer les SR qui doivent donner comme sortie une liste ordonnée avec un seul élément pertinent. Le rang réciproque (RR) d'une liste est égale à $1/r$, où r est le rang donné par l'approche évaluée à l'élément pertinent. La moyenne du rang réciproque est la valeur moyenne de RR de toutes les listes. La valeur de cette mesure varie entre 0 et 1, où 1 est le meilleur score de précision. Cette mesure est appropriée à certaines applications comme les systèmes de question-réponse, où le SR doit rendre la réponse la plus exacte à une question donnée.

Couverture

La couverture dans les systèmes de recommandation peut avoir deux sens : la couverture du catalogue d'items, et la couverture d'utilisateurs. Dans le premier cas (couverture d'items), on calcule le pourcentage d'items que l'approche est capable de recommander aux utilisateurs pour l'ensemble d'items disponibles [Lü *et al.*2012]. Dans le deuxième cas (couverture d'utilisateurs), on calcule le pourcentage d'utilisateurs pour lesquels le SR est capable de générer des recommandations. Nous nous intéressons seulement à la deuxième (couverture d'utilisateurs).

1.3 Résumé

Dans ce chapitre, nous avons présenté les systèmes de recommandation. Nous avons examiné et comparé plusieurs approches de recommandation. Nous avons ensuite analysé la dominance du

FC dans le domaine, et l'émergence des SRC dans le web social. Puis, nous avons présenté plusieurs facteurs importants pour le fonctionnement du SR, et comparé les approches disponibles en fonction de ces facteurs. Nous avons résumé cette étape dans notre tableau 1.2 de comparaison, en rappelant que les approches de recommandation souffrent en général d'un problème de fluctuation qui baisse la performance pour un sous-ensemble d'utilisateurs. Nous comptons sur les SR hybrides pour surmonter ce problème. Nous avons alors indiqué les techniques d'hybridation disponibles. Pour finir, nous avons présenté les méthodes d'évaluation de SR.

Le chapitre actuel de l'état de l'art est dédié aux SR globalement. Nous sommes intéressés spécifiquement par les SR à base de confiance. Dans le chapitre suivant, nous continuons notre état de l'art en évoquant le sujet de la confiance et ses applications dans les systèmes de recommandation.

La confiance computationnelle

Sommaire

2.1	Introduction	25
2.2	La confiance sociale	26
2.2.1	Les zones de confiance	27
2.2.2	Le contexte de la confiance	28
2.2.3	Les deux paradigmes de confiance	28
2.2.4	Mise à jour de la confiance	31
2.3	La modélisation de la confiance sociale	32
2.4	Tour d’horizon des algorithmes de confiance	33
2.4.1	Advogato	33
2.4.2	AppleSeed	34
2.4.3	EigenTrust	35
2.4.4	MoleTrust	37
2.4.5	TidalTrust	39
2.4.6	Le modèle d’Abdul-Rahman	40
2.4.7	Le modèle de Richardson	41
2.4.8	Le modèle d’O’Donovan	42
2.4.9	Bitreillis	43
2.4.10	La logique subjective et les réseaux de confiance	44
2.5	Résumé	49

2.1 Introduction

Cette thèse examine l’apport de la modélisation et de l’exploitation de la confiance aux SR. Ce chapitre est consacré aux approches de la modélisation de la confiance en informatique, et leur application dans le domaine des SR.

La confiance joue un rôle fondamental au sein de la société humaine depuis la nuit des temps, ce qui en fait un sujet captivant, étudié dans plusieurs disciplines de sciences humaines.

Notre démarche pour étudier la confiance computationnelle dans ce chapitre commence par une investigation sur la confiance sociale. Nous avons choisi d’expliquer le déroulement du scénario de confiance par un exemple de la vie réelle, en montrant plusieurs sous-scénarios possibles, et différents aspects et caractéristiques de la confiance sociale. Notre objectif dans cette partie est de dresser un panorama des paramètres et aspects fondamentaux de la confiance sociale.

Nous étudierons ensuite la confiance computationnelle. Nous listons plusieurs modèles qui tentent de modéliser la confiance pour des objectifs divers. Nous discutons de leurs qualités, inconvénients et leur intérêt au regard des attentes précédentes. Nous résumons cela dans un tableau qui compare les modèles, et qui explique notre choix de la logique subjective [Jøsang *et al.*2006] comme cadre des modèles que nous allons proposer plus tard dans la partie consacrée à notre contribution.

2.2 La confiance sociale

Le sujet de la confiance a attiré de nombreux chercheurs en sciences humaines. Pourtant, jusqu'à ce jour, aucun d'entre eux n'a proposé une définition faisant l'unanimité. Georg Simmel (1858-1918) fut l'un des premiers sociologues à écrire sur la confiance. Pour lui, elle est l'une des forces de socialisation la plus importante dans la société [Watier2008]. [luh2013] la considère comme un mécanisme pour réduire la complexité du monde social. Pour [Arrow1974,Deutsch1977, Gambetta1988], elle est ce qui fait réussir les efforts de coopération. Pour [Fox1974,Lewis and Weigert1985], elle est une clé positive dans les relations interpersonnelles. Pour [Gabarro1987], elle est un compostant central dans les relations de travail effectif. Pour [Yu *et al.*,Li *et al.*2003, Paolucci *et al.*2003], elle est l'équivalent de la délégation. Pour [Cofta2007], elle est un point central de toute décision qui peut comprendre du risque.

[Mcknight and Chervany1996] a montré qu'il y a une tendance dans les sciences humaines à considérer que la confiance a un effet positif, mais sans pouvoir en donner une définition unanime. [Castaldo2007] est allé plus loin et a collecté 72 définitions de la confiance venant de quatre disciplines (gestion, marketing, psychologie, sociologie). Puis il a extrait les termes les plus fréquents (avec les synonymes). Ses résultats sont :

No	Terme	f	No	Terme	f	No	Terme	f
1	sujet (individu)	180	11	Prendre / Accepter	11	21	Situation	6
2	Action, comportement	42	12	Risque	11	22	Attitude	5
3	Future auxiliaire	29	13	Vulnérabilité	11	23	Décision	5
4	Attente	24	14	Relation	10	24	Réalisation	5
5	Croyance	23	15	Échange	9	25	Maintenir	5
6	Résultat	19	16	À base de	8	26	Intention	5
7	Fiabilité	18	17	Compétence	7	27	Impliquer	5
8	Confiance "trust"	17	18	Positive	7	28	Mutuel	5
9	Confiance "Confidence"	16	19	Coopération	6	29	Mot	5
10	Volonté	14	20	Exploitation	6	30	Conditionnel	5

TABLE 2.1 – Fréquence des termes dans les définitions de la confiance

Bien que les sciences humaines ne fournissent pas une définition concrète de la confiance, ce tableau montre une synthèse des termes qui gravitent autour d'elle. Dans un système de recommandation, l'objectif final pour l'utilisateur est de trouver des items convenables. La confiance doit servir cet objectif. Par conséquent, nous proposons une définition de la confiance qui prend en compte les perspectives des SR, et qui s'inspire des avis de [Arrow1974,Deutsch1977, Gambetta1988, Jøsang1996] :

La confiance est la croyance d'un individu nommé confiant, qu'un autre nommé crédible (ou digne de confiance), a la compétence et la volonté de coopérer, pour accomplir une tâche en faveur du confiant. Ce dernier est souvent invité à choisir

le crédible parmi un groupe de candidats. Sa croyance est basée sur les précédentes interactions avec chacun des candidats, les conseils de ses amis fiables, les caractéristiques de la tâche demandée (domaine, contexte, urgence ...), ainsi que d'autres composantes subjectives (l'apparence du candidat, sa façon de s'exprimer ...). Cela a donné naissance à la confiance computationnelle, qui est censée représenter une adaptation de la confiance sociale au web.

Comparée au tableau 2.1, cette définition contient 7 des 30 termes du tableau, les termes sont situés entre la position 1 et 17 en terme de fréquence.

Notre présentation de la confiance s'illustre sur l'exemple que nous allons exposer. Cet exemple est construit autour de notre réflexion sur le problème de la confiance, et des travaux dans [Golbeck2009, Jøsang *et al.*2006, Victor *et al.*2009].

2.2.1 Les zones de confiance

Supposons qu'Arnaud possède une voiture de la marque x, et qu'elle soit tombée en panne. Il ne peut pas la réparer lui-même car il n'a aucune connaissance dans ce domaine. Arnaud a besoin de confier cette tâche à quelqu'un de "crédible".

Ainsi, le scénario de confiance commence par un besoin exprimé par un individu. L'objectif est d'accomplir une tâche qu'il ne peut pas faire lui-même pour des raisons diverses (dans l'exemple c'est le manque de compétence d'Arnaud en réparation de voitures). Le confiant (Arnaud) doit maintenant trouver une personne crédible pour exécuter la tâche. Pour ce faire, Arnaud pense à plusieurs candidats possibles. Il estime la confiance qu'il peut attribuer à chacun d'entre eux selon deux facteurs : son estimation des intentions du candidat à son égard, et son estimation de la compétence du candidat à propos de la tâche à effectuer.

Dans [Marsh and Briggs2009], les auteurs placent la confiance entre deux individus sur une barre en trois zones. Le confiant classe les candidats selon sa confiance en eux sur les zones de la barre. La figure 2.1 montre la barre de confiance : la confiance varie entre deux valeurs -1 (défiance complète) et $+1$ (confiance complète). La valeur 0 représente la neutralité. Le seuil de coopération est la valeur minimale de confiance pour laquelle le confiant accepte de coopérer avec le candidat.

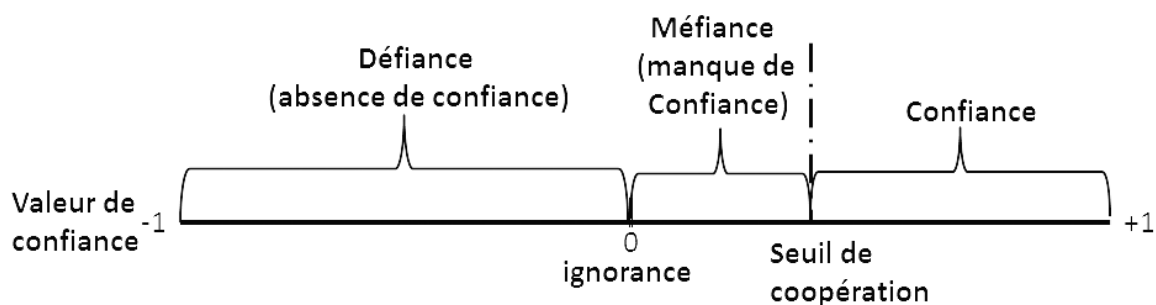


FIGURE 2.1 – Les zones de confiance

On peut identifier trois zones :

1. La défiance : la valeur est négative parce que le confiant pense que le candidat a des intentions négatives.
2. La méfiance : l'estimation du confiant est positive, les intentions du candidat ne présentent pas de danger pour lui, mais la valeur reste en-dessous du seuil de coopération. Cela parce

que le confiant juge le candidat incapable d'exécuter la tâche, même si ses intentions sont positives. Par conséquent, le confiant ne va pas coopérer avec ce candidat tant qu'il émet des doutes sur sa compétence.

3. La confiance : la valeur ici est au-dessus du seuil de coopération, alors le confiant prendra le risque de coopérer avec ce candidat.

Les zones de confiance numérisent l'opinion du confiant à l'égard des candidats. Le point le plus déterminant sur cette barre est le seuil de coopération : où se positionne-t-il exactement sur la barre? Quand et comment change-t-il sa position? Afin de mieux comprendre la notion du seuil de coopération, il faut prendre conscience du contexte de la confiance.

2.2.2 Le contexte de la confiance

La confiance devient plus centrale lors des périodes critiques et des crises [Mishra *et al.*1996, Weick and Roberts1993].

Si la panne arrivait à Arnaud sur l'autoroute, compte tenu de l'urgence de la situation, il serait poussé d'accepter l'aide du premier mécanicien susceptible de se présenter immédiatement, sans poser de question sur la qualité de son travail.

L'urgence de la tâche a une influence directe sur la décision de coopérer. Quand le confiant est dans l'urgence, il doit décider rapidement, il devient plus tolérant et moins exigeant, et par conséquent, il a intérêt à baisser le seuil de coopération.

Si la voiture est dans le garage d'Arnaud, et qu'il l'utilise très rarement, parce qu'il n'a pas un besoin fréquent d'une voiture, ou qu'il en possède une autre plus performante, tous ces éléments rendent l'impact de cette panne moins important sur la vie d'Arnaud, et le rendent moins motivé à la réparer (il peut même, dans ce cas, penser à s'en débarrasser définitivement).

L'importance de la tâche pour le confiant, et son impact sur sa vie jouent un rôle décisif dans le positionnement du seuil de coopération. Ce dernier augmente quand la tâche est importante et diminue si elle ne l'est pas.

L'ensemble des éléments précédents (le sujet de la tâche, l'urgence, l'importance) constitue le contexte. Le contexte encadre la relation de confiance, et influence le seuil de coopération. À notre connaissance, tous les modèles de confiance computationnelle d'aujourd'hui ne sont sensibles ni à l'urgence ni à l'importance de la tâche. Certains ne sont pas sensibles au sujet non plus. D'autres résumement le contexte dans le sujet de la tâche. Généralement, les modèles qui existent considèrent implicitement un seuil de coopération unique et égal à zéro pour tous les cas.

Le seuil de coopération est aussi influencé par d'autres facteurs en dehors de la confiance comme le prix, la distance géographique, mais ces facteurs ne sont pas pris en compte dans cette thèse.

Après avoir déterminé le seuil de coopération, le confiant doit placer les candidats sur la barre de confiance pour choisir celui qui est le plus crédible pour cette coopération. Dans la section suivante, nous allons aborder comment le confiant attribue les scores de confiance aux candidats afin de les placer sur la barre de confiance.

2.2.3 Les deux paradigmes de confiance

Après avoir déterminé le besoin d'Arnaud et son contexte, on passe à l'étape suivante qui consiste à trouver une personne digne de confiance pour effectuer la tâche. Arnaud connaît trois mécaniciens, Eric, Fabien et Georges.

Chaque modèle de confiance possède une fonction qui sert à calculer la confiance mutuelle entre chaque couple d'individus. Il existe deux grands paradigmes pour les modèles à base de confiance.

2.2.3.1 La confiance globale

Le premier paradigme : Arnaud pense à utiliser un magazine spécialisé qui publie le classement des mécaniciens selon une estimation de leurs compétences, ou par un sondage qui collecte les avis de leur clientèle.

Ce paradigme s'appelle la confiance globale, ou réputation, ou encore confiance quantitative. Dans cette catégorie de modèles, chaque utilisateur est caractérisé par une valeur de crédibilité ou de réputation. Cette valeur même est considérée à chaque fois que quelqu'un essaie de coopérer avec cet utilisateur. Dans notre exemple, le magazine donnera à Arnaud les scores de réputation des trois mécaniciens ainsi que d'autres.

2.2.3.2 La confiance locale

Dans le deuxième paradigme : Arnaud s'appuie sur sa propre connaissance, et sur celle de ses amis fiables, afin de construire sa propre opinion vis-à-vis de chacun des trois mécaniciens. Supposons qu'Arnaud considère :

- *Éric : du point de vue d'Arnaud, Éric est un bon professionnel. Il lui a rendu service à plusieurs reprises pour des pannes de voiture.*
- *Fabien : Arnaud avait changé une fois une pièce de sa voiture chez Fabien. La pièce n'était pas bien fixée. En plus, il a eu une facture importante. Arnaud trouve Fabien malhonnête, il ne veut absolument pas traiter avec lui. Arnaud retire Fabien de la liste des candidats.*
- *Georges : Arnaud ne connaît pas Georges en personne, et il ne sait rien sur la qualité de son travail. Il sait juste que près de chez lui, il y a le garage de Georges.*

Le confiant ici se sert de sa propre connaissance des candidats. Cette connaissance est le fruit de ses interactions avec chacun d'entre eux.

Arnaud peut se contenter de ses expériences personnelles pour décider. Il peut être aussi plus prudent et perfectionniste, alors il décide de consulter ses amis fiables (Bruno et Claire) à propos de Georges. Bruno pense que Georges est bon, il a déjà réparé sa voiture chez lui et cela s'est bien passé, alors que Claire trouve que Georges n'est pas ponctuel et donc pas trop fiable.

Le confiant peut compléter sa connaissance à propos des candidats qu'il ne connaît pas directement par les conseils des amis fiables. Le bouche-à-oreille est une manière efficace pour échanger et propager la connaissance. Quand le confiant ne dispose d'aucune connaissance sur le candidat, il peut se renseigner auprès de ses amis fiables. Les amis, à leur tour, peuvent se renseigner auprès de leurs amis et ainsi de suite.

Les modèles à base de confiance locale représentent l'ensemble des utilisateurs sous la forme d'un graphe orienté, comme illustré dans la figure 2.2. Les sommets ou les nœuds représentent les utilisateurs, et les arcs représentent les liens de confiance entre les utilisateurs. Les liens sont fondés sur les interactions directes entre chaque couple d'utilisateurs. Pour que l'utilisateur A sache s'il peut coopérer avec l'utilisateur E , il s'appuie sur la puissance du lien de confiance de A en E . Si ce lien n'existe pas, A demande à ses amis s'ils peuvent le conseiller à propos de E . Cette demande est propagée dans le graphe par A et ses amis jusqu'à trouver un chemin qui relie A avec E ; chaque nœud sur le chemin partage sa connaissance avec le nœud qui le précède, qui considère cette connaissance conditionnée par sa confiance au participant, et ainsi de suite jusqu'à

retourner chez *A*. Les modèles qui suivent ce paradigme de confiance locale doivent maîtriser la propagation de la confiance. Ils doivent aussi pouvoir traiter des informations contradictoires du fait qu'il se peut que les amis interrogés fournissent des avis divergents.

Les avis de Bruno et Claire peuvent influencer Arnaud et changer son choix, tant qu'il leur fait confiance, et qu'il considère qu'ils veulent lui apporter de l'aide même s'ils ne sont pas capables de résoudre son problème par eux-mêmes.

Dans ce scénario, Arnaud a demandé l'aide de ses amis à propos de Georges seulement (là où il n'a aucune information pour prendre une décision). Dans un autre scénario, il pouvait demander leurs avis à propos de la liste entière. Bruno alerte Arnaud du fait qu'Éric n'est pas compétent spécifiquement pour cette panne de voiture. Claire affirme que son ami Damien avait la même panne qu'Arnaud et qu'il l'a réparée chez un autre mécanicien nommé Henry.

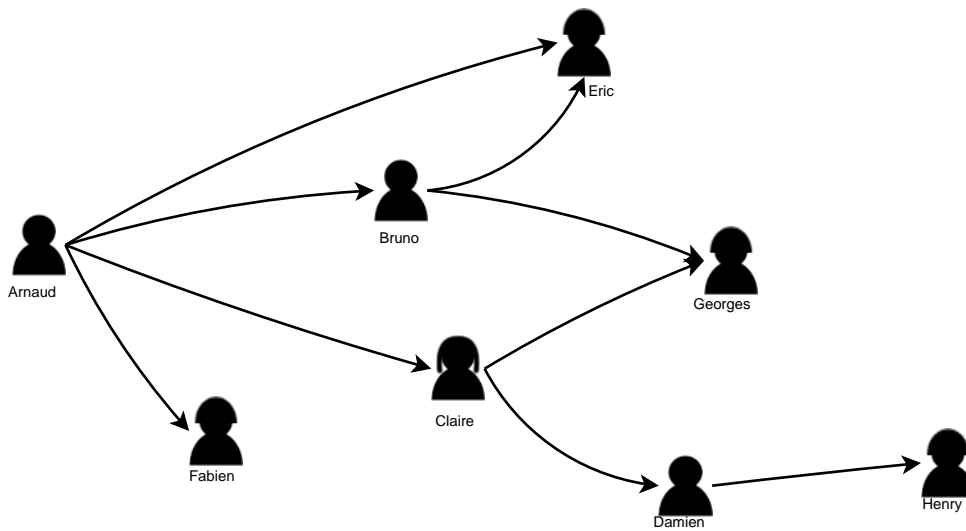


FIGURE 2.2 – Le réseau de confiance

Consulter les amis peut toujours apporter de nouvelles informations utiles. Un simple changement de stratégie de propagation peut affecter fortement le choix du confiant. Les amis peuvent affiner la connaissance du confiant à propos des candidats et à propos du contexte. La modélisation de confiance est très sensible à la stratégie de propagation. Dans cet exemple, les amis pouvaient dévoiler la faiblesse d'Éric pour cette tâche (en modifiant le sujet de la tâche de "réparer la voiture" qui est une demande vague et imprécise à "réparer la panne spécifique"). Ils pouvaient aussi proposer de nouveaux candidats "Henry" qu'Arnaud ne connaissait pas avant. La question qui se pose ici est, sachant qu'Arnaud fait confiance à Claire, cela garantit-il qu'il fera aussi confiance à ses amis (les amis de Claire) ? Si oui, à quel point ? La transitivité de la confiance est un grand défi pour les modèles de la confiance. Peu de modèles considèrent que la confiance n'est pas transitive [Christianson and Harbison, Abdul-Rahman and Hailes2000]. En revanche, l'écrasante majorité considère la confiance transitive, sans se mettre en accord sur la manière de la propager. Communément, les modèles considèrent qu'Arnaud accordera aux amis de Claire une confiance proportionnelle à celle qu'il lui accorde. Il est clair que le rôle des amis fiables est décisif, même quand ils ne peuvent pas eux-mêmes accomplir la tâche. Les liens de confiance ne s'utilisent donc pas toujours de la même manière.

En outre, ce scénario nous permet de distinguer quatre types de relation de confiance, ces types sont également discutés dans [Jøsang *et al.*2006] :

1. Confiance directe : la confiance est le résultat d'interactions directes exclusives entre le confiant et le crédible ; telles que les relations (Arnaud, Bruno) et (Arnaud, Éric).
2. Confiance indirecte : les deux personnes ne se connaissent pas directement. La confiance est établie par une ou des personnes intermédiaires ; par exemple, la relation (Arnaud, Georges) et (Arnaud, Damien).
3. Confiance fonctionnelle : l'attente du confiant est que le crédible accomplit la tâche lui-même ; par exemple, les relations (Arnaud, Éric), (Arnaud, Georges) et (Arnaud, Henry).
4. Confiance référentielle : l'attente du confiant est que le crédible lui recommande quelqu'un pour accomplir la tâche, par exemple (Arnaud, Bruno), (Claire, Damien) et (Arnaud, Damien). A noter que la recommandation de Claire est également basée sur sa confiance référentielle en Damien. En d'autres termes, il n'y a aucune obligation pour le crédible de la confiance référentielle à fonder sa recommandation sur une relation fonctionnelle. Toutefois, une série de relations de confiance référentielle doit se terminer par une relation de confiance fonctionnelle [Jøsang and Pope2005].

D'autres types de confiance ont été identifiés et discutés, dans [Herzig *et al.*2010]. On fait la distinction entre : la confiance immédiate, où l'utilisateur A fera confiance à B pour un besoin immédiat, et qu'il attend que B accomplisse la tâche à l'instant ; et la confiance potentielle⁴ où A pense qu'il doit faire confiance à B pour un éventuel futur besoin, et que B sera prêt à l'aider si ce besoin se déclençait.

2.2.4 Mise à jour de la confiance

Quel que soit le choix d'Arnaud, il va évaluer les conséquences de la coopération. Si la coopération se passe bien, la confiance d'Arnaud en la personne crédible va légèrement augmenter. Dans le cas contraire, la confiance va fortement baisser. La confiance s'édifie progressivement et se perd facilement. La confiance perdue est difficile à regagner, mais cela n'est pas impossible. Rétablir la confiance peut être fait par le *regret* ; si le crédible arrive à expliquer les raisons de l'échec au confiant, il peut diminuer l'impact de l'échec sur le lien de confiance. Il est aussi possible de regagner la confiance grâce à l'intervention d'un ami commun qui tente de convaincre le confiant de donner une deuxième chance au crédible. Cette notion de regret est très importante dans un contexte de confiance mais elle est rarement considérée ou étudiée.

Pour résumer, un scénario de confiance suit les étapes suivantes :

1. Déclenchement d'un besoin de coopération : cela peut être annoncé directement par l'utilisateur, ou bien estimé par quelqu'un d'autre (l'exemple des clés qui tombent, l'utilisateur ne se rend pas compte de son besoin de récupérer ses clés), c'est exactement comme les systèmes de recommandation qui anticipent le besoin d'un utilisateur pour un item donné.
2. Précision du contexte : identification du type de besoin, importance et urgence.
3. Trouver les candidats potentiels (dépendants de l'utilisateur seul, ou de lui et de ses amis recommandateurs, ou bien du système).
4. Calculer la confiance accordée à chaque membre de la liste en utilisant un sous-ensemble de trois méthodes : la réputation de chacun, les interactions précédentes avec le confiant, la propagation de bouche-à-oreille.
5. Ordonner la liste en fonction des valeurs de confiance.
6. Coopérer avec le ou les meilleurs candidats.

4. en anglais : dispositional

7. Évaluer la coopération en modifiant les liens de confiance ou les valeurs de réputation des utilisateurs impliqués dans la coopération.

2.3 La modélisation de la confiance sociale

*Facebook me demande ce que je pense, Twitter me demande ce que je fais, Foursquare me demande où je suis. L'Internet est transformé en une petite amie obsédée.*⁵

Le web social a livré un environnement de partage universel. Un environnement facilement accessible et qui dépasse les limites géographiques. Un environnement où les communautés se forment autour de thèmes, d'intérêts communs, ou d'objectifs partagés.

Aujourd'hui, d'innombrables applications intègrent cet aspect social. Cela comprend (sans être limité à) :

- Encyclopédie collaborative : Wikipedia⁶ est la première encyclopédie collaborative et pourtant la plus grande dans l'histoire de l'humanité à ce jour. Elle a démarré en 2001, et il lui a fallu seulement 3 ans pour dépasser, en terme d'utilisation, les géants de son domaine comme Encarta l'encyclopédie de Microsoft, et Britanica qui régnait sur le domaine depuis près de 200 ans. Cela montre la force redoutable des mobilisations sociales.
- Réseau de partage de vidéos : Youtube, dailymotion ou keek offrent un espace de partage des vidéos permettant de les évaluer, et de les commenter.
- Réseaux sociaux : avec Facebook, twitter et "google+", les gens peuvent presque partager leur vie en temps réel avec leurs amis, voire avec tout le monde sur Internet. LinkedIn offre aussi un réseau social dédié aux entreprises professionnelles et instituts de recherche.
- Blogs : Blogspot et tumblr offrent à leurs utilisateurs la possibilité d'avoir un blog (journal personnel) qui sert à partager leurs avis et connaissances et recevoir les commentaires du public. Cela permet l'apparition d'une "activité journalistique" parallèle dont les héros sont les nouveaux "bloggers" qui peuvent se faire entendre par le grand public sans passer par des maisons d'édition ou des journaux.
- Réseaux de partage de photos : instagram et flickr offrent des espaces pour partager les photos, les évaluer, et les commenter.
- Sites moins connus dédiés à certaines communautés ou certains objectifs, comme le site Epinions qui permet à ses utilisateurs de discuter de la qualité des produits qu'ils consomment.
- Systèmes de questions-réponses : comme experts-exchange et stackoverflow, où des développeurs informatiques peuvent s'aider les uns les autres pour résoudre des problèmes techniques.
- Apprentissage social de masse : les cours en ligne ouverts et collectifs (Moocs) sont apparus dans l'éducation. Ils permettent une participation massive des élèves. De nombreuses plateformes, sites universitaires et hors-université offrent aujourd'hui des Moocs. Cela comprend des grandes universités comme Harvard et Stanford.

Il est devenu clair que le web social est entré dans notre vie, et est devenu aujourd'hui important et capable d'influencer nos décisions quasiment à tous les niveaux : de l'achat d'un simple produit à quelques euros, jusqu'à l'achat d'une voiture ou une maison à plusieurs dizaines de milliers d'euros. On devient de plus en plus dépendant du web pour faire des achats, pour recevoir des informations, pour enrichir notre culture, pour se faire des amis, et pour se faire entendre. Comme d'autres sociétés coopératives, il devient très important d'enrichir le web social par la notion de confiance.

5. L'acteur américain Tracy Morgan

6. <http://www.wikipedia.com>

La modélisation de la confiance pour le web social a commencé à la fin des années 90. Plusieurs modèles ont été proposés pour être utilisés dans des applications diverses. Cela comprend : les systèmes de sécurité et de cryptographie [Gong *et al.*1990], les réseaux pair-à-pair pour le partage des fichiers [Kamvar *et al.*2003, Tang *et al.*2005], les systèmes de recommandation, les réseaux sociaux, le web sémantique [Artz and Gil2007, Nejdil *et al.*2004, Golbeck *et al.*2003], les systèmes multi-agents [Montaner *et al.*2002, Mui2002, Herzig *et al.*2010], les systèmes distribués [Jøsang1996], le cloud computing [Li and Ping2009], les systèmes de vente en ligne [Zhang *et al.*2008, Wang and Gui2013, Resnick and Zeckhauser2002], et bien d'autres domaines.

Les modèles de confiance computationnelle se démarquent selon le type de confiance considérée (locale, globale), leurs échelles de valeurs de confiance (binaire, graduelle, probabiliste), leur prise en compte ou pas de la défiance, leur stratégie pour propager la confiance.

Les modèles à base de confiance locale souffrent du problème de démarrage à froid, ils ont du mal à agir avec les nouveaux utilisateurs qui n'ont encore aucune interaction avec les autres membres du système. C'est exactement le même problème que le filtrage collaboratif, sauf que les systèmes de confiance locale agissent plus vite que le filtrage collaboratif et avec beaucoup moins de notes (souvent le FC demande à l'utilisateur de noter une quinzaine d'items, alors que les systèmes de confiance locale peuvent souvent démarrer dès le premier lien d'amitié que l'utilisateur exprime). Les systèmes à base de confiance globale n'ont pas ce problème, ils détectent les utilisateurs les plus réputés, et leur associent les nouveaux utilisateurs. Ainsi, dès que l'utilisateur s'inscrit dans le système, il est déjà relié aux personnes auxquelles il peut faire confiance.

La section suivante est dédiée à un tour d'horizon des modèles de confiance.

2.4 Tour d'horizon des algorithmes de confiance

L'objectif d'un modèle de confiance est d'aider le confiant en lui trouvant un bon crédible. Pour atteindre cet objectif, l'algorithme doit pouvoir gérer la propagation de la confiance, l'agrégation des avis, et il doit considérer le contexte ou le sujet de la tâche afin d'optimiser le choix du crédible.

Dans la section suivante, nous allons examiner plusieurs modèles de confiance, et nous concluons ce chapitre par un tableau comparatif qui montre les qualités et les défauts de chacun des modèles.

2.4.1 Advogato

C'est l'un des premiers modèles de confiance qui existe depuis 1999 [Levien2002]. L'application consiste en un réseau d'échanges et de partage pour les développeurs de logiciels libres. Le but est d'utiliser la méthode d'évaluation par les pairs, où les utilisateurs s'évaluent entre eux afin d'éviter les attaques malveillantes sur le site [Levien2004].

Le réseau des membres est présenté sous forme d'un graphe avec les nœuds représentant les membres, et les liens entre eux représentant la confiance entre les membres.

Les utilisateurs du site se certifient entre eux, en attribuant trois niveaux possibles (apprenti, aventurier, maître).

L'idée est similaire à celle de l'algorithme PageRank [Brin and Page1998, Page *et al.*1999] utilisé par Google pour classer les pages web. PageRank considère qu'un hyperlien vers une page est une reconnaissance de son importance. Plus la page est référencée par les autres pages, mieux elle est classée. L'ensemble des pages est visualisé sous forme de graphe, où les pages représentent les nœuds et les hyperliens les liens. On peut considérer cela comme un indice de la réputation

d'une page web. Par conséquent, Google l'utilise comme un indice (parmi d'autres) pour ordonner ses résultats.

La mesure de confiance d'Advogato réagit de la même manière que PageRank, mais elle remplace les pages web par les utilisateurs, et les hyperliens par des liens de certification.

L'algorithme s'appuie sur l'existence d'un petit groupe de membres crédibles appelés grains ou sources⁷; l'objectif de l'algorithme est de diviser les nœuds du graphe en deux groupes (bons nœuds, mauvais nœuds) au regard d'un grain donné. La réputation d'un nœud n'est pas liée seulement au nombre de personnes ayant confiance en lui, mais aussi à sa position par rapport au grain (plus il est proche, meilleure est la réputation). Levien appelle ce principe le "goulot d'étranglement", qui constitue une parade très efficace contre les attaques malveillantes.

L'algorithme prend en entrée un nombre entier n (la capacité du grain) qui doit donner un indice du nombre de membres crédibles à la fin de la procédure.

Même si un groupe de malveillants s'allie pour se certifier entre eux et augmenter leur réputation, cet acte ne donnera rien tant qu'ils ne recevront pas des certifications des membres fiables.

La sortie de l'Advogato est une fonction qui attribue à chaque membre une valeur binaire indiquant s'il est crédible ou pas :

$$trust_M : 2^V \times \mathbb{N}_0^+ \rightarrow (V \rightarrow \{true, false\})$$

où V est l'ensemble des utilisateurs.

La capacité d'un nœud donné dans le graphe est égale à la capacité du niveau précédent l divisée par le nombre de liens sortant de l .

Analyse du modèle

Le modèle adopte une stratégie de confiance globale ou réputation. Ce genre de modèles est normalement fragile face aux attaques de groupe. Mais cette notion de grain rend le modèle résistant à ce genre d'attaques.

Les notations des membres sont binaires, l'algorithme ne considère pas la confiance pondérée (le nœud est soit parfaitement fiable, soit parfaitement non fiable).

Il n'est pas toujours facile de trouver le groupe de grains. Dans le cas où un nœud reçoit une capacité inférieure au nombre des utilisateurs qu'il a certifiés, il passera cette capacité à un sous-ensemble de ses certifiés tiré au hasard sans aucune justification.

Aucune considération de contexte n'est prévue. La réputation du membre est une valeur absolue, valable pour tous les contextes et les sujets.

2.4.2 AppleSeed

Ce modèle est proposé par [Ziegler and Lausen2004]. C'est un modèle de confiance locale, il permet aux utilisateurs de se noter entre eux. Quelle que soit l'échelle de notation, les valeurs de notes seront normalisées. Ce modèle adopte une stratégie de confiance locale, l'ensemble des utilisateurs est représenté sous forme d'un graphe orienté.

Pour estimer la confiance entre un utilisateur source et un utilisateur cible, AppleSeed active le nœud source en y injectant une somme d'énergie. L'énergie coule dans le graphe comme un courant électrique, chaque nœud partage l'énergie qu'il reçoit avec les nœuds suivants. La quantité d'énergie passée à chaque nœud est proportionnelle à la confiance qui lui est faite par le nœud passeur. Elle est calculée selon la formule suivante :

7. Seeds en anglais

$$e_{A \rightarrow B} = d \cdot in_A \frac{tr_{AB}}{\sum_i tr_{Ai}}$$

Où $e_{A \rightarrow B}$ est la quantité d'énergie à passer du nœud A au nœud B .

d correspond à un facteur de diffusion (normalement $d = 0,85$).

in_A est l'afflux d'énergie à A .

tr_{AB} est la valeur de confiance que A accorde à B .

$\sum_i tr_{Ai}$ est la somme des valeurs de confiance que A accordait à tous ses amis crédibles.

AppleSeed considère la coopération entre A et B possible seulement si l'énergie arrivée à B dépasse un seuil pré-établi.

Analyse du modèle

Le modèle utilise le même principe que celui qui est utilisé par Advogato (le goulot d'étranglement) contre les attaques malveillantes ; la résistance de l'algorithme est prouvée mais seulement au plan théorique. AppleSeed utilise la même base théorique, mais il n'est pas testé sur des données réelles. Tant que le système contrôle la quantité d'énergie à diffuser au départ, ainsi que le facteur de diffusion, il contrôle le niveau de tolérance durant l'attribution de bons et de mauvais nœuds.

Le problème essentiel de ce modèle réside dans le fait qu'il considère la confiance comme une ressource limitée par utilisateur. Il pénalise donc les utilisateurs qui font confiance à beaucoup d'utilisateurs, ce qui est incohérent avec la définition de confiance et avec l'intérêt du système (qui veut que ces utilisateurs interagissent plus entre eux).

Contrairement à Advogato, AppleSeed est déterministe, il garantit toujours une répartition équitable d'énergie entre un confiant et ses amis crédibles, quelle que soit la quantité d'énergie. A cet égard, il se comporte mieux qu'Advogato.

Ce modèle est plus flexible qu'Advogato, car tout utilisateur peut être un grain pour lui-même. Advogato est très strict en ce qui concerne ses décisions sur la fiabilité des utilisateurs, car il considère que le nœud peut être strictement fiable ou non-fiable, alors qu'AppleSeed rend une liste ordonnée selon la fiabilité des utilisateurs, la liste est plus maniable et plus facile à utiliser selon l'application par rapport au jugement binaire.

2.4.3 EigenTrust

Ce modèle est développé pour protéger un réseau pair-à-pair d'échanges de fichiers contre les comportements malveillants [Kamvar *et al.*2003]. Les utilisateurs de ce genre de réseaux partagent des fichiers avec le public. Les utilisateurs peuvent aussi télécharger des fichiers depuis le réseau.

Un comportement aberrant consiste à remplacer le contenu d'un fichier recherché par des contenus publicitaires ou par un virus, tout en gardant le nom et les descriptions du fichier original.

Afin de réduire l'impact de ce comportement, les utilisateurs sont invités à informer le système après chaque téléchargement, si le fichier est le bon (+1) ou non (-1).

Ce modèle s'appuie fortement sur la transitivité de la confiance. Il construit une matrice T de taille $N \times N$ où N est le nombre des utilisateurs du système. L'élément t_{ij} représente la confiance que l'utilisateur i accorde à l'utilisateur j . Suite à chaque interaction, i note si l'interaction a réussi $tr_{ij} = +1$ ou non $tr_{ij} = -1$. La confiance de i en j correspond à la somme des notes que i attribuait à j suite aux téléchargements qu'il a effectués de chez lui $s_{ij} = \sum tr_{ij}$. Cette valeur est nommée confiance locale.

Le modèle prévoit une attaque de groupe où des utilisateurs malveillants s'arrangent pour s'attribuer des notes positives entre eux, et attribuer des notes négatives aux autres utilisateurs en dehors du groupe. Afin de réduire l'impact de ce comportement, les valeurs de confiance locale sont normalisées.

$$c_{ij} = \frac{\max(s_{ij}, 0)}{\sum_j \max(s_{ij}, 0)}$$

Les valeurs vides de matrice sont remplies grâce à la transitivité ; supposons que l'utilisateur i n'a jamais téléchargé des fichiers de chez k , EigenTrust calcule la confiance locale de i en k en s'appuyant sur les amis de i qui connaissent k :

$$c_{ik} = \sum_j c_{ij} \cdot c_{jk}$$

C correspond à la matrice des confiances locales normalisées. c_i est le vecteur de confiance de l'utilisateur i qui comprend la confiance de i en ses amis directs. On peut alors calculer la confiance de i aux amis de ses amis (vecteur de confiance de deuxième degré) :

$$\bar{t}_i = C^T \cdot c_i$$

Ainsi nous pouvons calculer le vecteur de degré x :

$$\bar{t}_i^x = (C^T)^x \cdot c_i$$

Pour de grandes valeurs de x , les vecteurs \bar{t} convergent vers une valeur propre de C . Les valeurs de confiance dans la matrice deviennent des valeurs de confiance globale (réputation) pour les membres, ce qui est le but du modèle.

Analyse du modèle

Il est vrai que, grâce à la normalisation, le modèle permet d'éviter les attaques de groupes. Mais d'un autre côté, à cause de la phase de normalisation, le modèle pénalise les utilisateurs qui font confiance à beaucoup d'utilisateurs ; il considère que la confiance est une ressource limitée par utilisateur, ce qui suscite un conflit avec le concept de confiance. Dans un autre contexte, cela pourrait être expliqué par le fait qu'un utilisateur de nature réservée ne fait pas confiance facilement, et quand il fait confiance, c'est pour une bonne raison, contrairement à quelqu'un de nature confiante qui fait confiance plus facilement, et dont les notes ont une valeur moins importante que le premier.

Les valeurs sont relatives ; si $c_{ij} = c_{ik}$, on sait que j et k possèdent la même réputation du point de vue de i , mais on ne sait pas s'ils ont une bonne réputation ou l'inverse.

Le modèle n'est pas capable d'expliquer ces valeurs de confiance. Supposons que l'utilisateur i a strictement deux bonnes interactions avec l'utilisateur j , et qu'il a avec l'utilisateur k cinq bonnes interactions et trois mauvaises, j et k auraient la même valeur de confiance locale, alors qu'il est clair que j est nettement plus crédible que k .

Le modèle prévoit une solution pour le démarrage à froid ; les nouveaux utilisateurs et les utilisateurs inactifs (ceux qui n'ont pas de liens de confiance) peuvent toujours compter sur une communauté pré-crédible (un groupe d'utilisateurs les plus réputés) pour leur calcul de confiance.

Le fait que tous les utilisateurs dans le système soient obligés de faire confiance à la communauté pré-crédible, permet d'affaiblir les groupes malveillants. Le modèle se défend ainsi contre ce genre d'attaque.

Le modèle considère que lorsque i accorde une valeur négative de confiance à j , i va faire le contraire de ce que j lui conseille de faire, ce qui n'est pas toujours le cas dans la vie réelle (il est possible que i va simplement négliger l'avis de j).

Le modèle est censé fonctionner dans des systèmes distribués.

Le modèle ne prend pas en considération le contexte de confiance.

2.4.4 MoleTrust

Ce modèle est développé par [Massa and Avesani2004] dans un contexte de recommandation. Il était utilisé pour faire de la recommandation à base de confiance sur les données du site Epinions⁸. Le site représente un réseau social dédié aux consommateurs de produits commerciaux, en leur offrant un espace pour exprimer et partager leurs avis et connaissances à ce propos. L'utilisateur d'Epinions peut noter des produits qu'il a achetés ou consommés avec une échelle numérique de 1 (le produit est peu appréciable) à 5 (le produit est trop appréciable). Il peut aussi rédiger un commentaire au sujet du produit pour développer son avis. Il peut noter les commentaires des autres utilisateurs sur la même échelle (1 à 5). Finalement, il peut exprimer sa confiance et sa défiance envers d'autres utilisateurs. Les valeurs de confiance et de défiance sont absolues. L'utilisateur peut exprimer sa confiance à un autre en l'ajoutant à son domaine, parce qu'il trouve que cet utilisateur fournit des avis pertinents, et qu'il veut que le système mette les avis de cette personne en avant. Dans le cas contraire (défiance), l'utilisateur l'ajoute sur sa "liste noire". Par conséquent, le système ne lui affiche plus les avis de cette personne.

La confiance, dans ce contexte, est une valeur binaire et non-contextuelle. MoleTrust est un modèle de confiance locale qui prend en considération les tendances personnelles des utilisateurs. Quand un utilisateur A ajoute un utilisateur B à son domaine, le modèle considère que B est légitime pour recommander des produits à A . Si B à son tour ajoute C à son domaine, le modèle considère que A fait aussi confiance à C , mais ne lui donne pas autant de confiance qu'à B . La confiance de A en C est calculée selon l'équation 2.1 :

$$tr(A, C) = \begin{cases} \frac{(d-n+1)}{d} & \text{if } n \leq d \\ 0 & \text{if } n > d \end{cases} \quad (2.1)$$

Le paramètre n correspond à la distance entre A et C ($n = 2$ dans ce cas parce qu'il y a un seul ami entre les deux, cela correspond à deux étapes, de A à B , et de B à C).

Le paramètre d est la distance maximale de propagation (dans la version de [Massa and Avesani2004], il est fixé à 4).

En cas d'existence de plusieurs chemins entre A et C , l'algorithme choisit celui qui est le plus court, donc celui qui maximise la confiance. La formule 2.1 est appliquée sous réserve que C n'est pas dans la liste noire de A , car le modèle ne propage pas la défiance.

2.4.4.1 Caractéristiques du modèle

L'absence de contexte dans ce modèle fait qu'il ne tient pas compte des domaines de compétence des utilisateurs. Supposons que A fasse confiance à B parce qu'il trouve que B fournit des avis pertinents à propos de produits d'électroménager, et que B fasse confiance à C pour la qualité de ses avis à propos des films cinématographiques, il n'est pas évident que A accepterait les recommandations de C pour les téléphones portables.

8. <http://www.epinions.com>

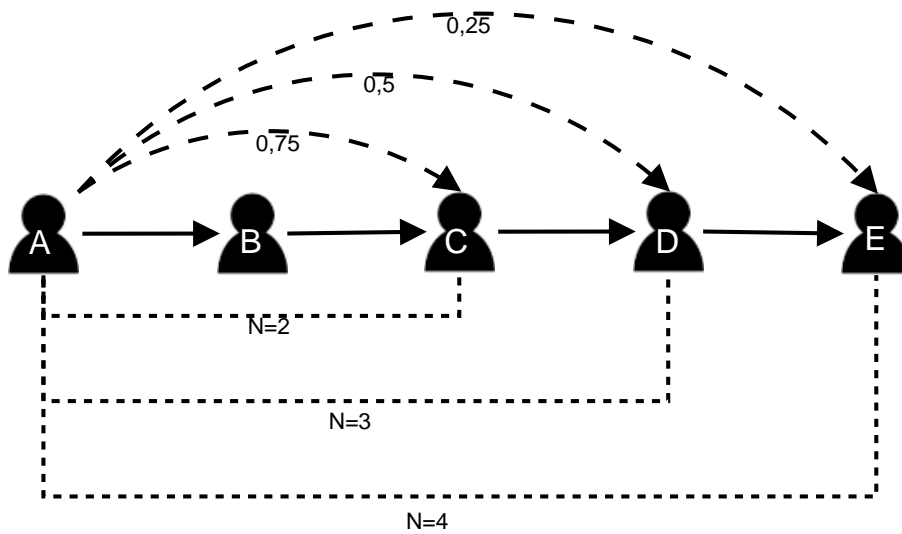


FIGURE 2.3 – La transitivité de confiance selon Moletrust avec $d=4$

L'expression de la confiance est limitée à une valeur binaire, la seule mesure qui compte est la distance au nœud source ; pour un utilisateur A il n'y a pas de moyen pour exprimer le fait qu'il fait confiance à B plus qu'à F (figure 2.4).

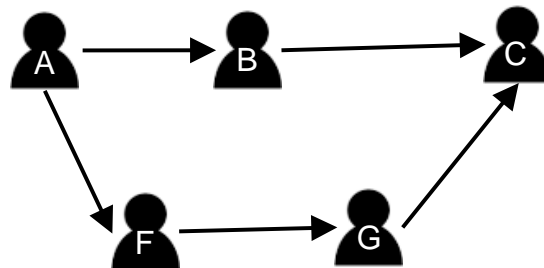


FIGURE 2.4 – MoleTrust avec plusieurs chemins

Le choix du chemin le plus court ne garantit pas toujours la meilleure performance. Dans la figure 2.4, MoleTrust choisit le chemin (A, B, C) pour lier A à C , et néglige complètement les informations en provenance de F et G . Ce modèle ne permet pas de savoir si B connaît C bien ou peu. Cela est important pour savoir à quel point B est sûr de ses propos quand il confirme que C est crédible. Fondamentalement, il est possible que G ait une connaissance plus profonde sur C que la connaissance de B sur C . Dans ce cas, en éliminant G , on perd beaucoup d'informations importantes sur C , ces informations pourraient être utiles pour A .

MoleTrust ne conçoit aucun traitement pour les relations de défiance, mais il cherche à maximiser la confiance par la stratégie du chemin le plus court.

Le modèle souffre du problème du démarrage à froid. Mais son impact reste moins important qu'avec le filtrage collaboratif.

Dans tous les cas, MoleTrust fait preuve d'efficacité dans le contexte de la recommandation par rapport au filtrage collaboratif [Massa and Bhattacharjee2004]. Sa performance surpasse celle du FC surtout en termes de précision, même si l'application actuelle (Epinions) permet à ses utilisateurs de s'exprimer par deux moyens différents, en notant des items ou des utilisateurs. Il se trouve qu'une partie des utilisateurs focalise sur un moyen en négligeant l'autre ; ce qui fait

qu'une portion considérable de la population ne fait que noter des items, et est donc privée des recommandations générées par ce modèle.

2.4.5 TidalTrust

Ce modèle est développé pour les réseaux sociaux [Golbeck2005]. Il a été appliqué et testé sur le site filmtrust [Golbeck and Hendler2006]. Il s'agit d'un réseau social dédié à la recommandation de films, où l'utilisateur peut, à la fois, noter des films sur une échelle de 5, rédiger des articles pour exprimer son avis détaillé à propos des films, créer des liens d'amitié avec les autres utilisateurs selon son accord avec leurs avis, et les noter en fonction de sa confiance en eux sur une échelle de 1 à 10. Le modèle adopte la notion de confiance locale. L'ensemble des utilisateurs est donc représenté sous forme d'un graphe orienté.

Dans la terminologie du modèle, la profondeur correspond à la distance entre le nœud actuel dans le chemin et le nœud source A . La force d'un chemin correspond à la valeur minimale de confiance entre deux utilisateurs le long du chemin. Comme MoleTrust, ce modèle choisit toujours le chemin le moins profond (le plus court) entre le nœud source (A), et le nœud cible (G). En cas d'existence de plusieurs chemins de même profondeur, l'algorithme choisit celui qui maximise la force.

Pour prédire si un utilisateur A apprécie, ou pas, un item x , tidalTrust extrait Q , la liste des utilisateurs qui ont déjà noté x . Il choisit parmi cette liste les utilisateurs que A avait déjà notés. La note de A sera calculée selon l'équation suivante :

$$v_{A,x} = \frac{\sum_{u \in Q} t_{Au} \cdot v_{ux}}{\sum_{u \in Q} t_{Au}} \quad (2.2)$$

où $v_{A,x}$ est la note attribuée par l'utilisateur A à l'item x .

u est un membre de Q qui a noté x .

$t_{A,u}$ est la confiance de A en u .

Supposons que la confiance de A en B est 9 et la confiance de A en C est 4, et que les notes de B et C pour x sont 4 et 2. L'algorithme prédit la note de A comme étant la moyenne des notes de ses amis pondérées par sa confiance en eux :

$$v_{A,x} = \frac{9 \times 5 + 4 \times 2}{9 + 4} = 4,08$$

Dans le cas où aucun membre de Q n'est ami avec A , tidalTrust va chercher les membres de Q dans les amis des amis de A , et ainsi de suite. La recherche s'arrête alors dès qu'on trouve au moins un membre de Q . Ceci signifie que seulement les membres les plus proches de A et qui minimisent la profondeur du chemin sont impliqués dans le calcul. Ensuite, en cas d'existence de liens avec plusieurs membres de Q , tidalTrust applique l'équation 2.2.

Analyse du modèle

Ce modèle utilise strictement des relations de confiance, en permettant à l'utilisateur d'exprimer une confiance graduelle. Contrairement à MoleTrust, il offre à l'utilisateur la possibilité de juger un ami comme étant plus crédible qu'un autre. (voir Figure 2.5).

La stratégie du chemin le plus court n'est pas toujours efficace. Revenons à l'exemple précédent ; si B n'avait pas noté x , la note de A aurait été calculée en se basant seulement sur la note de C en sachant que A ne lui fait pas trop confiance (4 sur 10). Supposons dans le même temps que D a noté x , et que B a attribué une note de confiance à D égale à 10, voir (figure 2.5). Le

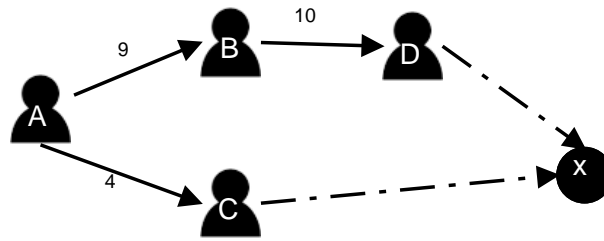


FIGURE 2.5 – tidalTrust

fait que D se situe un pas plus loin de A par rapport à C n'empêche pas que A peut être plus influencé par D que par C , surtout avec des notes plus élevées sur le chemin entre A et D , par rapport à la note attribuée à C .

2.4.6 Le modèle d'Abdul-Rahman

Ce modèle est proposé par [Abdul-Rahman and Hailes2000]. C'est un modèle de confiance locale. Bien qu'il soit proposé parmi les premiers, il fait partie des modèles qui prennent en compte le contexte. Il permet à ses utilisateurs d'exprimer à la fois leur confiance et défiance l'un à l'égard de l'autre. La notation prend la forme d'une échelle de quatre valeurs possibles (très bien, bien, mauvais, très mauvais).

Dans ce modèle, la confiance est basée sur les interactions passées. Les utilisateurs peuvent identifier leurs expériences avec les autres ainsi que les contextes. Le contexte est réduit au sujet de recherche de l'utilisateur actif. Par exemple, s'il cherche un livre de science fiction, il prend en compte seulement les notes attribuées à ce genre de livres et les relations de confiance fondées sur la recommandation des livres de science fiction.

Les utilisateurs sont capables d'échanger des informations concernant les réputations des autres, afin de soutenir les décisions de confiance locale. Pour Abdul-Rahman, la confiance est subjective, donc les utilisateurs peuvent avoir des avis contradictoires à propos d'un utilisateur. Les valeurs de confiance sont inférées. Elles s'appuient sur plusieurs interactions entre un couple d'utilisateurs, et sur leurs évaluations après chaque nouvelle interaction.

Selon ce modèle, la confiance entre deux utilisateurs se construit par les interactions entre eux, et seulement par ces interactions parce que le modèle ne considère pas que la confiance soit transitive, ainsi il n'y a pas de propagation de confiance. Le modèle reconnaît deux types d'interactions possibles : l'évaluation d'une expérience entre les deux utilisateurs, et l'acceptation d'une recommandation en provenance d'un utilisateur. C'est l'un des rares modèles qui sépare la confiance fonctionnelle de la confiance référentielle.

Chaque utilisateur A dispose de deux ensembles : Q où il stocke les résultats de ses interactions, et R où il stocke ses avis à propos des utilisateurs qui lui fournissent des recommandations. Chaque élément de Q ou de R se compose d'un triplet :

- L'utilisateur concerné : l'utilisateur auquel A a fait confiance dans le cas de Q , ou l'utilisateur qui a recommandé un item à A dans le cas de R .
- Le contexte : le modèle dispose d'une liste de domaines d'intérêt ; chaque domaine est considéré comme un contexte indépendant. Le contexte est résumé au sujet de la tâche.
- S est l'ensemble des avis de A , un avis peut avoir quatre valeurs (très mauvais, mauvais, bien, très bien). Chaque élément $s \in S$ représente un avis de A sur un utilisateur B dans un contexte c . L'avis s est constitué par un vecteur de quatre compteurs correspondant aux quatre valeurs de l'avis. Lorsque A exprime un avis sur une interaction avec B dans

le contexte c , le compteur correspondant à la valeur est mis à jour.

$$S_A = \{(s_{tm}, s_m, s_b, s_{tb})_{B_{c_0}}, (s_{tm}, s_m, s_b, s_{tb})_{B_{c_1}}\}$$

où s_{tm} correspond au compteur de "très mauvais", s_m au compteur de "mauvais" et ainsi de suite.

Pour l'évaluation de la confiance de A en B dans un contexte c_i , on prend la valeur de Q ou de R (cela dépend si on évalue la confiance fonctionnelle ou bien référentielle), et on choisit l'accumulateur maximal parmi les quatre disponibles. En cas d'existence de deux valeurs identiques, la valeur finale de confiance est incertaine et est donnée dans le tableau suivant :

valeur	sens
très bien \wedge bien \wedge ?	tendance positive (plutôt digne de confiance)
très mauvais \wedge mauvais \wedge ?	tendance négative (plutôt indigne de confiance)
autrement	tendance neutre

TABLE 2.2 – La confiance incertaine

Analyse du modèle

Le modèle ne dispose pas de moyens pour propager la confiance. La propagation de confiance est importante pour la modélisation de la confiance. L'auteur présume que la confiance n'est pas transitive, par conséquent, il n'applique pas de fonction de propagation. Il est vrai qu'il n'y a aucune preuve ni pour la transitivité de la confiance ni pour son intransitivité, mais dans des modèles qui prennent en considération le contexte, il est plutôt intuitif de considérer que les conseils des amis crédibles ont un effet positif sur la performance. La confiance à la base est un esprit coopératif, même dans le modèle actuel où on accepte la recommandation des utilisateurs tant que son objet porte sur des items (le groupe R contient les amis qui recommandent des produits appréciables), mais le modèle ne considère pas la propagation de confiance, alors il ne supporte pas que l'utilisateur recommande d'autres utilisateurs. Le travail de [Guha2003, Jøsang *et al.*2003] confirme que la transitivité de confiance (avec défiance) apporte beaucoup de bénéfices pour les systèmes de confiance et améliore leurs performances.

2.4.7 Le modèle de Richardson

Développé par [Richardson *et al.*2003], ce modèle a comme but d'utiliser la confiance pour filtrer les informations sur le web selon leur plausibilité. Il considère que la croyance d'un utilisateur en une information est une fonction de sa confiance en sa source. Il offre aux utilisateurs deux manières pour exprimer leurs avis :

- La croyance personnelle : pour exprimer la croyance d'un utilisateur à propos d'une ressource disponible, la valeur doit être entre 0 et 1. La valeur ici représente la croyance de l'utilisateur en la précision, la crédibilité et la pertinence de la ressource en question.
- La confiance locale personnelle : pour noter un autre utilisateur, la valeur doit être aussi entre 0 et 1. L'utilisateur exprime ici à quel point il considère l'autre comme étant fiable, crédible, et partage les mêmes intérêts avec lui.

Le modèle utilise une terminologie plus ou moins particulière ; à partir de deux expressions précédentes, il calcule des "croyances fusionnées" et des "confiances fusionnées". Dans ce contexte, le terme "fusionné" correspond à "calculé" ou "implicite", et le terme "personnel" correspond à "déclaré" ou "explicite". La croyance fusionnée est le résultat d'agrégation

des croyances de plusieurs utilisateurs à propos d'un même item. Selon l'auteur, une variété de fonctions peut être utilisée sous conditions d'être commutatives et associatives; il propose par exemple l'addition et la valeur maximale.

La confiance fusionnée est le résultat de la propagation de confiance suivant un chemin entre deux utilisateurs. Il s'agit d'une concaténation des valeurs de confiance entre chaque couple d'utilisateurs le long du chemin. La multiplication et la valeur minimale peuvent être de bons exemples pour des fonctions de concaténation; il est possible aussi d'utiliser d'autres fonctions sous conditions qu'elles soient associatives et distributives sur la fonction d'agrégation.

Le modèle existe en deux versions : centralisée et distribuée.

Analyse du modèle

Ce modèle est une bonne généralisation de la propagation de confiance et de l'agrégation d'avis. Il propose une base théorique solide mais classique, alors que la véritable contribution réside dans la version distribuée. Le modèle ne se préoccupe pas de la collecte de notes. De ce fait, on ne sait pas exactement comment les relations de confiance sont constituées. La question des notes est critique pour les systèmes à base de confiance. La notation des items est déclarative, la relation entre l'utilisateur et l'item est souvent une relation d'appréciation déclarative ou instantanée. Le modèle n'est pas sensible au contexte, et il ne considère pas la défiance.

2.4.8 Le modèle d'O'Donovan

Développé par [O'Donovan and Smyth2005] pour un SRC. Le modèle ajoute une couche de confiance au FC, avec des changements dans la terminologie; l'utilisateur actif est nommé ici *consommateur*, les voisins sont nommés *producteurs*. Le modèle part de la même formule que celle du FC pour prédire les notes. Il défend la thèse qu'un utilisateur qui produit de bonnes recommandations à ses voisins sera considéré plus fiable, et sera privilégié et retenu pour d'autres prédictions par rapport à un autre qui produit peu de recommandations.

La valeur prédite par le FC est jugée correcte si la différence absolue entre cette dernière et la note réelle est inférieure à un seuil précis ϵ . Tant que la valeur est calculée avec la contribution de plusieurs voisins, il est difficile de déterminer l'utilité de chacun. C'est pourquoi le modèle refait ses calculs avec un seul voisin à la fois, de telle sorte qu'il peut juger de la justesse de leurs contributions. Dans ce contexte, la réputation (les auteurs l'appellent confiance au niveau du profil) d'un utilisateur est le pourcentage de prédictions *correctes* pour lesquelles il était impliqué comme producteur, sur le nombre total des prédictions pour lesquelles il était également impliqué comme producteur. Le modèle considère que la réputation d'un utilisateur peut varier par item, il calcule donc aussi sa réputation par item. Elle est représentée par le pourcentage de prédictions *correctes* de l'item où l'utilisateur actuel était producteur, sur le nombre total des prédictions de cet item où l'utilisateur actuel était producteur.

Les auteurs proposent trois méthodes pour insérer la confiance dans le filtrage collaboratif :

- Pondération : remplacer la similarité dans le filtrage collaboratif par la valeur $w(c, p, i)$. où c correspond au consommateur, p au producteur, i à l'item.

$$w(c, p, i) = \frac{2 \times \text{similarite}(c, p) \times \text{reputation}(p, i)}{\text{similarite}(c, p) + \text{reputation}(p, i)}$$

- Filtrage : après avoir utilisé la mesure de similarité du FC pour trouver l'ensemble des voisins, on extrait un sous-ensemble en appliquant un filtre de confiance, alors on garde seulement les voisins dont la réputation pour l'item considéré dépasse un seuil T . On applique la formule 1.2 originelle en impliquant seulement les voisins fiables.

- Combinaison : il s'agit de combiner les deux méthodes précédentes. Après avoir extrait le sous-ensemble des voisins fiables (filtrage), on applique la formule de pondération (en remplaçant la similarité en 1.2 par $w(c, p, i)$).

Analyse du modèle

La particularité de ce modèle réside dans le fait qu'il est le seul modèle implicite ; les valeurs de confiance ne sont pas exprimées directement par les utilisateurs. Les auteurs affirment que le terme compétence pourrait remplacer le terme confiance dans ce contexte, mais nous trouvons que la confiance telle qu'elle est présentée ici pourrait être remplacée par l'utilité de l'utilisateur pour le système.

Ce modèle a été testé sur les données du site Epinions ; les résultats montrent que le modèle améliore la performance du FC. Il est plus résistant que le FC, et les recommandations sont plus faciles à expliquer.

2.4.9 Bitreillis

Présenté par [Victor *et al.*2009], le modèle est graduel, il se base sur les bitreillis [Ginsberg1988] qui se montrent utiles pour raisonner à partir d'informations imprécises. L'élément de base pour ce modèle est le score de confiance, ce score est représenté sous la forme d'un couple $(c \in [0, 1], d \in [0, 1])$ où la variable c représente la confiance et la variable d représente la défiance.

L'espace des scores de confiance est :

$$\mathcal{BL} = ([0, 1]^2, \leq_c, \leq_k, \neg)$$

où $[0, 1]^2$ correspond aux couples (confiance, défiance).

\leq_c à l'ordre des confiances : $(c_1, d_1) \leq_c (c_2, d_2)$ ssi $c_1 \leq c_2$ et $d_1 \geq d_2$.

\leq_k à l'ordre de connaissance : $(c_1, d_1) \leq_k (c_2, d_2)$ ssi $c_1 \leq c_2$ et $d_1 \geq d_2$.

\neg à la négation : $\neg(c_1, d_1) = (d_1, c_1)$.

Les valeurs de confiance dans ce modèle varient de confiance absolue $(1, 0)$, à défiance absolue $(0, 1)$. Il est aussi capable d'exprimer l'ignorance $(0, 0)$, et la surcharge d'informations $(1, 1)$. En général, la somme de c et de d détermine si la valeur est en manque d'informations ($c + d < 1$), ou si elle contient des informations contradictoires ($c + d > 1$), ou si la quantité d'informations est optimale ($c + d = 1$). La partie connaissance est importante dans ce modèle parce qu'il mesure la qualité des propos et des liens de confiance. La propagation de connaissance est monotone, c'est-à-dire que sa valeur croît toujours avec les interactions indépendamment de leurs résultats. La confiance entre deux utilisateurs peut baisser ou augmenter en raison de nouvelles interactions, et aussi en raison de la propagation de confiance et l'acquisition des avis des amis, mais concernant la connaissance, elle s'enrichit toujours par les nouvelles interactions et les avis acquis. L'ignorance est une valeur absorbante de la connaissance. L'avis d'un inconnu n'apporte pas de connaissance d'autant qu'on ne peut pas mesurer la fiabilité de son avis ; c'est la même chose quand on demande l'avis d'un utilisateur qui n'a aucune connaissance du sujet (et qui n'a pas d'ami qui peut l'aider) $F((0, 0), (c, k)) = F((c, k), (0, 0)) = (0, 0)$.

Une relation de confiance entre deux utilisateurs a et b est divisée en deux parties : $R^+(a, b)$ qui représente la confiance de a en b , et $R^-(a, b)$ qui représente la défiance de a envers b .

Le modèle actuel propose quatre fonctions de propagation qui s'accordent pour le traitement des valeurs de confiance et qui agissent différemment à l'égard de la défiance et de l'ignorance. Le tableau 2.3 illustre ces quatre fonctions de propagation, qui définissent quatre comportements

différents : *prop1* ignore les avis des gens indignes de confiance ainsi que ceux des inconnus sans faire de différence entre eux. *prop2* ignore les avis de gens indignes de confiance, et elle considère seulement les valeurs de défiance des utilisateurs inconnus, cela rend les utilisateurs, en général, plus réservés. *prop3* ignore les avis des inconnus, et considère le contraire des avis des utilisateurs jugés indignes de confiance. *prop4* ignore aussi les avis des inconnus, pour ce qui concerne les utilisateurs indignes, elle considère le contraire de leurs avis seulement pour tout jugement positif. La proposition de ces quatre fonctions repose sur des statistiques faites sur les données Epinions ; les auteurs recensent les liens de confiance et défiance dans cette base de données et les ont croisés pour extraire les comportements que les utilisateurs mènent avec leurs homologues crédibles et non-crédibles.

	Confiance ((1,0),(c,d))	Défiance ((0,1),(c,d))	Ignorance ((0,0),(c,d))
prop1	(c,d)	(0,0)	(0,0)
prop2	(c,d)	(0,0)	(0,d)
prop3	(c,d)	(d,c)	(0,0)
prop4	(c,d)	(0,c)	(0,0)

TABLE 2.3 – Les quatre fonctions de propagation du modèle bitreillis

Analyse du modèle

Le modèle sépare la confiance et la défiance. Il traite l'ignorance et les informations contradictoires, et propose plusieurs fonctions de propagation basées sur des observations des comportements réels. Malgré cela, on remarque l'absence d'une fonction d'agrégation d'avis. On ne sait comment réagit un utilisateur qui reçoit plusieurs avis sur un même thème. De plus, le contexte n'est pas pris en compte.

2.4.10 La logique subjective et les réseaux de confiance

La logique subjective (LS)⁹ est une extension de la logique probabiliste, qui associe un degré d'incertitude à chaque probabilité. Elle permet de construire des modèles qui traitent des situations de preuves incomplètes. La LS n'est pas elle-même un modèle de confiance. Elle est un cadre formel efficace pour formaliser un modèle de confiance.

La théorie de la croyance [Dempster2008, Yager *et al.*1994] est un cas particulier de la théorie des probabilités dédié à traiter une connaissance incomplète. La somme des probabilités de cas possibles peut être inférieure à 1. La logique subjective propose un calcul de la croyance en utilisant une représentation de croyance sous la forme d'une opinion [Jøsang2001]. L'opinion d'un individu U sur une déclaration x est représentée par :

$$\omega_x^U = (b, d, u, a)$$

Où : $b, d, u \in [0, 1]$ sont respectivement la croyance, l'incrédulité et l'incertitude de U sur x . La somme des trois valeurs est égale à 1 ($b + d + u = 1$). Le taux de base $a \in [0, 1]$ représente la probabilité à priori. Fondamentalement, le taux de base est une mesure statistique appliquée dans le cas de l'absence totale de preuves. Par exemple, sachant que le pourcentage d'une maladie

9. Josang Audun. "Subjective logic" [Jøsang and Pope2005]. Projet du livre, 2013.
http://folk.uio.no/josang/papers/subjective_logic.pdf.

z dans une population donnée est de 1%, alors, le taux de base de l'infection par z est de 1%. Lorsqu'on rencontre une personne de cette population qui n'a pas effectué un test pour la maladie, à priori on suppose que la probabilité qu'il soit infecté est de 1%. Dans le cas de la confiance computationnelle, alors qu'aucune statistique à priori n'est présente, habituellement on considère qu'une personne inconnue a une demi-chance d'être jugée comme digne de confiance. Donc, on utilise un taux de base $a = 0,5$. Dans la logique subjective, le taux de base dirige la contribution de l'incertitude dans le calcul de la valeur d'espérance mathématique de la probabilité selon l'équation 2.3 :

$$E(\omega_x^U) = b + a \times u \quad (2.3)$$

L'opinion dans la logique subjective est basée sur l'accumulation d'expériences réussies et échouées. Après chaque expérience, U met son opinion sur x à jour de façon à être en conformité avec le résultat de l'expérience. Selon cette description, l'opinion peut être représentée comme une variable aléatoire binaire. La distribution bêta est normalement utilisée pour modéliser le comportement de ce type de variables. En conséquence, l'opinion correspond à la fonction de densité de probabilité (PDF) de distribution bêta. PDF est notée par deux paramètres de preuves α et β qui peuvent être écrits comme des fonctions du nombre d'expériences réussies et ratées respectivement.

$$\begin{aligned} \alpha &= r + W \times a \\ \beta &= s + W \times (1 - a) \end{aligned} \quad (2.4)$$

Où r est le nombre d'expériences réussies (preuves). s est le nombre d'expériences qui ont échoué. W est le poids non-informatif apriori, il assure qu'apriori (c'est-à-dire quand $r = s = 0$) Beta PDF avec un taux de base par défaut $a = 0,5$ est un PDF uniforme. Normalement $W = 2$.

La valeur moyenne de la bêta PDF est :

$$E(\text{Beta}(p|\alpha, \beta)) = \frac{\alpha}{\alpha + \beta} = \frac{r + Wa}{r + s + W} \quad (2.5)$$

Dans la logique subjective, la correspondance entre les paramètres d'opinion et les paramètres PDF bêta est donnée comme suit :

$$b = \frac{r}{r + s + W} \quad (2.6)$$

$$d = \frac{s}{r + s + W} \quad (2.7)$$

$$u = \frac{W}{r + s + W} \quad (2.8)$$

Le tableau 2.4 montre un exemple de l'évolution d'une opinion avec des interactions successives.

No	status	r	s	croissance	incrédulité	incertitude
0	pas d'interaction	0	0	0	0	1
1	interaction réussite	1	0	1/3	0	2/3
2	interaction échouée	1	1	1/4	1/4	2/4
3	interaction réussite	2	1	2/5	1/5	2/5

TABLE 2.4 – L'évolution d'opinion avec des interactions successives

Dans la première ligne du tableau 2.4, on remarque le cas de l'absence de preuves (expériences). L'opinion est complètement incertaine ($u = 1$). Dans ce cas, selon l'équation 2.3, la valeur moyenne est égale à la valeur du taux de base. L'arrivée de nouvelles expériences fait que la connaissance augmente et que l'incertitude diminue, peu importe si ces expériences ont réussi ou échoué. Les expériences réussies permettront d'accroître la croyance, alors que les expériences infructueuses viendront compléter l'incrédulité.

L'opinion dans la logique subjective peut être illustrée par l'intérieur d'un triangle équilatéral. Les trois sommets du triangle sont appelés croyance, incrédulité et incertitude. Le lien entre le sommet d'incertitude et le bord opposé (bord croyance-incroyance) forme l'axe d'incertitude, la valeur de l'incertitude de l'opinion est tracée sur cet axe, étant donné que son intersection avec le bord croyance-incroyance représente la valeur 0 (opinion complètement certaine), alors que l'intersection avec le sommet d'incertitude représente la valeur 1 (opinion complètement incertaine). De la même manière, on décrit l'axe de croyance et l'axe d'incrédulité.

L'opinion est représentée par le point d'intersection des trois projections sur les trois axes (la croyance, l'incrédulité et la certitude) et est illustrée dans l'exemple de la figure 2.6. La base du triangle est l'axe de probabilité, la valeur de la probabilité est la projection du point d'opinion sur l'axe de probabilité par rapport à la ligne qui relie le sommet d'incertitude avec le point de taux de base sur l'axe de probabilité (la ligne rouge dans la figure 2.6). La figure 2.6 illustre un exemple de représentation de l'opinion dans la logique subjective. L'opinion est représentée par un point à l'intérieur du triangle. Le point est l'intersection de la projection des trois valeurs b , d et u sur l'axe de l'incrédulité, de la croyance et de l'incertitude respectivement. La valeur de l'espérance mathématique $E(x)$ est la projection de ω_x sur l'axe de probabilité dirigé par l'axe reliant a_x avec le sommet de l'incertitude.

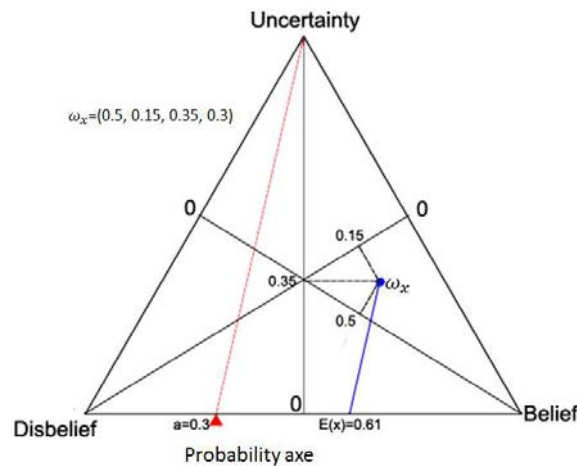


FIGURE 2.6 – Opinion

Notez que le changement de la valeur du taux de base peut rendre les gens plus téméraires ou plus prudents. Ce qui permet au concepteur de modéliser la confiance tout en tenant compte de la différence entre les utilisateurs de nature réservée et ceux de nature confiante.

Après avoir défini la structure de l'opinion dans la logique subjective, nous avons besoin d'expliquer certains des opérateurs de la logique subjective qui sont utiles pour la construction de réseaux de confiance. La logique subjective propose une grande variété d'opérateurs comme l'addition, la soustraction, la multiplication, la négation, la transitivité et l'agrégation des opi-

nions. Sa structure flexible permet aussi d'en ajouter d'autres selon le besoin [Cerutti *et al.*2013]. La section suivante est limitée aux opérateurs impliqués dans le modèle de SRC. Les réseaux de confiance locale sont généralement représentés par un graphe orienté, où les sommets représentent les utilisateurs, et les arcs représentent des relations de confiance. En conséquence, le calcul de la confiance entre deux utilisateurs est réduite à trouver un chemin ou plus les reliant l'un à l'autre.

2.4.10.1 La transitivité de la confiance

Si un utilisateur A fait confiance à un autre utilisateur B , et B fait confiance à C , l'objectif d'un opérateur de transitivité de la confiance est de dériver la relation entre A et C . Cela ne paraît pas compliqué pour des valeurs positives, A fera confiance à C et la valeur sera conditionnée par sa confiance en B et la confiance de ce dernier en C . La transitivité devient compliquée quand A considère B indigne de confiance, il n'y a pas de preuve sur la manière avec laquelle il va considérer l'avis de B à propos de C . Pour cette raison, la logique subjective propose trois fonctions de propagation de confiance basées sur trois scénarios possibles :

2.4.10.1.1 Transitivité en favorisant l'incertitude Lorsque A considère B plutôt indigne de confiance, il considère que B ignore la vérité de C , ou bien qu'il n'a pas autant de légitimité pour juger C . En conclusion, son avis ne peut guère influencer l'avis de A . La formule 2.9 illustre comment l'opinion de A en C est calculée selon cette fonction.

$$\begin{aligned}\omega_B^A &= (b_B^A, d_B^A, u_B^A, a_B^A) \\ \omega_C^B &= (b_C^B, d_C^B, u_C^B, a_C^B) \\ \omega_B^A \otimes \omega_C^B &= \begin{cases} b_C^{A:B} = b_B^A \cdot b_C^B \\ d_C^{A:B} = b_B^A \cdot d_C^B \\ u_C^{A:B} = d_B^A + u_B^A + b_B^A \cdot u_C^B \\ a_C^{A:B} = a_C^B \end{cases} \quad (2.9)\end{aligned}$$

Prenons l'exemple où $\omega_B^A = (0, 1; 0, 8; 0, 1; 0, 5)$ et $\omega_C^B = (0, 7; 0, 2; 0, 1; 0, 5)$, l'opinion de A en C sera $\omega_C^{A:B} = (0, 07; 0, 02; 0, 91; 0, 5)$. Remarquons la valeur élevée de l'incertitude dans ω_C^A . Supposons que l'avis de A en B était inversé $\omega_B^A = (0, 8; 0, 1; 0, 1; 0, 5)$, en gardant la même valeur de $\omega_C^B = (0, 7; 0, 2; 0, 1; 0, 5)$, on obtient $\omega_C^{A:B} = (0, 56; 0, 16; 0, 28; 0, 5)$.

2.4.10.1.2 Transitivité en favorisant l'incrédulité Tant que A juge B indigne de confiance, il considère que ce dernier va lui fournir une opinion opposée à la vérité de C . Par conséquent, il réagira à l'opposé de la recommandation de B . Cette fonction est représentée par la formule suivante 2.10 :

$$\omega_B^A \otimes \omega_C^B = \begin{cases} b_C^{A:B} = b_B^A \cdot b_C^B + d_B^A \cdot d_C^B \\ d_C^{A:B} = b_B^A \cdot d_C^B + d_B^A \cdot b_C^B \\ u_C^{A:B} = u_B^A + (b_B^A + d_B^A) \cdot u_C^B \\ a_C^{A:B} = a_C^B \end{cases} \quad (2.10)$$

Revenons à l'exemple précédent : $\omega_B^A = (0, 1; 0, 8; 0, 1; 0, 5)$ et $\omega_C^B = (0, 7; 0, 2; 0, 1; 0, 5)$, cela donne $\omega_C^{A:B} = (0, 23; 0, 58; 0, 19; 0, 5)$. La tendance de A est plutôt de ne pas faire confiance à C tant qu'il est apprécié par B . A est plus certain de son opinion sur C qu'avec la fonction précédente.

2.4.10.1.3 Transitivité sensible au taux de base Supposons que la voiture de A tombe en panne alors qu'il est dans une ville où il ne connaît personne, tout ce qu'il sait est que les habitants de cette ville sont connus pour être honnêtes et serviables. Il demande à la première personne (B) qu'il croise dans la rue où il peut trouver un bon garagiste. B lui recommande C . Comment A va interpréter l'avis de B ? La seule information sur laquelle il peut se reposer est que la réputation préalable des habitants de la ville est bonne (le taux de base). La formule de cette fonction est :

$$\omega_B^A \otimes \omega_C^B = \begin{cases} b_C^{A:B} = E(\omega_B^A) \cdot b_C^B \\ d_C^{A:B} = E(\omega_B^A) \cdot d_C^B \\ u_C^{A:B} = 1 - E(\omega_B^A) \cdot (b_C^B + d_C^B) \\ a_C^{A:B} = a_C^B \end{cases} \quad (2.11)$$

où $E(\omega_B^A)$ est la même que dans l'équation 2.3.

En revanche, il faut être vigilant lors de l'utilisation de cette fonction, car elle peut générer facilement de fausses certitudes. Prenons le cas où $\omega_B^A = (0; 0; 1; 0, 9)$ A est quasiment sûr de la crédibilité des habitants de la ville de B , et si $\omega_C^B = (1; 0; 0; 0, 9)$. Le résultat sera $\omega_C^{A:B} = (0, 9; 0; 0, 1; 0, 9)$. La valeur haute de croyance dans $\omega_C^{A:B}$ est dérivée de la valeur haute d'incertitude dans ω_B^A , ce qui paraît inhabituel.

2.4.10.2 Agrégation d'opinions

L'agrégation des avis sert à unifier les avis de deux utilisateurs ou plus à propos du même utilisateur. Quand A demande les avis de ses amis B et C à propos de D , chacun retourne une opinion différente de l'autre. A a besoin d'une opinion unique sur D , il doit donc fusionner les deux avis. La logique subjective propose deux opérateurs pour agréger les avis. Ils sont décrits aux paragraphes suivants.

2.4.10.2.1 Agrégation moyenne Cet opérateur présume que les deux utilisateurs (B et C) ont observé D durant la même période, et que leurs avis sont les fruits des mêmes interactions entre eux et lui. Supposons que B et C soient un couple, et que D a réparé leur voiture, B trouve que le travail de D est satisfaisant, alors que C trouve que D est cher. Les avis de B et C sont basés sur deux évaluations différentes de la même interaction. La formule de cet opérateur peut prendre deux formes :

Lorsque $u_D^B \neq 0$ ou $u_D^C \neq 0$:

$$\omega_B^C \oplus \omega_D^C = \begin{cases} b_D^{B \circ C} = \frac{b_D^A \cdot u_B^B + b_D^B \cdot u_D^A}{u_D^B + u_D^C} \\ d_D^{B \circ C} = \frac{d_D^A \cdot u_B^B + d_D^B \cdot u_D^A}{u_D^B + u_D^C} \\ u_D^{B \circ C} = \frac{2 \cdot u_D^B \cdot u_D^C}{u_D^B + u_D^C} \end{cases} \quad (2.12)$$

Le deuxième cas est lorsque $u_D^B = 0$ et $u_D^C = 0$:

$$\omega_D^B \oplus \omega_D^C = \begin{cases} b_D^{B \circ C} = \gamma^B \cdot b_D^B + \gamma^C \cdot b_D^C \\ d_D^{B \circ C} = \gamma^B \cdot d_D^B + \gamma^C \cdot d_D^C \\ u_D^{B \circ C} = 0 \end{cases} \quad \text{où} \quad \begin{cases} \gamma^B = \lim_{\substack{u^B \rightarrow 0 \\ u^C \rightarrow 0}} \frac{u^C}{u^B + u^C} \\ \gamma^C = \lim_{\substack{u^B \rightarrow 0 \\ u^C \rightarrow 0}} \frac{u^B}{u^B + u^C} \end{cases} \quad (2.13)$$

2.4.10.2.2 Agrégation cumulative Cet opérateur est utilisé lorsque les avis de B et C ne sont pas le fruit de la même observation de D . Chacun a construit son avis sur D grâce à ses propres interactions. La formule de cet opérateur peut prendre deux formes :

Lorsque $u_D^B \neq 0$ ou $u_D^C \neq 0$:

$$\omega_D^B \oplus \omega_D^C = \begin{cases} b_D^{B \circ C} = \frac{b_D^A \cdot u_D^B + b_D^B \cdot u_D^A}{u_D^B + u_D^C - u_D^B \cdot u_D^C} \\ d_D^{B \circ C} = \frac{d_D^A \cdot u_D^B + d_D^B \cdot u_D^A}{u_D^B + u_D^C - u_D^B \cdot u_D^C} \\ u_D^{B \circ C} = \frac{u_D^B \cdot u_D^C}{u_D^B + u_D^C - u_D^B \cdot u_D^C} \end{cases} \quad (2.14)$$

Le deuxième cas est lorsque $u_D^B = 0$ et $u_D^C = 0$:

$$\omega_D^B \oplus \omega_D^C = \begin{cases} b_D^{B \circ C} = \gamma^B \cdot b_D^B + \gamma^C \cdot b_D^C \\ d_D^{B \circ C} = \gamma^B \cdot b_D^B + \gamma^C \cdot b_D^C \\ u_D^{B \circ C} = 0 \end{cases} \quad \text{où} \quad \begin{cases} \gamma^B = \lim_{\substack{u^B \rightarrow 0 \\ u^C \rightarrow 0}} \frac{u^C}{u^B + u^C} \\ \gamma^C = \lim_{\substack{u^B \rightarrow 0 \\ u^C \rightarrow 0}} \frac{u^B}{u^B + u^C} \end{cases} \quad (2.15)$$

Analyse du modèle

Le modèle est proposé théoriquement sans être testé sur des données réelles. Il repose sur une base théorique solide et riche de la logique subjective. Ses multiples opérateurs lui permettent d'être ajustable selon plusieurs contextes et applications. Il sépare la confiance et la défiance ce qui lui permet de distinguer les individus inconnus des individus indignes de confiance. Les notes sont inférées, ce qui est cohérent avec le principe de confiance.

Il ajoute aussi un argument très intéressant dans le domaine de la confiance : la certitude. Il ne défend que ce qui compte dans une opinion, c'est-à-dire la certitude du porteur de cette opinion indépendamment de la distance entre lui et le confiant. Plusieurs modèles de confiance mettent la distance entre le confiant et le crédible en avant, et considèrent que plus la distance entre les deux est petite, plus ils vont accepter de coopérer. Ce n'est pas le cas dans ce modèle, car plus l'utilisateur fait des interactions (gagne de l'expérience), plus sa certitude augmente (ou son incertitude baisse). Ce modèle considère que le confiant a intérêt à coopérer avec des gens qui ont plus d'expérience (moins d'incertitude), et pas avec ceux qui sont simplement plus proches de lui dans le réseau.

2.5 Résumé

Le tableau 2.5 illustre les différences essentielles entre les modèles de confiance étudiés dans ce chapitre.

		advogato	eigenTrust	MoleTrust	TidalTrust	AppleSeed	Abdulrahmen	Marche aléatoire	Odonovan	Bitreillis	logique subjective
Type d'appli- cation		Réseau social	pair-à-pair	SR	SR	-	SR	Web séma- ntique	SR	-	-
Local	Global	g	g	l	l	l	l	l	g/l	l	l
Notes	EXplicite IMplicite	x	x	x	x	-	-	-	m	-	-
	Déclarative Inférée	d	i	d	d	d	i	d	d	i	i
	Probabiliste Graduel	g	p	g	g	g	g	p	g	g	p
Propagation	oui	oui	oui	oui	oui	non	oui	oui	oui	oui	
Agrégation d'avis	non	non	non	oui	oui	non	non	oui	oui	oui	
Contextuel	non	non	non	non	non	non	oui	non	non	non	non
Défiance	non	non	non	non	non	non	oui	non	non	oui	oui

TABLE 2.5 – Résumé des modèles de confiance

La caractéristique principale choisie pour classer les modèles de confiance est souvent la forme des relations de confiance qu'ils adoptent (confiance locale, confiance globale). Les modèles à base de confiance locale souffrent d'un léger problème de démarrage à froid. D'un autre côté, l'utilisateur a un contrôle quasi total sur le comportement du système, d'autant qu'il choisit ses amis lui-même. Ce genre de système est résistant face aux attaques malveillantes. Le modèle qui utilise la confiance globale souffre d'une autre forme de problème de démarrage à froid, d'autant qu'un nouvel arrivant doit pouvoir construire sa réputation au sein de la communauté. Cette tâche devient difficile si le système continue à diriger les demandeurs de services vers les utilisateurs déjà réputés. Les modèles de confiance globale sont aussi fragiles face à des attaques de groupes. Advogato a proposé une solution efficace à ce problème en liant tous les membres à un petit groupe d'utilisateurs fiables, ce que rend le système extrêmement dépendant de ce petit groupe d'utilisateurs.

Concernant les notes dans les modèles de confiance computationnelle, nous pouvons décrire deux attributs pour les notes :

- explicite/implicite : comme dans les systèmes de recommandation, la différence ici est entre les notes déclarées directement par l'utilisateur, ou calculées par le modèle à partir des traces laissées par les utilisateurs.
- déclarative/inférée : plusieurs des modèles vus dans ce chapitre traitent des notes de confiance entre utilisateurs comme des notes d'appréciation des items. Dans un contexte de confiance, on peut plutôt apprécier les résultats d'une interaction. La confiance est plutôt cumulative, et n'est pas instantanée comme l'appréciation des items. Quand un utilisateur

change sa note pour un item, il est normal de supprimer l'ancienne note, alors qu'avec la confiance, la nouvelle note est ajoutée à l'ancienne en quelque sorte. Les notes inférées peuvent avoir des formes probabilistes, graduelles ou bien autres. Indépendamment de leur forme, nous sommes convaincus que les notes inférées sont mieux capables de représenter la confiance.

A part un seul modèle qui considère la confiance intransitive, tous les modèles proposent au moins une fonction de propagation de confiance. Les fonctions varient selon plusieurs critères :

- Choisir entre le chemin le plus court ou le chemin le plus sûr.
- Traiter ou pas la défiance.
- Choisir un chemin unique ou agréger les avis en provenance de plusieurs chemins.

La propagation de confiance est plus compliquée pour les modèles qui considèrent la défiance. Pour propager la confiance, on peut trouver un comportement intuitif en accordant une confiance plus faible aux amis de nos amis ; cela est plus difficile lors du traitement de la défiance et de l'ignorance. Il n'y a pas une manière unique de traiter ces situations. Dans le cas de l'ignorance par exemple, il n'est pas évident de savoir si l'utilisateur ignorera l'avis ou fera le contraire.

La confiance n'est pas une valeur absolue, elle est liée fortement au contexte. Au début du chapitre, nous avons expliqué les trois composants principaux du contexte (le sujet de la tâche, son importance et son urgence). Bien que la plupart des définitions de la confiance apportées dans le domaine informatique mentionnent de près ou de loin le contexte, peu de modèles proposés l'intègrent dans leurs propositions. Parmi les modèles étudiés dans ce chapitre, il n'y a que le modèle d'Abdul-Rahman et al (voir 2.4.6) qui intègre explicitement le contexte (représenté seulement par le sujet de tâche) dans son modèle. Il y a aussi le modèle d'O'Donovan et al (voir 2.4.8) qui calcule une sorte de réputation de l'utilisateur par item, et engage seulement les utilisateurs qui sont connus comme de bons recommandeurs *pour cet item* dans ce calcul. Cela peut être aussi considéré comme une forme de conscience du contexte. Encore une fois, le contexte est résumé par le sujet de la tâche. L'intégration du contexte donne plus de sens à la confiance.

Finalement, les modèles varient aussi par leur interprétation de la confiance sociale et de ses caractéristiques (plage des valeurs, représentation de la défiance et de l'ignorance, contexte ...).

Deuxième partie

Travail réalisé

Hybridation du filtrage collaboratif avec la confiance

Sommaire

1.1	Introduction	55
1.2	Données et contexte	56
1.3	Motivation	57
1.4	Hybridation	58
1.4.1	Stratégies d'hybridation	59
1.4.2	Méthode d'évaluation	60
1.4.3	Analyse des résultats	61
1.5	Analyse du comportement des utilisateurs	65
1.5.1	Les caractéristiques du comportement de l'utilisateur	65
1.5.2	Classification d'utilisateurs	72
1.6	Conclusion	79

1.1 Introduction

Nous nous sommes intéressés à la recommandation dans le contexte du web social, où les utilisateurs interagissent entre eux et expriment leurs avis les uns à l'égard des autres. Ces avis permettent aux SR à base de confiance de remplacer la similarité entre utilisateurs calculée par le FC par des relations de confiance exprimées directement par les utilisateurs.

Les deux approches (FC, SRC) souffrent d'un problème de fluctuation des performances selon les utilisateurs. Certaines catégories d'utilisateurs sont privées de recommandation ou en reçoivent qui sont inappropriées.

Dans ce chapitre, nous nous intéressons à ce problème de fluctuation. Nous proposons l'hybridation des deux approches comme solution, car certains utilisateurs ont un comportement adapté à seulement une des deux approches. L'hybridation permet d'assurer une bonne recommandation pour ce genre d'utilisateurs. Nous testons plusieurs hybridations possibles afin de trouver la meilleure solution.

Ensuite, nous tentons d'expliquer ce phénomène de fluctuation en analysant le comportement des utilisateurs. Nous introduisons de nouveaux critères qualitatifs, et nous montrons leurs effets sur la qualité de recommandation.

Dans la dernière étape, nous construisons un profil utilisateur constitué des attributs qualitatifs et quantitatifs de son comportement. Puis, nous utilisons un algorithme de classification pour définir des classes comportementales d'utilisateurs. Par la suite, nous analysons la performance des approches de recommandation sur ces classes pour identifier l'approche la plus appropriée par classe.

Dans ce travail, nous utilisons des données réelles pour valider nos propositions.

1.2 Données et contexte

Nous emploierons le corpus Epinion pour notre travail, car il contient à la fois des notes utilisateur-item, et des notes utilisateur-utilisateur. Par conséquent, il met à notre disposition les données nécessaires pour le FC et le SR à base de confiance. De plus, ce corpus fait l'objet de plusieurs travaux qui comparent les deux approches [Massa and Avesani2004, Ma *et al.*2011, Guo *et al.*2014].

Epinions est un réseau social dédié aux avis des consommateurs. L'utilisateur est invité à créer son compte, à naviguer sur le site et à explorer son immense catalogue de produits. Les utilisateurs peuvent noter les produits avec une échelle de 1 (le produit est mauvais) à 5 (le produit est appréciable). Ils peuvent aussi rédiger des commentaires afin d'expliquer leurs avis, noter les commentaires des autres utilisateurs, exprimer leur confiance/défiance envers leurs homologues. Quand un utilisateur *A* trouve qu'un autre utilisateur *B* fournit souvent des commentaires utiles, il peut l'ajouter à son "web de confiance", autrement dit il déclare sa confiance en lui. Dans le cas où *A* trouve que les commentaires de *B* sont pauvres ou inutiles, il ajoute *B* à sa "liste noire", ce qui signifie la défiance. La figure 1.1 illustre un commentaire rédigé sur le site, le commentaire contient toujours les "pros" (les qualités du produit), les "cons" (les inconvénients du produit), et l'avis de l'utilisateur sous forme de textes entre 20 et 100 mots. A droite, on trouve un bref profil de l'utilisateur, et en-dessous de ce dernier, deux choix pour exprimer la confiance ou la défiance en cet utilisateur.

Les données d'Epinions ont été publiées la première fois par [Massa and Avesani2004]. La version publiée contient au total 49290 utilisateurs, dont 3470 n'ont jamais ni noté d'items ni d'utilisateurs, ce qui réduit le nombre d'utilisateurs inclus dans nos calculs à 45820 utilisateurs. Cette base de données contient également 139738 items, et un nombre total de 664824 notes utilisateur-item. La figure 1.2 fournit un histogramme de la distribution des notes dans ces

See all Reviews | Write a Review

Great utility laptop - has everything you need

★★★★☆ by shawnfraser

Aug 11, 2013 (Updated Aug 12, 2013)

Rated a **Very Helpful Review** by the Epinions community

User Rating: Excellent	Pros: HP support. Lots of RAM. Big drive. Fast reasonable processor. Backlit LED display. INEXPENSIVE!
Ease of Use: ██████████	Cons: No numlock indicator. Annoying wakeup issues with the trackpad driver.
Quality of Tech Support: ██████████	The Bottom Line: A great choice for a value machine ... and Epinions needs to update their computer profiles

There are so many variations of HP's laptops that it is virtually impossible to write one for a specific model, since what YOU run into will almost certainly differ from what I bought - it almost seems as if they equip these with varying processors, hard drives, memory and accessories that no two are quite alike.

But I figure that as long as you have the same case, display and processor, the review is valid.

Last month my trusty, 11 year old Sony Vaio ultralight laptop finally died and I needed a replacement quickly - I suppose that while I could have had the Sony repaired, it was slow (750 MHz), lacking in memory (512 megs) and had a small (40 Gig) hard drive. But it did what I needed.

I wanted to keep costs down, so I looked at the various offerings from Asus, Toshiba, Lenovo, Dell and so on, but I decided on an HP for several reasons, not the least of which is excellent support.

About the Author

Epinions.com ID: shawnfraser
Member: shawnfraser
Location: Montreal, Canada
Reviews written: 25
Trusted by: 1 member

Web of Trust

Trust shawnfraser
Block shawnfraser
Whom should I trust?

FIGURE 1.1 – Un commentaire rédigé dans Epinions

données, en sachant que la note moyenne est 3,99.

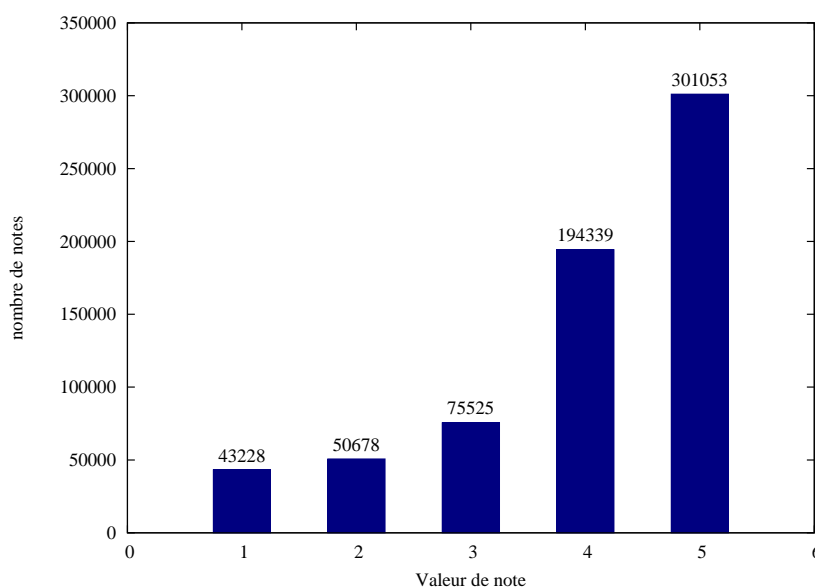


FIGURE 1.2 – Histogramme des notes aux données Epinion

Dans cette version de la base de données, les éditeurs du site ont seulement fourni les notes de confiance (sans les notes de défiance), cette base de données contient 487182 notes utilisateur-utilisateur. Toutes les notes ont une valeur unique égale à +1.

1.3 Motivation

[Massa and Avesani2004] ont proposé un SR à base de confiance, qui utilise l’algorithme Moletrust pour modéliser la confiance (voir paragraphe 2.4.4). Ils ont comparé sa performance avec celle du FC sur les données Epinions [Massa and Bhattacharjee2004, Massa and Avesani2004]. Leurs résultats montrent que la performance de MoleTrust surpasse celle du FC, et qu’elle continue à s’améliorer avec l’augmentation de la distance maximale de propagation(d) jusqu’à ($d = 4$), ensuite l’amélioration ne devient plus significative.

MoleTrust améliore la précision de la recommandation par rapport au FC, car elle réduit l’impact de certains de ses problèmes comme le démarrage à froid, le manque de données, le passage à l’échelle, et l’explication de recommandation [Massa and Avesani2004]. Cela, grâce aux notes utilisateur-utilisateur qui sont plus facilement exploitables, et plus transparentes pour l’utilisateur. De plus, la propagation de confiance sert à enrichir la matrice de similarité entre utilisateurs, et par conséquent améliore la qualité de la recommandation.

Bien qu’elle soit plus performante que le FC, l’approche à base de confiance ne montre pas plus d’adaptation aux comportements des utilisateurs. Car cette amélioration de performance n’est pas forcément en faveur de tout le monde. Dans la base de données d’Epinions, 11858 utilisateurs (soit 25,8%) n’ont fait confiance à personne, un système de recommandation à base de confiance ne peut rien leur recommander. Ces utilisateurs ne sont pas complètement inactifs non plus, car ils ont attribué au total 75109 notes aux items, ce qui fait une moyenne de 6,3 notes utilisateur-item par utilisateur. Un système de filtrage collaboratif peut être efficace pour ces utilisateurs. On distingue aussi un autre groupe de 5655 utilisateurs qui n’ont noté aucun

item alors qu'ils ont fait confiance en moyenne à 4,3 utilisateurs chacun.

Il apparaît que certains utilisateurs ne s'intéressent pas à l'aspect social du site, leur intérêt est focalisé sur la notation des items. Alors que d'autres font l'inverse, ils s'intéressent plus à l'aspect social du site et aux activités des autres utilisateurs. Le contraste de comportement fait que les deux SR (FC et confiance) souffrent d'un problème de fluctuation de performance par utilisateur, chacun est satisfaisant pour un groupe d'utilisateurs et pas pour l'autre. Cela nous conduit à examiner les systèmes de recommandation hybrides qui combinent deux techniques de recommandation afin de palier mutuellement les problèmes.

Quand on dit que le comportement de tel utilisateur n'est pas adapté à tel algorithme, le but n'est pas de mettre en cause son comportement, mais au contraire d'aider le système à s'adapter à ce type de comportement.

Les questions que nous traitons dans ce chapitre sont : comment construire un système de recommandation hybride qui agit face à une variation comportementale de ses utilisateurs en maximisant le nombre d'utilisateurs qui reçoivent des recommandations ? Quelles sont les caractéristiques qualitatives et quantitatives qui aident à identifier le comportement de l'utilisateur et son influence sur la performance du système ? Y a-t-il d'autres groupes comportementaux qui se forment au sein de la population des utilisateurs ? Comment utiliser les caractéristiques extraites pour les identifier et pour trouver un algorithme convenant à chaque groupe d'utilisateur ?

Nos travaux suivent les étapes suivantes : tout d'abord, nous allons proposer une solution pour hybrider les deux approches de recommandation (FC et confiance) ; l'évaluation de l'hybridation consiste à mesurer l'amélioration de la couverture apportée par la méthode, tout en faisant attention à la précision de la prédiction. En effet, le but est d'augmenter le nombre d'utilisateurs profitant de recommandations tout en gardant un bon niveau de précision.

La phase suivante est dédiée à l'extraction d'erreurs pour identifier les raisons comportementales qui font qu'un utilisateur ne reçoit pas de recommandation ou en reçoit de mauvaise qualité. Pour cela, nous définissons un ensemble de caractéristiques qualitatives et quantitatives liées à la stratégie de l'utilisateur pour noter des items et pour faire confiance à ses amis. L'objectif est de mettre en lumière le lien entre chaque caractéristique et la performance de chaque approche de recommandation.

Dans la dernière phase, nous rassemblons les caractéristiques par utilisateur pour créer un vecteur comportemental de l'utilisateur. Ensuite nous traçons les comportements composés et leurs impacts sur la performance de chaque approche de recommandation. Nous utilisons pour cela une classification non supervisée des vecteurs comportementaux, et une étude descriptive de chaque classe.

1.4 Hybridation

Les systèmes de recommandation hybrides sont utilisés pour combiner deux approches de recommandation, dont le but est souvent de pallier les inconvénients de chacune [Burke2002]. C'est exactement ce dont nous avons besoin pour réduire la fluctuation du FC et de MoleTrust.

L'hybridation peut être envisagée de deux manières. La première repose sur une prédiction indépendante ; chaque approche utilise sa propre fonction de prédiction afin de prédire les notes utilisateur-item et générer une liste de recommandations. Dans ce cas, le rôle du système d'hybridation est de combiner les deux listes. La deuxième manière s'appuie sur une prédiction commune ; les deux approches coopèrent pour calculer une note. Dans ce cas le modèle hybride a une fonction de prédiction composée.

Dans notre application, nous ne proposons pas une liste de recommandations à l'utilisateur.

En revanche, nous tentons de prédire toutes ses notes pour avoir une vision plus complète de son comportement, autrement dit, trouver les items qu'il aime, et ceux qu'il n'aime pas. De plus, nous ajoutons une mesure supplémentaire pour évaluer l'efficacité du SR à prédire les items les plus appréciés par l'utilisateur, pour assurer sa capacité à générer des recommandations pertinentes.

1.4.1 Stratégies d'hybridation

Nous examinons les deux approches initiales (FC et MoleTrust), et cinq stratégies d'hybridation : pondérée, alternance, mixte, cascade et probabiliste (voir 1.1.4). La cascade et l'alternance sont sensibles à l'ordre des approches hybridées, par conséquent nous allons les appliquer deux fois chacune en changeant l'ordre. La stratégie pondérée contient un paramètre $\alpha \in [0; 1]$, nous allons l'appliquer avec plusieurs valeurs de ce paramètre. Voici un résumé des méthodes de recommandation utilisées :

1. FC : Une méthode du filtrage collaboratif classique, qui utilise la similarité de Pearson (voir l'équation 1.1). En sachant que dans cette version, on intègre tous les voisins ayant un score de similarité autre que 0 (positif ou négatif).
2. MoleTrust : C'est le même algorithme de MoleTrust que dans la section 2.4.4, il est appliqué ici avec une distance maximale de propagation égale à 4. Cela veut dire que dans le calcul de la prédiction, on intègre les amis crédibles, leurs amis, et ainsi jusqu'à une distance de quatre étapes par rapport à l'utilisateur actif. Le choix de cette valeur s'appuie sur l'optimisation de MoleTrust par Massa sur ces données [Massa and Bhattacharjee2004].
3. Hybride cascade (FC, Mole) : FC est la méthode dominante dans cette stratégie. Elle fournit C_{FC} l'ensemble des voisins extraits, en utilisant le FC. Seulement les voisins en C_{FC} seront passés à MoleTrust. Le score final sera celui de MoleTrust sur une collection de voisins choisis par le FC :

$$simil(u_a, u_b) = \begin{cases} simil_{Mole}(u_a, u_b) & \text{si } simil_{FC}(u_a, u_b) > 0 \\ 0 & \text{sinon} \end{cases}$$

où $simil$ est la similarité, u_a, u_b sont deux utilisateurs.

4. Hybride cascade (Mole, FC) : Ici nous faisons l'inverse de la méthode précédente ; c'est MoleTrust qui propose C_{mole} au départ, ensuite on calcule le coefficient de Pearson entre chaque membre de C_{mole} et l'utilisateur actif. Cela veut dire qu'on choisit les utilisateurs ayant un comportement similaire à u_a parmi ses amis fiables :

$$simil(u_a, u_b) = \begin{cases} simil_{FC}(u_a, u_b) & \text{si } simil_{Mole}(u_a, u_b) > 0 \\ 0 & \text{sinon} \end{cases}$$

5. Hybride mixte : FC et MoleTrust proposent un score de similarité entre deux utilisateurs, cette méthode sélectionne la valeur maximale entre les deux :

$$simil(u_a, u_b) = \max(simil_{FC}(u_a, u_b), simil_{Mole}(u_a, u_b))$$

6. Hybride probabiliste : L'intérêt de cette méthode est qu'elle favorise les utilisateurs qui sont à la fois similaires et fiables à ceux qui sont seulement similaires ou seulement fiables.

$$simil(u_a, u_b) = 1 - (1 - simil_{FC}(u_a, u_b)) * (1 - simil_{Mole}(u_a, u_b))$$

en sachant que la similarité de Pearson est normalisée pour cette méthode.

7. Hybride alternance (FC, Mole) : Cette stratégie utilise FC comme méthode de base ; quand FC échoue à calculer un score pour un utilisateur donné, on alterne avec MoleTrust pour calculer son score :

$$simil(u_a, u_b) = \begin{cases} simil_{FC}(u_a, u_b) & \text{si } simil_{FC}(u_a, u_b) \neq nul \\ simil_{Mole}(u_a, u_b) & \text{si } simil_{FC}(u_a, u_b) = nul \end{cases}$$

8. Hybride alternance (Mole, FC) : Ici on inverse les rôles par rapport à la stratégie précédente. MoleTrust est consulté en premier, et le FC est utilisé comme méthode de secours en cas d'échec de MoleTrust :

$$simil(u_a, u_b) = \begin{cases} simil_{Mole}(u_a, u_b) & \text{si } simil_{Mole}(u_a, u_b) \neq nul \\ simil_{FC}(u_a, u_b) & \text{si } simil_{Mole}(u_a, u_b) = nul \end{cases}$$

9. Hybride pondérée : Cette stratégie combine les scores de FC et MoleTrust, elle considère une somme pondérée des deux scores :

$$simil(u_a, u_b) = \alpha \times simil_{FC}(u_a, u_b) + (1 - \alpha) \times simil_{Mole}(u_a, u_b)$$

où $\alpha \in [0, 1]$.

Cette stratégie est sensible à la valeur de α . Nous l'appliquons avec cinq valeurs différentes de α , ce seront (0,1 ; 0,3 ; 0,5 ; 0,7 ; 0,9)

Cela fait un total de 13 méthodes à comparer. L'objectif de ces méthodes est de prédire les notes que les utilisateurs vont fournir aux items.

Pour réaliser le test, nous divisons l'ensemble des notes utilisateur-item de la base de données en deux parties. 80% pour l'apprentissage et 20% pour le test de performance. La répartition des notes dans les deux parties est aléatoire, mais elle respecte le pourcentage par utilisateur. Autrement dit, les notes de chaque utilisateur sont divisées dans les deux parties (apprentissage, test) en respectant les proportions (80% et 20%).

1.4.2 Méthode d'évaluation

Ce qu'on demande aux algorithmes de recommandation dans ce contexte se résume à prédire les valeurs de notes utilisateur-item le plus précisément possible pour le maximum de notes. C'est pourquoi notre évaluation comprend deux volets : la précision et la couverture. Dans l'application présentée (Epinions), l'algorithme utilise les données d'apprentissage afin de prédire les notes de la partie test. Les notes de la partie test sont les valeurs réelles qu'on va utiliser comme référence pour évaluer la prédiction.

Précision Une évaluation simple et efficace de la précision peut être faite en calculant l'erreur moyenne absolue (MAE) (section 1.2.1). L'erreur est résumée par la différence entre la valeur prédite et la valeur réelle.

L'évaluation de la précision dans notre application présente trois aspects.

- La précision de la prédiction générale. Cela est réalisé en appliquant MAE sur toutes les valeurs de la partie test du corpus. L'objectif est de mesurer la capacité de l'algorithme à prédire les notes de tous les utilisateurs.
- La précision par utilisateur. Il s'agit de calculer la MAE par utilisateur (UMAE), et ensuite de calculer la moyenne de ces valeurs. L'importance de cette mesure vient du fait qu'elle est capable de détecter la distribution des recommandations sur l'ensemble des utilisateurs.

Par conséquent, elle permet de distinguer les utilisateurs satisfaits/non-satisfaits. Cette mesure se situe au cœur de notre évaluation, parce que c'est à travers celle-ci qu'on verra la fluctuation de performance par utilisateur.

- La précision de recommandation (HMAE). La différence avec le premier aspect (précision de la prédiction) est que nous nous focalisons ici seulement sur les items recommandables. En effet, l'utilisateur va recevoir une liste limitée aux meilleurs produits qu'il devrait apprécier, il ne verra donc pas les autres prédictions du système.

Nous considérons que l'utilisateur s'intéresse à l'item s'il lui fournit une note égale à 4 ou 5. Ainsi, si le système prédit la valeur 3 alors que la valeur réelle est 1, ce ne sera pas considéré comme une erreur significative parce que l'item est, de fait, non recommandable. En revanche, si la valeur réelle était 5, l'erreur serait grave parce qu'un item à recommander serait éliminé de la liste des recommandations.

Couverture L'importance de cette mesure vient du fait qu'elle donne du sens à la précision. Quand un SR fournit des recommandations très précises à peu d'utilisateurs, il a un bon score de précision, mais en réalité, il perd beaucoup de ses utilisateurs. La couverture sert à savoir qui profite de la précision du système.

Comme la précision, la couverture a trois aspects correspondant à ceux de la précision.

- La couverture de la prédiction s'écrit comme le rapport entre le nombre total de notes prédites sur le nombre total de notes réelles dans le corpus de test, indépendamment des valeurs numériques de ces notes.
- La couverture d'utilisateurs correspond au nombre d'utilisateurs ayant reçu au moins une fois des recommandations divisé par le nombre total d'utilisateurs.
- La couverture de la recommandation (ou la couverture des hautes valeurs) correspond au rapport entre le nombre de notes prédites où la valeur réelle est égale à 4 ou 5, et le nombre total de valeurs hautes.

Nous allons accompagner les mesures de précision avec celles de couverture pour avoir un retour sur la répartition de la précision sur les utilisateurs et par conséquent, caractériser la fluctuation de performance.

1.4.3 Analyse des résultats

La stratégie de l'hybridation pondérée est sensible au paramètre (α). Pour régler ce paramètre, nous l'associons à 5 valeurs comprises entre 0,1 et 0,9 avec un pas de 0,2. Le tableau 1.1 illustre les variations de MAE selon les valeurs de α . On constate que la variation est minime. Pour ce qui concerne la couverture, elle est stable pour toutes les valeurs de α , ce qui est normal car changer α affecte la valeur de la prédiction, mais pas la capacité à prédire. Néanmoins, nous allons choisir $\alpha = 0,3$ dans nos prochaines comparaisons, car il fournit le meilleur score de MAE.

α	0,1	0,3	0,5	0,7	0,9
MAE	0,822	0,821	0,8212	0,823	0,829

TABLE 1.1 – Les valeurs MAE dans l'hybridation pondérée selon plusieurs valeurs du paramètre α

Le tableau 1.2 contient les valeurs des six mesures de qualité pour toutes les méthodes, rassemblées par couple (précision, couverture), chaque couple mesure la performance selon la prédiction, la recommandation et la prédiction par utilisateur.

Méthode	MAE	Couverture de prédiction	HMAE	Couverture des valeurs hautes	UMAE	Couverture d'utilisateurs
Filtrage Collaboratif	0,840	61,15%	0,636	44,44%	0,823	47,46%
MoleTrust	0,817	69,28%	0,619	51,12%	0,808	52,21%
Cascade (FC,Mole)	0,832	53,19%	0,626	38,57%	0,891	36,78%
Cascade (Mole,FC)	0,836	53,44%	0,634	38,94%	0,813	36,97%
Mixte	0,821	76,34%	0,622	56,43%	0,812	62,15%
Probabiliste	0,821	76,31%	0,621	56,44%	0,812	62,07%
Alternance (Mole,FC)	0,822	76,38%	0,622	56,19%	0,816	61,96%
Alternance (FC,Mole)	0,822	76,38%	0,623	56,44%	0,815	62,06%
Pondérée ($\alpha = 0,3$)	0,8210	76,38%	0,621	56,42%	0,812	62,22%

TABLE 1.2 – Les trois formes de MAE et de couverture

Certaines méthodes hybrides ont une performance similaire. Nous pouvons identifier deux catégories. La première contient les méthodes : mixte, probabiliste, en alternance, et pondérée. La deuxième contient les deux méthodes cascades. Dans l'analyse suivante, quand nous parlons des méthodes hybrides, cela concerne uniquement la première catégorie, et pour évoquer la deuxième, nous utilisons le terme "méthodes cascades".

Précision de prédiction (MAE) MoleTrust réalise le meilleur score de MAE (0,817). Les méthodes hybrides réalisent des scores un peu plus élevés (moins bons $MAE \simeq 0,82$). L'écart entre les deux scores est faible. Cela signifie que la perte de précision avec les méthodes hybrides n'est pas significative.

Les méthodes cascades se placent derrière les méthodes hybrides en termes de précision de la prédiction avec un $MAE \simeq 0,83$. Fondamentalement, les cascades utilisent le score de confiance entre deux utilisateurs ayant un comportement similaire, ou le score de similarité quand un lien de confiance existe entre les deux utilisateurs. Autrement dit, elles utilisent la similarité des amis fiables, ou la confiance des utilisateurs similaires. Les méthodes hybrides acceptent que le similaire soit un voisin de l'utilisateur actif, ou qu'il soit son ami, alors que pour les cascades, le similaire doit être les deux à la fois. Si les cascades sont moins précises que les méthodes hybrides et la méthode à base de confiance, c'est parce qu'il n'est pas toujours évident que l'utilisateur fasse confiance à ses voisins, ou qu'il ait un comportement similaire à ceux à qui il fait confiance.

Le filtrage collaboratif est le moins performant avec un $MAE = 0,84$. Il a une précision inférieure à celle de l'approche à base de confiance. Cela rejoint les résultats de Massa sur les mêmes données [Massa and Avesani2007].

La conclusion est que les méthodes hybrides causent une légère baisse de précision par rapport à l'approche à base de confiance mais elles restent meilleures que le filtrage collaboratif.

Couverture de prédiction Les écarts entre les algorithmes sur le plan de la couverture sont plus importants que ceux du point de vue de la précision. Les méthodes hybrides surpassent largement les autres méthodes. Elles prédisent à peu près 7% de plus de notes que MoleTrust et près de 15% de plus que le FC. Cela est expliqué par le fait que les méthodes hybrides considèrent l'utilisateur similaire s'il est voisin ou ami, ce qui leur permet de générer des recommandations

pour les utilisateurs qui n'ont pas d'amis s'ils ont des voisins, et pour ceux qui n'ont pas de voisins s'ils ont des amis. Cette stratégie des méthodes hybrides s'oppose à celle des méthodes cascades. En effet, ces dernières intègrent seulement les utilisateurs qui sont à la fois amis et voisins dans leurs calculs. Il n'est pas toujours évident de trouver des utilisateurs qui remplissent les deux conditions à la fois, ce qui explique la faible performance de ces méthodes.

La figure 1.3 illustre un résumé des valeurs de MAE, et de couverture de prédictions des méthodes utilisées. Il est clair que les méthodes hybrides apportent une augmentation importante de couverture sans une notable perte en précision.

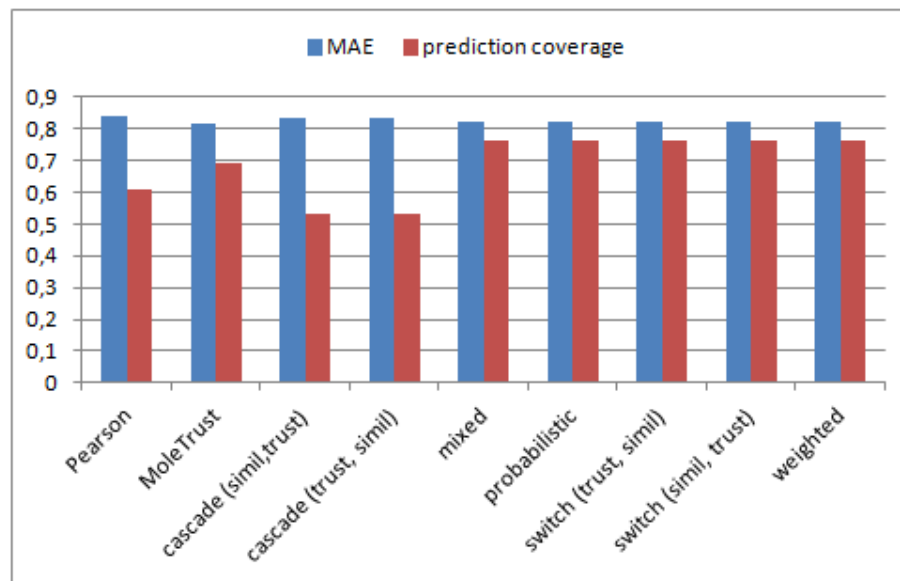


FIGURE 1.3 – MAE

HMAE Les résultats de cette mesure suivent la même tendance que MAE. Le classement des méthodes est le même. Comme dans MAE, les écarts entre les méthodes ne sont pas très importants. MoleTrust reste la méthode la plus précise avec $HMAE = 0,619$. En deuxième position, les méthodes hybrides qui réalisent des HMAE entre 0,621 et 0,623. L'écart entre MoleTrust et les méthodes hybrides est encore minime et n'est pas significatif. L'écart devient "plus" significatif avec les méthodes cascades et le FC qui ont des HAME entre 0,626 et 0,636. Ces résultats nous permettent de conclure que les méthodes hybrides assurent une précision acceptable au niveau de la recommandation par rapport à MoleTrust.

Couverture des valeurs hautes La figure 1.4 montre que les méthodes hybrides améliorent la capacité à recommander des items d'environ 5% par rapport à MoleTrust et 12% par rapport au FC. Par conséquent, les méthodes hybride améliorent la couverture des recommandations, et pas seulement la couverture des prédictions. Cette amélioration est importante car elle garantit que les méthodes hybrides assurent plus de "bonnes recommandations" pour l'ensemble d'utilisateurs.

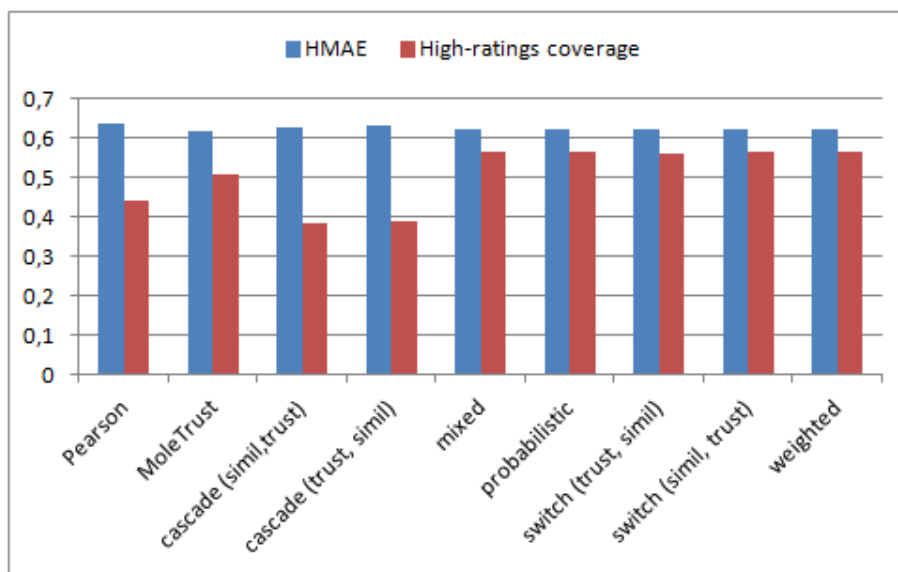


FIGURE 1.4 – HMAE

UMAE La Figure 1.5 montre les valeurs de UMAE pour toutes les méthodes. Il n’y a pas de changement important dans le classement des méthodes. Les méthodes hybrides ont une précision un peu moins bonne que celle de la méthode à base de confiance et meilleure que le FC. Ce résultat est important pour montrer que les méthodes hybrides n’affectent pas la satisfaction des utilisateurs, car elles ne causent pas une sérieuse chute de précision par utilisateur.

Couverture d’utilisateurs Les méthodes hybrides recommandent près de 10% de plus d’utilisateurs par rapport à MoleTrust et 15% de plus par rapport à FC. Ce ratio n’est pas négligeable. Il montre que l’amélioration de couverture est répartie sur tous les utilisateurs et n’est pas concentrée sur une petite communauté. Par conséquent, les méthodes hybrides, comparés à MoleTrust, n’améliorent pas la précision par utilisateur, mais elles augmentent le nombre d’utilisateurs profitant de cette précision.

Bilan

Notre bilan est homogène pour les trois aspects, à savoir, la prédiction, la recommandation et les utilisateurs. Les méthodes hybrides améliorent nettement la couverture, tout en assurant un niveau acceptable de précision, car la baisse de précision par rapport à MoleTrust n’est pas significative.

L’augmentation de la couverture d’utilisateurs montre que les méthodes hybrides réduisent la fluctuation, et qu’elles s’adaptent au comportement de l’utilisateur. Elles lui laissent le choix de noter des items ou des utilisateurs, tout en restant capable de lui générer des recommandations.

Les méthodes hybrides sont donc capables de générer des recommandations à une plus grande variété d’utilisateurs, ce qui permet d’attirer plus d’utilisateurs vers le SR. Ceci constitue un objectif capital pour le SR.

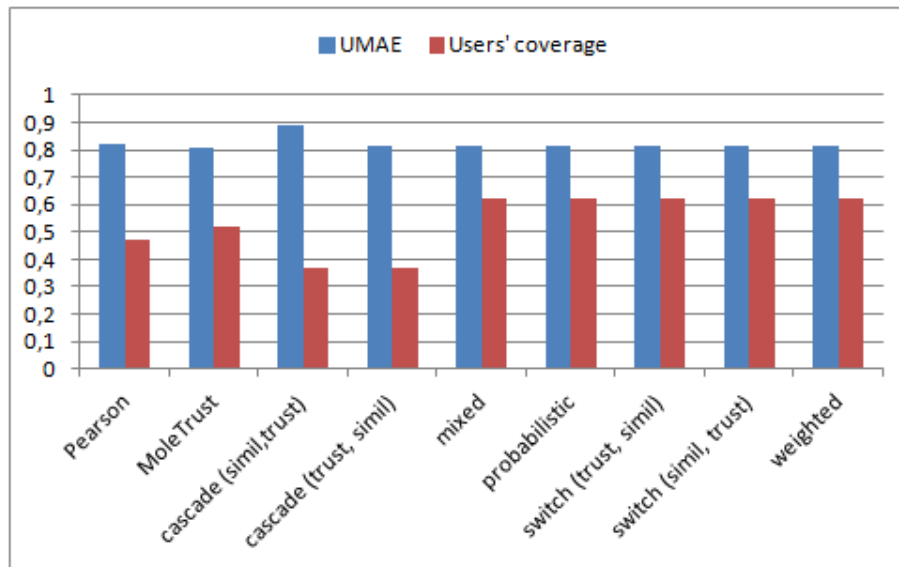


FIGURE 1.5 – UMAE

1.5 Analyse du comportement des utilisateurs

Cette étape consiste à analyser de plus près les résultats obtenus à l'étape précédente, pour essayer d'expliquer la fluctuation des recommandations. Pour ce faire, nous allons souligner cinq caractéristiques du comportement de l'utilisateur, et étudier leurs relations avec la qualité des recommandations rendues par les SR à l'étape précédente.

Les méthodes hybrides ont des performances assez similaires entre elles. Par conséquent, nous allons intégrer seulement une représentante de cette catégorie dans les étapes suivantes. Ce sera la méthode pondérée avec $\alpha = 0,3$, car elle a les meilleurs scores. La performance des méthodes cascades est la moins bonne, surtout pour ce qui concerne la couverture. C'est pourquoi elles seront éliminées lors de l'analyse suivante.

1.5.1 Les caractéristiques du comportement de l'utilisateur

Les caractéristiques que nous présentons ici examinent les notes que l'utilisateur attribue aux items (nombre de notes, la popularité et la déviation de l'utilisateur), et aux autres utilisateurs (nombre d'amis fiables et leur réputation). Elles retracent les aspects quantitatifs (nombre de notes, nombre d'amis fiables), et qualitatifs de ces notes (la popularité des items notés, la déviation de l'utilisateur et la réputation de ses amis fiables).

Dans les sections suivantes, nous illustrons les relations entre chacune des caractéristiques et la valeur UMAE de l'utilisateur, en essayant d'expliquer l'impact de cette caractéristique sur la performance du SR.

Pour chaque utilisateur, nous calculons un score par caractéristique. Chaque caractéristique a une plage de variation des scores. Nous découpons cette plage en 20 groupes. Le découpage est fait de telle sorte que chaque groupe contient environ 5% d'utilisateurs. Ce qui fait que les utilisateurs d'un groupe couvrent une sous-plage de la plage initiale. Le groupe est représenté par la moyenne des UMAE de ses utilisateurs. L'intérêt de cette étape de catégorisation est de réduire la représentation de l'ensemble des utilisateurs pour faciliter la visualisation des courbes.

1.5.1.1 Nombre de notes par utilisateur (NNU)

Cette caractéristique représente un simple comptage des items notés par l'utilisateur. Elle est souvent la seule à être prise en compte lors de l'analyse du comportement des utilisateurs dans l'état de l'art. De notre point de vue, NNU est importante mais ne peut pas résumer à *elle seule* le comportement de l'utilisateur. La figure 1.6 illustre le rapport entre NNU et la UMAE. Il est clair que la UMAE a tendance à baisser avec l'augmentation du NNU moyen sur l'ensemble des utilisateurs. Certes, cette tendance est importante, mais ne dévoile pas tout sur la satisfaction des utilisateurs.

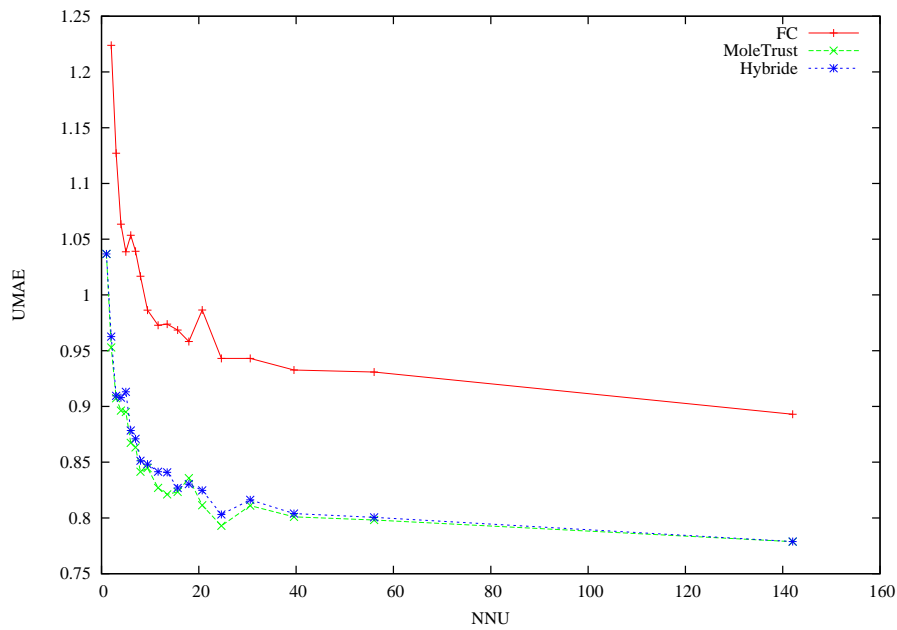


FIGURE 1.6 – Nombre de notes des utilisateurs (NNU) et UMAE

L'amélioration de la qualité de recommandations avec l'augmentation du nombre de notes n'est pas systématique. Pour indiquer l'insuffisance du NNU, nous considérons quatre niveaux de satisfaction d'utilisateur selon la valeur d'UMAe :

1. Très satisfait : La valeur d'UMAe est inférieure ou égale à 0,5.
2. Satisfait : La valeur d'UMAe est supérieure à 0,5 et inférieure ou égale à 1.
3. Insatisfait : La valeur d'UMAe est supérieure à 1.
4. Complètement insatisfait : Absence de valeur d'UMAe.

La figure 1.7 illustre le pourcentage d'utilisateurs pour chaque niveau de satisfaction par catégorie pour le FC. Nous avons choisi le FC pour cette illustration parce qu'elle est l'approche la plus dépendante des notes d'utilisateurs.

La première catégorie contient les utilisateurs qui ont noté 0 item. Ces utilisateurs sont à 100% "complètement insatisfaits", le même résultat est visible dans la deuxième catégorie, où le nombre de notes par utilisateur est égal à 1. Nous commençons à trouver des utilisateurs qui reçoivent des recommandations à partir de la troisième catégorie, là où chaque utilisateur a fourni exactement 2 notes. Il est vrai que la courbe de "complètement insatisfaits" fait une considérable chute avec l'augmentation du nombre de notes, mais nous remarquons qu'elle n'atteint pas le zéro même

dans des catégories où le nombre de notes par utilisateur dépasse les 70 notes. Il est vraiment rare qu'un utilisateur qui dépasse les 70 notes ne reçoive toujours pas de recommandation de la part d'un système qui s'appuie sur les notes.

La courbe de "insatisfait" remonte légèrement avec l'augmentation du nombre de notes et redescend à la fin.

La courbe de "satisfait" a un comportement normal et suit une relation linéaire avec l'augmentation du nombre de notes par utilisateur.

La courbe de "très satisfait" affiche un tracé inattendu, elle croît au départ (ce qui est normal) jusqu'à atteindre sa meilleure valeur en dépassant le 20% dans la catégorie de 7 notes, avant qu'elle ne prenne un chemin décroissant jusqu'à près de 5% dans la catégorie de plus de 70 notes.

Nous concluons de cette figure qu'il est vrai que le pourcentage des utilisateurs satisfaits augmente avec l'augmentation des nombres de notes par utilisateur, mais il est clair aussi que certains utilisateurs arrivent à obtenir de bonnes recommandations avec très peu de notes, alors que d'autres attribuent beaucoup de notes sans être satisfaits par leurs recommandations. Cela prouve que le nombre de notes attribuées ne peut pas seul garantir la satisfaction d'un utilisateur, et que la fluctuation du FC est liée à d'autres caractéristiques.

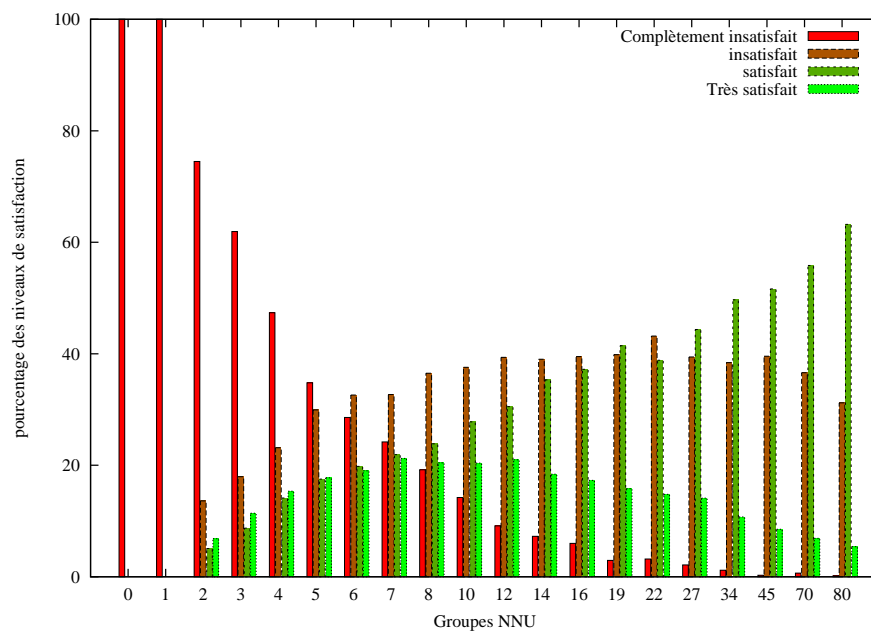


FIGURE 1.7 – La satisfaction des utilisateurs par catégorie

L'importance de la figure 1.7 vient du fait qu'elle montre que la tendance observée sur la figure 1.6 n'est pas une règle universelle, et qu'elle n'est pas suffisante pour décrire le comportement de l'utilisateur. Pour ce qui concerne les autres caractéristiques, nous savons déjà qu'aucune d'entre elles n'est suffisante pour décrire le comportement d'un utilisateur. Par conséquent, nous nous focalisons seulement sur leurs tendances.

1.5.1.2 La moyenne de popularité des items notés (PIN)

Dans [Steck2011], l'auteur considère les items populaires comme étant les items les plus appréciés par l'ensemble des utilisateurs. Il associe la popularité au nombre de notes de valeurs

hautes que l’item reçoit. Plus l’item reçoit des notes hautes plus il est populaire. L’auteur améliore la performance du FC en l’incitant à recommander les items les plus populaires sous prétexte qu’ils ont plus de chance d’être appréciés.

Notre définition de la popularité est différente. Nous considérons qu’un item populaire n’est pas forcément appréciable. Nous associons plutôt la popularité d’un item au nombre total de notes qu’il reçoit. Formellement :

$$pop_i = \text{card}(u|r(u,i) \neq \text{null})$$

où pop_i est la popularité de l’item i .

$r(u,i)$ est la note fournie par l’utilisateur u à l’item i .

Noter un item en FC aide à établir un lien de similarité avec les utilisateurs qui ont noté cet item. Si l’item est noté par beaucoup d’utilisateurs, il peut rendre l’utilisateur qui le note plus connecté. Cela influence la qualité des recommandations qu’il recevra. Nous associons à chaque utilisateur la moyenne de popularité des items qu’il note :

$$PIN(u) = \frac{\sum_{i \in U_r} pop_i}{\text{card}(U_r)}$$

où $PIN(u)$ est la popularité moyenne des items notés par un utilisateur.

U_r est l’ensemble des items notés par l’utilisateur u .

Le tableau 1.3 représente un exemple d’une matrice de notes dans un SR. La popularité de i_0 est égale à 5 parce qu’il est noté par 5 utilisateurs, et la popularité de i_4 est égale à 1. Si un utilisateur note i_4 (un item peu populaire), il aura seulement u_0 comme voisin. Mais s’il note i_0 (un item très populaire), il sera voisin avec tous les utilisateurs sauf u_4 .

Par exemple : L’utilisateur u_3 a noté deux items i_0 et i_2 dont la popularité est 5 et 4. Par conséquent :

$$PIN(u_3) = \frac{5 + 4}{2} = 4,5$$

	i_0	i_1	i_2	i_3	i_4	i_5
u_0	5		4	2	4	
u_1	4	5		2		1
u_2	1	5				3
u_3	2		5			
u_4			1	5		
u_5	5	4	5	1		

TABLE 1.3 – Exemple de matrice de notes

La figure 1.8, illustre la relation entre PIN et UMAE.

Les utilisateurs sont ordonnés par leurs valeurs ascendantes de PIN, ensuite ils sont répartis en 20 groupes. Chaque groupe contient environ 5% de la population. Rappelons que plus UMAE est faible, plus précises sont les recommandations, et plus l’utilisateur est considéré satisfait.

La courbe a une légère tendance à décroître avec l’augmentation de PIN au début, mais elle croît à nouveau pour des valeurs PIN supérieures à 100. L’explication de ce phénomène est que l’utilisateur qui note seulement des items très populaires risque de subir un effet de *sur-voisinage*. C’est-à-dire qu’il aura un nombre excessif de voisins de différents types, ce qui perturbe le SR qui n’arrive plus à bien déterminer les préférences de l’utilisateur. Ce résultat montre que les items très populaires sont généralement moins discriminants et peu informatifs pour le SR. Ce

résultat rejoint celui de plusieurs études qui ont tendance à montrer des effets négatifs des items populaires sur la performance du SR [Cremonesi *et al.*2010, Kouki2012].

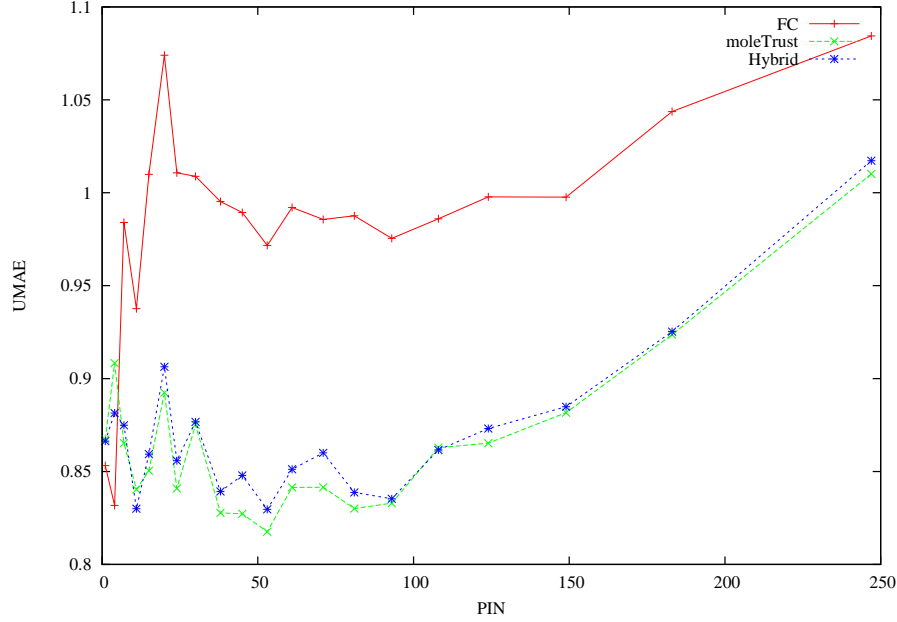


FIGURE 1.8 – La popularité des items (PIN)

1.5.1.3 Les utilisateurs déviants (coefficient de déviation CODEV)

Nous caractérisons ici les utilisateurs ayant des goûts particuliers. Nous mesurons la distance entre l'orientation de l'utilisateur et l'orientation globale de la communauté.

Formellement, nous commençons par calculer la note moyenne de l'item, ensuite la différence entre cette moyenne et la note attribuée par l'utilisateur actif. Le coefficient de déviation de l'utilisateur est la moyenne des différences entre ses notes et la note moyenne des items qu'il a notés.

$$CODEV|u| = \frac{\sum_{i \in U_r} |r(u, i) - \bar{r}_i|}{card(U_r)} \quad (1.1)$$

$r(u, i)$: est la note attribuée par l'utilisateur u à l'item i . \bar{r}_i : est la note moyenne de l'item i .

La figure 1.9 a la même structure que la figure 1.8 avec une différence, c'est que la catégorisation des utilisateurs se fait en fonction de leur CODEV.

En ce qui concerne la figure 1.9, UMAE est relativement très élevé pour les utilisateurs avec un grand CODEV, ce qui signifie que les utilisateurs dont les notations sont proches des notes moyennes des items reçoivent des recommandations plus précises que ceux dont les notations sont opposées à la tendance de la communauté. En effet, la relation n'est pas linéaire entre les deux. Les utilisateurs les plus satisfaits sont ceux qui ont un CODEV compris entre 0,4 et 0,6, donc un avis peu différent de l'orientation de la communauté.

La figure 1.10 illustre la distribution du nombre moyen de notes dans les groupes de CODEV. Les utilisateurs dans les groupes à CODEV élevé (plus de 1,4) et les groupes à faible CODEV (moins de 0,4) ont presque le même nombre de notes. En regardant ces mêmes groupes dans la

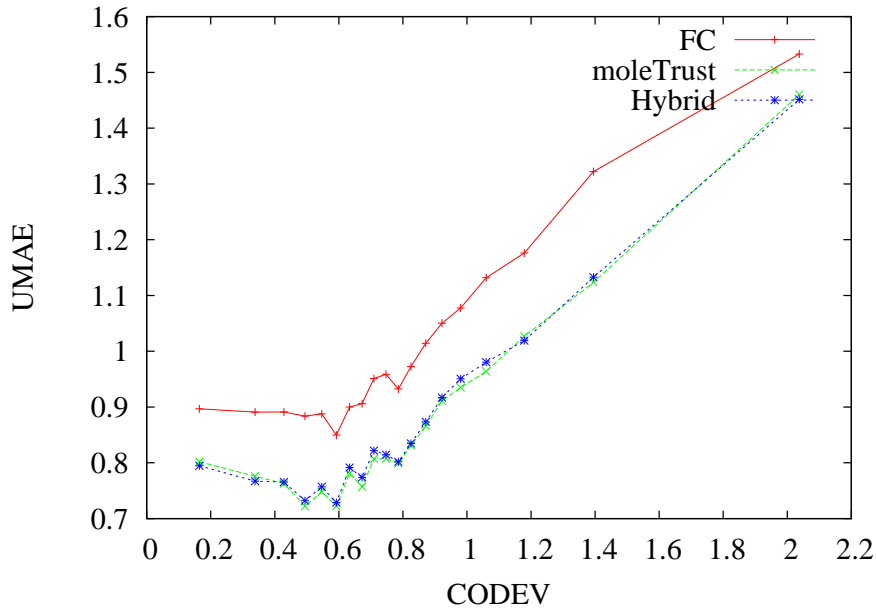


FIGURE 1.9 – codev et UMAE

figure 1.9, nous remarquons qu'ils sont sur deux extrêmes de UMAE. Cela montre l'existence de deux groupes d'utilisateurs en situation de démarrage à froid (utilisateurs qui notent peu d'items). Un des deux groupes est largement plus satisfait que l'autre, parce que ses membres ont un CODEV plus bas. Il est donc évident que, pour les utilisateurs en situation de démarrage à froid, CODEV est une caractéristique discriminante de la performance du SR, plutôt que le nombre de notes.

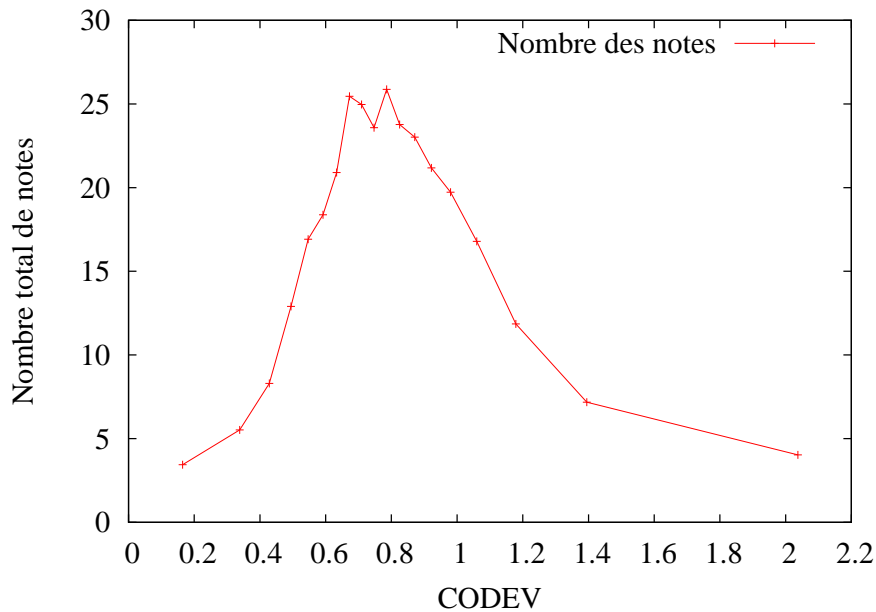


FIGURE 1.10 – CODEV et le nombre de notes par utilisateur

1.5.1.4 Les utilisateurs de nature confiante/réservée UCR

Cette caractéristique est quantitative. Elle relie le nombre d'amis fiables d'un utilisateur avec son UMAE. La question ici est : est-ce qu'en faisant plus confiance à plus d'utilisateurs, on arrive à être plus satisfait ?

Comme le montre la figure 1.11, faire plus confiance est toujours profitable dans ce contexte. La précision de recommandation a une tendance à s'affiner avec l'augmentation de UCR. Mais cette relation n'est pas parfaitement linéaire. La chute de UMAE est plus importante pour les utilisateurs ayant peu d'amis. Le système hybride réalise un UMAE de 1,04 pour les utilisateurs avec un seul ami fiable, et 0,84 pour les utilisateurs avec 10 amis fiables, soit une amélioration de 0,20 pour 9 amis de plus. Le même système atteint un UMAE de 0,78 pour 110 amis, c'est-à-dire une amélioration de 0,06 pour 100 amis de plus. Le profit des nouvelles relations de confiance est plus important pour les utilisateurs qui en ont peu, et devient moins important pour ceux qui ont déjà beaucoup de relations.

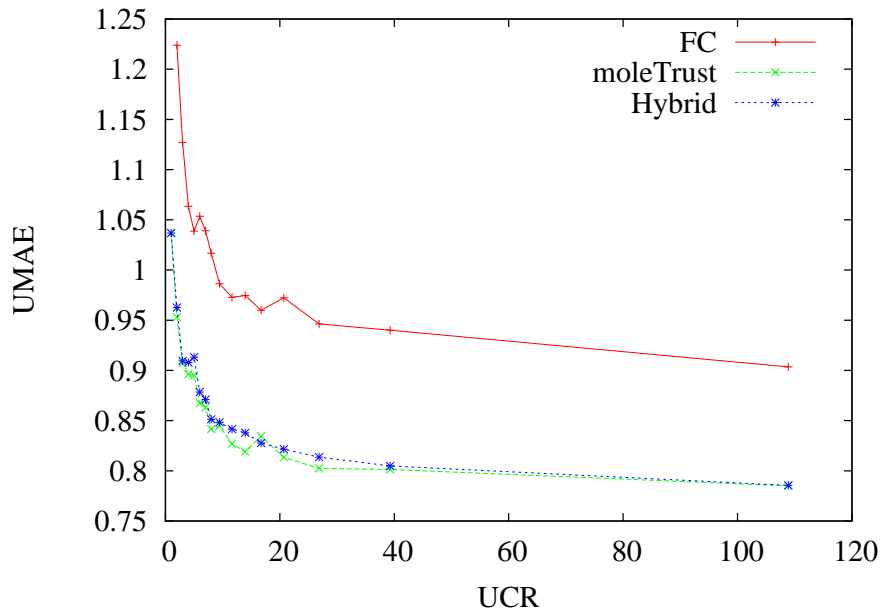


FIGURE 1.11 – Les utilisateurs de nature confiante/réservée

1.5.1.5 La réputation des amis crédibles (RAC)

La caractéristique précédente s'intéresse au fait que l'utilisateur fasse ou pas confiance à beaucoup d'utilisateurs. La caractéristique actuelle s'intéresse à la réputation des utilisateurs auxquels il fait confiance.

Au vu du type des relations de confiance (valeurs binaires), nous proposons une mesure simple de réputation. La réputation d'un utilisateur est le nombre d'utilisateurs qui lui font confiance.

$$Rep(u) = card(S_u) \quad (1.2)$$

où : S_u est le sous-ensemble des utilisateurs qui ont fait confiance à u .

RAC pour un utilisateur u est la moyenne des réputations des utilisateurs auxquels u fait confiance.

$$RAC(u) = \frac{\sum_{u_i \in C_u} Rep(u_i)}{card(C_u)} \quad (1.3)$$

où C_u est le sous-ensemble d'utilisateurs auxquels u fait confiance.

La courbe de la figure 1.12 montre que UMAE est relativement élevé lorsque la moyenne des réputations est très faible (moins de 10), alors qu'il est quasiment stable par la suite. Cela montre que, dans ce contexte, il suffit que l'utilisateur montre des intentions positives à quelques utilisateurs pour avoir une légitimité suffisante dans la communauté, et d'un autre côté, il est suffisant pour un utilisateur de trouver une dizaine d'utilisateurs qu'il peut qualifier comme crédibles pour qu'il puisse recevoir de bonnes recommandations.

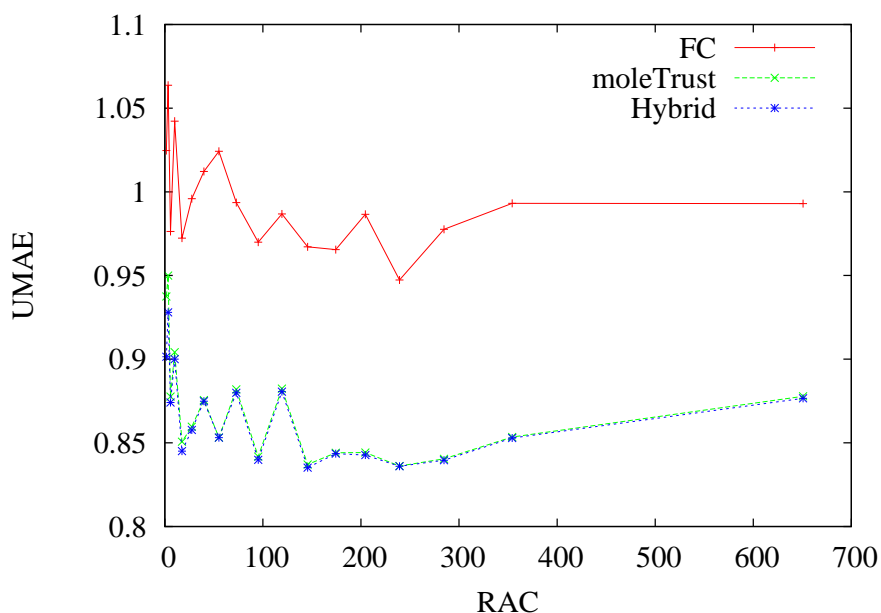


FIGURE 1.12 – Réputation des amis crédibles (RAC) et UMAE

1.5.2 Classification d'utilisateurs

L'étape précédente a montré l'influence des caractéristiques qualitatives et quantitatives du comportement de l'utilisateur sur la qualité des recommandations qu'il reçoit. Nous avons montré la tendance de chacune d'entre elles et sa relation avec la précision de recommandation. La question suivante concerne la possible influence composée de ces caractéristiques.

Dans l'étape actuelle, nous représentons l'utilisateur par un vecteur de valeurs des caractéristiques définies dans l'étape précédente. La classification non supervisée permet de regrouper les utilisateurs selon leur comportement similaire, de détecter l'impact de ce comportement sur la performance de chaque approche de recommandation, et de trouver si une approche est mieux adaptée qu'une autre pour ce type de comportement.

Nous utilisons l'algorithme Kmeans [Alsabti *et al.*1997] pour la classification. C'est un algorithme de classification non-supervisé connu pour son efficacité dans beaucoup de contextes. Le seul inconvénient est qu'il faut lui préciser K le nombre désiré de classes. Afin de rester ouvert sur

toutes sortes de comportements qui peuvent être identifiées, nous n'allons pas limiter le nombre de classes qui peuvent exister. Pour cela, nous allons appliquer l'algorithme Kmeans en faisant varier le nombre de classes, et nous allons utiliser une mesure de qualité pour trouver le nombre optimal de classes ; la mesure Davies–Bouldin (DB) [Davies and Bouldin1979,Halkidi *et al.*2001] qui est une mesure de la qualité de classification indépendante de l'algorithme utilisé.

L'idée de cette mesure est de privilégier une classification qui à la fois maximise les distances entre les barycentres des classes, et minimise les distances entre les membres d'une même classe. En premier lieu, on calcule la dispersion dans chaque classe :

$$S_i = \sqrt{\frac{1}{T_i} \sum_{j=1}^{T_i} |X_j - A_i|^2}$$

où S_i est la dispersion de la classe i .

T_i représente la taille de la classe i .

A_i correspond au barycentre de la classe i .

X_j est le vecteur représentant le membre j de la classe i .

Ensuite, on calcule la distance euclidienne entre chaque paire de classes :

$$M_{i,j} = \|A_i - A_j\|_p$$

L'étape suivante consiste à calculer $R_{i,j}$ qui représente la qualité de la classification par couple de classes i et j :

$$R_{i,j} = \frac{S_i + S_j}{M_{i,j}}$$

Puis, on calcule D_i pour chaque classe :

$$D_i = \max_{j:i} R_{i,j}$$

L'indice de Davies–Bouldin est égal à la moyenne des valeurs de D_i

$$DB = \frac{1}{N} \sum_{i=1}^N D_i$$

Plus la valeur de DB est petite, meilleure est la classification.

On fait varier K entre 4 et 10. Dans le tableau 1.4, la valeur de DB est minimale pour 8 classes.

No.de classes	4	5	6	7	8	9	10
Davies-Bouldin	1,44	1,35	1,41	1,34	1,28	1,3	1,33

TABLE 1.4 – Davies-Bouldin pour optimiser le nombre de classes

Le tableau 1.5 illustre les résultats de cette classification à 8 classes. La taille de chaque classe est exprimée en nombre d'utilisateurs classés dedans, suivie par le pourcentage de sa taille par rapport au nombre total d'utilisateurs dans ce corpus.

La deuxième colonne du tableau 1.5 fournit les valeurs moyennes des caractéristiques pour l'ensemble du corpus. Les colonnes suivantes illustrent les moyennes par classe. Cela permet

de montrer les caractéristiques de chaque classe afin de les comparer avec les caractéristiques générales du corpus, et ainsi comparer les classes entre elles.

Classe	Corpus	C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7
Taille	32424	3221	765	4458	5547	7909	1674	8324	526
Pourcentage	100%	9,93%	2,36%	13,75%	17,11%	24,39%	5,16%	25,67%	1,62%
Nb notations	20,27	14,8	6,63	34,09	11,12	22,3	4,47	8,98	251,01
PIN	80,86	79,43	453,76	64,67	54,19	88,24	26,66	79,8	44,02
CODEV	0,8	1,26	0,84	0,72	0,42	0,8	2,18	0,81	0,76
UCR	13,86	13,57	9,16	28,26	10,05	16,88	6,7	0,08	136,47
RAC	143,65	109,76	90,13	320,4	82,97	74,02	119,47	1126,51	225,77

TABLE 1.5 – Classification Kmeans avec 8 classes

1.5.2.1 Méthode d'évaluation

Nous nous intéressons à voir comment les méthodes de recommandation réagissent avec chacune des classes. Pour cela nous allons mesurer leur performance par classe. Nous proposons une seule mesure pour la précision et la couverture, une mesure de type F-mesure qui agrège les deux aspects en une seule valeur :

$$F = \frac{2 \times \text{precision} \times \text{rappel}}{\text{precision} + \text{rappel}}$$

En général, la F-mesure est utilisée pour mesurer la performance du système qui doit agir face à des requêtes. Le rappel est le rapport entre le nombre de requêtes auxquelles le système a pu proposer une réponse sur le nombre total de requêtes posées. La précision est le rapport entre le nombre de requêtes auxquelles le système a pu proposer une *bonne* réponse sur le nombre de requêtes auxquelles le système a proposé une réponse.

Dans notre contexte, une requête est de pouvoir calculer des recommandations pour un utilisateur donné. Par conséquent, le rappel est le nombre d'utilisateurs ayant eu des recommandations sur le nombre total d'utilisateurs (même définition que la couverture). La précision mesure la qualité des recommandations. La précision est souvent le résultat d'un jugement binaire sur la justesse de la réponse, dans notre contexte ce jugement est graduel, il est porté par la mesure UMAE. Le seul problème est qu'il n'est pas normalisé. Nous appliquons la formule suivante pour normaliser UMAE :

$$NUMAE = \frac{4 - UMAE}{4}$$

Pour mémoire, la valeur d'UMAE est comprise entre 4 et 0.

Le tableau 1.6 montre les valeurs de couverture, UMAE, et F-mesure pour toutes les classes.

Les deux tableaux 1.5 et 1.6 nous permettent de mener notre analyse sur chaque classe et le comportement qu'il dévoile :

La classe C0 La classe C_0 contient environ 10% de la population ; la particularité de cette classe est la valeur de CODEV élevée avec un score de 1,26. Cela provoque une augmentation de la valeur UMAE (baisse de la précision), qui devrait tirer la valeur de F vers le bas. Néanmoins, la valeur de F pour cette classe est légèrement plus élevée que celle du corpus. Ceci résulte des valeurs élevées de couverture de la classe (par rapport à l'ensemble du corpus). La classe est plus dense et moins dispersée que le corpus entier, les utilisateurs sont plus proches du centre de la classe, ce qui améliore la valeur de la couverture, et par conséquent la valeur de F.

	Classes	Corpus	C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7
Précision	FC	0,75	0,69	0,73	0,77	0,78	0,76	0,64	0,74	0,78
	MoleTrust	0,79	0,74	0,76	0,8	0,81	0,79	0,71	0,78	0,81
	Hybride	0,78	0,74	0,76	0,8	0,81	0,79	0,96	0,77	0,81
Rappel	FC	0,64	0,69	0,66	0,85	0,55	0,82	0,12	0,49	0,99
	MoleTrust	0,69	0,90	0,90	0,98	0,89	0,92	0,76	0,02	1
	Hybride	0,82	0,93	0,92	0,98	0,91	0,96	0,77	0,47	1
F	FC	0,69	0,69	0,70	0,81	0,65	0,79	0,21	0,60	0,88
	MoleTrust	0,73	0,81	0,82	0,88	0,85	0,85	0,74	0,06	0,89
	Hybride	0,80	0,83	0,83	0,88	0,86	0,87	0,74	0,59	0,89

TABLE 1.6 – La performance des recommandeurs par classes

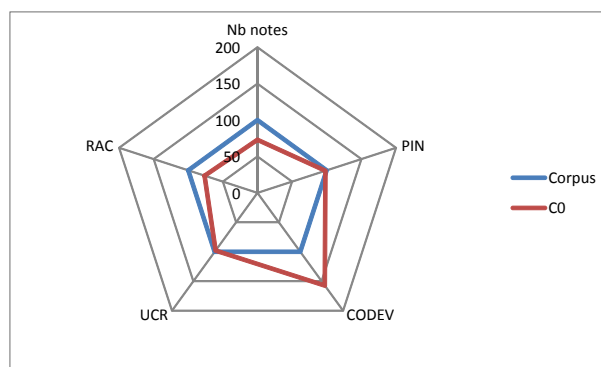


FIGURE 1.13 – La structure de la classe C0

La classe C1 La classe C_1 contient 2,36% des utilisateurs. Ce sont des utilisateurs qui disposent d'un faible NNU (6,6), mais ils notent des items populaires (moyenne de popularité est 453.76). Nous avons montré dans 1.5.1.3 que noter des items populaires a un impact positif pour les utilisateurs en situation de démarrage à froid, et un impact négatif pour les utilisateurs qui notent beaucoup d'items. Les utilisateurs de cette classe sont des utilisateurs en situation de démarrage à froid, ce qui explique la légère amélioration de la performance pour cette classe. En général, le FC améliore légèrement sa performance (1%). L'importance de cette amélioration vient du fait qu'elle est réalisée sur une communauté d'utilisateurs qui notent beaucoup moins que la moyenne générale. Encore une fois, cela prouve la limite d'une analyse basée seulement sur le nombre de notes. MoleTrust améliore un peu son rappel, car il n'est pas affecté par le manque de notes ; la moyenne des notes de confiance est satisfaisante. Cette amélioration lui permet de s'approcher de la performance du système hybride sans la dépasser.

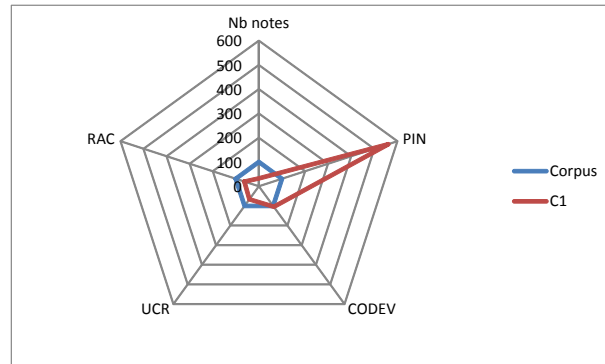


FIGURE 1.14 – La structure de la classe C1

La classe C2 La classe C_2 contient 13,75% des utilisateurs. Nous remarquons une augmentation considérable du NNU. Les utilisateurs de cette classe notent des items un peu moins populaires que la moyenne, leur CODEV est légèrement en-dessous de la moyenne, ils sont de nature confiante, et ils font confiance aux utilisateurs réputés. Ces caractéristiques ont un effet positif pour les trois approches de recommandation, les utilisateurs de cette classe sont faciles à satisfaire. Il est donc normal que les trois SR aient une meilleure précision et un meilleur rappel.

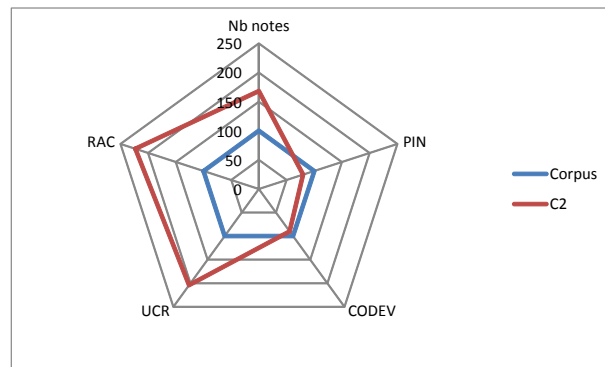


FIGURE 1.15 – La structure de la classe C2

La classe C3 La classe C_3 contient 17% de la population. Les utilisateurs de cette classe notent moins que la moyenne générale. Les items qu'ils notent ne sont pas très populaires, mais leurs notes s'accordent bien avec l'orientation générale de la communauté (CODEV=0,42). Cette dernière caractéristique empêche une baisse de précision pour le FC, elle fait même une légère hausse de précision. Le choix des items moins populaires fait que le FC trouve difficilement des voisins et par conséquent le rappel est en baisse, ainsi que la F-mesure pour CF.

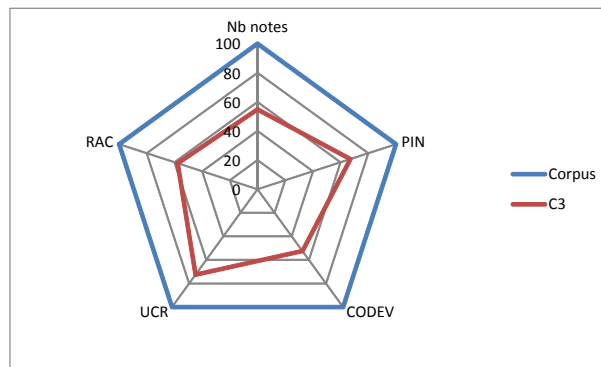


FIGURE 1.16 – La structure de la classe C3

La classe C4 La classe C_4 comporte environ 24% des utilisateurs. Toutes les caractéristiques sont légèrement supérieures à la moyenne obtenue dans le corpus, à l'exception de la réputation d'amis RAC qui est à son niveau le plus bas (74,02). Mais cela n'affecte pas sérieusement la performance du SR car $RAC > 10$. La performance des trois approches est meilleure parce qu'ils profitent de la légère augmentation des quatre autres caractéristiques, alors que l'abaissement de RAC n'a pas un fort impact.

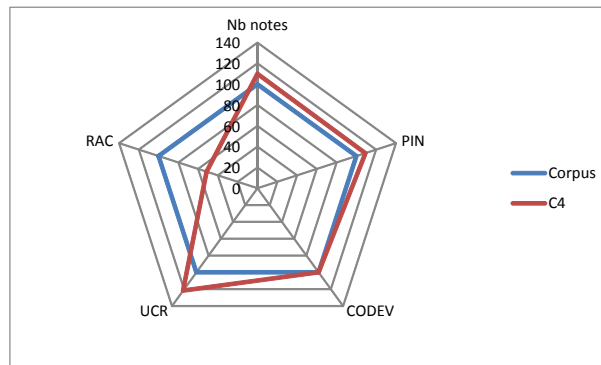


FIGURE 1.17 – La structure de la classe C4

La classe C5 La classe C_5 contient 5,16% des utilisateurs. Ces utilisateurs sont des utilisateurs en situation de démarrage à froid qui notent des items impopulaires, et leurs notes sont nettement loin de l'orientation de la communauté. Cela met le FC à son plus faible niveau, avec la précision la plus basse, et un rappel à seulement 12%. Par conséquent, la valeur de F est très basse, et l'approche hybride repose complètement sur MoleTrust. Cela montre aussi la robustesse du modèle à base de confiance qui reste fonctionnel et performant même avec peu de données (UCF=6,7).

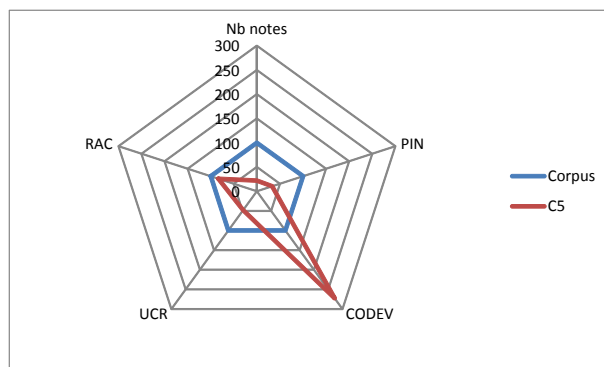


FIGURE 1.18 – La structure de la classe C5

La classe C6 La classe C_6 est la plus grande avec 25% des utilisateurs. Les valeurs des trois premières caractéristiques semblent convenables pour le FC. Les utilisateurs de cette classe font peu de relations de confiance. Ils sont très réservés et ne font confiance qu'aux utilisateurs très réputés. Il est clair qu'ils se focalisent sur la notation des items (près de 9 notes par utilisateur) plutôt que sur l'expression de leur confiance aux autres. Cela affecte fortement la performance de MoleTrust surtout au niveau du rappel. Par conséquent, le FC est le meilleur SR pour cette classe, il dépasse même le système hybride.

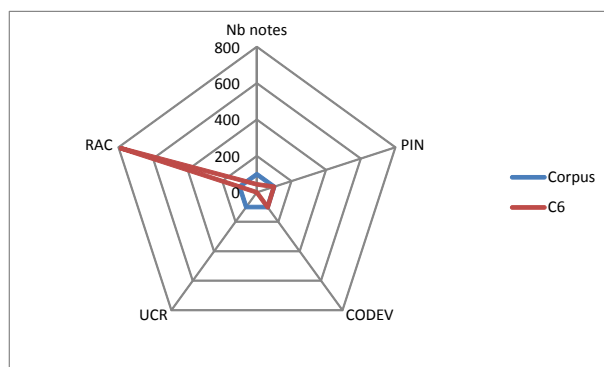


FIGURE 1.19 – La structure de la classe C6

La classe C7 La dernière et la plus petite classe est C_7 , avec 1,62 % des utilisateurs. Les utilisateurs de cette classe ont la moyenne la plus élevée des notes, ils notent des items moyennement populaires, leur CODEV est proche de la moyenne générale. Ils ont un nombre considérable de relations de confiance avec quelques amis bien réputés. Toutes ces valeurs font que cette classe a la meilleure qualité parmi les classes obtenues, et les trois approches réalisent leurs meilleures performances sur cette classe.

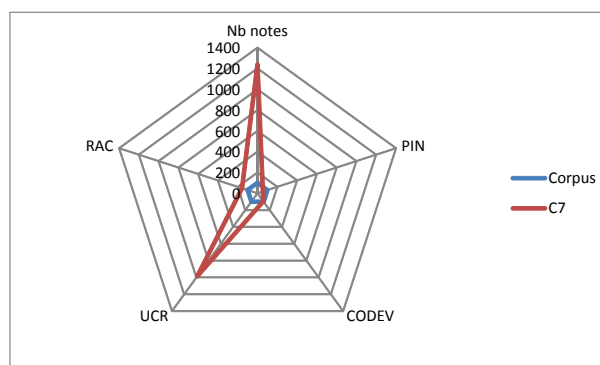


FIGURE 1.20 – La structure de la classe C7

1.6 Conclusion

Dans ce chapitre, nous nous sommes intéressés à la fluctuation de performance dans les systèmes de recommandation. Nous savions déjà que le SRC dépassait le FC au niveau de la performance, mais qu'il ne convenait pas à tout le monde. L'hybridation sert à augmenter la couverture et à élargir le pourcentage d'utilisateurs ayant des recommandations, et diminue la fluctuation.

Afin d'ajuster l'hybridation, nous avons testé plusieurs techniques. Globalement, toutes les techniques améliorent la couverture sauf les cascades ; ces méthodes tablent sur le fait que l'utilisateur fera toujours confiance à ses voisins, ou qu'il se comportera comme ses amis, ce qui n'est pas toujours évident.

Les autres méthodes sont moins exigeantes, car il suffit souvent que le voisin soit ami ou qu'il ait un comportement similaire à celui de l'utilisateur actif pour qu'il soit éligible à la recommandation. Les méthodes hybrides sont plus accommodantes que les cascades, elles améliorent la qualité de recommandation et surtout la couverture, et réduisent la fluctuation.

L'analyse des fluctuations de performances de SR ne prend souvent en compte que les aspects quantitatifs des activités des utilisateurs. Notre analyse va au-delà de cela et intègre les aspects qualitatifs des activités. Cette intégration est fructueuse pour effectuer une analyse plus profonde sur le comportement des utilisateurs afin d'améliorer le service proposé par les modèles de recommandation.

Nous avons proposé trois caractéristiques qualitatives (PIN, CODEV et RAC) qui ont permis de dévoiler des phénomènes comportementaux inexplicables par les caractéristiques quantitatives ; comme l'effet de noter des items populaires/impopulaires, et l'effet de suivre ou pas l'orientation de la communauté.

Après avoir montré l'effet individuel de chaque caractéristique, nous nous sommes intéressés à des effets plus complexes et au croisement des caractéristiques afin d'identifier des comportements globaux existant dans le corpus. La phase de classification permet de réaliser ce but. Grâce à la classification, plusieurs styles de comportements ont été identifiés, les approches de recommandation ont parfois des performances similaires pour certaines classes, des performances opposées pour d'autres. Cela permet de construire un système de recommandation ajustable à plusieurs classes de comportements, et de choisir à chaque fois l'algorithme qui s'accorde le mieux à chaque classe comportementale.

Le modèle d'analyse proposé dans ce chapitre ne se limite pas aux caractéristiques présentées ici, d'autres peuvent être intégrées, cela varie selon le contexte, le type de données, et l'appli-

cation. Il est vrai que certains des phénomènes observés peuvent être propres à ces données, d'autres peuvent apparaître selon les données utilisées. Mais les données utilisées sont devenues un standard dans le domaine, et ont fait l'objet de nombreuses études, ce qui motive leur choix, et justifie l'intérêt des tendances observées.

L'apport positif des relations de confiance dans le domaine de la recommandation est évident. Dans le chapitre suivant nous allons nous focaliser sur ces relations, et nous allons essayer de trouver un cadre pour optimiser leur modélisation afin d'améliorer la performance du SR.

Modèles de confiance à base de la logique subjective

Sommaire

2.1	Motivation	81
2.2	Données et contexte	82
2.3	Confiance locale	84
2.4	Confiance collective	86
2.5	Confiance globale	88
2.5.1	Le traitement des réponses non-acceptées	90
2.6	Mesures de qualité	93
2.7	Résultats et discussions	93
2.7.1	Précision	93
2.7.2	Complexité	94
2.7.3	Résistance face aux attaques malveillantes	95
2.8	Conclusion	96

2.1 Motivation

Pouvoir choisir ses amis, et la propagation de confiance sont les deux arguments qui permettaient à MoleTrust de surpasser la performance du FC. Toutefois, nous sommes convaincus que MoleTrust ne fournit pas une modélisation optimale de la confiance. Nous sommes convaincus aussi que la logique subjective (LS) présente un meilleur cadre de la confiance que MoleTrust, pour les raisons suivantes.

Dans MoleTrust, l'utilisateur ne peut pas exprimer différents degrés de confiance. Il ne peut pas choisir d'avoir des amis plus fiables que d'autres. Cela n'est pas conforme à la confiance dans la vie réelle. La LS offre cette possibilité.

Pour modéliser la confiance entre deux utilisateurs, MoleTrust prend le chemin le plus court et la valeur maximale de la confiance, ce qui ne garantit pas d'être sur le meilleur chemin. La LS prend le chemin le plus sûr, ce que nous trouvons plus réaliste.

MoleTrust ne prend pas en compte la défiance, alors que celle-ci est prise en compte dans la LS.

De plus, la LS offre plus de flexibilité dans la modélisation grâce à ses multiples opérateurs.

Dans ce chapitre, nous proposons trois modèles de confiance à base de la LS. Le premier utilise une confiance locale, où l'utilisateur s'appuie sur ses propres opinions, et consulte celles de ses amis en cas d'absence de la sienne. Le deuxième utilise une confiance collective, où l'utilisateur consulte toujours ses amis pour renforcer son opinion. Le troisième utilise une confiance globale, où l'utilisateur s'appuie sur la réputation du candidat pour décider de coopérer avec lui ou non.

Nous donnons aussi beaucoup d'importance au contexte de la confiance, car nous considérons que la confiance est contextuelle. Malgré l'importance du contexte, ce dernier est rarement pris en compte dans les modèles de la confiance.

Les données utilisées dans le chapitre précédent ne sont pas compatibles avec les modèles étudiés ici, car elles ne contiennent aucune information sur le contexte, et que les notes entre les utilisateurs ne supportent pas la constitution des opinions de la LS. Par conséquent, nous allons utiliser de nouvelles données venant du corpus StackExchange¹⁰.

Ce chapitre est organisé comme suit : nous commençons par détailler les données utilisées. Ensuite, nous présentons les trois modèles proposés. Puis nous examinons les mesures de qualité, avant de présenter les résultats et les conclusions.

2.2 Données et contexte

StackExchange est un réseau de 106 sites web équipés d'une plateforme de question-réponse. Il traite de domaines divers comme la programmation, les mathématiques, la langue anglaise, la photographie, la cuisine, et bien d'autres. En septembre 2013, l'ensemble des sites comptait près de 4 millions d'utilisateurs, 7,1 millions de questions, et 12,9 millions de réponses. Le site le plus connu est celui de programmation Stackoverflow¹¹.

Quand un utilisateur a besoin d'une aide technique, il est invité à choisir le domaine de son problème, et à poser sa question. Il doit associer entre 1 et 5 mots clés à sa question à partir d'un dictionnaire des mots clés propres au domaine. Il reçoit alors plusieurs réponses. Il doit ensuite indiquer la *réponse acceptée*, c'est-à-dire celle qu'il trouve la plus pertinente. Les autres utilisateurs peuvent aussi participer à l'évaluation des réponses en attribuant un vote positif (+1) (pouce vers le haut), ou un vote négatif (-1) (pouce vers le bas) pour cette réponse. Ils peuvent aussi le faire pour la question, sur son utilité et sa clarté, en utilisant le même système de (+1) ou (-1) (ce qui permet de distinguer les questions populaires). La figure 2.1 illustre une question sur le site. Par défaut, les réponses sont ordonnées en fonction de la somme des votes qu'elles reçoivent, en mettant la réponse acceptée toujours en tête de la liste (même quand d'autres réponses la dépassent en termes de somme des votes). La réponse acceptée est marquée par un signal vert.

Afin de renforcer la sécurité, le site dispose d'un système de réputation qui s'appuie sur la distribution de badges aux utilisateurs. Il existe 53 badges distribués sur 6 catégories (badges de question, badges de réponse, badges de participation, badges de mots clés, badges de modération, autres). Les badges déterminent la légitimité et les droits de chaque utilisateur. Par exemple, le badge *étudiant* est donné à l'utilisateur dès qu'il pose sa première question, le badge *écolier* quand il accepte une réponse pour la première fois, le badge *enseignant* quand il répond à une question et que sa réponse obtient une somme de votes supérieure à 1.

Chaque action est associée à une valeur numérique de points, cela permet à l'utilisateur d'accumuler des points et d'avoir un score de réputation. Par exemple, l'utilisateur peut gagner deux points en acceptant une réponse à sa question, il gagne aussi cinq points à chaque fois

10. <http://www.stackexchange.com>

11. <http://www.stackoverflow.com/>

foreign libraries in a shared library

▲ 2
▼

I've written a c++ shared library that is based on several other libraries. Now when using the library I get a lot of undefined references concerning the functions of the 'foreign' libraries. Compiling and linking works fine if I explicitly add the 'foreign' libs to the g++ command. Is there away to cope with that so that the library user does not have to link against the 'foreign' libraries manually?

★ Thanks for your help.

`c++` `g++` `shared-libraries` `shared`

share | edit | flag

asked Nov 7 '12 at 10:05

add comment

2 Answers active oldest votes

▲ 1
▼

If you don't want users of your shared-library having to declare all the link-time dependencies of your library, you can inform the linker already of those dependencies when building the shared-library itself.

In a Makefile, this would look like this:

```
mylib.so:
g++ -o mylib.so $OBJ$ -llib1 -llib2 -llib3
```

Then the users of `mylib.so` don't need to specify `lib1`, `lib2` or `lib3`, but they would still need to be available (in the right version) for the program to be able to run.

share | edit | flag

answered Nov 7 '12 at 14:14

add comment

▲ 0
▼

You should exclude all external datatypes and functions from your published (exported) interface. It means, for GCC, you can use `-fvisibility=hidden` to hide all implementation details, so, export only your parts with `attribute ((dllexport))`. Pimpl idiom (aka [Opaque pointer](#)) helps you with removing dependencies from published headers. Quick example below:

```
#include<memory>

#if EXPORT_DLL
```

FIGURE 2.1 – Une question dans Stackoverflow

que l'une de ses réponses reçoit un vote positif. Le score accumulé par l'utilisateur est appelé "réputation". Certaines actions exigent un niveau minimal de réputation avant que l'utilisateur ait le droit de les faire. Par exemple, pour attribuer un vote négatif à une réponse, il faut avoir plus de 125 points de réputation.

Ultérieurement dans ce chapitre, nous allons proposer un système de recommandation à base de confiance globale (la réputation). Nous aimerions dès à présent clarifier ce point : hormis la différence méthodologique entre notre système à base de confiance globale et le système de réputation que nous décrivons, les deux systèmes n'ont pas les mêmes objectifs. Notre système vise à recommander l'utilisateur qui va fournir la bonne réponse à une question donnée, alors que le système en place a des objectifs plutôt sécuritaires pour guider les activités des utilisateurs et lutter contre les comportements aberrants.

Les données utilisées dans ce chapitre sont celles du réseau "Stackexchange". Elles ont été extraites en septembre 2011, et contiennent seulement 30 domaines. Le nombre total des comptes est de 374008 pour environ 153000 utilisateurs (puisque un utilisateur peut avoir un compte par domaine). Il y a 371594 questions et 816487 réponses. Les données existent sous forme XML. Elles donnent accès à la question même, aux réponses fournies, au profil du propriétaire de la question, aux profils des répondants, à la réponse acceptée, aux mots clés associés à la question,

au nombre de votes positifs et négatifs que la question/réponse a reçus (sans savoir qui les a attribués). Voici un exemple de données concernant une question :

```
<row
Id="1"
PostTypeId="1" /*1=question; 2= reponse*/
AcceptedAnswerId="13"
CreationDate="2010-09-13T19:16:26.763"
Score="45"
ViewCount="8799"
Body="This is a common question by those who have just rooted their phones.
What apps, ROMs, benefits, etc. do I get from rooting?
What should I be doing now?"
OwnerUserId="10"
LastEditorUserId="267"
LastEditorDisplayName=""
LastEditDate="2011-01-25T16:00:21.373"
LastActivityDate="2011-08-02T14:44:12.740"
CommunityOwnedDate="2011-01-25T08:44:10.820"
Title="I've rooted my phone. Now what?"
Tags="&rooting"
AnswerCount="8"
CommentCount="5"
FavoriteCount="31" />
```

Notre objectif est de prédire l'utilisateur qui va fournir la meilleure réponse (la réponse acceptée). Nous comparons la performance de quatre modèles de confiance :

1. MoleTrust : c'est le même modèle que celui utilisé dans le chapitre précédent. Nous l'utilisons ici comme modèle de référence. Nous considérons que quand un utilisateur fournit la réponse acceptée, un lien de confiance est établi entre l'auteur de la question et celui-ci.
2. Modèle de confiance locale à base de logique subjective.
3. Modèle des opinions collectives à base de logique subjective.
4. Modèle de confiance globale à base de logique subjective.

2.3 Confiance locale

Ce modèle est inspiré du travail de [Jøsang *et al.*2006]. Dans l'état actuel de notre connaissance, il n'a jamais été appliqué à des données réelles, dans le domaine de la recommandation. Il consiste à construire un réseau de confiance locale entre les utilisateurs. Ce réseau figure sous forme d'un graphe orienté où les nœuds représentent les utilisateurs, et les liens entre les nœuds représentent les relations de confiance. Le lien de confiance entre deux utilisateurs A et B est donné par l'opinion (selon la logique subjective) de A sur B . Ce lien est construit par les interactions directes entre les deux utilisateurs. Les interactions viennent de l'évaluation par A de la réponse de B à une question posée par A .

L'opinion dans la logique subjective se construit par les interactions. Lorsque A n'a aucune information sur B , son opinion sera complètement incertaine $\omega(A, B) = (b; d; u; a_x) = (0; 0; 1; a)$, nous rappelons que b représente la croyance de A en B , d représente l'incrédulité, u l'incertitude,

et a le taux de base, et les deux compteurs : r et s (qui comptabilisent les interactions réussies et échouées selon l'ordre) sont mis à zéro. Quand A pose une question dans un domaine donné, si B répond et A accepte sa réponse, cela est considéré comme une interaction réussie entre les deux.

Nous avons déjà montré que, de notre point de vue, un scénario de confiance repose sur deux arguments ; la volonté de B d'aider A , et sa compétence dans le contexte. Nous considérons ici que proposer une réponse est une preuve de la volonté d'aider. Pour que A accepte la réponse de B , il faut qu'il soit convaincu de la qualité de la réponse (la compétence de B dans ce domaine). Si cela est vérifié, l'opinion de A sur B sera mise à jour selon les formules 2.6, 2.7, et 2.8 du chapitre 2, où r représente le nombre des interactions réussites, s représente le nombre des interactions échouées. Dans le cas où la première interaction entre deux utilisateurs réussit : $r = 1$ et $s = 0$. La croyance (belief) sera :

$$b = \frac{r}{r + s + 2} = 1/3$$

l'incrédulité (disbelief) :

$$d = \frac{s}{r + s + 2} = 0$$

et l'incertitude :

$$u = \frac{2}{r + s + 2} = 2/3$$

finalemeent l'opinion sera :

$$\omega(A, B) = (1/3; 0; 2/3; a)$$

Maintenant, A connaît mieux B . Sa connaissance repose sur une interaction réussie, ce qui explique une tendance chez A à croire en B lors d'une prochaine interaction. L'incertitude de A a baissé suite à cette interaction mais elle est encore fortement présente car une seule interaction n'est pas suffisante pour avoir une opinion certaine sur B .

Notre modèle tente de prédire la réponse que l'utilisateur va accepter pour sa question. Le groupe \mathcal{R} représente les utilisateurs ayant proposé des réponses à la question. Un score sera associé à chaque membre $R \in \mathcal{R}$ pour savoir si A acceptera sa réponse ou pas. Le score de R correspond à l'espérance mathématique du lien de confiance entre A et R , nous appelons cela opinion personnelle. En cas d'absence d'une opinion personnelle, A va consulter ses amis pour savoir si quelqu'un parmi eux connaît R . Chacun des amis de A en fera de même (s'il a une opinion à propos de R , il la transfère à A , sinon il consulte à son tour ses amis). Ensuite, A agrège toutes les opinions de ses amis sur R . Nous appelons cela l'opinion collective.

Pour que l'opération précédente puisse se dérouler, il faut avoir un opérateur de transitivité et un opérateur d'agrégation. Dans la section 2.4.10, nous avons montré que la logique subjective possède plusieurs fonctions pour ces deux catégories.

Notre choix d'opérateur de transitivité sera celui favorisant l'incertitude, l'opérateur est présenté dans la section 2.4.10.1. Il considère que A néglige les avis des utilisateurs qu'il estime ne pas être dignes de confiance. Nous considérons que cette option est plus appropriée dans le contexte actuel, où les réponses doivent être plutôt factuelles avec le minimum de subjectivité. Souvent, la raison de l'échec de l'interaction est l'incompétence du fournisseur de la réponse. Il est donc préférable d'adopter une attitude de négligence vis-à-vis de ces utilisateurs plutôt que d'inverser leurs avis.

Pour ce qui est de l'agrégation, nous allons utiliser l'agrégation cumulative parce que quand A consulte ses amis (B et C) à propos de R , leurs réponses seront basées sur leurs propres

expériences avec R . Cela signifie que leurs opinions ne sont pas basées sur la même observation du comportement de R .

$$score(r) = \begin{cases} e(a, r) & \text{si } e(a, r) \in E \\ \sum_j \oplus [e(a, f_j) \otimes e(f_j, r)] & \text{sinon} \end{cases} \quad (2.1)$$

Le pseudo-code 1 montre comment ce modèle fonctionne en demandant l'opinion de vos amis.

Algorithm 1 fonction confianceLocale(A,B)

```

if ( $e(A, B) \in E$ ) then
  return  $e(A, B)$ 
else
   $e(A, B) \leftarrow e(0; 0; 1)$  {<opinion neutre>}
  for all  $f \in A.amis$  do
     $e(A, B) \leftarrow e(A, f) \oplus [e(A, f) \otimes confianceLocale(f, B)]$ 
  end for
  return  $e(A, B)$ 
end if

```

2.4 Confiance collective

Ce modèle repose complètement sur les opinions collectives. Dans le modèle précédent, les opinions collectives étaient utilisées seulement en cas d'absence d'opinions personnelles. Alors que dans ce modèle, les opinions collectives sont utilisées dans tous les cas. Cela signifie que A demandera automatiquement l'avis de ses amis à propos de tous les membres de R , même s'il a un avis personnel.

Supposons que l'opinion de A sur B est le fruit des 15 interactions réussies entre les deux utilisateurs, cela veut dire que A fait beaucoup confiance à B . Supposons aussi que B a eu 25 interactions avec F dont 23 réussies, la confiance de B en F sera également élevée. L'opinion de A sur F peut être extraite de ces deux opinions par transitivité avec un taux élevé de certitude. Si A a une seule interaction avec F , il appliquera son avis personnel et abandonnera la connaissance venant de 40 autres interactions (entre lui et B , et entre B et F). Cela signifie qu'il va construire son avis avec peu de certitude. Si le résultat de cette seule interaction entre A et F était un échec, F serait considéré indigne de confiance pour A . Il ne pourra plus jamais inverser cette décision dans un système de confiance locale. La confiance collective permet à B de jouer un rôle d'intermédiaire pour convaincre A de donner une nouvelle chance à F . Le regret fait partie de la confiance car il permet de rétablir les relations de confiance. Ce modèle permet d'intégrer cette notion, mais sa principale particularité reste la base élargie d'interactions sur laquelle il s'appuie pour formuler les opinions.

La fonction pour construire les opinions dans ce modèle est donnée formellement par l'équation suivante :

$$score(r) = \begin{cases} e(a, r) \oplus \sum_j \oplus [e(a, f_j) \otimes e(f_j, r)] & \text{si } e(a, r) \in E \\ \sum_j \oplus [e(a, f_j) \otimes e(f_j, r)] & \text{sinon} \end{cases} \quad (2.2)$$

Ce modèle suppose que les interactions directes sont souvent insuffisantes pour assurer une information suffisante sur les utilisateurs. Dans le modèle précédent, l'utilisateur pouvait fournir

une opinion personnelle sur un autre utilisateur une fois qu'il avait au moins une interaction avec lui. Nous pensons que cela affecte la qualité de l'opinion, en raison du manque d'expérience. L'utilisateur du modèle actuel agrège son opinion avec celles de ses amis, l'opinion de chaque ami étant conditionnée par la confiance que lui accorde l'utilisateur actif.

Algorithm 2 fonction $\text{ConfianceCollective}(A, \mathcal{R})$

```

Declare  $\text{scores}[\mathcal{R}]$ 
for all  $\text{score} \in \text{scores}$  do
   $\text{score} = \text{Opinion}(0; 0; 1)$  {opinion neutre}
end for
for all  $r \in \mathcal{R}$  do
  if  $\text{opinion}(A, r) \in E$  then
     $\text{scores}[r] = \text{opinion}(A, r) \otimes \text{scores}(r)$ 
  end if
end for
for all  $f \in A.\text{friends}$  do
   $f\text{score} = \text{ConfianceCollective}(f, \mathcal{R})$ 
  for all  $r \in \mathcal{R}$  do
     $\text{scores}[r] = \text{scores}[r] \oplus f\text{score}[r]$ 
  end for
end for
return  $\text{scores}$ 

```

Exemple :

Nous proposons ici un petit exemple pour mieux comprendre la différence entre les deux modèles :

Dans le graphe 2.2 :

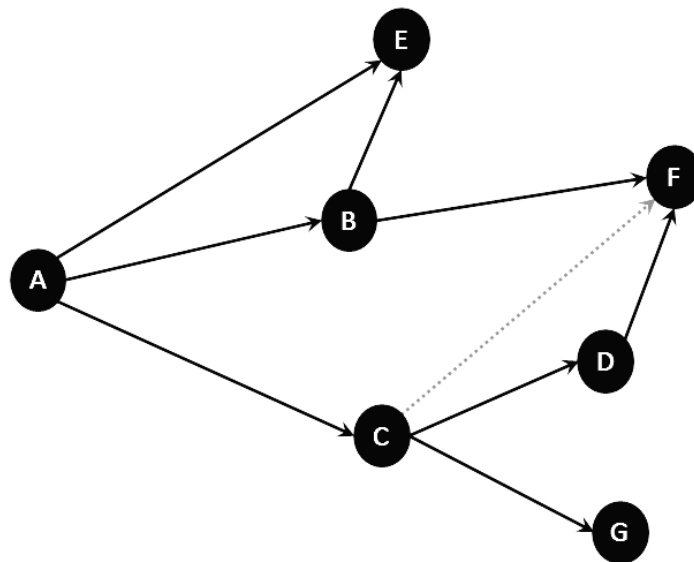


FIGURE 2.2 – Graphe de confiance

Supposons que A pose une question et qu'il reçoit trois réponses de E , F et G . C'est-à-dire

que $\mathcal{R} = \{E, F, G\}$. L'objectif de chaque modèle est d'ordonner \mathcal{R} pour A par la crédibilité de ses membres.

Le modèle à base de confiance locale proposera les équations suivantes (sans compter le lien en pointillé entre C et F) :

$$\text{score}(E) = \text{Op}(A, E)$$

$$\text{score}(F) = [\text{Op}(A, B) \otimes \text{Op}(B, F)] \oplus [\text{Op}(A, C) \otimes \text{Op}(C, D) \otimes \text{Op}(D, F)]$$

$$\text{score}(G) = \text{Op}(A, C) \otimes \text{Op}(C, G)$$

Le modèle collectif proposera les mêmes scores pour F et G , mais le score de E sera :

$$\text{score}(E) = \text{Op}(A, E) \oplus [\text{Op}(A, B) \otimes \text{Op}(B, E)]$$

Maintenant, si nous prenons en considération le lien gris entre C et F , le score de F dans le modèle local deviendra :

$$\text{score}(F) = [\text{Op}(A, B) \otimes \text{Op}(B, F)] \oplus [\text{Op}(A, C) \otimes \text{Op}(C, F)]$$

Et dans le modèle collectif :

$$\text{score}(F) = [\text{Op}(A, B) \otimes \text{Op}(B, F)] \oplus [[\text{Op}(A, C) \otimes \text{Op}(C, F)] \oplus [\text{Op}(A, C) \otimes \text{Op}(C, D) \otimes \text{Op}(D, F)]]$$

Le modèle local est très sensible à l'ajout des liens. Nous remarquons que dès que le lien est ajouté, le modèle néglige l'opinion de D sur F , même si D dispose de plus de certitude sur F que de C sur F .

2.5 Confiance globale

Le modèle que nous proposons est à base de confiance globale et de logique subjective. Ce modèle est contextuel, ce qui signifie que le score de réputation de l'utilisateur n'est pas absolu, mais lié aux mots clés des questions auxquelles l'utilisateur a donné une bonne réponse.

Chaque question est identifiée par un ensemble de mots clés \mathcal{M} associé à son sujet. Dans ce modèle, nous créons un profil d'utilisateur à base de sa réputation par mot clés. A chaque fois qu'un utilisateur fournit une réponse acceptée à une question, un lien (sous forme d'une opinion de la logique subjective) est établi entre lui et chacun des mots clés associés à la question. Les liens dans ce modèle ne sont pas de forme utilisateur-utilisateur mais mot clé-utilisateur. L'interprétation d'un lien correspond au degré de l'expérience ou de réputation de l'utilisateur relatif à chaque mot-clé. Un utilisateur réputé pour un mot-clé a le statut d'expert.

Pour ce qui concerne les réponses non acceptées, nous appliquons les mêmes quatre hypothèses que dans le modèle précédent, en tenant compte du fait que les relations lient les mots clés avec les utilisateurs.

Pour prédire la réponse acceptée à une question donnée Q posée par l'utilisateur A , ce modèle tente aussi d'ordonner la liste \mathcal{R} des personnes ayant répondu à la question. Le score de $R \in \mathcal{R}$ est la moyenne de ses espérances mathématiques de réputation avec les mots clés associés à la question. Le modèle estime que l'utilisateur ayant le score le plus élevé fournira la réponse acceptée.

Pour ce qui concerne l'algorithme de ce modèle

Exemple : Supposons qu'un utilisateur A pose une question Q_1 contenant les mots clés $\mathcal{M}_{Q_1} = \{\#jquery; \#security; \#browser\}$, et que A accepte la réponse de B à cette question.

Trois liens seront créés, entre B et chacun de mots clés de \mathcal{M}_{Q_1} . Ces liens sont des opinions de la LS avec chacun une seule interaction réussite et zéro interaction échouée ($r = 1$ et $s = 0$) :

$$\text{rep}(B, \#jquery) = \omega(1/3; 0; 2/3; 0, 5)$$

$$\text{rep}(B, \#security) = \omega(1/3; 0; 2/3; 0, 5)$$

Algorithm 3 fonction $\text{ConfianceGlobale}(A, \mathcal{R}, \mathcal{M})$

```

Declare  $scores[\mathcal{R}]$ 
for all  $score \in scores$  do
   $score = \text{Opinion}(0; 0; 1)$  {opinion neutre}
end for
for all  $r \in \mathcal{R}$  do
  for all  $m \in \mathcal{M}$  do
    if  $\text{opinion}(r, m) \in E$  then
       $scores[r] = \text{opinion}(r, m) \oplus scores[r]$ 
    end if
  end for
end for
return  $scores$ 

```

$$\text{rep}(B, \#browser) = \omega(1/3; 0; 2/3; 0, 5)$$

Supposons que par la suite B répond à une autre question Q_2 qui est liée aux mots clés $\mathcal{M}_{Q_2} = \{\#captcha; \#security\}$, et que la réponse est acceptée. Dans ce cas un nouveau lien sera créé entre B et le mot clé $captcha$:

$$\text{rep}(B, captcha) = \omega(1/3; 0; 2/3; 0, 5)$$

Alors que son lien avec le mot clé *sécurité* sera mis à jour ($r = 2$ et $s = 0$) :

$$\text{rep}(B, security) = \omega(0, 5; 0; 0, 5; 0, 5)$$

Puis l'utilisateur C répond à une troisième question Q_3 liée aux deux mots clés $\mathcal{M}_{Q_3} = \{\#legal; \#security\}$, et la réponse de C est acceptée.

$$\text{rep}(C, \#legal) = \omega(1/3; 0; 2/3; 0, 5)$$

$$\text{rep}(C, \#security) = \omega(1/3; 0; 2/3; 0, 5)$$

Supposons aussi qu'une quatrième question Q_4 est posée, elle contient trois mots clés associés : $\mathcal{M}_{Q_4} = \{\#security; \#hacking; \#legal\}$. L'algorithme de confiance globale va calculer la réputation de B et de C pour cette question :

$$\text{rep}(B, Q_4) = \text{rep}(B, \#security) \oplus \text{rep}(B, \#hacking) \oplus \text{rep}(B, \#legal)$$

$$\text{rep}(B, Q_4) = (0, 5; 0; 0, 5; 0, 5) \oplus (0; 0; 1; 0, 5) \oplus (0; 0; 1; 0, 5)$$

$$\text{rep}(B, Q_4) = (0, 2; 0; 0, 8; 0, 5)$$

L'espérance mathématique pour la réputation de B dans ce cas est :

$$E(B_{Q_4}) = b + a_b \times u = 0, 2 + 0, 8 \times 0, 5 = 0, 6$$

$$\text{rep}(C, Q_4) = \text{rep}(C, \#security) \oplus \text{rep}(C, \#hacking) \oplus \text{rep}(C, \#legal)$$

$$\text{rep}(C, Q_4) = (1/3; 0; 2/3; 0, 5) \oplus (0; 0; 1; 0, 5) \oplus (1/3; 0; 2/3; 0, 5)$$

$$\text{rep}(C, Q_4) = (0, 27; 0; 0, 73; 0, 5)$$

L'espérance mathématique pour la réputation de C dans ce cas est :

$$E(C_{Q_4}) = b + a_b \times u = 0,27 + 0,73 \times 0,5 = 0,635$$

L'algorithme considère que C est plus apte à fournir une réponse acceptable pour cette question. Il est vrai que b et C ont à priori le même nombre d'interactions réussies avec les éléments de \mathcal{M}_{Q_4} (deux interactions chacun), mais la connaissance de C est plus répertoriée. Ce qui fait que son espérance est légèrement plus élevée.

2.5.1 Le traitement des réponses non-acceptées

Le traitement des réponses acceptées est clair. Le problème est maintenant de traiter les réponses non-acceptées. En effet, quand un utilisateur n'a pas accepté une réponse, nous ne savons pas si la raison est que la réponse est mauvaise, ou correcte mais moins bonne que la réponse acceptée, ou encore meilleure que la réponse acceptée mais arrivée après que l'utilisateur ait déjà trouvé une solution satisfaisante avec la première réponse.

Face à cette situation, nous proposons quatre hypothèses pour interpréter une réponse non acceptée :

1. hypothèse rigoureuse : les réponses non acceptées sont considérées comme des interactions échouées.
2. hypothèse ignorante : les réponses non acceptées ne sont pas considérées.
3. hypothèse probabiliste subjective indépendante : dans les deux hypothèses précédentes, la valeur de l'interaction est soit $+1$ (succès) ou -1 (échec). Dans l'hypothèse actuelle, nous introduisons des interactions relativement réussies/échouées. Nous utilisons les votes attribués par la communauté à la réponse pour estimer un jugement de son succès/échec. Le jugement est sous la forme d'une opinion de logique subjective. En fait, le vote positif (pouce vers le haut) représente une interaction réussie avec un utilisateur inconnu. De même, le pouce vers le bas est considéré comme une interaction échouée avec un utilisateur inconnu. Dans cette hypothèse, nous tenons compte de la réaction globale de la communauté à l'égard de la réponse. Cette réaction est formulée sous la forme d'une opinion subjective résultant des interactions des membres avec la réponse. Ici, nous utilisons la logique subjective dans deux phases indépendantes ; la première, pour estimer la valeur de l'interaction. La deuxième, pour les liens de confiance entre les utilisateurs (comme dans les deux hypothèses précédentes).

Pour calculer la valeur de l'interaction, nous considérons que pour une réponse acceptée :

$$\omega(A, rep_B) = (1; 0; 0; 0,5)$$

où rep_B est la réponse de B .

Par conséquent le résultat de l'interaction est égal à l'espérance de $\omega(A, rep_B)$ selon l'équation 2.3 :

$$Ex(\omega_{rep_B}^A) = b + u \times a_x = 1 + 0 \times 0,5 = 1$$

Quand la réponse n'est pas acceptée, nous considérons que : $r = card(pouce_{Haut})$ et $s = card(pouce_{bas})$:

$$\omega(A, rep_B) = \left(\frac{r}{r+s+2}; \frac{s}{r+s+2}; \frac{2}{r+s+2}; 0,5 \right)$$

Ensuite nous calculons l'espérance mathématique selon 2.3.

Notons que l'intervalle de l'espérance mathématique est $[0, 1]$, alors que l'interaction peut avoir une valeur négative, qu'elle est dans l'intervalle $[-1, +1]$. Nous ramenons donc la valeur de l'espérance à l'intervalle $[-1, +1]$ avant de l'utiliser dans l'étape suivante.

Normalement, le résultat de l'interaction est une valeur concrète -1 (échec) ou $+1$ (réussi). Mais l'hypothèse actuelle permet d'exprimer le résultat de l'interaction avec des valeurs probabilistes.

4. hypothèse probabiliste subjective dépendante : cette hypothèse est comparable à la précédente, sauf qu'ici les probabilités de réussite et d'échec des interactions sont dépendantes l'une de l'autre. Les votes qu'une réponse reçoit traduisent sa qualité relativement aux qualités des autres réponses à la même question. Dans cette hypothèse, la certitude est influencée par le nombre global de votes attribués à toutes les réponses à une question. En conséquence, l'opinion sur la réponse dépend des autres opinions sur les autres réponses :

$$certitude_j = \frac{\sum_j p}{2 + \sum_{i=rep_1}^{rep_n} \sum_i p}$$

où p est la valeur absolue d'un vote.

j est la réponse actuelle.

n est le nombre de réponses pour la question actuelle.

Ainsi, les trois composants de l'opinion seront :

$$u_j = 1 - certitude_j$$

$$b_j = certitude_j \times \frac{\sum_j p_h}{\sum_j p}$$

$$d_j = certitude_j \times \frac{\sum_j p_b}{\sum_j p}$$

où p_h est la somme des votes positifs, p_b est la somme de votes négatifs.

Exemple : Supposons qu'un utilisateur A pose une question, que les utilisateurs B et C proposent chacun une réponse à cette question, et que A accepte la réponse de B . Supposons aussi que la réponse de B obtient 3 votes positifs et 2 votes négatifs, et que la réponse de C obtient un vote positif et 4 négatifs.

Un lien de confiance entre A et B est créé. Ce lien est une opinion de la LS avec une seule interaction réussie et zéro interaction échouée ($r = 1$ et $s = 0$) :

$$Confiance(A, B) = \omega(1/3; 0; 2/3; 0, 5)$$

S'agissant de la réponse de C , cela dépend de l'hypothèse prise en compte :

1. L'hypothèse rigoureuse : un lien est aussi créé entre A et C représentant une opinion avec une interaction échouée et zéro interaction réussie ($r = 0$ et $s = 1$) :

$$Confiance(A, C) = \omega_r(0; 1/3; 2/3; 0, 5)$$

2. L'hypothèse ignorante : la réponse de C est ignorée, et aucun lien n'est créé.

3. L'hypothèse probabiliste indépendante : il y a une personne qui a attribué un vote positif à la réponse de C , c'est l'équivalent d'une interaction réussie avec cette réponse $r = 1$. Il y a quatre personnes qui ont attribué un vote négatif, c'est l'équivalent de quatre interactions échouées avec cette réponse $s = 4$. Le jugement de cette réponse est représenté par l'opinion ω_{C_i} :

$$\omega_{C_i} = (1/7; 4/7; 2/7; 0, 5)$$

Par la suite, nous calculons l'espérance mathématique de ω_{C_i} :

$$E(x) = b + a \times u = \frac{1}{7} + 0,5 \times \frac{2}{7} = 0,29$$

Nous ramenons la valeur de l'espérance à l'intervalle $[1, +1]$:

$$E(x) = -0,42$$

La réputation de C vis-à-vis des trois mots clés est une opinion avec ($r = 0$ et $s = 0,42$) :

$$rep(A, C) = \omega_i(0; 0,17; 0,83; 0,5)$$

4. L'hypothèse probabiliste dépendante : la réponse de C a obtenu 5 votes sur un total de 10 votes attribués aux deux réponses.

$$Certitude = \frac{5}{2 + 5 + 5} = 0,42$$

$$u = 1 - 0,42 = 0,58$$

$$b = 0,42 \times \frac{1}{5} = 0,08$$

$$d = 0,42 \times \frac{4}{5} = 0,34$$

Le jugement de cette réponse est représenté par l'opinion ω_{C_d} :

$$\omega_{C_d} = (0,08; 0,34; 0,58; 0,5)$$

Par la suite, nous calculons l'espérance mathématique de ω_{C_d} :

$$E(x) = b + a \times u = 0,08 + 0,5 \times 0,58 = 0,3$$

Nous ramenons la valeur de l'espérance à l'intervalle $[1, +1]$:

$$E(x) = -0,26$$

La réputation de C vis-à-vis les trois mots clés est une opinion avec ($r = 0$ et $s = 0,26$) :

$$Confiance(A, C) = \omega_d(0; 0,12; 0,88; 0,5)$$

2.6 Mesures de qualité

Le problème de l'identification de la réponse acceptée est posé comme un problème de tri d'une liste à un seul élément pertinent.

Le rang moyen réciproque¹² (MRR) est une mesure de performance des algorithmes de prédiction des listes ordonnées. Néanmoins, il n'est pas parfaitement adapté à notre cas. MRR est habituellement utilisé pour les systèmes qui doivent prévoir une liste d'éléments à l'intérieur de laquelle un élément spécifique existe. Dans le cas présenté ici, nous essayons plutôt de trouver la réponse acceptée par reclassement d'une liste existante de réponses, nous savons que la réponse existe déjà dans la liste et nous cherchons à l'identifier. Ainsi, dans le cas particulier du classement en dernière position de l'élément pertinent, MRR récompense l'algorithme pour avoir fait figurer l'élément dans la liste. Dans notre cas, la liste est prédéfinie, l'algorithme ne doit pas être récompensé pour le classement de l'élément pertinent à la fin de la liste. La plage de valeurs de MRR est de $[1/r; 1]$ où r est le rang donné par l'algorithme évalué à l'élément pertinent. Nous en proposons une version modifiée, dont la plage est $[0; 1]$, et qui rend 1 si l'élément pertinent est dans le haut de la liste, et 0 s'il est à la fin de la liste. Nous l'appelons rang moyen de liste prédéfinie (RMLP), où le rang de la liste prédéfinie (RLP) est donné par la formule :

$$RLP = \frac{N - r}{N - 1}$$

Où : N est la taille de la liste.

RMLP est la moyenne de toutes les questions pour RLP. Nous employons une stratégie de classement des *ex aequo*. Par exemple, si deux éléments en tête de la liste ont le même score, tous les deux sont considérés comme seconds, aucun n'est classé en première position.

2.7 Résultats et discussions

Notre discussion inclut trois volets. Nous comparons les précisions de prédiction des modèles de la LS sont entre eux et avec MoleTrust. Nous calculons la complexité de chacun des modèles de la LS, pour juger leurs temps d'exécution. Nous discutons de la résistance face aux attaques malveillantes pour juger leurs capacités à fonctionner dans des conditions défavorables.

2.7.1 Précision

L'objectif des modèles proposés dans ce chapitre est de prédire la réponse acceptée pour une question dans le corpus de "stack exchange". Les questions intégrées dans cette expérimentation sont les questions qui ont plus d'une seule réponse dont une est acceptée. Il y en a au total 118778. Nous les avons distribuées aléatoirement en 5 groupes de même taille, avant d'effectuer un test croisée à cinq itérations ; à chaque itération nous utilisons 4 groupes pour l'apprentissage des modèles (construction du réseau de confiance) et le dernier groupe pour tester la performance. L'évaluation est faite selon les deux mesures définies dans la section 2.6.

Le tableau 2.1 illustre les valeurs de (MRR) des quatre modèles cités dans ce chapitre en utilisant les quatre hypothèses. Le tableau 2.2 illustre les valeurs de (RMLP). Il est normal que les valeurs de ce dernier soient inférieures à celles de (MRR). Mais les deux mesures suivent la même tendance et conduisent aux mêmes conclusions.

12. En anglais : Mean reciprocal rank

Méthode	MoleTrust	Confiance locale	Confiance collective	Confiance globale
Rigoureuse	-	0,57	0,88	0,884
Ignorante	0,53	0,58	0,75	0,7
Probabiliste dépendante	-	0,62	0,87	0,815
Probabiliste indépendante	-	0,617	0,86	0,78

TABLE 2.1 – Résultats de MRR

Méthode	MoleTrust	Confiance locale	Confiance collective	Confiance globale
Rigoureuse	-	0,37	0,85	0,85
Ignorante	0,3	0,36	0,69	0,6
Probabiliste dépendante	-	0,442	0,84	0,76
Probabiliste indépendante	-	0,438	0,83	0,73

TABLE 2.2 – Résultats de MPLR

Il apparaît clairement que la performance des modèles à base de logique subjective dépasse largement celle de MoleTrust. Cette comparaison n'est valable que pour l'hypothèse d'ignorance, car les autres hypothèses ne sont pas applicables à MoleTrust.

La flexibilité de la logique subjective permet l'application des trois autres hypothèses, qui apportent des améliorations de performance. Cela indique que les réponses non acceptées peuvent apporter des informations importantes et non négligeables. Nous remarquons que pour le modèle de confiance locale, ce sont les hypothèses probabilistes, qui atteignent la meilleure performance par rapport à l'hypothèse rigoureuse. C'est le cas inverse pour le modèle de confiance globale où c'est l'hypothèse rigoureuse qui dépasse toutes les autres hypothèses. Finalement, le modèle collectif est presque indifférent pour les trois hypothèses (la rigoureuse et les deux hypothèses probabilistes). Par conséquent, on ne peut pas décider de la meilleure hypothèse, cela dépend du modèle de confiance. Cela peut avoir un lien avec la quantité d'interactions dont le modèle dispose pour sa prise de décision. Le modèle local dispose de peu d'informations, le modèle collectif en utilise plus, et le modèle global en utilise encore plus. On remarque alors que plus on a d'informations, plus l'hypothèse rigoureuse est de mise.

S'agissant des trois modèles de logique subjective, il se trouve que le modèle collectif et le modèle global ont des performances proches, et qu'ils dépassent le modèle local. Ceci s'explique simplement par le fait que ces deux modèles basent leur raisonnement sur un nombre plus élevé d'interactions par rapport au modèle local, ce qui leur permet de fournir des opinions plus confortées.

2.7.2 Complexité

L'importance de l'évaluation de la complexité est d'estimer le temps nécessaire pour exécuter chaque modèle. Un bon SR doit être capable de générer des recommandations dans un délai raisonnable.

La complexité de l'algorithme est une fonction $t(n)$, où n est la taille de l'entrée. La fonction de complexité donne une idée de la durée d'exécution prévue de l'algorithme étant donné une entrée de taille n . La complexité du calcul est indépendante du matériel, du langage de programmation, du compilateur, et des détails de l'implémentation. Elle ne prend en considération que les opérations élémentaires de l'algorithme telle que : affectation d'une variable ($t(n) = 1$), la comparaison ($t(n) = 1$), la boucle sur une liste de taille n ($t(n) = n$), la comparaison des valeurs d'un tableau l'une à l'autre ($t(n) = n^2$), le parcours d'un graphe ($t(n) = V + E$), où V est le nombre de sommets, et E est le nombre de liens).

La notation O est utilisée pour représenter la complexité, cette notation ne conserve que l'élément primaire qui maximise la complexité de l'algorithme. Par exemple, ayant un algorithme de $t(n) = n^2 + 4 \times n + 2$, l'équivalent en notation O est $O(n) = n^2$.

En règle générale, l'évaluation de la complexité prend en compte le *pire des cas* et le *cas moyen*. Le pire des cas représente la limite supérieure du temps nécessaire à l'exécution de l'algorithme, et le cas moyen est la borne inférieure.

Le pire des cas

Le modèle local et le modèle collectif procèdent en parcourant le graphe R fois, où R est le nombre d'utilisateurs qui ont proposé des réponses à la question. Ceci fait qu'ils ont une complexité de $O(R \times (V + E))$ (V est le nombre d'utilisateurs et E est le nombre des liens de confiance). En revanche, la complexité du modèle global est égale à $O(R \times M)$, où M est la taille de la liste de mots-clés associés au profil du membre de R .

Le cas moyen

Nous définissons R' un sous-ensemble qui contient les membres de R avec lesquels l'auteur de la question n'a pas un lien direct. Dans ce cas, la complexité en cas moyen du modèle local est égale à $O(R' \times (V + E))$, alors que la complexité en cas moyen du modèle collectif reste la même que celle au pire des cas. Nous concluons que le modèle local est moins complexe que le modèle collectif car $R' \subset R$.

Pour ce qui concerne le modèle global, nous utilisons les tables de hachage pour stocker le profil d'utilisateur, celles-ci ont une complexité en cas moyen égale à $O(1)$. Cela permet de réduire la complexité du cas moyen du modèle global à $O(R)$.

Pour conclure, nous constatons que le modèle global est le moins complexe, suivi par le modèle local. Le modèle collectif est le plus complexe, celui dans qui consomme le plus de temps. Cette analyse illustre la limitation du modèle collectif pour les applications menées d'énormes graphes.

2.7.3 Résistance face aux attaques malveillantes

Dans un scénario d'attaque de groupe malveillant, plusieurs utilisateurs s'allient pour attaquer le SR et contaminer les recommandations des autres utilisateurs. On peut alors identifier trois communautés d'utilisateurs. Les attaquants qui participent à l'exécution de l'attaque. Les utilisateurs concernés dont les recommandations sont contaminées à cause de l'attaque. Et les utilisateurs purs qui ne sont pas touchés par l'attaque.

Dans l'application actuelle, les attaquants peuvent continuellement ajouter des questions, y répondre, et accepter leurs réponses pour devenir des utilisateurs recommandables par le système, et capables par la suite de contaminer les questions des autres communautés.

La topologie de graphe des modèles local et collectif sert à isoler la communauté d'attaquants. Le comportement communautaire les rend très connectés l'un à l'autre, mais faiblement reliés aux autres communautés. Ainsi, un utilisateur pur ne peut pas être affecté à moins qu'il ne se décide à faire confiance à un ou plusieurs attaquants, ce qui est très peu probable. Même si cela se produit une fois par accident, le lien résultant ne suffit pas (en particulier dans le modèle collectif). Car ce lien est basé sur une seule interaction, et il est plus incertain que d'autres liens, donc avec une influence médiocre.

La Figure 2.3 illustre un résumé de notre discussion.

Dans [Levien2004], les auteurs proposent la propriété de goulot pour discuter la résistance d'un modèle de confiance à l'attaque du groupe. Elle signifie que pour une relation de confiance

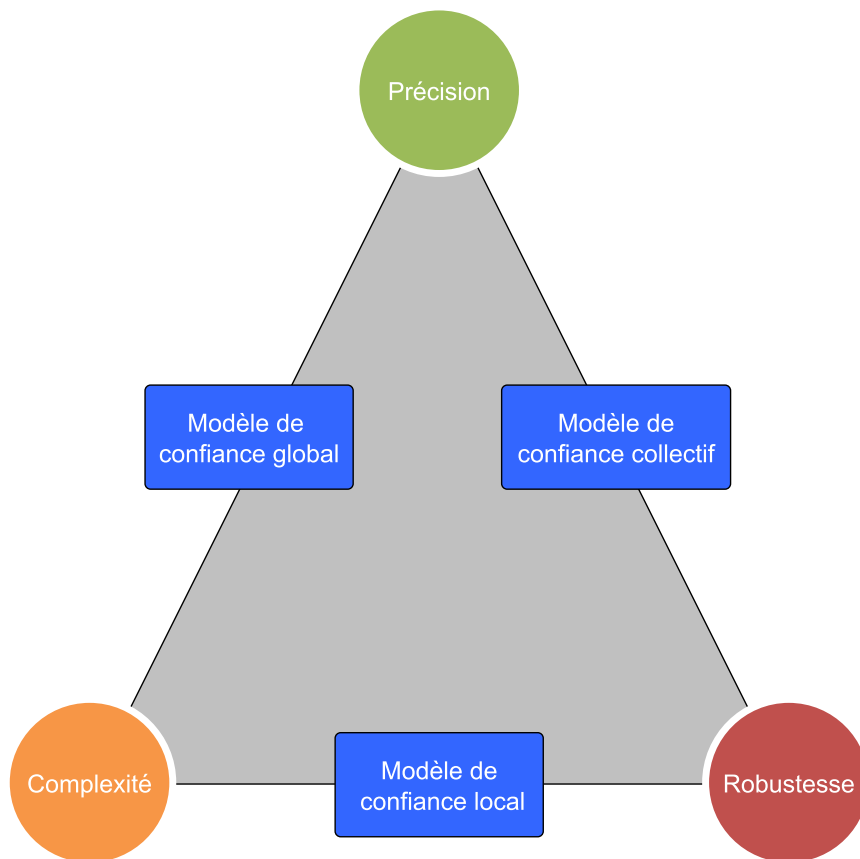


FIGURE 2.3 – évaluation des trois modèles de confiance à base de la logique subjective

entre un utilisateur pur s et un attaquant t ($s \rightarrow t$), la relation ne doit pas être affectée par les successeurs de t . Les graphes de nos modèles se conforment à cette propriété, car les liens de la LS sont fortifiés seulement par les interactions réussies entre s et t ou par les utilisateurs qui précèdent t , ce qui permet d’isoler la communauté des attaquants.

Bien que le modèle global soit vulnérable aux attaques de groupes, le fait qu’il soit contextualisé complique l’exécution d’une telle attaque. La réputation d’un utilisateur est calculée par domaine, et est associée aux mots clés. C’est-à-dire que les attaquants sont obligés de cibler une notable quantité des mots clés, et de répéter l’attaque par domaine pour qu’elle réussisse.

2.8 Conclusion

La performance de la logique subjective surpasse nettement celle de MoleTrust. La logique subjective est plus fidèle à la définition de la confiance sociale que MoleTrust. Elle utilise des liens inférés de confiance qui la rendent plus dynamique et réaliste dans sa représentation de la confiance, ce qui n’est pas le cas de MoleTrust. Les modèles à base de logique subjective ne se limitent pas à la réponse générée par le chemin le plus court, mais au contraire, ils encouragent les réponses les plus sûres qui intègrent et analysent plus de connaissances avant d’arriver à une décision. Ceci est clairement établi dans les liens inférés qui se basent sur de multiples interactions, mais aussi dans les fonctions d’agrégation des opinions.

Dans des systèmes où l'on manque d'informations, il faut tenter d'exploiter d'autres indicateurs pour chercher de la connaissance. Dans ce chapitre, nous n'avons pas d'informations concrètes sur la défiance, mais avec nos hypothèses nous pouvions en extraire une estimation et améliorer les performances des trois modèles.

La logique subjective fait preuve d'une flexibilité qui nous a permis de proposer ces trois modèles. Elle est aussi ajustable selon le contexte et l'application, grâce à la variété des opérateurs proposés.

Le modèle collectif et le modèle global prennent largement l'avantage sur le modèle local au niveau de la prédiction. Néanmoins, chacun des trois modèles a ses qualités et ses inconvénients. Les deux modèles (collectif et local) souffrent du problème du démarrage à froid, ils ne peuvent rien prédire pour un utilisateur qui n'a pas d'ami ; le modèle global n'a pas ce problème. Il peut prédire des réponses pour les nouveaux utilisateurs. D'un autre côté, le modèle global risque de converger vers une petite communauté d'utilisateurs réputés, car les nouveaux arrivants auront du mal à gagner en réputation.

Le modèle local souffre aussi d'un problème de manque de données, il prend ses décisions avec un minimum de connaissances, ce qui fait qu'elles sont moins optimales par rapport à celles des deux autres modèles. Le modèle collectif, qui renforce l'utilisation des opinions collectives afin de pallier ce problème, peut poser un problème de passage à l'échelle, car ce modèle fait ses calculs en temps réel. Cela risque de consommer un temps considérable pour parcourir le graphe notamment si l'utilisateur a beaucoup d'amis. Le modèle local est moins exposé à ce problème parce qu'il fait moins de propagation.

Théoriquement, le modèle global est plus exposé à une attaque de groupe. Le modèle local ainsi que le modèle collectif offrent une meilleure protection contre les attaques malveillantes. En effet, ces deux derniers ne prennent en compte que l'information en provenance des amis de l'utilisateur même, ainsi le groupe d'attaque sera écarté et les votes de ses membres ne seront pas utilisés par d'autres utilisateurs.

Le modèle global est le plus rapide parmi le trois modèles de LS, alors que le modèle collectif consomme le plus de temps.

Dans ce chapitre, nous avons proposé et comparé trois modèles à base de la LS. Nous avons montré les points forts et faibles de chacun d'entre eux. Cela permet de choisir le modèle et le type de modélisation de confiance selon l'application.

Troisième partie

Conclusion

Conclusion et perspectives

Sommaire

1.1 Bilan	101
1.2 Perspectives	104

1.1 Bilan

Le filtrage collaboratif à base d'utilisateurs est l'algorithme de recommandation le plus répandu et le plus fréquent. Récemment, les modèles de recommandation à base de confiance ont souvent pu dépasser la performance du FC dans les réseaux sociaux. Néanmoins, ces modèles souffrent encore d'un léger problème de démarrage à froid et de fluctuation. Dans cette thèse, nous avons montré que l'hybridation de ces deux algorithmes peut garantir des recommandations à une partie plus importante d'utilisateurs, sans une perte significative de précision. Le système de recommandation doit toujours pouvoir attirer de nouveaux clients pour réussir, et ne pas se contenter de fournir de bonnes recommandations à une communauté limitée d'utilisateurs. Les comportements des utilisateurs sont hétérogènes ; cela exige que le système de recommandation soit flexible pour maximiser la satisfaction au sein de ses utilisateurs. Dans [Schwartz2004], l'auteur cite l'exemple d'un utilisateur qui vient, au temps des smart-phones, chercher un téléphone qui ne fait que téléphoner, et il parle de la difficulté à satisfaire ce genre d'utilisateurs sans les inciter à changer leurs comportements. Les approches de recommandation hybrides permettent de réaliser ce but en s'adaptant à l'utilisateur. Voilà d'où vient aussi l'importance d'effectuer une analyse quantitative (et qualitative) du comportement de l'utilisateur afin de mieux comprendre ses besoins et sa manière d'utiliser le système.

L'analyse comportementale des utilisateurs d'un système de recommandation privilégie le côté quantitatif (nombre de notes attribuées, nombre d'amis crédibles, etc.). Cette analyse peut détecter les utilisateurs en situation de démarrage à froid sans aller plus loin. Elle tend aussi à limiter l'explication de bonnes/mauvaises performances par la masse d'activité générée par l'utilisateur, ce qui n'est pas suffisant à notre sens. Nous avons montré que le comportement de l'utilisateur est plus complexe, et qu'il est sensible à d'autres caractéristiques. Notre analyse s'intéresse aussi aux aspects qualitatifs du comportement.

Nous avons trouvé une corrélation entre la popularité des items notés par l'utilisateur et la performance du système de recommandation à l'égard de cet utilisateur. Cette relation a deux effets distincts selon le nombre de notes attribuées par l'utilisateur. Noter des items populaires

est profitable pour les utilisateurs en situation de démarrage à froid, car cela maximise la probabilité de trouver des voisins pour l'utilisateur concerné, ce qui n'est pas évident dans le cas où l'utilisateur note des items peu communs ou rarement notés. Dans le cas des utilisateurs qui notent beaucoup d'items c'est l'inverse ; car si l'utilisateur continuait à noter des items très populaires, il s'identifierait à beaucoup d'utilisateurs, et par conséquent aurait une identité floue et son profil contiendrait beaucoup de bruit, tant que le système n'arrive plus à déterminer ses voisins réels, et l'impact sur la qualité de recommandations serait négative.

Nous avons défini aussi le coefficient de déviation pour exprimer la déviation du comportement d'un utilisateur par rapport à l'orientation de la communauté. Cette mesure est importante pour identifier les utilisateurs ayant un goût atypique. Ce coefficient est discriminant surtout pour les utilisateurs en situation de démarrage à froid. Ces utilisateurs sont nettement plus faciles à satisfaire quand ils suivent l'orientation de la communauté. La discrimination devient moins importante pour le reste des utilisateurs mais elle existe toujours. Cela peut expliquer le fait que certains utilisateurs en situation de démarrage à froid reçoivent des recommandations plus précises que d'autres utilisateurs normaux.

La dernière caractéristique qualitative est liée aux relations de confiance exprimées par l'utilisateur. On a mesuré la réputation des amis auxquels il fait confiance. La réputation d'un utilisateur est le nombre d'utilisateurs lui faisant confiance. Plus les amis d'un utilisateur sont réputés, meilleure est la qualité des recommandations qu'il reçoit. Ceci est valable seulement jusqu'à 10 points de réputation (selon le modèle actuel), car au delà, la qualité se stabilise. L'impact d'un ami crédible sur la qualité de recommandation s'améliore tant qu'il est jugé digne de confiance par près d'une dizaine d'utilisateurs au moins, mais au delà de 10 l'impact devient plus ou moins fixe. Dans le contexte étudié, un utilisateur est jugé fiable dès qu'il a dix utilisateurs qui lui font confiance.

Ce dernier résultat (la fiabilité à partir de 10) est propre aux données et au contexte applicatif utilisés dans nos expérimentations. Cela peut être aussi le cas des autres conclusions tirées à partir des autres caractéristiques discutées ci-dessus. Néanmoins, pour généraliser, notre affirmation porte sur l'importance des caractéristiques qualitatives du comportement, et à leur capacité à mieux comprendre le profil comportemental de l'utilisateur, et à expliquer la fluctuation des algorithmes de recommandation. Nous soutenons l'importance du côté qualitatif de l'analyse de recommandation sans être limité aux caractéristiques présentées ici, car d'autres peuvent aussi se trouver selon le contexte et le type de données disponibles.

Une autre analyse plus approfondie est effectuée par la classification des utilisateurs. La classification permet d'identifier des sous-ensembles d'utilisateurs ayant des comportements communs vis-à-vis de multiples caractéristiques. Les caractéristiques qualitatives et quantitatives forment les dimensions d'un vecteur qui décrit le comportement de l'utilisateur. La classification des utilisateurs sert à détecter les comportements composés, regrouper les utilisateurs selon ces comportements, et mesurer la performance de chaque algorithme de recommandation pour une communauté comportementale d'utilisateurs. Cette analyse de performance locale permet de détecter à la fois les types d'utilisateurs et les comportements pratiqués, et aussi les points forts et faibles de chaque algorithme de recommandation. La performance des algorithmes change radicalement d'une classe à l'autre, il est donc envisageable d'associer un algorithme par classe d'utilisateurs.

Dans cet exemple, nous nous sommes limités à trois approches à base d'utilisateurs, mais la méthode est susceptible d'intégrer d'autres approches selon la nature des données. Une question se pose sur l'évolution du profil de l'utilisateur (le vecteur comportemental de l'utilisateur). Comment traiter cette évolution qui peut rendre l'algorithme choisi moins efficace pour l'utilisateur ? Une autre question peut se poser sur la classification des nouveaux utilisateurs. La méthode dans

son état actuel ne propose pas de réponse concrète à ces questions, mais plusieurs solutions sont envisageables : dédier une classe aux nouveaux utilisateurs, employer un algorithme de classification incrémental comme "growing neural gas" [Fritzke and others1995], ou même refaire la classification intégralement avec un intervalle temporel prédéfini.

Les résultats de la première partie accordent une supériorité à l'approche à base de confiance par rapport au FC. Dans la deuxième partie, nous nous intéressons aux caractéristiques des modèles de confiance. Dans l'état de l'art, nous avons passé en revue des modèles existants en tenant compte de leur fidélité et complétude par rapport à la confiance sociale.

La logique subjective possède une base théorique très solide. Elle représente à la fois la confiance et la défiance. Elle adopte une notion de notation évolutive. Elle fonctionne dans des environnements comportant des carences en matière de preuves. Finalement, elle dispose de plusieurs fonctions de propagation de confiance, et d'agrégation des avis même en cas de contradiction. Elle est même censée pouvoir représenter les utilisateurs de nature réservée ou confiante.

Notre travail était de mettre la logique subjective à l'épreuve, et de faire usage de sa flexibilité pour proposer plusieurs modèles et comparer leurs performances.

Nous avons proposé un modèle à base de confiance locale, ce modèle s'appuie sur les opinions des utilisateurs les uns vis-à-vis des l'autres et réciproquement. L'intérêt de cette approche réside dans le fait qu'elle propose des solutions personnalisées dépendant du réseau construit par l'utilisateur lui-même, ce qui rend le modèle plus transparent et capable d'expliquer ses recommandations à l'utilisateur, d'autant plus que ce dernier connaît déjà ses voisins. Le modèle favorise l'utilisation des opinions individuelles (personnelles) de l'utilisateur. Néanmoins, ce type d'opinions donne des résultats personnalisés certes, mais imprécis. Les interactions personnelles sont souvent insuffisantes pour assurer une connaissance juste et complète. Dans cette approche, même lorsque la valeur d'incertitude est grande, l'utilisateur reste enfermé dans son opinion. Il peut se contenter d'une seule interaction avec un autre utilisateur pour apporter un jugement sur sa crédibilité. Si la première interaction entre deux utilisateurs venait à échouer, il n'aurait plus de moyen pour rétablir la confiance entre eux.

Nous avons proposé un modèle à base de confiance globale, mais qui prend en compte le contexte. Ce modèle s'appuie sur la réputation des utilisateurs, qui naît de leurs compétences à fournir de bonnes recommandations, en tenant compte du contexte. Il répond à une question fondamentale souvent posée au sujet des modèles à base de confiance globale, à propos de la capacité de l'utilisateur à être réputé dans différents contextes. La réputation ici est limitée aux mots clés associés aux contextes où l'utilisateur a été efficace. La réputation contextualisée augmente nettement la précision de la prédiction par rapport au modèle de confiance locale. La recommandation est moins explicable, et les résultats ne sont pas personnalisés dans le sens où ils sont indépendants de l'identité de l'utilisateur à qui les recommandations sont faites. Cela peut faire que le système repose sur une minorité d'experts (utilisateurs réputés), ce qui peut être bien ou pas selon l'application. Il peut arriver aussi que pour les nouveaux utilisateurs, il sera difficile de trouver une place parmi les experts d'autant que le modèle favorise toujours ces derniers. C'est un problème classique dans les modèles à base de confiance globale. Néanmoins, nous estimons que l'effet de ce problème est réduit grâce à la contextualisation de la réputation, car un utilisateur ne peut pas être expert dans tous les domaines. D'un autre côté, ce modèle n'a pas de problème de démarrage à froid, et il est capable de générer des recommandations même pour un utilisateur qui n'a encore fait confiance à quiconque.

Le problème potentiel pour ce modèle, c'est le manque de robustesse face à des attaques de groupes. Si un groupe de profils s'allie pour augmenter le score d'un utilisateur dans un contexte donné, le modèle ne pourra pas détecter cet acte, et cet utilisateur sera privilégié par le modèle

en le considérant comme expert de ce contexte.

Notre troisième modèle proposé est un modèle de confiance locale à base d'opinions collectives. Contrairement au modèle de confiance locale, ce modèle s'appuie toujours sur les opinions de l'utilisateur et de ses amis, même quand l'utilisateur possède une opinion personnelle. Le problème du modèle à base de confiance locale est qu'il favorise les opinions personnelles qui sont souvent très incertaines parce qu'elles s'appuient souvent sur peu d'interactions. Les opinions collectives garantissent une meilleure précision parce qu'elles possèdent plus de preuves par rapport aux opinions individuelles. Ce modèle souffre du même problème de démarrage à froid que le modèle local, ce qui présente un avantage par rapport au modèle global sur ce point. D'un autre côté, gagner de la réputation est plus facile. Là où il se démarque aussi par rapport au modèle local, c'est parce qu'il encourage le regret et permet de rétablir et de corriger les relations de confiance.

Au niveau de la sécurité, il est théoriquement plus robuste que le modèle global aux attaques de groupe, car les interactions faites par un groupe de profils malveillants sont utilisées par le modèle seulement pour générer des recommandations pour les membres du groupe, car les opinions collectives ne donnent pas à l'utilisateur source l'accès à toutes les interactions de l'utilisateur cible, mais seulement aux interactions effectuées par ses amis (les amis de l'utilisateur source). Cela fait que l'utilisateur reste à l'abri de cette attaque tant qu'il n'a pas d'ami qui sont membres du groupe d'attaque.

Le bilan des trois modèles porte sur les qualités et les inconvénients de chacun, et cela comprend : le modèle global et le modèle collectif sont nettement plus précis que le modèle local. Le modèle global est faible face aux attaques de groupes, ce qui n'est pas le cas pour les deux autres modèles. Les modèles local et collectif sont plus transparents que le modèle global. Les modèles local et collectif souffrent du problème de démarrage à froid, alors que dans le modèle global, la difficulté pour les nouveaux utilisateurs est de construire leurs réputations. Le modèle local utilise peu de preuves dans ses décisions, le modèle collectif en utilise plus, et le modèle global en utilise encore plus. Les trois méthodes supportent une confiance qui favorise les preuves et les chemins les plus sûrs entre les utilisateurs des réseaux de confiance (entre les nœuds d'un graphe dirigé), nos résultats montrent que cette stratégie des chemins sûrs est plus pertinente que la stratégie des chemins les plus courts.

1.2 Perspectives

Dans tous les travaux dédiés à la modélisation de la confiance et qui prennent le contexte en compte, ce dernier est restreint à la tâche. Néanmoins, dans ce texte nous avons évoqué de l'importance et de l'urgence de la tâche comme composants du contexte non-exploités pour l'instant. Dans l'avenir nous envisageons d'explorer ces composants sur des données réelles, et mesurer leur rôle dans un scénario de confiance.

Notre conclusion concernant la sécurité et la robustesse des modèles est théorique, elle doit être soutenue par des épreuves pratiques (des tests sur des données réelles).

Dans la première partie de la contribution, nous défendons une analyse du comportement à base de caractéristiques qualitatives et quantitatives. Dans nos perspectives, nous comptons adapter cette méthode d'analyse à la deuxième partie de notre contribution, afin de mieux saisir les phénomènes comportementaux et mieux affiner les modèles proposés.

Faire confiance, en quelque sorte, c'est accepter de prendre un risque et de coopérer sans avoir une connaissance complète pour prédire les résultats. Cela fait que les modèles qui fonctionnent avec des données manquantes et avec peu de preuves sont les plus adaptés. La logique subjective

est un cadre riche qui possède encore beaucoup de qualités à exploiter dans ce domaine, et qui a été déjà adapté au traitement de la confiance, ainsi qu'à d'autres cadres de même genre (qui traitent les données manquantes) comme les méthodes d'imputation ou le modèle des chaînes de Markov cachées. Il est donc souhaitable de continuer à exploiter et à développer ce cadre et d'essayer d'adapter les autres.

Bibliographie

- [Abdul-Rahman and Hailes2000] A. Abdul-Rahman and S. Hailes. Supporting trust in virtual communities. In *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, 2000*, page 9 pp. vol.1, January 2000.
- [Adomavicius and Tuzhilin2004] Gediminas Adomavicius and Alex Tuzhilin. Recommendation technologies : Survey of current methods and possible extensions. Working paper, Stern School of Business, New York University, 2004. Information Systems Working Papers Series.
- [Almazro *et al.*2010] Dhoha Almazro, Ghadeer Shahatah, Lamia Albdulkarim, Mona Khrees, Romy Martinez, and William Nzoukou. A survey paper on recommender systems. arXiv e-print 1006.5278, June 2010.
- [Alpert2008] N. Alpert, J. Hajaj. We knew the web was big... 2008.
- [Alsabti *et al.*1997] Khaled Alsabti, Sanjay Ranka, and Vineet Singh. An efficient k-means clustering algorithm. *Electrical Engineering and Computer Science*, January 1997.
- [Amatriain *et al.*2009a] Xavier Amatriain, Neal Lathia, Josep M. Pujol, Haewoon Kwak, and Nuria Oliver. The wisdom of the few : a collaborative filtering approach based on expert opinions from the web. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval, SIGIR '09*, page 532–539, New York, NY, USA, 2009. ACM.
- [Amatriain *et al.*2009b] Xavier Amatriain, Josep M. Pujol, and Nuria Oliver. I like it... i like it not : Evaluating user ratings noise in recommender systems. In Geert-Jan Houben, Gord McCalla, Fabio Pianesi, and Massimo Zancanaro, editors, *User Modeling, Adaptation, and Personalization*, number 5535 in Lecture Notes in Computer Science, pages 247–258. Springer Berlin Heidelberg, January 2009.
- [Andersen and Fenandez-Luna2012] S. Andersen and Fenandez-Luna. On the use of weighted mean absolute error in recommender systems.pdf. In *RUE 2012 – Workshop on Recommendation Utility Evaluation : Beyond RMSE*, Dublin, Ireland, 2012.
- [Appelquist *et al.*2010] Daniel Appelquist, Dan Brickley, Melvin Carvahlo, Renato Iannella, Alexandre Passant, Christine Perey, and Henry Story. A standards-based, open and privacy-aware social web. *W3C Incubator Group Report*, 6, 2010.
- [Arrow1974] Kenneth J. Arrow. *The Limits of Organization*. 1974.
- [Artz and Gil2007] Donovan Artz and Yolanda Gil. A survey of trust in computer science and the semantic web. *Web Semantics : Science, Services and Agents on the World Wide Web*, 5(2) :58–71, 2007.
- [Audun2013] Jøsang Audun. Subjective logic (draft), 2013.
- [Avesani *et al.*2005] P. Avesani, P. Massa, and R. Tiella. A trust-enhanced recommender system application : Moleskiing. In *Proceedings of the 2005 ACM symposium on Applied computing*, page 1589–1593, 2005.

- [Baeza-Yates *et al.*1999] Ricardo Baeza-Yates, Berthier Ribeiro-Neto, et al. *Modern information retrieval*, volume 463. ACM press New York, 1999.
- [Balabanović and Shoham1997] Marko Balabanović and Yoav Shoham. Fab : content-based, collaborative recommendation. *Commun. ACM*, 40(3) :66–72, March 1997.
- [Bambini *et al.*2011] Riccardo Bambini, Paolo Cremonesi, and Roberto Turrin. A recommender system for an IPTV service provider : a real large-scale production environment. In Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors, *Recommender Systems Handbook*, pages 299–331. Springer US, January 2011.
- [Bank and Cole2008] Jacob Bank and Benjamin Cole. Calculating the jaccard similarity coefficient with map reduce for entity pairs in wikipedia. *Wikipedia Similarity Team*, 2008.
- [Basu *et al.*1998] Chumki Basu, Haym Hirsh, and William Cohen. Recommendation as classification : Using social and content-based information in recommendation. In *In Proceedings of the Fifteenth National Conference on Artificial Intelligence*, page 714–720. AAAI Press, 1998.
- [Bell *et al.*2007] Robert Bell, Yehuda Koren, and Chris Volinsky. Modeling relationships at multiple scales to improve accuracy of large recommender systems. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '07, page 95–104, New York, NY, USA, 2007. ACM.
- [Ben Ticha *et al.*2013] Sonia Ben Ticha, Azim Roussanaly, Anne Boyer, and Khaled Bsaïes. Feature frequency inverse user frequency for dependant attribute to enhance recommendations. pages 45–50, November 2013.
- [Bergholz2003] A. Bergholz. Coping with sparsity in a recommender system. *WEBKDD 2002-Mining Web Data for Discovering Usage Patterns and Profiles*, page 86–99, 2003.
- [Berners-Lee1992] T.J. Berners-Lee. The world-wide web. *Computer Networks and ISDN Systems*, 25(4–5) :454–459, November 1992.
- [Billsus and Pazzani2000] Daniel Billsus and Michael J. Pazzani. User modeling for adaptive news access. *User Modeling and User-Adapted Interaction*, 10(2-3) :147–180, 2000.
- [Birdsall2007] William F. Birdsall. Web 2.0 as a social movement, 2007. Web 2.0 research and development ; Right to communicate ; Web 2.0 as social movement ; Communication rights, Web sociology.
- [Bouquillion and Matthews2010] Philippe Bouquillion and Jacob Thomas Matthews. *Le Web collaboratif : mutations des industries de la culture et de la communication*. Presses universitaires de Grenoble, 2010.
- [Breese *et al.*1998] John S. Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithm for collaborative filtering. In *Proceedings of the 14 th Conference on Uncertainty in Artificial Intelligence*, page 43–52, 1998.
- [Bridge *et al.*2005] Derek Bridge, Mehmet H. Göker, Lorraine McGinty, and Barry Smyth. Case-based recommender systems. *Knowl. Eng. Rev.*, 20(3) :315–320, September 2005.
- [Brin and Page1998] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the seventh international conference on World Wide Web 7, WWW7*, page 107–117, Amsterdam, The Netherlands, The Netherlands, 1998. Elsevier Science Publishers B. V.
- [Brun *et al.*2011] Armelle Brun, Sylvain Castagnos, and Anne Boyer. From community detection to mentor selection in rating-free collaborative filtering. *Advances in Multimedia*, 2011, March 2011.

-
- [Burke *et al.*2005] R. Burke, B. Mobasher, R. Zabicki, and R. Bhaumik. Identifying attack models for secure recommendation. In *Beyond Personalization : A Workshop on the Next Generation of Recommender Systems*, 2005.
- [Burke1999a] Robin Burke. *Integrating Knowledge-based and Collaborative-filtering Recommender Systems*. 1999.
- [Burke1999b] Robin Burke. The wasabi personal shopper : a case-based recommender system. In *AAAI/IAAI*, pages 844–849, 1999.
- [Burke2002] Robin Burke. Hybrid recommender systems : Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4) :331–370, November 2002.
- [Burke2007] R. Burke. Hybrid web recommender systems. *The adaptive web*, page 377–408, 2007.
- [Castagnos2008] Sylvain Castagnos. *Modélisation de comportements et apprentissage stochastique non supervisée de stratégies d’interactions sociales au sein de systèmes temps réel de recherche et d’accès à l’information*. PhD thesis, Nancy 2, Nancy, 2008.
- [Castaldo2007] Sandro Castaldo. *Trust in Market Relationships*. Edward Elgar Publishing, 2007.
- [Cerutti *et al.*2013] Federico Cerutti, Alice Toniolo, Nir Oren, and Timothy J. Norman. Context-dependent trust decisions with subjective logic. arXiv e-print 1309.4994, September 2013.
- [Chen *et al.*2010] Zhimin Chen, yi Jiang, and Yao Zhao. A collaborative filtering recommendation algorithm based on user interest change and trust evaluation.pdf. *JDCTA*, pages 106–113, 2010.
- [Cho *et al.*2002] Yoon Ho Cho, Jae Kyeong Kim, and Soung Hie Kim. A personalized recommender system based on web usage mining and decision tree induction. *Expert Systems with Applications*, 23(3) :329–342, October 2002.
- [Christianson and Harbison] B. Christianson and W. Harbison. Why isn’t trust transitive? In *Security Protocols*, page 171–176.
- [Cofta2007] Piotr Cofta. *Trust, complexity and control confidence in a convergent world*. [USA] : Wiley, [S.l.], 2007.
- [Cremonesi *et al.*2010] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 39–46. ACM, 2010.
- [Croft *et al.*2009] Bruce Croft, Donald Metzler, and Trevor Strohman. *Search Engines : Information Retrieval in Practice*. Addison-Wesley Publishing Company, USA, 1st edition, 2009.
- [Davies and Bouldin1979] David L. Davies and Donald W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2) :224–227, 1979.
- [De Gemmis *et al.*2009] Marco De Gemmis, Leo Iaquinta, Pasquale Lops, Cataldo Musto, Fedelucio Narducci, and Giovanni Semeraro. Preference learning in recommender systems. *PREFERENCE LEARNING*, 41, 2009.
- [de Kunder2006] Maurice de Kunder. *{G}eschatte {G}rootte van het {G}e\{|{i}\}ndexeerde {W}orld {W}ide {W}eb*. PhD thesis, Tilburg University, 2006.
- [Delgado and Ishii1999] Joaquin Delgado and Naohiro Ishii. Memory-based weighted majority prediction. In *SIGIR Workshop Recomm. Syst. Citeseer*. Citeseer, 1999.
- [Dempster2008] A.P. Dempster. The Dempster–Shafer calculus for statisticians. *International Journal of Approximate Reasoning*, 48(2) :365–377, 2008.

- [Deshpande and Karypis2004] Mukund Deshpande and George Karypis. Item-based top- N recommendation algorithms. *ACM Trans. Inf. Syst.*, 22(1) :143–177, January 2004.
- [Desrosiers and Karypis2011] Christian Desrosiers and George Karypis. A comprehensive survey of neighborhood-based recommendation methods. In Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors, *Recommender Systems Handbook*, pages 107–144. Springer US, January 2011.
- [Deutsch1977] Morton Deutsch. *The Resolution of Conflict : Constructive and Destructive Processes*. Yale University Press, 1977.
- [Elmisery and Botvich2012] A. M. Elmisery and D. Botvich. Multi-agent based middleware for protecting privacy in IPTV content recommender services. *Multimedia Tools and Applications*, page 1–27, 2012.
- [Esslimani et al.2010] I. Esslimani, A. Brun, and A. Boyer. Detecting leaders in behavioral networks. In *2010 International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 281–285, 2010.
- [Esslimani2010] Ilham Esslimani. *Vers une approche comportementale de recommandation : apport de l'analyse des usages dans un processus de personnalisation*. PhD thesis, Université Nancy II, 2010.
- [Fox1974] Alan Fox. *Beyond contract : Work, power and trust relations*. Faber London, 1974.
- [Fritzke and others1995] Bernd Fritzke et al. A growing neural gas network learns topologies. *Advances in neural information processing systems*, 7 :625–632, 1995.
- [Gabarro1987] John J Gabarro. *The dynamics of taking charge*. Harvard Business Press, 1987.
- [Gambetta1988] Diego Gambetta. *Trust : Making and Breaking Cooperative Relations*. Blackwell, 1988.
- [Ginsberg1988] Matthew L. Ginsberg. Multivalued logics : a uniform approach to reasoning in artificial intelligence. *Computational Intelligence*, 4(3) :265–316, 1988.
- [Golbeck and Hendler2006] J. Golbeck and J. Hendler. Filmtrust : Movie recommendations using trust in web-based social networks. In *Proceedings of the IEEE Consumer communications and networking conference*, volume 96, 2006.
- [Golbeck et al.2003] Jennifer Golbeck, Bijan Parsia, and James Hendler. Trust networks on the semantic web. In *In Proceedings of Cooperative Intelligent Agents*, page 238–249, 2003.
- [Golbeck2005] Jennifer Ann Golbeck. *Computing and Applying Trust in Web-based Social Networks*. 2005.
- [Golbeck2009] Jennifer Golbeck, editor. *Computing with Social Trust*. Human-Computer Interaction Series. Springer, 2009.
- [Goldberg et al.1992] David Goldberg, David Nichols, Brian M. Oki, and Douglas Terry. Using collaborative filtering to weave an information tapestry. *Commun. ACM*, 35(12) :61–70, 1992.
- [Gong et al.1990] Li Gong, R. Needham, and R. Yahalom. Reasoning about belief in cryptographic protocols. In *, 1990 IEEE Computer Society Symposium on Research in Security and Privacy, 1990. Proceedings*, pages 234–248, 1990.
- [Graham2005] Paul Graham. Web 2.0. *Consultado (21/12/2008) en : <http://www.nosolousabilidad.com/articulos/Web20.htm>*, 2005.
- [Grčar et al.2006] Miha Grčar, Blaž Fortuna, Dunja Mladenič, and Marko Grobelnik. kNN versus SVM in the collaborative filtering framework. In Prof Dr Vladimir Batagelj, Prof Dr

-
- Hans-Hermann Bock, Prof Dr Anuška Ferligoj, and Aleš Žiberna, editors, *Data Science and Classification*, Studies in Classification, Data Analysis, and Knowledge Organization, pages 251–260. Springer Berlin Heidelberg, January 2006.
- [Gruber2008] Tom Gruber. Collective knowledge systems : Where the social web meets the semantic web. *Web Semantics : Science, Services and Agents on the World Wide Web*, 6(1) :4–13, 2008.
- [Guha2003] R. Guha. *Open Rating Systems*. 2003.
- [Guo et al.2014] Guibing Guo, Jie Zhang, and Daniel Thalmann. Merging trust in collaborative filtering to alleviate data sparsity and cold start. 57 :57–68, 2014.
- [Halkidi et al.2001] Maria Halkidi, Yannis Batistakis, and Michalis Vazirgiannis. On clustering validation techniques. *Journal of Intelligent Information Systems*, 17 :107–145, 2001.
- [Haydar et al.2012a] Charif Haydar, Anne Boyer, and Azim Roussanally. Hybridising collaborative filtering and trust-aware recommender systems. In *8th International Conference on Web Information Systems and Technologies - WEBIST'2012*, Porto, Portugal, 2012.
- [Haydar et al.2012b] Charif Haydar, Azim Roussanally, and Anne Boyer. Analyzing recommender system’s performance fluctuations across users. In Gerald Quirchmayr, Josef Basl, Ilsun You, Lida Xu, and Edgar Weippl, editors, *Multidisciplinary Research and Practice for Information Systems*, number 7465 in Lecture Notes in Computer Science, pages 390–402. Springer Berlin Heidelberg, January 2012.
- [Haydar et al.2012c] Charif Haydar, Azim Roussanally, and Anne Boyer. Clustering users to explain recommender systems’ performance fluctuation. In Li Chen, Alexander Felfernig, Jiming Liu, and Zbigniew W. Raś, editors, *Foundations of Intelligent Systems*, number 7661 in Lecture Notes in Computer Science, pages 357–366. Springer Berlin Heidelberg, January 2012.
- [Haydar et al.2013a] Charif Haydar, Azim Roussanally, and Anne Boyer. Individual opinions versus collective opinions in trust modelling. pages 92–99, November 2013. 00000.
- [Haydar et al.2013b] Charif Haydar, Azim Roussanally, and Anne Boyer. Local trust versus global trust networks in subjective logic. In *2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, volume 1, pages 29–36, 2013. 00000.
- [Herlocker et al.1999] Jonathan L. Herlocker, Joseph A. Konstan, Al Borchers, and John Riedl. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR ’99, page 230–237, New York, NY, USA, 1999. ACM.
- [Herlocker et al.2002] Jon Herlocker, Joseph A. Konstan, and John Riedl. An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Information Retrieval*, 5(4) :287–310, October 2002.
- [Herlocker et al.2004] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, John, and T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 22 :5–53, 2004.
- [Herzig et al.2010] Andreas Herzig, Emiliano Lorini, Jomi F. Hübner, and Laurent Vercouter. A logic of trust and reputation. *Logic Journal of IGPL*, 18(1) :214–244, February 2010.
- [Hill et al.1995] Will Hill, Larry Stead, Mark Rosenstein, and George Furnas. Recommending and evaluating choices in a virtual community of use. In *Proceedings of the SIGCHI Conference*

- on *Human Factors in Computing Systems*, CHI '95, page 194–201, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co.
- [Hoffman *et al.*2009] Kevin Hoffman, David Zage, and Cristina Nita-Rotaru. A survey of attack and defense techniques for reputation systems. *ACM Comput. Surv.*, 42(1) :1 :1–1 :31, 2009.
- [Hofmann2003] Thomas Hofmann. Collaborative filtering via gaussian probabilistic latent semantic analysis. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, SIGIR '03, page 259–266, New York, NY, USA, 2003. ACM.
- [Huynh *et al.*2006] Trung Dong Huynh, Nicholas R. Jennings, and Nigel R. Shadbolt. An integrated trust and reputation model for open multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 13(2) :119–154, September 2006.
- [Iaquinta *et al.*] L. Iaquinta, M. de Gemmis, P. Lops, G. Semeraro, M. Filannino, and P. Molino. Introducing serendipity in a content-based recommender system. In *Eighth International Conference on Hybrid Intelligent Systems, 2008. HIS '08*, pages 168–173.
- [Jamali and Ester2009] Mohsen Jamali and Martin Ester. Using a trust network to improve top-n recommendation. In *Proceedings of the third ACM conference on Recommender systems*, RecSys '09, page 181–188, New York, NY, USA, 2009. ACM.
- [Jones2010] Nicolas Jones. *User Perceived Qualities and Acceptance of Recommender Systems : The Role of Diversity*. PhD thesis, ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE, 2010.
- [Jøsang and Pope2005] A. Jøsang and S. Pope. Semantic constraints for trust transitivity. In *Proceedings of the 2nd Asia-Pacific conference on Conceptual modelling-Volume 43*, page 59–68, 2005.
- [Jøsang *et al.*2003] Audun Jøsang, Elisabeth Gray, and Michael Kinateder. Analysing topologies of transitive trust. pages 9–22, 2003.
- [Jøsang *et al.*2006] A. Jøsang, R. Hayward, and S. Pope. Trust network analysis with subjective logic. In *Proceedings of the 29th Australasian Computer Science Conference-Volume 48*, page 85–94, 2006.
- [Jøsang1996] Audun Jøsang. The right type of trust for distributed systems. In *Proceedings of the 1996 workshop on New security paradigms*, NSPW '96, page 119–131, New York, NY, USA, 1996. ACM.
- [Jøsang2001] A. Jøsang. A logic for uncertain probabilities. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9(03) :279–311, 2001.
- [Kagal *et al.*2001] L. Kagal, T. Finin, and A. Joshi. Trust-based security in pervasive computing environments. *Computer*, 34(12) :154–157, 2001.
- [Kamvar *et al.*2003] Sepandar D. Kamvar, Mario T. Schlosser, and Hector Garcia-Molina. The eigentrust algorithm for reputation management in P2P networks. In *Proceedings of the 12th international conference on World Wide Web*, WWW '03, page 640–651, New York, NY, USA, 2003. ACM.
- [Kohavi *et al.*2009] Ron Kohavi, Roger Longbotham, Dan Sommerfield, and Randal M. Henne. Controlled experiments on the web : survey and practical guide. *Data Mining and Knowledge Discovery*, 18(1) :140–181, February 2009.
- [Konstan *et al.*1997] Joseph A. Konstan, Bradley N. Miller, David Maltz, Jonathan L. Herlocker, Lee R. Gordon, and John Riedl. GroupLens : applying collaborative filtering to usenet news. *Commun. ACM*, 40(3) :77–87, March 1997.

-
- [Koren and Bell2011] Yehuda Koren and Robert Bell. Advances in collaborative filtering. In Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors, *Recommender Systems Handbook*, pages 145–186. Springer US, January 2011.
- [Koren *et al.*] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. 42(8) :30–37.
- [Koren2008] Yehuda Koren. Factorization meets the neighborhood : a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '08, page 426–434, New York, NY, USA, 2008. ACM.
- [Kouki2012] Alejandro Bellogín Kouki. *Recommender System Performance Evaluation and Prediction : An Information Retrieval Perspective*. PhD thesis, Universidad Autónoma de Madrid, 2012.
- [Kramer and Tyler] Roderick M. Kramer and Tom R. Tyler. *Trust in Organizations : Frontiers of Theory and Research*. SAGE Publications.
- [Lü and Zhou2011] Linyuan Lü and Tao Zhou. Link prediction in complex networks : A survey. *Physica A : Statistical Mechanics and its Applications*, 390(6) :1150–1170, March 2011.
- [Lü *et al.*2012] Linyuan Lü, Matúš Medo, Chi Ho Yeung, Yi-Cheng Zhang, Zi-Ke Zhang, and Tao Zhou. Recommender systems. *Physics Reports*, 519(1) :1–49, October 2012.
- [Lam and Riedl2004] S. K. Lam and J. Riedl. Shilling recommender systems for fun and profit. In *Proceedings of the 13th international conference on World Wide Web*, page 393–402, 2004.
- [Lee and Brusilovsky2009] D. H. Lee and P. Brusilovsky. Does trust influence information similarity? *Recommender Systems & the Social Web*, page 10, 2009.
- [Levien2002] Raphael L. Levien. *Attack resistant trust metrics. (phD draft)*. PhD thesis, UNIVERSITY of CALIFORNIA, 2002.
- [Levien2004] Raph Levien. Attack resistant trust metrics. Technical report, 2004.
- [Lewis and Weigert1985] J David Lewis and Andrew Weigert. Trust as a social reality. *Social forces*, 63(4) :967–985, 1985.
- [Li and Ping2009] Wenjuan Li and Lingdi Ping. Trust model to enhance security and interoperability of cloud environment. In Martin Gilje Jaatun, Gansen Zhao, and Chunming Rong, editors, *Cloud Computing*, number 5931 in Lecture Notes in Computer Science, pages 69–79. Springer Berlin Heidelberg, January 2009.
- [Li *et al.*2003] Ninghui Li, Benjamin N Grosz, and Joan Feigenbaum. Delegation logic : A logic-based approach to distributed authorization. *ACM Transactions on Information and System Security (TISSEC)*, 6(1) :128–171, 2003.
- [Linden *et al.*2003] G. Linden, B. Smith, and J. York. Amazon.com recommendations : item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1) :76–80, 2003.
- [Loh *et al.*2008] S. Loh, F. Lorenzi, G. Simões, L. K. Wives, and J. P. M. de Oliveira. Comparing keywords and taxonomies in the representation of users profiles in a content-based recommender system. In *Proceedings of the 2008 ACM symposium on Applied computing*, page 2030–2034, 2008.
- [Lousame and Sánchez2009] Fabián P. Lousame and Eduardo Sánchez. A taxonomy of collaborative-based recommender systems. In Giovanna Castellano, Lakhmi C. Jain, and Anna Maria Fanelli, editors, *Web Personalization in Intelligent Environments*, number 229 in Studies in Computational Intelligence, pages 81–117. Springer Berlin Heidelberg, January 2009.

- [Lü *et al.*2012] Linyuan Lü, Matúš Medo, Chi Ho Yeung, Yi-Cheng Zhang, Zi-Ke Zhang, and Tao Zhou. Recommender systems. *Physics Reports*, 519(1) :1–49, 2012.
- [luh2013] Niklas luhmann, la confiance, un mécanisme de réduction de la complexité sociale, January 2013.
- [Ma *et al.*2009] H. Ma, M. R. Lyu, and I. King. Learning to recommend with trust and distrust relationships. In *Proceedings of the third ACM conference on Recommender systems*, page 189–196, 2009.
- [Ma *et al.*2011] Hao Ma, Dengyong Zhou, Chao Liu, Michael R. Lyu, and Irwin King. Recommender systems with social regularization. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, WSDM '11*, pages 287–296, New York, NY, USA, 2011. ACM.
- [Maes *et al.*1999] Pattie Maes, Robert H. Guttman, and Alexandros G. Moukas. Agents that buy and sell. *Commun. ACM*, 42(3) :81–ff., March 1999.
- [Mahmood and Ricci2009] Tariq Mahmood and Francesco Ricci. Improving recommender systems with adaptive conversational strategies. In *Proceedings of the 20th ACM conference on Hypertext and hypermedia, HT '09*, page 73–82, New York, NY, USA, 2009. ACM.
- [Maltz and Ehrlich1995] David Maltz and Kate Ehrlich. Pointing the way : active collaborative filtering. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '95*, page 202–209, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co.
- [Marsh and Briggs2009] Stephen Marsh and Pamela Briggs. Examining trust, forgiveness and regret as computational concepts. In Jennifer Golbeck, editor, *Computing with Social Trust, Human–Computer Interaction Series*, pages 9–43. Springer London, January 2009.
- [Martínez *et al.*2008] Luis Martínez, Manuel J. Barranco, Luis G. Pérez, and Macarena Espinilla. A knowledge based recommender system with multigranular linguistic information. *International Journal of Computational Intelligence Systems*, 1(3) :225–236, 2008.
- [Massa and Avesani2004] P. Massa and P. Avesani. Trust-aware collaborative filtering for recommender systems. *On the Move to Meaningful Internet Systems 2004 : CoopIS, DOA, and ODBASE*, page 492–508, 2004.
- [Massa and Avesani2007] Paolo Massa and Paolo Avesani. Trust-aware recommender systems. In *Proceedings of the 2007 ACM conference on Recommender systems, RecSys '07*, page 17–24. ACM, 2007.
- [Massa and Bhattacharjee2004] P. Massa and B. Bhattacharjee. Using trust in recommender systems : an experimental analysis. *Trust Management*, page 221–235, 2004.
- [Mcknight and Chervany1996] D. Harrison Mcknight and Norman L. Chervany. The meanings of trust. Technical report, 1996.
- [Melville *et al.*2002] P. Melville, R. J. Mooney, and R. Nagarajan. Content-boosted collaborative filtering for improved recommendations. In *Proceedings of the National Conference on Artificial Intelligence*, page 187–192, 2002.
- [Miller *et al.*2004] B. N. Miller, J. A. Konstan, and J. Riedl. PocketLens : toward a personal recommender system. *ACM Transactions on Information Systems (TOIS)*, 22(3) :437–476, 2004.
- [Mishra *et al.*1996] Aneil K Mishra, R Kramer, and Tom R Tyler. Organizational responses to crisis : The centrality of trust, 1996.

-
- [Mobasher *et al.*2004] Bamshad Mobasher, Xin Jin, and Yanzan Zhou. Semantically enhanced collaborative filtering on the web. In Bettina Berendt, Andreas Hotho, Dunja Mladenic, Maarten van Someren, Myra Spiliopoulou, and Gerd Stumme, editors, *Web Mining : From Web to Semantic Web*, number 3209 in Lecture Notes in Computer Science, pages 57–76. Springer Berlin Heidelberg, January 2004.
- [Mobasher *et al.*2006] Bamshad Mobasher, Robin Burke, and Jeff J Sandvig. Model-based collaborative filtering as a defense against profile injection attacks. In *AAAI*, volume 6, page 1388, 2006.
- [Mobasher *et al.*2007] Bamshad Mobasher, Robin Burk, Runa Bhaumik, and Chad Williams. Toward trustworthy recommender systems an analysis of attack models and algorithm robustness.pdf. 7(4), 2007.
- [Montaner *et al.*2002] M. Montaner, B. López, and J. L. de la Rosa. Developing trust in recommender agents. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems : part 1*, page 304–305, 2002.
- [Mui2002] L. Mui. *Computational Models of Trust and Reputation : Agents, Evolutionary Games, and Social Networks*. PhD Thesis, Massachusetts Institute of Technology, 2002.
- [Nakamura and Abe1998] Atsuyoshi Nakamura and Naoki Abe. Collaborative filtering using weighted majority prediction algorithms. In *ICML*, volume 98, pages 395–403, 1998.
- [Nejdl *et al.*2004] Wolfgang Nejdl, Daniel Olmedilla, and Marianne Winslett. PeerTrust : automated trust negotiation for peers on the semantic web. In Willem Jonker and Milan Petković, editors, *Secure Data Management*, number 3178 in Lecture Notes in Computer Science, pages 118–132. Springer Berlin Heidelberg, January 2004.
- [Oard and Kim1998] Douglas Oard and Jinmook Kim. Implicit feedback for recommender systems. *Faculty Publications*, pages 81–83, January 1998.
- [O’Donovan and Smyth2005] J. O’Donovan and B. Smyth. Trust in recommender systems. In *Proceedings of the 10th international conference on Intelligent user interfaces*, page 167–174, 2005.
- [O’Mahony *et al.*2004] Michael O’Mahony, Neil Hurley, Nicholas Kushmerick, and Guénolé Silvestre. Collaborative recommendation : A robustness analysis. *ACM Trans. Internet Technol.*, 4(4) :344–377, November 2004.
- [O’Reilly2005] Tim O’Reilly. What is web 2.0, 2005.
- [Page *et al.*1999] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. *The PageRank Citation Ranking : Bringing Order to the Web*. 1999.
- [Paolucci *et al.*2003] Massimo Paolucci, Katia P Sycara, Takuya Nishimura, and Naveen Srinivasan. Using daml-s for p2p discovery. In *ICWS*, pages 203–207. Citeseer, 2003.
- [Paterrek2007] Arkadiusz Paterrek. Improving regularized singular value decomposition for collaborative filtering. In *Proceedings of KDD cup and workshop*, volume 2007, pages 5–8, 2007.
- [Pazzani and Billsus2007] Michael J. Pazzani and Daniel Billsus. Content-based recommendation systems. In Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl, editors, *The Adaptive Web*, number 4321 in Lecture Notes in Computer Science, pages 325–341. Springer Berlin Heidelberg, January 2007.
- [Pazzani1999] Michael J. Pazzani. A framework for collaborative, content-based and demographic filtering. *Artif. Intell. Rev.*, 13(5-6) :393–408, 1999.

- [Perugini and Gonçalves2002] Saverio Perugini and Marcos André Gonçalves. Recommendation and personalization : a survey. Technical report, 2002.
- [Pitsilis and Knapskog2012] G. Pitsilis and S. J. Knapskog. Social trust as a solution to address sparsity-inherent problems of recommender systems. *arXiv preprint arXiv :1208.1004*, 2012.
- [Prasad2012] Rvsv Prasad. A categorical review of recommender systems. *International Journal of Distributed and Parallel systems*, 3(5) :73–83, September 2012.
- [Ramchurn *et al.*2004] Sarvapali D. Ramchurn, Dong Huynh, and Nicholas R. Jennings. Trust in multi-agent systems. *The Knowledge Engineering Review*, 19 :1–25, 3 2004.
- [Resnick and Varian1997] P. Resnick and H. R. Varian. Recommender systems. *Communications of the ACM*, 40(3) :56–58, 1997.
- [Resnick and Zeckhauser2002] Paul Resnick and Richard Zeckhauser. Trust among strangers in internet transactions : Empirical analysis of eBay’ s reputation system. *Advances in Applied Microeconomics*, 11 :127–157, October 2002.
- [Resnick *et al.*1994] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. GroupLens : an open architecture for collaborative filtering of netnews. page 175–186. ACM Press, 1994.
- [Ricci *et al.*2006] Francesco Ricci, Dario Cavada, Nader Mirzadeh, and Adriano Venturini. Case-based travel recommendations. *Destination Recommendation Systems : Behavioural Foundations and Applications*, pages 67–93, 2006.
- [Ricci *et al.*2011] Francesco Ricci, Lior Rokach, and Bracha Shapira. Introduction to recommender systems handbook. In Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors, *Recommender Systems Handbook*, pages 1–35. Springer US, January 2011.
- [Richardson *et al.*2003] Matthew Richardson, Rakesh Agrawal, and Pedro Domingos. Trust management for the semantic web. In *In proceedings of the second international semantic web conference*, page 351–368, 2003.
- [Rotter1980] Julian B Rotter. Interpersonal trust, trustworthiness, and gullibility. *American psychologist*, 35(1) :1, 1980.
- [Salakhutdinov *et al.*2007] Ruslan Salakhutdinov, Andriy Mnih, and Geoffrey Hinton. Restricted boltzmann machines for collaborative filtering. In *Proceedings of the 24th international conference on Machine learning, ICML ’07*, page 791–798, New York, NY, USA, 2007. ACM.
- [Sarwar *et al.*2000] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Analysis of recommendation algorithms for e-commerce. In *Proceedings of the 2nd ACM conference on Electronic commerce, EC ’00*, page 158–167, New York, NY, USA, 2000. ACM.
- [Sarwar *et al.*2001] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, page 285–295, 2001.
- [Schafer *et al.*2007] J. Schafer, D. Frankowski, J. Herlocker, and S. Sen. Collaborative filtering recommender systems. *The adaptive web*, page 291–324, 2007.
- [Schein *et al.*2001] Andrew I Schein, Alexandrin Popescul, Lyle H Ungar, and David M Pennock. Generative models for cold-start recommendations. In *Proceedings of the 2001 SIGIR Workshop on Recommender Systems*, volume 6. Citeseer, 2001.
- [Schwartz2004] Barry Schwartz. *The paradox of choice : why more is less*. Ecco, New York, 2004.

-
- [Shani and Gunawardana2011] Guy Shani and Asela Gunawardana. Evaluating recommendation systems. In Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors, *Recommender Systems Handbook*, pages 257–297. Springer US, January 2011.
- [Shapiro1987] Susan P Shapiro. The social control of impersonal trust. *American journal of Sociology*, pages 623–658, 1987.
- [Shardanand and Maes1995] Upendra Shardanand and Pattie Maes. Social information filtering : algorithms for automating “word of mouth”:. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '95, page 210–217, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co.
- [Shi et al.2006] Lei Shi, Zhimin Gu, Lin Wei, and Yun Shi. An applicative study of zipf's law on web cache. *International Journal of Information Technology*, 12(4) :49–58, 2006.
- [Shin et al.2008] H. Shin, N. Y. Kim, E. Y. Kim, and M. Lee. Behaviors-based user profiling and classification-based content rating for personalized digital TV. In *Consumer Electronics, 2008. ICCE 2008. Digest of Technical Papers. International Conference on*, page 1–2, 2008.
- [Smyth B. and Cotter P.2000] Smyth B. and Cotter P. A personalised TV listings service for the digital TV age. *Knowledge-Based Systems*, 13(2) :53–59, 2000.
- [Steck2011] Harald Steck. Item popularity and recommendation accuracy. In *Proceedings of the fifth ACM conference on Recommender systems*, RecSys '11, page 125–132, New York, NY, USA, 2011. ACM.
- [Su and Khoshgoftaar2009] Xiaoyuan Su and Taghi M. Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, 2009 :1–19, 2009.
- [Takacs et al.2008] G. Takacs, I. Pílaszy, B. Nemeth, and Domonkos Tikk. Investigation of various matrix factorization methods for large recommender systems. In *IEEE International Conference on Data Mining Workshops, 2008. ICDMW '08*, pages 553–562, 2008.
- [Takács et al.2007] Gábor Takács, István Pílaszy, Bottyán Németh, and Domonkos Tikk. Major components of the gravity recommendation system. *SIGKDD Explor. Newsl.*, 9(2) :80–83, 2007.
- [Tang et al.2005] Wen Tang, Yun-Xiao Ma, and Zhong Chen. Managing trust in peer-to-peer networks. *Journal of Digital Information Management*, 3(2) :58, June 2005.
- [Terveen et al.1997] Loren Terveen, Will Hill, Brian Amento, David McDonald, and Josh Creter. PHOAKS : a system for sharing recommendations. *Commun. ACM*, 40(3) :59–62, March 1997.
- [Tintarev and Masthoff2007] N. Tintarev and J. Masthoff. A survey of explanations in recommender systems. In *2007 IEEE 23rd International Conference on Data Engineering Workshop*, pages 801–810, 2007.
- [Towle and Quinn2000] B. Towle and C. Quinn. Knowledge based recommender systems using explicit user models. In *Papers from the AAI Workshop, AAI Technical Report WS-00-04*, pages 74–77. Menlo Park, CA : AAI Press, 2000.
- [Trewin2000] S. Trewin. Knowledge-based recommender systems. *Encyclopedia of Library and Information Science : Volume 69-Supplement 32*, 69 :180, 2000.
- [Van Meteren and Van Someren2000] R. Van Meteren and M. Van Someren. Using content-based filtering for recommendation. In *Proceedings of the Machine Learning in the New Information Age : MLnet/ECML2000 Workshop*, 2000.
- [Vekariya and Kulkarni2012] V. Vekariya and G. R. Kulkarni. Notice of violation of IEEE publication principles< BR> hybrid recommender systems : Survey and experiments. In *Digital*

- Information and Communication Technology and its Applications (DICTAP), 2012 Second International Conference on*, page 469–473, 2012.
- [Victor *et al.*2009] Patricia Victor, Chris Cornelis, Martine De Cock, and Paulo Pinheiro Da Silva B. Gradual trust and distrust in recommender systems. fuzzy sets and systems. In *In press*, 2009.
- [Voorhees2002] EllenM. Voorhees. The philosophy of information retrieval evaluation. In Carol Peters, Martin Braschler, Julio Gonzalo, and Michael Kluck, editors, *Evaluation of Cross-Language Information Retrieval Systems*, volume 2406 of *Lecture Notes in Computer Science*, pages 355–370. Springer Berlin Heidelberg, 2002.
- [Vozalis and Margaritis2003] E. Vozalis and K. G. Margaritis. Analysis of recommender systems algorithms. In *Proceedings of the 6th Hellenic European Conference on Computer Mathematics and its Applications (HERCMA-2003), Athens, Greece*, 2003.
- [Walter *et al.*2008] Frank Edward Walter, Stefano Battiston, and Frank Schweitzer. A model of a trust-based recommendation system on a social network. *Autonomous Agents and Multi-Agent Systems*, 16(1) :57–74, February 2008.
- [Wang and Gui2013] Gang Wang and Xiaolin Gui. Dynamic recommendation trust model based on information entropy and heuristic rules in e-commerce environment. *Electronics and Electrical Engineering*, 19(4) :71–76, October 2013.
- [Wang *et al.*2006] Jun Wang, Arjen P. de Vries, and Marcel J. T. Reinders. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '06, page 501–508, New York, NY, USA, 2006. ACM.
- [Watier2008] Patrick Watier. *Eloge de la confiance*. Belin, February 2008.
- [Wei *et al.*2007] Kangning Wei, Jinghua Huang, and Shaohong Fu. A survey of e-commerce recommender systems. In *2007 International Conference on Service Systems and Service Management*, pages 1–5, 2007.
- [Weick and Roberts1993] Karl E Weick and Karlene H Roberts. Collective mind in organizations : Heedful interrelating on flight decks. *Administrative science quarterly*, pages 357–381, 1993.
- [Williams *et al.*2007] Chad A. Williams, Bamshad Mobasher, and Robin Burke. Defending recommender systems : detection of profile injection attacks. *Service Oriented Computing and Applications*, 1(3) :157–170, November 2007.
- [Wray2009] Richard Wray. Internet data heads for 500bn gigabytes, May 2009.
- [Xue *et al.*2005] Gui-Rong Xue, Chenxi Lin, Qiang Yang, WenSi Xi, Hua-Jun Zeng, Yong Yu, and Zheng Chen. Scalable collaborative filtering using cluster-based smoothing. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '05, page 114–121, New York, NY, USA, 2005. ACM.
- [Yager *et al.*1994] Ronald R. Yager, Janusz Kacprzyk, and Mario Fedrizzi, editors. *Advances in the Dempster-Shafer theory of evidence*. John Wiley & Sons, Inc., New York, NY, USA, 1994.
- [Yu *et al.*] B. Yu, M. P. Singh, and K. Sycara. Developing trust in large-scale peer-to-peer systems. In *Multi-Agent Security and Survivability, 2004 IEEE First Symposium on*, page 1–10.

-
- [Yu *et al.*2004] K. Yu, A. Schwaighofer, V. Tresp, Xiaowei Xu, and H.-P. Kriegel. Probabilistic memory-based collaborative filtering. *IEEE Transactions on Knowledge and Data Engineering*, 16(1) :56–69, 2004. 00160.
- [Zeng *et al.*2010] Wei Zeng, Ming-Sheng Shang, Qian-Ming Zhang, Linyuan Lü, and Tao Zhou. Can dissimilar users contribute to accuracy and diversity of personalized recommendation? *International Journal of Modern Physics C*, 21(10) :1217–1227, October 2010.
- [Zeng *et al.*2011] Wei Zeng, Yu-Xiao Zhu, Linyuan Lü, and Tao Zhou. Negative ratings play a positive role in information filtering. *Physica A : Statistical Mechanics and its Applications*, 390(23–24) :4486–4493, November 2011.
- [Zhang *et al.*2008] Yu Zhang, Hua-jun CHEN, Xiao-hong JIANG, Hao SHENG, Tong YU, and Lin-hua ZHOU. A survey of trust management for e-commerce systems [j]. *Acta Electronica Sinica*, 10 :026, 2008.
- [Zhang *et al.*2009] Jinbo Zhang, ZhiQing Lin, Bo Xiao, and Chuang Zhang. An optimized item-based collaborative filtering recommendation algorithm, 2009.
- [Ziegler and Lausen2004] C. N. Ziegler and G. Lausen. Spreading activation models for trust propagation. In *e-Technology, e-Commerce and e-Service, 2004. EEE'04. 2004 IEEE International Conference on*, page 83–97, 2004.
- [Zitnick and Kanade2012] Lawrence Zitnick and Takeo Kanade. Maximum entropy for collaborative filtering. arXiv e-print 1207.4152, July 2012.
- [Zuva *et al.*2012] Tranos Zuva, Sunday O.Ojo, Seleman Ngwira, and Keneilwe Zuva. A survey of recommender systems techniques, challenges and evaluation metrics. *International Journal of Emerging Technology and Advanced Engineering*, 2012.

Résumé

La divergence comportementale des utilisateurs sur le web résulte un problème de fluctuation de performance chez les systèmes de recommandation (SR) qui exploitent ce comportement pour recommander aux utilisateurs des items qu'ils vont apprécier. Ce problème est observé dans l'approche de filtrage collaboratif (FC) qui exploite les notes attribuées par les utilisateurs aux items, et l'approche à base de confiance (SRC) qui exploite les notes de confiance que les utilisateurs attribuent l'un à l'autre. Nous proposons une approche hybride qui augmente le nombre d'utilisateurs bénéficiant de la recommandation, sans perte significative de précision. Par la suite, nous identifions plusieurs caractéristiques comportementales qui permettent de constituer un profil comportemental de l'utilisateur. Ce qui nous permet de classer les utilisateurs selon leur comportement commun, et d'observer la performance de chaque approche par classe.

Par la suite, nous focalisons sur les SRC. Le concept de confiance a été abordé dans plusieurs disciplines. Il n'existe pas véritablement de consensus sur sa définition. Cependant, toutes s'accordent sur son effet positif. La logique subjective (LS) fournit une plateforme flexible pour modéliser la confiance. Nous l'utilisons pour proposer et comparer trois modèles de confiance, dont l'objectif est de prédire à un utilisateur source s'il peut faire confiance à un utilisateur cible. La recommandation peut s'appuyer sur l'expérience personnelle de la source (modèle local), un système de bouche à oreille (modèle collectif), ou encore la réputation du cible (modèle global). Nous comparons ces trois modèles aux termes de la précision, la complexité, et la robustesse face aux attaques malicieuses.

Mots-clés: système de recommandation, confiance, Logique subjective, filtrage collaboratif, réputation.

Abstract

recommender systems (RS) exploit users' behaviour to recommend to them items they would appreciate. Users Behavioral divergence on the web results in a problem of performance fluctuations to (RS). This problem is observed in the approach of collaborative filtering (CF), which exploits the ratings attributed by users to items, and in the trust-based approach (TRS), which exploits the trust relations between the users. We propose a hybrid approach that increases the number of users receiving recommendation, without significant loss of accuracy. Thereafter, we identify several behavioral characteristics that define a user profile. Then we classify users according to their common behavior, and observe the performance of the approaches by class.

Thereafter, we focus on the TRS. The concept of trust has been discussed in several disciplines. There is no real consensus on its definition. However, all agree on its positive effect. Subjective logic (LS) provides a flexible platform for modeling trust. We use it to propose and compare three trust models, which aims to predict whether a user source can trust a target user. Trust may be based on the personal experience of the source (local model), or on a system of mouth (collective model), or the reputation of the target (global model). We compare these three models in terms of accuracy, complexity, and robustness against malicious attacks.

Keywords: recommender systems, trust, subjective logic, collaborative filtering, reputation.

