Les oubliés de la recommandation sociale

Benjamin Gras

UNIVERSITÉ DE LORRAINE

École doctorale IAEM Lorraine

Les Oubliés de la Recommandation Sociale

THÈSE

présentée et soutenue publiquement le 18 janvier 2018

pour l'obtention du

Doctorat de l'Université de Lorraine

(mention informatique)

par

Benjamin Gras

Composition du jury

Rapporteurs :	Florence Sèdes	*Professeur, Université Paul Sabatier*
	Laurent Vercouter	*Professeur, INSA de Rouen*
Examinateurs :	Cécile Favre	*Maître de conférences, Université Lyon 2*
	Vincent Guigue	*Maître de conférences, Université Pierre et Marie Curie*
	Bernard Girau	*Professeur, Université de Lorraine*
	Armelle Brun	*Maître de conférences, Université de Lorraine*
Directrice :	Anne Boyer	*Professeur, Université de Lorraine*

Laboratoire Lorrain de Recherche en Informatique et ses Applications — UMR 7503

Mis en page avec la classe thesul.

Sommaire

Chapitre 1

Introduction

1.1 Contexte

Il y a plus de vingt-cinq ans maintenant, Tim Berners-Lee inventait le protocole de transfert HTTP, à l'origine d'Internet [Berners-Lee and Cailliau, 1990]. Cette avancée technologique a profondément changé notre société en permettant à chacun de rendre accessibles et d'accéder publiquement à des informations sur des pages Web. La sphère du Web n'a jamais cessé de s'étendre, de telle façon qu'il est désormais difficile d'en évaluer la taille. Bien que controversée [1], une méthodologie a récemment été proposée et permet de consulter en temps réel l'estimation de son envergure sur une page en ligne [2] [Bosch et al., 2016]. Selon cette dernière, environ 4,48 * 10^9 pages Web seraient indexées en 2017. C'est pourquoi, lorsqu'un utilisateur souhaite une information en particulier, il est nécessaire de l'assister et de le guider pour qu'il puisse l'identifier et y accéder dans cet océan d'informations.

Indexer pour mieux trouver.
Les moteurs de recherche [Filo and Yang, 1994, Brin and Page, 1998] ont été dans un premier temps la clé pour aider les utilisateurs à trier et/ou filtrer la surabondance d'informations sur Internet. A l'origine, les moteurs de recherche utilisaient une simple indexation par mots-clés, et mettaient en évidence les résultats les mieux référencés. Un utilisateur n'avait donc qu'à renseigner un ou plusieurs mot(s)-clé(s) correspondant à ses besoins et le moteur de recherche le guidait vers les sites Web correspondant à ces mots-clés. Plusieurs limitations sont rapidement venues gêner les utilisateurs de ces systèmes. Par exemple, les moteurs de recherche étant plus à même de référencer les sites populaires, l'accès aux informations contenues sur les pages peu consultées a alors été freiné. De plus, la difficulté des utilisateurs à exprimer les bons mots-clés pour une recherche limitait la qualité des résultats des moteurs de recherche. Enfin, des utilisateurs Grand Public peuvent avoir besoin d'accéder à des informations différentes même si leurs besoins ou leurs mots-clés semblent identiques. Prenons l'exemple de deux personnes souhaitant suivre un cours de bricolage organisé par une grande enseigne du domaine. Le premier a souvent entendu parler de bricolage, il sait ce qu'il est possible de réaliser soi-même, avec quel(s) outil(s), mais ne l'a jamais fait. Le second est quant à lui un néophyte, ignorant jusqu'à la manière de se servir correctement d'un marteau. Ils seront donc tous les deux à la recherche d'un cours d'initiation au bricolage, sans que le moteur de recherche ne puisse tenir compte de leur niveau de connaissance dans le domaine. Si plusieurs formations d'initiation au bricolage sont disponibles et qu'elles

1. https ://www.livescience.com/54094-how-big-is-the-internet.html
2. www.worldwidewebsize.com

concernent des publics différents, il est important de pouvoir guider chacun de ces deux profils vers la formation la mieux adaptée. Il s'agit du rôle des outils de personnalisation de l'accès à l'information.

Personnaliser pour trouver mieux.
Parmi les services de personnalisation de l'accès à l'information figurent les systèmes de recommandation (SR) [Goldberg, 1992]. Les SR sont étudiés depuis plus de vingt ans et sont désormais performants en moyenne. Ils sont très utilisés par de grands sites de e-commerce tels que Amazon [3] et beaucoup d'autres sites spécialisés comme Allociné [4]. Un SR a pour objectif de recommander à l'utilisateur des ressources pertinentes pour lui. Une ressource peut être un livre, un film, une page Web, une ressource éducative, etc. Pour permettre cette recommandation, le système utilise les informations qu'il a collectées sur l'utilisateur actif (c'est-à-dire l'utilisateur auquel le système doit fournir des recommandations) ou sur les ressources. La collecte des informations sur l'utilisateur peut être explicite, au travers d'un formulaire à remplir (préférence sur une ressource, centres d'intérêts, ...), ou implicite, grâce à l'analyse des traces de l'utilisateur (temps passé sur une ressource, nombre de ressources consultées, fréquence de consultation, analyse des commentaires...). Lorsque la collecte est implicite, une étape supplémentaire de déduction des préférences explicites de l'utilisateur est nécessaire. La manière dont les préférences explicites des utilisateurs sont collectées varie d'un système à l'autre en fonction de sa conception. Par exemple, pour collecter la préférence d'un utilisateur sur une ressource de manière explicite, le plus souvent, le système demande à l'utilisateur d'évaluer la ressource au travers d'une note. Cette note peut être binaire (pouce vert/rouge), discrète (note $\in [1;5]$, note $\in \mathbb{N}$) ou encore continue (un curseur sur une réglette, ...), etc. La définition formelle et l'impact des différentes manières de collecter les préférences (implicites et explicites) des utilisateurs ont été largement étudiés dans la littérature [Gena et al., 2011, Jawaheer et al., 2014].

Comment permettre cette personnalisation ?
Plusieurs types de SR peuvent exploiter les données collectées pour effectuer des recommandations. Les deux approches de recommandation les plus répandues sont le filtrage par contenu [Belkin and Croft, 1992, Oard and Marchionini, 1998] et le filtrage social ou collaboratif (FC) [Resnick et al., 1994, Su and Khoshgoftaar, 2009]. Dans le filtrage par contenu, le système utilise les informations qu'il possède sur les ressources (indexation, mots-clés, type, ...) pour sélectionner celles qui correspondent aux préférences collectées **uniquement** sur l'utilisateur actif, sans se préoccuper des préférences des autres utilisateurs. Dans le FC, le système utilise les données de préférences qu'il possède sur **l'ensemble** des utilisateurs. Le FC repose sur l'hypothèse que les préférences des utilisateurs sont cohérentes entre elles, ce qui permet d'inférer les préférences d'un utilisateur à partir des préférences des autres utilisateurs, sans nécessiter la connaissance des méta-données ou de données de contenu d'une ressource. Le FC ne dépend donc pas de la nature des ressources (livres, films, ...), ce qui allège sa mise en œuvre. De plus, les bonnes performances des systèmes à base de FC en font l'approche la plus populaire à l'heure actuelle. C'est pour cette raison que nous avons focalisé ce travail sur cette approche en particulier.

L'importance des données.
La qualité des recommandations apportées par une approche de FC est donc entièrement liée à la qualité et à la quantité des données collectées sur les utilisateurs. Il est nécessaire que les

3. www.amazon.com
4. www.allocine.fr

données collectées soient fiables, pour que le système ne se base pas sur des informations erronées pour calculer ses recommandations, et nombreuses sur chaque utilisateur, pour que le système puisse identifier les utilisateurs aux préférences similaires et les spécificités de chaque utilisateur. Face à cette étape sensible de collecte d'informations viennent se dresser plusieurs obstacles pouvant nuire à la qualité des recommandations : (1) il n'est pas possible de collecter toutes les préférences d'un utilisateur, car elles sont presque infinies et l'utilisateur ne peut pas connaître l'intégralité des ressources, (2) les lois visant à protéger la vie privée des utilisateurs sur Internet n'autorisent pas à sauvegarder des informations permettant de retrouver l'identité d'un utilisateur [Castagnos, 2008], et, (3) les données collectées, même de manière explicite, ne sont pas toujours fiables [Cosley et al., 2003, Brun et al., 2011], à cause de la présence de biais ou de bruit dans celles-ci [Jones et al., 2011, Yera Toledo et al., 2013]. Ce phénomène peut être en partie expliqué par l'inconsistance des utilisateurs au cours du temps. Par exemple, [Bellogín et al., 2014] montre que lorsque l'on demande à un utilisateur deux fois sa préférence sur une ressource à quelques mois d'intervalle, les deux préférences sont souvent différentes. Ce phénomène peut être expliqué par la modification du contexte [5] dans lequel se trouvait l'utilisateur au moment de la collecte de sa préférence [Adomavicius and Tuzhilin, 2005], ou encore par l'incertitude de l'utilisateur lorsqu'il évalue une ressource [Jones et al., 2011]. À ces obstacles s'ajoute le cas dans lequel le service ne possède que peu d'information sur l'utilisateur, il est alors difficile d'identifier les caractéristiques distinguant ou rapprochant cet utilisateur des autres utilisateurs. Il s'agit du problème du démarrage à froid [Schein et al., 2001]. De nombreux chercheurs travaillent sur le traitement de ce problème pour tenter d'en minimiser l'impact [Bobadilla et al., 2012a, Guo et al., 2014, Ben Ticha, 2015, Aleksandrova et al., 2016]. Les données collectées sur un utilisateur ne permettent donc pas toujours d'offrir un service de personnalisation adéquat aux utilisateurs.

Les conséquences de la moyenne.
Même dans le cas où les données collectées sont nombreuses et de bonne qualité, certaines limites peuvent résulter des caractéristiques propres aux techniques de FC. Historiquement, la technique de filtrage collaboratif la plus utilisée est la technique des k plus proches voisins [Resnick et al., 1994]. Cette technique repose sur l'évaluation de la similarité entre les utilisateurs et considère que la préférence de l'utilisateur actif sur une ressource r_1, qu'il n'a pas évaluée, est la *moyenne* des préférence sur la ressource r_1 des k utilisateurs lui étant les plus similaires. L'efficacité de cette technique repose sur trois conditions : (1) le système doit pouvoir identifier des utilisateurs similaires à l'utilisateur actif, sans lesquels il n'est pas possible d'inférer la préférence de ce dernier, (2) la *moyenne* des préférences des k utilisateurs les plus similaires doit avoir un sens (la moitié des voisins a adoré et l'autre moitié a détesté, que prédire ?) et (3) les utilisateurs similaires doivent avoir évalué des ressources que l'utilisateur actif n'a pas encore évalué. Plus tard, une autre technique de FC a émergé : la factorisation de matrice (FM) [Breese et al., 1998, Koren et al., 2009]. Cette technique factorise les préférences des utilisateurs de manière à en extraire les principales caractéristiques, en minimisant l'erreur *moyenne* obtenue lorsque l'on recompose les préférences connues des utilisateurs à partir des caractéristiques calculées pour les utilisateurs et les ressources. Intuitivement, le processus d'apprentissage de la FM va chercher les caractéristiques optimales, en *moyenne*, pour l'ensemble des utilisateurs et des ressources. De plus, quelle que soit la technique de FC utilisée, les performances des systèmes de recommandation sont le plus souvent calculées et affichées sous la

5. Le contexte est ici à considérer dans son sens le plus vaste, il peut être physique (lieu, température corporelle, fatigue, ...), émotionnel (joie, tristesse, amour, ...) ou encore par exemple temporel (matin, midi, soir, ...).

forme d'une moyenne. Une métrique d'évaluation est utilisée pour calculer l'erreur commise sur chacune des prédictions d'un algorithme de recommandation, mais ce n'est que la moyenne globale sur toutes les prédictions calculées par l'algorithme qui sera affichée. A performances globales similaires, il n'est pas possible de distinguer un algorithme dont les performances sont en permanence moyennes d'un algorithme dont les performances varient fortement. La notion de *moyenne* est donc centrale à ces deux techniques de FC qui sont les plus populaires. Ces techniques exploitent les informations qu'elles ont collectées sur les utilisateurs et ne tiennent, en général, pas suffisamment compte des caractéristiques qui distinguent les utilisateurs entre eux. Ces techniques vont alors plus souvent utiliser toutes les caractéristiques qu'elles possèdent sur un utilisateur, en les moyennant. Or, lorsqu'un utilisateur possède des caractéristiques qui distinguent son profil de celui des autres, si ces caractéristiques ne sont pas correctement exploitées, alors les recommandations proposées par le système ne correspondront pas forcément aux attentes de l'utilisateur concerné. Il s'agit du problème des **Grey Sheep Users** (GSU) [6] dans les SR [Claypool et al., 1999, Ghazanfar and Prugel-Bennett, 2011].

Dans un service à base de préférences tel que le FC, il est nécessaire qu'au moins une communauté d'utilisateurs partage les préférences de l'utilisateur actif si l'on souhaite pouvoir lui proposer des recommandations de bonne qualité. Je définis une **préférence spécifique** comme une préférence qui ne serait partagée pour aucun groupe d'utilisateurs. Chaque être humain possède au moins une préférence permettant de le distinguer des autres [Tooby and Cosmides, 1990], un utilisateur possédant une seule préférence spécifique ne doit donc pas être considéré comme un GSU. Cependant, un utilisateur possédant *un ensemble de plusieurs préférences spécifiques*, qu'il ne partage avec aucun autre utilisateur du service, sera probablement mal servi par une approche de FC classique. Je propose ici une définition plus générale que la définition originale d'un GSU [Claypool et al., 1999] : un GSU est un utilisateur dont les préférences spécifiques, le distinguant des autres utilisateurs, empêchent une utilisation cohérente de la moyenne pour le calcul de ses recommandations. La figure 1.1 illustre intuitivement la difficulté pour un système de recommandation sociale à prédire le sens de nage préféré du poisson rouge s'il se base sur les préférences des autres poissons.

Cette thèse porte sur l'analyse de l'identification et de la modélisation des GSU dans les systèmes de recommandation à base de filtrage collaboratif.

6. Ce terme peut être traduit littéralement par "utilisateur moutons gris" en français, nous conserverons l'appellation anglaise plus largement utilisée.

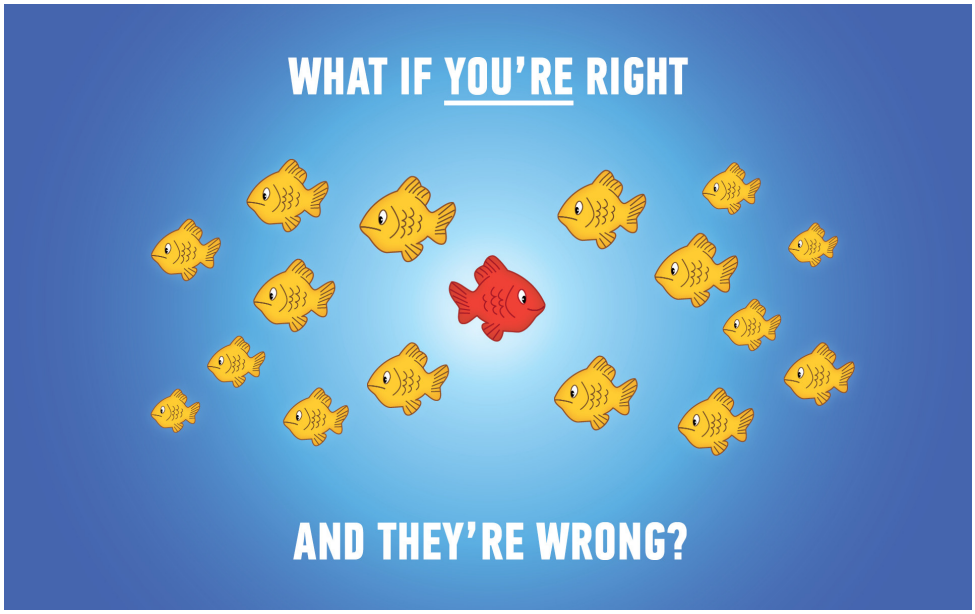


FIGURE 1.1 – Illustration de l'atypisme

1.2 Problématique

Pour aborder le problème des GSU dans les systèmes de recommandation à base de FC, il convient de se demander ce qu'est exactement une préférence. Dans son essai sur le goût de 1757, écrit pour l'Encyclopédie publiée sous la direction de Diderot, Montesquieu définit le goût comme la faculté de découvrir avec rapidité et délicatesse le degré de plaisir que nous devrions recevoir de chaque objet qui entre dans la sphère de notre perception. La définition de Montesquieu possède alors un double sens. Le goût se réfère à la fois aux sentiments de plaisir que l'on éprouve lorsqu'on est confronté à un objet, mais aussi aux normes intrinsèques de la beauté matérialisée par ces objets. Mais quel est le lien entre le goût et la préférence ?

En sociologie, le goût est défini par les motifs récurrents (personnels et culturels) à l'origine des préférences d'un individu [Bourdieu, 1984a]. S'intéresser aux préférences d'un individu est donc une manière de capter les goûts de ce dernier. Néanmoins, des mécanismes sociaux peuvent perturber le lien qui existe entre les notions de préférence et de goût. En effet, les préférences sont l'expression consciente des goûts de l'individu, il est donc possible que l'individu mente (consciemment ou inconsciemment) sur ses préférences et cache ses véritables goûts. De plus, selon Bourdieu, les phénomènes sociaux et culturels qui sont à l'origine des préférences des individus sont étroitement associés aux relations sociales et à la dynamique entre les gens. C'est en partant de ce constat que le sociologue Don Slater a montré que les personnes désirent se distinguer de celles qu'elles pensent avoir un statut inférieur ou égal à elles-même dans la hiérarchie sociale et, pour cela, elles imitent celles qui occupent des postes supérieurs [Slater, 1997]. Pour illustrer ce propos, la théorie originale de l'espace social de Pierre Bourdieu montre le lien entre les préférences des individus et leurs places dans la société [Bourdieu, 1984b]. Par exemple, un individu qui apprécie le champagne et l'équitation sera jugé plus apte à occuper des postes importants dans la société, tandis qu'un individu appréciant la bière et le football sera quant à lui déprécié. Selon la sociologie, de nombreux individus préfèreraient ainsi exprimer une préférence

pour le champagne même si leur boisson préférée est la bière.

Ce phénomène social s'applique également au domaine de la recommandation. Les individus sont alors des utilisateurs et leurs préférences concernent des ressources. Un utilisateur qui se distingue volontairement de certains autres utilisateurs au niveau de ses préférences va alors porter une attention toute particulière au respect de ses **préférences spécifiques** par un SR. Or, les systèmes actuels de FC, tels que la technique des k plus proches voisins, n'utilisent que l'information contenue dans la similarité entre deux utilisateurs pour estimer la préférence d'un utilisateur sur une ressource. Les algorithmes de FC n'accordent pas une attention particulière aux préférences qui distinguent les utilisateurs les uns des autres, bien que ces algorithmes proposent des recommandations considérées satisfaisantes en moyenne [Castagnos et al., 2013].

Les utilisateurs possédant un grand nombre de **préférences spécifiques**, ou **GSU**, ne permettent pas au système de FC de trouver une cohérence entre leurs préférences et celles des autres utilisateurs et le système ne sera donc pas en mesure de leur fournir des recommandations de bonne qualité.

Dans ce travail, nous nous demandons :

1. **Que signifie avoir des préférences spécifiques dans les systèmes de recommandation à base de filtrage collaboratif ?**

   Les préférences d'un individu peuvent présenter deux types de spécificités : les spécificités intrinsèques et les spécificités extrinsèques. Les spécificités intrinsèques concernent les préférences qui ne sont pas cohérentes avec le reste des préférences d'un même utilisateur [Bellogín et al., 2011], tandis que les spécificités extrinsèques concernent les préférences qui ne sont pas cohérentes avec les préférences des autres utilisateurs du système [Del Prete and Capra, 2010]. Dans cette thèse, nous nous concentrons sur les préférences spécifiques extrinsèques aux individus. Rappelons que les sociologues sont d'accord pour affirmer que les individus apprécient se distinguer les uns par rapport aux autres au niveau de leurs préférences [Bourdieu, 1984a, Slater, 1997]. Ce comportement entraîne une très grande variabilité des préférences extrinsèques des individus. Il s'agit donc de mesurer le degré de spécificité des préférences que possède chaque utilisateur par rapport à celles du reste des utilisateurs. En d'autres termes, nous sommes ici à la recherche d'un indicateur permettant d'évaluer à quel point un utilisateur est différent de tous les autres.

   Nous savons que les utilisateurs possédant beaucoup de préférences spécifiques sont mal servis par un système de FC [Claypool et al., 1999]. Il est donc important d'identifier ces utilisateurs, les GSU, afin d'éviter qu'ils soient mal servis par un SR.

2. **Comment identifier les GSU dans le cadre des systèmes de recommandation sociale ?**

   Les approches de recommandation sociale nécessitent toutes une phase de collecte des données (implicites ou explicites) de préférences d'un utilisateur pour permettre la recommandation de ressources. L'identification des GSU dans les systèmes de recommandation reste un sujet encore très peu étudié. Le domaine de la fouille de données, et plus particulièrement, la détection des *Outliers* (ou "données aberrantes") répondent à un problème similaire. "Un *outlier* est une observation qui dévie tellement des autres observations que l'on peut suspecter qu'elle a été générée par un mécanisme différent." [Hawkins, 1980]. La

détection des *outliers* est le plus souvent utilisée pour détecter des fraudes, des tentatives d'intrusion, des données aberrantes (lors de l'application de méthodes statistiques par exemple), etc.

En recommandation sociale, une observation est un utilisateur possédant un ensemble de préférences connues. Identifier un utilisateur dont les préférences seraient aberrantes ou très spécifiques est une tâche difficile de par la nature des données. En effet, les données sont volumineuses, en grande dimension et parcimonieuses. Dans un algorithme de recommandation sociale, chaque utilisateur est représenté sous la forme d'un vecteur possédant autant de dimensions que de ressources référencées par le système. Toutefois, la majeure partie des utilisateurs n'aura exprimé ses préférences que sur une infime partie des ressources du système. On parle alors de manque de données. De plus, les données sont exprimées par des utilisateurs, ce qui peut introduire un biais utilisateur, rendant les données collectées peu fiables [Jones et al., 2011]. La nature des données peut donc impacter les performances de certaines méthodes du domaine du *data mining*.

Tenant compte de ces contraintes, pour mesurer la différence entre un utilisateur et tous les autres à partir de ses préférences, deux types de méthodes du domaine de l'*outlier detection* sont en concurrence : les méthodes statistiques et les méthodes à base de distance [Ramaswamy et al., 2000, Ghazanfar and Prügel-Bennett, 2014, Gras et al., 2015b]. Chacune de ces méthodes présente des avantages et des inconvénients qu'il convient d'analyser pour ensuite pouvoir correctement exploiter les résultats obtenus.

3. **Comment modéliser les GSU afin de leur apporter de meilleures recommandations ?**

   Nous avons vu que les GSU possèdent plus de préférences spécifiques que les autres utilisateurs et sont donc mal servis par les algorithmes de FC [Del Prete and Capra, 2010, Gras et al., 2015b], qui exploitent la similarité. En effet, ces préférences spécifiques complexifient la recherche d'utilisateurs similaires à un GSU et conduisent à de mauvaises recommandations. Les problèmes plus classiques de la recommandation sociale, tels que le démarrage à froid [Schein et al., 2001] ou encore le manque de données [Grcar et al., 2005b, Grcar et al., 2005a], viennent s'additionner aux préférences spécifiques des utilisateurs. Il est donc nécessaire de proposer de nouvelles approches de recommandation pour permettre aux GSU de bénéficier de la préférences des autres utilisateurs de la même manière qu'un utilisateur possédant moins de préférences spécifiques, et ainsi recevoir des recommandations de meilleur qualité.

En résumé, dans cette thèse nous nous intéressons à trois questions distinctes. La première question est : qu'est-ce qu'une préférence spécifique ? Nous y apportons une réponse en proposant des hypothèses associées que nous validons expérimentalement. Ensuite, nous nous demandons comment identifier les GSU dans les données ? Cette identification est importante afin d'anticiper les mauvaises recommandations qui seront fournies à ces utilisateurs. Enfin, comment modéliser ces GSU pour améliorer la qualité des recommandations qui leurs sont fournies ? La modélisation des GSU en particulier nous permet de leur offrir un service de meilleure qualité. **L'objectif de ce travail est donc de proposer une amélioration de l'identification et de la modélisation des GSU dans les systèmes de recommandation.**

1.3 Contributions

Les contributions de cette thèse s'étendent de la proposition d'une définition théorique claire du concept de GSU à la conception de modèles dédiés à l'identification et à la modélisation des GSU dans un système de recommandation en passant par des mesures adaptées.

1.3.1 La définition du concept de GSU

Nous avons évoqué le lien entre le concept de GSU et celui d'*outlier*. Les préférences d'un utilisateur peuvent s'écarter de celle des autres, et ainsi devenir spécifiques, de bien des manières. Nous avons par exemple brièvement défini les natures extrinsèque et intrinsèque des spécificités que peuvent avoir les préférences des utilisateurs. Dans ce travail, nous partons d'une définition simple du concept de GSU que nous affinons grâce notamment aux définitions issues des sciences humaines. Par exemple, la présence de préférences spécifiques et non spécifiques implique la définition d'une norme permettant de les distinguer. Nous apportons un nouveau regard au phénomène des GSU dans les SR sociale à la fois dans la manière dont nous les définissons et dans la manière dont nous les considérons. Bon nombre des travaux portant sur les *outliers*, ou sur les GSU, considèrent que les observations (ou utilisateurs) les plus déviantes d'un système ne doivent pas être prise en compte et qu'il faut les oublier. Cette approche du problème peut être valable lorsqu'il s'agit de données aberrantes issues d'attaques malveillantes ou du dysfonctionnement d'un capteur par exemple, mais pas lorsque les données à écarter sont des utilisateurs. Dans ce manuscrit, nous défendons fermement l'idée que les GSU peuvent et doivent être correctement traités par un SR social.

1.3.2 L'identification des GSU

Notre contribution pour l'identification des GSU [Gras et al., 2015a] [7] est triple : tout d'abord une modification d'une mesure d'identification de l'état de l'art [Del Prete and Capra, 2010, Haydar et al., 2012] permettant la prise en compte des caractéristiques des ressources dans le calcul de l'*Anormalité* d'un utilisateur. Le résultat, la mesure d'*AnormalitéCR*, nous a permis de réduire fortement le taux de fausses identifications de la mesure originale. De plus, la définition d'un second indicateur, l'*AnormalitéCRU*, a permis d'étudier l'effet de la prise en compte des particularités de chaque utilisateur sur la précision de l'identification des GSU. Les expérimentations menées montrent la pertinence du coefficient d'*AnormalitéCRU* pour l'identification d'utilisateurs appartenant au groupe des GSU.

Dans [Gras et al., 2015b] [8], nous montrons la généricité de notre coefficient d'*AnormalitéCRU* en vérifiant les mêmes résultats avec plusieurs techniques de recommandation. Une version étendue de cet article a fait l'objet d'un chapitre de livre [Gras et al., 2015c] [9] dans lequel nous dévoilons d'avantages d'expérimentations permettant de comparer nos mesures avec plusieurs mesures de l'état de l'art. Nous en profitons d'ailleurs pour exposer les hypothèses scientifiques

7. Gras, b., Brun, A., and Boyer, A. (2015a). Identification des utilisateurs atypiques dans les systèmes de recommandation sociale. In *EGC - Extraction et Gestion de Connaissances*, Esch sur Alzette, Luxembourg

8. Gras, B., Brun, A., and Boyer, A. (2015b). Identifying users with atypical preferences to anticipate inaccurate recommendations. In *Proceedings of the 11th International Conference on Web Information Systems and Technologies*

9. Gras, B., Brun, A., and Boyer, A. (2015c). When users with preferences different from others get inaccurate recommendations. In *International Conference on Web Information Systems and Technologies*, pages 191–210. Springer

qui n'ont pas pu être validées, ainsi que les expérimentations associées.

Dans un second temps, nous définissons une deuxième mesure d'identification des GSU pour deux raisons : d'une part, bien qu'elle soit plus performante que les mesures de l'état de l'art, le nombre de faux positifs du coefficient d'*AnormalitéCRU* nous semblait encore trop élevé, et d'autre part, pour capturer les utilisateurs qui possèdent une autre forme de spécificité au niveau de leurs préférences. Nous avons pour cela proposé une méthode probabiliste basée sur le principe de la *Vraisemblance*, appelée VraisemblanceID [Gras et al., 2016] [10]. Cette méthode se distingue des méthodes classiques de la littérature car elle n'utilise pas uniquement la moyenne des préférences d'une ressource, mais elle modélise la distribution de ces préférences sur cette ressource et se base sur ce modèle pour définir la *Vraisemblance* d'une préférence. Notamment, en comparant les performances de la mesure VraisemblanceID avec celles des autres mesures de la littérature, nous montrons que cette mesure permet de sélectionner plus d'utilisateurs possédant des préférences spécifiques que les mesures que nous utilisions précédemment, en veillant à ne pas dépasser un seuil maximum de faux positifs. Cette publication à d'ailleurs fait l'objet d'une mention "Outstanding Paper" dans la plus importante conférence dédiée à la modélisation des utilisateurs et de leurs usages sur internet (UMAP).

1.3.3 La modélisation des GSU

Concernant les recommandations apportées aux GSU identifiés [Gras et al., 2017] [11], nous proposons plusieurs approches de recommandation dédiées. Nous étudions tout d'abord l'impact de l'utilisation de la dissimilarité dans une approche des k plus proches voisins sur la qualité des recommandations fournies aux GSU. Ensuite, nous définissons une nouvelle méthode de calcul de la similarité entre deux utilisateurs avec pour objectif de favoriser l'émergence de liens entre les utilisateurs qui partagent le même voisin dans une approche des k plus proches voisins. Enfin, nous proposons plusieurs modèles basés sur un algorithme de factorisation de matrice [Koren et al., 2009] dont l'objectif est d'apporter de meilleures recommandations aux GSU. Un premier modèle, *GSUOnly*, ne modélise que les GSU qu'il sépare du reste des utilisateurs. Un second modèle, *WeightedGSU*, donne la priorité aux GSU lors de la modélisation de l'ensemble des utilisateurs, sans séparer les utilisateurs. Un dernier modèle, *SingleGSU*, ne permet de modéliser qu'un seul GSU par modèle. Nous visons l'étude au cas par cas de la capacité à modéliser un GSU. Ces travaux ont montré qu'il est effectivement possible d'améliorer les recommandations fournies à un GSU uniquement à partir de leurs préférences [Gras et al., 2017].

1.4 Plan de la thèse

Ce manuscrit est organisé de la manière suivante : dans le chapitre 2, nous présentons les concepts et les approches de la littérature liés à la définition, l'identification et la modélisation des GSU. Dans le chapitre 3, nous introduisons les mesures d'identification des GSU que nous avons élaborées. Ensuite, nous exposons les modèles de recommandation spécialement conçus pour

10. Gras, B., Brun, A., and Boyer, A. (2016). Identifying grey sheep users in collaborative filtering: a distribution-based technique. In *Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization*, pages 17–26. ACM

11. Gras, B., Brun, A., and Boyer, A. (2017). Can matrix factorization improve the accuracy of recommendations provided to grey sheep users? In *Proceedings of the 13th International Conference on Web Information Systems and Technologies - Volume 1: WEBIST,*, pages 88–96. INSTICC, ScitePress

apporter de meilleures recommandations aux GSU dans le chapitre 4. Enfin, nous concluons et présentons les perspectives dans le chapitre 5.

Chapitre 2

État de l'art

Sommaire

Dans ce chapitre, nous détaillons et analysons le fonctionnement des systèmes de recommandation sociale qui sont le contexte applicatif de ces travaux, puis nous étudions les travaux réalisés sur le sujet de l'atypisme. Nous mettons ensuite en perspective le fonctionnement de la recommandation sociale par rapport au problème des GSU de manière à mettre en évidence les éléments à l'origine des recommandations de mauvaise qualité fournies aux utilisateurs possédant des préférences spécifiques [Su and Khoshgoftaar, 2009], appelés GSU dans les systèmes de recommandation. Bien que le problème des GSU [Claypool et al., 1999] soit toutefois très peu étudié dans la littérature, nous verrons que si nous explicitons le lien entre ce problème et celui des *outliers* du domaine de la fouille de données, alors il existe de nombreux outils et méthodes sur lesquels nous appuyer pour proposer des solutions.

2.1 Recommandation sociale et atypisme

Dans cette première partie de notre état de l'art, nous présentons la recommandation sociale et définissons le concept d'atypisme, puis nous mettons en évidence la place et les enjeux de l'atypisme dans la recommandation sociale

Lorsqu'un utilisateur accède, par exemple, à un site Web et qu'il souhaite y trouver une ressource correspondant à ses besoins, si le nombre de ressources est très important, il est nécessaire pour lui de filtrer les ressources. Prenons l'exemple d'un utilisateur à la recherche d'un livre de poésie française sur Amazon [12], ce sont pas moins de 55 000 livres qui correspondent à cette description. Dans cet exemple, si l'utilisateur n'a pas d'avantage de critères quant au livre qu'il recherche, la sélection d'un livre qui correspond à ses préférences risque d'être très complexe. C'est pourquoi de nombreuses applications commerciales ont émergé de ces travaux comme le SR d'Amazon [Linden et al., 2003] ou encore Netflix [13] [Koren et al., 2009], ainsi que d'autres applications non commerciales comme l'application Tapestry [Goldberg, 1992], GroupLens [Resnick et al., 1994], ou encore Ringo [Shardanand and Maes, 1995].

Les systèmes de recommandation sont donc un outil de personnalisation de l'accès à l'information [Goldberg, 1992, Adomavicius and Tuzhilin, 2005]. Les deux approches de recommandation les plus utilisés sont le filtrage par contenu [Belkin and Croft, 1992] et le filtrage collaboratif [Resnick et al., 1994]. Le filtrage par contenu est une approche de recommandation qui repose sur les préférences de l'utilisateur actif u [14] pour lui recommander des ressources similaires à celles qu'il a appréciées dans le passé. Cette approche de filtrage nécessite donc de disposer d'informations de "contenu" sur les ressources référencées par le système. Cela rend difficile la mise en œuvre du filtrage par contenu dans les systèmes pour lesquels ce recueil d'informations ne peut être automatisé. Le filtrage social ou collaboratif (FC) repose sur l'hypothèse que les préférences des utilisateurs sont cohérentes entre elles. Cela signifie qu'il est possible de se reposer sur les préférences d'un utilisateur similaire à l'utilisateur actif pour inférer les préférences de l'utilisateur actif sur les ressources qu'il n'a pas encore évaluées. Il n'est donc pas nécessaire de renseigner la moindre information sur les ressources référencées par le système. Dans ce travail, nous nous sommes focalisés sur la recommandation sociale, c'est-à-dire l'approche à base de FC (Filtrage Collaboratif) car il s'agit de la technique la plus utilisée à ce jour, notamment grâce à ses performances [Bobadilla et al., 2013].

12. www.amazon.fr
13. www.netflix.com
14. Nous appellerons l'utilisateur actif "u" tout au long de ce chapitre.

2.1.1 Les systèmes de recommandation sociale

Nous présentons d'abord le principe générique de la recommandation sociale, nous décrivons ensuite différentes techniques de FC, puis nous explicitons les méthodes d'évaluation de ces systèmes, et enfin nous présentons les limites que nous entrevoyons à cette approche.

2.1.1.1 Le principe de base du FC

Le filtrage collaboratif (FC) est une forme de recommandation sociale qui repose sur l'exploitation des préférences des utilisateurs sur des ressources. Grâce aux préférences d'une partie des utilisateurs sur une ressource, le système estime les préférences d'autres utilisateurs sur cette ressource. De cette manière, un système à base de FC va pouvoir isoler les ressources pertinentes pour l'utilisateur actif. En général, les préférences des utilisateurs sur des ressources sont recueillies sous la forme de notes explicitement données par ces utilisateurs. Lorsque celles-ci ne sont pas disponibles, elles peuvent être inférées en fonction des traces laissées par les utilisateurs [Agrawal and Srikant, 1995, Castagnos, 2008]. Ces notes sont souvent stockées dans une matrice représentant en ligne les utilisateurs du système et en colonne les ressources que le système recense. Cette matrice est appelée matrice de notes. Le tableau 2.1 présente un exemple de matrice de notes collectant des notes allant de 1 à 5 (valeurs entières) avec 4 utilisateurs (u_1 à u_4) et 6 ressources (r_1 à r_6) :

	r_1	r_2	r_3	r_4	r_5	r_6
u_1		5	2	2	1	
u_2	2	1		5	5	5
u_3	4		1	3		3
u_4	5	4	1		2	4

TABLE 2.1 – Exemple de matrice de notes *(Pour clarifier l'explication, nous présentons ici une matrice très renseignée. Dans les faits, on observe plus de 95% de valeurs non renseignées dans les matrices de notes)*

Dans le tableau 2.1, les cases non renseignées correspondent aux ressources pour lesquelles les utilisateurs n'ont pas exprimé leur préférence. Les cases renseignées correspondent aux notes qu'un utilisateur a données aux ressources. L'objectif d'un algorithme de FC est d'inférer les valeurs manquantes pour ensuite recommander à l'utilisateur actif les ressources dont les valeurs sont les plus élevées.

Il existe deux grands types de techniques de recommandation sociale, l'approche à base de mémoire [Resnick et al., 1994, Sarwar et al., 2001] et l'approche à base de modèle [Breese et al., 1998, Lemire and Maclachlan, 2005, Koren et al., 2009]. C'est la manière dont est utilisée la matrice de notes qui représente la principale distinction entre ces deux types de recommandation sociale. Une approche à base de mémoire calcule les recommandations directement à partir de la matrice de notes, tandis qu'une approche à base de modèle apprend un modèle à partir de la matrice de notes et utilise ce modèle pour calculer des recommandations. Nous présentons ci-dessous plus en détails le fonctionnement de chacune de ces deux approches.

2.1.1.2 Les techniques de recommandation de l'approche à base de mémoire

L'approche à base de mémoire repose sur le calcul de la similarité entre l'utilisateur actif et chacun des autres utilisateurs du système. La technique appelée k plus proches voisins

(KNN)[Fix and Jr, 1951] sélectionne les k utilisateurs les plus similaires à l'utilisateur actif. Intuitivement, lorsque l'on souhaite estimer la note que mettrait un utilisateur u sur une ressource r, la technique des k plus proches voisins isole les k utilisateurs les plus similaires à u et s'inspire ensuite de leurs préférences sur r (approche *User-User*) [15] [Resnick et al., 1994]. Nous poursuivons les explications concernant cette technique en utilisant l'exemple de l'approche *User-User*.

Pour effectuer une recommandation, l'approche *User-User* utilise les préférences des k plus proches voisins de l'utilisateur actif sur les ressources qu'il n'a pas notées et sélectionne les ressources dont les préférences ainsi estimées sont les plus élevées.

Dans l'exemple précédent (tableau 2.1), si le système souhaite effectuer une recommandation à u_3, il calcule la similarité entre u_3 et chacun des autres utilisateurs. Voici un exemple de résultat obtenu :

Similarité	u_1	u_2	u_3	u_4
u_3	**0,4**	0,15	*(1,0)*	**0,8**

TABLE 2.2 – Exemple de similarité inter-utilisateurs

L'étape suivante est la sélection des k plus proches voisins. Le nombre de voisins plus proches (k) sélectionnés est choisi en fonction du nombre d'utilisateurs du système. Ici, prenons $k = 2$. Il faut donc sélectionner les deux plus proches voisins de u_3 qui sont u_1 et u_4.

Le système effectue enfin une moyenne pondérée (avec la similarité) des notes des $k = 2$ plus proches voisins et obtient ce type de prédiction :

	r_1	r_2	r_3	r_4	r_5	r_6
u_1		5	2	2	1	
u_2	2	1		5	5	5
u_3	4	**4,6**	1	3	**1,4**	3
u_4	5	4	1		2	4

TABLE 2.3 – Exemple de prédictions pour l'utilisateur u_3

Étant donnés ces résultats, le système va sélectionner la ressource r_2 pour la recommander à l'utilisateur u_3 car il estime que l'utilisateur u_3 donnera la note la plus élevée à cette ressource.

Cette approche est simple à mettre en œuvre et les recommandations fournies évoluent dynamiquement puisque lorsqu'un utilisateur renseigne une note supplémentaire, elle est directement prise en compte par le processus de recommandation. Cependant, cette technique souffre d'un problème de passage à l'échelle dû à l'explosion combinatoire lorsque l'on ajoute de nouveaux utilisateurs ou de nouvelles ressources, notamment pour le calcul des similarités. Il s'agit là du principal défaut de cette approche. Nous allons voir en détails comment cette similarité peut être calculée.

Les mesures de similarité

Lorsque l'on utilise l'approche à base de mémoire, l'étape critique est la détermination des k plus proches voisins de l'utilisateur actif qui vont permettre au système de calculer ses prédictions. La similarité entre deux utilisateurs u et v est notée sim(u,v). Certains algorithmes

15. Cette approche peut également isoler les k ressources les plus similaires à r et en s'inspirant des préférences de u sur ces derniers (approche *Item-Item*)

utilisent également les similarités entre ressources (approche Item-Item) de manière à pouvoir ajouter de l'information sur les utilisateurs [Sarwar et al., 2001]. Il existe de nombreuses méthodes pour calculer cette similarité, nous présentons les plus populaires dans cette partie.

Le coefficient de corrélation de Pearson

Le coefficient de *corrélation de Pearson* mesure la corrélation linéaire entre deux variables, c'est-à-dire la dépendance linéaire entre ces variables [Resnick et al., 1994, Shardanand and Maes, 1995]. La *corrélation de Pearson* entre les utilisateurs u et v est notée $Pearson(u, v)$ et est calculée de la façon suivante :

$$Pearson(u,v) = \frac{\sum_{r \in R_{uv}} (n_{u,r} - \overline{n}_u)(n_{v,r} - \overline{n}_v)}{\sqrt{\sum_{r \in R_{uv}} (n_{u,r} - \overline{n}_u)^2} \sqrt{\sum_{r \in R_{uv}} (n_{v,r} - \overline{n}_v)^2}}$$

où R_{uv} est l'ensemble des items co-votés par u et v, $n_{u,r}$ est la note de l'utilisateur u sur la ressource r et $\overline{n}_u$ est la note moyenne donnée par l'utilisateur u sur l'ensemble des ressources qu'il a évaluées. Par exemple, dans le cas du tableau 2.1, les corrélations de Pearson entre utilisateurs sont :

	u_1	u_2	u_3	u_4
u_1	1,0	-0,97	0,01	0,83
u_2	-0,97	1,0	-1,0	-0,61
u_3	0,01	-1,0	1,0	0,98
u_4	0,83	-0,61	0,98	1,0

TABLE 2.4 – Corrélations de Pearson pour le tableau 2.1

L'exploitation des corrélations négatives est très rare en recommandation sociale. En effet, il a été montré expérimentalement que l'utilisation des corrélations négatives diminue la précision des algorithmes classiques [Bobadilla et al., 2013]. A l'opposé, dans [Zeng et al., 2010], les auteurs montrent qu'il est possible d'améliorer la qualité des recommandations en utilisant les corrélations négatives, et proposent une nouvelle approche qui n'écarte pas les utilisateurs les plus anti-corrélés. Dans les faits, l'amélioration de la qualité des recommandations reste faible, en revanche l'utilisation des corrélations négatives permet de recommander des ressources qui ne sont pas recommandées par les systèmes classiques, ce qui apporte de la diversité aux recommandations proposées par l'algorithme.

Il existe des mesures de corrélation qui dérivent de la *corrélation de Pearson*, mais à nouveau les gains en précision sont faibles [McLaughlin and Herlocker, 2004, Su and Khoshgoftaar, 2009]. La *corrélation de Pearson* reste néanmoins la mesure de similarité la plus utilisée dans les algorithme de FC à base de voisinage.

La similarité Cosinus

La *similarité Cosinus* calcule l'angle formé par deux vecteurs possédant un même nombre de dimensions [Breese et al., 1998]. Dans [Salton, 1986], l'auteur calcule la similarité Cosinus entre des documents textes en considérant chaque texte comme un vecteur associant à chaque mot différent sa fréquence d'apparition.

La *similarité Cosinus* entre les utilisateurs u et v est calculée de la manière suivante :

$$cos(u,v) = \frac{\sum_{r \in R_{uv}} n_{u,r} * n_{v,r}}{\sqrt{\sum n_{u,r}^2} \sqrt{\sum n_{v,r}^2}}$$

La *similarité Cosinus* a pour principal atout ses performances pour le calcul de similarités sur les vecteurs à haute dimension en situation de manque de données [Grčar et al., 2006]. Cependant, cette mesure ne tient pas compte du biais de l'utilisateur, contrairement au coefficient de corrélation de Pearson, ce qui implique, par exemple, qu'un utilisateur très sévère (avec une moyenne de 2 par exemple) aura une similarité faible avec les autres utilisateurs, s'ils ne sont pas dans le même cas. Le mot "biais" tel que nous l'employons ici est dévié de sa signification statistique originale, mais il est utilisé ainsi dans le domaine du filtrage collaboratif. Le terme biais représente ici la part de chaque préférence qui n'est pas due à l'interaction brute entre une utilisateur et une ressource. Par exemple, le biais utilisateur est la part d'une préférence qui ne dépend pas de la ressource sur laquelle la préférence à été exprimée, mais uniquement de l'utilisateur. Un utilisateur qui a tendance à surévaluer les ressources aura par exemple un biais utilisateur positif.

Le coefficient de Jaccard

Le *coefficient de Jaccard* entre les utilisateurs u et v est obtenu de la manière suivante :

$$Jaccard(u,v) = \frac{|R_u \cap R_v|}{|R_u \cup R_v|},$$

où R_u représente l'ensemble des ressources votées par u et R_u l'ensemble des ressources votées par v. Ce coefficient permet donc de tenir compte du nombre de ressources communes à deux utilisateurs en fonction de leur nombre total d'évaluations.

Le *coefficient de Jaccard* est très souvent utilisé pour pondérer d'autres mesures de similarité. Dans [Candillier et al., 2008], les auteurs montrent qu'il est possible d'améliorer la qualité des recommandations d'un FC utilisant la *corrélation de Pearson* en pondérant cette mesure par le *coefficient de Jaccard*. Le coefficient de Jaccard est également utilisé en cas de démarrage à froid lorsque le coefficient de corrélation de Pearson ne peut pas être utilisé [Bobadilla et al., 2012b].

Autres mesures de similarité

De nombreux articles définissent de nouvelles mesures de similarité permettant d'améliorer la qualité de la recommandation [Bobadilla et al., 2012a, Liu et al., 2014]. Cependant, ces mesures sont très souvent dédiées à des problèmes spécifiques et n'améliorent que peu les performances des approches classiques. Elles sont donc peu utilisées et ne seront pas expliquées plus en détail dans ce manuscrit.

Le calcul des recommandations

Le calcul de la recommandation s'effectue ensuite en deux étapes. La première consiste à sélectionner les utilisateurs dont exploitera les préférences (les k plus proches voisins) et la seconde étape consiste à estimer les notes manquantes de l'utilisateur actif en faisant, par exemple, une moyenne pondérée (par la similarité) des préférences des plus proches voisins.

Sélection des plus proches voisins

On isole un ensemble composé des k voisins présentant la plus forte similarité avec l'utilisateur actif. Le nombre de voisins sélectionnés k peut être modifié en fonction des données sur lesquelles le système effectue ses calculs. En général, on fixe cette valeur entre 10 et 50 en fonction du jeu de données sur lesquelles on travaille [Resnick et al., 1994, Gras et al., 2015c].

Moyenne pondérée des notes des plus proches voisins
Pour calculer une prédiction pour l'utilisateur actif u sur une ressource r, on utilise une moyenne pondérée des notes attribuées à cette ressource par ses k plus proches voisins [Resnick et al., 1994] représentée par l'équation 2.1.

$$Prédiction_{u,r} = \overline{n}_u + \frac{\sum_{v \in V_u}(n_{v,r} - \overline{n}_v) * sim(u,v)}{\sum_{v \in V_u} |sim(u,v)|}, \tag{2.1}$$

où V_u représente l'ensemble des k plus proches voisins de u, $n_{v,r}$ est la note mise par l'utilisateur v sur la ressource r et *sim(u,v)* est la similarité précédemment calculée entre l'utilisateur actif u et son voisin v.

Si on reprend l'exemple du tableau 2.1, avec $k = 2$, la prédiction de la note de u_3 sur la ressource r_2 est la suivante :

$$\begin{aligned} Prédiction_{u_3,r_2} &= \overline{n}_{u_3} + \frac{\sum_{v \in V}(n_{v,r_2} - \overline{n}_v) * sim(u_3,v)}{\sum_{v \in V} |sim(u_3,v)|} \\ &= \overline{n}_{u_3} + \frac{(n_{u_4,r_2} - \overline{n}_{u_4}) * sim(u_3,u_4) + (n_{u_1,r_2} - \overline{n}_{u_1}) * sim(u_3,u_1)}{|sim(u_3,u_4)| + |sim(u_3,u_1)|} \\ &= 2,75 + \frac{(4-3,2)*0,98 + (5-2,5)*0,01}{0,98+0,01} \\ &= 3,57 \end{aligned}$$

D'autres méthodes ont été proposées pour la prédiction de notes [Bobadilla et al., 2012a, Su and Khoshgoftaar, 2009], mais elles ne s'appliquent souvent qu'à des cas spécifiques, comme le problème du démarrage à froid. La méthode de la moyenne pondérée des plus proches voisins étant de loin la plus populaire, nous ne détaillerons pas ces autres méthodes.

2.1.1.3 Les techniques de recommandation de l'approche à base de modèle

L'approche à base de modèle apprend un modèle qui décrit le lien existant entre les utilisateurs et les ressources : les notes. C'est ensuite ce modèle qui est utilisé pour calculer une recommandation pour l'utilisateur actif. Le principal avantage de ces techniques est qu'elles sont mieux adaptées au problème du passage à l'échelle. En effet, une fois le modèle appris, le calcul de la recommandation est relativement simple, contrairement au cas de l'approche à base de mémoire. Un grand nombre de techniques de FC basées sur un modèle ont été proposées dans la littérature. On peut citer les techniques basées sur le clustering [Ungar and Foster, 1998, Esslimani et al., 2009], les réseaux Bayesiens [Chickering et al., 1997, Breese et al., 1998], la factorisation de matrice [Billsus and Pazzani, 1998, Koren et al., 2009] ou encore l'analyse par la sémantique latente [Hofmann, 2004], etc.

Parmi les approches à base de modèle, la factorisation de matrice permet de fournir des recommandations de meilleur qualité aux utilisateurs [Koren et al., 2009, Yu et al., 2014], améliorant les résultats des approches à base de mémoire, tout en favorisant le passage à l'échelle. Nous détaillons son fonctionnement dans cette section car il s'agit de l'approche de recommandation de référence dans les systèmes de recommandation sociale et peut donc constituer une base à la conception d'une approche dédiée à la modélisation des GSU. Le clustering est une méthode qui est très souvent employée pour la classification des utilisateurs [Ghazanfar and Prugel-Bennett, 2011, Haydar et al., 2012]. Le clustering peut donc à la fois être

utilisé pour identifier des utilisateurs possédant des caractéristiques particulières, mais également pour permettre la modélisation de ces utilisateurs.

Les méthodes à base de clustering

Un algorithme de clustering a pour principal objectif de regrouper les données dans des clusters homogènes. Il existe plusieurs techniques de clustering dont les plus connues sont l'algorithme DBSCAN [Ester et al., 1996, Das et al., 2014], l'algorithme espérance-maximisation (EM) [Dempster et al., 1977, Ungar and Foster, 1998] ou l'algorithme *k-means* [MacQueen, 1967, Ghazanfar and Prügel-Bennett, 2014].

L'algorithme DBSCAN regroupe les points en cluster par densité. Le principal avantage de cet algorithme est qu'il ne nécessite pas de préciser le nombre de clusters à identifier. Cependant, il n'est pas capable de gérer des clusters de densités différentes, et cela est nécessaire sur les données de recommandation sociale car certaines préférences sont beaucoup plus représentées que d'autres chez les utilisateurs. L'algorithme espérance-maximisation se base sur la *Vraisemblance* statistique de l'appartenance d'un point à un cluster. Le principal inconvénient de cet algorithme est qu'il est très complexe en temps et qu'il converge trop souvent sur un minimum local qui ne représente pas une solution acceptable. Enfin, l'algorithme *k-means* est le plus populaire dans le domaine de la recommandation de par sa robustesse. En effet, cette méthode donne de bons résultats sur tout type de données (données incomplètes, aberrantes, à haute dimension, ...) [Kim and Ahn, 2008, Haydar et al., 2012]. Les deux principaux inconvénients de cette méthode sont qu'elle nécessite de connaître à l'avance le nombre de clusters que l'on souhaite identifier et que le résultat dépend de l'initialisation de l'algorithme. La figure 2.1 présente un ensemble de données en deux dimensions que l'on souhaite regrouper en clusters à l'aide de l'algorithme *k-means* (chaque point représente une donnée) :

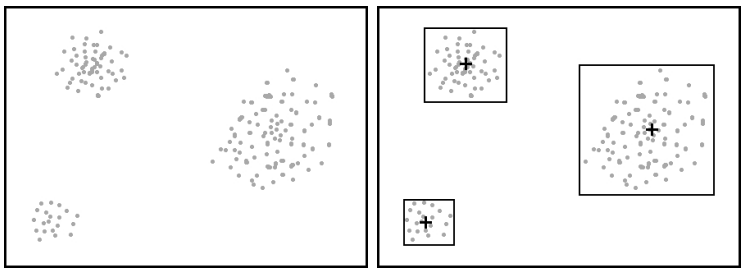


FIGURE 2.1 – Exemple d'exécution d'un algorithme de clustering *k-means*

Dans le cadre de la recommandation sociale, cet algorithme est utilisée pour rassembler les utilisateurs ou les ressources similaires en clusters. L'approche de recommandation associée ne sélectionne plus les k plus proches voisins de l'utilisateur actif mais exploite les préférences de l'ensemble des utilisateurs qui partagent son cluster [Sarwar et al., 2002, Esslimani et al., 2009]. Cette méthode est donc plus adaptée aux grandes quantités de données (passage à l'échelle) que les méthodes de FC à base de mémoire [Xue et al., 2005]. En effet, l'étape de la sélection des k voisins les plus similaires est la plus coûteuse de l'approche KNN puisqu'elle nécessite de comparer les préférences de tous les utilisateurs deux à deux, et l'utilisation d'un modèle regroupant les utilisateurs les plus similaires entre eux permet de réduire considérablement l'espace de recherche des k voisins les plus similaires.

Néanmoins, dans les systèmes de recommandation, le *clustering* est également souvent utilisé, a posteriori, pour expliquer la fluctuation de la qualité des recommandations fournies aux utilisateurs. Dans [Haydar et al., 2012], les auteurs distinguent plusieurs clusters d'utilisateurs et isolent ensuite les caractéristiques communes aux utilisateurs d'un même cluster, recevant des recommandations de mauvaise qualité.

La factorisation de matrice

La factorisation de matrice est l'approche à base de modèle la plus répandue dans le domaine de FC, notamment grâce à sa capacité à représenter de très grandes quantités de données [Koren et al., 2009, Takacs et al., 2009]. Cette technique a également permis d'obtenir les meilleurs résultats lors de la compétition organisée par Netflix [Bennett et al., 2007] et lors du tournoi KDD de 2011 [Dror et al., 2012], affichant des scores supérieurs à ceux obtenus avec les méthodes à base de K plus proches voisins. Depuis, les techniques de factorisation de matrice n'ont cessé d'être perfectionnées et connaissent actuellement un regain d'intérêt avec de nouvelles techniques inspirées des réseaux de neurones profonds [van Baalen, 2016, He et al., 2017, Xue et al., 2017].

L'idée générale de la factorisation de matrice est de modéliser les interactions utilisateur-ressource au travers de facteurs qui représentent des caractéristiques latentes des utilisateurs et des ressources, comme des classes de préférence pour les utilisateurs ou des catégories pour les ressources.

Dans la pratique, la factorisation de matrice factorise la matrice de notes N (dimension $n*m$) en deux matrices possédant chacune k caractéristiques latentes [16] : la matrice P (dimension $n*k$) représentant les facteurs des utilisateurs et la matrice Q (dimension $m*k$) représentant les facteurs des ressources [Billsus and Pazzani, 1998, Yu et al., 2014]. La figure 2.2 illustre le rapport entre les matrices N, P et Q.

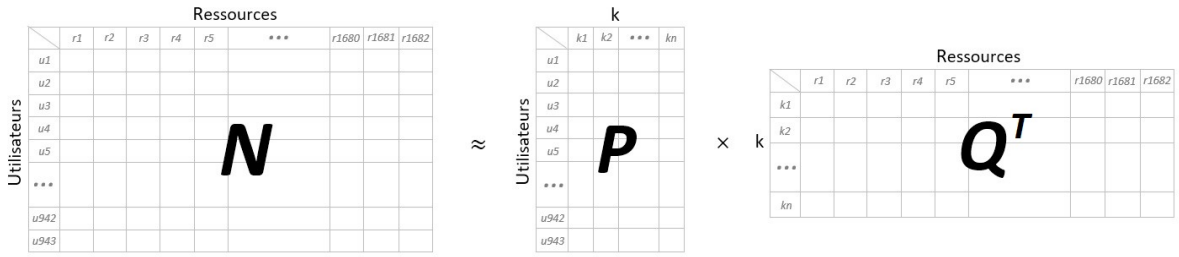


FIGURE 2.2 – Factorisation de matrice

L'objectif est de former les matrices P et Q dont le produit reconstruit le plus précisément possible la matrice N : $PQ^T \approx N$. La précision de la factorisation est évaluée en fonction de l'erreur sur chaque note $n_{u,r}$ dans la matrice N, s'appuyant sur l'équation (2.2) :

$$e_{u,r} = n_{u,r} - P_u Q_r^T, \tag{2.2}$$

16. La lettre k est traditionnellement utilisée par l'état de l'art pour désigner le nombre de caractéristiques latentes d'un modèle de FM et n'a aucun lien avec la lettre k utilisée précédemment pour décrire l'algorithme *k-means*.

avec P_u le vecteur associé à l'utilisateur u dans P et Q_r le vecteur associé à la ressource r dans Q.

Le nombre k de caractéristiques latentes que l'on souhaite utiliser est déterminé en amont de l'exécution de l'algorithme de factorisation, en fonction des données sur lesquelles l'algorithme est appliqué. On peut donc optimiser ce nombre de caractéristiques latentes pour obtenir le modèle le plus performant possible, mais cette optimisation est extrêmement coûteuse en temps de calcul. Une solution à cela est d'optimiser le nombre de caractéristiques latentes sur un sous-ensemble de données sélectionnées aléatoirement, et ensuite d'inférer le nombre optimal de caractéristique nécessaire pour représenter l'ensemble du jeu de données [Gemulla et al., 2011].

La faible taille des matrices obtenues à l'issue de la factorisation permet de stocker de manière plus efficace les informations nécessaires à la recommandation sociale (le modèle). Les plus gros services du Web comme Netflix, Amazon ou encore Google utilisent la factorisation de matrice pour représenter leurs données de manière plus concise, et gagnent ainsi en temps de calcul et en espace de stockage mémoire, tout en conservant une qualité de recommandation élevée.

Il existe plusieurs techniques de factorisation de matrice parmi lesquelles on trouve la technique de descente de gradient stochastique *SGD* [Ruszczynski and Syski, 1983, Koren et al., 2009] et la technique des moindres carrés alternés *ALS* [Paatero, 1997, Hu et al., 2008]. Ces différentes techniques cherchent toutes à optimiser la procédure de factorisation de la matrice N et ont donc pour but de minimiser la fonction d'erreur quadratique suivante :

$$argmin_{p,q} \sum_{u,r \in L} (e_{u,r})^2 \tag{2.3}$$

où L est l'ensemble des paires (u, r) pour lesquelles $n_{u,r}$ est connue.

La prise en compte des biais

Le processus de minimisation de l'erreur quadratique de l'équation (2.3) a pour objectif de capturer les interactions entre les utilisateurs et les ressources à l'origine des différences observées sur les notes dans la matrice N. Cependant, la plupart des différences observées dans les notes des utilisateurs sont liées soit à l'utilisateur en question, soit à la ressource sur laquelle la note porte, sous la forme d'un biais indépendant des interactions pures entre l'utilisateur et la ressource. Les données de préférence sont très exposées à ce type d'effet. Par exemple, nous avons déjà évoqué la présence de biais sur les utilisateurs ou les ressources pouvant biaiser les données présentes dans la matrice N. Cela s'explique simplement par différents niveaux d'exigence chez les utilisateurs et différents niveaux de qualité pour les ressources. Dans ce cas, il n'est plus correct d'expliquer toute l'information d'une note par la simple interaction utilisateur-ressource de la forme $P_u Q_r^T$. Il convient d'essayer d'identifier la part de chaque note qui peut être expliquée par un biais de l'utilisateur ou de la ressource. Cela permet donc ensuite d'isoler l'interaction pure entre l'utilisateur et la ressource.

Le biais utilisateur-ressource $b_{u,r}$ présent dans chaque note $n_{u,r}$ est alors noté :

$$b_{u,r} = \mu + b_r + b_u \tag{2.4}$$

La note moyenne du système est μ et les paramètres b_u et b_r correspondent aux déviations de l'utilisateur et de la ressource par rapport à la moyenne. Pour illustrer cette formule dans un système de recommandation de films, supposons que l'on veuille estimer la note de l'utilisateur actif sur la ressource Forrest Gump. Admettons que la moyenne des notes, tous films confondus, soit de 3,7 / 5. Le film Forrest Gump est très apprécié en moyenne, avec 0,6 point de plus que les

autres films en moyenne. De son côté, l'utilisateur actif n'est pas un utilisateur sévère, il a pour habitude d'attribuer 0,2 points de plus que les autres utilisateurs en moyenne aux films. Dans ce cas, l'estimation de la note de l'utilisateur actif sur le film Forrest Gump serait 4,5 / 5 (3,7 + 0.6 + 0.2). Les biais s'ajoutent à l'équation (2.2) du calcul de l'erreur de la manière suivante :

$$e_{u,r} = n_{u,r} - (b_{u,r} + p_u^T q_r) \tag{2.5}$$

La prise en compte des biais utilisateur et ressource dans le calcul du modèle permet une réelle amélioration de la précision du modèle obtenu. Il existe d'autres manières plus évoluées d'inclure les biais de manière plus précise, notamment en considérant que le biais de l'utilisateur et de la ressource évoluent en fonction du temps [Koren, 2010].

La régularisation

Dans le domaine de l'apprentissage automatique, le sur-apprentissage est un problème récurrent. Intuitivement, le calcul d'un modèle présentant une erreur très faible peut sembler intéressant, mais en pratique, un modèle parfait des données passées (notes connues) sera moins apte à prédire les données futures (prédictions) qu'un modèle plus approximatif. En d'autres termes, parmi tous les modèles permettant de représenter une solution acceptable au problème, il faut sélectionner le modèle le moins complexe. En recommandation sociale, la faible quantité d'information disponible dans la matrice N permet même aux modèles les plus simples de représenter parfaitement les données présentes dans la matrice N.

Pour prévenir ce phénomène, lors de l'apprentissage d'un modèle de factorisation de matrice, on utilise un terme de régularisation dans l'expression de la fonction d'erreur quadratique. Parmi tous les termes de régularisation, deux ont été mises au point spécialement pour être appliquées sur des matrices creuses : les régularisations L_1 et L_2.

La régularisation L_1 [Zheng et al., 2004] s'intéresse aux valeurs absolues des données présentes dans les matrices P et Q, tandis que la régularisation L_2 s'intéresse aux carrés des valeurs [Nigam, 1999] (équations (2.6) et (2.7)). La régularisation L_2 est donc plus stricte que la régularisation L_1, ce qui la rend plus sensible aux valeurs extrêmes.

Un algorithme de factorisation de matrice utilisant la régularisation L_1 cherche à minimiser la fonction d'erreur suivante :

$$min_{P,Q} \sum_{n_{u,r} \in N} [(e_{u,r})^2 + \lambda * (||P_u|| + ||Q_r||)], \tag{2.6}$$

avec λ le coefficient permettant de moduler l'impact du terme de régularisation lors de l'apprentissage du modèle.

La régularisation L_1 a montré de meilleurs résultats dans certaines applications telles que la modélisation d'un signal épars [Liu and Ihler, 2011]. Bien que la régularisation L_2 soit plus sensible aux valeurs extrêmes dans les données [17], elle reste en pratique la régularisation la plus utilisée dans le domaine du FC car elle permet d'obtenir de meilleurs résultats. Cela s'explique par le fait que la régularisation L_2 est plus adaptée aux jeux de données comprenant un écart important entre le nombre de lignes et le nombre de colonnes de la matrice N.

Un algorithme de factorisation de matrice utilisant la régularisation L_2 cherche à minimiser la fonction d'erreur suivante :

$$min_{P,Q} \sum_{n_{u,r} \in N} [(e_{u,r})^2 + \lambda * (||P_u||^2 + ||Q_r||^2)] \tag{2.7}$$

17. Les différences d'opinion et les préférences extrêmes sont naturelles dans un système de recommandation sociale

En pratique, la régularisation L_2 semble toujours donner de meilleurs résultats et reste donc la plus utilisée. Nous utiliserons donc l'équation (2.7) par défaut pour la suite de ce travail. L'état de l'art ne nous a pas permis d'avoir un aperçu de l'impact des différents types de régularisation sur les recommandations faites aux GSU. Nous avons donc réalisé des expérimentations dans le chapitre 4 de cette thèse pour obtenir cette information.

La technique ALS

En se focalisant sur la fonction d'erreur quadratique à minimiser, on voit qu'il y a deux types de variables à optimiser : les variables de la matrice P et celles de la matrice Q. De plus, ces variables sont liées, puisque c'est la multiplication de P par Q^T qui permet de recomposer la matrice originale. Le point important sur lequel s'appuie la technique ALS est le suivant : si on fixe la matrice P pour optimiser la matrice Q seule, le problème est alors réduit à une simple régression linéaire. Dans notre cas, il s'agit de trouver β (qui représente Q) tel qu'il minimise l'erreur au carré $||N - P\beta^T||^2$ avec N et P connus. La solution est donnée par la formule de la technique ordinaire des moindres carrés : $\beta = \left(P^T P\right)^{-1} P^T N$. Il s'agit du principe de fonctionnement de la technique des moindres carrés alternés. C'est un processus d'optimisation itératif à deux étapes. A chaque itération, on fixe Q pour optimiser P, puis on fixe P pour optimiser Q. Puisque la solution obtenue avec la technique ordinaire des moindres carrés est unique et garantit une erreur minimale, pour chaque étape, la fonction de l'erreur quadratique générale ne peut que diminuer ou rester stable, elle ne peut pas augmenter, et cela jusqu'à la convergence de l'algorithme ALS. La technique ALS permet donc d'atteindre un minimum local qui dépend de l'état d'initialisation des matrice P et Q.

La mise à jour des valeurs dans les matrices P et Q au cours de l'optimisation se fera alors de la manière suivante pour chaque note $n_{u,r}$:

$$P_u = \left(Q_r^T \times Q_r + \lambda||P_u||^2\right)^{-1} \times Q_r^T \times n_{u,r}, \tag{2.8}$$

avec Q_r fixé.

$$Q_r = \left(P_u^T \times P_u + \lambda||Q_r||^2\right)^{-1} \times P_u^T \times n_{u,r}, \tag{2.9}$$

avec P_u fixé.

Puisque chaque ligne de la matrice P ne dépend pas des autres lignes dans la matrice P, il est possible de calculer toutes les lignes simultanément, ce qui permet une large parallélisation de cet algorithme.

Les tableaux 2.5 et 2.6 présentent les matrices P et Q obtenues lorsque l'on factorise la matrice du tableau 2.1 en représentant chaque utilisateur et chaque ressource par $k = 2$ caractéristiques (c_1,c_2) (technique ALS).

	c_1	c_2
u_1	1,77	0,54
u_2	0,31	2,66
u_3	1,55	0,71
u_4	1,8	0,69

TABLE 2.5 – Matrice P

	r_1	r_2	r_3	r_4	r_5	r_6
c_1	2,28	2,41	0,12	0,67	0,22	1,29
c_2	0,51	0,11	1,78	1,76	1,81	1,69

TABLE 2.6 – Matrice Q

La technique SGD

La descente de gradient stochastique (SGD) est une technique d'optimisation qui est très largement utilisée dans le domaine de l'apprentissage automatique. La technique SGD apprend les matrices en évaluant de manière itérative l'erreur $e_{u,r}$ pour chaque note $n_{u,r} \in N$. A chaque itération, le gradient de la fonction d'erreur quadratique (cf. équation (2.7)) est recalculé en fonction des variables à optimiser (P et Q), et les met à jour dans la direction opposée à celle du gradient de la fonction d'erreur quadratique, de manière à minimiser cette fonction d'erreur. Cette opération est répétée jusqu'à la convergence de l'algorithme sur un minimum local, de la même manière que la technique ALS. La technique SGD montre de très bons résultats pour l'optimisation de modèles à base de factorisation de matrice, notamment dans le domaine de la recommandation [Koren et al., 2009]. A l'inverse de la technique ALS, la technique SGD ne peut pas être parallélisée puisque les deux matrices P et Q évoluent à chaque étape.

La mise à jour des caractéristiques des utilisateurs et des ressources dans les matrices P et Q est décrite dans les équations (2.10) et (2.11).

$$P_u = P_u + \alpha(e_{u,r} * Q_r - \lambda * P_u), \tag{2.10}$$

$$Q_r = Q_r + \alpha(e_{u,r} * P_u - \lambda * Q_r), \tag{2.11}$$

avec α le taux d'apprentissage, qui représente la vitesse à laquelle l'algorithme apprend.

Les tableaux 2.7 et 2.8 présentent un exemple de factorisation obtenu à partir de la matrice du tableau 2.1 en représentant chaque utilisateur et chaque ressource par $k = 2$ caractéristiques (c_1,c_2) (technique *SGD*).

	c_1	c_2
u_1	1,32	0,94
u_2	0,52	1,87
u_3	0,98	1,21
u_4	1,93	0,57

TABLE 2.7 – Matrice P

	r_1	r_2	r_3	r_4	r_5	r_6
c_1	2,10	2,41	0,69	0,27	0,22	1,09
c_2	0,7	0,11	1,6	1,03	1,73	1,12

TABLE 2.8 – Matrice Q

Le calcul de la prédiction

A partir des matrices P et Q ainsi obtenues, il est très simple de calculer une prédiction de note pour un utilisateur et une ressource. Par exemple, la valeur estimée pour la note de l'utilisateur u_3 sur la ressource r_2 sera :

$$\begin{aligned} \textit{Prédiction}_{u_3,r_2} &= \sum_{c \in C} P_{u_3,c} * Q_{c,r_2} \\ &= P_{u_3,c_1} * Q_{c_1,r_2} + P_{u_3,c_2} * Q_{c_2,r_2} \\ &= 1,55 * 2,41 + 0,71 * 0,11 \\ &= \mathbf{3,81} \end{aligned}$$

La factorisation de matrice présente de nombreux avantages. Le premier est la rapidité de calcul d'une prédiction ainsi que la précision de cette dernière [Koren et al., 2009]. Les avantages suivants sont communs à toutes les approches à base de modèle. Ces approches réduisent l'impact du problème du passage à l'échelle, ce qui favorise leur exécution sur les jeux de données actuels. De plus, le stockage du modèle nécessite moins d'espace mémoire que le stockage de la matrice N complète (approche à base de mémoire). A l'inverse, l'inconvénient de ces approches est qu'il est nécessaire de recalculer le modèle si l'on souhaite prendre en compte les nouvelles préférences émises par les utilisateurs.

2.1.1.4 Tableau comparatif des techniques de filtrage collaboratif

Je présente ici de manière synthétique les deux approches que j'ai détaillées ci-dessus. J'y résume le type de données collectées sur l'utilisateur, les principales techniques utilisées ainsi que les avantages et les inconvénients de ces approches.

	Techniques utilisées	**Avantages**	**Inconvénients**
FC à base de mémoire	- Calcul du voisinage	- dynamicité	- Démarrage à froid - Complexité en temps / mémoire - Passage à l'échelle
FC à base de modèle	- Clustering - Factorisation de matrice	- Rapidité de calcul de la prédiction - Passage à l'échelle - qualité des recommandations	- Construction du modèle complexe en temps / mémoire - Mise à jour complexe du modèle - Démarrage à froid - Perte d'informations

2.1.1.5 Les systèmes hybrides

L'hybridation de SR est le fait d'utiliser plusieurs approches (approche sociale, à base de contenu, à base de connaissance, etc.) au sein d'un même système. Dans [Burke, 2002], l'auteur présente sept techniques d'hybridation. On peut citer par exemple la méthode par commutation

qui consiste à sélectionner l'approche la plus adaptée, en fonction d'une heuristique à définir, parmi une liste d'approches implémentées (illustrée sur la figure 2.3), ou encore la méthode par pondération qui combine les résultats de plusieurs approches différentes en pondérant de manière spécifique l'impact de chaque approche sur la recommandation (illustrée sur la figure 2.4).

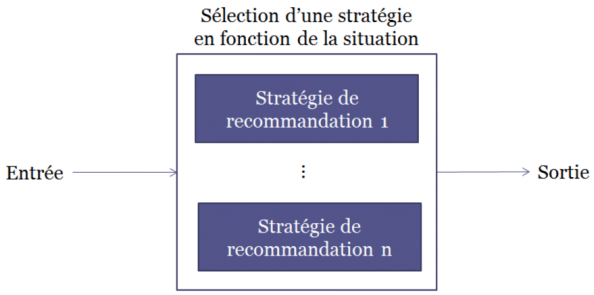


FIGURE 2.3 – La technique hybride des commutations

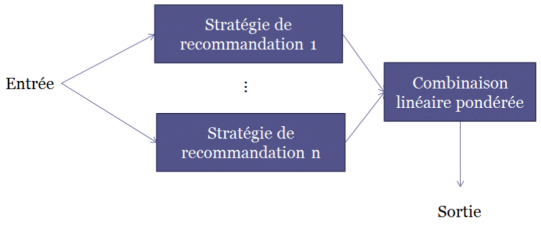


FIGURE 2.4 – La technique hybride par pondération

L'avantage de ce type de systèmes est qu'il permet de faire profiter certaines approches des avantages d'autres approches. Par exemple, en combinant un filtrage collaboratif et un filtrage par contenu, le filtrage par contenu permet de diminuer l'impact du démarrage à froid sur les recommandations fournies aux utilisateurs. C'est le cas du SR *Fab*, proposé par les auteurs de [Balabanović and Shoham, 1997]. Cette solution au problème du démarrage à froid du FC a été confirmée à plusieurs reprises [Pazzani and Billsus, 2007, Volkovs et al., 2017].

Il existe d'autres techniques d'hybridation [Su and Khoshgoftaar, 2009], mais elles ont toutes l'inconvénient d'être complexes à implémenter de part la quantité de paramètres à optimiser pour obtenir de bonnes performances. Prenons l'exemple de la technique hybride qui a remporté la compétition Netflix en 2008, proposée par l'équipe "BellKor's Pragmatic Chaos" [Koren, 2009, Töscher et al., 2009]. Cette technique utilise un arbre de décision boosté avec le gradient pour hybrider les résultats de plus de 500 approches de recommandation.

2.1.1.6 Les quatre propriétés nécessaires à une recommandation sociale de qualité

Nous venons de décrire le fonctionnement de plusieurs approches de recommandation sociale, mais comment comprendre l'origine des forces et des faiblesses de ces différentes approches ? Pour nous aider dans cette démarche, M. Pennock and E. Horvitz ont défini un cadre formel composé de 4 grands principes que devraient respecter chaque algorithme de FC [Pennock and Horvitz, 1999].

Ces principes sont les suivants :

— Propriété 1, *l'universalité* : Un algorithme de FC doit toujours être en mesure de retourner des estimations de préférence, quel que soit l'utilisateur ou la ressource concerné(e). De nombreux problèmes comme le démarrage à froid ou encore le problème des GSU, central à ce travail, empêchent le respect de cette propriété par les systèmes de FC classiques.

— Propriété 2, *l'unanimité* : Si une ressource j reçoit toujours des notes strictement supérieure à k, alors la note prédite pour l'utilisateur actif pour la ressource j sera strictement supérieure à k. Cette propriété n'est pas respectée lorsque le système de FC utilise un coefficient de corrélation pour déterminer les similarités entre les utilisateurs. Par exemple, un utilisateur négativement corrélé à tous les autres utilisateurs dans un système utilisant les corrélations négatives peut obtenir une prédiction inférieure à k sur la ressource j. Les approches à base de similarité ou à base de factorisation de matrice respectent cette propriété.

— Propriété 3, la *stabilité* : L'ordre des préférences d'un utilisateur sur les ressources doit rester inchangé lorsqu'une nouvelle ressource est ajoutée. Par exemple, si un algorithme de FC prédit que l'utilisateur actif devrait préférer la ressource j à la ressource k, alors l'ajout d'une ressource m, que l'utilisateur actif n'a pas encore évaluée, à la base de données, ne doit pas modifier l'ordre des préférences estimées, et la prédiction de la préférence sur la ressource j devra rester supérieure à celle sur la ressource k. L'ordre des préférences peut néanmoins être affecté si l'utilisateur actif évalue de nouvelles ressources, complétant ainsi les informations quant à ses préférences.
Cette propriété est respectée par tous les algorithme de FC, mais est-elle en adéquation avec la nature des données de recommandation ? Dans un système à base de FC, les notes émises par les utilisateurs sont incertaines [Amatriain et al., 2009], ce qui implique l'instabilité de l'ordre des préférences réelles d'un utilisateur. Bien que cette propriété soit respectée par toutes les approches de recommandation, nous pensons que tant que l'approche de recommandation exploite des données incertaines, cette propriété ne garantit pas de bonnes recommandations pour les utilisateurs.

— Propriété 4, *invariance de l'échelle* : Depuis de nombreuses années, l'économie [Arrow, 1963, Sen, 1986] considère que l'échelle de notation interne d'un utilisateur n'est pas comparable avec celle des autres utilisateurs. Ce phénomène concerne également le domaine de la recommandation sociale. Lorsque deux utilisateurs utilisent la même note, cela ne signifie pas toujours la même chose, même à échelle identique. Par exemple, certains utilisateurs peuvent utiliser majoritairement la partie inférieure de l'échelle de notation, tandis que d'autres utilisateurs, peut-être moins sévères, n'utilisent que la partie supérieure de cette même échelle.
Une manière d'inclure en partie cette propriété dans un algorithme de FC est de soustraire la moyenne de chaque utilisateur à chacune de ses notes. Cela permet de recentrer les échelles de notation des utilisateurs et d'améliorer la comparabilité de leurs notes. Il s'agit du "biais utilisateur" que nous avons déjà évoqué.

Le non respect de l'une de ces propriétés par un système de FC peut être à l'origine de mauvaises recommandations. Les différentes approches de recommandation que nous avons présentées respectent partiellement ces quatre propriétés. Plus précisément, la propriété 3 est dicutable et la propriété 4 n'est qu'en partie respectée. En effet, il n'est pas possible d'assurer la stabilité des préférences des utilisateurs en raison de l'imprécision de ces dernieres lors de la notation, ce

qui remet en cause la légitimité de la propriété 3. De plus, les différentes propositions visant à intégrer les biais des utilisateurs et des ressources dans le calcul des prédictions sont minimalistes. La propriété 4 n'est donc pas pleinement respectée. Par exemple, à notre connaissance, les approches intégrant le biais utilisateur ne considèrent qu'un seul et unique biais par utilisateur, même si celui-ci peut évoluer au cours du temps. Or, un utilisateur peut avoir des biais différents en fonction de son contexte (météo, maladie, etc.) mais également en fonction du type de ressource (genre de films, auteurs de livres, etc.) [Adomavicius and Tuzhilin, 2011]. Ces propriétés sont un outil d'analyse pour expliquer les mauvaises performances des certains algorithmes de FC sur certains utilisateurs. Pour étudier cela il est donc nécessaire d'évaluer la qualité des recommandations fournies par un algorithme de FC.

2.1.1.7 Les méthodes d'évaluation

Les méthodes d'évaluation sont utilisées pour comparer l'efficacité de différentes approches de recommandation [Avazpour et al., 2014]. Dans cette partie, je présente la manière dont les performances des différentes approches sont classiquement évaluées, ainsi que les différentes métriques utilisées.

Le protocole : évaluation *online* ou *offline* ?

Le protocole d'évaluation d'un système de recommandation peut être soit *offline*, soit *online*. Le protocole *online* propose des recommandations à des utilisateurs réels et observe leur comportement pour évaluer la pertinence de la recommandation. Le protocole *online* requiert donc d'avoir accès à un système en ligne et à une communauté d'utilisateurs conséquente. La méthode d'évaluation *online* la plus populaire est la méthode par A/B testing. Cette méthode consiste à mettre en ligne plusieurs versions d'un système de recommandation pour comparer les résultats des différentes versions sur les utilisateurs réels du site en ligne. C'est pourquoi cette méthode est la plus réputée pour évaluer la performance des algorithmes de recommandation. Son principal inconvénient est que le temps nécessaire pour évaluer et comparer les performances des différentes versions proposées du SR est important [18], et également qu'il est difficile de disposer d'utilisateurs réels.

A l'inverse, le protocole *offline* nécessite uniquement de posséder un jeu de données composé de préférences d'utilisateurs sur des ressources. Cette technique sépare les préférences des individus en deux sous-ensembles : le jeu de données d'apprentissage et le jeu de données de test. Pour comparer l'efficacité des différentes approches de recommandation, on leur donne accès au jeu de données d'apprentissage puis on évalue leur capacité à inférer les données du jeu de test. Cette méthode s'exécute en peu de temps et permet donc d'avoir des résultats rapidement. Cependant, [Gomez-Uribe and Hunt, 2015] montre que les méthodes qui obtiennent les meilleurs résultats en évaluation *offline* ne sont que très rarement celles qui sont les plus efficaces dans un protocole *online*.

Dans le travail mené dans cette thèse, nous nous intéressons à un sous-ensemble des utilisateurs : les GSU. Une validation *online* n'est possible que si cette sous-partie des utilisateurs est suffisamment grande pour être divisée en plusieurs groupes de test. Or, nous n'avons pas eu accès à un service en ligne possédant ce très grand nombre d'utilisateurs, nous avons donc réalisé une évaluation *offline* des approches de recommandation que nous avons implémentées. Nous présentons un état de l'art des métriques les plus populaires pour l'évaluation de la qualité des

18. https ://medium.com/netflix-techblog/netflix-recommendations-beyond-the-5-stars-part-2-d9b96aa399f5

prédictions d'un algorithme de recommandation dans un protocole *offline* dans la suite de cette section.

MAE et NMAE

La MAE (*Mean Absolute Error* ou Moyenne des écarts absolus) représente l'erreur moyenne d'un SR dans la tâche de prédiction de préférences. Il s'agit de la mesure historique utilisée pour évaluer les SR [Breese et al., 1998]. Elle représente la moyenne des différences absolues entre les prédictions et les notes réelles :

$$MAE = \frac{\sum_{u,r} |prédiction_{u,r} - n_{u,r}|}{m},$$

où m représente le nombre de prédictions évaluées. Plus la MAE est faible, meilleur est l'algorithme de recommandation.

La NMAE (MAE normalisée), quant à elle, normalise la MAE en fonction des notes min et max que le système a enregistrées :

$$NMAE = \frac{MAE}{n_{max} - n_{min}},$$

où n_{max} et n_{min} sont les notes maximale et minimale théoriques. Cette mesure a l'avantage de permettre de comparer des systèmes dont les échelles de notation sont différentes.

RMSE

La RMSE (*Root Mean Squarred Error* ou Racine de la moyenne des écarts au carrés) repose sur le même principe que la MAE, à la différence près qu'elle élève au carré les erreurs de prédiction. Cela a pour impact d'augmenter l'influence des fortes erreurs sur la métrique d'évaluation. La RMSE se calcule de la manière suivante :

$$RMSE = \sqrt{\frac{1}{m} \sum_{u,r} (prédiction_{u,r} - n_{u,r})^2},$$

Cette métrique d'évaluation a été popularisée par la compétition Netflix [19].

Le principal inconvénient des métriques citées précédemment est qu'elles proposent une vision moyenne des résultats du système. Elles ne permettent pas de faire la différence entre un système proposant de très bonnes recommandations en parallèle de très mauvaises recommandations et un système proposant uniquement des recommandations de qualité moyenne. D'autres mesures ont alors été proposées, parmi lesquelles figure le HitRatio.

Le HitRatio

Le HitRatio des recommandations représente la proportion de prédictions correctes effectuées par un algorithme de recommandation.

Une prédiction correcte est une $prédiction_{u,r}$ dont l'écart à la valeur réelle $n_{u,r}$ est inférieur à E_{max}, un paramètre fixé a priori.

$$HitRatio = \frac{\sum_{u,r} (|prédiction_{u,r} - n_{u,r}| < E_{max})}{m}$$

19. www.netflixprize.com

La mesure de HitRatio est comprise dans l'intervalle [0 ;1], où plus la valeur est proche de 1, plus l'algorithme est performant.

Le HitRatio permet d'avoir un aperçu de la proportion de recommandations correctes fournies par l'algorithme, là où la RMSE donne un aperçu moyen ne permettant pas de connaître le nombre de bonnes recommandations. Le HitRatio mesure donc une information différente.

L'inconvénient de cette mesure est qu'elle n'est pas impactée lorsque le système fournit une recommandation très écartée de la note réelle de l'utilisateur. Une note légèrement écartée (de plus de E_{max}) aura le même impact.

Une mesure de classement : NDPM

La mesure NDPM (*Normalized Distance-Based Performance Measure*) calcule l'erreur de classement de deux listes de préférences ordonnées [Yao, 1995]. On peut alors comparer l'ordre des notes prédites avec l'ordre des notes réelles. Cette métrique permet d'évaluer si l'algorithme de recommandation a estimé correctement l'ordre de préférence des ressources pour un utilisateur donné [Shani and Gunawardana, 2009].

Voici le détail du calcul de la NDPM :

$$C^u = \sum_{ij} sgn^2(r_{u,i} - r_{u,j})$$

$$C^+ = \sum_{ij} sgn(r_{u,i} - r_{u,j}) = sgn(\overline{r}_{u,i} - \overline{r}_{u,j})$$

$$C^- = \sum_{ij} sgn(r_{u,i} - r_{u,j}) \neq sgn(\overline{r}_{u,i} - \overline{r}_{u,j})$$

$$C^{u0} = C^u - (C^+ + C^-)$$

$$NDPM = \frac{2C^- + C^{u0}}{2C^u}$$

$$\forall x \in \mathbb{R}, sgn(x) = \begin{cases} -1 & \text{si x} < 0 \\ 0 & \text{si x} = 0 \\ 1 & \text{si x} > 0 \end{cases} \tag{2.12}$$

Rappelons que la fonction *sgn* est définie selon l'équation 2.12. $r_{u,i}$ est le rang de la ressource i dans la liste ordonnée des préférences de l'utilisateur u. On utilise l'écriture $\overline{r}_{u,i}$ pour désigner le rang de la ressource i dans la liste des préférences estimées par l'algorithme pour l'utilisateur u. Le symbole C^u représente le nombre de paires de ressources qui ne partagent pas le même rang selon les préférences exprimées par l'utilisateur. Deux ressources peuvent posséder le même rang si l'utilisateur a donné la même note aux deux ressources. On peut donc considérer C^u comme le nombre de fois où une différence de classement devrait être constatée. On a ensuite le symbole C^+ qui représente le nombre de paires de ressources correctement ordonnées par l'algorithme en comparaison avec les préférences réelles de l'utilisateur et à l'inverse, C^- le nombre de paires mal ordonnées par l'algorithme de prédiction. C_{u0} représente alors le nombre de paires de ressources ordonnées selon les préférences réelles de l'utilisateur et non ordonnées selon les estimations des algorithmes de recommandation. Cette mesure permet d'évaluer la capacité de l'algorithme de recommandation a correctement hiérarchiser les préférences d'un utilisateur. Le principal inconvénient de cette méthode est que ses résultats dépendent de la manière dont sont arrondies les valeurs prédites pour que les listes de notes prédites et réelles soient comparables. De plus,

en général, le nombre de rangs possibles (correspondant aux notes réelles possibles) est faible, ce qui limite les comparaisons des rangs des ressources dû au grand nombre de ressources qui partagent le même rang.

Il n'y a pas de consensus à propos de la métrique à utiliser pour évaluer un système de recommandation, les métriques que nous avons présentées ont toutes des avantages et des inconvénients [Shani and Gunawardana, 2009]. Les travaux du domaine des SR utilisent en général plusieurs métriques pour donner une meilleure analyse de leurs résultats. Nous analysons donc les résultats de nos expérimentations avec plusieurs métriques d'évaluation.

2.1.2 Les limites de la recommandation sociale

Bien que les performances des SR à base de FC sont satisfaisantes [Castagnos et al., 2013], il reste des défis à relever dans ce domaine. La liste que nous allons dresser est non exhaustive, mais elle en référence les principaux défis [Su and Khoshgoftaar, 2009, Beel et al., 2016]. Nous avons classé ces défis en quatre grandes catégories : les problèmes liés au manque de données, le problème du passage à l'échelle, le problème des utilisateurs malveillants et enfin le problème central à cette thèse, les *Grey Sheep Users*.

2.1.2.1 Le manque de données

Les matrices de notes comptent en moyenne plus de 95% de valeurs non renseignées. Le problème du manque de données est donc commun à toutes les approches de filtrage collaboratif. La rareté des notes entraîne plusieurs sous-problèmes que nous détaillons ci-dessous.

Le démarrage à froid

Le démarrage à froid est un problème qui peut concerner soit les utilisateurs, soit les ressources. Le démarrage à froid utilisateur survient lorsque le système ne possède pas suffisamment d'informations (notes) sur un utilisateur pour pouvoir effectuer une recommandation pertinente [Adomavicius and Tuzhilin, 2005]. Plusieurs solutions ont été proposées pour traiter ce problème comme l'hybridation avec un système à base de contenu (section 2.1.1.5), l'utilisation d'utilisateurs représentatifs pour compléter les préférences manquantes [Aleksandrova et al., 2016], l'utilisation d'une approche à base de confiance [Haydar et al., 2012], ou plus simplement la proposition d'un coefficient de similarité moins sensible au manque de notes [Bobadilla et al., 2012b].

La couverture

La couverture concerne les prédictions et se définit comme le pourcentage de ressources non votées par l'utilisateur actif dont le système peut estimer la note. Lorsqu'une ressource n'a été notée que par très peu d'utilisateurs, la probabilité que cette ressource soit recommandée par l'algorithme est très faible (il faut qu'un voisin de l'utilisateur actif l'ait évaluée positivement). Lorsque beaucoup de ressources se trouvent dans ce cas, alors on fait face à un problème de couverture réduite.

Le voisinage

Le voisinage réduit ne touche quant à lui que les utilisateurs. Lorsque deux utilisateurs ne covotent pas les mêmes ressources, en raison du grand nombre de ressources, il n'est pas possible de calculer la similarité entre ces deux utilisateurs. Lorsqu'un grand nombre d'utilisateurs sont concernés, il devient difficile de trouver k voisins similaires à chaque utilisateur. Le système souffre alors du problème du voisinage réduit qui diminue la qualité des recommandations qu'il fournit. De la même façon, si les interactions (co-votes) entre les utilisateurs sont rares, alors une

approche à base de modèle ne parviendra pas à modéliser ces interactions et le modèle fournira des recommandations de mauvaise qualité.

2.1.2.2 La fiabilité des données

Les utilisateurs malveillants
L'un des inconvénients des approches de FC est que chacune des données de préférence est importante, en raison du problème du manque de données. Par exemple, exprimer un grand nombre de préférences positives sur une ressource augmente la probabilité qu'elle soit recommandée. Des utilisateurs malveillants peuvent profiter de ce biais pour qu'un SR recommande une ressource plus souvent qu'une autre. Ceci peut s'illustrer par exemple dans un SR de site de ventes en ligne sur lequel un fabricant peut être tenté d'évaluer positivement à plusieurs reprises ses produits pour qu'ils soient plus souvent recommandés que ceux de ses concurrents. Les préférences sont donc désormais filtrées pour empêcher ce genre d'abus. Il s'agit d'attaques visant à altérer le fonctionnement des systèmes de recommandation, mettant leur robustesse à l'épreuve [Wang and Tang, 2015].

L'imprécision des notes
Dans [Amatriain et al., 2009], les auteurs ont montré que les notes exprimées par les utilisateurs ne sont pas précises. Par exemple, lorsque l'on demande à un utilisateur de noter deux fois une même ressource à des moments différents, il pourrait l'évaluer différemment, tout en restant cohérent. Cette différence n'est pas seulement due au contexte de l'utilisateur, mais aussi à l'existence d'une échelle de notation discrète qui force les utilisateurs à choisir lorsque leur préférence se situe entre deux valeurs de note proposées.

2.1.2.3 Le passage à l'échelle

Le FC à base de mémoire souffre du problème du passage à l'échelle. Alors qu'il parvient à exploiter de petites quantités de données, lorsque l'on augmente le nombre de ressources et d'utilisateurs, le temps d'exécution de l'algorithme explose. Plusieurs solutions ont été adoptées pour contrer ce problème comme les approches modèles [Koren et al., 2009].

2.1.2.4 Les *Grey Sheep Users*

La majeure partie des utilisateurs correspondent à des utilisateurs dits *White Sheep Users*, c'est-à-dire des utilisateurs dont les préférences sont cohérentes avec celles de plusieurs autres utilisateurs. Pour ce type d'utilisateur, il est, en général, simple d'effectuer des recommandations de qualité. Cependant, dans la matrice de données, certains utilisateurs ont des préférences qui ne sont pas systématiquement en accord ou en désaccord avec un groupe d'utilisateurs [Claypool et al., 1999]. Ces utilisateurs présentent des préférences spécifiques (non partagées par une majorité d'utilisateurs). Même en isolant leurs plus proches voisins, ces utilisateurs n'auront que très peu de points communs avec ceux-ci. En conséquence, ces utilisateurs recevront très probablement de mauvaises recommandations. Ces utilisateurs sont appelés des *Grey Sheep Users* (GSU) et sont le centre d'intérêt de ce travail. L'enjeu est donc de les identifier pour intervenir de manière ciblée sur les recommandations qui leur seront faites pour en améliorer la qualité.

Pour lever les ambiguïtés sur la définition d'un GSU, il convient donc d'expliciter les notions d'atypisme, de déviance et de norme (section 1.2). Les sciences humaines telles que la sociologie, la psychologie, la philosophie ou encore l'histoire font état de nombreux travaux sur ces notions

centrales à la définition des GSU. Nous exposons une partie de ces travaux dans la section suivante pour apporter un regard nouveau sur le problème des GSU.

2.1.3 L'atypisme en sciences humaines : une étude de la différence

L'atypisme (ou phénomène de la différenciation) a été largement étudié dans les sciences humaines et sociales. C'est en sociologie que l'on trouve le plus de références à ce phénomène, ainsi que de multiples études sur la catégorisation des différentes formes d'atypisme [Merton, 1968, Cusson, 1992, Ogien, 1995]. En philosophie, la question de la marginalisation de l'individu et des conséquences sur sa place dans la société sont également en lien direct avec le phénomène que nous étudions [Wittgenstein and Ogden, 1921].

2.1.3.1 Un même phénomène, plusieurs points de vue...

Les personnes atypiques sont considérées de différentes manières en fonction du domaine dans lequel elles sont traitées. Déviants, marginaux, anormaux, atypiques, etc., autant de termes pour faire référence à ces personnes différentes des autres et que l'on ne parvient pas correctement à caractériser. Mais quelles sont les nuances cachées derrière chacune de ces formulations ?

Partons de la définition officielle du terme atypisme du dictionnaire Larousse [20] pour éclairer ce point.

Atypisme
nom masculin
Absence de conformisme relativement à un modèle que l'on prend comme référence.

Tout ce qui n'est pas conforme à un modèle servant de référence est alors considéré atypique. Dans cette thèse, nous avons pour principal objectif d'identifier et de mieux servir les utilisateurs possédant des préférences spécifiques ou atypiques en vue d'un système de recommandation sociale. L'aspect social de l'atypisme est donc prédominant pour le problème des GSU. Il existe un terme issu des domaines de la psychologie et de la sociologie qui s'adapte mieux à notre problème : la déviance [21].

Déviance
nom féminin
Position d'un individu ou d'un groupe qui conteste, transgresse et qui se met à l'écart de règles et de normes en vigueur dans un système social donné.

Trois conditions sont nécessaires pour qu'une personne soit considérée déviante [Mucchielli, 1999a] :

1. l'existence d'une norme sociale
2. l'existence d'une observation qui transgresse cette norme
3. l'existence d'une stigmatisation de cette transgression

20. http ://www.larousse.fr/dictionnaires/francais/atypie/6335
21. http ://www.larousse.fr/dictionnaires/francais/déviance/24988

Ces trois points représentent chacun des champs de recherche pour les sciences humaines telles que l'histoire [Massin, 1998, Duby, 1995] ou le droit [Robert et al., 1997, Mucchielli, 1999b] pour la définition des normes, ainsi que la philosophie [Wittgenstein and Ogden, 1921, Bergson, 2012], la psychologie [Amblard et al., 2011, Gosling and Ric, 1996] ou enfin la sociologie [Maunier, 1929, Becker, 1985] pour la définition des différents types de déviance et l'analyse de leurs stigmatisations. La transdisciplinarité de ce sujet implique un grand nombre de manières de le définir. Nous essayons dans cette partie de rassembler un maximum de points de vue sur le sujet et d'en synthétiser les idées.

Norme sociale, entre pluralité et relativisme

La norme correspond à une standardisation arbitraire des comportements propres à chaque société s'appuyant sur des codes sociaux, culturels, ethniques, législatifs, ou encore moraux. Il existe donc plusieurs normes et un même individu peut être déviant par rapport à une norme et non déviant par rapport à une ou plusieurs autres normes.

Les mentalités évoluent et cela impacte les normes. L'évolution des normes due aux progrès techniques ou à l'évolution des croyances religieuses et politiques en sont un très bon exemple. Dans nos sociétés occidentales, il y a encore seulement une centaine d'années, l'avortement était un crime jugé particulièrement cruel et sévèrement puni [Duby, 1995], la mendicité était un délit qui pouvait conduire un sans abris à une peine de prison accompagnée de travaux forcés [Damon, 1998], etc. Le temps et l'évolution des mœurs sont donc à l'origine d'un changement des normes qui sont alors relatives au temps.

Aujourd'hui encore, la norme sociale, par exemple en matière d'avortement, est très différente en fonction du pays dans lequel on se trouve. Il existe donc également un relativisme géographique qui s'applique à la norme sociale.

Enfin, les normes sociales sont également relatives au contexte précis de la situation dans laquelle un individu est jugé. Un individu vêtu intégralement de couleur verte est déviant toute l'année, sauf le soir de la Saint Patrick, où il est de coutume de porter du vert (dans certaines cultures).

Pascal écrivait : « le larcin, l'inceste, le meurtre des enfants et des pères, tout a eu sa place entre les actions vertueuses » [Pascal and Faugère, 1897]. La déviance d'un individu peut donc s'appliquer à de nombreuses normes qui varient en fonction de l'époque et du lieu dans lequel on se trouve.

En considérant la diversité de ces normes sociales, chaque personne est déviante vis-à-vis d'au moins une norme. Il est donc nécessaire de préciser la norme selon laquelle un individu est jugé déviant.

La déviance sous toutes ses formes

Au travers de la définition sociologique de la déviance (section 2.1.3.1), nous avons pu voir qu'elle représente un comportement ou une attitude qui transgresse ou conteste les normes en vigueur. La déviance peut être individuelle, lorsqu'un seul individu dévie de la norme, ou collective, lorsqu'ils sont plusieurs à le faire conjointement. Les individus qui respectent les normes

sociales peuvent quant à eux considérer les déviants comme une menace à l'intégrité de la société et vont les stigmatiser. Il existe trois grandes formes de déviance [Ogien, 1995] :

1. La délinquance. Cette forme de déviance caractérise les transgressions des normes sociales définies par la loi. Cette définition est la plus couramment utilisée dans les domaines du droit et de la criminologie. Le contexte de cette thèse ne permet pas d'aborder la déviance sous cette forme. Nous n'explorerons pas d'avantage ce pan de la recherche sur la déviance.
2. La marginalité. Plus souvent abordée en philosophie et en psychologie, cette forme de déviance s'applique si la norme est purement sociale ou religieuse et n'entraîne pas de sanction juridique. Elle peut néanmoins mener à une stigmatisation du comportement déviant et à l'exclusion sociale.
3. La variance. Il s'agit d'un simple écart du comportement qui n'est pas socialement répréhensible. Il peut être qualifié "d'excentricité" ou "d'originalité".

L'ensemble du travail réalisé dans cette thèse s'inspire des deux dernières formes de déviance présentées ci-dessus : la **marginalité** et la **variance**. Nous allons donc présenter plus en détails ces deux formes.

Dans le domaine de la psychologie, la **marginalité** des individus est considérée comme une maladie, qu'il est possible de traiter, voire de soigner. Au travers de leur étude portant sur la schizophrénie [Amblard et al., 2011], les auteurs ont montré que, bien que le comportement des personnes atteintes de schizophrénie ne soit pas normal, il n'en est pas moins **logique**. Le problème lié à ce trouble du comportement porte alors sur la valeur associée aux idées des patients et non pas sur la manière dont ils raisonnent. Cela implique qu'il est possible de comprendre et de traiter les marginaux grâce à des méthodes plus subtiles que les méthodes cliniques. La marginalité correspond donc parfaitement au type de déviance que l'on souhaite identifier dans les données de recommandation sociale. Nous voulons identifier des utilisateurs qu'il est possible de traiter, de manière à améliorer leur taux de satisfaction.

L'aspect logique du comportement humain a quant à lui été longuement abordé dans le domaine de la philosophie, notamment dans les travaux sur l'identité de l'individu. Dans le quatrième chapitre de son ouvrage [Wittgenstein and Ogden, 1921], Ludwig Wittgenstein considère que la pensée humaine est nécessairement logique, qu'elle a forcément du sens. Cela signifie que même les pensées d'un individu marginal sont issues d'un raisonnement logique, et qu'**il est donc possible de les comprendre**. En philosophie [Cabiria, 2011], être marginal signifie ne pas être accepté par les principaux groupes sociaux. Le simple fait d'appartenir à un de ces groupes permet à une personne de pouvoir se comparer aux autres membres du groupe, et ainsi de profiter de leur influence. Ce support ainsi apporté par les autres permet à l'individu de développer sa personnalité dans de bonnes conditions, et donc d'ajuster son comportement. La philosophie considère alors qu'une personne marginale qui choisit de se conformer aux règles d'un groupe social pour en devenir un membre change alors de statut et perd sa marginalité.

Cette définition semble indiquer qu'il n'est pas possible de regrouper des marginaux qui se ressemblent, car si c'était possible, ils ne seraient pas marginaux. Il faut donc identifier les utilisateurs qui sont éloignés de tous les groupes d'utilisateurs.

Il existe donc plusieurs catégories de marginaux, tous très différents entre eux. Mais si un individu n'est pas marginal au sens des définitions introduites par la psychologie et la philosophie, il peut tout de même être déviant au sens de la sociologie sous la forme de la **variance**. La

variance correspond à des modifications de certains comportements spécifiques qui n'entraînent pas une dégradation de l'image de l'individu dans la société. On qualifiera alors l'individu déviant d'excentrique ou d'original. A la différence des autres formes de déviance, l'originalité est un point positif dans notre société occidentale actuelle [Weyers, 2012]. Cela implique la volonté de chacun de se démarquer des autres individus de la société, pour être considéré comme original. Certains individus déviants au sens de la variance peuvent alors être des précurseurs [Aleksandrova et al., 2016], qui influencent l'évolution des normes sociales.

Pourquoi certains individus font preuve de variance ?
Ce n'est plus un secret pour personne, la société occidentale est une société de consommation [Benson, 1994]. Dans son ouvrage sur l'apparition de la société de consommation britannique, l'historien John Benson décrit une société dans laquelle il y a un désir, au dessus de tout, pour tout ce qui est moderne, nouveau, excitant et original. Les sociologues se sont alors intéressés à la question de la distinction entre les individus et de l'ordre social ainsi généré [Slater, 1997]. Slater appelle ici à une analyse sociale de la consommation. Par un résumé de Durkheim [Durkheim, 1895], Douglas [Douglas J. D., 1982] ou encore Bourdieu [Bourdieu, 1984a], il montre le rôle des relations sociales et de l'ordre social. Notre utilisation, et pas seulement notre achat, de biens matériels nous intègre dans un ordre social que nous rencontrons constamment dans nos vies quotidiennes. Il va même plus loin en affirmant que les individus se démarquent des autres individus du même niveau social en imitant les individus des niveaux supérieurs. C'est principalement pour cette raison qu'il existe une très importante diversité de comportements chez l'être humain socialisé. La sociologie défend donc l'idée que nous sommes définis par ce que nous consommons. Nous verrons plus tard l'importance de cette vision dans la modélisation des utilisateurs et de leurs usages.

2.1.3.2 Les motivations des travaux sur les déviants et le marginaux

A l'origine, l'objectif des études menées sur les délinquants et les marginaux était la protection de la population. Pour cela, les chercheurs étudiaient les conditions qui conduisent un individu à la délinquance et à la marginalité, faisant preuve de compassion à l'égard des déviants. Ce n'est que bien plus tard que les études sur les causes de la déviance de type variance se sont greffées à ce domaine [Slater, 1997, Bourdieu, 1984a]. L'identification des causes de la déviance permet alors d'une part, de prévenir l'apparition de nouveaux cas de déviance, et d'autre part, de proposer des solutions pour améliorer les conditions de vie des individus déjà en situation de déviance, que ce soit de la délinquance, de la marginalité ou de la variance.

2.1.3.3 L'atypisme en recommandation sociale

L'étude de la déviance, ou plus généralement de l'atypisme, dans les systèmes informatiques s'inscrit dans cette même démarche. L'objectif est d'améliorer le service apporté par le système informatique aux utilisateurs qui ont des besoins différents des autres utilisateurs pour respecter leur droit à la même qualité de service que celui fourni aux autres utilisateurs. Le service apporté à ces utilisateurs peut par exemple prendre la forme de recommandations. La prise en compte du caractère atypique de certains utilisateurs d'un système de recommandation va permettre d'améliorer la qualité des recommandations fournies par ce système.

Reprenons les trois éléments à l'origine d'une déviance, et appliquons les au FC :

— **La norme sociale** : La norme sociale est définie par les autres individus. Dans un système à base de FC, la norme sociale est donc définie par les préférences des autres utilisateurs. Cette norme peut être modélisée de très nombreuses manières, allant de l'utilisation de la moyenne à la mise au point de modèle probabilistes décrivant la probabilité d'apparition d'une préférence.

— **La transgression** : Il existe trois grands types de transgression : la délinquance, la marginalité et la variance. En recommandation sociale, la délinquance correspond aux utilisateurs malveillants. Ces utilisateurs ne sont pas le centre d'intérêt de nos travaux donc nous ne détaillerons pas ce type de transgression. La marginalité correspond à la définition des GSU, telle qu'elle est présentée dans [Claypool et al., 1999], c'est-à-dire qu'un utilisateur marginal lorsqu'il n'appartient à aucune communauté d'utilisateurs. La variance concerne les utilisateurs dont seulement certaines préférences bien ciblées l'éloignent de la norme sociale. Sur des données de préférences, la marginalité et la variance sont très difficile à distinguer. Les utilisateurs faisant preuve de variance n'appartiennent en général pas non plus à une communauté d'utilisateurs, en raison de la variance de leurs préférences. Nous avons vu que tout individu transgresse au moins une norme. En recommandation sociale, cela signifie que nous avons tous au moins une préférence spécifique, même si elle n'est pas connue du système. Nous devons donc mesurer le niveau de transgression des utilisateurs, et les GSU correspondent aux utilisateurs qui transgressent le plus la norme définie par les préférences des autres.

— **La stigmatisation** : Nous avons peu développé ce troisième et dernier point car la stigmatisation des transgressions n'est pas directement le sujet de cette thèse. En recommandation sociale, la stigmatisation des GSU correspond aux recommandations de mauvaise qualité fournies aux GSU. La stigmatisation est donc une conséquence de l'atypisme et, dans ce travail, nous avons choisi de nous intéresser aux causes de l'atypisme en recommandation sociale pour empêcher cette stigmatisation des GSU.

Dans cette section, nous avons défini les notions en lien avec l'atypisme pour mieux comprendre notre problématique. Nous nous sommes inspirés de la définition originale d'un GSU [Claypool et al., 1999] pour proposer la définition suivante d'un GSU qui sera utilisée dans la suite de ce manuscrit :

Grey Sheep Users

Un GSU est un utilisateur qui n'est pas en accord ou en désaccord systématique avec aucune communauté d'utilisateurs, en raison du nombre (relatif ou absolu) de préférences spécifiques qu'il a exprimées.

2.2 Zoom sur le problème des GSU en recommandation sociale

Les travaux portant sur l'identification ou la modélisation des GSU sont peu nombreux dans le domaine de la recommandation. Jusqu'à présent, la littérature avait pour objectif d'améliorer la qualité des recommandations dont bénéficient l'ensemble des utilisateurs. Les GSU étant une minorité (voire paragraphe 2.1.2.4), la littérature considérait qu'il n'était pas intéressant de les traiter en priorité. Cependant, la recommandation atteint aujourd'hui une qualité globale satisfaisante, ce qui encourage la communauté à se pencher sur d'autres aspects, comme les GSU. Dans cette partie, nous analysons les causes de la stigmatisation des GSU dans le FC, puis nous explicitons les motivations derrière l'étude de ce problème pour finir par une synthèse des travaux du domaine de la recommandation qui s'intéressent au problème des GSU.

2.2.1 Pourquoi les GSU reçoivent des recommandations de mauvaise qualité ?

A l'origine, le problème des GSU (voir définition paragraphe 2.1.3.3) s'est révélé suite au constat que des utilisateurs reçoivent de mauvaises recommandations [Claypool et al., 1999]. Il est donc intéressant de comprendre pourquoi un utilisateur aux préférences spécifiques reçoit de mauvaises recommandations. Selon les propriétés fondamentales du FC établies dans [Pennock and Horvitz, 1999], quatre conditions sont à réunir pour qu'un système de FC fournisse de bonnes recommandations (voir paragraphe 2.1.1.6). Voici la liste de ces conditions associées à une analyse des raisons pour lesquelles un GSU pourrait invalider chacune de ces conditions :

— L'universalité : Cette propriété n'est vérifiée que lorsque le système est en mesure de prédire les préférences des utilisateurs. Or, nous savons que dans le cas des utilisateurs en situation de démarrage à froid ou des GSU, le système échouera à prédire précisément leurs préférences. Les GSU remettent donc au cause cette propriété.

— L'unanimité : cette propriété impose l'idée que lorsqu'une ressource est appréciée par une forte majorité d'utilisateurs, alors le système ne pourra pas estimer qu'un utilisateur n'apprécie pas cette ressource. Cette propriété va à l'encontre de la présence de GSU dans les données. Un utilisateur qui n'apprécie pas une ressource très populaire et très appréciée se verra alors mal recommandé par un système respectant cette condition. Nous pouvons donc remettre en question la légitimité de cette propriété. En effet, l'unanimité permet certainement d'assurer un service de qualité pour une large majorité d'utilisateurs, mais dessert fortement les utilisateurs dont les préférences sont spécifiques.

— La stabilité : nous avons explicité la notion de variance issue du domaine de la sociologie. Un individu variant peut, à un instant donné, décider qu'il n'appréciera plus à l'avenir les ressource d'un certain type, qu'il juge désormais dégradantes ou rabaissantes par exemple. Les préférences d'un GSU ne sont pas nécessairement stables dans le temps, sans pour autant être incohérentes. Si le système de recommandation n'intègre pas ce changement de comportement et respecte le critère de la stabilité, un individu variant recevra alors des recommandations pour lesquelles il pourrait avoir une aversion nouvelle. Cette propriété vient donc également limiter les performances d'un système de FC classique pour effectuer des recommandations aux GSU.

— L'invariance de l'échelle : la manière de noter ainsi que le biais utilisateur des GSU peut être à l'origine de leurs mauvaises recommandations. La gestion d'un biais utilisateur plus

précis ou différent ou intégrant plusieurs types de biais, pourrait limiter l'impact de cette propriété sur la qualité des recommandations fournies aux GSU.

On voit donc que parmi les quatre propriétés fondamentales du FC, chacune est susceptible d'interférer avec la qualité des recommandations fournies aux GSU. Il est donc important et nécessaire d'identifier les utilisateurs GSU pour éviter qu'un système de FC classique ne lui fournisse des recommandations de mauvaise qualité.

Une information importante à propos de ce problème est que d'un point de vue statistique, plus le nombre d'utilisateurs d'un système augmente, plus la probabilité de trouver un groupe d'individus semblables augmente [Terveen and Hill, 2001]. On peut donc en déduire que plus le jeu de données comprend d'utilisateurs, plus la proportion de GSU sera faible.

Dans ce travail, nous cherchons à identifier tout type d'utilisateur déviant. La marginalité et la variance peuvent se manifester de plusieurs manières dans les préférences d'un utilisateur.

2.2.2 La fluctuation des performances

L'identification des GSU est souvent motivée par l'explication des fluctuations des performances des SR [Bellogín et al., 2011, Haydar et al., 2012, Griffith et al., 2012]. Les auteurs cherchent à établir un lien entre l'erreur commise sur chaque utilisateur et ses caractéristiques (nombre de notes, nombre de voisins, ...). Par exemple dans le cas du démarrage à froid (utilisateur ayant noté peu de ressources), il est très simple d'observer un fort lien entre le faible nombre de notes d'un utilisateur et la forte erreur de prédiction commise lors de l'étape de recommandation. En effet, avec peu de données, le système ne peut pas effectuer des recommandations de qualité. Dans [Bellogín et al., 2011], les auteurs définissent un indicateur dit de clarté permettant d'identifier les utilisateurs les plus ambigus dans leur notation. Cet indicateur rend compte de la stabilité des notes d'un utilisateur, basé sur la mesure d'entropie. Les auteurs ont montré qu'il existe un lien entre la qualité des recommandations faites à un utilisateur et la stabilité de ses notes. Cette étude leur a également permis d'introduire une nouvelle manière de considérer les GSU. Ces utilisateurs seraient alors ceux dont la stabilité des notes est très faible, et non pas ceux dont les préférences sont différentes de celles des autres. Les auteurs considèrent les GSU comme du bruit (qu'il supprime) et ne leur fournissent pas de recommandations. Or, nous défendons l'idée qu'il existe des utilisateurs aux préférences atypiques dans les SR et que leurs préférences ne sont pas nécessairement instables ou incohérentes. Notre objectif est également d'apporter un service de qualité à ces utilisateurs que d'autres choisissent d'oublier. De plus, l'instabilité des notes données par un utilisateur peut également être expliquée par l'évolution de ses préférences au cours du temps.

Dans [Haydar et al., 2012], les auteurs identifient des clusters d'utilisateurs et tentent d'identifier les caractéristiques communes aux utilisateurs d'un même cluster pour expliquer les fluctuations des performances entre les clusters. Le clustering est la manière la plus utilisée pour identifier les GSU. Cependant, le principe du clustering étant de rassembler les utilisateurs similaires, nous pensons que les GSU ne peuvent pas être rassemblés dans un même cluster car les GSU peuvent ne pas se ressembler. Les auteurs identifient néanmoins des clusters d'utilisateurs pour lesquels l'erreur commise est très élevée et dont les utilisateurs présentent un indice d'*Anormalité* ([Del Prete and Capra, 2010]) très élevé.

Pour conclure sur les fluctuations des performances des SR, dans [Griffith et al., 2012], les auteurs montrent qu'en apprenant un modèle à base de règles faisant le lien entre les informations qu'ils extraient sur un utilisateur et son erreur, il est possible de prédire l'erreur qui sera

commise sur un utilisateur. Les informations utilisées sont :

— Le nombre de notes de l'utilisateur ;
— La note moyenne de l'utilisateur ;
— L'écart-type des notes de l'utilisateur ;
— Le nombre de voisins de l'utilisateur ;
— La similarité moyenne avec les 20 plus proches voisins ;
— La popularité moyenne (très votée ou peu votée ?) des ressources votées par l'utilisateur ;
— L'appréciation générale (note moyenne) des utilisateurs sur les items votés ;
— La moyenne du coefficient de Jaccard entre l'utilisateur et tous les autres utilisateurs.

Les règles ainsi apprises permettent alors d'estimer l'erreur de prédiction que l'on va commettre sur chaque utilisateur. Cela permet donc aux auteurs d'obtenir deux listes d'erreurs (MAE) propres à chaque utilisateur, celle prédite par leur système à base de règles et celle des erreurs réelles observées. Avec une corrélation de 0,76 entre ces deux listes, les auteurs montrent ainsi qu'il est possible d'anticiper l'erreur commise par les SR. Cependant, cette méthode à base d'extraction de règles est opaque et ne sélectionne pas que les utilisateurs présentant des préférences atypiques, puisqu'elle semble désigner l'ensemble des utilisateurs qui souffrent de mauvaises recommandations. Il est donc ensuite très complexe de savoir pourquoi ces mauvaises recommandations apparaissent et comment modifier l'approche de recommandation pour s'adapter à ces différents cas. Cette méthode ne garantit pas une précision d'identification supérieure à 76%, ce qui implique que plusieurs utilisateurs recevant des recommandations de mauvaise qualité ne sont pas identifiés par cette méthode. Nous avons donc choisi de ne pas utiliser ces résultats pour isoler un sous-ensemble des utilisateurs parmi lesquels il serait plus rapide d'identifier les GSU, afin de ne pas risquer la non-identification de certains GSU.

2.2.3 Domaine de la détection des *outliers*

La détection des *outliers* est une branche importante du domaine du *data mining*, étudié depuis de nombreuses années [Hawkins, 1980, Hodge and Austin, 2004, Aggarwal, 2013]. Les applications de la détection d'*outliers* sont très variées, allant de la détection de fraudes aux analyses médicales, en passant par la prédiction de la météo. Un *outlier* est défini comme une observation qui dévie tellement des autres observations que l'on peut suspecter qu'elle a été générée par un mécanisme différent [Hawkins, 1980]. En fonction du domaine d'application, un *outlier* peut également être référencé comme une anomalie [Chandola et al., 2009], du bruit [Eskin, 2000], une erreur [Weimer and Necula, 2005], de la nouveauté [Markou and Singh, 2003], une donnée aberrante [Cousineau and Chartier, 2010], etc. et il est opposé aux observations "normales".

Plusieurs types de scénarios sont possibles pour la détection d'*outliers* : le cas supervisé, non supervisé ou semi-supervisé [Hodge and Austin, 2004]. Le cas supervisé concerne les applications dans lesquelles suffisamment d'*outliers* sont étiquetés pour qu'un algorithme apprenne à les détecter [Aggarwal, 2013]. La cas semi-supervisé correspond aux applications dans lesquelles seule une partie des données sont étiquetées [Gao et al., 2006]. Dans ce travail, nous nous concentrons sur le cas non supervisé de la détection d'*outliers* puisque les GSU ne sont pas étiquetés dans les données de recommandation. Dans ce cas, un paramètre qui représente le nombre ou le pourcentage d'*outliers* que l'on souhaite détecter doit être fixé à la main. Cependant, puisqu'il s'agit par définition d'observations qui sont rares, les valeurs que peut prendre ce paramètre sont limitées.

De nombreuses techniques ont été proposées pour la détection non supervisée d'*outliers* :

les techniques à base de distance, à base d'écart, à base de densité et à base de distribution [Han and Kamber, 2006, Ben-Gal, 2005]. Le choix de la technique dépend alors de l'organisation, la nature, la taille et de la densité des données [Ben-Gal, 2005].

Les techniques à base de distance

Les techniques à base de distance exploitent la distance entre chaque paire d'observations. Deux observations proches sont alors considérées semblables.

Les mesures de distance

Les mesures de distance permettent d'évaluer les similitudes présentes dans les données entre différentes observations. Dans le cas de l'identification des GSU [Claypool et al., 1999], on mesure la distance entre un utilisateur et les autres. Ensuite plusieurs techniques existent pour déterminer si un utilisateur est un GSU ou non. Voici une liste non exhaustive des mesures de distances qui sont utilisées dans l'état de l'art :

- La distance Euclidienne représente la moyenne des écarts brutes entre les vecteurs de notes de deux utilisateurs.
- Le coefficient de corrélation de Pearson est également souvent utilisé de la même manière qu'une mesure de distance. On ne s'intéresse alors qu'aux corrélations positives, une corrélation élevée représentant une distance faible.
- La distance cosinus correspond à (1 - la similarité cosinus) (cf. équation (2.1.1.2)).
- La distance de Bray-Curtis est une mesure de dissimilarité issue du domaine de la biologie. Elle permet de calculer la différence en terme d'abondance des espèces présentes dans deux échantillons.
- La distance de Mahalanobis est une mesure de distance très utilisée dans le domaine de la détection d'*outliers*. Son calcul est néanmoins très coûteux en temps et en mémoire, ce qui rend difficilement utilisable cette mesure sur d'importants jeux de données.

Les k plus proches voisins

Nous avons déjà présentée l'approche des KNN pour le recommandation sociale. Telle qu'elle est décrite dans le domaine du *data mining*, l'approche KNN cherche à identifier les utilisateurs suffisamment éloignés de l'ensemble de leurs voisins et les considère comme des *outliers* [Ramaswamy et al., 2000]. La partie la plus importante consiste donc à définir la mesure de distance entre les observations. Trouver la mesure la plus adaptée pour calculer la distance entre chaque observation est fastidieux, il est même parfois préférable de redéfinir une mesure dédiée. Le principal inconvénient de cette méthode est que sa complexité est liée au nombre d'observations, ainsi qu'à la dimension des données. Un autre inconvénient est qu'il est nécessaire de définir un seuil de distance maximale au delà duquel les observations sont considérées comme des *outliers*. Une variante de cette technique est appelée méthode du $k^{ième}$ plus proche voisin [Wong and Lane, 1981]. Cette technique est une méthode classique du domaine de la détection des *outliers*. Il s'agit de définir k de manière à pouvoir comparer le $k^{ième}$ plus proche voisin de chaque utilisateur. Il suffit ensuite d'identifier les utilisateurs dont le $k^{ième}$ plus proche voisin est trop éloigné, à l'aide d'un seuil par exemple.

Le clustering

Les techniques à base de clustering sont très utilisées dans le domaine de la détection d'*outliers* [Hodge and Austin, 2004, Loureiro et al., 2004, Chawla and Gionis, 2013]. Nous avons déjà évoqué le fonctionnement de quelques approches de clustering (section 2.1.1.3). Les méthodes de

clustering du domaine du *data mining* se sont rapidement exportées aux autres domaines tels que la recommandation sociale pour l'identification des GSU. Nous présentons les méthodes résultantes dans la section suivante.

Les techniques à base de densité

Les techniques à base de densité utilisent la densité autour d'une observation, en comparaison avec la densité autour des observations voisines. Une observation avec une densité significativement plus faible que celle de ses voisins est considérée comme un *outlier*. Cette méthode est très complexe à mettre en œuvre sur des données de recommandation sociales car la matrice de notes est extrêmement vide, ce qui limite fortement les densités observées parmi les observations. Nous avons donc choisi de ne pas exploiter ce type de technique.

Les techniques à base de distribution

Enfin, les techniques à base de distribution posent l'hypothèse que les observations suivent une distribution prédéfinie (qui peut être estimée), le modèle des données. Cette technique utilise en général des distributions standards (par exemple gaussienne). Les *outliers* sont les observations qui appartiennent aux queues de la distribution : il est peu probable qu'ils aient été générés par ce modèle. Le principal inconvénient de cette méthode est que l'identification de la distribution des données peut être longue et coûteuse à calculer.

Il existe plusieurs indicateurs pour évaluer le degré de spécificité des préférences d'un utilisateur en fonction de la loi suivie par les distributions de notes des ressources. Si la distribution est gaussienne, la distance de Mahalanobis peut être utilisée [Aggarwal, 2013], si elle est normale, une analyse en composante principale sera plus adaptée [Jolliffe, 2002], ... Cette manière de détecter des *outliers* a longuement été étudiée [Rousseeuw and Leroy, 2005] et il existe aujourd'hui de nombreuses techniques pour effectuer cette détection. Le principal inconvénient de ces méthodes est à la fois qu'il est complexe d'estimer les lois de distribution, et que les méthodes à appliquer ensuite sont également très lourdes. Les données de recommandation possèdent un très grand nombre de dimensions et l'identification d'un si faible nombre d'utilisateurs en vue d'une amélioration de leur modélisation ne doit pas ralentir le système pour l'intégralité des utilisateurs, il est donc nécessaire de se concentrer sur des méthodes légères, capable de passer à l'échelle.

Les techniques à base de densité de probabilité sont très proches des technique à base de distribution. On trouve par exemple les techniques basées sur la *Vraisemblance* statistique [Akaike, 1998] qui permettent de calculer la probabilité d'apparition d'une observation en fonction des observations connues. Cette technique travaille directement sur les données, sans nécessiter l'identification d'une loi de distribution des variables. La maximisation de la *Vraisemblance* permet alors d'estimer quelle est l'observation la plus probable que l'on pourrait observée, et permet donc de prédire la probabilité d'apparition d'un élément futur. Dans notre cas, il est possible d'appliquer ce principe de manière très simple aux données de recommandation pour estimer la *Vraisemblance* des notes d'un utilisateur. Un utilisateur dont les notes sont peu vraisemblables sera alors considéré comme un GSU.

2.2.4 Identification des GSU

L'Anormalité, la mesure classique

L'*Anormalité* est l'indice le plus utilisé par la littérature pour identifier les utilisateurs présen-

tant des préférences spécifiques [Del Prete and Capra, 2010, Haydar et al., 2012]. Le terme de *deviance* en anglais à été traduit *Anormalité* en français par des travaux antécédents aux miens [Haydar et al., 2012], nous conserverons cette appellation. Cette mesure calcule l'écart entre les notes attribuées par un utilisateur aux ressources qu'il a consultées et la note moyenne attribuée à ces ressources. L'*Anormalité* d'un utilisateur u est notée *Anormalité(u)* et se calcule de la manière suivante :

$$Anormalité(u) = \frac{\sum_{r \in R_u} |n_{u,r} - \overline{n_r}|}{\|R_u\|}$$

où $\|R_u\|$ représente le nombre de ressources notées par l'utilisateur u. Il est ensuite possible d'isoler les utilisateurs dont l'*Anormalité* est très élevée. Cette mesure est peu coûteuse en temps et en mémoire et permet de quantifier la spécificité des préférences d'un utilisateur. Cependant, elle souffre également de nombreuses limites. Par exemple, les ressources sur lesquelles personne n'est d'accord vont injustement augmenter l'*Anormalité* des utilisateurs. En effet, une ressource pour laquelle 50% des utilisateurs ont attribué la note de 5 et 50% des utilisateurs ont attribué la note de 1 aura une moyenne de 3 ce qui implique que 100% des utilisateurs auront un écart de 2 à la moyenne de cette ressource. Il s'agit une fois de plus d'une conséquence de l'utilisation de la moyenne. Une autre limite est liée au fait que cette mesure ne tient pas compte du comportement propre à chaque utilisateur. Par exemple un utilisateur avec un important biais utilisateur risque d'être perçu comme anormal alors qu'il a des préférences similaires à celles des autres.

L'*Anormalité* est notamment utilisée dans le projet DiffeRS, qui est un système de recommandation sociale de musique [Del Prete and Capra, 2010]. Les auteurs ont choisi de séparer les utilisateurs en deux sous-ensembles en exploitant cette mesure : les déviants et les non déviants. Ce choix part du constat que dans le domaine de la musique, il existe toujours une partie des utilisateurs qui sont à contre-courant. Un seuil d'*Anormalité* au dessus duquel un utilisateur est étiqueté "déviant" est fixé. Une nouvelle approche pour la modélisation de ces GSU, que nous détaillerons plus tard, est proposée.

Le clustering, la solution évidente

Dans [Ghazanfar and Prugel-Bennett, 2011], les auteurs proposent d'utiliser un algorithme de clustering *K-Means++* pour identifier les GSU. Ils considèrent un utilisateur comme un GSU s'il n'est proche d'aucun des clusters de manière significative. Ils définissent pour cela un seuil minimal de distance et, à chaque itération, ils séparent les utilisateurs qui ne respectent pas ce seuil minimal de distance au cluster le plus proche. Ensuite, ils séparent les GSU des autres utilisateurs pour pouvoir appliquer une approche de recommandation différente sur leurs données de préférence.

L'article [Ghorbani and Novin, 2016] propose une analyse théorique des capacités de différentes techniques de clustering à regrouper les utilisateurs des systèmes de recommandation. Les auteurs appliquent la technique KNN pour regrouper les utilisateurs et distinguent un cluster de GSU. Aucune expérimentation n'est présentée, ce qui ne nous permet pas d'analyser les performances de cette méthode. De la même manière, une technique à base de clustering conçu pour la détection de plusieurs types d'utilisateurs (GSU, influenceurs, etc.) a récemment été proposée [Srivastava, 2016]. Les auteurs exposent leurs résultats mais ne présentent pas en détail la méthode d'identification des GSU. Nous n'avons donc pas pu comparer nos résultats à ceux de leur méthode.

La barrière magique : une autre forme de déviance ?

Dans [Bellogín et al., 2014], les auteurs considèrent des classes de ressources. Par exemple dans

le cas où les ressources sont des films, une classe peut être le genre "Action", ou "Horreur", etc. Les utilisateurs inconsistants sont ceux qui mettent des notes avec de forts écarts au sein d'une même classe. Ces utilisateurs sont considérés comme du bruit (il sont donc écartés) et les auteurs ont montré que les utilisateurs consistants reçoivent de meilleurs recommandations si l'on ne tient pas compte des utilisateurs inconsistants dans le processus de recommandation. A l'inverse, les utilisateurs inconsistants ne peuvent pas se satisfaire entre eux, ils ont besoin des informations provenant des utilisateurs consistants pour que la qualité de leurs recommandations soit satisfaisante. Cependant, les auteurs se limitent à considérer qu'un utilisateur doit être consistant vis-à-vis de ses propres notes au sein d'une même classe. Ils défendent l'idée qu'un utilisateur inconsistant vis-à-vis de lui-même ne peut pas être correctement modélisé. Les utilisateurs identifiés par les auteurs ne correspondent donc pas à la définition d'origine d'un GSU [Claypool et al., 1999] sur laquelle repose notre définition (voir paragraphe 2.1.3.3).

Dans notre travail, nous choisissons une approche très différente :

— nous n'ajoutons pas d'information sur le contenu des ressources telles que le genre sur les films, etc.
— nous ne considérons pas les GSU comme du bruit, nous considérons qu'ils sont logiques et qu'on peut imaginer des solutions pour les satisfaire.
— l'inconsistance des utilisateurs est mesuré de manière interne à l'utilisateur. Un utilisateur n'est donc inconsistant que vis-à-vis de lui-même, ce qui ne correspond pas à notre définition d'un GSU (cf. paragraphe 2.1.3.3). En effet, nous pensons qu'il n'est possible d'être atypique ou déviant que par rapport aux autres.

2.2.5 Modélisation des GSU

Une fois que les GSU sont identifiés, il faut trouver une solution permettant d'améliorer la qualité des recommandations qui leur sont fournies. L'approche historique de l'état de l'art consiste à renseigner des informations de "contenu" sur les ressources pour pouvoir utiliser un algorithme de filtrage par contenu [Claypool et al., 1999, López-Nores et al., 2012]. Cette solution est efficace mais elle n'est applicable que si les informations de "contenu" peuvent être étiquetées automatiquement. Dans de nombreux cas, cela n'est pas possible, il est donc nécessaire d'apporter une solution plus générique au problème de la modélisation des GSU.

[Griffith et al., 2012] a mentionné qu'il est possible d'apprendre un modèle spécifique pour chaque utilisateur, tandis que [Penn and Zalesne, 2007] exprime clairement que les préférences des GSU sont difficiles à comprendre et à inférer. La prise en compte de leurs préférences, ainsi que la conception d'une approche de recommandation dédiée à la modélisation des GSU, semble être un véritable défi scientifique. En effet, la majeure partie des travaux choisit d'écarter ces utilisateurs de la recommandation plutôt que d'essayer de les modéliser.

L'état de l'art montre que les GSU reçoivent de mauvaises recommandations car la corrélation entre leurs notes et celles des autres utilisateurs pourrait être faible. Bien que les approches de recommandation à base de voisinage reposent sur la recherche de voisins proches ou similaires [Bobadilla et al., 2013], la plupart des travaux dont le but est d'améliorer les recommandations fournies aux GSU exploitent tout de même ces approches.

2.2.5.1 La mise à l'écart des GSU

Dans le cas du système de recommandation de musique diffeRS [Del Prete and Capra, 2010], nous avons vu comment les auteurs ont procédé pour distinguer les utilisateurs déviants des non déviants. Contrairement à [Bellogín et al., 2014] vu précédemment, avec diffeRS, les auteurs

choisissent de ne tenir compte que des utilisateurs déviants lorsque l'utilisateur actif est identifié comme déviant et de ne tenir compte que des utilisateurs non déviants lorsque l'utilisateur actif est non déviant. Les expérimentations montrent que cette solution n'améliore pas la modélisation des utilisateurs déviants, mais améliore celui des utilisateurs non-déviants. Les auteurs proposent de nouvelles idées pour la modélisation des utilisateurs déviants, comme la mise au point d'un nouvel indice de similarité ou d'une nouvelle formule de prédiction. Ces idées ne sont ni détaillées, ni expérimentées.

La plupart des travaux sur le problème des GSU choisissent de mettre les GSU à l'écart de la recommandation pour améliorer les recommandations apportées aux autres utilisateurs. Nous ne sommes pas opposés à l'idée d'écarter les GSU et de les traiter séparément, mais nous défendons la possibilité qu'une méthode dédiée permette de satisfaire les GSU.

2.2.5.2 L'apprentissage d'un modèle dédié

Dans [Ghazanfar and Prugel-Bennett, 2011], les auteurs proposent une méthode de clustering *KMeans++* qui regroupe tous les GSU dans un cluster. Pour proposer des recommandations aux utilisateurs non-GSU, ils utilisent un système de recommandation à base de clustering, qui ne cherche les k plus proches voisins qu'au sein du cluster auquel appartient l'utilisateur actif. Sur le cluster des GSU, les auteurs utilisent un apprentissage par régression SVM (approche modèle) pour proposer des recommandations. Une remarque générale que l'on peut faire sur ce travail est que les techniques utilisées sont complexes en temps de calcul, ne permettent pas une évolution dynamique du système et n'apportent que très peu d'amélioration par rapport à un système de filtrage collaboratif classique. Cette approche a d'ailleurs été abandonnée par l'auteur dans la suite de ses travaux [Ghazanfar and Prügel-Bennett, 2014], avec l'utilisation d'une approche par contenu sur les utilisateurs GSU. L'approche par contenu est la solution originale proposée pour le problème des GSU [Claypool et al., 1999]. Rappelons que, dans la plus grande partie des cas, le renseignement des informations à propos des ressources d'un système est fastidieux et coûte cher à mettre en place. Il ne s'agit donc pas d'une solution adéquate pour une si petite partie des utilisateurs. Dans ce travail, nous défendons l'idée que toute l'information nécessaire pour effectuer une recommandation de qualité aux GSU se trouve dans la matrice de notes et qu'il faut alors trouver de quelle manière extraire cette information.

2.2.5.3 Une nouvelle mesure de similarité

Dans [Bobadilla et al., 2012a], les auteurs ont quant à eux proposé une solution plus générique à la modélisation des GSU. L'identification des GSU n'est pas nécessaire puisqu'ils choisissent de modifier la manière dont sont estimées les recommandations de l'intégralité des utilisateurs. Ils proposent une nouvelle manière de calculer la similarité entre deux utilisateurs basée sur le principe de la singularité d'une note. Une note est dite singulière si elle ne correspond pas à la note majoritaire. Les auteurs définissent une échelle de notes binaire : les notes positives (correspondant aux notes {4,5}) et les notes négatives (correspondant aux notes {1,2,3}). Le principe de la singularité d'une note est ensuite de considérer que deux utilisateurs partageant la même préférence sur une ressource sont plus similaires s'ils font tous les deux partie d'une minorité d'utilisateurs à posséder cette préférence sur la ressource. Les auteurs espèrent ainsi capter et donner plus d'influence aux préférences singulières ou spécifiques des utilisateurs. Pour y parvenir, ils associent à chaque ressource r une singularité positive s_P^r et une singularité négative s_N^r de la manière suivante : plus le ratio de notes positives sur le nombre total de notes d'une ressource est élevé, plus la singularité positive est faible puisqu'il est de moins en moins singulier

de mettre une note positive. Le même principe est appliqué pour la singularité négative suivant l'équation (2.13).

$$s_P^r = 1 - \frac{\#U_P^r}{\#U^r} \qquad et \qquad s_N^r = 1 - \frac{\#U_N^r}{\#U^r}$$
$$avec \quad s_P^r + s_N^r = 1 \tag{2.13}$$

où $\#U_P^r$ est le nombre d'utilisateurs qui ont attribué une note positive à la ressource r et $\#U^r$ est le nombre d'utilisateurs qui ont noté la ressource r. La similarité entre 2 utilisateurs est ensuite calculée en pondérant les notes positives par s_P^r et les notes négatives par s_N^r. Le reste du processus de recommandation est identique à celui d'un filtrage collaboratif classique. L'auteur montre dans ce papier que moins de voisins sont nécessaires pour calculer une recommandation de qualité, et que la couverture des ressources est accrue. Il observe également une légère diminution de l'erreur, non significative. La première remarque que l'on peut apporter à ce papier est qu'il ne semble pas apporter une solution directe au problème des GSU. En effet, cette mesure de similarité semble mieux rendre compte des similarités entre utilisateurs. Cependant, elle ne permet pas pour autant de trouver de bons voisins aux GSU. Intuitivement, cette mesure semble plutôt sélectionner plus finement les voisins des utilisateurs non atypiques qui ont quelques préférences singulières, et non pas s'adapter aux utilisateurs qui ont une majorité de préférences singulières. Cette démarche est néanmoins efficace pour améliorer la personnalisation des systèmes de recommandation et peut être une bonne approche pour limiter l'impact du démarrage à froid (moins de voisins sont nécessaires).

De mon point de vue, le problème des GSU n'a pas encore de solution performante, aussi bien pour l'identification de ces utilisateurs que pour leur modélisation. Nous avons vu qu'il n'y a pas une unique définition du concept de GSU dans la littérature, ce qui entraine différentes mesures quant à leur identification. Dans ce travail, nous souhaitons identifier des utilisateurs qui ont besoin d'une amélioration du service qui leur est proposé. Pour atteindre cet objectif, nous proposons des mesures d'identification et des approches de modélisation correspondant au regard nouveau que nous avons apporté sur ce sujet.

Chapitre 3

Proposition de mesures pour l'identification des Grey Sheep Users

Sommaire

La proposition de nouvelles mesures dédiées à l'identification des GSU dans les systèmes de recommandation (SR) à base de filtrage collaboratif (FC) est l'objectif de ce chapitre. Afin de garantir que les mesures que nous proposons puissent être génériques à tout système de recommandation à base de filtrage collaboratif, nous avons choisi de n'exploiter que les données de préférence, celles présentes dans la matrice de notes N (qui représente les préférences des utilisateurs) pour identifier les GSU. De plus, nous nous imposons la contrainte de mettre au point des mesures dont la complexité est limitée et qui permettent la dynamicité du système.

Dans ce chapitre, nous présentons les mesures que nous avons conçues pour cette tâche d'identification, puis nous décrivons les expérimentations menées et analysons les résultats obtenus avec chacune des mesures.

3.1 Les mesures d'identification

Le FC repose sur l'hypothèse que les préférences des utilisateurs sont cohérents entre eux. C'est pourquoi il exploite les préférences des autres utilisateurs pour inférer la préférence de l'utilisateur actif sur une ressource qu'il n'a pas encore évaluée. Le FC requiert donc qu'au moins une communauté d'utilisateurs possède des préférences similaires à celles de l'utilisateur actif pour pouvoir lui proposer des recommandations de bonne qualité. Si une telle communauté n'existe pas, l'utilisateur actif est alors considéré comme un GSU.

Un GSU (voir définition paragraphe 2.1.3.3) possède donc plusieurs préférences, dites spécifiques, qui sont éloignées de la norme des préférences du système (norme inférée des préférences des utilisateurs). Ces préférences spécifiques empêchent les approches classiques de FC de fournir aux GSU des recommandations de bonne qualité. L'identification de GSU ne peut se faire que si l'on dispose d'un modèle décrivant la norme [Robert et al., 1997] des préférences, permettant de représenter les utilisateurs dits normaux, ou non-GSU. Grâce à ce modèle, il est possible d'identifier les utilisateurs qui dévient le plus de cette norme, c'est-à-dire les GSU. Chacune des mesures que nous proposons dans ce chapitre repose sur une définition propre de la norme des préférences, et permet ainsi de distinguer des utilisateurs différents.

3.1.1 Voisinage et popularité

Les premières mesures que nous proposons ont pour but de vérifier l'hypothèse proposée par [Bellogín et al., 2011] selon laquelle les utilisateurs qui reçoivent de mauvaises recommandations sont ceux qui n'ont pas d'utilisateurs suffisamment similaires dans le système. La similarité peut être mesurée de très nombreuses manières au travers par exemple de la distance Cosinus, le coefficient de corrélation de Pearson, la distance de Bray-Curtis, la Singularité, etc. (cf. chapitre 2). La mesure la plus populaire en FC pour mesurer la proximité entre deux utilisateurs est le coefficient de corrélation de Pearson (cf. équation 2.1.1.2). Cette mesure permet à la fois d'identifier les utilisateurs similaires mais également les utilisateurs dont les préférences sont les plus opposées dans le système, contrairement aux mesures à base de distance qui ne mesurent qu'un écart moyen et ne s'intéressent pas précisément à l'opposition systématique des préférences.

— Tout d'abord, nous étudions le lien entre la présence d'utilisateurs très corrélées ou anti-corrélées à l'utilisateur actif et la qualité des recommandations qu'il reçoit. Nous avons considéré les valeurs maximales des corrélations négatives et positives d'un utilisateur (**Corr *T* MaxPos / Corr *T* MaxNeg**) pour connaître le lien entre les utilisateurs possédant des voisins très dissimilaires/similaires et l'erreur de recommandation observée sur

ces utilisateurs.

— Ensuite, pour compléter cette mesure, nous observons le lien entre la similarité/dissimilarité moyenne d'un utilisateur (**CorrMoyPos / CorrMoyNeg**) avec les autres utilisateurs pour voir si cela peut suffire à expliquer la qualité des recommandations.

— Enfin, nous analysons la qualité des recommandations fournies aux utilisateurs dont la taille du voisinage est faible. Il s'agit donc de mesurer le nombre d'utilisateurs voisins qu'il est possible d'associer à l'utilisateur actif, en fonction de ses préférences (**Popularité**). En l'occurrence, un nombre minimal de cinq co-votes [Schickel-Zuber and Faltings, 2006] est nécessaire pour pouvoir estimer la similarité entre deux utilisateurs, c'est-à-dire considérer ces utilisateurs comme des voisins. Si avec peu de notes, un utilisateur a beaucoup de voisins, alors les ressources notées sont des ressources populaires. Un utilisateur avec un indice de popularité faible est donc un utilisateur qui ne consulte pas en moyenne les mêmes ressources que les autres, ce qui peut révéler un comportement déviant. La popularité de l'utilisateur u est alors notée :

$$Popularité(u) = \frac{|Voisins(u)|}{|R_u|} \tag{3.1}$$

où $|R_u|$ représente le nombre de ressources notées par l'utilisateur u et $|Voisins(u)|$ représente le nombre de voisins de l'utilisateur u.

Notre objectif est de proposer des mesures génériques qui puissent s'adapter à toutes les approches de recommandation sociale. Ces différentes mesures sont tout spécialement adaptées pour les systèmes à base de mémoire comme l'approche des k plus proches voisins puisqu'elles reposent sur l'exploitation de la similarité entre les utilisateurs qui est centrale à cette approche. Elles peuvent néanmoins être utilisées dans d'autres types de systèmes, comme les systèmes à base de modèle, mais elles représenteraient une complexité supplémentaire puisque ces approches ne reposent pas sur leur calcul.

3.1.2 Extensions de l'*Anormalité*

Les deux mesures suivantes que nous proposons sont des extensions de l'*Anormalité* de l'état de l'art qui a montré son efficacité pour l'identification des GSU. Nous proposons des extensions permettant de tenir compte des biais que nous avons évoqués au chapitre précédent (cf. paragraphe 2.2.4).

3.1.2.1 AnormalitéCR

L'*AnormalitéCR* (anormalité exploitant la controverse sur les ressources) est basée sur l'hypothèse que l'écart à la moyenne des notes (cf. équation 2.1.3.1) n'a pas la même signification selon la ressource. Prenons l'exemple d'une ressource sur laquelle les préférences sont divisés. Cette dernière fera augmenter l'*Anormalité* des utilisateurs qui l'ont consultée et évaluée sans que cela soit justifié, puisque chaque note sera écartée de la moyenne des notes sur la ressource. Pour réduire l'impact des ressources controversées sur cette mesure, nous choisissons d'introduire

un indice de controverse, basé sur l'écart-type des notes d'une ressource. Cet indice de controverse permet de pondérer l'impact d'une note sur l'*Anormalité* d'un utilisateur, de manière à donner plus d'importance aux écarts constatés sur les ressources peu controversées, c'est-à-dire consensuelles.

De plus, nous défendons l'idée que si la préférence d'un utilisateur est très écartée de celles des autres utilisateurs, cet écart doit être d'avantage pris en considération. En effet, les préférences très écartées de la moyenne sont des préférences spécifiques et nous voulons identifier les utilisateurs possédant plusieurs préférences spécifiques. Pour cela, nous avons renforcé l'impact des forts écarts, et diminué l'impact des très faibles écarts que l'on peut associer à la variabilité naturelle des notes des utilisateurs. L'*AnormalitéCR* d'un utilisateur u est évaluée de la manière suivante :

$$\text{Anormalité}_{CR}(u) = \frac{\sum_{r \in R_u} [(n_{u,r} - \overline{n_r}) * sig(r)]^2}{\|R_u\|}$$

où *sig(r)* représente l'indice basé sur l'écart-type normalisé des notes de la ressource r calculé de la manière suivante :

$$sig(r) = 1 - \frac{\sigma_r - \sigma_{min}}{\sigma_{max} - \sigma_{min}}$$

avec σ_r l'écart-type des notes de la ressource r, σ_{min} l'écart-type minimum possible de notes dans l'ensemble des ressources et σ_{max} l'écart-type maximum possible de notes dans l'ensemble des ressources. Le calcul de l'*AnormalitéCR* est peu complexe en temps et en mémoire et permet une évolution dynamique du SR.

La figure 3.1 illustre l'ordre des différentes étapes du calcul de l'*AnormalitéCR*. Les préférences moyennes sur chaque ressource sont représentées par le vecteur $\overline{u}$, représentant l'utilisateur moyen du système. Le vecteur $sig(r)$ est quant à lui composé de l'indice de controverse de chaque ressource. Le vecteur $sig(r)$ est donc utilisé pour pondérer les écarts entre chaque utilisateur et l'utilisateur moyen $\overline{u}$.

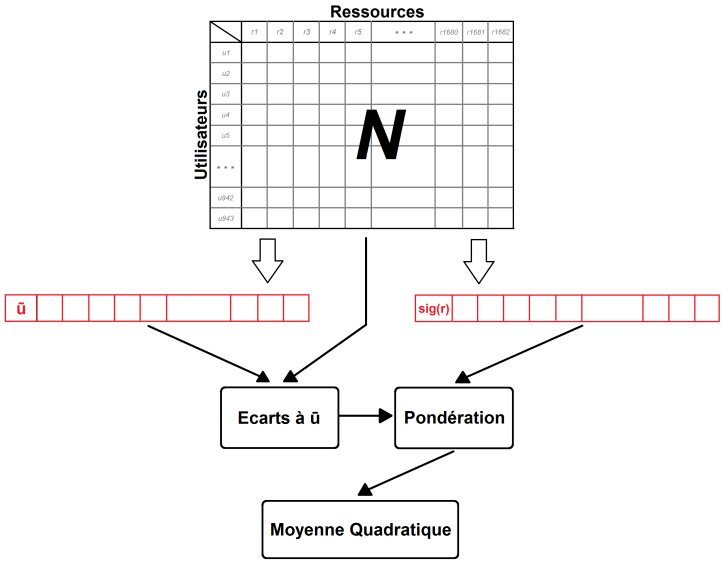


FIGURE 3.1 – Calcul de l'*AnormalitéCR*

3.1.2.2 AnormalitéCRU

L'*AnormalitéCR* (cf. équation (3.1.2.1) ne tient pas compte de l'échelle de notation des utilisateurs. Par exemple, un utilisateur sévère qui met souvent des notes basses risque d'être identifié comme GSU alors que seule son échelle de notation est différente. Nous nous sommes inspiré du coefficient de corrélation de Pearson pour minimiser l'impact du biais de l'utilisateur sur l'*AnormalitéCR*. Nous proposons la mesure d'*AnormalitéCRU* qui est une extension de l'*AnormalitéCR* qui considère chaque note de la matrice N par rapport à la moyenne de l'utilisateur qui l'a exprimée. Chacune des notes des ressources est centrée par rapport aux moyennes des utilisateurs qui les ont votées. De cette manière, chaque note contient plus d'information, ce qui permet de savoir plus finement si la ressource a été plutôt appréciée ou non par l'utilisateur. L'*AnormalitéCRU* d'un utilisateur u est alors notée $Anormalité_{CRU}(u)$ et se calcule de la manière suivante :

$$Anormalité_{CRU}(u) = \frac{\sum_{r \in R_u} [1 + (\left| n_{u,r} - \overline{n_u} - \overline{nC_r} \right|) * sigC(r)]^2}{\|R_u\|}$$

où $\overline{nC_r}$ représente la moyenne des notes centrées des utilisateurs sur la ressource r, *sigC(r)* représente un indice de controverse "centré" calculé à partir de l'écart-type des notes centrées (par rapport aux utilisateurs) de la ressource r. *sigC(r)* est calculé de la même manière que *sig(r)* à la différence que les notes utilisées pour calculer l'écart type sont des notes centrées par rapport aux moyennes des utilisateurs.

Les deux mesures d'anormalité que j'ai proposées permet d'étudier séparément les impacts du biais de notation des utilisateurs et de la controverse des ressources dans le processus d'identification des GSU.

D'autres extensions de l'*Anormalité* peuvent être imaginées, avec plusieurs manières de mesurer la controverse d'une ressource, mais également plusieurs manières d'incorporer cet indice dans la formule globale d'*Anormalité*.

3.1.3 Méthode à base de clustering, ClustGSU

Une seconde méthode d'identification des GSU de l'état de l'art qui a retenu notre attention repose sur le clustering, c'est la technique KMeans++ [Ghazanfar and Prugel-Bennett, 2011]. Il s'agit d'un algorithme KMeans dans lequel les centres des clusters sont initialisés à l'aide d'une heuristique permettant de déterminer quels sont les meilleurs utilisateurs pour représenter les centres initiaux des clusters. Cette amélioration de l'algorithme original du KMeans permet d'accélérer la vitesse de convergence de l'algorithme et de s'assurer d'une meilleure répartition des utilisateurs dans les clusters (afin d'éviter le cas dans lequel la majorité des utilisateurs se retrouvent dans le même cluster). Cependant, cette amélioration semble présenter une défaut de taille relativement à notre problème. Comme nous l'avons constaté dans l'état de l'art (cf. chapitre 2), l'initialisation des centroïdes des clusters aux utilisateurs les plus éloignés dans le jeu de données peut amener l'algorithme à identifier des GSU comme points d'initialisation et donc perturber leur identification. Rappelons que l'idée de [Ghazanfar and Prugel-Bennett, 2011] est de considérer comme GSU les utilisateurs qui ne sont suffisamment proches d'aucun des centroides à l'aide d'un seuil de distance minimale aux centres (cf. paragraphe 2.2.3). Ces utilisateurs sont ensuite isolés dans un cluster supplémentaire.

Nous avons donc pris en compte cette limitation pour proposer une autre technique à base de clustering, mieux adaptée à notre problème. Nous proposons d'utiliser une solution de KMeans

classique dans laquelle nous initialisons les centres des clusters avec des utilisateurs tirés au hasard, et ce n'est qu'après la convergence de l'algorithme KMeans que nous isolons les utilisateurs les plus éloignés de leur centre de cluster, à l'aide d'un seuil Δ_{max}. Il s'agit donc d'une version simplifiée de l'algorithme de clustering de [Ghazanfar and Prugel-Bennett, 2011], mais cette proposition semble mieux adaptée à notre problème et permettra d'avoir un point de comparaison avec les autres mesures puisque le clustering est souvent utilisé pour les tâches d'identification d'anomalies en général ou de GSU dans le cas de la recommandation.

Un exemple d'exécution de la méthode *ClustGSU* est présenté sur les figures 3.2 et 3.3.

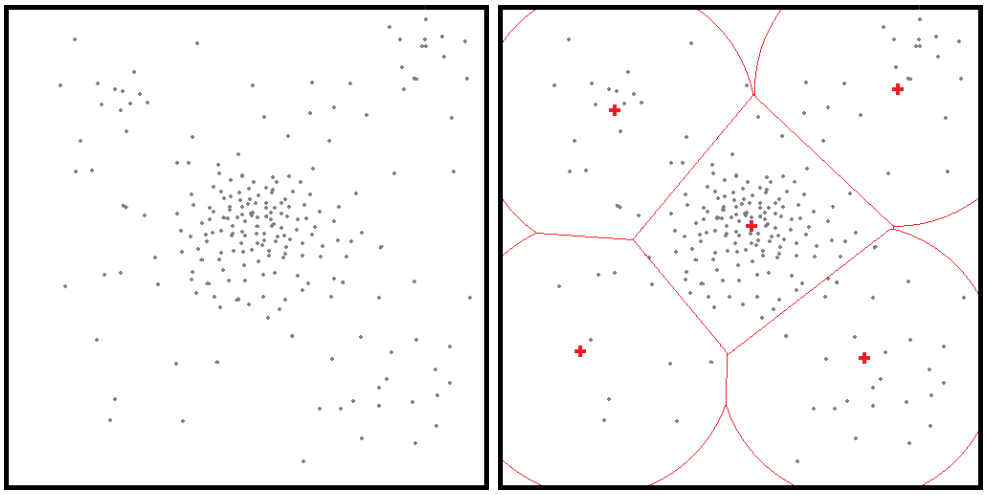


FIGURE 3.2 – Exécution d'un KMeans classique

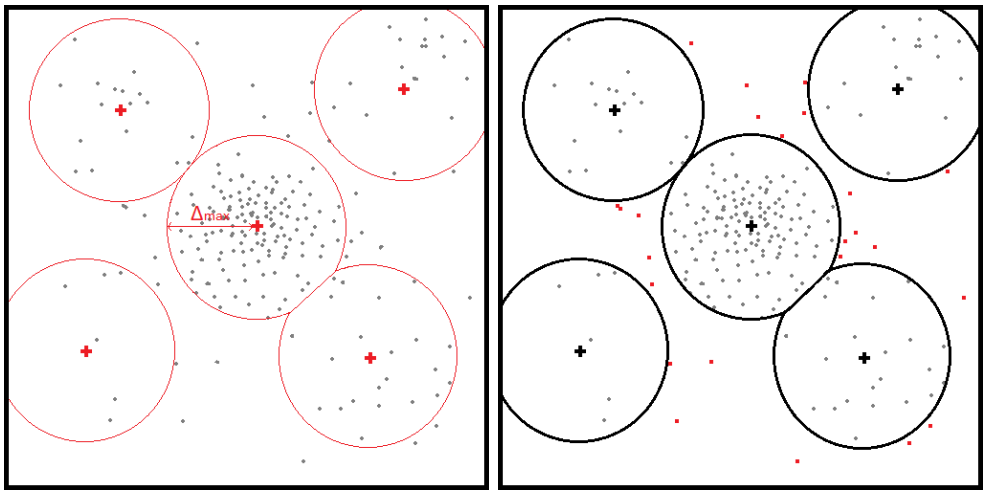


FIGURE 3.3 – Définition d'une distance maximale au centre des clusters et isolement des utilisateurs les plus éloignés

Plusieurs mesures de distances peuvent être utilisées pour l'exécution de l'algorithme *ClustGSU* parmi lesquelles figurent les normes L_1 et L_2, la distance cosinus, le coefficient de corrélation de Pearson, etc. Le choix de la mesure dépend des données sur lesquelles l'algorithme est appliqué.

3.1.4 Mesures à base de distribution

Les mesures précédentes reposent sur des écarts ou des distances, soit avec l'utilisateur moyen du système, soit avec les centroïdes des clusters. La moyenne est donc au cœur des mesures précédentes, et je pense que la norme des préférences des utilisateurs peut être représentée de manière plus fine qu'au travers de l'utilisation de la moyenne des notes sur chaque ressource.

Les mesures à base de distribution peuvent apporter plus de finesse à l'identification des GSU en considérant directement la distribution des notes sur une ressource plutôt qu'uniquement la moyenne de ces notes. Les mesures précédentes exploitent des indices permettant de donner un aperçu de la distribution des notes (moyenne, écart-type, etc.), mais dans le cas des données de recommandation, le faible taux de remplissage de la matrice de préférences permet de travailler directement sur les distributions des notes des ressources, sans en inférer la loi de distribution. En effet, bien qu'il soit possible d'associer chaque distribution de notes des ressources à une loi de distribution paramétrique connue (Zipf, Poisson, etc.), cette opération est coûteuse en temps de calcul et ne garantit pas de meilleurs résultats. Je propose de ne pas inférer les lois de distributions associées aux données observées, mais de considérer une estimation de la densité non-paramétrique pour chaque ressource. Cette solution est souvent adoptée dans le domaine de la recherche d'information, et plus précisément pour la modélisation statistique du langage [Croft, 2003]. La principale différence dans notre travail est que l'estimation de la densité est effectuée sur chaque ressource séparément, puisque c'est la distribution de chaque ressource qui permet d'évaluer la spécificité d'une préférence, tandis qu'une seule estimation de densité pour l'ensemble des ressources est calculée dans le domaine de la recherche d'information.

La *Vraisemblance* est une mesure statistique permettant d'estimer s'il est "vraisemblable" qu'une observation appartienne ou non à une distribution d'observations. Cette mesure n'a, à notre connaissance, jamais été utilisée pour identifier les GSU. Elle est néanmoins adaptée puisque les distributions de notes de chaque utilisateur et de chaque ressource sont calculables.

Puisque les données de préférences sont discrètes, estimer la fonction de probabilité de densité revient à estimer un petit ensemble de $|V|$ probabilités, une pour chaque valeur possible de note ($v \in V$) pour chaque ressource r. Ces probabilités, calculées à partir de la fonction d'estimation $f(r, v)$ (voir équation (3.2)), forment le modèle θ.

$$f(r, v) = \frac{\sum_u \delta(u, r, v)}{\sum_v \sum_u \delta(u, r, v)}, \tag{3.2}$$

où $\delta(u, r, v) = 1$ lorsque $n_{u,r} = v$ et $\delta(u, r, v) = 0$ dans les autres cas. Puisque l'élément $\sum_u \delta(u, r, v)$ est facilement calculable lorsque les données sont chargées par l'approche de recommandation, l'estimation de θ n'est pas complexe, elle s'effectue en $O(m)$, avec m le nombre de ressources dans le jeu de données.

Ensuite, pour déterminer si un utilisateur u est un GSU, nous nous basons sur le modèle θ qui représente les distributions des préférences de la population entière sur chaque ressource, suivant $f(r, v)$. Nous choisissons d'exploiter la *Vraisemblance* [Edwards, 1972], qui évalue la probabilité qu'une donnée appartienne à un modèle. Dans le domaine de la recherche d'information, cette formule est traditionnellement utilisée pour déterminer si un document peut vraisemblablement avoir été généré par un modèle donné. Dans les systèmes de recommandation à base de FC, la $Vraisemblance$ pourrait être utilisée pour évaluer à quel point les préférences d'un utilisateur (les notes) peuvent avoir été générées par θ, le modèle des préférences de la population d'utilisateurs. Plus concrètement, la $Vraisemblance$ évalue les probabilités conditionnelles d'apparition de chaque préférence d'un utilisateur en fonction du modèle décrit par θ.

En raison du problème de manque de données que nous avons déjà évoqué, les valeurs de la mesure de $Vraisemblance$ de deux utilisateurs distincts peuvent ne pas avoir été calculés sur le même nombre de ressources, ce qui ne permet pas de les comparer. Il est donc nécessaire de modifier l'équation originale de la $Vraisemblance$, pour tenir compte de ce nombre de termes. Nous proposons d'exploiter la racine $||R_u||^{ième}$ (voir équation (3.3)).

$$Vraisemblance(u,\theta) = \sqrt[||R_u||]{\prod_{r \in R_u} f(r, n_{u,r})} \tag{3.3}$$

Notons que certains utilisateurs peuvent avoir une *Vraisemblance* nulle puisqu'il est possible qu'un terme de la multiplication soit 0. Pour éviter ce cas, dans l'équation (3.2) nous utilisons une technique de lissage des données pour attribuer une faible probabilité d'apparition (non nulle) aux notes qui n'ont jamais été exprimées sur les ressources. Cette technique est inspirée du domaine de la modélisation du langage [Croft, 2003]. Nous utilisons le lissage proposé par Kneser et Ney [Kneser and Ney, 1995], comme suggéré dans [Chen and Goodman, 1998]. La complexité de cette mesure est alors en $O(nm)$, avec n le nombre d'utilisateurs.

La principale limitation de f est qu'elle ne tient pas compte du biais des utilisateurs. En effet, les notes des utilisateurs sévères ne devraient pas être considérées de la même manière que celles des utilisateurs plus tolérants. Pour tenir compte de ce biais, nous proposons ici de centrer les notes de chaque utilisateur par rapport à sa moyenne. L'équation résultante, la *Vraisemblance* sans biais utilisateur ($VraisemblanceSBU$), est présentée dans l'équation (3.4).

$$VraisemblanceSBU(u,\theta) = \sqrt[||R_u||]{\prod_{r \in R_u} f'(r, (n_{u,r} - \overline{n_u}))}, \tag{3.4}$$

où $(n_{u,r} - \overline{n_u})$ représente une note sans le biais de l'utilisateur. La complexité de cette mesure est comparable à celle de la $Vraisemblance$, $O(nm)$. Puisque les notes sans biais utilisateur sont continues, la fonction d'estimation $f(r, v)$, qui a été conçue pour des variables discrètes, ne peut plus être utilisée. Nous proposons une fonction d'estimation de densité pour histogrammes [Silverman, 1986], $f'(r, v)$. Avec l'objectif de faciliter la comparaison des résultats, nous choisissons de définir f' pour autant d'intervalles de notation que le nombre de valeurs discrètes considérées par f, chaque intervalle ayant la même taille. Le principal inconvénient de cette fonction d'estimation est que le choix des positions du premier intervalle et du dernier peut impacter les résultats. Nous avons choisi d'utiliser les plus petite et grande valeurs de note sans biais utilisateur observées dans les données comme bornes inférieure et supérieure des intervalles considérés par f'.

Les figures 3.4 et 3.5 représentent deux manières de délimiter les intervalles de f', avec $|V| = 5$. La figure 3.4 utilise les minimum et maximum théoriques pour délimiter les cinq intervalles dans la distribution de notes, tandis que la figure 3.5 exploite les plus petite et grande valeurs de note sans biais utilisateur observées dans les données. Nous avons choisi la solution de la figure 3.5 car elle permet de répartir plus équitablement les notes dans les différents intervalles.

Une seconde limitation des équations (3.3) et (3.4) est qu'elles ne traitent pas le problème connu de l'imprécision des notes [Amatriain et al., 2009]. Les équations (3.3) et (3.4) n'exploitent que la probabilité associée à la note exacte ou à la barre de l'histogramme (représentant un intervalle) correspondant à la note courante. Quand une note sans biais utilisateur se situe à la limite d'un intervalle, elle pourrait en réalité appartenir à l'intervalle voisin, en raison de l'imprécision de la notation. Puisque la fonction d'estimation f' ne tient pas compte du lien entre deux intervalles différents, la probabilité obtenue avec f' peut ne pas refléter correctement

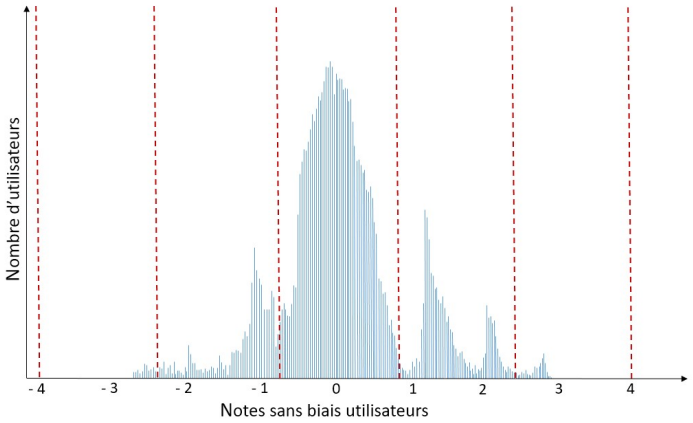


FIGURE 3.4 – Découpe des intervalles en fonction des valeurs minimale et maximale théoriques

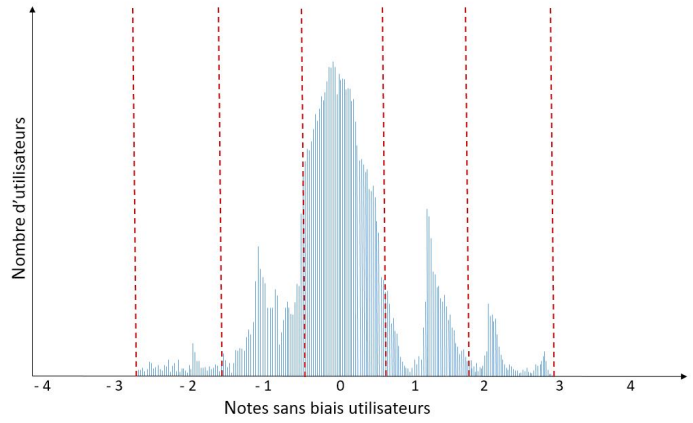


FIGURE 3.5 – Découpe des intervalles en fonction des valeurs minimale et maximale observées

la réalité. Nous proposons de former, pour chaque note sans biais utilisateur, un intervalle de confiance centré sur la note de l'utilisateur, permettant à la fonction d'estimation de tenir compte de l'imprécision des notes. La fonction d'estimation qui en résulte est alors appelée fonction d'estimation naïve [Silverman, 1986].

Soit h l'estimation de l'imprécision de la notation des utilisateurs. Dans ce contexte, h est un paramètre de tolérance permettant de délimiter l'intervalle de confiance représentant chaque note. Les intervalles de confiance résultant ne sont ainsi pas déterminés en amont du calcul de la fonction d'estimation, seule leur amplitude (h) est déterminée en amont. Nous pensons que l'ajout de cette tolérance à chaque note permettra de mieux modéliser les préférences d'un utilisateur que les intervalles prédéfinis utilisés pour les équations (3.3) et (3.4). Le résultat est une mesure de $Vraisemblance$, avec des Intervalles Dynamiques ($VraisemblanceID$), qui est définie dans l'équation (3.5).

$$VraisemblanceID(u) = \sqrt[\|R_u\|]{\prod_{r \in R_u} f''(n_{u,r} - \overline{n_u}, h)}, \tag{3.5}$$

où f'' est une fonction d'estimation naïve qui se base sur les notes centrées des utilisateurs. $f''(n_{u,r} - \overline{n_u}, h)$ estime la probabilité qu'une note sans biais utilisateur $(n_{u,r} - \overline{n_u})$ soit dans

l'intervalle $[n_{u,r} - \overline{n_u} - h; n_{u,r} - \overline{n_u} + h]$. Une fois de plus, nous calculons la distribution des préférences des utilisateurs sur chaque ressource en amont du processus d'identification des GSU. La complexité de la *VraisemblanceID* reste donc en $O(nm)$. Notons néanmoins que l'utilisation d'intervalles dynamiques est due à la mise en place d'une méthode d'estimation de la densité non-paramétrique. Cela n'aurait pas été nécessaire de se servir d'intervalles dynamiques si nous avions choisi d'utiliser des lois de distribution paramétriques.

3.2 Expérimentations

3.2.1 Les données

Au cours de cette thèse, nous avons utilisé plusieurs jeux de données proposés par la société GroupLens [22]. Ces jeux de données ont été collectés sur le site MovieLens [23] et sont les jeux de données de référence de l'état de l'art. Ce site propose d'effectuer des recommandations cinématographiques à ses utilisateurs sur la base d'un algorithme de filtrage collaboratif. Il s'agit d'un service gratuit en ligne qui existe depuis plus de 10 ans, ce qui explique l'impressionnante quantité d'informations présente dans ces jeux de données. Nous avons utilisé deux jeux de données Movielens, MovieLens100K et MovieLens20M.

Le plus petit des deux, MovieLens100K, contient 100 000 notes exprimées par 943 utilisateurs sur 1 682 films (les ressources). Chaque note est comprise entre 1 et 5 et prend toujours une valeur entière. Un utilisateur a voté en moyenne 95 films parmi les 1 682, ce qui signifie que la matrice de notes N est constituée d'environ 94% de données manquantes. Quelques informations complémentaires sur la répartition des notes dans ce jeu de données sont présentées sur la figure 3.6.

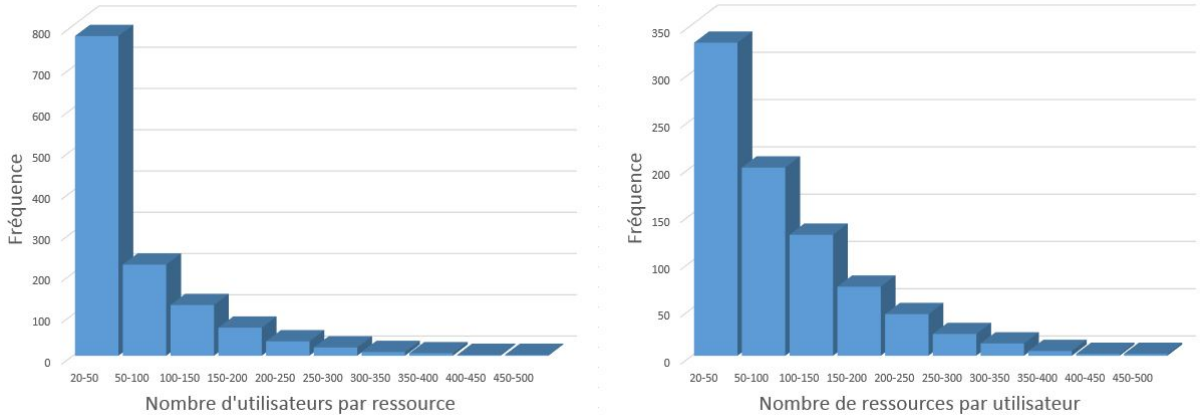


FIGURE 3.6 – Distributions des préférences par utilisateur et par ressource sur MovieLens100K

La figure 3.6 montre que même si le nombre moyen de notes par utilisateur est de 95, près de la moitié des utilisateurs a noté entre 20 et 50 films. Les ressources sont s'avantage concernées par ce problème avec plus de 60% des ressources évaluées par moins de 50 utilisateurs. Notons, à titre indicatif, que peu d'utilisateurs ont noté plus de 200 ressources et que peu de ressources ont été notées par plus de 150 utilisateurs.

22. www.grouplens.org
23. www.movielens.org

Lorsque l'on souhaite évaluer un système de recommandation, on divise le jeu de données initial en deux sous parties, le jeu de données d'apprentissage et le jeu de données de test. Plus précisément, on retire une partie des notes déjà observées du jeu de données initial (elles constituent le jeu de données de test) et l'objectif est de les retrouver en exploitant les notes qui ont permis d'apprendre les préférences des utilisateurs. Nous avons choisi d'exploiter 80% des données pour la phase d'apprentissage et 20% des données pour le test.

Dans [Schickel-Zuber and Faltings, 2006], l'auteur montre qu'il est nécessaire de posséder au minimum 20 notes sur un utilisateur dans les données d'apprentissage pour qu'un filtrage collaboratif classique ait des performances acceptables sur MovieLens100K. Or, dans la base MovieLens100K, 122 utilisateurs possèdent moins de 25 notes, c'est-à-dire que lorsque l'on sépare la base de départ en 80/20, 122 utilisateurs ont moins de 20 notes dans le jeu de données d'apprentissage. Notre travail ne portant pas sur le problème du démarrage à froid, nous avons choisi de supprimer ces utilisateurs pour qu'ils ne perturbent pas les expérimentations. Nous avons donc travaillé sur une version réduite de MovieLens100K avec 821 utilisateurs et 1 649 films restant.

Le second jeu de données de MovieLens que nous avons utilisé est MovieLens20M. Ce jeu de données contient 20 millions de notes exprimées par 138 000 utilisateurs sur 27 000 films. La matrice de notes est 6 fois moins dense que celle de MovieLens100K (1% de taux de remplissage contre 6,3%). Le principal intérêt de ce jeu de données est d'apporter une validation de nos mesures sur un plus grand nombre d'utilisateurs. En effet, le problème des GSU dans les systèmes de recommandation sociale est en lien direct avec le nombre d'utilisateurs [Claypool et al., 1999]. Dans un système possédant des millions d'utilisateurs, il est bien moins probable que l'on puisse être différents des autres utilisateurs et cela limite fortement le nombre de GSU.

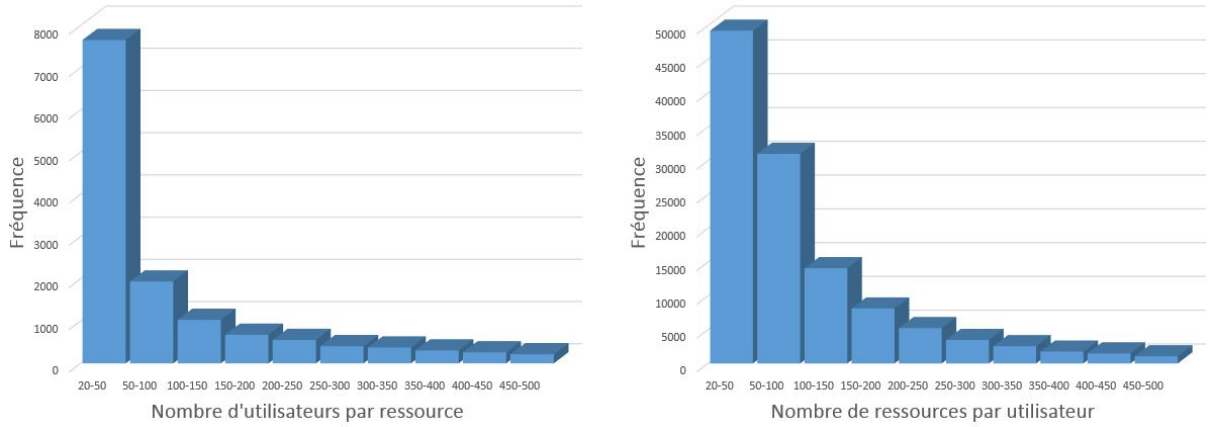


FIGURE 3.7 – Distributions des préférences par utilisateur et par ressource sur MovieLens20M

La figure 3.7 montre la répartition des nombres de notes par ressource et par utilisateur. Cette répartition est très similaire à celle du jeu de donnée MovieLens100K, bien que ce second jeu de données soit moins dense. Il y a donc également peu d'utilisateurs ayant évalué plus de 200 ressources et peu de ressource ayant été évaluées par plus de 150 utilisateurs.

Nous avons également supprimé les utilisateurs possédant moins de 20 notes dans les données d'apprentissage car ils relèvent du problème du démarrage à froid. Le jeu de données résultant est composé d'environ 123 000 utilisateurs et 17 000 ressources.

Pour chacun de ces deux jeux de données, nous avons séparé les données en 5 sous-ensemble

	Approche KNN		Approche ALS			
	k	Similarité	k	α	λ	Régularisation
MovieLens100K	20	Pearson	10	0,01	0,1	L_2
MovieLens 20M	//	//	20	0,001	0,02	L_2

TABLE 3.1 – Paramètres des approches de recommandation

pour effectuer une validation croisée de nos résultats avec une répartition 80%/20% des données entre le base d'apprentissage et de test respectivement. Nous avons donc 5 jeux de données d'apprentissage associés à 5 jeux de données de test pour chacun des deux jeux de données MovieLens.

3.2.2 Protocole d'expérimentation

3.2.2.1 Les approches de recommandation

Nous avons utilisé les deux principales techniques de recommandation sociale proposées dans l'état de l'art : les k plus proches voisins (KNN) (approche à base de mémoire, voir paragraphe 2.1.1.2) et la factorisation de matrice (approche à base de modèle, voir paragraphe 2.1.1.3). Pour la technique des k plus proches voisins, nous avons utilisé le coefficient de corrélation de Pearson (voir équation 2.1.1.2) lors du calcul des similarités inter-utilisateurs. Il s'agit de la mesure de similarité la plus populaire dans le domaine du filtrage collaboratif. Le paramètre k a été fixé à différentes valeurs en suivant toujours les indications présentes dans la littérature [Yu et al., 2014]. Pour la méthode de factorisation de matrice, nous avons utilisé la technique ALS (moindre carrés alternés). La technique ALS n'est pas toujours la plus performante (en fonction des données), mais il s'agit d'une technique très populaire dans les systèmes de recommandation [Koren, 2010]. Les paramètres tels que le nombre de caractéristiques latentes, le taux d'apprentissage et le paramètre de régularisation ont été fixés en fonction du jeu de données sur lequel la technique était appliquée.

Nous avons choisi d'utiliser les paramètres les plus utilisés dans la littérature (tableau 3.1) pour que les résultats obtenus avec ces deux approches soient représentatifs du domaine du filtrage collaboratif. Bien que ces deux approches de recommandation n'exploitent que les préférences des utilisateurs, elle les utilisent différemment et offrent des recommandations différentes aux utilisateurs. En utilisant ces deux types d'approches, nous souhaitons également montrer la nature générique de nos mesures d'identification.

3.2.2.2 L'évaluation de la qualité des recommandations

La méthode offline de validation a été utilisée avec des jeux de données d'apprentissage et de test. Connaissant la note mise par l'utilisateur sur une ressource n_{ur}, nous pouvons calculer les deux estimations fournies par les approches de recommandation sociale que nous avons implémentées : $\textit{Prédiction}_{u,r}^{knn}$ pour l'approche des k plus proches voisins et $\textit{Prédiction}_{u,r}^{FM}$ pour l'approche à base de factorisation de matrice.

À partir de ces information brutes, il est possible de calculer plusieurs types d'erreurs pour évaluer la performance des deux approches de recommandation. Nous avons sélectionné deux mesures d'évaluation de l'erreur : la RMSE (*Root Mean Squared Error*, voir section 2.1.1.7) et le HitRatio (cf. section 2.1.1.7). Ces mesures mesurent une information différente concernant la qualité de la recommandation. La RMSE mesure l'écart moyen entre les notes réelles et les

notes estimées. Plus la RMSE est faible, meilleurs sont les résultats. Le HitRatio représente le pourcentage de recommandations satisfaisantes fournies par l'algorithme de recommandation.

Pour permettre la validation de nos mesures d'identification des GSU, nous avons calculé chacune des ces mesures d'erreur sur chaque utilisateur. Ainsi, nous pouvons observer la moyenne des erreurs commises sur les utilisateurs identifiés par les mesures et analyser leurs performances.

3.2.2.3 La validation des résultats

Les mesures d'identification que nous avons proposées évaluent le degré de différence d'un utilisateur par rapport au reste des utilisateurs, en exploitant uniquement ses préférences. Les utilisateurs présents dans les données ne sont pas étiquetés GSU / non-GSU, alors comment vérifier que les utilisateurs identifiés par nos mesures sont bien des GSU ?

Le concept de GSU a été défini pour expliquer l'incapacité des systèmes à base de FC à fournir des recommandations de qualité à certains de leurs utilisateurs [Claypool et al., 1999]. Puisque les utilisateurs de notre jeu de données ne sont pas étiquetés, il n'est en effet pas possible de s'assurer que nos mesures identifient uniquement des GSU. En revanche, la présence d'utilisateurs recevant des recommandations de bonne qualité parmi les utilisateurs identifiés par nos mesures nous donne une approximation des erreurs commises par les mesures.

Dans le contexte de la recommandation sociale, un utilisateur correspondant à la définition d'un GSU (voir paragraphe 2.1.3.3) va à l'encontre des quatre propriétés fondamentales permettant de lui fournir des recommandations de bonne qualité (voir paragraphe 2.2.1). En effet, l'objectif de l'identification des GSU est de proposer une nouvelle approche de recommandation adaptée à leurs besoins spécifiques, il est donc nécessaire de s'assurer que nos mesures n'identifient que des utilisateurs mal servis par les approches classiques.

3.2.2.4 La méthode ClustGSU

La méthode ClustGSU repose principalement sur deux paramètres : la mesure de distance utilisée pour calculer l'écart entre un utilisateur et le centre de chaque cluster, ainsi que le nombre de clusters à prendre en considération. Nous avons exécuté plusieurs fois notre algorithme de clustering en faisant varier ces paramètres afin d'étudier l'impact de ces paramètres sur les résultats. Pour cela, nous avons identifié 10% d'utilisateurs comme GSU dans le jeu de données MovieLens100K [Gras et al., 2015c]. Nous avons comparé la moyenne des RMSE obtenues pour ces utilisateurs en fonction de quatre mesures de distance : la norme L_2, le coefficient de corrélation de Pearson, la norme L_1 et la distance Cosinus (voir paragraphe 2.2.3). La norme L_2 correspond à la somme des écarts au carré entre 2 vecteurs et la norme L_1 correspond à la somme des écarts absolus. Nous avons également fait varier le nombre de clusters K recherchés par l'algorithme KMeans de 5 à 30. Les résultats d'un algorithme de KMeans dépendant énormément de l'état d'initialisation, nous avons effectués 5 exécutions pour chaque combinaison de paramètres (soit 80 exécutions au total) et avons considéré la moyenne des résultats obtenus.

Les résultats obtenus sont présentés dans la figure 3.8. Notons que plus les RMSE obtenues par les utilisateurs identifiés comme GSU sont élevées, plus la méthode d'identification est performante. L'utilisation de la norme L_2 pour mesurer la distance entre deux utilisateurs semble en tous points supérieure aux autres mesures de distance. Même si la norme L_1 propose des performances similaires lorsque l'on distingue 30 clusters, les résultats de la norme L_2 restent supérieurs. Les autres mesures proposent quant à elles des résultats plus faible pour la tâche d'identification des GSU. Avec une RMSE moyenne sur les GSU de $1,204$, la norme L_2 avec $K = 10$ présente les résultats les plus encourageants pour la méthode $ClustGSU$.

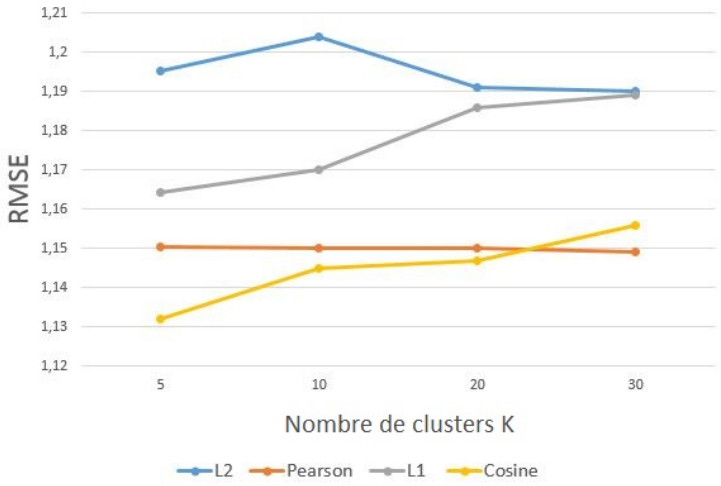


FIGURE 3.8 – RMSE des GSU identifiés en fonction des paramètres utilisés par l'algorithme ClustGSU

3.2.2.5 Les mesures d'identification évaluées

Dans les expérimentations que nous présentons, nous comparons un ensemble de mesures d'identification des GSU. Nous comparons tout d'abord trois mesures intuitives (Corr TMax, CorrMoy et Popularité) qui exploitent la qualité du voisinage d'un utilisateur. Nous avons fixé la variable T au nombre de voisins utilisés pour effectuer une recommandation, à savoir $T = 20$ pour MovieLens100K et $T = 50$ pour MovieLens20M. Rappelons que le nombre minimal de ressources à posséder en commun pour pouvoir calculer la similarité entre deux utilisateurs est 5 pour le calcul de la Popularité (voir paragraphe 3.1.1).

Ensuite, nous évaluons les trois mesures statistiques basées sur la mesure d'*Anormalité*. Nous comparons l'*Anormalité* de l'état de l'art [Haydar et al., 2012] avec les extensions à cette dernière que nous avons proposées (Équations (3.1.2.2) et (3.1.2.1)).

Nous étudions également les résultats obtenus avec deux méthodes à base de clustering. Une première méthode est basée sur l'algorihme *KMeans++* de [Ghazanfar and Prügel-Bennett, 2014], ainsi que *ClustGSU* qui a été mis au point par nos soins.

Nous présentons les résultats obtenus avec les trois mesures à base de distribution : la *Vraisemblance* de l'état de l'art, ainsi que la *VraisemblanceSBU* et la *VraisemblanceID* (équations (3.4) et (3.5)). L'état de l'art [Jones et al., 2011] a montré que l'imprécision des notes des utilisateurs est d'environ 1 point sur MovieLens100K. Le jeu de données MovieLens20M possédant la même échelle de notation, nous avons fixé la variable h à 1 dans l'équation (3.5) pour l'ensemble des expérimentations.

Enfin, nous avons évalué une méthode classique du domaine du Data Mining : la méthode du $K^{ième}$ plus proche voisin. Cette méthode sera notre référence pour le domaine de la détection d'*outliers*. On fixe ici K à 20 pour MovieLens100K et à 500 pour MovieLens20M.

Pour permettre cette comparaison, nous ajustons les seuils d'identification de manière à identifier le même nombre d'utilisateurs avec chacune des mesures. Ce nombre d'utilisateurs varie en fonction des différentes expérimentations ainsi qu'en fonction du jeu de données concerné.

3.2.2.6 Evaluation en trois étapes

Le protocole d'évaluation que nous avons utilisé pour évaluer et comparer les différentes mesures d'identification des GSU se divise en trois étapes.

Tout d'abord, nous évaluons la corrélation entre chaque mesure d'identification et la qualité des recommandations (RMSE et HitRatio) de chaque utilisateur. Cette corrélation sera un premier indice de la capacité de chaque mesure à estimer la qualité des recommandations fournies aux utilisateurs.

Ensuite, nous étudions la capacité de nos mesures à n'identifier que des utilisateurs mal servis par les systèmes de FC classiques. Il s'agit d'une forme de précision de nos mesures d'identification, nous l'avons donc nommée Précision (équation (3.6)). L'efficience d'une mesure, dans ce contexte, représente le ratio entre le nombre d'utilisateurs identifiés qui ont une forte erreur (nombre de bonnes détections) et le nombre total d'utilisateurs identifiés par la mesure (nombre total de détections).

$$Précision = \frac{\#RMSE(u) > Seuil}{\#Utilisateurs\ Identifis} \tag{3.6}$$

Enfin, nous analysons la répartition des erreurs des utilisateurs identifiés par les mesures. Plus une mesure identifie des utilisateurs très mal servis par les systèmes classiques, plus il est important que ces utilisateurs soient identifiés en amont de la recommandation. Nous comparons ici les minimums, 1^{er} quartile, médiane, $3^{ème}$ quartile et maximum de chaque distribution d'erreurs par utilisateur.

3.2.3 Corrélations entre les mesures d'identification et les erreurs de recommandation

Dans cette section, nous présentons les corrélations entre chaque mesure d'identification et la qualité des recommandations affectées à chaque utilisateur. Nous séparons notre analyse en deux sous-parties, la première dédiée aux corrélations obtenues avec l'approche KNN et la seconde dédiée aux corrélations obtenues avec une approche par factorisation de matrice (FM).

3.2.3.1 Corrélations obtenues à partir d'une approche mémoire : la technique des K plus proches voisins

Les GSU sont supposés recevoir des recommandations de mauvaise qualité dans le contexte de la recommandation sociale. Une bonne mesure d'identification de ces utilisateurs doit donc identifier des utilisateurs mal servis par les systèmes de FC classiques. La corrélation entre les mesures d'identification et les erreurs observées par utilisateur nous permet d'avoir un premier aperçu des capacités des mesures à être liées à la qualité des recomandations fournies et donc à isoler des utilisateurs mal servis. Puisqu'il s'agit d'obtenir un premier aperçu, nous avons effectué ces calculs sur MovieLens100K. Les corrélations sont présentées dans le tableau 3.2.

En statistique, l'analyse des résultats obtenus est traditionnellement faite à l'aide de la *p-value* de cette corrélation. On considère que la *p-value* est un indice pertinent lorsque les échantillons comparés sont d'une taille supérieure à 500 [Ratner, 2004]. Or, nous comparons les erreurs avec les mesures d'identification sur 821 utilisateurs dans le cadre de ce premier jeu de données, nous pouvons donc nous fier aux résultats de la *p-value*. Le tableau 3.2 présente entre parenthèse la valeur de la *p-value* pour chaque corrélation pertinente.

Mesure	RMSE	HitRatio
$Corr20MaxPos$	-0.10	0.04
$Corr20MaxNeg$	-0.02	0.05
$CorrMoyPos$	-0.10	0.01
$CorrMoyNeg$	-0.02	0.05
Popularité	-0.06	-0.01
$20^{th}NN$	-0.07	0.04
Anormalité	**0.41 (1.e-35)**	**-0.34 (5.e-25)**
AnormalitéCR	**0.42 (4.e-24)**	**-0.34 (1.e-36)**
AnormalitéCRU	**0.39 (8.e-31)**	**-0.39 (3.e-31)**
$ClustGhazanfar$	0.27 (5.e-15)	-0.20 (3.e-9)
$ClustGSU$	**0.40 (7.e-22)**	**-0.33 (6.e-33)**
$Vraisemblance$	**-0.38 (6.e-29)**	**0.35 (1.e-25)**
$VraisemblanceSBU$	**-0.42 (1.e-36)**	**0.42 (3.e-36)**
VraisemblanceID	**-0.49 (1.e-50)**	**0.45 (1.e-42)**

TABLE 3.2 – Corrélations (Pearson) entre les mesures d'identification et les erreurs observées avec l'approche des k plus proches voisins (MovieLens100K)

Précisons que plus une *p-value* est faible, plus la valeur de la corrélation est **significative**. On considère en général qu'une corrélation avec une *p-value* $< 0,01$ est une corrélation significative. Une corrélation **pertinente** est une corrélation $> 0,3$, car même si une corrélation de $0,10$ n'est pas due au hasard (c'est-à-dire significative), un lien aussi faible entre les deux variables n'est pas exploitable.

Les six premières mesures exploitent la qualité du voisinage d'un utilisateur. On remarque immédiatement qu'il n'existe pas de lien effectif entre ces mesures d'identification et l'erreur observée sur les utilisateurs. Cela signifie donc que, contrairement à l'intuition liée à la définition d'un GSU (cf. paragraphe 2.1.3.3), la qualité du voisinage d'un utilisateur n'est pas suffisante pour identifier des utilisateurs mal servis par un système de FC. La mesure $20^{th}NN$, issue du domaine de la détection des *outliers*, n'est donc pas adaptée au domaine de la recommandation sociale. Cela semble confirmer les conclusions de la littérature : les techniques à base de distance pure ne sont pas adaptées pour les jeux de données extrêmement vides.

Les corrélations observées entre les mesures d'*Anormalité* et les mesures d'erreur par utilisateur semblent très pertinentes. Ces trois mesures possèdent des corrélations significatives (*p-value* $< 0,01$) avec les mesures de RMSE et de HitRatio. Cela signifie que plus un utilisateur est anormal au sens de ces mesures, plus sa RMSE sera élevée et son HitRatio sera faible, donc plus l'erreur commise par les SR sera grande. Les valeurs de corrélation sont très proches avec néanmoins une corrélation plus élevée entre l'*AnormalitéCRU* et le HitRatio ($-0,39$ contre $-0,34$). L'écart entre les corrélations des trois mesures basées sur l'*Anormalité* avec la RMSE est plus faible, mais la conclusion est différente. C'est l'*AnormalitéCR* qui est la plus corrélée à la RMSE, avec une corrélation de $0,42$. A ce stade de notre analyse, rien ne permet d'établir laquelle des deux mesures (*AnormalitéCR* ou *AnormalitéCRU*) est la plus efficace pour l'identification des GSU. Néanmoins, ces hautes corrélations montrent qu'une mesure statistique dédiée à l'identification des GSU peut fonctionner sur des jeux de données creux.

Les corrélations entre la méthode de clustering présentée dans l'état de l'art (*ClustGhazanfar*) [Ghazanfar and Prugel-Bennett, 2011] et les différentes mesures d'erreur ne sont pas aussi

élevées que celles observées sur l'*Anormalité*. Cette manière de classifier les utilisateurs ne semble pas permettre d'estimer précisément la qualité des recommandations fournies aux utilisateurs. Bien que les corrélations soient moins pertinentes ($< 0,3$), elle sont néanmoins significatives. L'adaptation de cette méthode que nous avons proposée ($ClustGSU$) est quant à elle corrélée de manière pertinente et significative avec la RMSE et le HitRatio. Les résultats sont comparables à ceux obtenus avec l'*Anormalité* de l'état de l'art, une analyse plus en détails, permettant d'observer les différences entre ces deux méthodes d'identification, est présentée dans les prochaines sections.

Les trois dernières mesures sont des mesures basées sur la distribution des données. La $Vraisemblance$, qui est utilisée en général dans le domaine de la recherche d'information, a une corrélation avec la RMSE de $0,35$ et avec le HitRatio de $0,35$ ce qui signifie qu'il existe un lien entre l'erreur et la *Vraisemblance* observées sur un utilisateur. Ces corrélations sont néanmoins légèrement plus faibles que celles observées sur les mesures d'*Anormalité*. Cela semble confirmer les biais de la *Vraisemblance* que nous avons mis en évidence dans l'état de l'art (cf. paragraphe 2.2.3). De plus, chacune des deux améliorations que nous avons apportées à la $Vraisemblance$ au travers des mesures $VraisemblanceSBU$ et *VraisemblanceID* semble augmenter la corrélation entre les indices d'identification et les mesures de RMSE et de HitRatio. Les corrélations entre la *VraisemblanceSBU* et la RMSE ($-0,42$) et entre la *VraisemblanceSBU* et le HitRatio ($0,42$) représente une amélioration de 15% des résultats de la $Vraisemblance$. Cela semble indiquer que le biais utilisateur doit être traité pour améliorer les capacités des mesures d'identification basées sur la *Vraisemblance*. Ensuite, l'ajout d'une fenêtre glissante pour incorporer le biais de notation augmente quant à lui à nouveau d'environ 10% la corrélation entre la *VraisemblanceID* et les deux erreurs corrélées. Les corrélations les plus élevées observées sont de $-0,49$ avec la RMSE et de $0,45$ avec le HitRatio. Ces fortes corrélations suggèrent que la mesure de *VraisemblanceID* pourrait fournir de meilleurs résultats que les autres mesures dans la tâche d'identification des GSU.

3.2.3.2 Corrélations obtenues à partir d'une approche à base de modèle : la factorisation de matrice

Les corrélations observées entre les mesures d'identification et les erreurs observées sur les utilisateurs lorsque l'on utilise une approche par factorisation de matrice sont similaires à celles observées précédemment à l'aide de l'approche des k plus proches voisins.

Néanmoins, les corrélations présentées dans le tableau 3.3 sont plus élevées, atteignant jusqu'à $0,66$ pour la corrélation entre la mesure d'*AnormalitéCRU* et la RMSE et une corrélation de $0,65$ entre la *VraisemblanceID* et la RMSE.

Avec l'approche à base de modèle, toute les mesures sont plus corrélées avec la RMSE qu'avec le HitRatio, à l'exception de certaines techniques basées sur le voisinage. Une explication à ce phénomène est que la factorisation de matrice cherche à minimiser l'erreur quadratique, et donc utilise la RMSE pour optimiser son modèle. Les six premières techniques, à base de distance, ne sont pas plus corrélées à la qualité des recommandations que dans le cas de l'approche à base de mémoire. Cependant, parmi les mesures basées sur l'*Anormalité*, l'*AnormalitéCRU* affiche les plus fortes corrélations avec les deux mesures d'erreur : $0,66$ avec la RMSE et $0,40$ avec le HitRatio. Contrairement aux conclusions de la section précédente, la mesure d'*AnormalitéCR* présente les corrélations les plus faibles avec la qualité des recommandation : $0,48$ avec la RMSE et $0,35$ avec le HitRatio. L'*Anormalité* de l'état de l'art obtient de meilleurs résultats. Ces corrélations indiquent que la mesure d'*AnormalitéCRU* est la plus adaptée des trois mesures d'identification

Mesure	RMSE	HitRatio
$Corr20MaxPos$	0,02	-0,01
$Corr20MaxNeg$	-0,04	0,10
$CorrMoyPos$	0,048	-0,06
$CorrMoyNeg$	-0,04	0,10
Popularité	-0,04	0,01
$20^{th}NN$	0,01	0,01
Anormalité	**0,49 (1.e-35)**	**-0,37 (5.e-25)**
AnormalitéCR	**0,48 (4.e-24)**	**-0,35 (1.e-36)**
AnormalitéCRU	**0,66 (8.e-31)**	**-0,40 (3.e-31)**
$ClustGhazanfar$	0,28 (5.e-15)	-0,19 (3.e-9)
$ClustGSU$	**0,47 (7.e-22)**	**-0,35 (6.e-33)**
$Vraisemblance$	**-0,43 (6.e-29)**	**0,30 (1.e-25)**
VraisemblanceSBU	**-0,57 (1.e-36)**	**0,29 (3.e-36)**
VraisemblanceID	**-0,65 (1.e-50)**	**0,44 (1.e-42)**

TABLE 3.3 – Corrélations (Pearson) entre les mesures d'identification et les erreurs observées avec la technique ALS (MovieLens100K)

basées sur l'*Anormalité*.

La méthode $ClustGSU$ présente également des corrélations significatives et pertinentes avec une approche à base de modèle. La conclusion est la même, une analyse plus en détails des résultats est réalisée dans la section suivante pour évaluer la pertinence de cette méthode.

Enfin, les trois dernières mesures présentent des corrélations aussi encourageantes qu'avec l'approche à base de mémoire. Plus encore, la mesure de *VraisemblanceID* affiche des corrélations très fortes avec les deux mesures d'erreur : $0,65$ avec la RMSE et $0,44$ avec le HitRatio. La corrélation entre la *VraisemblanceID* et la RMSE est 12% plus élevée que celle entre la *VraisemblanceSBU* et la RMSE ($0,65$ contre $0,57$). La corrélation entre la *VraisemblanceID* et le HitRatio est plus élevée de 32% que celles observées entre les mesures de *Vraisemblance* ou de *VraisemblanceSBU* et le HitRatio ($0,44$ contre $0,30$). Ces corrélations sont très proches de celles de la mesure d'*AnormalitéCRU*, ce qui semble confirmer que les mesures d'*AnormalitéCRU* et de *VraisemblanceID* sont les plus adaptées pour identifier des GSU.

Seules les corrélations des différentes *Anormalités*, de $ClustGSU$ et des différentes *Vraisemblances* sont significatives et pertinentes. Nous étudierons donc en détails ces mesures dans les expérimentations suivantes. Nous conservons également la méthode *ClustGhazanfar* car ils s'agit de notre référence de comparaison pour la méthode $ClustGSU$.

Rappelons que la corrélation reflète le lien entre deux variables sur l'ensemble de leurs observations. Il pourrait donc y avoir une forte relation sur un sous-ensemble seulement des observations de ces variables. Dans ce cas, la corrélation ne permettrait pas d'identifier cette relation. C'est pourquoi, dans les expérimentations suivantes, nous n'utiliserons plus la corrélation entre les variables mais nous observerons directement la répartition des erreurs observées sur les utilisateurs identifiés par chaque mesure.

3.2.4 Précision des mesures d'identification

Comme nous l'avons déjà mentionné, il est très important qu'une mesure d'identification des GSU n'identifie pas en tant que GSU un utilisateur qui reçoit des recommandations de bonne

qualité (il n'est pas un GSU).

Alors, pour évaluer la capacité des mesures à identifier les GSU, nous avons défini une mesure de précision adaptée. Plus la précision d'une mesure est élevée, plus cette mesure est efficace.

Puisque nous sommes dans le cadre de l'identification non supervisée des GSU, le nombre de GSU est inconnu. Cependant, pour évaluer la précision des mesures d'identification, les GSU doivent être connus. Nous proposons de définir un seuil d'erreur sur les recommandations proposées à l'utilisateur, seuil au dessous duquel un utilisateur ne peut pas être considéré comme un GSU. Pour donner un meilleur aperçu, nous utilisons deux valeurs pour ce seuil :

— la médiane des erreurs observées sur l'ensemble des utilisateurs. Cela signifie que nous considérons que 50% des utilisateurs ont une forte erreur, et peuvent donc potentiellement être qualifiés de GSU. Sur MovieLens100K avec une approche par k plus proches voisins, la RMSE médiane est de $0,95$ et le HitRatio médian par utilisateur est de $0,6$. La précision de la mesure d'identification associée sera alors : $Prec@Med$.

— le $3^{ème}$ quartile des erreurs observées sur les utilisateurs. On considère alors que seules les 25% plus hautes valeurs des erreurs sont de fortes erreurs. Sur MovieLens100K, les $3^{ème}$ quartiles des erreurs sont alors : $1,15$ pour la RMSE et $0,7$ pour le HitRatio. La précision de la mesure d'identification associée sera alors : $Prec@Q3$.

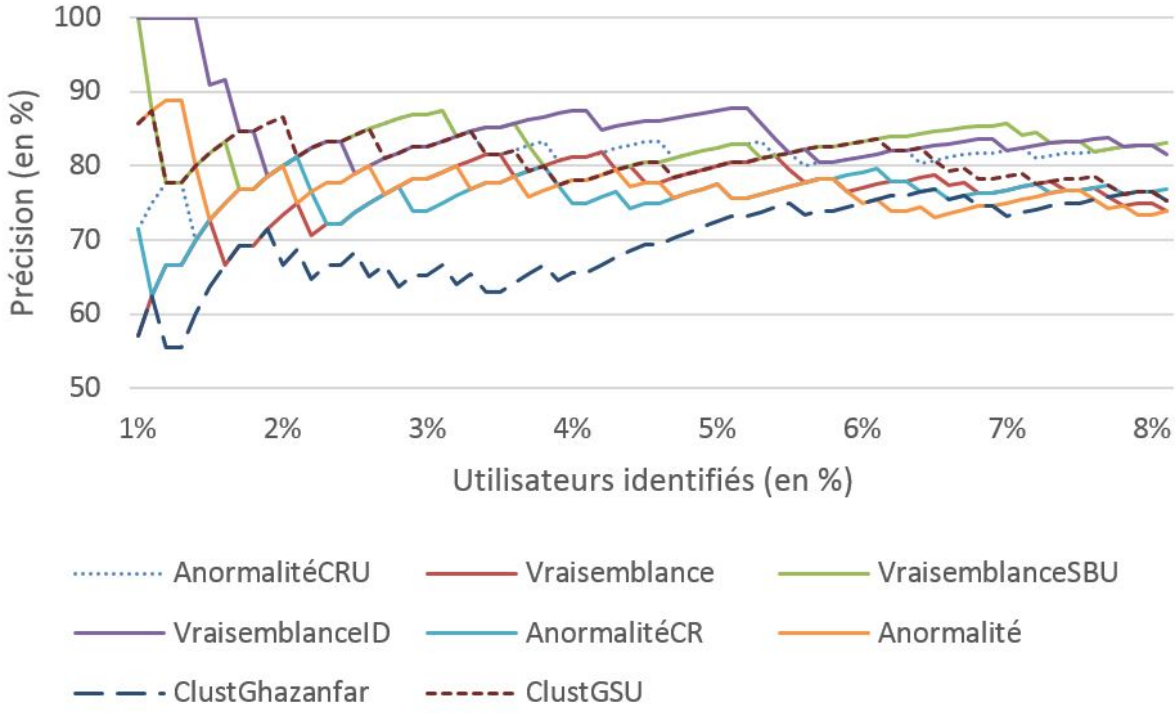
Précisons ici que nous ne pouvons pas calculer le rappel pour ces mesures d'identification puisque tous les utilisateurs qui ne reçoivent pas des recommandations de qualité de la part des algorithmes de FC ne sont pas des GSU.

3.2.4.1 Analyse des résultats sur MovieLens100K

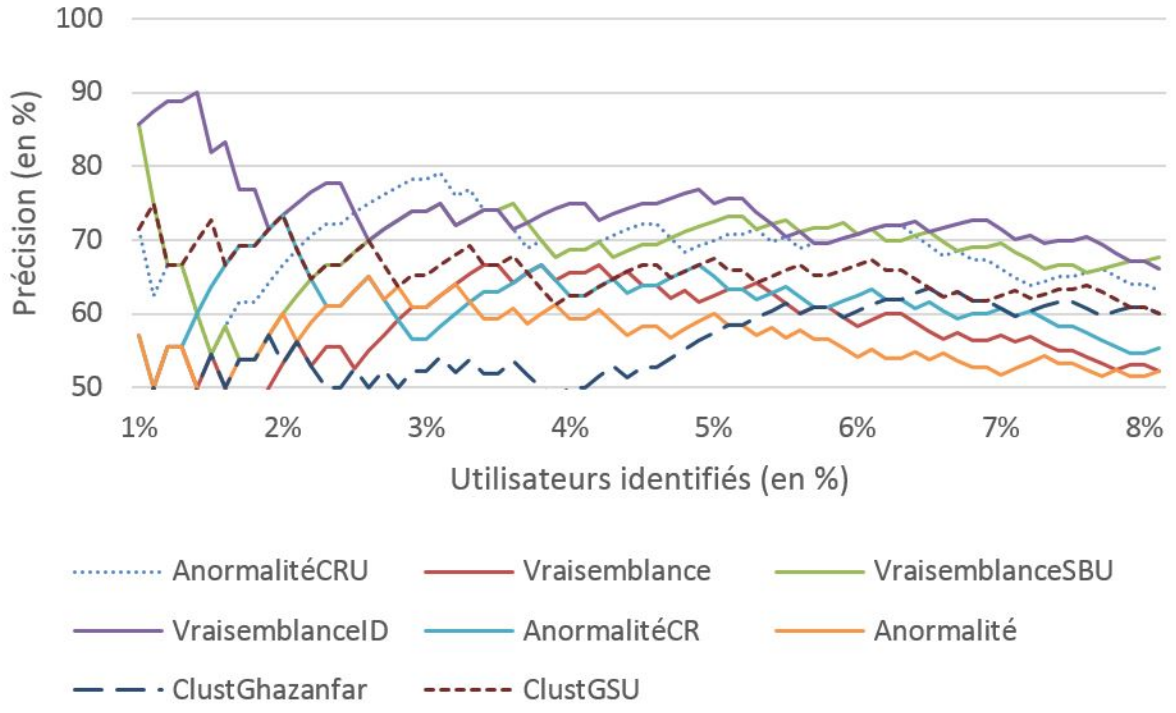
Les figures 3.9 et 3.10 présentent les précisions des mesures d'identification (que nous avons isolées dans la partie précédente) en fonction du nombre de GSU sur MovieLens100K. Nous identifions entre 1% (8 utilisateurs) et 8% d'utilisateurs (environ 65 utilisateurs). Nous choisissons de ne pas identifier plus de GSU car ils sont, par définition, rares.

La figure 3.9(a) représente la *précision* à la médiane ($Prec@Med$) des mesures d'identification obtenue avec la RMSE. Seules les mesures de *VraisemblanceID*, d'*VraisemblanceSBU* et d'*AnormalitéCRU* ont une précision supérieure à 80% jusqu'à 8% d'utilisateurs identifiés. La méthode *ClustGSU* permet d'identifier plus de 6% d'utilisateurs en respectant ce seuil minimum de précision de 80%. La *Vraisemblance* permet d'identifier jusqu'à 5% d'utilisateurs et les mesures d'*Anormalité* et d'*AnormalitéCR* ne peuvent identifier que 3% d'utilisateurs si elles doivent respecter une précision minimum de 80% d'utilisateurs identifiés possédant une RMSE supérieure à la médiane des RMSE du système. La *VraisemblanceID* nous permet même d'identifier jusqu'à 5,3% d'utilisateurs tout en garantissant une précision de plus de 87%, ce qui est largement supérieur aux résultats des autres mesures. L'*AnormalitéCRU* montre des résultats significativement supérieurs (entre 8 et 12%) aux deux autre mesures d'*Anormalité*.

La figure 3.9(b) présente la *précision* au troisième quartile ($Prec@Q3$) des mesures d'identification, toujours en fonction de la RMSE. De manière logique, les précisions sont plus faibles sur ce second graphique. Aucune mesure ne permet d'identifier plus de 2% des utilisateurs en respectant une précision de 80%. Par contre, si on diminue ce minimum de précision exigé à 70%, les trois mesures les plus performantes sont encore la *VraisemblanceSBU*, la *VraisemblanceID* et l'*AnormalitéCRU*, permettant d'identifier jusqu'à 7% d'utilisateurs. Parmi les mesures restantes, la méthode *ClustGSU* affiche des résultats proches de ceux des trois meilleures mesures, mais cette méthode ne permet d'identifier que 2,7% d'utilisateurs si elle respecte une précision de 70%. Les conclusions sur les autres mesures sont similaires à celles de la figure 3.9(a). Nous pouvons donc conclure que lorsque l'on utilise la RMSE pour évaluer la qualité des recommandations, les



(a) $Prec@Med - RMSE$



(b) $Prec@Q3 - RMSE$

FIGURE 3.9 – Précision des mesures d'identification sur MovieLens100K en fonction du nombre d'utilisateurs identifiés (RMSE)

mesures les plus efficaces pour l'identification des GSU sont la *VraisemblanceSBU*, la *VraisemblanceID*, l'*AnormalitéCRU* et *ClustGSU*.

La figure 3.10(a) représente la précision des mesures d'identification calculée à partir du HitRatio des recommandations fournies aux utilisateurs. Elle permet de constater que l'approche de clustering *ClustGSU* peut également garantir jusqu'à 85% de précision. Les bons résultats de *ClustGSU*, confirmés par la figure 3.10(b), affichent une précision en moyenne 10% supérieure à celle de la méthode de clustering de l'état de l'art : $ClustGhazanfar$. Sur la figure 3.10(a), il est difficile de déterminer laquelle des quatre mesures (*VraisemblanceSBU*, *VraisemblanceID*, *AnormalitéCRU* et *ClustGSU*) est la plus performante pour identifier des utilisateurs recevant de mauvaises recommandations selon la mesure de HitRatio.

La figure 3.10(b) présente la précision au troisième quartile des mesures d'identification et permet de distinguer les performances des meilleures mesures. C'est la mesure de *VraisemblanceSBU* qui permet d'identifier le plus d'utilisateurs tout en respectant une précision d'au moins 70%. Cette même mesure permet d'identifier jusqu'à 3,5% d'utilisateurs en respectant une précision de 80%. Les trois autres mesures les plus performantes présentent des résultats à peine inférieurs et sont également de très bonnes mesures pour l'identification de GSU.

Pour conclure, les performances des mesures d'identification ne sont pas les mêmes en fonction de la mesure utilisée pour évaluer la qualité des recommandations. Avec la RMSE, c'est la *VraisemblanceID* qui obtient les meilleures performances, tandis qu'avec le HitRatio, c'est la *VraisemblanceSBU* qui est mise en avant. Seules les mesures d'*AnormalitéCRU*, de *VraisemblanceID*, de *VraisemblanceSBU* et *ClustGSU* seront alors conservées pour les expérimentations qui suivent. Les trois mesures les plus précises (*AnormalitéCRU*, *VraisemblanceID* et *VraisemblanceSBU*) ont pour point commun de prendre en compte le biais des utilisateurs. Cela semble donc vérifier que l'utilisation des notes centrées des utilisateurs améliore la précision de l'identification de nos mesures.

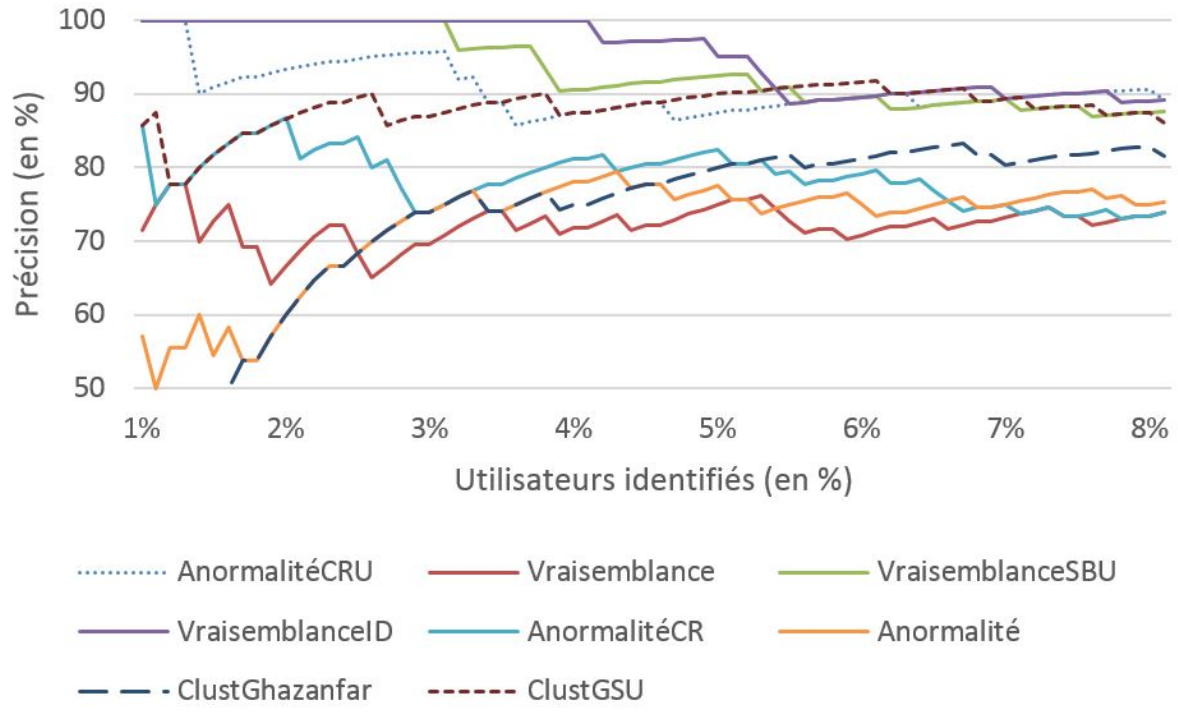
Cette première analyse des résultats obtenus sur MovieLens100K nous permet d'avoir un aperçu global de la précision des mesures d'identification en fonction du nombre d'utilisateurs identifiés comme GSU. Pour valider ces résultats, nous évaluons ces mesures sur un jeu de données plus vaste : MovieLens20M. Seule l'approche par modèle à base de factorisation de matrice peut supporter cette quantité d'informations, cela nous permettra donc également de valider la généricité à l'approche de recommandation des résultats précédents.

3.2.4.2 Validation des mesures proposées à grande échelle

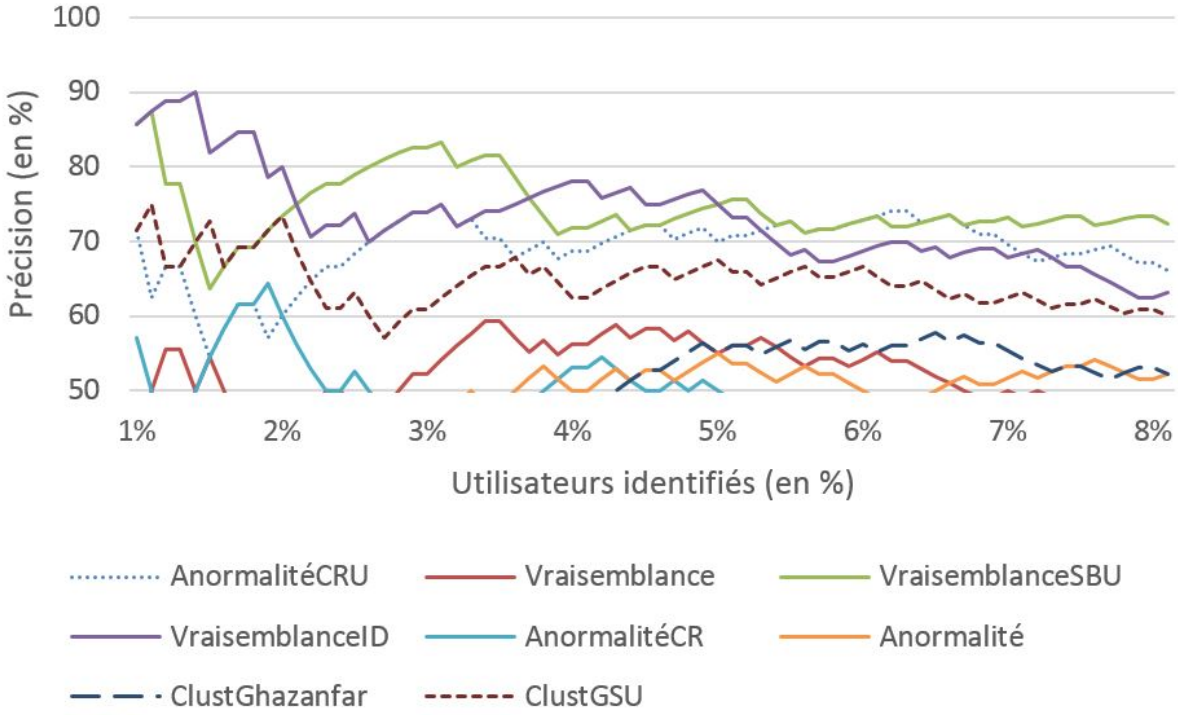
La factorisation de matrice a montré de très bons résultats lors de l'analyse des corrélations entre les mesures d'identification que nous avons proposées et les erreurs de recommandation observées sur ces utilisateurs. Les performances des mesures face à une très grande communauté d'utilisateurs pourraient se dégrader.

Sur la figure 3.11, qui montre la HitRatio@Med, on remarque en premier que la méthode *ClustGSU* obtient une précision significativement inférieure (entre 18 et 22%) que les autres mesures. Cette conclusion est en accord avec les valeurs de corrélation du tableau 3.3. Cependant, la différence en terme de corrélation entre *ClustGSU* et *AnormalitéCRU* est plus petite que la différence observée ici.

AnormalitéCRU a une précision légèrement plus faible (entre 2% et 3,5%) que *VraisemblanceSBU* et *VraisemblanceID*, tandis que la différence entre leurs corrélations aux mesures d'erreurs est plus élevée (environ 17%). *VraisemblanceID* et *VraisemblanceSBU* sont les deux mesures les plus précises, elles ont des précisions similaires, quel que soit le nombre d'utilisateurs identifiés.



(a) $Prec@Med - HitRatio$



(b) $Prec@Q3 - HitRatio$

FIGURE 3.10 – Précision des mesures d'identification sur MovieLens100K en fonction du nombre d'utilisateurs identifiés (HitRatio)

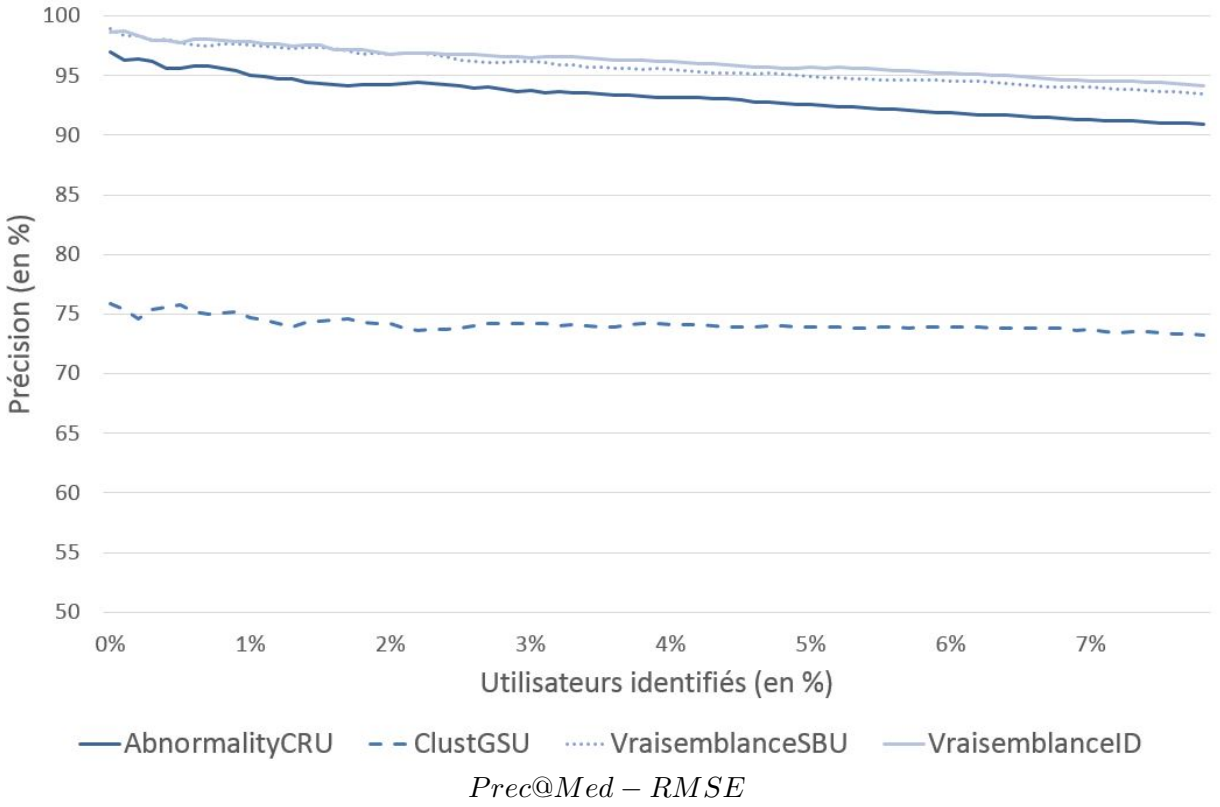


FIGURE 3.11 – Précision des mesures d'identification sur MovieLens20M en fonction du nombre d'utilisateurs identifiés (RMSE)

Si le nombre de GSU identifiés est fixé a 1% (environ 1 200 utilisateurs), utiliser l'*AnormalitéCRU* garantit d'identifier des utilisateurs avec une RMSE plus élevée que la médiane des RMSE du système avec une précision de 96%, tandis que la *VraisemblanceID* et l'*VraisemblanceSBU* garantissent d'identifier 1% des utilisateurs avec une précision de 98%.

Les résultats montrés par la figure 3.11 attribuent aux mesures d'identification sélectionnées des précisions supérieures à celles sur MovieLens100K.

Pour analyser plus en détails la figure 3.11, nous avons défini quatre seuils pour la précisions pour $Prec@Med$: 90%, 92,5%, 95% et 97,5% (C'est-à-dire 10%, 7,5%, 5% et 2,5% de mauvaises identifications respectivement) et nous étudions le nombre de GSU identifiés par les mesures (voir tableau 3.4, partie gauche). Nous ne présentons plus la méthode *ClustGSU* puisqu'elle n'atteint jamais ces seuils de précision.

Si nous voulons garantir 90% d'identification correcte avec l'*AnormalitéCRU*, le nombre d'uti-

	$Prec@Med$				$Prec@Q3$			
	90%	92,50%	95%	97,50%	90%	92,50%	95%	97,50%
AnormalitéCRU	10,00%	5,40%	1,30%	0,20%	0,60%	//	//	//
VraisemblanceSBU	15,40%	10,30%	5,20%	1%	2,70%	1,50%	0,60%	//
VraisemblanceID	17,90%	11,50%	**6,70%**	**1,90%**	3,50%	1,90%	**1%**	//

TABLE 3.4 – Nombre de GSU identifiés (en %) en fonction des seuils de précision

lisateurs identifiés doit être au maximum de 10%, soit environ 12 000 (valeur non affichée sur la figure 3.11). Cela pourrait aller jusqu'à 15,4% et 17,9% avec les mesure d'*VraisemblanceSBU* et de *VraisemblanceID*, ce qui correspond à une augmentation de plus de 50% du nombre d'utilisateurs identifiés avec le même seuil de précision. Rappelons que la différence de précision entre l'*AnormalitéCRU* et la *VraisemblanceID*, par exemple, n'est que de 3,5% maximum.

En étant plus strict pour garantir 92,5% de précision des mesures, le nombre d'utilisateurs identifiés est diminué (5,4% d'utilisateurs identifiés avec l'*AnormalitéCRU*). Avec les mesures *VraisemblanceSBU* et *VraisemblanceID* le nombre d'utilisateurs identifiés reste supérieur à 10%, soit environ deux fois plus qu'avec l'*AnormalitéCRU* .

Plus le seuil de précision choisi est élevé, plus l'écart entre la *VraisemblanceID* et l'*AnormalitéCRU* se creuse. Avec une précision de 97,5%, *VraisemblanceID* permet d'identifier dix fois plus d'utilisateurs que l'*AnormalitéCRU* : 1,9% des utilisateurs (environ 2 000 utilisateurs) peuvent être identifiés.

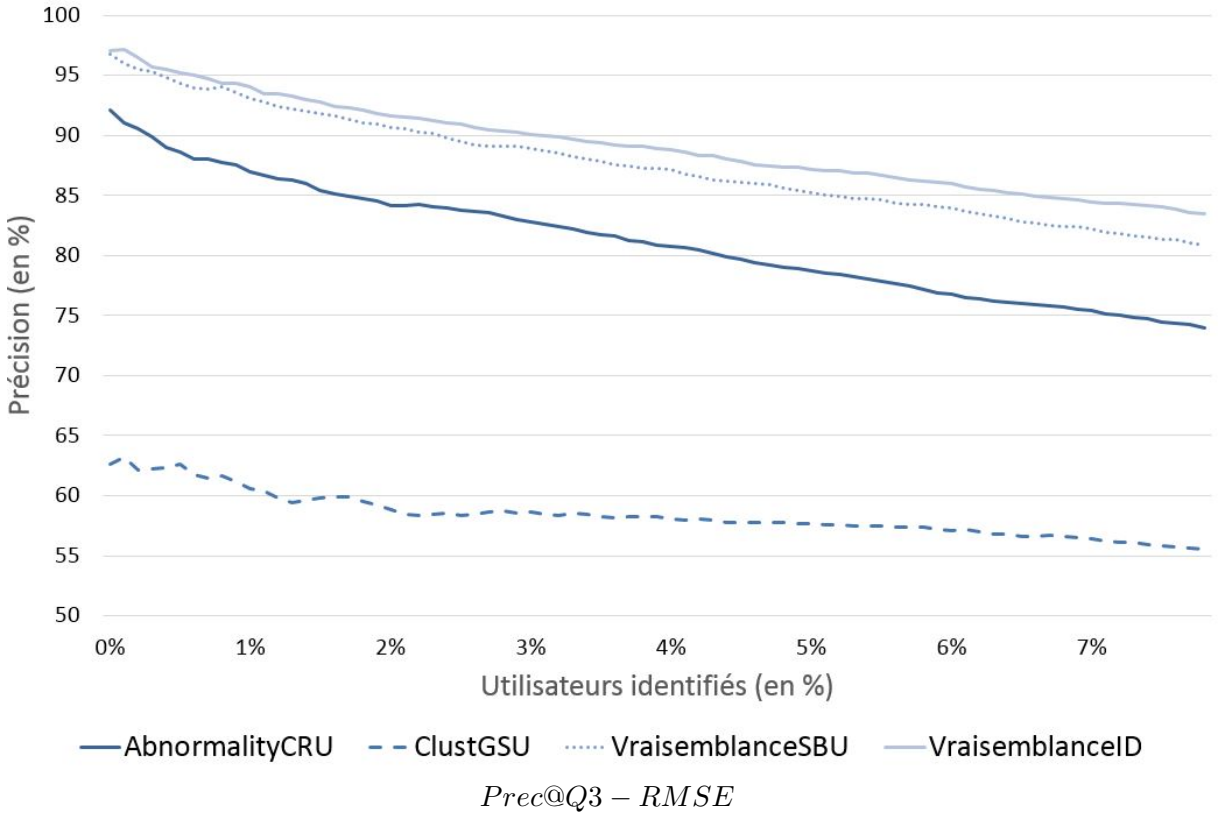


FIGURE 3.12 – Précision des mesures d'identification sur MovieLens20M en fonction du nombre d'utilisateurs identifiés (RMSE)

La figure 3.12 représente la précision au troisième quartile des mesures d'identification en fonction des RMSE observées. L'ordre des précisions des quatre mesures est préservé, par rapport à la figure 3.11 : la méthode *ClustGSU* reste la moins précise des quatre. Cependant, la différence entre la précision de la *VraisemblanceID* et des autre mesures s'est légèrement creusée (30% d'écart contre 20% auparavant pour l'*Anormalité* par exemple). Contrairement à la figure 3.11, *VraisemblanceID* et *VraisemblanceSBU* montrent une différence dans leur précisions respectives, notamment lorsque le nombre d'utilisateurs identifiés est supérieur à 1%. *VraisemblanceID* est celle qui permet d'obtenir la meilleure précision (environ 3% plus précis lorsque 8%

des utilisateurs sont identifiés).

Dans la partie droite du tableau 3.4, on peut voir que le nombre d'utilisateurs identifiés est plus faible lorsqu'on utilise $Prec@Q3$, ce qui était attendu. *AnormalitéCRU* ne peut pas garantir d'identifier des utilisateurs avec une RMSE supérieure au $3^{ème}$ quartile, avec une précision de la mesure supérieure ou égale à 92,5%. Néanmoins, la mesure de *VraisemblanceID* permet tout de même de garantir une précision de 95% lorsqu'elle identifie 1% des utilisateurs (environ 1 200), ce qui est très satisfaisant. Aucune des mesures ne peut garantir une précision de 97,5% avec Prec@Q3.

Nous avons également souhaité étudier le nombre d'utilisateurs recevant des recommandations de bonne qualité qui sont identifiés par les mesures d'identification des GSU. Une bonne RMSE est considérée ici comme une RMSE inférieure au premier quartile de la distribution totale des RMSE. Parmi les 1% d'utilisateurs identifiés par les mesures *VraisemblanceSBU* et *VraisemblanceID*, seulement 1% de ces utilisateurs (13 utilisateurs pour la *VraisemblanceSBU* et 11 utilisateurs pour la *VraisemblanceID*, voir tableau 3.5) ont une RMSE appartenant aux 25% plus faibles RMSE du système, alors que l'*AnormalitéCRU* en identifie le double.

AnormalitéCRU	*VraisemblanceSBU*	*VraisemblanceID*
22	13	**11**

TABLE 3.5 – Nombre d'utilisateurs bien recommandés identifiés en fonction de la mesure d'identification, avec une sélection de 1% des utilisateurs (1 200)

Nous pouvons donc conclure que *VraisemblanceSBU* et *VraisemblanceID*, que nous avons proposées, sont les deux mesures les plus précises. Sur MovieLens20M, lorsque l'on identifie 1% des utilisateurs avec une de ces deux mesures, ces utilisateurs font partie des 25% d'utilisateurs ayant les pires RMSE du système avec une précision de 95%, ce qui est très élevé. Cette expérimentation nous a permis de constater que pour garantir une identification optimale des GSU, le nombre d'utilisateurs identifiés ne doit pas dépasser 3% sur ce jeu de données.

3.2.5 Distribution des RMSE des GSU identifiés

Les expérimentations de la section précédente nous ont permis de conclure que l'*AnormalitéCRU*, la *VraisemblanceSBU* et la *VraisemblanceID* sont les mesures les plus précises. Pour mieux visualiser les utilisateurs identifiés avec ces différentes mesures, nous comparons ici les distributions de RMSE des GSU identifiés au travers du minimum, Q1, médiane, Q3 et maximum des valeurs de RMSE des GSU identifiés. Plus ces valeurs sont élevées, plus la mesure d'identification est précise. Pour être cohérent avec la conclusion de la partie précédente, nous observons les distributions des RMSE lorsque l'on identifie entre 1% et 3% de GSU avec chacune des mesures. Les valeurs sont présentées dans le tableau 3.6, qui montre à titre comparatif les informations sur la distribution des RMSE de l'ensemble complet des utilisateurs.

Le tableau 3.6 (colonne 1%) présente la distribution des valeurs de RMSE de chaque mesure d'identification lorsque 1% des utilisateurs sont identifiés. On peut voir que les mesures ont la même valeur pour le maximum des valeurs de RMSE, cette RMSE correspond d'ailleurs à la RMSE maximale observée sur l'ensemble complet des utilisateurs. L'analyse des valeurs minimales montre que l'utilisateur avec la plus faible RMSE identifié par l'*AnormalitéCRU* (0,43) possède une RMSE plus élevée que celle des utilisateurs avec la plus faible RMSE identifiés par les deux mesures basées sur la *Vraisemblance*, même si la précision de l'*Anormalité* est plus faible. La sélection d'un seul utilisateur bien recommandé par les mesure *VraisemblanceID* et

		1%			2%		
	Complet	*AnormCRU*	*VraisSBU*	*VraisID*	*AnormCRU*	*VraisSBU*	*VraisID*
Min.	0,06	**0,43**	0,11	0,11	0,35	0,11	0,11
1^{er} Q.	0,67	1,16	1,31	**1,34**	1,11	1,21	**1,24**
Med.	0,82	1,37	1,49	**1,51**	1,30	1,40	**1,42**
$3^{ème}$ Q.	1,00	1,59	1,68	**1,70**	1,52	1,58	**1,59**
Max.	**3,01**	**3,01**	**3,01**	**3,01**	3,01	3,01	3,01

		3%		
	Complet	*AnormCRU*	*VraisSBU*	*VraisID*
Min.	0,06	**0,11**	0,11	0,11
1^{er} Q.	0,67	1,08	1,17	**1,20**
Med.	0,82	1,27	1,34	**1,37**
$3^{ème}$ Q.	1,00	1,48	1,52	**1,54**
Max.	**3,01**	3,01	3,01	3,01

TABLE 3.6 – Informations sur les distributions de RMSE lorsque 1%, 2% et 3% des utilisateurs sont identifiés, en fonction de la mesure utilisée pour l'identification des GSU.

VraisemblanceSBU peut expliquer ce résultat. Le tableau 3.5 le confirme : l'*AnormalitéCRU* identifie d'avantage d'utilisateurs bien recommandés, ce qui est effectivement moins précis.

VraisemblanceSBU et *VraisemblanceID* ont des valeurs de précision similaires lorsque l'on identifie 1% d'utilisateurs (cf. figure 3.11). Les valeurs minimales et maximales des RMSE des utilisateurs identifiés avec ces mesures sont également similaires. Cependant, les premiers quartile, médiane et troisième quartile sont différents : *VraisemblanceID* obtient de meilleurs résultats (entre 1,1% et 2,5%). *VraisemblanceID* a tendance à identifier des utilisateurs avec des RMSE plus élevées, elle est plus précise.

Lorsque l'on s'intéresse à l'évolution des valeurs des quartiles entre 1% et 3% d'utilisateurs identifiés, ils diminuent puisque le nombre d'utilisateurs identifiés augmente. Cela était attendu puisque la précision des mesures diminue également (entre 5% et 7% de diminution) lorsque l'on augmente le nombre d'utilisateurs identifiés.

Une remarque importante est que lorsque l'on identifie 3% des utilisateurs, le premier quartile de la distribution des RMSE des utilisateurs identifiés avec la *VraisemblanceID* vaut 1,2, ce qui correspond au neuvième décile de l'ensemble complet des RMSE des utilisateurs. Cela signifie que 75% (2 750) des utilisateurs identifiés font partie des 10% plus mauvaises RMSE du système. De manière encore plus significative, lorsque l'on identifie 1% des utilisateurs avec la *VraisemblanceID*, 75% (922) des utilisateurs identifiés possèdent une RMSE appartenant aux 5% pires RMSE du système. La mesure est donc très précise.

3.3 Conclusion

Ces expérimentations permettent de conclure que, quelque soit le critère d'évaluation (corrélation, Précision, distribution), les mesures que nous avons proposées (*VraisemblanceSBU*, *VraisemblanceID* et *AnormalitéCRU*) présentent des résultats supérieurs à ceux des mesures de l'état de l'art. Cette conclusion reste vraie quel que soit le nombre de GSU identifiés. Les précisions de la mesure de *VraisemblanceSBU* et de *VraisemblanceID* sont proches, mais la plupart du temps c'est la mesure de *VraisemblanceID* qui montre de meilleurs résultats sur un grand jeu

de données. Cette mesure permet d'identifier jusqu'à 3% d'utilisateurs tout en garantissant une Prec@Q3 supérieure a 90% sur le jeu de données MovieLens20M. Le traitement du biais de notation ainsi que du biais utilisateurs avec un intervalle dynamique conduit à une meilleure précision d'identification des GSU.

Enfin, les résultats que nous venons de présenter sur le plus grand de nos jeux de données démontrent la généricité de nos mesures d'identification. En effet, la précision de l'identification ainsi que l'ordre entre les différentes mesures sont restés cohérents avec les deux approches de FC que nous avons expérimentées. L'utilisation d'un plus grand jeu de données nous a permis de constater que plus le nombre d'utilisateurs est important, moins le proportion d'utilisateurs aux préférences atypiques est élevée. Nous proposons donc des mesures d'identification génériques à l'approche de recommandation par FC utilisée dont la meilleur, la *VraisemblanceID*, permet d'identifier des GSU avec une précision suffisante pour pouvoir baser la suite de nos travaux sur cette mesure. Il s'agit donc désormais de se focaliser sur la tâche de modélisation des GSU afin d'améliorer la qualité des recommandations qui leur sont fournies.

Chapitre 4

Proposition de méthodes pour la modélisation des Grey Sheep Users

Sommaire

Dans la thèse que je défends, je pense qu'il est non seulement possible d'identifier les GSU en amont de toute recommandation (ce qui fait l'objet du chapitre précédent), mais qu'il est également possible de les modéliser, et de leur fournir des recommandations de meilleure qualité que ne le font les modèles classiques de recommandation sociale (ce qui fait, entre autres, l'objet de ce chapitre).

Dans ce chapitre, je présente et analyse tout d'abord les caractéristiques des GSU identifiés grâce aux mesures que j'ai proposées, en les comparant aux caractéristiques des non-GSU pour comprendre plus en détails les raisons qui entraînent une mauvaise qualité de recommandations pour les GSU, en outre d'être différent des autres. Ensuite, je propose de nouvelles stratégies pour sélectionner le voisinage d'un GSU, ainsi que des méthodes à base de factorisation de matrice, dédiées à l'amélioration de la qualité des recommandations fournies aux GSU.

Nous utiliserons les appellations $Train_{GSU}$ et $Test_{GSU}$ pour faire référence aux jeux de données composés des préférences d'apprentissage et de test des GSU et $Train_{non-GSU}$ et $Test_{non-GSU}$ pour les jeux de données composés des préférences des utilisateurs non-GSU.

Dans le chapitre précédent, la mesure d'identification la plus efficace étant la *VraisemblanceID*, j'ai donc choisi d'utiliser cette mesure pour identifier les GSU analysés et exploités dans ce chapitre.

4.1 Analyse des GSU identifiés

Nous sommes convaincus que la raison pour laquelle les GSU ne reçoivent pas des recommandations de qualité, même à l'aide d'une approche à base de FM, n'est pas seulement liée à la nature spécifique de leurs préférences, mais également au faible nombre d'utilisateurs représentant ces préférences spécifiques. Pour être en mesure d'améliorer la qualité des recommandations fournies aux GSU, je me suis intéressé dans un premier temps aux caractéristiques des GSU identifiés. Dans cette section, nous nous intéressons au nombre de votes moyen ou encore au nombre moyen de voisins des GSU, que nous comparons à ceux des utilisateurs non-GSU.

4.1.1 Les caractéristiques des GSU

Les éléments présentés ici sont tirés des jeux de données MovieLens (MovieLens100K et MovieLens20M). Pour garantir une précision de 90% à la médiane, nous avons utilisé la mesure *VraisemblanceID* pour identifier 6% des utilisateurs dans MovieLens20M comme GSU. Une précision de 90% à la médiane n'étant jamais atteinte par les mesures sur MovieLens100K (voir figure 3.9), nous avons garanti une précision de 80% à la médiane en identifiant 10% des utilisateurs comme GSU. Comme nous l'avons vu dans le chapitre précédent, l'erreur commise sur les prédictions des GSU est bien plus élevée que celle commise sur les prédictions des non-GSU (51% plus élevée en moyenne). Nous analysons dans cette section les caractéristiques des GSU pouvant être à l'origine de ces recommandations de mauvaise qualité.

Le tableau 4.1 présente les caractéristiques des préférences des GSU, comparées à celles des utilisateurs non-GSU, sur le jeu de données MovieLens100K.

	GSU	non-GSU
Nombre d'utilisateurs	82	739
Nombre moyen de notes	76	96
Nombre de notes (1^{er} quartile)	30	36
Note moyenne	3,19	3,64
Ecart-type moyen des notes	1,34	0,97

TABLE 4.1 – Caractéristiques des préférences des GSU et des non-GSU (MovieLens100K)

Dans les données de MovieLens100K, nous avons identifié 82 GSU parmi les 821 utilisateurs. Les GSU votent en moyenne moins de ressources que les non-GSU, avec une moyenne de 76 notes pour les GSU contre 96 notes pour les non-GSU, et cela s'explique par la manière dont nous avons défini la mesure *VraisemblanceID*, qui permet d'identifier les utilisateurs possédant la plus forte proportion de préférences spécifiques. Il est alors plus probable de posséder une forte proportion de préférences spécifiques (rares) lorsque l'on a exprimé moins de préférences. Afin de nous assurer que nous n'identifions pas uniquement des utilisateurs avec le nombre minimum

de notes (qui est de 20 pour ce jeu de données), nous calculons le premier quartile de cette distribution des nombre de notes par GSU. En effet, si nos mesures identifient en majorité des utilisateurs avec très peu de préférences, alors les recommandations de mauvaise qualité fournies à ces utilisateurs pourraient être dues au problème du démarrage à froid. Nous pouvons voir qu'avec un premier quartile de 30 pour les GSU, contre 36 pour les non-GSU, plus de 75% des GSU identifiés ont plus de 30 notes exprimées. La qualité des recommandations fournies aux GSU n'est donc pas directement liée au nombre de préférences qu'ils ont exprimées.

L'écart de 13% entre la note moyenne des GSU (3,19) et la note moyenne des non-GSU (3,64) montre que les GSU sont plus sévères dans leur notation que les non-GSU. La moyenne des notes est prise en compte dans la plupart des approches de recommandation sociale que nous avons présentées. Avoir une moyenne de notes différente ne devrait donc pas influencer la qualité des recommandations. Néanmoins, si nous observons l'écart-type moyen des GSU (1,34), en comparaison avec celui des non-GSU (0,97), alors nous pouvons conclure que non seulement les GSU sont plus sévères que les non-GSU, mais leurs notes sont également plus écartées de la moyenne. Avant d'aller plus loin dans nos conclusions sur les GSU identifiés, nous proposons d'analyser les GSU identifiés sur le jeu de données MovieLens20M au travers du tableau 4.2.

	GSU	non-GSU
Nombre d'utilisateurs	7 383	115 670
Nombre moyen de notes	92	130
Nombre de notes (1^{er} quartile)	29	35
Note moyenne	3,23	3,67
Ecart-type moyen des notes	1,37	0,9

TABLE 4.2 – Caractéristiques des préférences des GSU et des non-GSU (MovieLens20M)

Le grand nombre de GSU identifiés (7 383 utilisateurs) dans ce second jeu de données permet d'obtenir des indicateurs plus fiables concernant les caractéristiques des GSU. Nous pouvons voir que le nombre moyen de notes par utilisateur est plus élevé dans MovieLens20M. Cependant, si nous observons les premiers quartiles des distributions du nombre de notes par utilisateur pour les GSU (29 notes) et pour les non-GSU (35 notes) nous pouvons conclure que les 25% d'utilisateurs qui ont le moins voté dans ce jeu de données ont noté environ autant de ressources que les 25% d'utilisateurs qui ont le moins voté dans MovieLens100K. Les GSU identifiés dans MovieLens20M ne sont donc pas non plus exclusivement des utilisateurs avec peu de notes. Les notes moyennes des GSU (3,23) et des non-GSU (3,67) sont également très similaires à celles relevées sur le jeu de données MovieLens100K. Enfin, l'écart-type moyen des notes des GSU identifiés (1,37) est encore plus élevé dans MovieLens20M que dans MovieLens100K. A l'inverse, l'écart-type des non-GSU (0,9) est inférieur à celui observé dans MovieLens100K. Cela confirme que l'écart-type des notes des GSU identifiés est plus élevé que celui des non-GSU.

Parmi les approches de recommandation sociale que nous avons présentées dans ce manuscrit, l'approche KNN utilise la moyenne de l'utilisateur actif ainsi que les préférences de ses voisins pour estimer les préférences non exprimées par l'utilisateur actif. Puisque les préférences des GSU sont en général très écartées de leur moyenne (écart-type élevé), les préférences de leurs voisins sont encore plus importantes pour la prédiction des préférences non exprimées par les GSU que dans le cas des non-GSU. De plus, la littérature a principalement proposé des méthodes innovantes basées sur une approche KNN pour fournir des recommandations aux GSU (cf. paragraphe 2.2.5). Pour comprendre l'origine des recommandations de mauvaises qualité fournies à ces utilisateurs, je propose donc d'étudier leur voisinage.

4.1.2 Le voisinage des GSU

Un voisinage de qualité permet en général de fournir de bonnes recommandations aux utilisateurs. Les préférences spécifiques des GSU complexifient la sélection d'un voisinage de qualité pour l'estimation de leurs préférences. Pour cette raison, nous étudions ici le voisinage d'un GSU pour identifier les raisons des recommandations de mauvaise qualité des GSU. Intuitivement, les deux éléments les plus importants pour effectuer une recommandation basée sur le voisinage d'un utilisateur sont 1) le nombre de voisins utiles pour calculer une prédiction, et 2) la mesure permettant d'estimer la similarité entre deux utilisateurs.

Dans le chapitre précédent, nous avons utilisé les paramètres standards recommandés par l'état de l'art pour les approches de recommandation que nous avons implémentées. Dans ce chapitre, nous nous intéressons au sous-ensemble des utilisateurs qui sont des GSU. Nous comparons les différents paramètres présentés dans l'état de l'art pour proposer ensuite la stratégie la plus adaptée pour améliorer la qualité des recommandations fournies aux GSU.

Afin d'avoir un aperçu de l'impact du nombre de voisins utilisés k sur la qualité des recommandations fournies aux GSU, nous avons fait varier ce nombre de 1 à 50 sur le jeu de données MovieLens100K lorsque l'on utilise une approche KNN. Nous n'avons pas pu réaliser cette étude sur le jeu de données MovieLens20M car l'approche KNN ne passe pas à l'échelle. Par ailleurs, pour déterminer la mesure de similarité la plus adaptée aux GSU, nous comparons les résultats obtenus avec les méthodes les plus populaires de l'état de l'art : le coefficient de corrélation de Pearson, la similarité cosinus et la similarité basée sur la singularité des notes [Bobadilla et al., 2012a]. L'utilisation du coefficient de Jaccard pour pondérer la similarité entre deux utilisateurs permet d'améliorer la qualité des recommandations en général [Candillier et al., 2008]. Nous proposons donc d'étudier également l'impact de ce coefficient sur la qualité des recommandations fournies aux GSU lorsqu'il est combiné aux deux mesures de similarité de référence de l'état de l'art : le coefficient de Pearson (Pearson + Jaccard) et la similarité cosinus (Cosinus + Jaccard). La qualité des recommandations est mesurée à l'aide de la mesure de RMSE que nous avons utilisée dans le chapitre précédent.

Les graphiques 4.1 et 4.2 présentent les résultats obtenus lorsque l'on utilise les mesures de similarité détaillées dans l'état de l'art (paragraphe 2.1.1.2) pour l'approche des k plus proches voisins, respectivement sur les non-GSU et les GSU.

Rappelons que le nombre de voisins classiquement utilisé pour ce jeu de données est $k = 20$, pour l'ensemble des utilisateurs. Selon la figure 4.1, utiliser seulement $k = 13$ voisins avec le coefficient de Pearson + Jaccard permet d'obtenir l'erreur la plus faible, avec une RMSE moyenne de $0,95$ sur les non-GSU. Avec le même nombre de voisins utilisés pour effectuer les recommandations, la mesure de similarité cosinus permet d'obtenir une RMSE de $0,98$ lorsqu'elle est utilisée seule, et de $0,97$ lorsqu'elle est couplée avec le coefficient de Jaccard. La mesure de singularité est celle qui est la moins performante, avec une RMSE de $1,02$ avec $k = 13$ voisins utilisés. Dans [Bobadilla et al., 2012b], les auteurs présentent le coefficient de Jaccard comme un moyen de réduire l'impact du démarrage à froid utilisateur sur la qualité des recommandations. Sur la figure 4.1, on remarque que la mesure de similarité cosinus a besoin d'utiliser 24 voisins pour obtenir des résultats optimaux, tandis que lorsque pour Cosinus + Jaccard, seulement 17 voisins sont nécessaires. De la même manière, le coefficient de corrélation de Pearson seul a besoin de 20 voisins pour obtenir les meilleurs résultats tandis que seulement 13 voisins sont nécessaire avec Pearson + Jaccard. Cette observation semble confirmer les conclusions de [Bobadilla et al., 2012b], car si, en exploitant le coefficient de Jaccard, moins de voisins sont nécessaires pour effectuer des recommandations de qualité équivalente, alors les utilisateurs pour lesquels la sélection du voisinage est complexe (utilisateurs en situation de démarrage à froid, etc.) peuvent recevoir des

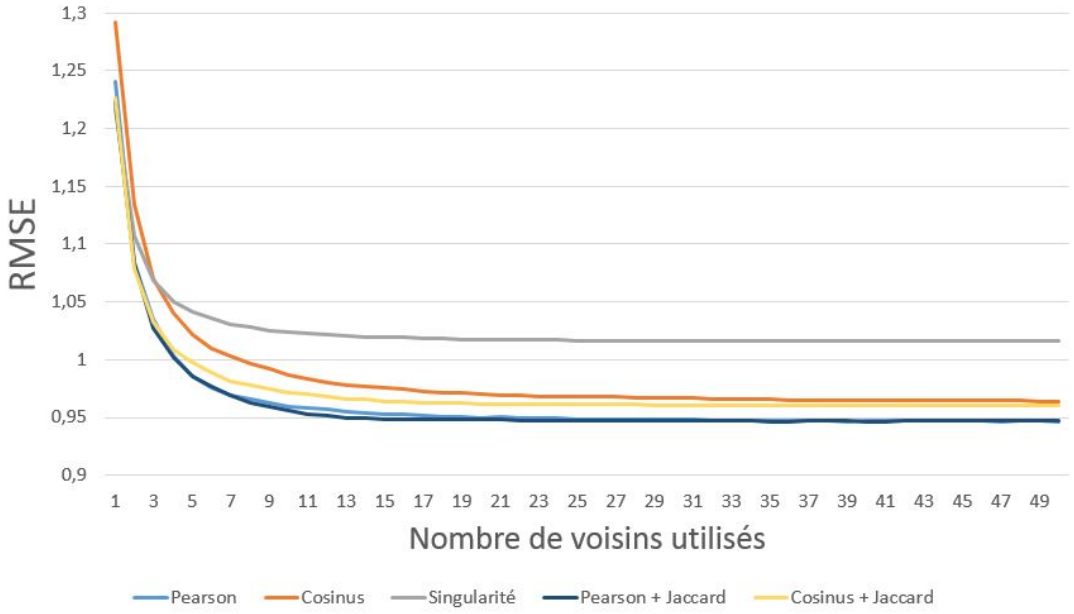


FIGURE 4.1 – Qualité des recommandations proposées aux utilisateurs non-GSU en fonction du nombre de voisins et de la mesure de similarité utilisés

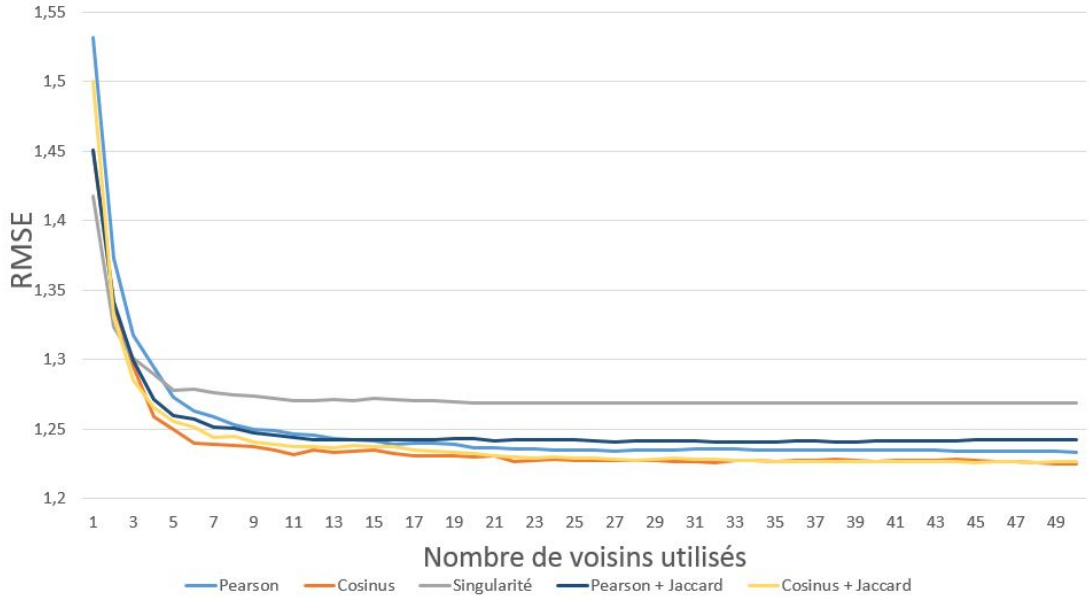


FIGURE 4.2 – Qualité des recommandations proposées aux GSU en fonction du nombre de voisins et de la mesure de similarité utilisés

recommandations de bonne qualité lorsqu'ils ont moins de voisins.

Sur la figure 4.2, on constate que sur la population des GSU, utiliser 11 voisins pour les recommandations est suffisant avec toutes les mesures de similarité, avec une RMSE de $1,23$ pour la mesure Cosinus. La similarité cosinus est la plus efficace sur les GSU et améliore de 0,02 la RMSE obtenue avec le coefficient de Pearson. Au delà de 11, l'amélioration des recommandations en terme de RMSE est faible. La similarité cosinus pondérée par le coefficient de Jaccard, le

coefficient de corrélation de Pearson, et le coefficient de corrélation de Pearson pondéré obtiennent également une RMSE moyenne inférieure à $1,25$ avec $k = 11$ voisins utilisés. Les différences entre ces mesures de similarité sont très faibles. Nous pouvons noter que pour les GSU, le coefficient de Jaccard ne permet pas d'améliorer les résultats des mesures classiques seules (Pearson et Cosinus), contrairement aux résultats obtenus sur les non-GSU.

La comparaison des graphiques (4.2 et 4.1) nous montre que la mesure de similarité la plus adaptée pour les utilisateurs non-GSU n'est pas la même que la mesure la plus adaptée pour les GSU. Les GSU sont en moyenne mieux servis lorsque la similarité est calculée à l'aide de la similarité cosinus tandis que les utilisateurs non-GSU semblent mieux servis par la combinaison des coefficients de Pearson et de Jaccard.

La mesure de similarité cosinus est réputée pour sa qualité sur des données extrêmement creuses. Le problème du manque de données touche tout particulièrement les GSU car, s'il est peu probable que deux utilisateurs aient co-voté un minimum ressources, il est encore moins probable qu'ils aient tous les deux exprimé des préférences spécifiques similaires sur ces ressources. Si tous les utilisateurs avaient évalués toutes les ressources, alors le nombre de notes par ressource serait bien plus élevé et il serait moins probable d'avoir une préférence spécifique sur cette ressource. Les GSU sont donc très impactés par le problème du manque de données et peuvent plus facilement bénéficier de l'utilisation de la mesure de similarité cosinus. De plus, le coefficient de corrélation de Pearson utilise la moyenne des notes de chaque utilisateur pour centrer ses notes. Or, je pense que la moyenne des notes d'un GSU a moins de sens que celle des autres utilisateurs puisque l'écart-type des préférences des GSU est très grand (10% plus élevé que celui des non-GSU).

Nous pouvons également faire deux remarques plus générales sur ces résultats. La première concerne la *Singularité* de l'état de l'art. Cette mesure, présentée dans [Bobadilla et al., 2012a], ne permet pas d'améliorer l'erreur observée ni sur les GSU (figure 4.2), ni sur les non-GSU (figure 4.1). Au contraire, les résultats obtenus par cette mesure sont très faibles et ne permettent pas d'envisager une solution au problème des GSU basée sur la singularité des notes. La seconde remarque que l'on peut faire est que la pondération avec le coefficient de Jaccard n'a pas le même effet sur les GSU et sur les non-GSU. En effet, l'utilisation de ce coefficient semble augmenter l'erreur sur les GSU, tandis que cela semble avoir un impact positif sur la qualité des recommandations fournies aux non-GSU. Cela peut s'expliquer par le fait que le coefficient de Jaccard donne la priorité aux voisins qui ont co-voté un maximum de ressources avec les GSU. Or, par définition, un GSU est très différent des utilisateurs qui ont co-voté les même ressources que lui. L'utilisation du coefficient de Jaccard n'est donc pas adapté à notre problème.

Lorsqu'une approche KNN est implémentée avec $k = x$, cela signifie qu'un maximum de x voisins seront utilisés pour le calcul de chaque prédiction. Les données extrêmement creuses auxquelles nous sommes confrontés dans ce travail ne permettent donc pas toujours de trouver ce maximum de voisins lorsque l'on souhaite prédire la note d'un utilisateur sur une ressource. Pour compléter les informations précédentes sur le voisinage des utilisateurs (GSU et non-GSU), nous présentons sur la figure 4.3 la proportion de prédictions pour chaque catégorie d'utilisateurs pour lesquels le nombre de voisins maximum k a pu être utilisé, en fonction de k.

Sur la figure 4.3, nous pouvons observer qu'une approche KNN utilise le nombre maximum de voisins k aussi souvent pour les GSU que pour les non-GSU. Cela signifie que les approches classiques peuvent trouver des voisins pour les GSU, mais que les voisins sélectionnés pour les GSU ne sont pas fiables (puisqu'ils conduisent à des recommandations de mauvaise qualité), en comparaison avec ceux trouvés pour les non-GSU. Notons que lorsque le paramètre classique pour ce jeu de données $k = 20$ est utilisé, à peine plus de la moitié des prédictions utilisent réellement 20 voisins.

La définition originale [Claypool et al., 1999] du problème des GSU signalait déjà que le

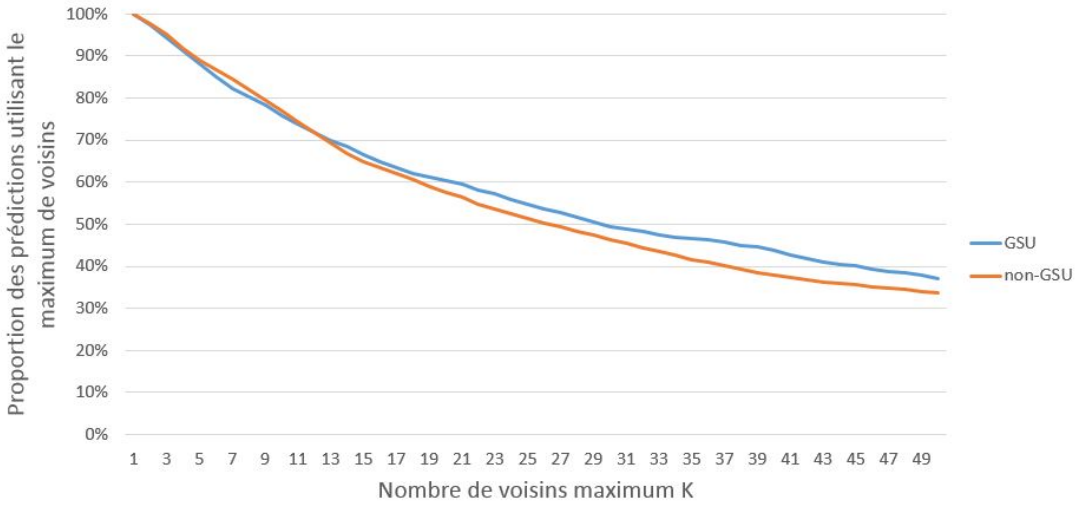


FIGURE 4.3 – Proportion des prédictions utilisant le maximum de voisins k en fonction de k

problème des GSU n'existe que lorsque la population d'utilisateurs est trop faible pour permettre de représenter efficacement l'intégralité des nuances de préférences de ses utilisateurs. En d'autres termes, plus le nombre d'utilisateurs est grand et plus il est difficile de posséder des préférences spécifiques. Le faible nombre d'utilisateurs du jeu de données MovieLens100K ne permet alors pas aux mesures de similarité que nous avons évaluées de trouver des utilisateurs qui présentent des préférences similaires à celles d'un GSU.

Un voisin fiable est un voisin permettant au système de fournir des recommandations de bonne qualité. Les expérimentations précédentes ont permis de montrer qu'une approche KNN peut trouver des voisins pour les GSU, mais ces voisins ne permettent pas de fournir des recommandations de bonne qualité à ces derniers. Les voisins ne sont donc pas suffisamment fiables. Nous pensons que redéfinir le voisinage nous permettra d'augmenter le nombre de voisins fiables (et donc la qualité des recommandations) trouvés pour les GSU.

4.2 Voisinage d'un GSU dans une approche KNN

A la vue des éléments relatifs aux voisinages des GSU et aux performances des différentes mesures de similarité présentées dans la section précédente, nous avons choisi de nous intéresser, dans cette section, à l'impact de mesures de sélection du voisinage dédiées, sur la qualité des recommandations apportées aux GSU.

Rappelons que les méthodes à base de voisinage ont l'inconvénient de ne pas passer à l'échelle. Nous avons donc à nouveau utilisé le jeu de données MovieLens100K pour évaluer l'impact de la sélection des voisins sur les recommandations faites au GSU. Nous avons une fois encore utilisé la mesure de *VraisemblanceID* pour identifier les 10% d'utilisateurs aux préférences les plus spécifiques (les GSU) sur MovieLens100K.

4.2.1 L'utilisation de la dissimilarité

Aucune des mesures de similarité ne permettant d'améliorer significativement la qualité des recommandations fournies aux GSU, nous souhaitons étudier si l'information ajoutée par les

préférences qui distinguent les utilisateurs entre eux permet d'améliorer la qualité des recommandations apportées aux GSU. Nous avons ainsi choisi de mesurer la différence entre deux utilisateurs au niveau de leurs préférences sous la forme d'une dissimilarité, car il s'agit de la mesure la plus proposée dans la littérature. La dissimilarité sera utilisée en complément de la similarité lors de l'estimation d'une prédiction.

L'état de l'art a déjà proposé d'exploiter la dissimilarité entre utilisateurs pour augmenter la précision des recommandations d'une méthode des k plus proches voisins [Zeng et al., 2010]. La dissimilarité est une valeur positive entre 0 et 1. Plus la valeur est proche de 1, plus les données sont dissimilaires. La dissimilarité entre deux utilisateurs peut être calculée de plusieurs manières. Nous avons retenu trois manières :

La dissimilarité $DCouv$ (équation 4.1) [Zeng et al., 2010] représente la proportion de notes que deux utilisateurs ne partagent pas. Cette mesure permet d'estimer à quel point deux utilisateurs consultent des ressources différentes, les utilisateurs les plus dissimilaires au sens de cette mesure étant ceux qui n'ont co-voté aucune ressource.

$$DCouv(u_1, u_2) = \frac{|N_{u_1} \cup N_{u_2}| - |N_{u_1} \cap N_{u_2}|}{|N_{u_1} \cup N_{u_2}|} \tag{4.1}$$

avec N_{u_i} représentant l'ensemble des ressources sur lesquelles des préférences sont exprimées par l'utilisateur u_i.

Nous proposons la dissimilarité $DPearson$ (équation 4.2) représentant le lien d'opposition qui peut exister entre les préférences de deux utilisateurs. Par exemple, si un utilisateur apprécie systématiquement les ressources qu'un autre utilisateur n'apprécie pas, alors les utilisateurs sont dissimilaires au sens de cette mesure. Lorsque le coefficient de corrélation de Pearson est utilisé pour calculer les similarités entre utilisateurs, les valeurs négatives (anti-corrélations) ne sont pas utilisées pour la prédiction de notes. Nous proposons ici d'exploiter ces anti-corrélations sur les GSU. Les utilisateurs doivent donc avoir co-voté des ressources pour que $DPearson$ puisse évaluer leur dissimilarité.

$$DPearson(u_1, u_2) = -1 * MIN(0, Pearson(u_1, u_2)) \tag{4.2}$$

avec $Pearson(u_1, u_2)$ correspondant à l'équation (2.1.1.2).

Enfin, nous proposons d'adapter la mesure $DBrayCurtis$ (équation 4.3), classiquement exploitée dans le domaine des sciences de la vie et de la terre pour comparer des échantillons organiques [Bray and Curtis, 1957].

$$DBrayCurtis(u_1, u_2) = 1 - \frac{2 * \sum_{r \in N_{u_1 u_2}} MIN(n_{u_1 r}, n_{u_2 r})}{\sum_{r \in N_{u_1 u_2}} (n_{u_1 r} + n_{u_2 r})} \tag{4.3}$$

avec $N_{u_1 u_2}$ l'ensemble des ressources co-votées pour u_1 et u_2. Cette mesure a été proposée à l'origine pour calculer la dissimilarité entre deux échantillons, en fonction de l'abondance des espèces présentes dans ces derniers. Cette mesure compare deux vecteurs dont les éléments représentent les concentrations de chaque espèce. En recommandation sociale, pour comparer deux utilisateurs, les vecteurs composés de leurs notes sur les ressources qu'ils ont co-votées sont classiquement exploités. Intuitivement, la mesure de dissimilarité de Bray-Curtis mesure la différence entre deux échantillons (deux utilisateurs) composés de différents niveaux d'importance des espèces qui le composent (les notes sur les ressources).

Notons Sim_{u_1,u_2} la similarité entre deux utilisateurs et $DSim_{u_1,u_2}$ la dissimilarité entre ces même utilisateurs. Nous proposons d'intégrer la dissimilarité à la formule de prédiction classique

de l'approche KNN (voir équation 2.1). Pour évaluer l'apport de la dissimilarité sur la qualité des recommandations des GSU (par rapport à une méthode classique exploitant uniquement la similarité), nous étudions la combinaison linéaire des utilisateurs similaires et dissimilaires pour le calcul de la prédiction [Zeng et al., 2010]. Nous analysons l'impact du poids de la dissimilarité β sur la qualité des recommandations fournies aux GSU.

Le calcul d'une prédiction pour un utilisateur u sur la ressource r suit alors l'équation (4.4).

$$\begin{aligned} Prédiction_{u,r} = \overline{n}_u + & \sum_{v \in V+_{u,r}} (n_{v,r} - \overline{n}_v) * Sim(u,v) \\ & + \beta * \sum_{v \in V-_{u,r}} (n_{v,r} - \overline{n}_v) * DSim(u,v), \end{aligned} \tag{4.4}$$

avec $V+_{u,r}$ et $V-_{u,r}$ les ensembles de voisins similaires et dissimilaires respectivement.

Les expérimentations sont réalisées avec une approche KNN, avec $k = 20$ et le coefficient de corrélation de Pearson pour mesurer les similarités entre les utilisateurs. Nous faisons varier β de -1 à 1 (par pas de 0,1) et nous présentons la RMSE moyenne obtenue sur les GSU sur la figure 4.4 avec les trois mesures de dissimilarité introduites précédemment. Les résultats obtenus lorsque $\beta = 0$ correspondent aux résultats d'une approche KNN classique sans l'utilisation de la dissimilarité.

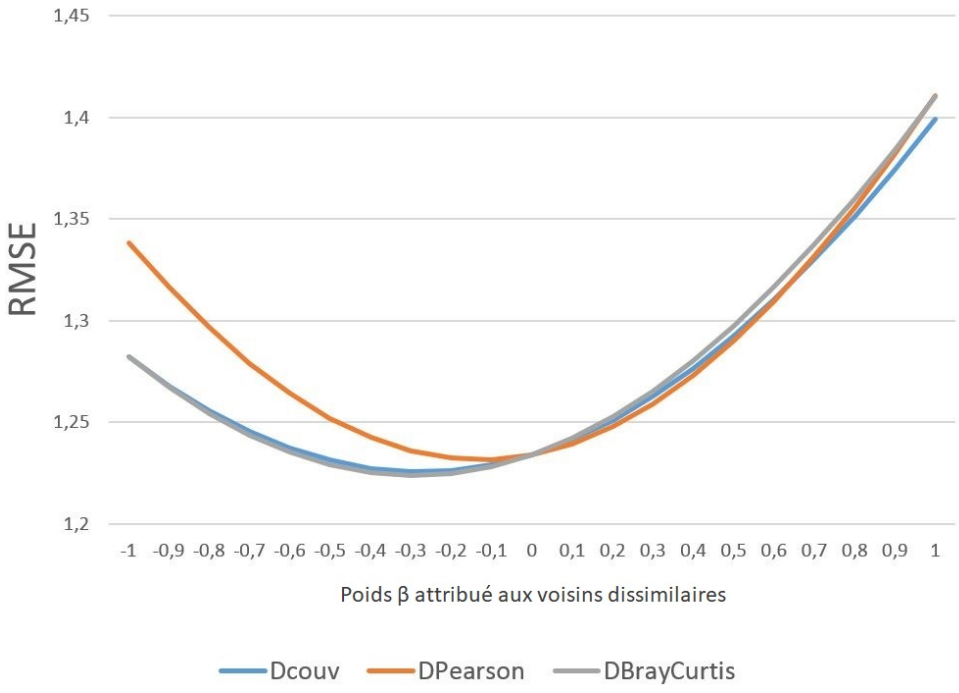


FIGURE 4.4 – RMSE des GSU en fonction de l'impact de la dissimilarité sur les recommandations

L'utilisation de la dissimilarité DCouv permet d'améliorer légèrement les résultats obtenus sur les GSU lorsque le poids β de la dissimilarité est de -0,3. Cela correspond à une formule de prédiction reposant pour 77% sur la similarité et 23% sur la dissimilarité. La dissimilarité de Bray-Curtis ne permet pas d'améliorer les performances de DCouv.

Cependant, l'amélioration maximale apportée par l'exploitation de la dissimilarité est seulement de 0,8% pour la RMSE observée sur les GSU (1,22 contre 1,23 sans utiliser la dissimilarité),

ce qui n'est pas significatif. Nous pouvons conclure qu'exploiter la dissimilarité ne permet pas d'apporter des recommandations de bonne qualité aux GSU, ni même d'améliorer significativement la qualité de ces recommandations.

4.2.2 Les utilisateurs pivots

Comme nous l'avons mentionné précédemment, nous avons l'intuition que la modélisation des GSU est d'avantage sensible au manque de données que celle des non-GSU. Le système ne peut pas trouver des voisins aux préférences suffisamment similaires aux GSU. Nous avons donc cherché à minimiser l'impact de ce manque de données, en augmentant artificiellement la couverture du voisinage des GSU à l'aide d'une idée très simple : " les amis de mes amis sont mes amis ".

En effet, si un GSU x pouvait bénéficier des préférences d'un utilisateur z même s'ils n'ont pas co-voté suffisamment de ressources, alors la qualité des recommandations de x pourrait être augmentée. Si un utilisateur y est similaire à x et que l'utilisateur y est également similaire à z, par transitivité, une similarité entre x et z peut être inférée. Nous pensons qu'il est plus fiable d'inférer une similarité entre deux utilisateurs très similaires à l'utilisateur y.

Pour mettre en pratique cette idée, il est d'abord nécessaire de répondre à deux questions :

— Quelle est la similarité minimale entre y et x, ainsi qu'entre y et z pour qu'une similarité entre x et z puisse être inférée ?

— Comment estimer la similarité entre x et z au travers de y ?

Nous appelons l'utilisateur y un utilisateur pivot, car il va permettre de faire le lien entre deux utilisateurs dont les préférences ne sont pas directement comparables. Nous considérons que deux utilisateurs x et z dont la similarité [24] avec un même troisième utilisateur y est supérieure à δ, alors la similarité entre x et z ne doit pas être supérieure au minimum des similarités entre x et y, et, y et z. Nous choisissons d'utiliser ce minimum pour maximiser l'impact de l'utilisation des utilisateurs pivots sur la qualité des recommandations fournies aux GSU. La similarité entre x et z peut alors être calculée selon l'équation 4.5.

$$Sim_y(x,z) = MIN(Sim(x,y) > \delta, Sim(y,z) > \delta) \tag{4.5}$$

Pour schématiser l'utilisation des utilisateurs pivots, la figure 4.5 montre un cas simple dans lequel l'utilisateur y serait un utilisateur pivot permettant de faire le lien entre l'utilisateur x et l'utilisateur z. La seule condition nécessaire pour permettre le rassemblement de deux utilisateurs x et z est que leurs similarités avec y soient supérieures à un seuil δ.

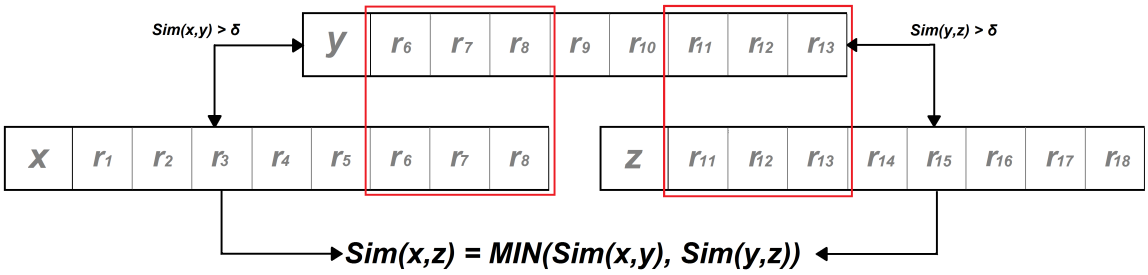


FIGURE 4.5 – Illustration de l'utilisation d'un utilisateur pivot

24. Par exemple : Pearson, Cosinus, ...

Dans un SR à base de KNN classique, seuls les liens entre les utilisateurs x et y **ou** y et z seraient exploités lors de la recommandation. Le système serait donc en mesure d'inférer les préférences de l'utilisateur x sur les ressources r_9 à r_{13}, mais n'aurait alors aucun moyen d'inférer les préférences de x sur les ressources r_{14} à r_{16}. D'après la définition originale du problème des GSU [Claypool et al., 1999], plus on augmente le nombre d'utilisateurs, plus on diminue la probabilité que les préférences d'un utilisateur soient spécifiques. L'utilisation des utilisateurs pivots pourrait ainsi permettre de trouver des utilisateurs partageant les mêmes préférences que celles d'un GSU mais qui n'ont pas co-voté suffisamment de ressources avec ce GSU. Nous pensons que grâce à cette nouvelle manière de calculer des similarités entre les utilisateurs, nous allons améliorer la probabilité de trouver des voisins fiables pour les GSU.

En pratique, plusieurs utilisateurs peuvent jouer le rôle d'utilisateur pivot (y), permettant de faire le lien entre deux utilisateurs (x et z). Nous choisissons de nous baser sur la moyenne des similarités obtenues avec l'équation (4.5) calculée avec tous les pivots, selon l'équation (4.6). Les expérimentations sont réalisées à l'aide d'une approche KNN, avec $k = 20$.

$$Sim(x, z) = \frac{\sum_{y \in P} Sim_y(x, z)}{|P|}, \tag{4.6}$$

avec P l'ensemble des utilisateurs pivots permettant de faire le lien entre les préférences des utilisateurs x et z.

Pour mesurer la similarité entre deux utilisateurs, nous avons utilisé la mesure cosinus qui permet de fournir les recommandations les plus précises aux GSU. Bien que cette mesure ne permet pas d'apporter des recommandations de bonnes qualité aux GSU, elle reste néanmoins la mesure la mieux adaptée pour l'identification de voisins fiables pour un GSU dans un jeu de données creux.

La figure 4.6 illustre l'erreur obtenue grâce à l'utilisation des utilisateurs pivots, sur les utilisateurs GSU et les utilisateurs non-GSU, en fonction du seuil δ utilisé pour l'acceptation des utilisateurs pivots. Lorsque $\delta = 1$, les résultats correspondent à ceux d'un modèle standard.

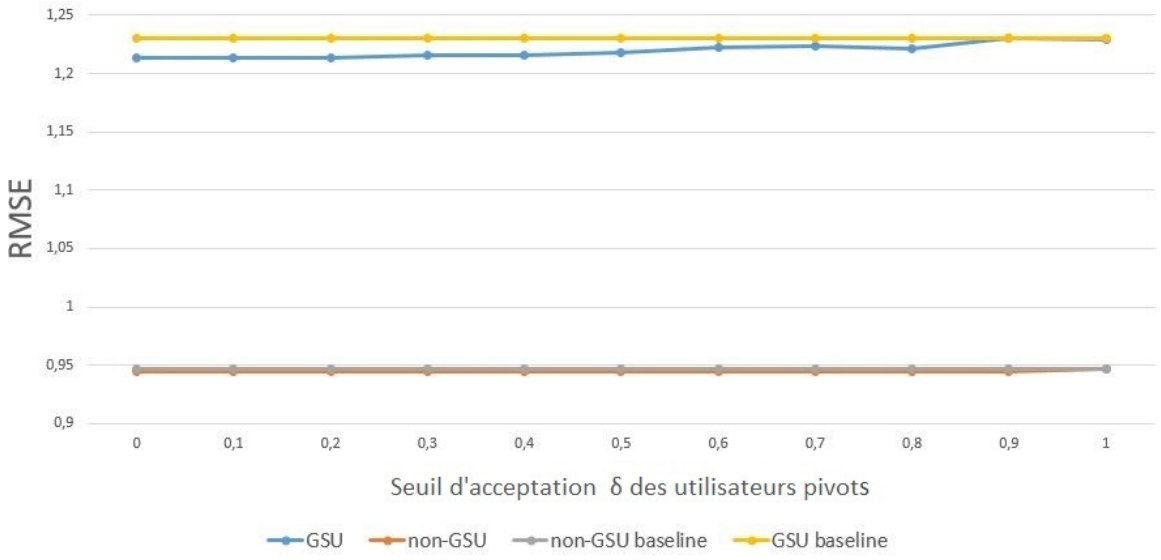


FIGURE 4.6 – Evolution de la RMSE des utilisateur non-GSU et des GSU en fonction de δ

On peut voir que l'utilisation des utilisateurs pivots ne permet pas d'améliorer les recomman-

dations apportées aux utilisateurs non-GSU (non-GSU et non-GSU *baseline*). Cela ne dégrade pas non plus les résultats, on peut donc dire que les utilisateurs pivots n'ont aucun impact sur ces utilisateurs. Une explication réside dans le fait que les utilisateurs non-GSU ont déjà un grand nombre de voisins très similaires et que les voisins ajoutés avec cette méthode ne font que très rarement partie des k plus proches voisins.

Les GSU profitent très légèrement de l'ajout de nouveaux voisins. Plus δ est petit (et donc plus on ajoute de nouveaux voisins aux GSU), plus l'erreur est faible. On parvient ainsi à réduire l'erreur de 2% sur les GSU lorsque $\delta = 0$ (1,21 avec les utilisateurs pivots contre 1,23 avec une approche classique), c'est-à-dire que l'on ajoute le maximum de nouveaux voisins aux GSU. L'erreur reste néanmoins très élevée puisqu'une RMSE de $1,21$ signifie que l'algorithme de recommandation se trompe de plus d'un point en moyenne sur les prédictions des GSU.

Dans cette section, nous avons étudié l'impact de l'utilisation de la dissimilarité et des utilisateurs pivots sur le renforcement du voisinage des GSU. Nous observons que l'ajout de voisins a un impact positif sur les recommandations fournies aux GSU. Il existe donc un lien entre la fiabilité du voisinage des GSU et les mauvaises recommandations qu'ils reçoivent. Les GSU que nous avons identifiés correspondent donc bien à notre définition du problème (voir définition 2.1.3.3), leurs voisins ne permettent pas de représenter leurs préférences.

Malgré les solutions innovantes que j'ai proposées, je n'ai pas réussi à améliorer de façon significative la fiabilité du voisinage d'un GSU pour qu'il reçoive des recommandations de bonne qualité. Plusieurs raisons peuvent être à l'origine de ce résultat. La première est le faible nombre d'utilisateurs du jeu de données MovieLens100K. En effet, plus le nombre d'utilisateurs est grand, moins il est probable qu'un utilisateur possède des préférences très spécifiques. Nous proposons donc de modéliser les GSU identifiés dans MovieLens20M dans la suite de ce chapitre. La seconde raison est l'importance de la moyenne de l'utilisateur dans un approche KNN. Je pense que l'utilisation de la moyenne des préférences des GSU n'a pas de sens puisque l'écart-type de leurs notes est très élevé (voir tableau 4.1).

Seules les approches à base de modèle sont capables de passer à l'échelle pour effectuer des recommandations sur le jeu de données MovieLens20M. Nous avons décidé d'utiliser l'approche par factorisation de matrice qui est la plus performante de l'état de l'art. La factorisation de matrice se concentre sur les interactions entre chaque utilisateur et chaque ressource, et permet donc d'extraire plus d'informations des données d'apprentissage que l'utilisation de la similarité entre les utilisateurs. De plus, lorsqu'elle exploite les biais des utilisateurs et des ressources elle peut s'adapter au cas particulier des GSU et ainsi permettre une modélisation précise de leurs préférences.

4.3 Modélisation des GSU dans une approche par factorisation de matrice

Pour apporter une solution au problème des Grey Sheep Users par une approche de FM, nous avons imaginé un système de recommandation basé sur l'exploitation conjointe de deux modèles de FM. Le premier modèle est un modèle standard de factorisation de matrice, dédié à la modélisation des utilisateurs non-GSU. L'état de l'art présente des performances acceptables sur ce sous-ensemble de la population, mais ce n'est pas le centre d'intérêt de nos recherches. Pour le second modèle, destiné à modéliser les GSU, nous proposons d'utiliser une méthode d'apprentissage adaptée aux spécificités de ces utilisateurs. Puisque la factorisation de matrice à montré qu'elle est la méthode de filtrage collaboratif la plus précise en moyenne [Koren et al., 2009],

nous nous sommes basés sur cette approche pour proposer un modèle dédié aux GSU. Il s'agit, à notre connaissance, de la première tentative de modélisation des GSU avec une approche à base de factorisation de matrice.

Par définition, les GSU sont rares, il y a donc bien plus d'utilisateurs non-GSU que d'utilisateurs GSU. Puisque le critère de convergence d'une FM est l'optimisation de la qualité globale des recommandations (sur l'ensemble des utilisateurs GSU et non-GSU), la précision sur les GSU n'impacte pas, ou en tout cas que très peu, la précision globale du modèle. Le modèle résultant va donc plus facilement représenter les préférences des utilisateurs non-GSU, indépendamment des résultats qu'il permet d'obtenir sur les GSU.

Les techniques de factorisation SGD et ALS sont les plus populaires à l'heure actuelle pour factoriser une matrice de notes [Yu et al., 2014]. Les modèles à base de FM, que nous souhaitons proposer, peuvent être utilisés avec n'importe quelle technique de factorisation. Nous avons choisi d'utiliser la technique SGD pour sa rapidité d'exécution et sa forte popularité actuelle, notamment grâce à son application aux réseaux de neurones. De plus, la technique SGD fournit en général des recommandations de meilleure qualité que la technique ALS [Yu et al., 2014].

Chacun des modèles proposés sera alors basé sur l'algorithme 1 présenté ci-après qui est un algorithme de SGD tenant compte des biais sur les utilisateurs et les ressources [Koren et al., 2009], et utilise la régularisation L_2. Cet algorithme parcourt chaque note $n_{u,r}$ du jeu de données d'apprentissage et calcule l'erreur de prédiction entre $n_{u,r}$ (la note réelle) et $(b_{u,r} + p_u q_r^T)$ (la note prédite). Le terme $b_{u,r}$ représente le biais qui s'applique sur la prédiction de la note de l'utilisateur u sur la ressource r, tel que décrit par l'équation 2.4. L'erreur ainsi calculée permet de mettre à jour le biais de l'utilisateur, le biais de la ressource et les caractéristiques latentes des vecteurs P_u et Q_r.

Algorithm 1 Descente de gradient stochastique - Régularisation L_2

Entrées :
$N = \{n_{u,r}\}$ - ensemble des notes des utilisateurs sur les ressources,
k - nombre de caractéristiques latentes du modèle,
α - taux d'apprentissage,
λ - paramètre de régularisation

Sortie :
Matrice de caractéristiques latentes P et Q

procedure SGDL2(N, k, α, λ)
  Initialisation au hasard des matrices P et Q
  while *non Convergence* **do**
    for *chaque* $n_{u,r} \in N$ **do**
      $e_{u,r} = (n_{u,r} - (b_{u,r} + p_u q_r^T))$ // **Estimation de l'erreur**
      $b_u \leftarrow b_u + \alpha * e_{u,r}$
      $b_r \leftarrow b_r + \alpha * e_{u,r}$
      $Q_r \leftarrow Q_r + \alpha * (\mathbf{e}_{u,r}.P_u - \lambda * Q_r)$
      $P_u \leftarrow P_u + \alpha * (\mathbf{e}_{u,r}.Q_r - \lambda * P_u)$
  retourner P **et** Q

Convergence est un booléen qui représente le critère d'arrêt de l'algorithme d'apprentissage. Deux scénarios sont alors possibles, soit l'algorithme atteint un nombre maximum d'itérations, soit l'erreur globale se stabilise.

La régularisation L_1 est réputée plus adaptée pour représenter des données creuses. Cette

régularisation a longtemps été utilisée de manière naïve dans les implémentations de SGD, n'utilisant que le signe du facteur latent, comme le suggère l'équation 4.7 de mise à jour des poids.

$$Q_r \leftarrow Q_r + \alpha * (e_{u,r} * P_u - \lambda.signe(Q_r)) \tag{4.7}$$

Cette implémentation est dite naïve car la régularisation L_1 ne devrait pas être à l'origine d'un changement de signe du facteur Q_r [Tsuruoka et al., 2009]. La solution apportée à ce problème tient dans l'algorithme 2. Nous comparerons ces deux approches de régularisation et leur impact sur les recommandations proposées aux GSU.

Algorithm 2 Descente de gradient stochastique - Régularisation L_1

procedure SGDL1(N, k, α, λ)
  Initialisation au hasard des matrices P et Q
  while *non Convergence* **do**
    for *chaque* $n_{u,r} \in N$ **do**
      $\mathrm{e}_{u,r} = (n_{u,r} - (b_{u,r} + p_u q_r^T))$ // **Estimation de l'erreur**
      $b_u \leftarrow b_u + \alpha * e_{u,r}$
      $b_r \leftarrow b_r + \alpha * e_{u,r}$
      $Q_r \leftarrow Q_r + \alpha * \mathbf{e}_{u,r}$
      if $q_r > 0$ **then**
        $Q_r \leftarrow MAX(0, Q_r - \lambda * \alpha)$
      else if $Q_r < 0$ **then**
        $Q_r \leftarrow MIN(0, Q_r - \lambda * \alpha)$
      $P_u \leftarrow P_u + \alpha * \mathbf{e}_{u,r}$
      if $P_u > 0$ **then**
        $P_u \leftarrow MAX(0, P_u - \lambda * \alpha)$
      else if $P_u < 0$ **then**
        $P_u \leftarrow MIN(0, P_u - \lambda * \alpha)$
  retourner P **et** Q

Rappelons que l'identification des GSU est réalisée en amont de la phase d'apprentissage des modèles et donc en amont de la recommandation. A l'issue de cette étape, chaque utilisateur peut donc être considéré, soit comme un GSU, soit comme un non-GSU. Le calcul des recommandations pour un utilisateur nécessite de connaître la catégorie à laquelle il appartient (GSU ou non-GSU), pour ensuite pouvoir utiliser le modèle adéquat. Comme précédemment, l'identification de la catégorie de chaque utilisateur est réalisée à l'aide de la mesure *VraisemblanceID* puisqu'il s'agit de la mesure la plus performante pour l'identification des utilisateurs aux préférences spécifiques.

4.3.1 Le modèle GSUOnly

Le premier modèle que nous avons conçu part de l'hypothèse que les GSU peuvent bénéficier des préférences des autres GSU uniquement, et uniquement de leurs préférences. La seule différence avec le modèle standard de l'état de l'art est que ce modèle est appris sur une sous-partie de la matrice de notes, rassemblant les préférences des GSU ($Train_{GSU}$). Nous avons écarté les préférences des utilisateurs non-GSU. Le temps de calcul de ce modèle est donc très largement inférieur à celui du modèle incluant la population entière des utilisateurs. Nous faisons référence à ce modèle sous le nom *GSUOnly*.

Ce modèle va donc apprendre précisément les préférences des GSU, et ainsi confirmer si la présence des non-GSU, de par leur nombre, n'empêcheraient pas l'algorithme d'apprendre les préférences spécifiques des GSU.

4.3.2 Le modèle WeightedGSU

Le second modèle, $WeightedGSU$, part du principe que les non-GSU doivent être présents lors de l'apprentissage, mais qu'il faut réduire leur impact sur l'erreur globale de la factorisation pour donner d'avantage d'importance aux préférences des GSU. Pour cela, ce modèle pondère l'erreur à l'apprentissage, en fonction de la catégorie à laquelle appartient l'utilisateur. Un algorithme standard accorde le même poids à tous les utilisateurs (GSU et non-GSU). Soit W_{GSU} le poids associé aux GSU et $W_{non-GSU}$ le poids associé aux non-GSU. L'équation (4.8) présente la relation entre ces deux poids.

$$W_{GSU} + W_{non-GSU} = 1.0 \tag{4.8}$$

Nous sommes convaincus que le poids des GSU doit être plus important que celui des utilisateurs non-GSU. De cette manière, les spécificités des notes des GSU seront d'avantage prises en compte, tout en continuant à utiliser les notes des utilisateurs non-GSU. Les notes des utilisateurs non-GSU sont une importante source d'informations supplémentaire pour la phase d'apprentissage et sont une manière de traiter le problème du manque de données. En effet, si on réduit le nombre d'utilisateurs, on diminue nécessairement le nombre de notes par ressource, ce qui entraîne un problème de démarrage à froid sur les ressources.

Nous souhaitons déterminer W_{GSU} de façon à ce qu'il maximise la qualité des recommandations fournies aux GSU. $WeightedGSU$ s'appuie donc sur les algorithmes 1 et 2, que nous avons modifiés, pour prendre en compte le poids des utilisateurs dans le calcul de l'erreur. L'étape d'estimation de l'erreur est donc modifiée suivant l'équation (4.9).

$$e_{u,r} = \begin{cases} W_{GSU} * (n_{u,r} - (b_{u,r} + p_u q_r^T)) & \text{si u est un GSU} \\ W_{non-GSU} * (n_{u,r} - (b_{u,r} + p_u q_r^T)) & \text{si u n'est pas un GSU} \end{cases} \tag{4.9}$$

4.3.3 Le modèle SingleGSU

Le dernier modèle que nous proposons part de deux postulats : (1) les GSU perturbent l'apprentissage des préférences des autres GSU, et (2) le nombre de non-GSU est trop important. Un modèle est alors calculé pour chaque GSU, à partir de ses préférences et de celles d'une sous-partie des non-GSU. L'idée nous vient de [Perlich, 2016] qui propose de modéliser chaque dimension du problème séparément. Nous considérons ici que chaque GSU est un problème distinct. Cette approche nous permet d'évaluer l'impact des préférences des autres GSU sur la qualité des recommandations fournies à un GSU donné.

Ce modèle, appelé $SingleGSU$, est appris sur les préférences d'un seul GSU (celui pour lequel le modèle est destiné) et sur les préférences d'utilisateurs supplémentaires. Nous choisissons de sélectionner les utilisateurs supplémentaires parmi l'ensemble des non-GSU, pour étudier l'influence de groupes d'utilisateurs non-GSU sur les recommandations fournies aux GSU. Nous excluons les préférences des autres GSU, qui pourraient impacter négativement la qualité du modèle [25]. Le nombre d'utilisateurs supplémentaires sélectionnés ainsi que la manière dont ils sont choisis doit être étudié.

25. Si les utilisateurs GSU sont très différents entre eux par exemple.

Notons que ce modèle est plus une preuve de concept qu'un modèle qui peut être utilisé sur de vrais jeux de données. En effet, il n'est pas acceptable d'imaginer calculer et gérer un modèle par GSU dans un contexte où la quantité de données explose. Cependant, ce modèle reste une manière d'étudier en profondeur la capacité d'une méthode de FM à modéliser précisément les préférences des GSU.

Nous venons donc de présenter trois approches de modélisation des GSU basées sur l'approche SGD. Nous avons implémenté ces approches pour évaluer la qualité de la modélisation des GSU qu'ils permettent d'obtenir.

4.3.4 Expérimentations

Cette section présente les expérimentations que nous avons menées pour étudier les capacités des modèles à base de FM que nous avons proposés, à fournir des recommandations de bonne qualité aux GSU.

4.3.4.1 Le protocole d'évaluation

Les approches par factorisation de matrice sont moins sensibles au problème du passage à l'échelle. Cette évaluation a donc été exécutée sur le plus important jeu de données à notre disposition : $MovieLens20M$. Rappelons que ce jeu de données est composé de 19,6 millions de préférences (98% du nombre total de préférences) exprimées par 123 053 utilisateurs (88,8% du nombre total des utilisateurs). Ce jeu de données est particulièrement très peu rempli avec moins de 1% de préférences renseignées.

Les modèles que nous étudions sont évalués sur la qualité des recommandations qu'il fournissent au travers de la RMSE que nous avons déjà présentée et exploitée dans les expérimentations précédentes.

Comme dans le chapitre précédent, les données sont séparées en deux sous-ensembles : les données d'apprentissage $Train_{GSU}$ et $Train_{non-GSU}$ (80%) et les données de test $Test_{GSU}$ et $Test_{non-GSU}$ (20%). Les préférences de test seront utilisées pour évaluer la précision des modèles.

La précision des modèles proposés est comparée à celle des méthodes standards de FM. Les techniques ALS et SGD sont à nouveau étudiées pour connaître l'impact de ces deux techniques de FM sur la qualité des recommandations fournies aux GSU. Puisque l'optimisation des paramètres de ces techniques de FM, tels que le taux d'apprentissage ou le taux de régularisation, ne sont pas l'objectif de notre travail, nous avons utilisé les valeurs de l'état de l'art pour le jeu de données utilisés [Yu et al., 2014]. Les mêmes paramètres seront utilisés pour tous les modèles présentés dans cette section.

Nous utilisons donc 20 caractéristiques latentes apprises avec un taux d'apprentissage de 0,001 et un taux de régularisation fixé à 0,02 (voir tableau 3.1). Les matrices P et Q sont initialisées aléatoirement. Pour obtenir des résultats comparables, nous remplissons aléatoirement les matrices P et Q une seule fois, et utilisons le même état initial pour chacune des méthodes que nous présentons ensuite. Une dizaine d'exécutions de cette étape seront réalisées pour éviter le biais d'une mauvaise initialisation aléatoire. Le critère d'arrêt utilisé pour les algorithmes d'apprentissage est la convergence de l'erreur sur les GSU. En pratique, si on définit e^t_{glob} l'erreur globale du modèle à l'étape t de l'apprentissage, alors le critère de convergence est défini par l'équation 4.10.

$$Convergence \leftarrow (e^{t-5}_{glob} - e^t_{glob} < 0,01) \tag{4.10}$$

Comme dans les expérimentations précédentes, l'identification des GSU a été réalisée à l'aide de la méthode de *VraisemblanceID*. Nous avons défini le nombre de GSU identifiés d'après les expériences menées dans le chapitre précédent. Pour garantir une précision de 90% de notre mesure d'identification, nous avons donc identifié 6% des utilisateurs possédant les plus faibles valeurs de *VraisemblanceID* pour former l'ensemble de GSU.

4.3.4.2 Les performances des modèles de l'état de l'art

Le tableau 4.3 présente les RMSE médianes obtenues sur les différents ensembles d'utilisateurs avec les approches à base de FM classiques de l'état de l'art. Ces résultats nous servent de point de référence pour l'analyse des expérimentations qui suivent.

Num. Ligne	Modèle	Régularisation	Données Test	RMSE Médiane
1	SGD	L_2	$Test$	0,80
2	SGD	L_1	$Test$	0,87
3	ALS	L_2	$Test$	0,82
4	SGD	L_2	$Test_{GSU}$	1,18
5	ALS	L_2	$Test_{GSU}$	1,28
6	SGD	L_1	$Test_{GSU}$	1,21
7	SGD	L_2	$Test_{non-GSU}$	0,78
8	SGD	L_1	$Test_{non-GSU}$	0,84

TABLE 4.3 – RMSE des modèles standards

Les trois premières lignes du tableau 4.3 présentent la RMSE obtenue à l'aide des deux méthodes de FM les plus populaires, ALS et SGD. Le modèle est entraîné sur les données d'apprentissage ($Train$) et ensuite testé sur l'ensemble des données de test ($Test$). Nous présentons les résultats obtenus pour la technique SGD avec deux types de régularisations (L_1 et L_2) ainsi que les résultats obtenus avec la technique ALS utilisée dans le chapitre précédent. La méthode SGD présente de meilleurs résultats que la méthode ALS (2,5% meilleurs pour la médiane), ce qui est cohérent avec les études de la littérature [Yu et al., 2014].

Les quatrième et septième lignes du tableau 4.3 présentent les RMSE des GSU ($Test_{GSU}$) et des utilisateurs non-GSU ($Test_{non-GSU}$) respectivement, obtenues à l'aide d'un modèle standard de SGD utilisé couramment dans les systèmes de recommandations. Avec une RMSE médiane de 1,18 sur les GSU et une RMSE médiane de 0,78 sur les utilisateurs non-GSU, la RMSE sur les GSU est 51% plus élevée que sur les utilisateurs non-GSU. La RMSE médiane de l'ensemble complet des utilisateurs ($Test$) est de 0,80, ce qui confirme que le faible nombre de GSU a peu d'impact sur la RMSE globale du système (environ 2,5% d'écart entre les lignes 1 et 7).

On peut également noter que l'approche SGD présente de meilleurs résultats sur les GSU ($Test_{GSU}$) que l'approche ALS (lignes 4 et 5 du tableau 4.3). Cette fois-ci, l'approche SGD est 8% plus précise que l'approche ALS pour modéliser les GSU. L'approche SGD est donc la plus adaptée pour notre travail.

Nous nous demandons enfin si la régularisation L_1 pourrait être un meilleur choix pour la modélisation des GSU, comme le suppose [Ng, 2004]. La régularisation L_1 est, en théorie, plus adaptée pour la factorisation de matrices creuses. Sur les GSU, la RMSE médiane obtenue avec un approche SGD et une régularisation L_1 est de 1,21, ce qui est 3% plus élevé que celle obtenue

avec la régularisation L_2. Nous pouvons conclure que l'utilisation du terme de régularisation L_1 n'est pas suffisante pour améliorer les recommandations fournies aux GSU.

L'analyse des modèles proposés a pour but d'identifier des clés permettant d'améliorer la modélisation des GSU.

4.3.4.3 Analyse des modèles proposés

Le modèle GSUOnly

Le modèle *GSUOnly* est appris sur le jeu de données $Train_{GSU}$ puis testé sur le jeu de données $Test_{GSU}$. Le tableau 4.4 montre les RMSE médianes obtenues en fonction du terme de régularisation utilisé.

La modélisation des GSU en utilisant uniquement les préférences des autres GSU mène à un modèle encore moins précis que le modèle standard entraîné sur la population entière des utilisateurs, avec une hausse de 8% de la RMSE médiane. Cette conclusion est cohérente avec une autre étude sur les préférences spécifiques de certains utilisateurs dans laquelle les auteurs séparent également les utilisateurs en deux groupes, les normaux et les inconsistants [Bellogín et al., 2014]. Ce papier montre en effet qu'utiliser uniquement les utilisateurs inconsistants pour effectuer des recommandations aux utilisateurs inconsistants entraine de mauvaises qualités de recommandations. Nous pensions que la manière dont nous avons défini les GSU, très différente de celle de [Bellogín et al., 2014], nous mènerait à des conclusions différentes.

Nous pouvons donc conclure que les GSU profitent des préférences des utilisateurs non-GSU lors de la phase d'apprentissage de l'approche à base de MF, puisque l'exclusion de ces utilisateurs lors de l'apprentissage du modèle entraîne une baisse de la qualité des recommandations fournies aux GSU. C'est également la conclusion de [Bellogín et al., 2014] : "les utilisateurs les moins cohérents ont besoin d'information venant de l'extérieur de leur propre cluster."

Modèle	Reg.	Apprentissage	Test	RMSE Médiane
SGD	L_2	$Train_{GSU}$	$Test_{GSU}$	1,27
SGD	L_1	$Train_{GSU}$	$Test_{GSU}$	1,29

Table 4.4 – RMSE obtenues avec le modèle *GSUOnly*

A titre indicatif, nous avons une nouvelle fois calculé les RMSE en utilisant à la fois la régularisation L_1 et la régularisation L_2. Bien que l'écart ne soit pas significatif (inférieur à 1%), la régularisation L_1 n'est encore une fois pas mieux adaptée à la modélisation des GSU puisqu'elle amoindrit légèrement les résultats. Nous pouvons donc confirmer que la régularisation L_2 est la technique de régularisation la plus efficace. Nous utiliserons la régularisation L_2 dans les expérimentations qui suivent.

Le modèle WeightedGSU

Les expérimentations menées sur le modèle *GSUOnly* montrent que n'utiliser que les GSU ne permet pas d'améliorer les recommandations qui leur sont proposées. Au contraire, on peut désormais affirmer qu'il est nécessaire d'utiliser les préférences des utilisateurs non-GSU pour effectuer une recommandation à un GSU. Nous pensons que le poids des GSU est bien trop faible pour permettre de les modéliser correctement. Donc nous allons étudier un modèle dont le

but est de moduler le poids des GSU et des utilisateurs non-GSU afin d'améliorer la qualité des recommandations fournies aux GSU : le modèle $WeightedGSU$.

La figure 4.7 présente l'évolution de la RMSE médiane sur les GSU en fonction du poids accordé aux GSU (W_{GSU}). W_{GSU} varie de 0,1 à 1,0 en suivant l'équation (4.8). Précisons que le cas où $W_{GSU} = 0$ ne peut pas être étudié puisque cela revient à ne pas utiliser les GSU dans la phase d'apprentissage, donc la matrice de facteurs latents P ne contiendrait pas les lignes correspondantes aux GSU et nous ne pourrions pas évaluer le modèle sur eux. A l'inverse, le cas où $W_{GSU} = 1,0$ représente le cas dans lequel les utilisateurs normaux sont ignorés, ce qui est équivalent au modèle $GSUOnly$, la RMSE médiane correspondante est donc 1,27. Un poids de 0,5 est équivalent à une exécution classique d'un algorithme de SGD puisque tous les utilisateurs (GSU et non-GSU) ont le même poids [26].

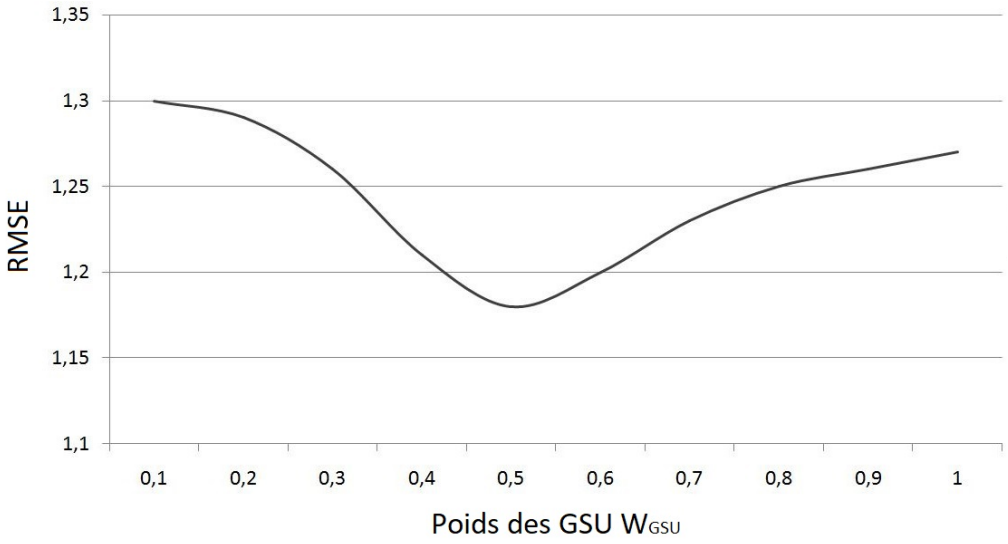


FIGURE 4.7 – RMSE médiane des GSU en fonction du poids W_{GSU} accordé à leurs préférences

Deux principales tendances se dégagent dans la figure 4.7. Dans le cas où $W_{GSU} < 0,5$, la RMSE médiane augmente avec la diminution du poids des GSU (jusqu'à 1,3). Plus l'impact des GSU est faible dans l'optimisation du modèle, plus la qualité des recommandations fournies aux GSU est faible, ce qui semble logique. Dans le cas où $W_{GSU} > 0,5$, la RMSE médiane augmente également avec le poids des GSU. Elle varie de 1,17 à 1,27. Ainsi, plus on donne d'importance aux préférences des GSU dans l'apprentissage du modèle, moins les recommandations qui leur sont fournies sont de qualité.

La valeur de RMSE médiane la plus faible est atteinte lorsque $W_{GSU} = 0,52$, ce qui est très proche du modèle standard de SGD. L'évolution de la RMSE médiane dans le cas où $W_{GSU} > 0,5$ n'est pas celle que nous attendions, nous pensions que donner plus de poids aux GSU pourrait améliorer la représentation de leurs préférences par le modèle ainsi obtenu.

Nous pensons que deux éléments impactent la qualité du modèle $WeightedGSU$. Le premier est la présence des autres GSU. Comme nous l'avons vu avec le modèle $GSUOnly$, la présence des autres GSU ne contribue pas à améliorer la qualité du modèle, il faut donc les séparer les uns des autres. Le second élément réside dans le nombre d'utilisateurs non-GSU. Nous sommes

26. En réalité, cela reste légèrement différent d'une exécution classique puisque toutes les erreurs sont divisées par 2, il est donc nécessaire de faire au moins le double d'itérations pour obtenir ce résultat.

face à une double conclusion ambiguë : les GSU sont nécessaires pour améliorer les résultats sur les GSU d'une part, et d'autre part lorsque l'on donne trop d'importance à ces utilisateurs, cela diminue la qualité des recommandations fournies aux GSU. Il ne faut donc peut-être pas diminuer le poids des utilisateurs normaux mais diminuer leur nombre pour simplifier l'extraction des caractéristiques latentes des GSU. Notre dernier modèle à été conçu dans ce but.

La modèle SingleGSU

Le dernier modèle que nous proposons, *SingleGSU*, apprend un modèle pour chaque GSU. Chaque modèle est appris sur les préférences de ce GSU ainsi que sur celles d'autres utilisateurs non-GSU. Nous avons choisi de ne pas exploiter les préférences des autres GSU en raison des faibles résultats obtenus avec le modèle *GSUOnly*. Nous avons donc choisi de les écarter. Deux questions se posent alors ici :

— Combien d'utilisateurs non-GSU doivent être utilisés pour apprendre le modèle d'un GSU ?

— Sur quel critère doit-on sélectionner ces utilisateurs non-GSU ?

Pour répondre à ces questions, dans une première expérimentation, nous avons fait évoluer le nombre d'utilisateurs non-GSU utilisés. Les utilisateurs les plus similaires au GSU auquel le modèle est dédié ont été privilégiés à l'aide du coefficient de corrélation de Pearson. Puisque nous considérons 6% des utilisateurs en tant que GSU (soit plus de 7 300 utilisateurs), le nombre de processus indépendants de factorisation de matrice à exécuter est trop important. Nous avons donc choisi d'identifier 300 GSU de manière aléatoire parmi l'ensemble total des GSU pour apprendre 300 modèles *SingleGSU* différents. Nous avons ajouté un filtre à l'identification aléatoire des GSU pour nous assurer que les *VraisemblanceID* des utilisateurs identifiés soient équitablement réparties de manière à bien représenter l'ensemble des GSU.

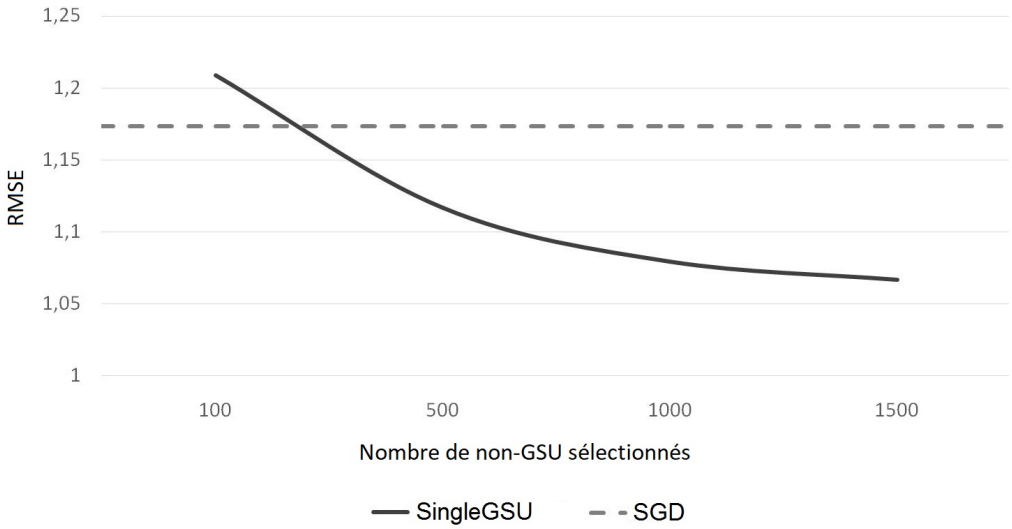


FIGURE 4.8 – RMSE médiane des GSU en fonction du nombre d'utilisateurs similaires utilisés

La figure 4.8 présente la RMSE médiane des 300 GSU en fonction du nombre d'utilisateurs similaires sélectionnés pour la phase d'apprentissage. Nous avons fait varier ce nombre de 100 à 1 500. Le graphique présente également en pointillés la RMSE standard obtenue sur les GSU (obtenue avec une approche SGD combinée à la régularisation L_2). Lorsque 100 utilisateurs

similaires sont sélectionnés, la RMSE médiane otenue avec la modèle *SingleGSU* est de 1,21, ce qui est supérieur à la RMSE des GSU avec un modèle standard. On peut donc conclure que 100 utilisateurs ne sont pas suffisants pour apporter de bonnes recommandations à un GSU.

Plus le nombre d'utilisateurs similaires augmente, plus la RMSE médiane obtenue baisse. Environ 250 utilisateurs non-GSU similaires suffisent à égaler les performances d'une approche standard sur les GSU. A partir d'environ 1 000 utilisateurs similaires sélectionnés, l'amélioration des résultats stagne, avec une RMSE médiane d'environ 1,07. Lorsque 1 500 utilisateurs non-GSU sont sélectionnés, la RMSE médiane est de 1,06, ce qui correspond à une amélioration de 9% de la qualité des recommandations fournies aux GSU, en comparaison d'une méthode standard. Évidemment, plus le nombre de non-GSU sélectionnés est important, plus le temps nécessaire pour apprendre le modèle d'un GSU est important. Le meilleur compromis entre le temps d'exécution et le résultat obtenu semble donc être aux alentours de 900 utilisateurs non-GSU sélectionnés, avec une RMSE médiane de 1,08 correspondant à une amélioration moyenne de 7,7% de la qualité des recommandations faites aux GSU. Ces premiers résultats représentent une amélioration réelle des résultats obtenus sur les GSU.

Pour aller plus loin nous avons étudié la répartition individuelle de ces améliorations. Premièrement, le modèle *SingleGSU* améliore la RMSE médiane de plus de 72% des 300 GSU identifiés. Si nous nous concentrons sur les améliorations les plus significatives, 54% de ces 300 GSU ont une amélioration de leur RMSE de plus de 10%. Cela signifie que le modèle *SingleGSU*, calculé à partir des préférences des non-GSU les plus similaires à celle du GSU, permet d'améliorer significativement la qualité des recommandations pour plus de la moitié des GSU.

De plus, 32% des GSU identifiés reçoivent ainsi des recommandations dont la qualité est d'au moins 20% supérieure et 20% des 300 GSU peuvent désormais recevoir des recommandations d'une qualité égale à celles destinées aux 50% des utilisateurs les mieux servis par le système, ce qui est une très grande avancée.

Nous n'avons pour le moment pas identifié de lien entre la *VraisemblanceID* des 300 GSU et la capacité ou non du modèle *SingleGSU* à améliorer les recommandations qui leurs sont fournies.

4.3.5 Analyse critique des résultats

Bien que ces résultats présentent une amélioration significative de la qualité des recommandations fournies à certains GSU, nous ne pouvons pas garantir des recommandations de qualité à l'ensemble des GSU.

Les 46% de GSU recevant encore des recommandations de mauvaise qualité, même avec cette première approche, ne sont pas simples à distinguer des autres GSU. Nous avons néanmoins voulu nous assurer qu'il est tout de même possible de leur fournir des recommandations de qualité en proposant de sélectionner les utilisateurs non-GSU de manière aléatoire en respectant la simple condition que cet utilisateur non-GSU doit posséder un minimum de ressources co-votées avec l'unique GSU. Pour permettre une comparaison avec les expériences précédentes, nous conservons 900 utilisateurs sélectionnés. Pour la sélection de ces 900 non-GSU, nous avons effectué 500 tirages aléatoires successifs pour chacun des 138 GSU (46%) recevant toujours des recommandations de mauvaise qualité. Pour 85 de ces 138 GSU (62%), nous avons pu trouver un groupe de 900 utilisateurs non-GSU permettant d'améliorer de plus de 10% la qualité des recommandations qui leur sont fournies. Cela montre que l'information nécessaire pour calculer un modèle précis des GSU est effectivement présente dans la matrice de notes de l'ensemble des utilisateurs, et qu'il est nécessaire d'isoler cette information pour qu'elle ne soit pas noyée dans la large masse de préférences du jeu de données complet.

A l'heure actuelle, nous n'avons pas pu identifier en amont de la modélisation, quels utilisateurs bénéficient plus facilement d'un type de sélection que d'autres. Cependant, nous savons désormais qu'il est possible de modéliser les GSU au travers d'une approche par modèle et que la difficulté se situe dans la sélection des données d'apprentissage.

Nous avons mentionné précédemment que cette approche serait difficile à mettre en place en situation réelle puisqu'il est extrêmement coûteux de calculer un modèle par utilisateur, mais il serait intéressant d'étudier la possibilité de rassembler des GSU qui ne partagent aucune préférence pour ainsi pouvoir calculer un seul modèle pour représenter plusieurs GSU, sans qu'ils ne viennent perturber les recommandations des autres. Si deux GSU ne partagent pas de co-votes, peut-être que les préférences spécifiques de l'un ne perturberont pas l'apprentissage des préférences spécifiques de l'autre. De plus, nous pensons que dans un future proche, l'accès aux informations pertinentes uniquement sera un enjeu tel pour les utilisateurs que le calcul de modèles sur mesure rentrera dans les mœurs et deviendra une norme. Chaque modèle représentera alors un utilisateur et sera parfaitement adapté à toutes les spécificités de l'utilisateur auquel le modèle est dédié.

4.4 Conclusion

Dans ce chapitre, nous avons étudié les caractéristiques des GSU, et notamment de leur voisinage, pour proposer des méthodes innovantes pour l'identification du voisinage d'un GSU. Nous avons montré que l'utilisation de la dissimilarité pour effectuer des recommandations à un GSU ne permet pas d'améliorer significativement la qualité des recommandations qui sont fournies. Une seconde méthode, dont l'objectif est d'estimer la similarité entre deux utilisateurs qui n'ont pas co-voté suffisamment de ressources pour qu'il soit possible de calculer directement leur similarité, a permis d'améliorer la qualité des recommandations d'avantage sur les GSU que sur les non-GSU. L'amélioration des résultats (2%) n'est pas suffisante pour fournir des recommandations de bonne qualité aux GSU.

Les méthodes d'apprentissage automatique dont l'objectif est de modéliser les données ont l'avantage d'extraire l'information contenue dans chaque interaction (préférence) entre un utilisateur et une ressource. Ces méthodes permettent l'exploitation du jeu de données MovieLens20M pour nos expérimentations. Nous nous sommes inspirés des techniques de factorisation de matrice, qui sont les plus efficaces dans le domaine de la recommandation sociale, pour modéliser les GSU. L'approche *GSUOnly* ne modélise que les GSU entre eux. Les résultats obtenus par ce modèle mettent en évidence les importants écarts de préférences au sein des GSU. Les GSU ne peuvent donc pas être utilisés pour recommander un GSU. L'information permettant de modéliser les GSU doit donc être trouvée parmi les préférences des non-GSU.

Nous avons proposé de ne pas sélectionner un sous-ensemble des utilisateurs pour l'apprentissage du modèle, mais de pondérer l'importance des préférences des deux catégories d'utilisateurs (GSU et non-GSU). L'approche $WeightedGSU$ modélise l'ensemble complet des utilisateurs en donnant plus de poids aux préférences des GSU. Les expérimentations menées sur cette approche ont montré qu'elle ne permet pas d'améliorer la qualité des recommandations fournies aux GSU pour deux raisons. La première est que lorsque trop d'importance est donnée aux préférences des GSU, comme pour le modèle *GSUOnly*, les préférences des GSU parasitent celles des autres GSU. La seconde raison est que, lorsque trop d'importance est donnée aux préférences des non-GSU, les spécificités des préférences des GSU ne sont plus suffisamment représentées par le modèle $WeightedGSU$.

Les deux premières approches de modélisation (*GSUOnly* et *WeightedGSU*) nous ont ainsi permis d'identifier les paramètres importants à prendre en considération pour permettre à un processus d'apprentissage par FM de conserver l'information pertinente concernant les préférences d'un GSU.

Pour cela, nous avons proposé le modèle *SingleGSU*, qui tient compte de l'influence négative des GSU, ainsi que du nombre important de non-GSU, sur la qualité de la modélisation des GSU. L'approche *SingleGSU* isole chaque GSU et le modélise à partir de ses préférences et de celles d'utilisateurs non-GSU sélectionnés à l'aide d'une heuristique. Cette approche permet d'améliorer significativement (par plus de 10%) la précision des recommandations fournies par un modèle standard de FM, pour plus de 82% des GSU identifiés.

Il est donc possible d'améliorer de plus de 10% la qualité des recommandations fournies à la majorité des GSU en sélectionnant les utilisateurs desquels on souhaite les faire bénéficier lors de la phase d'apprentissage du modèle. Les informations concernant les préférences des GSU sont donc modélisables au travers d'une approche de FM, mais il faut sélectionner précisément les données d'apprentissage de chaque modèle pour ne pas noyer l'information importante dans l'ensemble global des préférences du jeux de données.

Ce travail ouvre un grand nombre de perspectives que nous détaillerons en conclusion de ce manuscrit.

Chapitre 5

Conclusions et Perspectives

Ce travail porte sur l'identification et la modélisation des GSU dans les systèmes de recommandation à base de filtrage collaboratif. Les GSU sont des utilisateurs dont les préférences ne sont en accord ou en désaccord avec celles d'aucune communauté d'utilisateurs, en raison du nombre de préférences spécifiques qu'ils ont exprimées. Les spécificités des GSU empêchent les systèmes de recommandation sociale classiques de leur fournir des recommandations de bonne qualité. L'état de l'art met en évidence la vue partielle des travaux actuels sur le problème des GSU. En effet, la majeure partie des travaux considère qu'après avoir identifié les GSU, les GSU doivent être écarter de la recommandation et ne doivent pas recevoir le service offert aux autres utilisateurs. Dans ce travail, nous avons refusé de considérer qu'un GSU ne peut pas bénéficier des informations issues des préférences des autres utilisateurs du système. Pour améliorer la qualité des recommandations fournies aux GSU, nous avons distingué deux grandes questions autour desquelles s'articule notre recherche. La première question est, comment peut-on identifier des GSU dans des données de préférences ? Nous nous sommes inspirés à la fois des sciences humaines et du domaine du *data mining* pour proposer des mesures d'identification des GSU. Ensuite, la deuxième question est, comment modéliser les GSU, à l'aide d'une approche de filtrage collaboratif, pour améliorer la qualité des recommandations qui leur sont fournies ? L'analyse axiomatique du fonctionnement d'un algorithme de filtrage collaboratif proposée par [Pennock and Horvitz, 1999] a mis en évidence les points à respecter par une approche de modélisation qui serait dédiée aux GSU. Nous avons donc proposé des méthodes de modélisation des GSU qui permettent d'améliorer la qualité des recommandations qui leur sont fournies.

La première partie de ce travail concerne donc l'identification des GSU dans des données de préférences. Nous avons remarqué que la tâche bien connue de la détection d'*outliers* a un objectif très similaire et que les techniques sont applicables à l'identification des GSU, même si ces deux domaines ont rarement été réunis. C'est en nous inspirant de cet autre domaine, bien plus étudié, que nous avons pu proposer des méthodes innovantes permettant d'améliorer les résultats de l'état de l'art. Notre première contribution a été de proposer deux mesures statistiques inspirées de l'*Anormalité* de l'état de l'art [Haydar et al., 2012], que nous avons appelées *AnormalitéCR* et *AnormalitéCRU* [Gras et al., 2015c]. Nous avons utilisé un indice de controverse permettant d'écarter du processus d'identification les ressources dont la répartition des notes ne permet pas d'isoler de réelles préférences spécifiques sur ces ressources. La comparaison des résultats obtenus avec ces deux mesures a mis en évidence l'importance de prise en compte du biais de l'utilisateur dans la tâche d'identification des GSU. La mesure d'*AnormalitéCRU* a ainsi apportée une amélioration significative des résultats de l'état de l'art. Cependant, malgré cette amélioration, le nombre d'identification de GSU erronées obtenu avec l'*AnormalitéCRU* reste encore élevé.

L'*AnormalitéCRU* et l'*AnormalitéCR* sont des mesures basées sur des indices permettant de représenter les lois de distribution des préférences sur les ressources (moyenne, écart-type, etc.). Nous pensons que ces indices ne sont pas suffisamment précis pour l'identification des GSU. Nous nous sommes donc inspirés des techniques basées directement sur la distribution des préférences observées, et de la *Vraisemblance* pour proposer une seconde contribution ayant pour objectif de maximiser la précision de l'identification des GSU. Pour cela, nous avons pris en compte les spécificités induites par les données de préférences des systèmes de recommandation sociale : l'imprécision des données, le biais utilisateur ou encore le manque de données. Les mesures que nous avons proposées, *VraisemblanceSBU* et *VraisemblanceID* [Gras et al., 2016], ne partent pas du postulat que les notes sur les ressources suivent une loi unique pour leur distribution. Les notes des utilisateurs sont centrées pour éviter le biais de l'utilisateur et le problème de l'imprécision des notes des utilisateurs est traité à l'aide d'un intervalle dynamique représentant les valeurs possibles qu'aurait pu prendre une note donnée. Le protocole d'évaluation que nous avons appliqué aux mesures que nous proposons montre que les résultats ainsi obtenus améliorent significativement les résultats de l'état de l'art dans le cas des mesures d'*AnormalitéCRU*, de *VraisemblanceSBU* et de *VraisemblanceID*. La mesure de *VraisemblanceID* est la plus performante et permet par exemple d'identifier jusqu'à 6 fois plus d'utilisateurs que la mesure d'*AnormalitéCRU* en respectant le même niveau de précision d'identification. Ces excellents résultats nous ont montré que la mesure de *VraisemblanceID* était suffisamment précise pour que nous nous basions dessus pour la suite de nos travaux.

La deuxième partie de ce travail a concerné la modélisation des GSU. Nous avons commencé par l'étude du voisinage des GSU pour trouver un moyen d'améliorer la qualité des recommandations fournies aux GSU. Bien que nous pensions intuitivement qu'un lien entre le voisinage des utilisateurs et leur propension à recevoir des recommandations de mauvaise qualité nous permettrait de mieux comprendre l'origine des mauvaises recommandations fournies aux GSU, les méthodes que nous avons proposées pour améliorer le voisinage des GSU n'ont pas permis d'améliorer la qualité des recommandations qu'ils reçoivent. Nous avons ensuite étudié la possibilité de modéliser précisément les GSU à l'aide d'une approche à base de factorisation de matrice. Pour atteindre cet objectif, nous nous sommes basés sur l'idée que si l'on souhaite modéliser finement les GSU avec une approche par FM, les notes des GSU doivent être considérées différemment de celles des utilisateurs non-GSU durant l'apprentissage. Nous avons proposés trois modèles : un qui exploite uniquement les notes des GSU ($GSUOnly$), un autre qui pondère différemment les notes des GSU et celles des non-GSU ($WeightedGSU$) et un dernier qui modélise chaque GSU indépendamment ($SingleGSU$). $GSUOnly$ a montré que l'utilisation des GSU uniquement n'est pas adaptée pour le calcul des recommandations d'un GSU, il est donc nécessaire d'introduire des utilisateurs non-GSU dans l'apprentissage du modèle. $WeightedGSU$ a permis de constater que donner trop de poids aux GSU revient à parasiter leurs propres recommandations et donner trop peu de poids aux GSU revient à ne pas tenir compte de leurs préférences. Il fallait donc à la fois isoler les GSU les uns des autres et limiter le nombre d'utilisateurs non-GSU utilisés pour la phase d'apprentissage du modèle. C'est ce que nous avons fait avec le modèle $SingleGSU$, qui permet alors d'améliorer la précision des recommandations fournies aux GSU. Ce modèle utilise les préférences d'un seul et unique GSU qu'il accompagne des préférences des utilisateurs non-GSU les plus similaires à ce dernier. Grâce à ce modèle, 72% des GSU reçoivent des recommandations de meilleure qualité, ce qui est très satisfaisant. De plus 52% des GSU bénéficient d'une amélioration de plus de 10% de la qualité des recommandations qui leurs sont fournies.

Alors que l'état de l'art n'envisageait que l'utilisation d'une approche par contenu pour satisfaire les besoins des GSU, nous avons montré qu'une approche purement sociale permet également

d'améliorer la satisfaction de plus de la moitié de ces utilisateurs. Il n'est donc désormais plus nécessaire d'utiliser deux approches de recommandation entièrement différentes pour satisfaire les GSU et le non-GSU. Les résultats montrent néanmoins qu'une partie des GSU ne sont pas d'avantage satisfaits par notre modèle, une étude approfondie de la manière de sélectionner les utilisateurs à associer à un GSU lors de la phase d'apprentissage reste donc à envisager.

Au cours de ce travail nous avons donc traité les trois aspects principaux du problèmes des GSU dans les systèmes de recommandation sociale : leur définition, leur identification et leur recommandation. Chacun de ces trois aspects a représenté des challenges scientifiques différents que nous avons su relever au fil du temps. Les résultats que nous avons obtenus à la fois pour l'identification et la modélisation des GSU ont été publiés dans des conférences internationales, offrant une forme de complétude à nos travaux.

De nombreuses perspectives ont été mentionnées tout au long de ce manuscrit. A court terme, trois défis scientifiques restent à relever :

— Pour la recommandation des GSU, les résultats que nous avons obtenus, notamment avec la méthode *SingleGSU*, doivent être approfondis pour étudier dans quelle mesure il est possible d'améliorer et de systématiser ces résultats. La première étape consistera à identifier les raisons pour lesquelles le modèle *SingleGSU* a permis d'améliorer la qualité des recommandations fournies aux GSU ou non. Pour identifier ces raisons, il est nécessaire d'analyser précisément les profils des GSU et de mettre en évidence les caractéristiques similaires à tous les GSU, qui bénéficient du modèle *SingleGSU*, ou à leur voisinage. Ensuite, le coefficient de corrélation de Pearson nous a permis de sélectionner des utilisateurs non-GSU à inclure dans le processus d'apprentissage du modèle dédié aux GSU (*SingleGSU*), permettant d'améliorer la qualité des recommandations fournies à plus de la moitié des GSU. La modification de la mesure de similarité utilisée pour la sélection des utilisateurs non-GSU dans le modèle *SingleGSU*, ou la mise au point d'une toute nouvelle méthode de FM, pourrait nous permettre d'atteindre cet objectif. L'utilisation d'une autre similarité, comme la couverture des ressources par exemple, permettrait peut-être de satisfaire d'avantage de GSU.
  De plus, comme nous l'avons évoqué, il n'est peut être pas nécessaire d'isoler les GSU un par un. Une approche permettant d'apprendre un modèle pour plusieurs GSU selon une méthodologie à définir permettrait de largement simplifier la mise en œuvre du modèle *SingleGSU*, lorsqu'il y a des milliers de GSU. Il s'agira d'étudier l'impact du regroupement de plusieurs GSU sur les performances du modèle *SingleGSU* dans la mesure où les GSU n'auraient, par exemple, aucune ressource co-votée.
— Considérons qu'un utilisateur qui reçoit des recommandations de mauvaise qualité est un utilisateur dont l'erreur sur l'estimation de ses préférences fait partie des 25% plus hautes erreurs enregistrées par le système. Dans ce cas, les mesures d'identification que nous proposons permettent de n'identifier que la moitié des utilisateurs qui reçoivent des recommandations de mauvaise qualité. Nous avons évoqué plusieurs raisons possibles à ces mauvaises recommandations, comme le démarrage à froid ou encore l'imprécision des notes. Les utilisateurs qui reçoivent des recommandations de mauvaise qualité et qui ne sont pas identifiés par les mesures que nous proposons ne sont donc pas nécessairement des GSU. Il sera néanmoins intéressant d'analyser les caractéristiques des utilisateurs qui ne sont pas détectés par nos mesures pour nous assurer qu'il n'existe pas d'autres formes de spécificité des préférences qui ne seraient pas prises en compte par les mesures d'identification que nous avons proposées.
— Une dernière perspective à court terme sera de valider nos résultats pour l'identification

et la modélisation des GSU sur d'autres jeux de données. Nous avons expérimenté nos contributions sur des jeux de données dont les ressources sont des films. Nous vérifierons nos conclusions sur un jeu de données d'un domaine différent, comme par exemple le e-learning. L'identification d'apprenants présentant des caractéristiques spécifiques pourra permettre d'éviter que des ressources pédagogiques inadaptées ne lui soient fournies par exemple.
Par ailleurs, l'accès à un jeu de données dans lequel certains utilisateurs sont identifiés sur la base de leurs spécificités permettrait de confirmer nos résultats sur un cas concret. Je pense par exemple à un jeu de données comportant à la fois les préférences d'utilisateurs neurotypiques et d'utilisateurs en situation d'autisme, de dyslexie, etc.

A moyen terme, je souhaite inclure l'ordre dans lequel les préférences ont été évaluées dans le processus d'identifications des GSU pour tenir compte de l'évolution des préférences au cours du temps. En effet, de nombreuses techniques de *data mining* permettent d'extraire des motifs peu fréquents dans des séquences d'événements. Je pense qu'il sera alors possible d'isoler une toute autre forme de spécificité. Par exemple, dans le cas des préférences cinématographiques, certains films peuvent être plus facilement appréciés s'ils ne sont pas vus après un film qui utilise les mêmes leviers scénaristiques. Une préférence spécifique peut donc s'expliquer par l'ordre dans lequel les ressources sont consultées, et l'inverse est également vrai. Certaines préférences spécifiques qui n'étaient pas spécifiques avant d'inclure l'ordre de notation pourraient donc émerger et faire apparaître une nouvelle forme de spécificité. De plus, l'ajout de la notion de temps et de séquence dans la représentation des préférences des utilisateurs pourrait permettre de distinguer une spécificité due à l'évolution d'une préférence dans le temps d'une spécificité récurrente qui n'est pas liée au temps. Cela permettrait peut être d'améliorer la précision d'identification des mesures que nous avons proposées. Pour la modélisation des GSU, l'impact de l'utilisation de préférences ordonnées sur la qualité des recommandations qui leurs sont fournies sera également une voie à explorer.

Ensuite, je vais chercher à identifier d'autres types de comportements spécifiques dans des données de préférences. Les précurseurs, les influenceurs ou encore les utilisateurs représentatifs sont des profils qu'il peut être intéressant d'identifier en amont de la recommandation. La prise en compte de l'ordre dans lequel les ressources ont été évaluées nous permet, par exemple, d'envisager l'identification des utilisateurs qui auront été les premiers à exprimer un certain type de préférences, les précurseurs. Nous pourrons exploiter les travaux présentés de ce manuscrit pour identifier les utilisateurs dont les préférences spécifiques dans le passé sont ensuite devenues des préférences communes par exemple. Nous pourrons donc généraliser nos travaux à la détection d'autres types d'utilisateur.

Enfin, depuis quelques années, les méthodes d'apprentissage automatique à base de réseaux de neurones profonds sont devenues une référence en matière de performances. Ces méthodes commencent à être utilisées dans le domaine de la recommandation sociale pour la modélisation des utilisateurs par exemple. L'adaptation de ces méthodes pour l'identification des GSU sera donc une perspective de ce travail. Le principal avantage des méthodes à base de réseaux de neurones profonds est la précision des modèles obtenus sur des problèmes d'apprentissage automatique supervisé, c'est-à-dire dans lesquels les données à identifier sont connues, ce qui facilite l'évaluation du modèle. A moins que nous ayons accès à un jeu de données compatible, il faudra étudier les capacités de ces méthodes sur les tâches d'apprentissage non-supervisé et les défis scientifiques à relever seront nombreux.

Bibliographie

[Adomavicius and Tuzhilin, 2005] Adomavicius, G. and Tuzhilin, A. (2005). Toward the next generation of recommender systems : A survey of the state-of-the-art. *IEEE transactions on knowledge and data engineering*, 17(6) :734–749.

[Adomavicius and Tuzhilin, 2011] Adomavicius, G. and Tuzhilin, A. (2011). Context-aware recommender systems. In *Recommender systems handbook*, pages 217–253. Springer.

[Aggarwal, 2013] Aggarwal, C. (2013). An introduction to outlier analysis. In *Outlier Analysis*, pages 1–40. Springer New York.

[Agrawal and Srikant, 1995] Agrawal, R. and Srikant, R. (1995). Mining sequential patterns. In *Data Engineering, 1995. Proceedings of the Eleventh International Conference on*, pages 3–14. IEEE.

[Akaike, 1998] Akaike, H. (1998). *Information Theory and an Extension of the Maximum Likelihood Principle*, pages 199–213. Springer New York, New York, NY.

[Aleksandrova et al., 2016] Aleksandrova, M., Brun, A., Boyer, A., and Chertov, O. (2016). Identifying Representative Users in Matrix Factorization-based Recommender Systems : Application to Solving the Content-less New Item Cold-start Problem. *Journal of Intelligent Information Systems*.

[Amatriain et al., 2009] Amatriain, X., Pujol, J. M., and Oliver, N. (2009). I like it... i like it not : Evaluating user ratings noise in recommender systems. In *Proceedings of the User Modeling, Adaptation, and Personalization (UMAP'09)*.

[Amblard et al., 2011] Amblard, M., Michel, M., and Manuel, R. (2011). Une analyse basée sur la S-DRT pour la modélisation de dialogues pathologiques. In Lafourcade, M. and Prince, V., editors, *Traitement Automatique des Langues Naturelles - TALN 2011*, page 6, Montpellier, France. Laboratoire d'Informatique de Robotique et de Microélectronique.

[Arrow, 1963] Arrow, K. J. (1963). *Social choice and individual values*, volume 12.

[Avazpour et al., 2014] Avazpour, I., Pitakrat, T., Grunske, L., and Grundy, J. (2014). *Dimensions and Metrics for Evaluating Recommendation Systems*, pages 245–273. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Balabanović and Shoham, 1997] Balabanović, M. and Shoham, Y. (1997). Fab : Content-based, collaborative recommendation. *Commun. ACM*, 40(3) :66–72.

[Becker, 1985] Becker, H. (1985). Outsiders. études de sociologie de la déviance, trad. de l'américain par j. *P. Briand, J.-M. Chapoulie, Paris, Métaillé*.

[Beel et al., 2016] Beel, J., Gipp, B., Langer, S., and Breitinger, C. (2016). Research-paper recommender systems : a literature survey. *International Journal on Digital Libraries*, 17(4) :305–338.

[Belkin and Croft, 1992] Belkin, N. J. and Croft, W. B. (1992). Information filtering and information retrieval : Two sides of the same coin ? *Communications of the ACM*, 35(12) :29–38.

[Bellogín et al., 2011] Bellogín, A., Castells, P., and Cantador, I. (2011). Predicting the performance of recommender systems : An information theoretic approach. In *Proc. of the Third Int. Conf. on Advances in Information Retrieval Theory*, ICTIR'11, pages 27–39. Springer.

[Bellogín et al., 2014] Bellogín, A., Said, A., and de Vries, A. (2014). The magic barrier of recommender systems – no magic, just ratings. In *Proc. of the 22nd Conf. on User Modelling, Adaptation and Personalization (UMAP)*.

[Ben-Gal, 2005] Ben-Gal, I. (2005). *Data Mining and Knowledge Discovery Handbook : A Complete Guide for Practitioners and Researchers*, chapter Outlier Detection. Kluwer Academic Publishers.

[Ben Ticha, 2015] Ben Ticha, S. (2015). *Recommandation personnalisée hybride*. PhD thesis. Thèse de doctorat dirigée par Boyer, AnneBsaïes, Khaled et Roussanaly, Azim Informatique Université de Lorraine 2015.

[Bennett et al., 2007] Bennett, J., Lanning, S., and Netflix, N. (2007). The netflix prize. In *In KDD Cup and Workshop in conjunction with KDD*.

[Benson, 1994] Benson, J. (1994). *The Rise of Consumer Society in Britain*. Longman.

[Bergson, 2012] Bergson, H. (2012). *Les deux sources de la morale et de la religion*. Flammarion.

[Berners-Lee and Cailliau, 1990] Berners-Lee, T. and Cailliau, R. (1990). Worldwideweb : Proposal for a hypertext project. *Retrieved on February*, 26 :2008.

[Billsus and Pazzani, 1998] Billsus, D. and Pazzani, M. J. (1998). Learning collaborative information filters. In *Proc. of the Fifteenth Int. Conf. on Machine Learning*, ICML '98, pages 46–54, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Bobadilla et al., 2012a] Bobadilla, J., Ortega, F., and Hernando, A. (2012a). A collaborative filtering similarity measure based on singularities. *Inf. Process. Manage.*, 48(2) :204–217.

[Bobadilla et al., 2012b] Bobadilla, J., Ortega, F., Hernando, A., and Bernal, J. (2012b). A collaborative filtering approach to mitigate the new user cold start problem. *Know.-Based Syst.*, 26 :225–238.

[Bobadilla et al., 2013] Bobadilla, J., Ortega, F., Hernando, A., and Gutiérrez, A. (2013). Recommender systems survey. *Know.-Based Syst.*, 46 :109–132.

[Bosch et al., 2016] Bosch, A., Bogers, T., and Kunder, M. (2016). Estimating search engine index size variability : A 9-year longitudinal study. *Scientometrics*, 107(2) :839–856.

[Bourdieu, 1984a] Bourdieu, P. (1984a). *Distinction : A social critique of the judgement of taste*. Harvard University Press.

[Bourdieu, 1984b] Bourdieu, P. (1984b). Espace social et genèse des "classes". *Actes de la recherche en sciences sociales*, 52(1) :3–14.

[Bray and Curtis, 1957] Bray, J. R. and Curtis, J. T. (1957). An ordination of the upland forest communities of southern wisconsin. *Ecological monographs*, 27(4) :325–349.

[Breese et al., 1998] Breese, J. S., Heckerman, D., and Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, UAI'98, pages 43–52, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Brin and Page, 1998] Brin, S. and Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1) :107–117.

[Brun et al., 2011] Brun, A., Castagnos, S., and Boyer, A. (2011). From Community Detection to Mentor Selection in Rating-Free Collaborative Filtering. *Advances in Multimedia Journal*, 2011 :1–19.

[Burke, 2002] Burke, R. (2002). Hybrid recommender systems : Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4) :331–370.

[Cabiria, 2011] Cabiria, J. (2011). Virtual worlds and identity exploration for marginalised people. In *reinventing ourselves : Contemporary Concepts of Identity in virtual Worlds*, pages 301–321. Springer.

[Candillier et al., 2008] Candillier, L., Meyer, F., and Fessant, F. (2008). Designing specific weighted similarity measures to improve collaborative filtering systems. In *Proceedings of the 8th Industrial Conference on Advances in Data Mining : Medical Applications, E-Commerce, Marketing, and Theoretical Aspects*, ICDM '08, pages 242–255, Berlin, Heidelberg. Springer-Verlag.

[Castagnos, 2008] Castagnos, S. (2008). *Modélisation de comportements et apprentissage stochastique non supervisé de stratégies d'interactions sociales au sein de systèmes temps réel de recherche et d'accès à l'information.* These, Université Nancy II.

[Castagnos et al., 2013] Castagnos, S., Brun, A., and Boyer, A. (2013). When diversity is needed... but not expected ! In *IMMM 2013, The Third Int. Conf. on Advances in Information Mining and Management.*

[Chandola et al., 2009] Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection : a survey. *ACM CSUR*, 41(3).

[Chawla and Gionis, 2013] Chawla, S. and Gionis, A. (2013). k-means– : A unified approach to clustering and outlier detection. In *Proceedings of the 2013 SIAM International Conference on Data Mining*, pages 189–197. SIAM.

[Chen and Goodman, 1998] Chen, S. and Goodman, J. (1998). An empirical study of smoothing techniques for language modeling. Technical Report TR-10-98, Harvard University.

[Chickering et al., 1997] Chickering, D. M., Heckerman, D., and Meek, C. (1997). A bayesian approach to learning bayesian networks with local structure. In *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence*, UAI'97, pages 80–89, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Claypool et al., 1999] Claypool, M., Gokhale, A., and Miranda, T. (1999). Combining content-based and collaborative filters in an online newspaper. In *Proceedings of the SIGIR Workshop on Recommender Systems : Algorithms and Evaluation.*

[Cosley et al., 2003] Cosley, D., Lam, S. K., Albert, I., Konstan, J. A., and Riedl, J. (2003). Is seeing believing ? : how recommender system interfaces affect users' opinions. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 585–592. ACM.

[Cousineau and Chartier, 2010] Cousineau, D. and Chartier, S. (2010). Outliers detection and treatment : a review. *International Journal of Psychological Research*, 3(1).

[Croft, 2003] Croft, B. (2003). *Language modeling for information retrieval.* Springer.

[Cusson, 1992] Cusson, M. (1992). Déviance.

[Damon, 1998] Damon, J. (1998). Des hommes en trop. essai sur le vagabondage et la mendicité.

[Das et al., 2014] Das, J., Mukherjee, P., Majumder, S., and Gupta, P. (2014). Clustering-based recommender system using principles of voting theory. In *Contemporary Computing and Informatics (IC3I), 2014 International Conference on*, pages 230–235. IEEE.

[Del Prete and Capra, 2010] Del Prete, L. and Capra, L. (2010). differs : A mobile recommender service. In *Proc. of the 2010 Eleventh Int. Conf. on Mobile Data Management*, MDM '10, pages 21–26, Washington, USA. IEEE Computer Society.

[Dempster et al., 1977] Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38.

[Douglas J. D., 1982] Douglas J. D., W. F. C. (1982). *The sociology of deviance : an introduction.* Little Brown.

[Dror et al., 2012] Dror, G., Koenigstein, N., Koren, Y., and Weimer, M. (2012). The yahoo! music dataset and kdd-cup'11. In Dror, G., Koren, Y., and Weimer, M., editors, *Proceedings of KDD Cup 2011*, volume 18 of *Proceedings of Machine Learning Research*, pages 3–18. PMLR.

[Duby, 1995] Duby, G. (1995). L'histoire des femmes : Introduction. *Journal of the Economic and Social History of the Orient*, 38 :121.

[Durkheim, 1895] Durkheim, É. (1895). *Les règles de la méthode sociologique.* Flammarion.

[Edwards, 1972] Edwards, A. (1972). *Likelihood.* Cambridge University Press.

[Eskin, 2000] Eskin, E. (2000). Anomaly detection over noisy data using learned probability distributions. In *In Proceedings of the International Conference on Machine Learning.* Citeseer.

[Esslimani et al., 2009] Esslimani, I., Brun, A., and Boyer, A. (2009). A collaborative filtering approach combining clustering and navigational based correlations. In Filipe, J. and Cordeiro, J., editors, *WEBIST*, pages 364–369. INSTICC Press.

[Ester et al., 1996] Ester, M., Kriegel, H.-P., Sander, J., Xu, X., et al. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231.

[Filo and Yang, 1994] Filo, D. and Yang, J. (1994). Yahoo home page. *URL : http ://www. yahoo. com.*

[Fix and Jr, 1951] Fix, E. and Jr (1951). Discriminatory Analysis : Nonparametric Discrimination : Consistency Properties. Technical Report Project 21-49-004, Report Number 4, USAF School of Aviation Medicine, Randolf Field, Texas.

[Gao et al., 2006] Gao, J., Cheng, H., and Tan, P.-N. (2006). Semi-supervised outlier detection. In *Proceedings of the 2006 ACM Symposium on Applied Computing*, SAC '06, pages 635–636, New York, NY, USA. ACM.

[Gemulla et al., 2011] Gemulla, R., Nijkamp, E., Haas, P. J., and Sismanis, Y. (2011). Large-scale matrix factorization with distributed stochastic gradient descent. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pages 69–77, New York, NY, USA. ACM.

[Gena et al., 2011] Gena, C., Brogi, R., Cena, F., and Vernero, F. (2011). *The Impact of Rating Scales on User's Rating Behavior*, pages 123–134. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Ghazanfar and Prugel-Bennett, 2011] Ghazanfar, M. and Prugel-Bennett, A. (2011). Fulfilling the needs of gray-sheep users in recommender systems, a clustering solution. In *2011 Int. Conf. on Information Systems and Computational Intelligence.*

[Ghazanfar and Prügel-Bennett, 2014] Ghazanfar, M. and Prügel-Bennett, A. (2014). Leveraging clustering approaches to solve the gray-sheep users problem in recommender systems. *Expert Systems with Applications*, 41 :3261–3275.

[Ghorbani and Novin, 2016] Ghorbani, H. and Novin, A. H. (2016). An introduction on separating gray-sheep users in personalized recommender systems using clustering solution. *International Journal of Computer Science and Software Engineering (IJCSSE)*, 5(2) :14–18.

[Goldberg, 1992] Goldberg, D. (1992). Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12) :61–70.

[Gomez-Uribe and Hunt, 2015] Gomez-Uribe, C. A. and Hunt, N. (2015). The netflix recommender system : Algorithms, business value, and innovation. *ACM Trans. Manage. Inf. Syst.*, 6(4) :13 :1–13 :19.

[Gosling and Ric, 1996] Gosling, P. and Ric, F. (1996). *Psychologie sociale*. Number vol. 1 in Collection Lexifac. Bréal.

[Gras et al., 2015a] Gras, b., Brun, A., and Boyer, A. (2015a). Identification des utilisateurs atypiques dans les systèmes de recommandation sociale. In *EGC - Extraction et Gestion de Connaissances*, Esch sur Alzette, Luxembourg.

[Gras et al., 2015b] Gras, B., Brun, A., and Boyer, A. (2015b). Identifying users with atypical preferences to anticipate inaccurate recommendations. In *Proceedings of the 11th International Conference on Web Information Systems and Technologies*.

[Gras et al., 2015c] Gras, B., Brun, A., and Boyer, A. (2015c). When users with preferences different from others get inaccurate recommendations. In *International Conference on Web Information Systems and Technologies*, pages 191–210. Springer.

[Gras et al., 2016] Gras, B., Brun, A., and Boyer, A. (2016). Identifying grey sheep users in collaborative filtering : a distribution-based technique. In *Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization*, pages 17–26. ACM.

[Gras et al., 2017] Gras, B., Brun, A., and Boyer, A. (2017). Can matrix factorization improve the accuracy of recommendations provided to grey sheep users ? In *Proceedings of the 13th International Conference on Web Information Systems and Technologies - Volume 1 : WEBIST,*, pages 88–96. INSTICC, ScitePress.

[Grčar et al., 2006] Grčar, M., Fortuna, B., Mladenič, D., and Grobelnik, M. (2006). knn versus svm in the collaborative filtering framework. *Data Science and Classification*, pages 251–260.

[Grcar et al., 2005a] Grcar, M., Mladenic, D., Fortuna, B., and Grobelnik, M. (2005a). *Advances in Web Mining and Web Usage Analysis*, volume 4198, chapter Data Sparsity Issues in the Collaborative Filtering Framework, pages 58–76. Springer.

[Grcar et al., 2005b] Grcar, M., Mladenic, D., and Grobelnik, M. (2005b). Data quality issues in collaborative filtering. In *Proc. of ESWC-2005 Workshop on End User Aspects of the Semantic Web*.

[Griffith et al., 2012] Griffith, J., O'Riordan, C., and Sorensen, H. (2012). Investigations into user rating information and predictive accuracy in a collaborative filtering domain. In *Proc. of the 27th ACM Symposium on Applied Computing*.

[Guo et al., 2014] Guo, G., Zhang, J., and Thalmann, D. (2014). Merging trust in collaborative filtering to alleviate data sparsity and cold start. *Knowledge-Based Systems*, 57(0) :57 – 68.

[Han and Kamber, 2006] Han, J. and Kamber, M. (2006). *Data Mining : Concepts and Techniques*. Morgan Kaufmann.

[Hawkins, 1980] Hawkins, D. M. (1980). *Identification of outliers*, volume 11. Springer.

[Haydar et al., 2012] Haydar, C., Roussanaly, A., and Boyer, A. (2012). Clustering users to explain recommender systems' performance fluctuation. In *Foundations of Intelligent Systems*, volume 7661 of *LNCS*, pages 357–366. Springer.

[He et al., 2017] He, X., Liao, L., Zhang, H., Nie, L., Hu, X., and Chua, T.-S. (2017). Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*, pages 173–182. International World Wide Web Conferences Steering Committee.

[Hodge and Austin, 2004] Hodge, V. and Austin, J. (2004). A survey of outlier detection methodologies. *Artificial Intelligence Review*, 22 :85–126.

[Hofmann, 2004] Hofmann, T. (2004). Latent semantic models for collaborative filtering. *ACM Trans. Inf. Syst.*, 22(1) :89–115.

[Hu et al., 2008] Hu, Y., Koren, Y., and Volinsky, C. (2008). Collaborative filtering for implicit feedback datasets. In *Proc. of the 2008 Eighth IEEE Int. Conf.e on Data Mining*, ICDM '08, pages 263–272, Washington, DC, USA. IEEE Computer Society.

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

[Jolliffe, 2002] Jolliffe, I. T. (2002). Principal component analysis and factor analysis. *Principal component analysis*, pages 150–166.

[Jones et al., 2011] Jones, N., Brun, A., and Boyer, A. (2011). Comparisons instead of ratings : Towards more stable preferences. In *2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, volume 1, pages 451–456.

[Kim and Ahn, 2008] Kim, K. and Ahn, H. (2008). A recommender system using ga k-means clustering in an online shopping market. *Expert Systems with Applications*, 34(2) :1200 – 1209.

[Kneser and Ney, 1995] Kneser, R. and Ney, R. (1995). Improved backing-off for m-gram language modeling. In *Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, volume 1, pages 181–184.

[Koren, 2009] Koren, Y. (2009). The bellkor solution to the netflix grand prize. *Netflix prize documentation*, 81 :1–10.

[Koren, 2010] Koren, Y. (2010). Collaborative filtering with temporal dynamics. *Communications of the ACM*, 53(4) :89–97.

[Koren et al., 2009] Koren, Y., Bell, R., and Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, 42(8) :30–37.

[Lemire and Maclachlan, 2005] Lemire, D. and Maclachlan, A. (2005). Slope one predictors for online rating-based collaborative filtering. In *Proceedings of the 2005 SIAM International Conference on Data Mining*, pages 471–475. SIAM.

[Linden et al., 2003] Linden, G., Smith, B., and York, J. (2003). Amazon.com recommendations : Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1) :76–80.

[Liu et al., 2014] Liu, H., Hu, Z., Mian, A., Tian, H., and Zhu, X. (2014). A new user similarity model to improve the accuracy of collaborative filtering. *Know.-Based Syst.*, 56 :156–166.

[Liu and Ihler, 2011] Liu, Q. and Ihler, A. (2011). Learning scale free networks by reweighted 1 regularization.

[Loureiro et al., 2004] Loureiro, A., Torgo, L., and Soares, C. (2004). Outlier detection using clustering methods : a data cleaning application. In *Proceedings of KDNet Symposium on Knowledge-based Systems for the Public Sector. Bonn, Germany.*

[López-Nores et al., 2012] López-Nores, M., Blanco-Fernández, Y., Pazos-Arias, J. J., and Gil-Solla, A. (2012). Property-based collaborative filtering for health-aware recommender systems. *Expert Systems with Applications*, 39(8) :7451 – 7457.

[MacQueen, 1967] MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1 : Statistics*, pages 281–297, Berkeley, Calif. University of California Press.

[Markou and Singh, 2003] Markou, M. and Singh, S. (2003). Novelty detection : a review—part 1 : statistical approaches. *Signal processing*, 83(12) :2481–2497.

[Massin, 1998] Massin, B. (1998). La science nazie et l'extermination des marginaux. *L'Histoire*, 217 :52–59.

[Maunier, 1929] Maunier, R. (1929). *Introduction à la sociologie*. F. Alcan.

[McLaughlin and Herlocker, 2004] McLaughlin, M. R. and Herlocker, J. L. (2004). A collaborative filtering algorithm and evaluation metric that accurately model the user experience. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, pages 329–336, New York, NY, USA. ACM.

[Merton, 1968] Merton, R. K. (1968). Social theory and social structure.

[Mucchielli, 1999a] Mucchielli, L. (1999a). La déviance : Entre normes, transgression et stigmatisation : Normes, interdits, déviances. *Sciences humaines*, (99) :20–25.

[Mucchielli, 1999b] Mucchielli, L. (1999b). Les champs de la sociologie pénale. vingt ans de recherches et de débats dans déviance et société (1977-1997). *Déviance et société*, 23(1) :3–40.

[Ng, 2004] Ng, A. Y. (2004). Feature selection, l1 vs. l2 regularization, and rotational invariance. In *Proceedings of the Twenty-first International Conference on Machine Learning*, ICML '04, pages 78–, New York, NY, USA. ACM.

[Nigam, 1999] Nigam, K. (1999). Using maximum entropy for text classification. In *In IJCAI-99 Workshop on Machine Learning for Information Filtering*, pages 61–67.

[Oard and Marchionini, 1998] Oard, D. W. and Marchionini, G. (1998). A conceptual framework for text filtering process. Technical report.

[Ogien, 1995] Ogien, A. (1995). Sociologie de la déviance.

[Paatero, 1997] Paatero, P. (1997). Least squares formulation of robust non-negative factor analysis. *Chemometrics and intelligent laboratory systems*, 37(1) :23–35.

[Pascal and Faugère, 1897] Pascal, B. and Faugère, P. (1897). *Pensées, fragments et lettres de Blaise pascal, publiés pour la première fois conformément aux manuscrits.* Number vol. 2 in Pensées, fragments et lettres de Blaise pascal, publiés pour la première fois conformément aux manuscrits. E. Leroux.

[Pazzani and Billsus, 2007] Pazzani, M. J. and Billsus, D. (2007). *Content-Based Recommendation Systems*, pages 325–341. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Penn and Zalesne, 2007] Penn, M. and Zalesne, K. (2007). *Mircotrends : the small forces behind tomorrow's big changes.* Twelve.

[Pennock and Horvitz, 1999] Pennock, D. M. and Horvitz, E. (1999). Analysis of the axiomatic foundations of collaborative filtering. *Ann Arbor*, 1001 :48109–2110.

[Perlich, 2016] Perlich, C. (2016). Automated machine learning in the wild. In *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys '16, pages 1–1, New York, NY, USA. ACM.

[Ramaswamy et al., 2000] Ramaswamy, S., Rastogi, R., and Shim, K. (2000). Efficient algorithms for mining outliers from large data sets. In *Proceedings of the ACM SIGMOD International Conference on Management of Data.*

[Ratner, 2004] Ratner, B. (2004). *Statistical modeling and analysis for database marketing : effective techniques for mining big data*. CRC Press.

[Resnick et al., 1994] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J. (1994). Grouplens : An open architecture for collaborative filtering of netnews. In *Proc. of the 1994 ACM Conf. on Computer Supported Cooperative Work*, CSCW'94.

[Robert et al., 1997] Robert, P., Soubiran-Paillet, F., and van de Kerchove, M. (1997). *Normes normes juridiques normes pénales. Pour une sociologie des frontières*. Logiques sociales. Droit et société.

[Rousseeuw and Leroy, 2005] Rousseeuw, P. J. and Leroy, A. M. (2005). *Robust regression and outlier detection*, volume 589. John wiley & sons.

[Ruszczynski and Syski, 1983] Ruszczynski, A. and Syski, W. (1983). Stochastic approximation method with gradient averaging for unconstrained problems. *IEEE Transactions on Automatic Control*, 28(12) :1097–1105.

[Salton, 1986] Salton, G. (1986). *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA.

[Sarwar et al., 2001] Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, pages 285–295, New York, NY, USA. ACM.

[Sarwar et al., 2002] Sarwar, B. M., Karypis, G., Konstan, J., and Riedl, J. (2002). Recommender systems for large-scale e-commerce : Scalable neighborhood formation using clustering.

[Schein et al., 2001] Schein, A. I., Popescul, A., Ungar, L. H., and Pennock, D. (2001). Generative models for cold-start recommendations. In *Proc. of the 2001 SIGIR workshop on recommender systems*.

[Schickel-Zuber and Faltings, 2006] Schickel-Zuber, V. and Faltings, B. (2006). Overcoming incomplete user models in recommendation systems via an ontology. In *Proc. of the 7th Int. Conf. on Knowledge Discovery on the Web*, WebKDD'05, pages 39–57, Berlin. Springer.

[Sen, 1986] Sen, A. (1986). Chapter 22 social choice theory. volume 3 of *Handbook of Mathematical Economics*, pages 1073 – 1181. Elsevier.

[Shani and Gunawardana, 2009] Shani, G. and Gunawardana, A. (2009). Evaluating recommender systems. Technical report.

[Shardanand and Maes, 1995] Shardanand, U. and Maes, P. (1995). Social information filtering : Algorithms for automating “word of mouth”. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '95, pages 210–217, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co.

[Silverman, 1986] Silverman, B. (1986). *Density estimation for statistics and data analysis*. Chapman and Hall, London.

[Slater, 1997] Slater, D. (1997). *Consumer Culture*. Wiley Online Library.

[Srivastava, 2016] Srivastava, A. (2016). Gray sheep, influential users, user modeling and recommender system adoption by startups. In *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys '16, pages 443–446, New York, NY, USA. ACM.

[Su and Khoshgoftaar, 2009] Su, X. and Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. *Adv. in Artif. Intell.*, 2009 :4 :2–4 :2.

[Takacs et al., 2009] Takacs, G., Pilaszy, I., Nemeth, B., and Tikk, D. (2009). Scalable collaborative filtering approaches for large recommender systems. *Journal of Machine Learning Research*, 10 :623–656.

[Terveen and Hill, 2001] Terveen, L. and Hill, W. (2001). Beyond recommender systems : Helping people help each other. *HCI in the New Millennium*, 1(2001) :487–509.

[Tooby and Cosmides, 1990] Tooby, J. and Cosmides, L. (1990). On the universality of human nature and the uniqueness of the individual : The role of genetics and adaptation. *Journal of personality*, 58(1) :17–67.

[Töscher et al., 2009] Töscher, A., Jahrer, M., and Bell, R. M. (2009). The bigchaos solution to the netflix grand prize. *Netflix prize documentation*, pages 1–52.

[Tsuruoka et al., 2009] Tsuruoka, Y., Tsujii, J., and Ananiadou, S. (2009). Stochastic gradient descent training for l1-regularized log-linear models with cumulative penalty. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP : Volume 1-Volume 1*, pages 477–485. Association for Computational Linguistics.

[Ungar and Foster, 1998] Ungar, L. H. and Foster, D. P. (1998). Clustering methods for collaborative filtering. In *AAAI workshop on recommendation systems*, volume 1, pages 114–129.

[van Baalen, 2016] van Baalen, M. (2016). Deep matrix factorization for recommendation.

[Volkovs et al., 2017] Volkovs, M., Yu, G. W., and Poutanen, T. (2017). Content-based neighbor models for cold start in recommender systems. In *Proceedings of the Recommender Systems Challenge 2017*, page 7. ACM.

[Wang and Tang, 2015] Wang, J. and Tang, Q. (2015). Recommender systems and their security concerns.

[Weimer and Necula, 2005] Weimer, W. and Necula, G. (2005). Mining temporal specifications for error detection. *Tools and Algorithms for the Construction and Analysis of Systems*, pages 461–476.

[Weyers, 2012] Weyers, B. (2012). The internet's impact on our thinking. *AP Literature.*

[Wittgenstein and Ogden, 1921] Wittgenstein, L. and Ogden, C. (1921). *Tractatus Logico-philosophicus.* International library of psychology, philosophy, and scientific method. Routledge.

[Wong and Lane, 1981] Wong, M. A. and Lane, T. (1981). A kth nearest neighbour clustering procedure. In *Computer Science and Statistics : Proceedings of the 13th Symposium on the Interface*, pages 308–311. Springer.

[Xue et al., 2005] Xue, G.-R., Lin, C., Yang, Q., Xi, W., Zeng, H.-J., Yu, Y., and Chen, Z. (2005). 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, pages 114–121, New York, NY, USA. ACM.

[Xue et al., 2017] Xue, H.-J., Dai, X.-Y., Zhang, J., Huang, S., and Chen, J. (2017). Deep matrix factorization models for recommender systems.

[Yao, 1995] Yao, Y. Y. (1995). Measuring retrieval effectiveness based on user preference of documents. *J. Am. Soc. Inf. Sci.*, 46(2) :133–145.

[Yera Toledo et al., 2013] Yera Toledo, R., Martinez Lopez, L., and Caballero Mota, Y. (2013). Managing natural noise in collaborative recommender systems. In *IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS), 2013 Joint*, pages 872–877.

[Yu et al., 2014] Yu, H.-F., Hsieh, C.-J., Si, S., and Dhillon, I. S. (2014). Parallel matrix factorization for recommender systems. *Knowl. Inf. Syst.*, 41 :793–819.

[Zeng et al., 2010] Zeng, W., Shang, M.-S., Zhang, Q.-M., Lü, L., and Zhou, T. (2010). Can Dissimilar Users Contribute to Accuracy and Diversity of Personalized Recommendation ? *International Journal of Modern Physics C*, 21 :1217–1227.

[Zheng et al., 2004] Zheng, A. X., Jordan, M. I., Liblit, B., and Aiken, A. (2004). Statistical debugging of sampled programs. In Thrun, S., Saul, L. K., and Schölkopf, B., editors, *Advances in Neural Information Processing Systems 16*, pages 603–610. MIT Press.

Résumé

Un système de recommandation a pour objectif de recommander à un utilisateur, appelé utilisateur actif, des ressources pertinentes pour lui. Pour permettre cette recommandation, le système utilise les informations qu'il a collectées sur l'utilisateur actif ou sur les ressources. Le filtrage collaboratif (FC) est une approche de recommandation très répandue. Les données exploitées par le FC sont les préférences exprimées par des utilisateurs sur des ressources. Le FC repose sur l'hypothèse que les préférences des utilisateurs sont cohérentes entre elles, ce qui permet d'inférer les préférences d'un utilisateur à partir des préférences des autres utilisateurs. Dans une approche de FC, il est nécessaire qu'au moins une communauté d'utilisateurs partage les préférences de l'utilisateur actif pour pouvoir lui proposer des recommandations de bonne qualité. Définissons une préférence spécifique comme une préférence qui ne serait partagée pour aucun groupe d'utilisateurs. Un utilisateur possédant plusieurs préférences spécifiques qu'il ne partage avec aucun autre utilisateur du service sera probablement mal servi par une approche de FC classique. Il s'agit du problème des Grey Sheep Users (GSU). Dans cette thèse, je réponds à trois questions distinctes. La première question est : qu'est-ce qu'une préférence spécifique ? J'y apporte une réponse en proposant des hypothèses associées que je valide expérimentalement. Ensuite, je me demande comment identifier les GSU dans les données ? Cette identification est importante afin d'anticiper les mauvaises recommandations qui seront fournies à ces utilisateurs. Je propose des mesures numériques permettant d'identifier les GSU dans un jeu de données de recommandation sociale. Ces mesures sont significativement plus performantes que celles de l'état de l'art. Enfin, comment modéliser ces GSU pour améliorer la qualité des recommandations qui leurs sont fournies ? Les préférences spécifiques des GSU complexifient la recherche d'utilisateurs cohérents avec eux et conduisent à de mauvaises recommandations. Je propose des méthodes inspirées du domaine de l'apprentissage automatique et dédiées à la modélisation des GSU permettant d'améliorer la qualité des recommandations qui leurs sont fournies.

Mots-clés: Modélisation utilisateur,modélisation de préférences,systèmes de recommandation,apprentissage automatique,données aberrantes,utilisateurs atypiques

Abstract

A recommender system aims at providing relevant resources to a user, named the active user. To allow this recommendation, the system exploits the information it has collected about the active user or about resources. The collaborative filtering (CF) is a widely used recommandation approach. The data exploited by CF are the preferences expressed by users on resources. CF is based on the assumption that preferences are consistent between users, allowing a user's preferences to be inferred from the preferences of other users. In a CF-based recommender system, at least one user community has to share the preferences of the active user to provide him with high quality recommendations. Let us define a specific preference as a preference that is not shared by any group of user. A user with several specific preferences will likely be poorly served by a classic CF approach. This is the problem of Grey Sheep Users (GSU). In this thesis, I focus on three separate questions. The first question is : what is a specific preference ? I give an answer by proposing associated hypotheses that I validate experimentally. Next, I wonder how to identify GSU in preference data ? This identification is important to anticipate the low quality recommendations that will be provided to these users. I propose numerical indicators to identify GSU in a social recommendation dataset. These indicators outperform those of the state of the art and allow to isolate users whose quality of recommendations is very low. Third, how can I model GSU to improve the quality of the recommendations they receive ? The specific preferences of GSU make the search of users who are consistent with them difficult and lead in general to poor recommendations. Therefore, I propose new recommendation approaches to allow GSU to benefit from the opinions of other users as in the case of a user with fewer specific preferences.

Keywords: User modeling,preference modeling,recommender systems,machine learning,outliers,grey sheep users