Inférence de liens signés dans les réseaux sociaux, par apprentissage à partir d'interactions utilisateur

Luc-Aurelien Gauthier

UPMC SORBONNE UNIVERSITÉS

EDITE-de-Paris

Inférence de liens signés dans les réseaux sociaux, par apprentissage à partir d'interactions utilisateur

THÈSE

présentée et soutenue publiquement le 2 décembre 2015

pour l'obtention du

Doctorat de l'Université Pierre et Marie Curie

(mention informatique)

par

Luc-Aurélien Gauthier

Composition du jury

Rapporteurs :	Bénédicte Le Grand	de l'Université Paris 1 Panthéon - Sorbonne
	Lynda Tamine-Lechani	de l'Université Paul Sabatier
Examinateurs :	Anne Doucet	de l'Université Pierre et Marie Curie
	Rushed Kanawati	de l'Université Paris 13
Directeur de Thèse :	Patrick Gallinari	de l'Université Pierre et Marie Curie
Co-Encadrant :	Benjamin Piwowarski	du Centre national de la recherche scientifique (CNRS)

Laboratoire d'Informatique de Paris 6

Mis en page avec la classe thesul.

Sommaire

Table des figures

Liste des tableaux

Introduction générale

L'étude du comportement des individus à travers l'analyse de leur façon de penser et de leur façon d'agir a toujours suscitée un fort intérêt, et ce, bien avant l'arrivée des premiers réseaux sociaux sur internet. Aujourd'hui, la tâche se démocratise car les utilisateurs sont de plus en plus liés entre eux via de multiples plate-formes. Les réseaux se densifient et la diversité des données augmente également : contenus personnels (Facebook, Google+) ou professionnels (LinkedIn, Viadeo), partage de vidéos en ligne (Youtube), avis divers sur les produits issus de ventes en ligne (Amazon, Epinions) ou encore des plate-formes de vidéos ou de musique à la demande (Netflix, Deezer). La popularité des usages sur le web ouvre donc la porte à de nombreuses nouvelles tâches, car il devient possible de collecter facilement de grosses quantités de données sur les relations entre les utilisateurs. Le traitement et l'analyse de ces données requièrent des outils sophistiqués ; on utilise souvent la théorie des graphes, particulièrement adaptée aux données structurées sous forme de réseau. Les utilisateurs, les items ou les contenus sont représentés par des noeuds et ces noeuds sont reliés par des liens qui peuvent alors prendre diverses formes selon leur définition (simples, orientés, valués).

L'augmentation de la quantité de données disponibles et leur diversification permettent une meilleure analyse des utilisateurs, car chaque contenu apporte un renseignement. Malheureusement, cette quantité phénoménale d'informations devient parfois complexe à manipuler, aussi bien du point de vue technique que du point de vue de l'utilisateur. Celui-ci rencontre de plus en plus de difficultés à s'orienter, noyé dans l'abondance des choix qui lui sont offerts. C'est un réel paradoxe puisque le nombre grandissant de possibilités offertes et la richesse des contenus a l'effet inverse de ce qui est attendu : les utilisateurs ont de moins en moins de chance de trouver ce qu'ils recherchent. Alors que les utilisateurs peuvent facilement passer d'un service à un autre, satisfaire et fidéliser devient un véritable enjeu pour n'importe quel service en ligne qui doit maintenir (et augmenter) le trafic nécessaire à son succès, voire sa survie.

Face à ce constat, de nombreuses tâches ont émergé, parmi lesquelles la recommandation dont l'objectif est d'identifier les items les plus susceptibles de plaire aux utilisateurs, la prédiction de liens (entre utilisateurs) ou encore la détection de communautés. De nombreux graphes contiennent implicitement ou explicitement des relations antagonistes (signées). Dans ce manuscrit, nous nous intéressons à leur sémantique. Nous proposons des mesures de similarité qui prennent en compte la polarité des jugements que les utilisateurs partagent à propos d'items et validons expérimentalement ce type d'approche en nous appuyant sur deux tâches : la recommandation et la prédiction de liens.

La problématique des relations signées

Généralement, nous distinguons deux formes de liens entre les utilisateurs : les liens explicites ou les liens implicites. Les liens sont explicites lorsque les utilisateurs sont reliés au sein d'un même réseau social, ils sont implicites lorsque les utilisateurs partagent les mêmes groupes ou ont évalué un même item par exemple. Ces liens sont exploités pour tirer profit de l'*homophilie* et la *régularité* stipulant que les individus connectés entre eux partagent les mêmes caractéristiques de profil (âge, profession, hobbies) et tendent à appartenir aux

mêmes communautés d'intérêt. Ainsi, ces liens sont exploités pour leur connotation positive (amitié, collaboration, partage). Pourtant, beaucoup de relations sociales opposent souvent deux forces antagonistes (ami/ennemi ou confiance/méfiance) et nous observons régulièrement des tensions entre les utilisateurs, résultats de controverses ou encore de désaccords.

Dans le domaine du web, l'analyse des liens signés en est à ses débuts et peu de réseaux contiennent des relations de défiance. Pourtant, l'information qu'apportent les liens négatifs s'avère bien utile pour de nombreuses tâches [Kunegis et al., 2013] car ils viennent compléter l'information des liens positifs. Quelques travaux s'en sont servi pour améliorer une tâche de prédiction de liens classiques [Kunegis and Lommatzsch, 2009, Leskovec et al., 2010a] où les liens négatifs permettent une augmentation du nombre de bonnes prédictions de l'ordre de 5%. D'autres travaux s'en sont servi pour affiner des modèles de recommandation [Ma et al., 2009, Yang et al., 2012] permettant un gain du même ordre (5% en prédiction de notes).

Les sources de liens négatifs

Les liens signés peuvent être collectés soit lorsqu'ils sont explicités par les utilisateurs, soit via les relations indirectes qui peuvent exister entre eux.

Les liens signés explicites entre les utilisateurs sont les moins faciles à collecter. Notre hypothèse est que les utilisateurs (comme les fournisseurs de services) considèrent ces liens intrusifs et inconvenants. En effet, un lien négatif envers un autre utilisateur peut être perçu comme une agression, ce que les plate-formes en ligne cherchent déjà à combattre. Autoriser les relations hostiles représente donc un risque, d'autant plus que les utilisateurs ne perçoivent souvent même pas l'intérêt de les expliciter puisque cela ne permet pas d'améliorer leur profil contrairement aux liens positifs.

Pour pallier l'absence de liens signés explicites, il est possible d'exploiter des sources issues d'interactions indirectes entre les utilisateurs. Il s'agit par exemple d'un utilisateur qui va juger le contenu d'un autre utilisateur lorsque ce dernier a exprimé un avis à propos d'un item (à l'instar des sites de ventes en ligne). En pratique, ces liens sont plus faciles à collecter, notamment parce que les utilisateurs s'adressent avant tout au contenu et les avis négatifs sont alors perçus comme tel ; les autres utilisateurs ne perçoivent pas cela comme une provocation.

Analyse de graphes sociaux signés

Les liens signés soulèvent des difficultés si on veut les exploiter pour des tâches classiques d'accès à l'information ; d'une part parce que les règles de transitivité du type *l'ami de mon ami est mon ami* ne fonctionnent plus, d'autre part car le signe (et souvent l'orientation) des liens rend caduc une partie de l'arsenal mathématique à notre disposition.

Les premiers travaux traitant le sujet des liens signés, à la frontière entre la sociologie et la théorie des graphes, ont avant tout porté sur leur sémantique [Harary, 1953, Davis, 1977]. Ces travaux ont permis d'étudier la manière dont interagissent les liens signés pour lesquels la simple transitivité ne peut être utilisée. De cette problématique est née la théorie de l'équilibre, stipulant que le produit du signe des trois liens d'une triade doit être positif (par exemple

l'ennemi de mon ami est mon ennemi). Toutefois, cette théorie ne traite que le cas des liens réciproques entre utilisateurs, ce qui l'exclut de la majorité des graphes de terrain actuels.

Les graphes de terrain ont été étudié bien plus tard, lorsque les premières plate-formes en ligne ont proposé ce type de relation. Les liens y sont plus systématiquement orientés et la sémantique est alors différente. Les premiers travaux [Guha et al., 2004, Kunegis and Lommatzsch, 2009, Kunegis et al., 2010, Kunegis et al., 2009, Kerchove and Dooren, 2008] ont étendu des méthodes existantes à base de marches aléatoires afin de prédire le signe des liens entre les utilisateurs, alors même que ces méthodes reposent sur une théorie qui n'est pas adaptée au cas des liens négatifs (la convergence des modèles n'est plus garantie). D'autres travaux [Leskovec et al., 2010a, Leskovec et al., 2010b] se sont intéressés à la sémantique de ces liens négatifs orientés qui peut être interprétée comme une relation d'ordre : c'est la théorie du statut. Ces auteurs ont alors proposé d'inférer le signe des relations en utilisant un classifieur basé sur des caractéristiques de voisinage, prenant en compte ces relations d'ordre.

Dans cette thèse, nous proposons d'utiliser des relations signées indirectes à travers le définition de mesures de similarité, en posant l'hypothèse que la sémantique des accords positifs (jugements positifs communs) et des accords négatifs (jugements négatifs communs) est différente.

Les tâches abordées

Notre travail s'appuie sur l'étude de deux grands domaines : les systèmes de recommandation et l'analyse des réseaux sociaux.

Les systèmes de recommandation

Les systèmes de recommandation trouvent de plus en plus leur place dans les services qui proposent de nombreux items à leurs utilisateurs (films, musiques, etc.) ; ils permettent de suggérer aux utilisateurs quels items ont le potentiel de les intéresser - qu'il s'agisse de musique, de films ou de n'importe quel produit proposé par des sites de streaming ou e-commerce. L'explosion du contenu disponible est telle que les systèmes de recommandation sont aujourd'hui indispensables pour garantir le succès d'une plate-forme en ligne. Dans le domaine académique, c'est en particulier grâce au challenge Netflix [Bennett and Lanning, 2007] que ces systèmes ont été popularisés.

Il existe deux principales familles de systèmes de recommandation [Bobadilla et al., 2013] : les systèmes de *filtrage par contenu* et les systèmes de *filtrage collaboratif*. Les systèmes de filtrage par contenu se basent sur la construction de profils explicites pour les items, basés sur leur contenu textuel (synopsis d'un film, résumé d'un livre) ainsi que sur les méta-données éventuellement à disposition comme par exemple le genre ou encore les noms des auteurs. L'idée est alors de recommander aux utilisateurs les items similaires à ceux qu'ils ont déjà évalués [Balabanović and Shoham, 1997]. Les systèmes de filtrage collaboratif construisent un profil par utilisateur et un profil par item en se basant sur l'historique des notes, de manière

à recommander les mêmes items aux utilisateurs dont le comportement est similaire [Schafer et al., 2007, Koren et al., 2009]. Le filtrage collaboratif donne d'excellents résultats lorsqu'il peut exploiter une très grande quantité de données [Dror et al., 2012] ou encore tirer profit du voisinage explicite [Crandall et al., 2008] ou de toute mesure de similarité calculée à l'aide de l'historique des notes comme l'indice de Pearson ou le cosinus [Bellogín, 2013].

Dans cette thèse, nous utilisons une méthode de filtrage collaboratif à base de modèle à facteurs latents qui consiste à représenter les utilisateurs et les items au sein d'un même espace latent de petite dimension. Puis, nous définissons des mesures de similarité dépendant de la polarité des jugements communs entre les utilisateurs et nous intégrons cette information de voisinage dans les modèles de filtrage collaboratif, de sorte que les utilisateurs proches selon ces mesures auront des représentations latentes similaires.

L'analyse des réseaux sociaux

Traiter les données issues des réseaux sociaux est une tâche encore bien plus complexe que la recommandation, car chaque utilisateur du réseau est susceptible de créer du nouveau contenu à chaque minute, ce qui fait que la taille de ces réseaux croît de manière exponentielle. Faire face à tout ce contenu demande alors des techniques de plus en plus sophistiquées pour capturer l'ensemble de l'information disponible. L'étude des relations sociales a donné naissance à une littérature extrêmement vaste [Aggarwal, 2011], appuyée par une théorie mathématique adaptée à la structure de ces réseaux : la théorie des graphes.

Les problématiques étudiées dans ce cadre sont assez nombreuses et se concentrent généralement sur la proximité entre les utilisateurs. Par exemple, deux utilisateurs qui partagent de nombreux voisins mais qui ne sont pas reliés entre eux sont plus susceptibles de l'être dans un avenir proche que deux utilisateurs pris au hasard dans le réseau. Parmi les grandes tâches d'analyse des réseaux sociaux, nous pouvons en distinguer trois en particulier. Tout d'abord, la détection de communautés qui s'intéresse à la structure des réseaux et à l'existence de sous-groupes d'utilisateurs davantage reliés entre eux qu'avec le reste du réseau (les communautés). Puis, la classification de noeuds dont le but est par exemple de compléter certaines données de profils utilisateur en leur attribuant certaines étiquettes de classe comme la catégorie socio-professionnelle, le sexe ou encore le niveau d'études. Enfin, la prédiction de liens qui consiste à identifier quels sont les liens qui devraient apparaître dans le futur, ce que de nombreuses plate-formes comme Facebook ou Twitter exploitent en suggérant de nouvelles relations sur la base des amis ou des personnes que les utilisateurs suivent.

Dans cette thèse, nous nous intéressons à la prédiction de la polarité des liens qui consiste à prédire le signe de liens déjà observés.

Les contributions

Dans ce manuscrit, nous nous intéressons aux données signées implicites issues des jugements des utilisateurs à propos d'items.

Première contribution

Jusque là, il était communément admis que les accords entre utilisateurs sont synonymes de similarité. Dans ce manuscrit, nous étudions l'hypothèse selon laquelle les accords pourraient ne pas avoir la même sémantique selon qu'ils concernent un item apprécié ou non. L'hypothèse de travail est que la sémantique d'un jugement dépend à la fois de la popularité de l'item concerné et du coût que représente l'évaluation de cet item pour les utilisateurs (certains notent tous les types d'items quand d'autres ne notent que ceux susceptibles de les intéresser). Il en résulte que les opinions positives ont une sémantique bien plus forte que les opinions négatives car les items qui ont reçu une note négative pourront être (1) moins satisfaisants que les autres items de la même catégorie ou (2) non pertinents vis-à-vis des goûts de l'utilisateur.

Notre première contribution consiste donc à analyser ce qui différencie la polarité des accords. En particulier, nous définissons des caractéristiques qui permettent de distinguer les utilisateurs qui partagent des opinions communes selon qu'ils aient apprécié les mêmes items (accords positifs) ou qu'ils n'aient pas apprécié les mêmes items (accords négatifs). Nous utilisons ensuite ces caractéristiques comme information de voisinage pour une tâche de filtrage collaboratif (via des méthodes de factorisation matricielle où utilisateurs et produits sont représentés dans un même espace latent). Ce travail a fait l'objet d'une publication dans une conférence nationale :

— **[Gauthier et al., 2014]** : Gauthier, L.-A., Piwowarski, B., and Gallinari, P. (2014). Filtrage collaboratif et intégration de la polarité des jugements. In *CORIA-CIFED 2014*

Seconde contribution

L'analyse de ces accords négatifs soulève cependant quelques problèmes sachant que tous les utilisateurs ne notent pas de la même manière et que les items ont des popularités distinctes. Par exemple, deux utilisateurs qui n'expriment que des avis positifs sur des items populaires ont plus de chance de partager des accords positifs. Inversement, deux utilisateurs qui ne s'intéressent qu'à des items peu évalués ont de moindres chances d'être en accord positif puisque ces items reçoivent en moyenne moins de notes positives que les items populaires. La sémantique d'un accord, qu'il soit positif ou négatif, va donc dépendre à la fois de l'item en question et des utilisateurs impliqués.

Notre seconde contribution consiste à prendre en compte les biais relatifs à chaque utilisateur et à chaque item. En particulier, nous proposons de mesurer la surprise apportée par un ensemble d'accords positifs ou négatifs. Nous étudions dans quelle mesure il y a un lien entre la proximité de deux utilisateurs (au sens des notes données) et les mesures proposées.

— **[Gauthier et al., 2015b]** : Gauthier, L.-A., Piwowarski, B., and Gallinari, P. (2015b). Polarité des jugements et des interactions pour le filtrage collaboratif et la prédiction de liens sociaux. In *CORIA 2015*

Troisième contribution

Dans une tâche d'analyse des réseaux sociaux, avoir recours à des données signées implicites peut devenir nécessaire lorsqu'aucun lien négatif n'est explicité par les utilisateurs. *Tang et al.* ont récemment proposé [Tang et al., 2015] un protocole permettant d'exploiter différentes sources d'information implicites afin de construire un graphe contenant à la fois les liens positifs observés dans le réseau et des liens négatifs inférés, et ce, pour l'exploiter en tant que données d'entraînement pour un classifieur. C'est une option qui s'avère particulièrement intéressante puisque la plupart des plates-formes en ligne ne permettent pas aux utilisateurs de définir explicitement les liens négatifs. Dans [Tang et al., 2015], les auteurs proposent d'inférer les liens négatifs utilisés en apprentissage à partir des évaluations des utilisateurs à propos du contenu d'autres utilisateurs (lorsqu'ils évaluent la pertinence ou l'utilité d'un contenu).

Pour notre troisième contribution, nous proposons d'étendre ce travail en utilisant les jugements des utilisateurs comme unique source d'information signée. Cette tâche nous semble plus adaptée aux données réelles puisque les jugements à propos d'items sont plus nombreux et ne concernent pas uniquement un type d'utilisateur (les auteurs de contenus). Nous proposons un modèle de prédiction de la polarité d'un lien. Ce travail a fait l'objet d'une publication dans une conférence internationale :

— **[Gauthier et al., 2015a]** : Gauthier, L.-A., Piwowarski, B., and Gallinari, P. (2015a). Leveraging rating behavior to predict negative social ties. In *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*

Le plan

Le manuscrit est organisé de la façon suivante. Dans la première partie nous faisons un état de l'art : le premier chapitre étudie les systèmes de recommandation et le second s'intéresse à l'analyse des réseaux sociaux. Dans la seconde partie, nous décrivons nos trois contributions à travers trois chapitres successifs ; le premier concerne l'analyse de la sémantique des jugements communs, le second prend en compte les biais utilisateur et item à travers leurs distributions de notes respectives, enfin le troisième chapitre exploite les résultats des deux précédents pour une tâche de prédiction de liens. Nous terminons ce manuscrit par une conclusion où nous discutons sur les perspectives ouvertes par ces recherches.

Première partie

État de l'art

1

Les systèmes de recommandation

Résumé

Les systèmes de recommandation regroupent un ensemble d'outils et de techniques permettant de suggérer, à partir d'informations de différentes natures, quels sont les items qu'un utilisateur pourrait apprécier. Ces systèmes peuvent exploiter le profil des items (filtrage par contenu), l'historique des notes (filtrage collaboratif) ou encore les relations d'amitiés explicitées par les utilisateurs dans un réseau social. Dans ce chapitre, nous présentons les différents modèles utilisés pour la recommandation. Nous nous sommes intéressés en particulier au filtrage collaboratif sur lequel nos travaux se basent et plus précisément sur les méthodes dérivées de la factorisation matricielle.

Sommaire

Introduction

Avec l'explosion du volume de données disponibles à travers l'ensemble du web, il est de plus en plus difficile pour l'utilisateur de trouver la bonne information ou le bon produit. Cette augmentation du contenu, en taille comme en variété, n'est alors plus vraiment perçue comme un bénéfice pour l'utilisateur qui se noie dans l'abondance. La multiplication des choix a l'effet inverse qu'escompté : il diminue ses chances de trouver un élément qui l'intéresse. Pourtant, maintenir la satisfaction des utilisateurs est un enjeu majeur avec la prolifération des services disponibles. La survie de la majorité de ces services passe donc par leur capacité à orienter l'utilisateur vers du contenu pertinent.

Les systèmes de recommandation sont d'ores et déjà incontournables lorsqu'il s'agit de sélectionner les items susceptibles de retenir l'attention de l'utilisateur, à l'instar de la *curation de contenu* (sélection des meilleurs contenus par des utilisateurs) devenue populaire ces dernières années, bien que cette dernière ne se fasse pas de manière automatique. Cette suggestion des meilleurs items va également au-delà de la simple satisfaction utilisateur, car les systèmes cherchent aussi à augmenter le trafic et le nombre d'items parcourus ; c'est le cas par exemple pour des sites de e-commerce qui incitent les clients à effectuer de nouveaux achats. Cet objectif de capter le trafic concerne également les moteurs de recherche.

Ces dernières années, l'étude de ces systèmes a engendré une vaste littérature et suscité beaucoup d'intérêt de la part de services en ligne, notamment depuis le challenge Netflix [Bennett and Lanning, 2007] qui a su démontrer l'importance et l'impact de bons modèles. Ces travaux ont en particulier mis en avant le filtrage collaboratif, une approche consistant à exploiter l'historique des notes de manière à recommander à partir des similarités de notation entre utilisateurs ou items [Schafer et al., 2007,Koren et al., 2009]. Une seconde approche basée sur le contenu (filtrage par contenu) a également été étudiée et s'appuie sur le profil des items obtenu à partir des contenus et méta-données disponibles (genres, auteurs, acteurs ou tout autre descripteur) afin de recommander aux utilisateurs les items aux profils similaires à ceux déjà appréciés [Balabanović and Shoham, 1997]. Au delà de ces deux cas d'école étudiés dans le milieu académique, il existe une multitude d'autres problématiques pour la recommandation, par exemple celles pour lesquelles on exploite les traces utilisateurs et diverses autres sources d'informations, comme les données démographiques ou relationnelles (réseaux sociaux) [Burke, 2002, Ricci et al., 2010].

La tâche de recommandation soulève de nombreuses difficultés parmi lesquelles nous trouvons notamment les biais de notation et le démarrage à froid. Les problèmes de biais concernent principalement deux phénomènes évoqués dans [Cremonesi et al., 2010, Pradel et al., 2012,Marlin et al., 2012]. D'une part, quelques items jouissent d'une certaine popularité (plus d'évaluations et une note moyenne plus élevée). D'autre part, tous les utilisateurs ne notent pas de la même manière et beaucoup n'évaluent que certains types d'items (ce qui mène notamment à une sur-représentation des notes positives). Il en ressort par exemple que les items populaires sont à la fois ceux qui sont les plus faciles à suggérer (statistiquement ceux qui plairont au plus de monde) et ceux qui fourniront une satisfaction minimale pour des utilisateurs en attente d'une recommandation plus fine, plus ciblée et plus pertinente. Le défi se situe donc surtout du côté des produits non populaires, un point qui s'avère délicat

tant sur la construction de modèles que sur l'évaluation de ces systèmes.

Le problème de démarrage à froid correspond au cas où il n'y a pas suffisamment d'informations disponibles pour la recommandation, un problème qui concerne les nouveaux items et les nouveaux utilisateurs arrivant dans le système. Pour les nouveaux items, il est nécessaire de se baser sur les méthodes de filtrage par contenu puisque l'absence d'historique ne permet ni la personnalisation, ni la recommandation par popularité. Quant aux nouveaux utilisateurs, l'absence d'historique empêche l'utilisation des techniques de filtrage collaboratif ou de filtrage par contenu ; il faut alors avoir recours à des données démographiques par exemple (pour s'appuyer sur des similarités de profil comme l'âge ou la catégorie socio-professionnelle), sans quoi la recommandation devient non-personnalisée et ne repose alors que sur la popularité des items. Certains services comme Netflix demandent à chaque nouvel utilisateur d'évaluer un certain nombre de films, mais une telle procédure intrusive est de fait moins justifiée pour un site de ventes en ligne par exemple.

Dans ce chapitre, nous présentons les diverses méthodes de recommandation proposées dans la littérature. En particulier, nous décrivons les deux principales approches qui sont le filtrage collaboratif et le filtrage par contenu. Ensuite, nous revenons sur les méthodes à base de modèles pour le filtrage collaboratif, parmi lesquelles nous trouvons les techniques dérivées de la factorisation matricielle qui exploitent l'historique des notes pour la construction d'un modèle de prédiction. Enfin, nous discutons de l'évaluation des modèles de recommandation et des différentes tâches étudiées.

1.1 Les familles de méthodes

Les systèmes de recommandation visent à suggérer à l'utilisateur un ensemble d'items de manière à maximiser sa satisfaction. De manière plus formelle, [Adomavicius and Tuzhilin, 2005] définissent cette satisfaction comme une *utilité*. Soient U et I l'ensemble des utilisateurs et l'ensemble des items, alors la fonction d'utilité est de la forme $r : U \times I \rightarrow R$ où R est un ensemble totalement ordonné. Plus l'item satisfait l'utilisateur et plus l'utilité associée est élevée. Dans le cas le plus courant, l'utilité est représentée par une note : R est alors l'ensemble des réels $\mathbb{R}$. Au final, la difficulté pour les systèmes de recommandation est de trouver l'ensemble des items qui maximisent cette utilité pour un utilisateur, alors même que celle-ci n'est pas connue pour la majorité des couples $U \times I$.

Lorsque l'on ne peut exploiter aucune information relative aux utilisateurs (profil ou historique de notes par exemple), la recommandation est non-personnalisée. Les méthodes les plus efficaces sont alors celles qui recommandent les items les plus populaires, c'est-à-dire ceux ayant une forte utilité pour un grand nombre d'utilisateurs. Cette prédiction repose sur une espérance de satisfaction supérieure à celle obtenue avec un item quelconque, étant donné que l'utilité moyenne d'un item populaire est supérieure à l'utilité moyenne d'autres items. En revanche, cette recommandation non-personnalisée atteint rapidement ses limites lorsque, par exemple, le nombre de catégories d'items est important. En effet, l'utilité moyenne n'est pas calculée sur l'ensemble des utilisateurs, mais sur un petit échantillon qui n'est souvent pas représentatif ; l'utilité moyenne augmente donc artificiellement car les items d'une thématique

donnée sont évalués principalement par les utilisateurs qui s'intéressent à cette thématique. Finalement, la diversité des goûts fait que la personnalisation devient nécessaire pour affiner l'inférence de la fonction d'utilité.

Afin de construire un profil personnalisé pour les utilisateurs, il est nécessaire de recourir à leur historique de notes ou de navigation. On distingue généralement deux sortes de retours utilisateur (en anglais *feedback*) : les retours explicites lorsque l'utilisateur émet directement un avis sur un item (par exemple une note) et les retours implicites issus principalement de l'analyse de sa navigation et de son comportement (ne nécessitant aucune action de sa part). Si les retours implicites sont plus nombreux et plus faciles à obtenir, ils sont également plus difficiles à interpréter puisqu'il est nécessaire d'attribuer un *score* à chaque comportement particulier. La sémantique associée à un comportement peut être différente selon les utilisateurs et il est donc parfois difficile de les exploiter.

Dans ce manuscrit, nous nous focalisons sur les retours explicites. Dans le contexte du filtrage collaboratif, il en existe de trois sortes [Ricci et al., 2010] :

— L'utilisateur évalue la pertinence de l'item en indiquant simplement s'il a aimé ou non l'item : une sortie binaire (+1/-1).

— L'utilisateur donne une note à l'item, par exemple entre 0 et 100.

— L'utilisateur laisse un avis textuel sur ce qu'il a pensé de l'item.

Dans la suite, nous nous intéressons aux retours explicites sous forme de notes, puisque cela correspond à la majorité des corpus à disposition.

Les systèmes de recommandation se regroupent en plusieurs grandes familles de méthodes selon la nature de l'information exploitée, parmi lesquelles on distingue généralement [Koren et al., 2009] :

— Le filtrage collaboratif : l'historique des notes est exploité pour identifier des similarités entre utilisateurs dans leur manière de noter, afin de recommander des items appréciés par d'autre utilisateurs similaires.

— Le filtrage par contenu : des profils explicites sont récoltés pour les items (genre, auteurs, synopsis ou résumé) afin de recommander aux utilisateurs des items aux profils similaires à ceux qu'ils ont déjà notés.

Dans cette section, nous décrivons les principes mis en oeuvre pour chacune de ces familles. Nous détaillerons des cas d'applications pour chacune de ces approches.

1.1.1 Filtrage collaboratif

Les techniques à base de filtrage collaboratif tirent profit de l'historique de notes pour trouver des similarités entre les utilisateurs, plus précisément des points communs dans leur manière de noter. L'idée est de recommander à un utilisateur donné les items qui ont été appréciés par des utilisateurs partageant un historique similaire : certains travaux parlent alors de recommandation *de personne à personne* [Schafer et al., 2001]. Cette technique a pour principal avantage de ne nécessiter aucune construction explicite de profil puisque seul

l'historique de notes est exploité pour comparer les utilisateurs ; elles sont très rapides à mettre en oeuvre et sont de fait très populaires dans la littérature.

Exemple : Nous pouvons illustrer cette approche de la manière suivante. Dans le tableau ci-dessous, cinq items sont évalués de manière *non exhaustive* par cinq utilisateurs. Chaque utilisateur a noté au plus trois produits et aucun couple n'a jugé le même ensemble de produits. Les notes observées sont indiquées en noir, elles sont entre 0 et 5.

	Évaluation des Items				
	i_1	i_2	i_3	i_4	i_5
u_1	●●●●○	-	-	●●●●●	●●○○○
u_2	●●○○○	●●●●○	★★★★	-	●●●●●
u_3	-	★★★★	●●●●○	●○○○○	●●●●●
u_4	-	●●●●●	●●●●○	-	★★★★★
u_5	●●●●●	-	-	★★★★★	●●○○○

Table 1.1 – *Un historique de notes (points noirs) exploité par un modèle de filtrage collaboratif pour prédire de nouvelles notes (étoiles rouges)*

Nous pouvons distinguer deux groupes de personnes dans cet exemple. D'un côté, nous avons les utilisateurs u_1 et u_5 qui ont évalué de manière similaire les items i_1 et i_5 ; à partir de ces notes, une approche collaborative va recommander à u_5 l'item i_4 apprécié par u_1. De l'autre côté, nous avons les utilisateurs u_2, u_3 et u_4 dont les items appréciés correspondent aux trois items i_2, i_3 et i_5. Mais, parce qu'aucun de ces utilisateurs n'a noté l'ensemble de ces trois items, il est alors facile de recommander pour chacun d'eux l'item qu'il n'a pas encore jugé.

Les premiers modèles de filtrage collaboratif ont été proposés par le *Xerox Palo Alto Research Center* en 1992 [Goldberg et al., 1992]. Ils nécessitaient cependant une communauté explicite d'utilisateurs : la recommandation selon l'historique des notes ne se faisait alors qu'entre les utilisateurs se connaissant déjà entre eux. C'est avec [Resnick et al., 1994] qu'est introduit le premier filtrage collaboratif *automatique* (pour la recommandation d'actualité sur GroupeLens) s'affranchissant de tout lien explicite entre utilisateurs.

Notation : Formellement, l'historique de notes se définit de la manière suivante : chaque utilisateur $u \in U = \{u_1, u_2, ..., u_n\}$ a noté un ensemble $I_u \subseteq I = \{i_1, i_2, ..., i_m\}$ d'item(s). Chacune des notes $r_{ui} \in I_u$ concernant un item peut se présenter sous forme binaire (aimer/ne pas aimer) ou réelle. Par exemple, sur le corpus GroupLens, les notes sont réelles et se situent entre 1 (mauvais) et 5 (bon).

La littérature distingue deux sous-familles de méthodes de filtrage collaboratif [Breese et al., 1998] : les méthodes basées sur la mémoire (*memory-based*), s'appuyant sur des mesures de similarité entre vecteurs de notes, et les méthodes à base de modèle (*model-based*) dont le but est d'apprendre un modèle de prédiction de notes à partir des données observées.

Méthodes basées sur la mémoire

La première sous-famille exploite directement l'historique des notes à travers des mesures de similarité entre utilisateurs. Ces similarités permettent alors d'appliquer des méthodes de k-plus-proches-voisins. Nous pouvons illustrer cela à l'aide d'un modèle simple [Adomavicius and Tuzhilin, 2005] : soit $\hat{r}_{ui}$ la note à prédire, $\overline{r}_u$ la moyenne des notes de l'utilisateur u et $\omega(u, v)$ la similarité calculée entre les deux utilisateurs u et v à partir de l'historique de leurs notes, alors :

$$\hat{r}_{ui} = \overline{r}_u + d \times \sum_{v \in V} \omega(u, v)(r_{vi} - \overline{r}_v) \tag{1.1}$$

où d est un facteur de normalisation et V le voisinage de u. Le poids ω peut être l'indice de Pearson [Resnick et al., 1994] ou le cosinus [Salton and McGill, 1986] par exemple.

Il est possible d'appliquer cette idée du point de vue des items [Ricci et al., 2010, Dror et al., 2012].

$$\hat{r}_{ui} = \overline{r}_i + d \times \sum_{j \in R(j)} \omega(i, j)(r_{uj} - \overline{r}_j) \tag{1.2}$$

avec $R(j)$ l'ensemble des k plus proches items de i (k un hyperparamètre) et $\overline{r}_j$ la moyenne des notes reçues par l'item i.

Méthodes à base de modèle

La seconde sous-famille, à base de modèle, exploite quant à elle l'historique de notes comme données d'apprentissage afin de construire un modèle de prédiction. Elle peut être vue simplement comme le calcul d'une espérance [Breese et al., 1998] :

$$\hat{r}_{ui} = \mathbb{E}(r_{ui}) = \sum_{r=1}^{max} P(r_{ui} = r | r_{uk}, k \in I_u) \times r \tag{1.3}$$

où I_u est l'ensemble des items évalués par u et max la valeur maximale que peut prendre une note r.

Les méthodes à base de modèle sont les méthodes les plus répandues et celles qui se sont montrées les plus efficaces lors de nombreuses compétitions [Bennett and Lanning, 2007, Dror et al., 2012]. Les plus étudiées dans la littérature sont celles à base de voisinage et, plus largement, celles à base de facteurs latents, en particulier depuis le challenge Netflix [Bennett and Lanning, 2007] qui a permis l'essor des techniques basées sur la factorisation matricielle. Les modèles à base de voisinage se concentrent en règle générale sur la recommandation d'items voisins à ceux déjà évalués par un utilisateur, c'est-à-dire des items qui ont reçu des notes équivalentes de mêmes utilisateurs. Les modèles à facteurs latents tentent quant à eux d'expliquer les notes des utilisateurs en regroupant ces derniers en ensembles d'utilisateurs notant de manière similaire [Koren et al., 2009] ; l'approche la plus commune consiste par exemple à factoriser la matrice de notes $user \times item$. Nous détaillons ces techniques dans la section 1.2.

Les limites rencontrées

Le filtrage collaboratif a été plébiscité par de nombreux travaux ces dernières années [Ricci et al., 2010, Koren et al., 2009, Schafer et al., 2007], notamment parce qu'il ne nécessite aucune connaissance particulière concernant les items ou les utilisateurs, ni même la construction de profil. Néanmoins, de bonnes performances ne sont généralement obtenues que lorsque les données sont en nombre suffisant, aussi bien pour chaque utilisateur que pour chaque item. En effet, puisque le filtrage collaboratif s'appuie sur l'historique de notes pour faire de nouvelles associations *utilisateur-item*, un nombre insuffisant de notes ne permettra pas de modéliser correctement le goût des utilisateurs ou la nature de l'item.

Ceci peut s'illustrer à travers les deux cas particuliers d'un nouvel utilisateur et d'un nouvel item arrivants dans le système :

— **Nouvel utilisateur :** Quand un nouvel utilisateur arrive, il ne possède pas d'historique. Dans ce cas, n'importe quel modèle de filtrage collaboratif est impuissant, il faut alors recourir à de la recommandation non-personnalisée (sélectionner des items parmi les plus populaires). Certaines techniques ou familles de méthodes permettent de remédier à cela. Il est par exemple possible de forcer l'utilisateur à évaluer un premier ensemble d'items, comme le font Netflix ou Apple Music. Il est également possible de demander à l'utilisateur de remplir certaines informations de profil.

— **Nouvel item :** Le cas du nouvel item est encore plus délicat parce qu'en l'absence d'évaluation utilisateur, il est tout simplement impossible de recommander cet item à qui que ce soit. En fait, tant que celui-ci n'a pas reçu un nombre suffisant de notes, il est difficile d'identifier à quel type d'utilisateur il se destine. Pour des services dans lesquels le nombre d'items évolue au fil du temps, il est nécessaire de prendre en compte de nouvelles sources d'information, comme le permettent notamment les méthodes à base de contenu.

C'est principalement pour limiter les problèmes de démarrage à froid que certains modèles intègrent des informations *implicites* pour combler le manque de données [Oard et al., 1998], par exemple en prenant en compte les items parcourus mais pas encore évalués par les utilisateurs [Marlin et al., 2012]. Une autre solution est d'avoir recours aux méthodes à base de connaissances. Cette approche s'appuie sur la connaissance précise et *a priori* des besoins des utilisateurs. De nombreux services proposent aujourd'hui de choisir, dès l'inscription, un nombre de catégories d'items correspondant aux attentes, de manière à proposer immédiatement des suggestions. Un exemple intéressant est proposé par [Burke, 2002] pour la recommandation de restaurant, où les utilisateurs peuvent remplir préalablement un questionnaire (voir l'illustration 1.1) afin de préciser ce qu'ils recherchent. Les plate-formes Netflix et Apple Music utilisent également ce genre de méthodes, en demandant à l'utilisateur de choisir les genres de films ou de musiques qu'il souhaite découvrir. Cette méthode issue de techniques de recherche d'information résout parfaitement le problème de démarrage à froid pour les utilisateurs, mais pas pour les items.

Figure 1.1 – *Première page du système de recommandation de restaurant Entree. Illustration de [Burke, 2002]*

1.1.2 Recommandation basée sur le contenu

Les systèmes de recommandation à base de contenu (*content-based*) s'appuient sur le profil des items pour construire une recommandation personnalisée. Ils s'inscrivent dans le contexte où les items sont décrits par un ensemble d'attributs qui les caractérisent, c'est-à-dire des méta-données sous la forme de balises, de valeurs numériques ou de texte que nous pouvons alors analyser. À partir de ces différents attributs, les modèles définissent des distances ou des similarités entre items, permettant alors de recommander aux utilisateurs des items proches de ceux qu'ils ont déjà appréciés.

Exemple : Pour illustrer l'approche par contenu, nous pouvons reprendre l'exemple précédent (tableau 1.1). Nous y ajoutons un nouvel item i'_1 qui n'a pas encore été évalué.

	Évaluation de films					
	Comédie		Action		Aventure	
	i_1	i'_1	i_2	i_3	i_4	i_5
u_1	●●●●○	★★★★	-	-	●●●●●	●●○○○
u_2	●●○○○	-	●●●●○	★★★★	★★★★★	●●●●●
u_3	-	-	★★★★	●●●●○	●○○○○	●●●●●
u_4	-	-	●●●●●	●●●●○	-	-
u_5	●●●●●	★★★★★	-	-	-	●●○○○

Table 1.2 – *Un historique de notes (ronds noirs) exploité par un modèle de filtrage par contenu pour prédire de nouvelles notes (étoiles rouges)*

Dans cet exemple, nous pouvons observer les différences avec l'approche collaborative (tableaux 1.1 et 1.2). Ici, le modèle n'essaye pas de recommander des films en comparant des utilisateurs aux goûts similaires, mais plutôt de recommander des films du même genre, c'est-à-dire ayant le même profil que les films déjà appréciés par l'utilisateur. Par exemple, le film i_4 sera recommandé aux utilisateurs qui ont apprécié i_5, ou encore l'item i_3 sera recommandé à ceux ayant aimé i_2. Quant à l'item i'_1, il illustre bien l'un des avantages de l'approche par contenu : un nouvel item qui n'a pas encore été évalué peut tout à fait être recommandé si son profil est similaire à celui d'items déjà présents et jugés par des utilisateurs actifs. i'_1 pourra être recommandé aux utilisateurs ayant apprécié l'item i_1.

Parmi les exemples de systèmes de filtrage par contenu, nous pouvons citer le *Music Genome Project* [Koren et al., 2009] qui appuie sa recommandation sur une représentation issue de l'analyse musicale. Ou encore le filtrage de groupe de discussion *NewsWeeder* [Lang, 1995] où chaque item (du texte) est représenté par un sac de mots (les attributs).

L'approche : Formellement, chaque item est défini par un ensemble d'attributs (genre, liste des acteurs/compositeurs, date, résumé, etc.). L'approche consiste alors à construire pour chaque utilisateur un vecteur de profil à partir des attributs des items qu'il a évalués. Le profil construit est alors une représentation des préférences de l'utilisateur permettant ainsi de lui associer un ensemble d'items pertinents. L'essentiel de cette approche repose donc dans la caractérisation des items - la recommandation s'appuyant sur celle-ci pour évaluer la similarité de contenu avec les items notés par les utilisateurs.

Ricci et al. [Ricci et al., 2010] décrivent deux approches pour exploiter les profils items. Une première approche consiste à quantifier directement la similarité entre les items pour ensuite recommander à un utilisateur les items les plus similaires à ceux qu'il a appréciés. La seconde approche consiste à évaluer la probabilité pour qu'un item donné soit apprécié ou non par l'utilisateur, sachant les attributs de cet item.

Modèle d'espace vectoriel basé sur les mots clés

Les systèmes basés sur le contenu exploitent largement les techniques de recherche d'information. Dans ce contexte, chaque item est considéré comme un document. En particulier, on utilise des mesures d'analyse fréquentielle de contenu dont la plus répandue est TF-IDF (*Term Frequency-Inverse Document Frequency*) [Salton, 1989]. Cela permet de pondérer la valeur d'un attribut de profil (un mot) selon l'importance qu'il a au sein du corpus, et ce, de la manière suivante :

— moins un attribut apparaît au sein du corpus, plus il est pertinent pour caractériser un item dans lequel il apparaît (hypothèse IDF)

— plus un attribut apparaît pour un item donné, plus cet attribut est pertinent pour caractériser cet item (hypothèse TF)

— la taille de contenu disponible pour un item n'impacte pas le score de chaque attribut (hypothèse de normalisation)

À partir du calcul du TF-IDF, on calcule pour un document d_j un poids $w_{k,j}$ pour chaque terme t_k rencontré :

$$w_{k,j} = \frac{\text{TF-IDF}(t_k, d_j)}{\sqrt{\sum_{s \in D} \text{TF-IDF}(t_s, d_j)^2}} \tag{1.4}$$

où D est l'ensemble des termes rencontrés.

Puisque le principe des méthodes à base de contenu consiste à recommander des items similaires à ceux déjà évalués par l'utilisateur, il suffit alors de définir une fonction de similarité entre deux items donnés. Le cosinus est souvent employé [Baeza-Yates et al., 1999] :

$$sim_{d_i,d_j} = \frac{\sum_k w_{k,i} \times w_{k,j}}{\sqrt{\sum_k w_{k,i}^2} \times \sqrt{\sum_k w_{k,j}^2}} \tag{1.5}$$

De récents travaux se sont intéressés à des techniques d'analyse de sentiments afin d'affiner la représentation vectorielle des items, comme [Lees-Miller et al., 2008] où les auteurs exploitent le contenu de Wikipedia afin d'améliorer la prédiction sur Netflix.

Modèle d'apprentissage des préférences utilisateur

Une autre approche consiste à apprendre un classifieur pour chaque utilisateur, le plus souvent un classifieur binaire. Dans [Sebastiani, 2002, Ricci et al., 2010] une classification naïve bayésienne estime si un item i est apprécié ou non (nous avons donc deux classes c_+ et c_-). Formellement, le modèle cherche à estimer la probabilité $P(c|i)$, c'est-à-dire la classe à laquelle appartient l'item i. La formule de Bayes nous donne :

$$P(c|i) = \frac{P(c) \times P(i|c)}{P(i)} \tag{1.6}$$

avec $P(c)$ la probabilité *a priori* de la classe c, $P(i|c)$ la probabilité d'observer l'item i sachant la classe c et enfin $P(i)$ la probabilité d'observer l'item i. Ce qui nous donne pour un item i donné la classe suivante :

$$C_i^* = argmax_{c_i} \frac{P(c) \times P(i|c)}{P(i)} = argmax_c P(c) \times P(i|c) \tag{1.7}$$

La probabilité $P(i|c)$ est difficile à calculer sachant le nombre limité d'observations et, notamment, le fait qu'un document ne soit généralement pas observé plus d'une fois. C'est pourquoi les documents sont décomposés en termes (et pour nous les items en caractéristiques). On utilise par exemple une approche *multinomiale* [McCallum and Nigam, 1998] dans laquelle on compte le nombre d'apparitions de chaque terme/caractéristique. Formellement,

$$P(c|i) = P(c) \prod_{t_k \in D} P(t_k|c)^{N(i,t_k)} \tag{1.8}$$

avec $N(i, t_k)$ le nombre d'occurrences du terme t_k dans pour l'item i, D le dictionnaire de termes.

Dans ce contexte, des études empiriques ont montré que les résultats étaient satisfaisants bien que l'hypothèse d'indépendance entre les termes au sein d'un item ne soit pas réellement respectée dans les faits [Billsus and Pazzani, 1996].

Un manque de sérendipité

Comme nous avons pu le voir dans l'exemple ci-dessus, le filtrage basé sur le contenu a l'avantage de ne pas dépendre exclusivement de l'historique de notes ; un item pourra donc être recommandé à un utilisateur sur l'unique base de son profil, même si cet item n'a pas encore été évalué par qui que ce soit. En revanche, un nouvel utilisateur ne pourra toujours pas obtenir de bonne recommandation en l'absence de notes, comme dans le cadre du filtrage collaboratif. Nous pourrons alors avoir recours aux méthodes à base de connaissances, comme nous l'avons vu précédemment, afin de limiter le problème de démarrage à froid.

Il est également possible d'utiliser des méthodes démographiques qui catégorisent les utilisateurs au travers des caractéristiques personnelles pour ensuite recommander en s'appuyant sur des informations démographiques. En d'autres termes, les utilisateurs appartenant à une même catégorie démographique (âge, catégorie professionnelle, niveau d'études, etc.) seront considérés comme ayant des goûts similaires [Pazzani, 1999]. Cette famille de méthodes ne bénéficie pas d'une riche littérature, mais a été particulièrement étudiée pour des applications marketing [Krulwich, 1997]. Ces méthodes peuvent également être utilisées pour le filtrage collaboratif.

Malheureusement, la recommandation basée sur le contenu souffre d'un autre problème. L'avantage de pouvoir recommander n'importe quel nouvel item n'est pas sans contrepartie : contrairement aux méthodes de filtrage collaboratif qui proposent des items appréciés par des utilisateurs aux notes similaires, les méthodes basées sur le contenu ne peuvent suggérer que des items aux mêmes attributs que ceux évalués par l'utilisateur. Ainsi, cette approche souffre d'un phénomène de *sur-spécialisation*, c'est-à-dire que les items recommandés se différencient peu de ceux déjà jugés. Les modèles n'ont donc pas la capacité de *surprendre* l'utilisateur avec des items différents (une caractéristique appelée *serendipité*), en particulier pour les nouveaux utilisateurs ayant peu ou pas de note.

1.2 Les modèles à facteurs latents pour le filtrage collaboratif

Dans nos travaux, nous utilisons des méthodes à base de modèle à facteurs latents. Cette approche consiste à représenter et expliquer les notes observées à travers une représentation des utilisateurs et des items dans un espace de petite dimension appelé *espace latent*. Cette technique permet de réduire l'espace de représentation des données, ce qui est particulièrement important pour les systèmes de recommandation où la matrice de notes est généralement très grande et creuse, c'est-à-dire où une grande partie des valeurs sont manquantes. Cette réduction de dimension repose alors sur l'idée que les préférences des utilisateurs sont influencées par un nombre limité de facteurs qu'il s'agit alors d'apprendre. Elle permet d'obtenir une meilleure capacité de généralisation notamment parce que les distances entre les points de cet

espace sont davantage significatives [Ricci et al., 2010], évitant notamment le phénomène de sur-apprentissage.

La construction de cet espace se fait en exploitant l'historique de notes comme données d'apprentissage via des modèles comme les réseaux de neurones [Salakhutdinov et al., 2007], l'allocation de Dirichlet latente (*Latent Dirichlet Allocation [LDA]*) [Blei et al., 2003] ou encore des techniques de factorisation basées sur la décomposition en valeurs singulières (*Singular value decomposition [SVD]*) [Golub and Reinsch, 1970,Deerwester et al., 1990]. Nous décrivons dans cette section une méthode de factorisation matricielle qui permet de construire les représentations des utilisateurs et des items. Puis, nous introduisons les méthodes pour prendre en compte les informations implicites et l'information de voisinage sur lesquelles nos travaux s'appuient pour intégrer les relations sociales signées.

1.2.1 Techniques de factorisation matricielle

En recommandation, la factorisation matricielle représente les items et les utilisateurs sous forme de vecteurs au sein d'un même espace latent $\mathbb{R}^n$ de sorte que la note d'un utilisateur u à propos d'un item i est donnée par le produit scalaire entre la représentation de l'utilisateur et celle de l'item. Pendant la phase d'apprentissage au cours de laquelle est construite cette représentation latente, il est possible d'exploiter d'autres sources d'informations, comme les interactions implicites (navigation, clics ou requêtes utilisateurs), ou encore la chronologie des évaluations (appelé *horodatage*) [Koren et al., 2009].

Décomposition en valeurs singulières

La décomposition en valeurs singulières (SVD) est couramment utilisée pour la réduction de dimension dans divers problèmes d'exploration de données et de recherche d'information. C'est une technique d'approximation matricielle de faible rang qui s'appuie sur un théorème selon lequel n'importe quelle matrice peut s'écrire sous la forme USV^T où S est une matrice diagonale contenant les valeurs singulières (le poids de chaque concept) et U,V sont des matrices orthogonales d'ordre m et n. Formellement, cela consiste à trouver les matrices U, S et V tel que :

$$min_{R \in \mathbb{R}^{m \times n}} \|R - USV^T\|_F^2 \tag{1.9}$$

où R est notre matrice de notes observées que l'on cherche à approximer et $\|\cdot\|_F^2$ la norme de Frobenius.

La solution exacte correspond à celle où le nombre de valeurs singulières est égal au rang de la matrice R. Mais, une approximation peut également être générée avec une dimension de l'espace latent fixée à une valeur k inférieure à ce rang : c'est une décomposition en valeurs singulières approchée [Eckart and Young, 1936]. Dans ce cas, on garde les k plus grandes valeurs singulières. La matrice de notes est alors approximée par la matrice M_k de rang k de sorte que l'on ait : $M_k = U_k S_k V_k^T$.

Une approche directe pour cette décomposition en valeurs singulières consiste à calculer les valeurs propres et les vecteurs propres des matrices carrées RR^T ou $R^T R$; la matrice U_k

(respectivement V_k) contient alors les k vecteurs propres associés aux k plus grandes valeurs propres de la matrice RR^T (resp. R^TR). Le principal inconvénient de cette méthode est qu'elle n'est pas appropriée pour les matrices creuses. En effet, il faut dans ce cas compléter les données manquantes par des valeurs neutres (par exemple la note 0) de manière à rendre la matrice dense et exploitable [Sarwar et al., 2000]. La quantité de valeurs ajoutées rend également cette technique non envisageable pour de gros jeux de données.

De nombreux travaux récents [Koren et al., 2009, Canny, 2002, Koren, 2008, Paterek, 2007, Salakhutdinov et al., 2007] ont contourné ce problème en modifiant la fonction de coût (1.9) pour apprendre uniquement sur les observations. Autrement dit, il s'agit de minimiser une fonction de coût égale à la somme des erreurs empiriques sur les données d'entraînement constituées d'une partie de l'historique à disposition. Formellement, la fonction objectif est définie de la manière suivante :

$$min_{q_i,p_u \in \mathbb{R}^n} \sum_{(u,i) \in J} (r_{ui} - q_i^T p_u)^2 \tag{1.10}$$

sur l'ensemble des couples J de notes (u, i) observées.

Prise en compte des biais utilisateur et item

L'un des points les plus délicats rencontrés dans les systèmes de recommandation provient de l'existence de biais utilisateur et item. Pour bien comprendre cela, il suffit de revenir à un système non-personnalisé qui suppose alors qu'un item ayant une forte *utilité* pour un grand nombre d'utilisateurs devrait satisfaire n'importe quel utilisateur actif mieux qu'un item choisi au hasard. Derrière cette approche se cache l'idée que les items ne sont pas tous évalués de la même manière et que certains sont plus populaires que d'autres : ils obtiennent des notes plus nombreuses et plus favorables, donnant alors une utilité moyenne plus élevée. C'est la notion de *biais item*. Ce qui est vrai pour les items l'est également pour les utilisateurs qui ne perçoivent pas le barème de notation de la même manière. Certains notent plus favorablement, d'autres n'évaluent que certains items. Cela correspond au *biais utilisateur* [Funk, 2006].

En ce qui concerne les données réelles, il se trouve qu'une partie des variations des notes observées provient des biais utilisateur et item [Koren et al., 2009] et pour en tenir compte, il est possible de modifier l'équation (1.10) en définissant la note prédite par :

$$\hat{r}_{ui} = \mu + b_i + b_u + q_i^T p_u \tag{1.11}$$

où μ correspond à la moyenne des notes sur l'ensemble du corpus et b_i (respectivement b_u) est la différence entre les notes de i (resp. u) et la moyenne du corpus.

La fonction objectif (1.10) devient alors dans ce contexte :

$$min_{q_i,p_u \in \mathbb{R}^n} \sum_{(u,i) \in T} (r_{ui} - \mu - b_i - b_u - q_i^T p_u)^2 \tag{1.12}$$

Optimisation

Le manque de données peut rapidement poser problème. Résoudre l'équation (1.12) est un problème *inverse mal posé*, c'est-à-dire un problème de minimisation dans lequel les paramètres sont identifiés à partir d'observations et dont la solution n'est pas une fonction continue des données. En d'autres termes, le fait que la matrice soit particulièrement creuse (i.e. le nombre de notes est faible) conduit au *surapprentissage* puisque chaque nouvelle donnée intégrée peut considérablement modifier les paramètres du modèle. Pour pallier ce phénomène qui pénalise la généralisation et donc la recommandation, un terme de régularisation est utilisé [Koren et al., 2009] :

$$min_{q_i,p_u \in \mathbb{R}^n} \sum_{(u,i)\in T} (r_{ui} - \mu - b_i - b_u - q_i^T p_u)^2 + \lambda(\|q_i\|^2 + \|p_u\|^2 + b_i^2 + b_u^2) \tag{1.13}$$

avec λ est le paramètre permettant de contrôler l'importance de la régularisation. Plus le terme est élevé ou moins l'utilisateur n'a de notes, plus les vecteurs sont de norme faible. Un utilisateur n'ayant qu'une seule note aura un vecteur proche de 0 et donc aura des notes proches de la moyenne.

Les méthodes de type descente de gradient sont des méthodes adaptées pour résoudre ce type de problème. D'un point de vue optimisation, on utilise souvent soit une descente stochastique soit une descente *par lot*. Pour la descente de gradient stochastique (ou *séquentielle*), chaque couple $(u, i) \in T$ de l'ensemble d'entraînement est considéré de manière indépendante et aléatoire pour la mise à jour des vecteurs p_u et q_i associés. Pour la descente *par lot*, les exemples d'entraînement sont parcourus par paquets, par exemple en mettant à jour le vecteur de l'utilisateur une fois parcouru l'ensemble de son historique.

Formellement, en posant $e_{ui} = r_{ui} - \mu - b_i - b_u - q_i^T p_u$ l'erreur pour le couple (u, i), la descente de gradient stochastique met à jour chaque vecteur p_u et q_i de la manière suivante [Funk, 2006]. Pour chaque observation (u, i) :

$$q_i \leftarrow q_i + \gamma(e_{ui} \cdot p_u - \lambda \cdot q_i)$$

$$p_u \leftarrow p_u + \gamma(e_{ui} \cdot q_i - \lambda \cdot p_u)$$

$$b_u \leftarrow b_u + \gamma(e_{ui} - \lambda \cdot b_u)$$

$$b_i \leftarrow b_i + \gamma(e_{ui} - \lambda \cdot b_i)$$

alors qu'une descente par lot alternée procédera de la façon suivante :

$$q_i \leftarrow q_i + \gamma(\sum_u e_{ui} \cdot p_u - \lambda \cdot q_i)$$

$$p_u \leftarrow p_u + \gamma(\sum_i e_{ui} \cdot q_i - \lambda \cdot p_u)$$

$$b_u \leftarrow b_u + \gamma(\sum_u e_{ui} - \lambda \cdot b_u)$$

$$b_i \leftarrow b_i + \gamma(\sum_i e_{ui} - \lambda \cdot b_i)$$

La facilité d'implémentation, la rapidité d'apprentissage et les très bonnes performances sont à l'origine de la popularité des techniques de descente de gradient stochastique parmi les systèmes de recommandation de la littérature. Nous les retrouvons d'ailleurs massivement utilisées dans les challenges Netflix [Bennett and Lanning, 2007] et Yahoo ! KDD Cup 2011 [Dror et al., 2012].

1.2.2 Exploitation d'informations supplémentaires

Dans beaucoup de situations, l'historique des notes n'est pas suffisamment fourni, que ce soit pour un utilisateur donné ou pour un item. On pense notamment aux situations où l'évaluation d'un item présente un coût pour l'utilisateur (en temps ou en argent notamment). Sur un site de e-commerce par exemple, les produits onéreux ont bien moins de retours utilisateurs. Dans ces situations particulières, certains modèles collectent d'autres données *implicites*, notamment à partir de l'observation du comportement des utilisateurs (historique de navigation ou requêtes passées).

Chaque item i possède un vecteur caractéristique x_i associé aux préférences implicites, de sorte qu'un utilisateur ayant exprimé une préférence implicite pour un ensemble $N(u)$ d'items est caractérisé par le vecteur de préférences suivant :

$$\sum_{j \in N(u)} x_j \tag{1.14}$$

Les informations implicites sont alors intégrées en ajoutant les vecteurs de préférences à la représentation latente de l'utilisateur [Koren et al., 2009], ce qui revient à dire que sa représentation est modifiée par un ensemble de vecteurs dans l'espace latent :

$$\hat{r}_{ui} = \mu + b_i + b_u + q_i^T (p_u + |N(u)|^{-1/2} \sum_{j \in N(u)} x_j) \tag{1.15}$$

À partir de là, il est possible d'exploiter plusieurs sources d'information implicite. Par exemple le voisinage de l'item, c'est-à-dire les items similaires susceptibles de recevoir la même note de la part d'un même utilisateur (sur la base de l'historique de notes). C'est la solution retenue par de nombreux modèles, par exemple lors du challenge Yahoo ! KDD Cup 2011 [Dror et al., 2012].

D'autres familles de techniques exploitent le voisinage de l'utilisateur, c'est-à-dire que ses préférences seront la moyenne pondérée de l'écart de notation de ses voisins. Formellement, x_j ($j \in N(u)$) correspond à un vecteur d'un utilisateur similaire. Ce voisinage pourra être obtenu grâce aux interactions explicites, comme dans le cas de graphes sociaux, soit inféré au sens d'une certaine métrique ou similarité dans l'espace des utilisateurs [Herlocker et al., 1999].

Enfin, des méthodes combinent ces deux idées comme par exemple [Jamali and Ester, 2010, Delporte et al., 2013, Yang et al., 2012]. L'information de voisinage est particulièrement intéressante du fait que l'utilisateur est rarement disposé à fournir autant de notes que ne peut

en fournir l'ensemble des différents voisins (item ou utilisateur). Parmi les travaux exploitant ce voisinage, [Koren, 2008] introduit le premier modèle à facteurs latents où la pondération entre les utilisateurs ou les items voisins est apprise lors de la factorisation matricielle. En ne considérant aucune autre information *implicite* et en se plaçant dans le cadre d'un voisinage item-item, ce modèle se définit de la manière suivante :

$$\hat{r}_{ui} = \mu + b_i + b_u + q_i^T p_u + |R^k(u;i)|^{-1/2} \sum_{j \in R^k(u;i)} \omega_{ij}(r_{uj} - \mu - b_j - b_u) \tag{1.16}$$

avec $R^k(u;i)$ l'ensemble des k-plus proches items de i qui ont été notés par u.

D'autres approches ont été proposées pour l'exploitation du voisinage, notamment [Yang et al., 2012] qui exploitent les liens de confiance/méfiance déclarés par des utilisateurs sur le contenu d'autres utilisateurs. Les auteurs sont partis du principe que la confiance de u à propos de l'utilisateur v est donnée par $p_u^T p_v$ (qui mesure une similarité entre les deux utilisateurs). De la même manière que $q_i^T p_u$ évalue l'intérêt de l'utilisateur u vis-à-vis de l'item i. Ils proposent alors un modèle dans lequel à la fois les notes sur les items et la confiance entre utilisateurs sont appris, et ce, de manière à exploiter les interactions sociales signées pour mieux modéliser le processus d'évaluation.

Pour prédire la note $\hat{r}_{ui}$, on suppose que l'utilisateur prend une décision en suivant une marche aléatoire dans laquelle il a une probabilité p de suivre ses propres goûts $q_i^T p_u$ et une probabilité $(1-p)$ de suivre l'avis $p_v^T q_i$ de l'un de ses voisins (avec la confiance $p_u^T p_v$) [Yang et al., 2012]. Soit :

$$\hat{r}_{ui} = q_i^T [pI + (1-p)(trM_p^{-1})M_p] p_u \tag{1.17}$$

avec $M_p = \sum_u p_u p_u^T$ (matrice carrée du même ordre que p_u). Le facteur (trM_p^{-1}) permet de normaliser M_p afin d'obtenir une matrice stochastique.

De la même manière, le lien de confiance $t_{uv} \in \{-1, 1\}$ entre deux utilisateurs u et v est déterminé selon une marche aléatoire, avec une probabilité p de suivre la confiance $p_u^T p_v$ entre u et v, et une probabilité $(1-p)$ de prendre en compte la concordance entre le jugement $p_u^T q_i$ de u et le jugement $p_v^T q_i$ de v vis-à-vis d'un item i :

$$\hat{t}_{uv} = p_u^T [pI + (1-p)(trM_q^{-1})M_p] p_v \tag{1.18}$$

avec $M_q = \sum_i q_i q_i^T$.

Afin d'apprendre les paramètres du modèle (p_u, q_i), L'article [Yang et al., 2012] propose alors d'utiliser la fonction objectif suivante :

$$min \lambda_y \sum_{(u,i)} (r_{ui} - \hat{r}_{ui})^2 + \lambda_c \sum_{(u,i)} (t_{uv} - |\hat{t}_{uv}|)^2 + \lambda_I \|q_i\|^2 + \lambda_U \|p_u\|^2 \tag{1.19}$$

Les méthodes proposées dans cette section permettent de prendre en compte des sources additionnelles pour l'apprentissage des représentations latentes des utilisateurs et des items. Nos travaux s'inscrivent dans cette perspective en exploitant les informations locales de voisinage utilisateur à l'aide d'un modèle similaire à celui défini par l'équation (1.16) où le w entre les utilisateurs est une fonction prenant en compte la polarité des accords entre les utilisateurs.

1.3 L'évaluation

Alors que les services en ligne peuvent rapidement mesurer l'impact de leurs systèmes en mesurant par exemple l'augmentation du trafic ou des ventes (le plus souvent sur un panel d'utilisateurs), il est bien plus difficile dans un cadre théorique de mesurer le retour de satisfaction et de comparer différents modèles. Dans la situation où il n'est pas possible de vérifier les performances en temps réel (expériences *en-ligne*), il est alors indispensable de mettre en place diverses métriques afin de mesurer si un modèle a un impact positif sur les performances. Nous rentrons alors dans le cadre des expériences *hors-ligne*.

Pour le filtrage collaboratif, les performances sont mesurées via un ensemble de notes récoltées. En particulier, pour les modèles décrits dans la section précédente, un ensemble de notes est utilisé pour l'apprentissage et un autre ensemble sert alors à évaluer les capacités de généralisation. En collectant les données des utilisateurs à un instant donné, cela permet d'effectuer à moindre coût de multiples expériences avec différents paramètres. La majorité des systèmes présents dans la littérature s'évaluent de cette manière, car il s'agit d'un protocole permettant une comparaison rapide et immédiate des performances.

Cette approche limite cependant les méthodes et les critères d'évaluation des performances et repose sur l'hypothèse forte que le comportement des utilisateurs, observé à travers leur historique des notes, ne changera pas entre la collecte et le déploiement du système. De plus, les choix qui sont fait au cours de la collecte pourront influencer les performances ; la collecte elle-même pose un certain nombre de problèmes notamment parce que les systèmes de notation sont sujets à de forts biais et qu'une collecte partielle peut tout à fait en apporter de nouveaux.

S'agissant des biais déjà présents (évoqués dans la section 1.2.1), nous pouvons distinguer deux sources différentes : (1) les biais items et (2) les biais utilisateurs. Le premier type prend en compte le fait que tous les items ne sont pas évalués selon les mêmes critères ou de la même manière. Le second porte sur la différence de perception des utilisateurs sur l'échelle de notes. Ces deux sources de biais sont notamment étudiées dans [Cremonesi et al., 2010, Pradel et al., 2012, Marlin et al., 2012]. Ils sont alors la source de deux biais majeurs :

— le biais de popularité : le nombre de notes reçues par un item peut varier fortement selon la nature de celui-ci.

— le biais de positivité : Certains utilisateurs ne noteront que les items qu'ils trouvent pertinents, alors que d'autres noteront tous les types d'items rencontrés.

Ces deux biais font d'une part que les notes positives sont sur-représentées, d'autre part que cette sur-représentation ne se fait pas de manière uniforme entre les items, notamment les items populaires pour lesquels nous n'observerons pas cette sur-représentation.

Comme le souligne [Ricci et al., 2010], la collecte des notes afin de construire les ensembles d'apprentissage et de test doit prendre en compte les différents biais sur les données. Par exemple, sélectionner les notes de manière aléatoire tout en gardant l'ensemble des utilisateurs et des items rend la matrice de notes observées très creuse, ce qui peut favoriser certains modèles. À l'inverse, une sélection plus minutieuse des utilisateurs et/ou des items, par exemple en ne gardant que ceux dont l'historique de notes est le plus fourni, peut détério-

rer les performances de généralisation en apportant de nouveaux biais auxquels les systèmes devront faire face.

Finalement, parmi les méthodes pratiquées, la méthode qui semble la plus pertinente consiste à décomposer l'ensemble des notes via l'horodatage. Par exemple, pour la construction de l'ensemble d'apprentissage et de l'ensemble de test, il suffit de dire qu'avant l'instant t, les notes observées sont utilisées pour l'apprentissage et que l'évaluation se fait à partir des notes observées après l'instant t. De cette manière, on préserve la nature des données, mais également les biais cités plus haut. Il est alors possible de contrôler ces biais au niveau des modèles, bien que la tâche demeure difficile [Ricci et al., 2010]. Ce découpage est celui qui a été utilisé pour la construction des corpus exploités dans les chapitres de contributions.

1.3.1 Tâches et mesures

Le critère de performance peut varier selon la tâche considérée : soit le modèle doit être capable d'inférer pour n'importe quel utilisateur la note sur un item qu'il n'a pas encore évalué, soit le modèle doit pouvoir prédire un ensemble ordonné d'items maximisant l'utilité pour cet utilisateur et en particulier, ordonner les items les plus pertinents en haut de la liste proposée à l'utilisateur. Dans le premier cas, le critère de performance consiste à minimiser la somme des erreurs empiriques sur l'ensemble des notes observées (ce qu'optimise la fonction objectif 1.13). Cependant, cette approche ne mesure finalement que la capacité d'un modèle à évaluer précisément les notes des utilisateurs sur certains items, non à anticiper lesquels seraient les plus pertinents. Le second cas a été popularisé ces dernières années parce qu'il répond à une évaluation plus proche de la réalité. Il évalue la capacité du modèle à identifier quels sont les items qu'il faut recommander en priorité. Des approches similaires sont utilisées en recherche d'information pour ordonner les items selon leur pertinence vis-à-vis d'une requête paticulière.

La prédiction de notes

La tâche de prédiction de notes a été la première étudiée, popularisée notamment par le challenge Netflix [Bennett and Lanning, 2007] à travers la mesure RMSE (pour *Root Mean Squared Error*). Les mesures utilisées consistent à mesurer l'erreur empirique entre des notes de test et celles prédites par les modèles. Il existe deux principales mesures d'erreur de prédiction qui sont :

- Root Mean Squared Error (RMSE) :

$$RMSE = \sqrt{\frac{\sum_{u,i \in Test} (r_{ui} - \hat{r}_{ui})^2}{|Test|}} \tag{1.20}$$

où r_{ui} est la note attribuée par l'utilisateur u à l'item i sur un ensemble de $Test$.

- Mean Absolute Error (MAE) :

$$MAE = \frac{\sum_{u,i \in Test} |r_{ui} - \hat{r}_{ui}|}{|Test|} \tag{1.21}$$

La différence entre les deux mesures est que la première pénalisera davantage les fortes erreurs.

Bien que ces mesures soient de moins en moins utilisées au profit des critères d'ordonnancement, elles restent toutefois encore très répandues puisqu'elles sont faciles à calculer et à interpréter. De plus, une valeur nulle indique également un ordonnancement parfait.

L'ordonnancement

Le calcul de la somme des erreurs quadratiques sur un ensemble de test permet uniquement de mesurer à quel point la modélisation peut expliquer les notes. Cependant, un bon système de recommandation doit surtout savoir ordonner les items selon leur pertinence, tout particulièrement pour ceux ayant les notes prédites les plus hautes. En effet, l'objectif d'un tel système consiste avant tout à cibler un nombre très limité d'items pertinents afin de retenir l'utilisateur. Pour cette tâche, des mesures de recherche d'information comme le rappel ou la précision sont particulièrement bien adaptées. Pour définir ce qu'est un item pertinent ou non, nous définissons au préalable un seuil de notation à partir duquel un item est considéré pertinent. Par exemple, sur un corpus où les utilisateurs peuvent attribuer des notes entre 0 et 10, nous pouvons considérer comme pertinent tous les items qui ont reçu une note supérieure à 5. L'ensemble des items pertinents pour l'utilisateur est noté I^+.

Sachant la liste ordonnée L_k des k meilleurs items renvoyés par le modèle, on peut calculer le rappel et la précision au rang k.

- Précision au rang k (P@k) :

$$P@k = \frac{|L_k \cap I^+|}{|k|} \tag{1.22}$$

- Rappel au rang k (R@k) :

$$R@k = \frac{|L_k \cap I^+|}{|I^+|} \tag{1.23}$$

La précision à k mesure la proportion d'items pertinents parmi les items renvoyés par le modèle. Le rappel à k mesure la proportion d'items pertinents retrouvés par le modèle (parmi les items pertinents observés sur l'ensemble de test).

À partir du calcul de la précision, nous pouvons calculer la précision moyenne correspondant à la moyenne des précisions calculées au rang des items pertinents. Par exemple, si dans la liste ordonnée L_k se trouvent deux items pertinents au rang 3 et 5, alors la précision moyenne calcule la moyenne entre la précision à 3 et la précision à 5. Formellement :

- Précision moyenne (AP) :

$$AP = \frac{\sum_{i=1}^{k} \mathbb{1}(i \in I^+) P@i}{|I^+ \cap L^k|} \tag{1.24}$$

avec $\mathbb{1}(i \in I^+) = 1$ si l'item i de la liste L_k est pertinent.

Ceci nous permet alors de calculer la précision moyenne sur l'ensemble des utilisateurs ; c'est le score MAP (*Mean Average Precision*), très utilisé pour mesurer les performances des systèmes de recherche d'information.

Le rappel, la précision et le score MAP sont les métriques les plus populaires, mais il existe d'autres mesures d'ordonnancement, parmi lesquels on retrouve l'AUC qui mesure la probabilité de placer une élément pertinent au-dessus d'un élément non pertinent, soit :

- AUC (*Area Under the Curve*) :

$$AUC = \frac{\sum_{i \in I^+} \sum_{j \in I^-} \mathbb{1}(f(i) > f(j))}{|I^+||I^-|} \tag{1.25}$$

avec f la fonction de score et $\mathbb{1}(x) = 1$ si x est vérifiée.

Cette mesure a pour principal inconvénient de ne pas prendre en compte le rang des paires *(pertinent, non pertinent)* mal ordonnées. Une paire mal ordonnée en bas de liste a donc autant d'influence sur le score qu'une paire mal ordonnée en haut de la liste. Or, l'objectif de la recommandation est avant tout de bien ordonner le haut de la liste. Au final, lorsque le nombre d'éléments pertinents est faible par rapport aux éléments non pertinents, comme pour la recommandation, le score MAP est davantage approprié.

Pradel et al. se sont penchés sur l'impact des biais positivité et popularité sur cette mesure AUC. Il s'avère que cette mesure favorise les items ayant reçu peu de notes, au détriment des items populaires (puisque les notes manquantes sont ignorées). À l'inverse, il est possible de corriger ce biais en pénalisant les items peu notés, en considérant les notes manquantes comme négatives (noté AUC^{AMAN}, pour *All Missing As Negatives*). Dans ce cas, cela accentue l'effet popularité et favorise alors les modèles les plus simples, c'est-à-dire ceux recommandant les items selon leur succès. *Pradel et al.* ont alors proposé [Pradel et al., 2012] une méthodologie afin d'optimiser un modèle entre valeurs optimales pour l'AUC et AUC^{AMAN}, de manière à ce que les biais positivité et popularité se compensent.

Conclusion

Dans ce chapitre, nous avons étudié les systèmes de recommandation. Leur popularité tient au fait que le succès de nombreux services sur internet en découle. Cibler les bons items permet d'augmenter à la fois la satisfaction de l'utilisateur et le nombre d'items consultés, ce qui est bon à la fois pour l'utilisateur et pour le fournisseur de service.

Nous avons décrit deux des approches les plus étudiées dans la littérature : le filtrage par contenu et le filtrage collaboratif. La première tâche consiste à exploiter le contenu de profil disponible sur chaque item afin de recommander aux utilisateurs des items qui vont correspondre à ceux qu'ils ont évalués. La seconde tâche repose quant à elle sur l'historique des notes et s'appuie sur les similarités de notation entre les utilisateurs (c'est-à-dire le fait d'avoir évalué les mêmes items de la même manière) afin de définir les items les plus pertinents.

En pratique, le filtrage collaboratif a été largement étudié et adopté par la communauté académique, notamment lors de différents challenges comme Netflix [Bennett and Lanning,

2007] ou Yahoo ! KDD Cup 2011 [Dror et al., 2012]. De nombreux travaux ont été proposés dans la littérature [Ricci et al., 2010], notamment les méthodes à base de modèle où utilisateurs et items sont représentés par des vecteurs caractéristiques au sein d'un espace latent. Entre autres, les approches dérivées de la factorisation matricielle ont reçu beaucoup d'attention grâce aux performances et à la rapidité d'exécution des algorithmes, notamment via les méthodes de type descente de gradient

Nous avons également présenté la manière d'évaluer ces systèmes. À la différence des services en ligne qui peuvent évaluer leurs modèles directement sur un panel d'utilisateurs (expériences *en-ligne*), la littérature scientifique s'évalue plus généralement à partir d'observations déjà collectées, c'est-à-dire de manière *hors-ligne*. Dans ce cas, un ensemble d'observations est utilisé pour l'apprentissage du modèle et l'autre partie des observations sert alors comme données de test. Les modèles peuvent ensuite se comparer sous deux aspects : la prédiction de notes ou l'ordonnancement. La première approche évalue les systèmes de recommandation dans leur capacité à prédire les notes manquantes, ce qui a notamment été utilisé pour le challenge Netflix. On évalue alors l'erreur empirique sur les observations de l'ensemble de test. La seconde approche évalue la capacité des modèles à cibler et ordonner un nombre limité d'items pertinents. Ce sont alors des métriques de recherche d'information qui sont utilisées, comme le rappel ou la précision par exemple.

2

L'analyse des réseaux sociaux

Résumé

L'analyse des réseaux sociaux est un sujet qui répond à une problématique sociologique dans laquelle on étudie diverses relations entre les utilisateurs : collaboration, amitié, partage de contenu, communauté d'intérêt, etc. Ce domaine de recherche s'est très largement développé depuis l'avènement du web et l'explosion de la quantité de contenus partagés. La généralisation des usages a donné naissance à de nombreuses tâches parmi lesquelles la détection de communautés, la classification de noeuds et la prédiction de liens. En plus des relations d'amitié, de suivi ou de partage, de nouveaux réseaux permettent aujourd'hui de collecter des liens négatifs entre les utilisateurs correspondant à des relations de défiance, de désaccord ou d'hostilité. Ces liens négatifs soulèvent des questions à la fois techniques, car l'arsenal mathématique classique utilisé pour les réseaux sociaux n'est plus suffisant, et sémantiques, car lorsque apparaissent deux forces contraires (amitié/hostilité, confiance/méfiance), l'interprétation des relations devient plus complexe.

Sommaire

Introduction

L'utilisation d'internet et des réseaux sociaux s'est très largement développée ces dernières années, notamment avec la multiplication et la démocratisation des appareils connectés (smartphones, tablettes, etc.) grâce auxquels les utilisateurs peuvent partager n'importe quel moment de leur vie. Avec un nombre d'utilisateurs qui ne cesse d'augmenter, l'étendue des usages s'accroît rapidement et les données partagées se diversifient : de la vidéo ou des photos via des services comme Youtube ou Instagram, des informations professionnelles (compétences, expériences) via des réseaux comme LinkedIn ou Viadeo, et des données privées à partager avec ses proches via Facebook, Google+ ou Twitter. Ces différents usages sont devenus très populaires et permettent de récolter des quantités phénoménales de données personnelles, ce qui représente à la fois une opportunité et un défi pour l'ensemble de cette industrie.

L'étude de ces réseaux a donné naissance à une littérature extrêmement vaste [Aggarwal, 2011], appuyée par une théorie mathématique adaptée à la structure des réseaux sociaux : la théorie des graphes. Cette théorie étudie en particulier tous les systèmes dans lesquels des nœuds peuvent être reliés entre eux par des relations, comme les réseaux informatiques (internet, réseaux locaux), les réseaux de télécommunications ou encore les réseaux routiers. La théorie des graphes permet alors d'étudier des flux, de repérer des nœuds influents, ou encore d'observer des phénomènes invariants comme l'*effet du petit monde* [Milgram, 1967] stipulant que chaque utilisateur d'un graphe est relié à n'importe quel autre utilisateur par une distance moyenne faible (par exemple 6.6 liens en moyenne entre n'importe quel couple d'utilisateurs du réseau social MSN [Leskovec and Horvitz, 2008]). À la différence de la majorité des tâches d'apprentissage où les données observées sont considérées indépendantes, l'étude de tels réseaux constitue un changement de paradigme puisque les modèles doivent prendre en compte l'influence que les nœuds exercent entre eux (apprentissage transductif).

Les problématiques étudiées en informatique dans le cadre des réseaux sociaux sont assez nombreuses et se concentrent généralement sur la proximité entre les utilisateurs. Il existe trois principales tâches très largement étudiées : la détection de communautés, la classification de nœuds et la prédiction de liens. La détection de communautés est l'un des tout premiers sujets [Zachary, 1977] qui a pour but de partitionner l'ensemble des utilisateurs en groupes fortement connectés (communautés géographiques, communautés d'opinions, etc.), ce qui permet d'analyser différentes structures sociales. La classification de nœuds permet quant à elle de classifier les utilisateurs, par exemple pour améliorer les données de profils à disposition (genre, âge, etc.) sur des réseaux comme FaceBook ou Google+, ou encore en suggérant des groupes d'intérêt ou des individus spécifiques pour des réseaux plus professionnels (LinkedIn, Viadeo) ou des réseaux plus orientés médias (comme Twitter). Enfin, la tâche de prédiction de liens consiste à prédire des liens entre utilisateurs : qu'il s'agisse de prévoir de futures interactions, de recommander de nouvelles connexions ou simplement d'identifier des liens sociaux déjà existants mais non établis, par exemple dans l'étude de réseaux criminels.

Ces différentes tâches doivent cependant prendre compte la constante évolution de ces réseaux puisque ceux-ci évoluent au gré des interactions utilisateurs qui sont généralement rapides et nombreuses [Aggarwal and Philip, 2005]. Twitter totalise par exemple des centaines de milliers de nouvelles interactions (tweets) chaque minute. Ainsi, même si les méthodes

hors ligne demeurent tout à fait envisageables pour des tâches d'analyse de la structure du réseau (comme la détection de communautés, voire la classification), la prédiction de liens ou la recommandation rencontrent plus de difficultés lorsque les liens peuvent apparaître et disparaître à tout moment. De plus, les outils mis en place pour les modèles doivent également considérer la sémantique des liens qui relient les utilisateurs, puisque ces liens peuvent ne pas signifier la même chose selon le contexte et le réseau. Il existe des liens réciproques (entre amis, collègues), mais aussi des liens non réciproques dirigés d'une personne vers une autre (les *followers*), voire des liens valués lorsque des utilisateurs peuvent juger le contenu ou le comportement d'un autre utilisateur. Ces différentes interactions apportent alors leur lot d'informations, mais aussi de nombreuses contraintes en soulevant quelques difficultés : les liens dirigés diminuent en quelque sorte la connectivité du graphe, alors que les liens valués mènent certaines fois à des relations antagonistes (méfiance, hostilité) qui sont plus difficiles à interpréter et à traiter.

Dans ce chapitre, nous allons étudier différentes problématiques qui se posent dans les réseaux sociaux. Dans un premier temps, nous allons nous intéresser à la définition de ces réseaux et aux différents graphes de terrain rencontrés, qu'ils soient simples, dirigés ou pondérés. À partir de là, nous présenterons trois des tâches les plus traitées dans la littérature afin de mieux appréhender les problématiques imposées par l'étude de ces relations sociales. Ensuite, nous discuterons du cas classique où les liens entre les utilisateurs sont simples (sans pondération) et introduirons enfin le cas plus complexe - celui qui nous intéresse plus particulièrement dans ce manuscrit - des relations signées.

2.1 Définitions, notations et exemples de graphes de terrain

La branche des mathématiques qui se prête au mieux à l'étude des réseaux sociaux est la *théorie des graphes*, une théorie qui permet de représenter et d'étudier des ensembles d'objets reliés entre eux. Cette théorie s'appuie sur un arsenal mathématique et algorithmique puissant pour la résolution de nombreux problèmes comme la recherche du plus court chemin entre deux points géographiques, ou encore la gestion de flux de données dans des réseaux de télécommunications.

Formellement, un graphe G se définit comme un couple (V, E) où V est l'ensemble des nœuds (ou sommets) du graphe et $E \in V \times V$ l'ensemble des liens entre ces nœuds. Il existe deux formes de lien : non orienté ou orienté. Dans le premier cas, la relation est réciproque, le lien (ou arête) entre deux utilisateurs u, v est alors noté $\{u, v\}$. Dans le second cas, le lien (ou arc) possède un sens, il est noté (u, v) lorsqu'il est dirigé de u vers v, (v, u) dans la direction inverse. Dans un contexte social, les nœuds représentent les utilisateurs et les liens du graphe représentent les relations entre ces utilisateurs, pouvant indiquer une collaboration, une amitié ou une confiance par exemple. Dans certains contextes, les liens entre les utilisateurs peuvent être pondérés par une application $v : E \to \mathbb{R}$ - appelée *valuation* du graphe - qui donne une valeur à chaque relation entre les utilisateurs. Certains réseaux sociaux permettent une valuation négative, nous parlerons alors de graphes signés.

Nous distinguons généralement trois grandes familles de réseaux sociaux : les réseaux simples non orientés, les réseaux simples orientés et les réseaux pondérés orientés. Nous décrivons chacune de ces familles à travers des exemples de graphe de terrain.

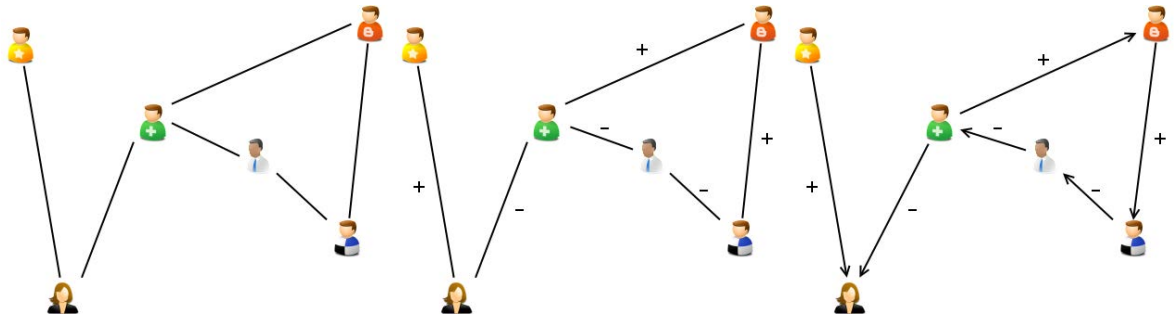


Figure 2.1 – *Les trois différents types de graphes de terrain rencontrés : les graphes simples non orientés (à gauche), les graphes simples orientés (au centre) et les graphes pondérés orientés (à droite).*

2.1.1 Graphes simples non orientés

La plupart des réseaux sociaux sur internet sont représentés par des graphes simples non orientés : les utilisateurs sont reliés avec d'autres utilisateurs via des liens réciproques qui n'ont pas de pondération, autrement dit les utilisateurs n'y ont pas associé de valeur. Souvent, cette absence de *valuation* ne tient pas de la nature ou de la sémantique de ces liens, mais de la difficulté de collecter de telles données. Puisque la relation va dans les deux sens, il est difficile d'obtenir ou d'évaluer la valeur d'un lien d'amitié, d'autant que les utilisateurs ont de fortes chances de ne pas y accorder la même valeur. Pour cette même raison, de tels réseaux sociaux n'autorisent pas les liens signés, c'est-à-dire des liens (réciproques) de discorde ou d'hostilité entre les utilisateurs.

L'exemple le plus connu de graphe simple non orienté est le réseau Facebook. Avec plus d'un milliard de comptes en ligne, il est le plus grand réseau social en ligne au monde. Le système s'appuie sur les relations réciproques entre les utilisateurs : un utilisateur n'aura accès au contenu partagé par un autre que si ce dernier l'accepte. Ce fonctionnement est celui de bien d'autres réseaux sociaux, notamment professionnels (LinkedIn ou Viadeo).

2.1.2 Graphes simples orientés

Pour certains réseaux, les liens sont dirigés d'un utilisateur vers un autre. Nous avons affaire à un graphe simple orienté. D'une manière générale, cela correspond à des *interactions sociales* lorsque par exemple un utilisateur exprime un intérêt pour un autre utilisateur - à partir de son profil, de son contenu, etc. Ces liens sont uniques et n'ont généralement aucune pondération ni signe puisqu'ils expriment la même chose : le personne souhaite suivre le contenu d'une autre personne.

L'exemple le plus connu est celui de Twitter dont le principe est de permettre aux utilisateurs de publier de courts messages, appelés tweets, qui seront lus par des suiveurs (*followers*). C'est le principe du *microblogage*. Les liens ne sont donc pas réciproques. D'autres réseaux au principe similaire autorisent des liens de différentes sémantiques, comme le réseau Google+. Sur ce réseau, la nature du lien orienté peut être explicitée par l'utilisateur parmi *ami, famille, connaissance* et *suivi* (ce dernier ayant alors la même sémantique qu'un lien Twitter). En revanche, ces liens ne peuvent pas avoir de sémantique antagoniste (*hostilité* ou *méfiance*).

2.1.3 Graphes pondérés orientés

Dans certains réseaux, les liens peuvent être orientés et dirigés. Il s'agit généralement de relations implicites où les utilisateurs évaluent une action ou un contenu d'un autre utilisateur. C'est donc un lien indirect d'un utilisateur vers un auteur. Ces dernières années, de nombreux sites ont sollicité les utilisateurs afin d'avoir leur retour sur la pertinence des avis laissés par d'autres. Le cas le plus courant est celui des sites de ventes en ligne où les utilisateurs sont autorisés à laisser un avis sur les produits qu'ils ont achetés : tout utilisateur peut alors indiquer si le commentaire laissé est utile ou non. C'est par exemple le cas sur Amazon avec la question *"Ce commentaire vous a-t-il été utile ?"*.

Figure 2.2 – *Amazon permet de laisser des avis sur les produits. Les autres utilisateurs peuvent ensuite évaluer la pertinence ou l'utilité de ces avis via la question "Ce commentaire vous a-t-il été utile ?"*

L'un des réseaux les plus utilisés pour l'étude des graphes pondérés orientés est celui d'Epinions, une plate-forme qui permet aux utilisateurs de partager leurs avis à propos d'items. La particularité d'Epinions est d'autoriser les utilisateurs, en plus d'évaluer la pertinence d'un jugement particulier, d'indiquer s'ils font confiance ou non à tel ou tel autre utilisateur. Autrement dit, un utilisateur donné peut émettre un avis négatif (donc orienté) vers un autre utilisateur pour indiquer qu'il n'a pas confiance ou qu'il n'est pas d'accord, de manière générale, avec les avis laissés par un utilisateur.

2.2 Les grandes tâches de l'analyse des réseaux sociaux

La structure de graphe peut être exploitée pour évaluer la similarité entre les nœuds. Les applications sont nombreuses une fois qu'il est possible de mesurer directement ou indirectement cette similarité : regroupement en communautés, recommandation de produits ou classification en catégories démographiques par exemple.

Nous distinguons généralement deux caractéristiques importantes sur l'information apportée par ces liens [Aggarwal, 2011, Bhagat et al., 2011] :

— homophilie : les liens sociaux entre les individus indiquent souvent des ressemblances du point de vue de certaines caractéristiques comme l'âge, la profession, les hobbies etc.

— régularité : les individus reliés entre eux tendent à partager certains centres d'intérêt ou à être connectés aux mêmes groupes par exemple.

Ces caractéristiques se retrouvent dans la structure du graphe et en particulier pour caractériser la relation entre deux utilisateurs : leur distance en terme de nombre de liens les séparant, ou encore le nombre de voisins qu'ils partagent. Par exemple, deux utilisateurs partageant un voisin en commun (distance = 2) ont une probabilité plus forte de partager des caractéristiques de profils ou des centres d'intérêt que deux utilisateurs pris au hasard dans ce graphe (distance moyenne autour de 6 d'après l'effet du petit monde). De la même manière, deux utilisateurs qui partagent plusieurs voisins communs, c'est-à-dire qui sont tout deux reliés aux mêmes individus, seront d'autant plus proches que ce nombre de voisins est grand.

Nous présentons dans la suite trois des tâches les plus étudiées dans la littérature [Aggarwal, 2011] : la détection de communautés, la classification de nœuds et la prédiction de liens.

2.2.1 Détection de communautés

La détection de communautés est un bon exemple d'application exploitant la structure de graphe. Elle a pour objectif d'identifier des groupes de nœuds dont la particularité est d'être davantage reliés entre eux qu'avec le reste du graphe : ce sont les communautés. L'existence de tels groupes d'utilisateurs est un sujet d'intérêt devenu extrêmement populaire ces dernières années avec le développement des réseaux sociaux. Cette structure particulière en composantes fortement connexes permet d'analyser ou de détecter des groupes d'utilisateurs partageant de fortes similitudes dans leurs goûts, leurs opinions ou leur comportement. Le partitionnement des individus en groupes et en communautés est un sujet étudié depuis de nombreuses années [Wasserman and Faust, 1994] et ce sont les travaux de [Girvan and Newman, 2002] qui ont défini plus précisément le problème de détection de communautés pour les graphes de terrain.

L'un des cas d'étude les plus repris dans la littérature est celui du Karaté Club proposé par l'anthropologue Zachary dans [Zachary, 1977]. Ce travail étudie les liens entre membres au sein d'un club de Karaté entre 1970 et 1972, en particulier l'étude porte sur l'existence de deux communautés de membres. Le groupe est composé de 34 personnes du club ayant eu des liens en dehors des seuls cours et réunions formelles. Plus récemment, dans le même domaine, [Adamic and Glance, 2005] ont proposé une étude portant sur les blogs politiques durant les élections de 2004 aux États-Unis. Cette étude a permis de montrer que la blogosphère se découpait en deux sous-groupes correspondant exactement aux deux groupes politiques, le groupe des libéraux et le groupe des conservateurs (figure 2.3), puis que les blogs en question n'étaient liés entre eux qu'au sein même de leur groupe et marginalement vers des blogs de l'opposition.

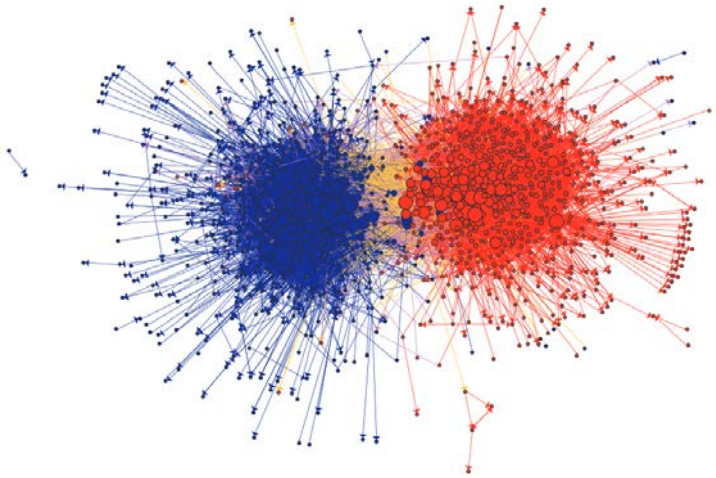


Figure *2.3 – Représentation de la structure du réseau de blogs politiques pendant les élections étasuniennes de 2004. Illustration de [Adamic and Glance, 2005].*

La tâche

La détection de communautés est basée sur une caractéristique particulière des graphes de terrain : un coefficient de clustering élevé (ou coefficient de regroupement) mesurant si le voisinage d'un nœud est plus ou moins connecté, c'est-à-dire à quel point les voisins d'un nœud sont reliés entre eux. En d'autres termes, il existe dans les graphes sociaux des composantes connexes locales pour lesquelles le nombre d'arêtes est plus important à l'intérieur même de la communauté qu'avec le reste du graphe. Formellement, la détection de communautés consiste à partitionner un graphe en composantes connexes denses avec un nombre limité de liens entre ces composantes. Il s'agit donc de subdiviser le graphe en sous-ensembles de nœuds de manière à ce qu'il y ait (1) beaucoup de liens intra-communautés et (2) peu de liens inter-communautés. Ces deux éléments sont caractéristiques des communautés et seront exploités pour évaluer la qualité de celles-ci.

Il existe de nombreuses approches dans la littérature qui diffèrent notamment dans leur manière de percevoir les communautés et d'évaluer les performances. En effet, comme n'importe quelle méthode non-supervisée, la détection de communautés s'appuie sur une fonction de qualité Q dont la définition correspond à une vision particulière des communautés et détermine les critères caractérisant ces communautés. Quelques fonctions de qualité sont présentées dans la section 2.3.1 afin d'illustrer les différentes approches de la littérature.

2.2.2 Classification de nœuds

Dans des graphes de terrain tels que les réseaux sociaux, les nœuds (i.e. les individus) peuvent être caractérisés par une multitude d'étiquettes telles que l'âge, le courant politique, le niveau d'études, etc. Ces étiquettes de profil peuvent à la fois permettre la recommandation de nouveaux amis, mais également d'items divers ou servir au ciblage publicitaire. Malheu-

reusement, la situation est souvent la même que dans les systèmes de recommandation où une partie des informations de profil ne sont pas (ou mal) renseignées : soit les individus font le choix de ne pas les indiquer, lorsque que par exemple elles sont considérées trop intrusives, soit les utilisateurs choisissent de donner de mauvaises informations ou de ne pas mettre à jour ces données.

La tâche de classification de nœuds s'inscrit dans ce contexte où il s'agit d'inférer certaines informations manquantes à propos des utilisateurs ; les étiquettes de nœuds sont alors obtenues à partir des étiquettes connues des voisins. Cette tâche de classification se distingue alors des approches classiques dans la mesure où la structure du graphe est exploitée grâce aux liens existants entre les nœuds ; les données ne sont alors plus considérées indépendantes. En particulier, l'idée est d'apprendre de nouvelles étiquettes à partir d'individus ayant renseigné ces informations, puis de compléter les profils d'autres utilisateurs, voire remplacer certains labels. Le cadre général est alors semi-supervisé [Chapelle et al., 2006] : les données sont presque toutes non étiquetées et il s'agit alors d'exploiter les quelques étiquettes disponibles.

La tâche

Du fait que la tâche soit semi-supervisée, on peut subdiviser l'ensemble des nœuds V en deux sous-ensembles : V_l les nœuds étiquetés (*labeled*) et V_u les nœuds non-étiquetés (*unlabeled*). La matrice Y contient les étiquettes des nœuds, telle que $Y_{ik} = 1$ si le nœud i est étiqueté k, 0 sinon. L'ensemble des nœuds étiquetés forme ainsi une sous-matrice de Y, noté Y_l et de taille $|V_l| \times c$ (où c est le nombre de classes possibles). Le reste de la matrice ne contient que des 0.

La matrice des étiquettes inférées est notée $\mathcal{F}$. La valeur $\mathcal{F}_{ij}$ correspond à la probabilité que le nœud i appartienne à la classe j. L'étiquette inférée d'un nœud i correspond alors à l'argument maximal de la ligne $\mathcal{F}_{i\bullet}$, c'est-à-dire la colonne j pour laquelle $\mathcal{F}_{ij}$ est maximal.

À partir de là, la classification peut se dérouler principalement de deux manières :

— Soit nous apprenons une fonction classifieur f sur l'ensemble des données afin d'inférer *par généralisation* la classe de n'importe quel nouveau nœud du graphe (classification inductive)

— Soit nous essayons d'inférer les seules étiquettes $\mathcal{F}_u$ de l'ensemble V_u des nœuds non-étiquetés sans passer par l'apprentissage d'une fonction classifieur (classification transductive)

La capacité de *généralisation* de la classification inductive est un avantage certain lorsque de nouvelles données sont incorporées. En effet, chaque nouvelle donnée nécessite une nouvelle phase d'apprentissage transductif, ce qui n'est pas le cas de la fonction classifieur qui elle s'applique quelles que soient les données [Bengio et al., 2006]. Toutefois, la classification transductive est plus facile et souvent plus performante [Chapelle et al., 2006].

Nous étudions dans la section suivante les deux principales approches transductives pour la classification de nœuds [Zhu, 2005]. La première approche consiste à *propager* les étiquettes dans le graphe à travers les liens entre les nœuds. C'est une méthode itérative [Zhu and Ghahramani, 2002] consistant à diffuser la valeur des étiquettes des nœuds de V_l à l'ensemble

du graphe jusqu'à convergence, et ce, afin d'obtenir les étiquettes des nœuds de V_u. Le seconde approche s'appuie sur le principe de régularisation afin de définir une fonction objectif qui respecte la structure du graphe en forçant les nœuds voisins à avoir la même étiquette [Belkin et al., 2004, Zhou et al., 2004]. Nous présentons chacune des deux approches dans la section 2.3.

2.2.3 Prédiction de liens

Dans les deux tâches précédentes, nous nous intéressions aux nœuds et notamment à leur appartenance à des classes ou des communautés. Un autre axe de recherche consiste à analyser et prédire de nouveaux liens entre les nœuds. Il peut s'agir de conseiller de nouvelles relations dans un réseau professionnel, de nouvelles personnes à suivre ou encore du contenu à recommander. La prédiction de liens est une tâche qui ne s'applique pas seulement aux réseaux sociaux, mais à une multitude de graphes de terrain. En particulier, elle a été étudiée sous divers aspects et a touché plusieurs domaines d'applications. Quelques exemples sont donnés dans [Al Hasan and Zaki, 2011], comme la prédiction d'interactions entre protéines [Airoldi et al., 2006], la prédiction d'hyper-liens entre sites web [Zhu and Ghahramani, 2002] ou encore la prédiction de données relationnelles [Taskar et al., 2003, Popescul and Ungar, 2003]. Pour les réseaux sociaux, la littérature s'y est largement intéressée ces dernières années [Aggarwal, 2011].

Le principe général de la prédiction de liens consiste à mesurer un score de similarité entre deux individus, selon plusieurs métriques pouvant dépendre du contenu de profil, du voisinage ou encore de la plus courte distance entre les deux utilisateurs en terme de nombre de liens les séparant. Si par exemple deux personnes ne sont pas connectées alors même que ces deux individus partagent une grande partie de leur réseau professionnel, alors un lien peut être raisonnablement suggéré.

La tâche

Plus formellement, cette tâche consiste à prédire des liens dans un graphe de terrain donné $G = (V, E_{train} \cup E_{test})$ où E_{train} est l'ensemble des liens étiquetés utilisés pour l'apprentissage et E_{test} ceux utilisés pour l'évaluation. Nous définissons cette tâche comme de la classification supervisée. Pour certains couples d'utilisateurs, nous avons une étiquette correspondant à la présence ou non d'un lien entre les deux utilisateurs donnés sur l'ensemble d'apprentissage. Soit :

$$y_{uv} = \begin{cases} +1 & \text{si } (u,v) \in E_{train} \\ -1 & \text{si } (u,v) \notin E_{train} \end{cases}$$

Il reste alors à inférer les étiquettes des couples d'utilisateurs étiquetés sur l'ensemble d'évaluation. Pour ce faire, un score pour chaque paire de nœuds est calculé. Celui-ci peut dépendre de diverses caractéristiques décrivant les couples d'utilisateurs qui sont calculées sur (V, E_{train}). Ces scores sont ensuite exploités afin d'appliquer une simple méthode de classification supervisée comme les machines à vecteurs de support (*Support Vector Machines*)

ou une méthode de k plus proches voisins. Nous présentons quelques caractéristiques pour la prédiction de liens dans les sections suivantes, dans le cas signé et le cas non signé.

2.3 Le cas non signé

Dans cette partie, nous décrivons les différentes méthodes utilisées pour résoudre les tâches présentées dans la section précédente dans le cas où les liens sont non signés. C'est le cas le plus courant où les utilisateurs sont reliés entre eux par de *simple* liens, et celui qui bénéficie de la littérature la plus abondante [Aggarwal, 2011].

Dans la section 2.3.1, nous présentons les fonctions de coût pour la détection de communautés et détaillons quelques approches algorithmiques. Dans la section 2.3.2 nous présentons des méthodes semi-supervisées de propagation d'étiquettes (classification de nœuds). Enfin, nous décrivons dans la section 2.3.3 les méthodes supervisées à base de caractéristiques de couples (pour la prédiction de liens).

2.3.1 Critères de qualité pour le détection de communautés

Comme nous l'avons précisé dans la section précédente, la détection de communautés s'appuie sur des critères choisis à l'avance et répondant à une vision particulière de ce qu'est une communauté. Pour cela, on utilise une fonction de coût qui permet d'évaluer la qualité du partitionnement selon ces critères. Nous présentons dans cette section deux fonctions de qualité : les fonctions *modularité* et *Ncut*.

Fonctions de qualité

Une fonction de qualité nous permet d'associer à chaque partition $\mathcal{P}$ du graphe une valeur $Q(\mathcal{P})$ de manière à quantifier à quel point la partition donnée vérifie ou non les critères de communauté. La détection de communautés consiste donc à trouver la partition qui maximise cette fonction qualité. Parmi les mesures les plus courantes, nous pouvons en citer deux d'entre elles :

— la fonction *modularité* permettant de maximiser le nombre de liens intra-communauté

— la fonction *Ncut* (ou coupe normalisée) est utilisée pour la minimisation du nombre de liens inter-communauté

Modularité

La modularité [Newman and Girvan, 2004] permet de mesurer à quel point les communautés sont denses, ce qui revient à évaluer pour chaque communauté la différence entre le nombre de liens observés et le nombre de liens qui seraient obtenus pour un graphe avec la même distribution de degrés. Un graphe aléatoire ne possède pas de structure communautaire,

c'est pourquoi il constitue en quelque sorte un référentiel à partir duquel on peut comparer les densités locales de chaque composante connexe.

Soit $G = (V, E)$ un graphe et $\mathcal{P}$ une partition donnée. En notant E_p l'ensemble des liens entre les utilisateurs de $p \in \mathcal{P}$, la modularité est définie de la manière suivante :

$$Q(\mathcal{P}) = \sum_{p \in \mathcal{P}} \left(\frac{|E_p|}{|E|} - \left(\frac{\sum_{i \in p} d_i}{2|E|} \right)^2 \right) \tag{2.1}$$

avec d_i le degré du nœud i.

Coupe normalisée ($Ncut$)

Cette mesure [Newman, 2005] est utilisée de manière à minimiser le nombre de liens entre les communautés du graphe. La notion de coupe se réfère en particulier aux liens entre communautés. Il s'agit de favoriser les communautés les moins connectées avec le reste du graphe, plutôt que les communautés les plus denses. Formellement, cette coupe normalisée se définit, pour un sous-graphe $S \subset V$ donné, de la manière suivante :

$$Ncut(S) = \frac{\sum_{i \in S, j \in \bar{S}} A(i,j)}{\sum_{i \in S} d_i} + \frac{\sum_{i \in S, j \in \bar{S}} A(i,j)}{\sum_{i \in \bar{S}} d_i} \tag{2.2}$$

où A est la matrice d'adjacence définie telle que A_{ij} contient la valeur du lien entre i et j (pour un graphe non pondéré, nous avons un 1 si les utilisateurs i et j sont reliés, 0 sinon).

Les approches algorithmiques

L'optimisation exacte d'une fonction de qualité comme $Ncut$ ou la *modularité* est un problème NP-difficile [Brandes et al., 2007], c'est pourquoi diverses heuristiques ont été proposées dans la littérature afin d'obtenir une solution approchée dans un temps raisonnable. Parmi l'ensemble des méthodes connues (voir [Fortunato, 2010] pour une liste exhaustive), la plupart reposent sur la structure hiérarchique des communautés, c'est-à-dire le fait que les communautés se décomposent elles-mêmes en sous-communautés. Ainsi, beaucoup de méthodes construisent une hiérarchie de communautés prenant la forme d'un *dendogramme* ; la meilleure partition étant alors celle maximisant la mesure choisie.

Ces méthodes se décomposent principalement en deux grandes familles d'algorithmes :

— les méthodes agglomératives qui partent de communautés de taille 1 (un nœud) et les fusionnent au fur et à mesure jusqu'à atteindre une taille prédéfinie ou la taille jugée optimale (une fusion diminuerait alors la modularité, par exemple).

— les méthodes divisives partent à l'inverse du graphe entier et retirent itérativement des arêtes du graphe de manière à obtenir des composantes connexes assimilables à des communautés.

Chaque famille est constituée d'une multitude d'algorithmes qui se définissent alors par leur manière de retirer les arêtes ou de fusionner les communautés. On pourra citer l'algorithme

divisif de Grivan-Newman [Newman and Girvan, 2004] ou encore l'algorithme agglomératif glouton introduit initialement dans [Newman, 2004] et amélioré dans [Clauset et al., 2004]. Plus récemment, la méthode (agglomérative) de Louvain [Blondel et al., 2008] est devenue très populaire en se montrant à la fois plus rapide et plus efficace que la majorité des autres méthodes de la littérature.

2.3.2 Méthodes semi-supervisées : proximité et propagation des étiquettes

Les méthodes de propagation ont pour principe de transmettre ou diffuser l'information à travers les liens du graphe (méthodes transductives). Elles s'appuient en particulier sur les deux principes que sont l'homophilie et la régularité. L'idée est simplement que les utilisateurs liés dans le graphe tendent à partager les mêmes étiquettes. Cette famille de techniques se révèle alors particulièrement pertinente lorsque les utilisateurs ne complètent pas leurs données de profil ou si certaines informations sont incomplètes ou difficiles à demander.

Ces méthodes reposent sur l'utilisation d'une matrice de transition dérivée de la matrice d'adjacence, et ce, afin de construire un processus stochastique Markovien. La popularité de cette approche provient en partie des travaux de Larry Page [Page et al., 1999] dont l'algorithme PageRank a donné naissance au moteur de recherche Google.

Dans cette section, nous décrivons une méthode itérative de propagation des étiquettes à laquelle s'ajoute une approche par régularisation. Le principe de cette approche repose sur les marches aléatoires : s'appuyer sur certaines propriétés de la matrice de transition pour associer au graphe un vecteur invariant de probabilités, traduisant le score de chaque sommet. Nous commençons donc par étudier les marches aléatoires au travers de l'étude de l'algorithme PageRank.

Méthodes spectrales

L'une des principales contributions de [Page et al., 1999] est d'avoir proposé une méthode de marche aléatoire adaptée aux graphes de terrain. En particulier, les auteurs ont proposé un modèle exploitant directement la matrice d'adjacence du graphe du web afin d'analyser les liens entre les sites et associer aux différentes pages un score proportionnel au pourcentage de fois qu'un *surfeur* passerait sur ce site en parcourant le graphe de manière aléatoire (vecteur de probabilité stationnaire).

La première étape consiste à transformer la matrice d'adjacence A en une matrice stochastique P, c'est-à-dire une matrice de transition (ou matrice de Markov). Pour cela, il suffit de calculer la matrice P correspondant à la matrice A normalisée par ligne, de sorte qu'un surfeur se situant sur le site i a une probabilité $P_{ij} = A_{ij} / \sum_j A_{ij}$ de suivre le lien menant du site i au site j.

Cette matrice de transition P est ensuite modifiée afin de la rendre ergodique. En effet, si le graphe associé n'est pas connexe par exemple, le surfeur peut rester indéfiniment à l'intérieur d'une composante connexe ou simplement bloqué sur un site. On suppose pour cela que le surfeur a une probabilité α de suivre un lien et une probabilité $1 - \alpha$ de se *télétransporter*

sur une page quelconque. Nous obtenons une nouvelle matrice de transition, en notant n le nombre de pages :

$$\bar{P} = \alpha P + (1-\alpha)\frac{1}{n}\mathbb{1}\cdot\mathbb{1}^T \tag{2.3}$$

Finalement, si on suppose que la probabilité que l'utilisateur se trouve sur une des pages est donnée par un vecteur π, alors l'hypothèse de stationnarité se traduit par l'équation $\pi^T\bar{P} = \pi^T$. Ce vecteur π est alors obtenu rapidement de manière itérative : à partir d'un vecteur stochastique quelconque π_0, il suffit d'itérer $\pi^{n+1} = \bar{P}\pi_n$ puisque la convergence de la suite $(\bar{P}^n)$ vers une matrice stochastique strictement positive est garantie par l'ergodicité.

Propagation d'étiquettes

L'approche par marche aléatoire se transpose au cas de la classification. De la même manière qu'un surfeur se déplace de site en site, il existe plusieurs approches itératives basées sur l'idée que les étiquettes de nœuds se propagent à travers les liens [Bengio et al., 2006] ; les nœuds connectés s'influencent alors entre eux et tendent à avoir des étiquettes similaires, alors que les nœuds non connexes ont plus de chance d'avoir des étiquettes différentes.

Nous décrivons ici une des approches de propagation de label proposée dans [Zhou et al., 2004]. Les auteurs proposent un algorithme dans lequel la matrice $\mathcal{F}$ des étiquettes inférées (voir section 2.2.2) est obtenue par N marches aléatoires (une par label possible) qui se déroulent de manière parallèle. En particulier, l'étiquette inférée d'un nœud donné est mise à jour à chaque itération en prenant en compte à la fois l'étiquette initiale du nœud ainsi que l'information reçue du voisinage.

Formellement, l'algorithme itératif est défini de la façon suivante :

$$\mathcal{F}_{t+1} = \alpha S\mathcal{F}_t + (1-\alpha)Y \tag{2.4}$$

où S est une matrice de transition que l'on définit dans la suite. Le principe est alors qu'à chaque itération, le nœud i reçoit une information de ses voisins avec un poids α tout en gardant l'information initiale avec un poids $1-\alpha$.

Dans [Zhou et al., 2004], la matrice de transition S est construite sur la base que l'information reçue d'un voisin est proportionnelle à la distance qui les sépare, soit $W_{ij} = exp(-\|x_i - x_j\|^2/2\sigma^2)$ où x_i correspond à la représentation d'un utilisateur.

W est alors transformée en une matrice ergodique, comme pour l'algorithme Pagerank, en utilisant la formule du laplacien normalisée. En particulier, on définit la matrice $S = I - \mathcal{L}$ et $\alpha \in]0,1[$, avec :

$$\mathcal{L}(u,v) = \begin{cases} 1 & \text{si u=v et } d_v \neq 0 \\ -\frac{w_{uv}}{\sqrt{d_u d_v}} & \text{si u et v sont adjacents,} \\ 0 & \text{sinon.} \end{cases} \tag{2.5}$$

avec d_u le degré du nœud u.

Approche par régularisation

Il est également possible d'aborder la classification par une approche par régularisation [Zhou et al., 2004]. Celle-ci est proche de la précédente car elle repose sur l'idée que les nœuds reliés entre eux ont en général des étiquettes similaires. Cette formulation part de la définition d'un risque empirique où l'on mesure l'erreur quadratique entre les étiquettes connues Y_i et les étiquettes inférées $\mathcal{F}_i$. Comme nous l'évoquions dans la section 1.2.1, la solution qui minimise un risque empirique n'est pas unique, c'est pour cette raison que les liens entre les utilisateurs sont utilisés pour la régularisation. L'idée est de restreindre l'espace des hypothèses en imposant par exemple que les nœuds voisins aient des étiquettes similaires. La fonction de coût Q peut alors être définie de la façon suivante :

$$Q(F) = \sum_{i=1}^{m} (\mathcal{F}_i - Y_i)^2 + \mu \left(\frac{1}{2} \sum_{u,v} W_{uv} \left(\frac{\mathcal{F}_u}{\sqrt{d_u}} - \frac{\mathcal{F}_v}{\sqrt{d_v}} \right)^2 \right) \tag{2.6}$$

avec μ (positif) le coefficient de régularisation, m le nombre d'utilisateurs et W une matrice de similarité qui peut être définie de la même manière que pour l'approche précédente.

Le premier terme correspond au risque empirique (formulation des moindres carrés) et le second terme est une régularisation utilisant la même matrice de transition S que dans la section précédente (c'est-à-dire avec le laplacien normalisé).

2.3.3 Méthodes supervisées : modèles à base de caractéristiques

Les méthodes supervisées à base de caractéristiques s'appuient sur la riche littérature d'apprentissage automatique en appliquant des méthodes supervisées sur des données représentées par des caractéristiques, calculées en fonction de la structure du graphe. L'idée est de représenter chaque paire d'utilisateurs via des caractéristiques dépendant de leur voisinage ou encore de leur distance dans le graphe (le nombre de liens les séparant), pour ensuite appliquer des méthodes plus génériques telles que les machines à vecteurs de support (*Support Vector Machines*) ou encore des techniques de k plus proches voisins. L'hypothèse est que les nœuds ou les liens qui ont des caractéristiques similaires tendent à appartenir aux mêmes classes.

Cette famille de méthodes est utilisée en particulier pour la tâche de prédiction de liens pour laquelle un ensemble de paires de nœuds est utilisé pour la supervision d'un modèle de classification exploitant les caractéristiques de ces paires afin d'inférer l'étiquette des liens (le lien existe ou non). L'avantage de cette approche est de pouvoir s'appliquer plus facilement à de nouvelles données (approche *inductive*) alors que la propagation d'étiquettes ne permet pas d'inférer les classes de données non parcourues lors de l'apprentissage (approche *transductive*).

La définition des caractéristiques joue naturellement un rôle essentiel. La proximité entre les individus est évaluée via des mesures de similarité basées sur la structure de graphe [Liben-Nowell and Kleinberg, 2007, Kashima and Abe, 2006, Al Hasan et al., 2006], en particulier à partir des deux éléments suivants :

— le voisinage commun : les utilisateurs partageant davantage de voisins tendent à appartenir à la même classe

— la taille des chemins séparant les utilisateurs : ils ont d'autant plus de chance d'appartenir à la même classe que leur distance dans le graphe est faible

Caractéristiques à base de voisinage

Il s'agit là de la première intuition consistant à associer la proximité des nœuds à la taille de leur voisinage. L'idée étant que deux utilisateurs sont plus susceptibles d'être reliés s'ils partagent déjà un grand nombre de voisins. Pour traduire cela, on note $\Gamma(x)$ l'ensemble des individus voisins (i.e les nœuds adjacents) de x parmi les liens observés, c'est-à-dire ceux de l'ensemble d'apprentissage. L'ensemble du voisinage commun à deux utilisateurs x et y est alors égal à $|\Gamma(x) \cap \Gamma(y)|$. Cette valeur représente déjà un bon indicateur pour la prédiction de liens [Newman, 2001].

Une autre façon d'utiliser le voisinage est de calculer le coefficient de Jaccard qui permet généralement de mesurer la similarité entre deux échantillons statistiques. Du point de vue du graphe, ce coefficient mesure simplement la probabilité de tirer un voisin commun aux deux utilisateurs parmi l'ensemble des voisins liés à l'un ou à l'autre des utilisateurs. Ainsi, nous avons :

$$score(x, y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x) \cup \Gamma(y)|} \tag{2.7}$$

Toutefois, cette mesure de Jaccard s'est révélée moins performante [Liben-Nowell and Kleinberg, 2007] en comparaison du nombre de voisins communs.

D'une manière similaire, on peut utiliser la mesure de Adamic et Adar [Adamic and Adar, 2003], originalement proposée de manière à mesurer la similarité entre deux pages web. Dans le cas d'étude des voisinages et de prédiction de liens, [Liben-Nowell and Kleinberg, 2007] ont adapté la mesure de la manière suivante :

$$score(x, y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{log|\Gamma(z)|} \tag{2.8}$$

Il s'agit dans ce cas de pondérer différemment les voisins communs en fonction de leur degré. Autrement dit, au lieu de compter simplement le nombre de voisins communs, on attribue une valeur d'autant plus forte que le voisin en question a peu de voisins.

Enfin, une dernière approche est *l'attachement préférentiel* [Mitzenmacher, 2001]. L'idée traduite par cette approche est que la probabilité d'avoir un nouveau lien impliquant x est directement proportionnelle à la taille de son voisinage $\Gamma(x)$. Ce que [Newman, 2001] traduit dans le cas de la prédiction de liens par :

$$score(x, y) = |\Gamma(x)| \cdot |\Gamma(y)| \tag{2.9}$$

Caractéristiques à base de chemins

Considérer les chemins entre les nœuds est une approche classique exploitant la structure du graphe. L'un des premiers points potentiellement intéressant est la distance séparant deux nœuds du graphe. Nous pouvons en effet supposer que deux utilisateurs séparés par un chemin de taille 2 seront plus susceptibles d'avoir des points communs que deux utilisateurs choisis au hasard. Cependant, compter le nombre de liens séparant les utilisateurs ne suffit pas puisque l'effet petit monde indique que tous les couples d'utilisateurs sont séparés par une distance faible. Cette caractéristique a donc peu d'importance [Liben-Nowell and Kleinberg, 2007]. C'est en partie pour résoudre ce problème que les mesures de similarité basées sur les chemins considèrent l'ensemble de *tous* les chemins entre deux sommets pour attribuer un score à un couple d'utilisateurs.

Une première mesure, nommée *mesure de Katz*, calcule la somme de tous les chemins, pondérés suivant leur taille. Soit plus formellement :

$$score(x,y) = \sum_{l=1}^{\infty} \beta^l \cdot |paths_{x,y}^{\langle l \rangle}| \tag{2.10}$$

où $|paths_{x,y}^{\langle l \rangle}|$ est le nombre de chemins de longueur l entre x et y. Le coefficient β permet de pondérer les chemins en fonction de leur longueur : les chemins plus longs ont des poids plus faibles.

Il est possible d'estimer la distance entre deux nœuds du graphe en utilisant des techniques basées sur des marches aléatoires [Liben-Nowell and Kleinberg, 2007]. Le nombre de pas espéré $H_{x,y}$ (*hitting time*) pour qu'une marche aléatoire partant du sommet x atteigne le sommet y est utilisé pour définir le score de l'arête $\langle x, y \rangle$. Puisque $H_{x,y}$ n'est en général pas symétrique, le score est défini de la façon suivante :

$$score(x,y) = -(H_{x,y} + H_{y,x}) \tag{2.11}$$

Ce calcul de score est cependant très dépendant de la *probabilité stationnaire* de l'un des deux sommets (section 2.3.2). En effet, l'arête $\langle x, y \rangle$ sera associée à un grand score si la *probabilité stationnaire* π_y est grande, quelque soit le sommet x de départ. C'est pour cela que la version normalisée est en général préférée :

$$score(x,y) = -(H_{x,y} \cdot \pi_y + H_{y,x} \cdot \pi_x) \tag{2.12}$$

Enfin, et pour éviter que la marche aléatoire ne parcoure une partie du graphe trop éloignée des deux sommets x et y qui nous intéressent, il est possible de *forcer* ce processus à retourner régulièrement au point de départ. C'est une méthode similaire à l'algorithme *Pagerank*, mais au lieu d'avoir une probabilité de "sauter" vers un sommet non nécessairement voisin, on a une certaine probabilité α de retourner au sommet de départ. Et ainsi, une probabilité $(1-\alpha)$ d'aller vers un voisin du sommet sur lequel on se trouve.

2.4 Le cas signé

L'étude des liens signés est une tâche qui motive de plus en plus de travaux, sans être pour autant encore très étudiée dans la littérature. Elle répond à une problématique sociologique évidente et étudiée depuis plusieurs années [Heider, 1946, Harary, 1953, Davis, 1977] puisque les liens antagonistes existent naturellement sous plusieurs formes (ami/ennemi, confiance/méfiance). Cette problématique appliquée à des données réelles en informatique voit les premiers travaux apparaître dans les années 2009-2010 [Kunegis and Lommatzsch, 2009, Leskovec et al., 2010a].

2.4.1 Données difficiles à collecter

Aujourd'hui, il s'avère difficile de collecter des données sociales signées entre les utilisateurs, c'est-à-dire des liens signés réciproques. Quelques plates-formes en ligne autorisent les liens négatifs entre individus, mais les utilisateurs ne semblent pas en faire une utilisation très massive (de l'ordre de 15 à 25% sur les plate-formes Slashdot, Epinions ou Wikipedia [Leskovec et al., 2010a]). Une hypothèse possible à cela est que ces liens sont considérés inconvenants et intrusifs par les utilisateurs. En effet, un lien négatif est synonyme de méfiance et d'hostilité et peut alors être source de conflit ou de tension ; déclarer un lien négatif vis-à-vis d'un second utilisateur est alors perçu comme un provocation, ce que les plates-formes en ligne cherchent déjà à éviter via la censure de certains contenus. De plus, les utilisateurs ne perçoivent pas nécessairement l'intérêt d'expliciter de tels liens alors qu'ils n'ont pas une incidence directe sur l'utilisation du site web.

Une autre source d'informations signées provient des interactions indirectes entre les utilisateurs, par exemple lorsqu'un utilisateur évalue le contenu généré par un autre utilisateur (lien orienté). C'est par exemple le cas de services en ligne comme Amazon où les utilisateurs peuvent évaluer la pertinence des avis laissés par d'autres utilisateurs sur certains produits. Ces liens sont plus fréquents parce que les utilisateurs jugent avant tout un contenu et non une personne. En revanche, ces liens demeurent limités puisqu'ils ne sont qu'en direction des auteurs de contenus, limitant ainsi le nombre d'interactions de cette forme. Il s'agit néanmoins de la majorité des liens signés disponibles dans les graphes de terrain.

2.4.2 Deux théories pour l'interprétation des liens signés

Traiter des interactions valuées soulève cependant de nombreux problèmes quant à l'utilisation des techniques et des modèles classiques de la littérature puisque l'arsenal mathématique utilisé dans le cas non signé n'est plus réellement adapté à ce contexte (les marches aléatoires notamment). De plus, l'interprétation sur la manière dont ces liens interagissent entre eux est plus complexe. Par exemple dans le cas non signé étudié jusque là, il est assez simple de mettre en place une règle de transitivité de la forme : si les utilisateurs u et v sont amis, alors tout ami de u sera potentiellement un ami de v, et vice-versa. En revanche dans le cas signé, si les utilisateurs u et v sont ennemis, que dire des ennemis de u vis-à-vis de v ?

Deux théories sur l'étude du signe dans les réseaux ont été proposées dans la littérature : la théorie de l'équilibre (graphe non orienté) et la théorie du statut (graphe orienté). La théorie de l'équilibre est issue d'études sociologiques sur les liens antagonistes datant du milieu du $20^{ème}$ siècle [Heider, 1946, Harary, 1953, Davis, 1977]. Ces études portent principalement sur des graphes non orientés représentant les relations naturelles possibles entre personnes. Plus tard, lorsque les premiers réseaux sociaux ont commencé à collecter des données signées, il a fallu redéfinir cette théorie dans un cadre plus adapté aux liens signés orientés (majoritaires), c'est la théorie du statut [Leskovec et al., 2010a]. Dans cette section, nous présentons les deux théories proposées et donnons quelques exemples d'observations pour chacune d'elles.

La première approche : la théorie de l'équilibre

Le cas du signe des liens dans un contexte social a été analysé bien avant l'étude des réseaux sociaux. En théorie des graphes, il y a par exemples des premiers travaux théoriques basés sur l'équilibre des signes dans les triades et les chemins d'un graphe. C'est en particulier à Heider [Heider, 1946] que nous devons la première théorie structurelle pour les graphes signés, ensuite formalisée pour la théorie des graphes par Harary [Harary, 1953, Harary, 1955] : la théorie de l'équilibre (en anglais *structural balance theory*). Cette théorie s'inscrit dans le cadre non orienté et émet l'hypothèse qu'un graphe signé est équilibré, c'est-à-dire que le produit du signe des liens de n'importe quel cycle de longueur 3 (aussi appelé triade) est positif.

Il ne peut y avoir qu'un nombre pair de liens négatifs dans une triade reliant trois individus : soit zéro lien négatif, soit deux liens négatifs. Concrètement, si nous revenons à l'exemple précédent, une telle théorie mène aux aphorismes suivants :

— l'ami d'un ami est un ami $(+ + +)$

— l'ennemi d'un ami est un ennemi $(+ - -)$

— l'ami d'un ennemi est un ennemi $(- + -)$

— l'ennemi d'un ennemi est un ami $(- - +)$

Le théorème de *Cartwright-Harary* [Harary, 1953] stipule qu'un graphe signé est équilibré si et seulement si l'ensemble de ses sommets peut être partitionné en deux sous-ensembles disjoints de telle sorte que tout lien positif relie deux sommets du même sous-ensemble et tout lien négatif relie deux sommets de différents sous-ensembles. Cependant, pour certains auteurs [Davis, 1977], un tel partitionnement en deux groupes est trop strict sachant que plusieurs études sociologiques ont évoqué des partitionnements en trois groupes, voire plus. Pour mieux comprendre cette situation, prenons deux ennemis u et v. La théorie de l'équilibre stipule alors que les ennemis de l'un seront les amis de l'autre (partitionnement en deux groupes). Dans la réalité, ce genre de situation n'est pas si évident. La contribution de [Davis, 1977] a ainsi consisté à nuancer la théorie de l'équilibre en stipulant que seules les triades possédant une seule arête négative sont proscrites, laissant ainsi la possibilité d'avoir des triades $(- - -)$ (*weak structural balance*).

Afin de vérifier cette théorie, *Leskovec et al.* ont proposé une étude exploratoire [Leskovec et al., 2010b] dans laquelle ils se sont intéressés à l'observation des triades dans des graphes

de terrain où les interactions sont dirigées - ce qui correspond à la majorité des réseaux sociaux signés actuels. Leur travail a consisté à vérifier si la propriété d'équilibre était bien respectée si l'on ignore le sens des liens entre les individus, et ce, en comparant la distribution statistique des différentes configurations de triades observées avec le nombre de triades qui seraient obtenues si l'ensemble des signes présents dans le graphe étaient permutés de manière aléatoire. Les conclusions de l'étude sont que les triades équilibrées, celles avec deux arêtes négatives sont sous-représentées et celles avec trois arêtes positives sont sur-représentées par rapport à l'aléatoire. En revanche, pour les configurations non équilibrées, le cas $(+ - +)$ est sous-représenté alors que la configuration $(- - -)$ ne l'est pas, ce qui penche en faveur de la notion d'équilibre *faible*.

La seconde approche : la théorie du statut

La théorie de l'équilibre suppose que les liens sont non dirigés ; or, de nombreux graphes de terrain le sont (par exemple Epinions ou Slashdot). La théorie du statut permet de traiter ce type de graphes. Elle a été introduite par l'étude exploratoire citée plus haut [Leskovec et al., 2010b]. Les auteurs sont partis du principe qu'un arc positif de u vers v signifie *"u pense que v a un meilleur statut que lui"* et qu'un arc négatif signifie l'opposé, *"u pense que v a un moins bon statut que lui"*. Cette approche traduit donc une intuition différente de celle de l'équilibre, à savoir une relation d'ordre entre les utilisateurs.

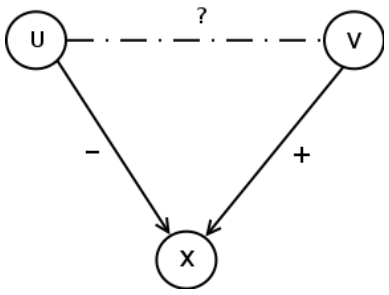


Figure 2.4 – *Premier cas : un arc négatif de u vers x et un arc positif de v vers x.*

L'avantage de cette théorie est de pouvoir, en principe, prédire à la fois le signe et l'orientation d'un lien entre deux utilisateurs (ce que la théorie de l'équilibre ne permet pas). Malheureusement, l'orientation diminue l'information disponible et il en résulte que de nombreuses situations ne pourront pas être résolues. Pour illustrer cela, prenons l'exemple de deux utilisateurs u et v dont nous cherchons à prédire la nature de leur lien (orientation et signe). Nous considérerons deux cas afin de bien comprendre le problème.

Dans le premier cas, un troisième utilisateur x a reçu un avis négatif de u et un avis positif de v (figure 2.4). Nous avons alors un arc négatif de u vers x et un arc positif de v vers x. Étant donnée la sémantique des liens (relation d'ordre), nous pouvons inverser l'arc de u vers x pour en faire un arc positif de x vers u. Ainsi, par transitivité (d'une relation d'ordre) nous obtenons un arc positif de v vers u.

Maintenant, considérons le deuxième cas similaire au premier à la différence que u a

exprimé un avis positif sur x plutôt que négatif (figure 2.5). Dans ce cas, la transitivité n'est pas possible et il nous est alors impossible de prédire le signe ou l'orientation de l'arc entre u et v. En fait, l'absence de réciprocité diminue le nombre d'inférences qu'il est possible de faire.

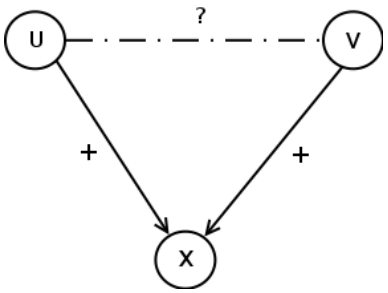


Figure *2.5* – *Second cas : un arc positif de u vers x et un arc positif de v vers x.*

Leskovec et al. ont étudié [Leskovec et al., 2010a] à travers plusieurs graphes de terrain le nombre d'observations pour chacune des 16 configurations de triades possibles. À partir de ces données, ils ont évalué la probabilité qu'un arc donné soit positif sachant la triade à laquelle il appartient. Par exemple, pour le premier cas de la figure 2.4, la probabilité que l'arc de v vers u soit positif est de 0.86 (ce qui va dans le sens de la théorie du statut). Dans notre second cas (figure 2.5) où la théorie du statut ne permet pas de prédire le signe ou l'orientation, la probabilité d'avoir un arc positif est de 0.94 (quelle que soit l'orientation).

Ensuite, les auteurs ont introduit les notions de *bases génératives et réceptives* qui correspondent à la proportion de liens positifs respectivement créés et reçus par l'utilisateur. Ces bases permettent de mesurer où se situe un utilisateur vis-à-vis des autres en comptant les différentes relations d'ordre dans lesquelles il est impliqué. Par exemple, si un utilisateur a une base réceptive à 1, alors il n'a reçu que des arcs positifs, ce qui revient à dire qu'il a un meilleur statut que tous les utilisateurs qui ont exprimé un avis sur lui. Inversement, s'il a une base réceptive à -1, alors tous les utilisateurs l'ayant jugé ont considéré avoir un meilleur statut que lui.

À partir de là, *Leskovec et al.* ont défini les surprises *génératives et réceptives* mesurant la différence entre le signe des arcs observés sur un graphe de terrain et les *bases génératives et réceptives* des utilisateurs impliqués. L'idée est que si le contexte dans lequel se situe l'arc n'a pas d'importance, alors le signe de l'arc devrait dépendre uniquement des bases de chacun des utilisateurs et la surprise devra donc être nulle. Sur les données considérées par les auteurs, la théorie du statut s'est avérée plus cohérente que la théorie de l'équilibre. Elle s'est montrée cohérente avec la surprise générative dans 14 cas sur 16 et avec la surprise réceptive dans 13 cas sur 16. La théorie de l'équilibre n'a été cohérente que dans, respectivement, 8 cas sur 16 et 7 cas sur 16.

2.4.3 Tâches étudiées dans le contexte signé

Si le cas signé a suscité l'intérêt de nombreux travaux, les tâches étudiées sont pourtant peu nombreuses et très majoritairement restreintes à la classification ou la prédiction du signe des liens entre les utilisateurs.

En ce qui concerne la classification, quelques travaux se sont intéressés aux méthodes de propagation d'étiquettes à base de marches aléatoires. Cependant, le signe soulève un gros problème puisque la matrice d'adjacence contient des valeurs négatives et ne peut alors pas être stochastique. Certains travaux ont tout simplement ignoré ces valeurs en les remplaçant par des 0 [Kamvar et al., 2003, Richardson et al., 2003]. D'autres travaux ont étendu cette approche de propagation à une matrice signée [Guha et al., 2004, Kunegis and Lommatzsch, 2009, Kunegis et al., 2010, Kunegis et al., 2009, Kerchove and Dooren, 2008], même si avec l'ajout de liens négatifs la convergence n'est pas garantie d'un point de vue théorique.

Le cas le plus étudié est finalement celui de la prédiction de liens signés où les liens sont déjà observés et il s'agit de prédire s'ils sont positifs ou négatifs ; c'est un problème de classification supervisée. Ainsi, alors que la prédiction de liens (classique) consiste généralement à associer à chaque paire de nœuds une étiquette (*relié* ou *pas relié*), il s'agit ici d'inférer le signe de liens connus. Ce protocole a été introduit dans [Guha et al., 2004] et repris par la suite [Leskovec et al., 2010a, Leskovec et al., 2010b, Chiang et al., 2011, Kunegis et al., 2010]. Parce que les graphes de terrain sont généralement déséquilibrés, c'est-à-dire qu'on y trouve souvent plus d'arcs positifs que négatifs, *Guha et al.* ont intégré au sein de leur apprentissage un re-équilibrage consistant à piocher un arc positif de manière aléatoire pour chaque arc négatif observé. Cela évite au modèle de sur-pondérer l'importance des exemples positifs au détriment des exemples négatifs - si le exemples sont trop déséquilibrées, alors prédire la classe aux exemples les plus nombreux permet d'augmenter les performances alors même que la qualité de la prédiction diminue.

Les méthodes à base de caractéristiques se sont avérées les plus adaptées pour la tâche de prédiction de liens signés [Leskovec et al., 2010a, Leskovec et al., 2010b, Chiang et al., 2011,Symeonidis et al., 2010]. Nous allons nous intéresser ici aux méthodes supervisées décrites dans [Leskovec et al., 2010a,Leskovec et al., 2010b] appliquant une régression logistique à partir de caractéristiques signées pour chaque couple de nœuds. Nous utiliserons ce protocole dans nos contributions.

La régression logistique est de la forme :

$$P(+|x) = \frac{1}{1 + e^{-(b_0 + \sum_i^n b_i x_i)}} \tag{2.13}$$

où x est le vecteur caractéristique du couple d'utilisateurs et les b_i les poids associés à chacune des caractéristiques, poids qui sont appris lors de la phase d'apprentissage.

Deux classes de caractéristiques sont définies dans [Leskovec et al., 2010a] : une première classe est basée sur une description des utilisateurs à travers le nombre et le signe de leurs arcs, la deuxième sur des statistiques liées aux voisins communs entre les sommets (basée sur des chemins de longueur 2).

— La première classe est composée de 7 caractéristiques. Le nombre d'arcs partants de u (degré sortant d_{out}) en distinguant les arcs positifs des arcs négatifs : $d^+_{out}(u)$, $d^-_{out}(u)$ et $d_{out}(u) = d^+_{out}(u) + d^-_{out}(u)$. Le nombre d'arcs en direction de v (degrés entrants d_{in}) selon le signe également :$d^+_{in}(v)$, $d^-_{in}(v)$ et $d_{in}(v) = d^+_{in}(v) + d^-_{in}(v)$. Enfin, le nombre de voisins communs aux deux sommets, noté $C(u, v)$.

— La seconde classe est composée de 16 caractéristiques associées aux 16 configurations de triades possibles impliquant l'arc (u, v) (théorie du statut). Ces 16 configurations correspondent aux 16 types de relations avec des voisins possibles que peuvent partager les utilisateurs u et v, selon le signe et la direction des liens de ces deux utilisateurs et le voisin.

Formellement, la valeur de ces 16 caractéristiques est donnée par la valeur (i, j) associée à chacune des 16 matrices possibles :

$$f_2(A^{b_1}) \cdot f_2(A^{b_2}) \tag{2.14}$$

avec A^+ est la matrice d'adjacence positive et A^- la matrice d'adjacence négative. $b_1, b_2 \in \{+, -\}$ et f_1, f_2 peuvent être l'identité ou la transposée.

Il est possible de compléter les 16 caractéristiques correspondantes aux 16 triades avec de nouvelles caractéristiques basées sur des cycles plus longs [Chiang et al., 2011]. La justification tient au fait que les caractéristiques précédentes ne prennent pas (ou peu) en compte les couples d'utilisateurs n'ayant pas de voisins, c'est-à-dire quand $C(u, v) = 0$. Il est possible de calculer ces caractéristiques pour des cycles de longueur $k \leq 5$:

$$f_1(A^{b_1}) \cdot f_2(A^{b_2}) \cdot f_3(A^{b_3}) \cdot ... \cdot f_{k-1}(A^{b_{k-1}}) \tag{2.15}$$

avec $b_i \in \{+, -\}$ et $f_i \in \{identité, transposée, 1\}$, ce qui donne 4^{k-1} caractéristiques possibles.

Les tâches exploitant d'autres sources d'information

Plus récemment *Tang et al.* [Tang et al., 2015] se sont intéressés au cas particulier où les utilisateurs pouvaient juger le contenu d'autres utilisateurs (des interactions indirectes). Ils sont partis de l'observation que ces évaluations étaient fortement corrélées aux liens sociaux signés entre les utilisateurs eux-mêmes. Or, les liens signés entre les utilisateurs sont une ressource extrêmement difficile à collecter, comme nous avons pu le constater au début de cette section. Les auteurs ont donc proposé un protocole permettant d'inférer les liens sociaux signés non pas à partir d'autres liens signés observés, mais à l'aide de ces interactions indirectes entre les utilisateurs. Une partie de nos travaux utilise ce même protocole, mais uniquement à partir d'observations sur des jugements communs à propos d'items et non plus via des interactions indirectes entre les utilisateurs. Le protocole est décrit dans le chapitre 5.

Notons également que le modèle de recommandation décrit dans le section 1.2.2 [Yang et al., 2012] permet aussi d'inférer le signe des liens entre les utilisateurs à partir de leurs jugements. Dans cette méthode, le signe des liens entre les utilisateurs est intégré dans la fonction de coût (voir équation 1.19) lors de la minimisation de l'erreur de prédiction, en

particulier via $\sum_{(u,i)}(t_{uv} - \hat{t}_{uv})^2$ où $\hat{t}_{uv}$ est le lien observé et t_{uv} la valeur inférée lors de l'apprentissage.

Les valeurs t_{uv} apprises sont alors utilisées pour la règle de prédiction suivante :

$$signe(t_{uv} - t_0) \tag{2.16}$$

où t_0 est un seuil permettant de modifier le pourcentage de liens négatifs souhaité. Ce seuil peut être fixé arbitrairement ou en inspectant sur un ensemble de validation le seuil optimal de sorte que la théorie du statut ou de l'équilibre soit le mieux respecté sur l'ensemble des données observées.

Conclusion

L'analyse des réseaux sociaux a suscité un intérêt croissant ces derniers années avec la démocratisation des réseaux en ligne, permettant aux utilisateurs de partager leurs contenus, leurs opinions, leurs souhaits. L'utilisation de plus en plus massive de ces réseaux ne fait qu'augmenter la difficulté de traitement puisque les données évoluent en quantité et en contenu de façon très rapide. À l'instar des systèmes de recommandation qui cherchent à guider les utilisateurs à travers l'ensemble des items disponibles, l'analyse des réseaux sociaux doit permettre aux utilisateurs de tirer profit de toute la richesse des contenus partagés.

Nous avons présenté trois applications majeures sur les réseaux sociaux : la détection de communautés, la classification de nœuds et la prédiction de liens. La première tâche analyse la structure des réseaux et cherche à regrouper les utilisateurs en communautés - des groupes de petite ou de grande taille où les utilisateurs sont davantage reliés entre eux qu'avec le reste du graphe. Le seconde tâche propose quant à elle d'attribuer des étiquettes aux utilisateurs pouvant indiquer leur appartenance politique, leur catégorie socio-professionnelle ou démographique, ou encore leur domaine d'expertise pour un réseau d'auteurs par exemple. La troisième tâche consiste à prédire de nouveaux liens entre utilisateurs à partir de liens observés à un instant t - une tâche très courante dans les réseaux sociaux où il s'agit alors de suggérer aux utilisateurs de nouvelles relations ou de nouvelles personnes à suivre.

Nous avons décrit plusieurs approches permettant de répondre à ces différentes problématiques. Nous avons pu remarquer que si elles étaient bien adaptées au cas classique où les liens ne sont pas signés, la présence de liens négatifs rend caduque une grande partie de l'arsenal mathématique à notre disposition. Pourtant, ces liens signés possèdent une valeur ajoutée [Kunegis et al., 2013], ce qui a suscité beaucoup d'intérêt ces dernières années. Outre les études portant sur l'interprétation de la sémantique de ces liens [Harary, 1953, Davis, 1977, Leskovec et al., 2010a] qui traduisent des relations antagonistes naturellement présentes, certains travaux [Guha et al., 2004, Kunegis and Lommatzsch, 2009] ont tenté de généraliser des méthodes à base de marches aléatoires au cas signé. D'autres travaux sur le sujet se sont focalisés sur des méthodes à base de caractéristiques [Leskovec et al., 2010b, Leskovec et al., 2010a, Tang et al., 2015] ou ont exploité indirectement la valeur ajoutée de ces liens pour améliorer d'autres tâches comme la recommandation [Yang et al., 2012].

Par ailleurs, nous avons pu constater que l'information négative - bien que pertinente et utile pour de nombreuses tâches non signées [Leskovec et al., 2010a, Yang et al., 2012] - est difficile à collecter puisque cela nécessite une démarche de la part des utilisateurs qu'ils considèrent eux-mêmes intrusive et malvenue. Dans cette thèse, nous proposons d'utiliser le comportement des utilisateurs pour inférer leur type de relation.

Deuxième partie

Contributions

3

Analyse de la sémantique des jugements communs dans le filtrage collaboratif

Résumé

Dans ce chapitre, nous nous intéressons à la sémantique de la polarité des jugements d'utilisateurs dans le cadre du filtrage collaboratif. Plus précisément, nous proposons d'étudier les accords entre les utilisateurs (lorsqu'ils partagent la même opinion sur un item) et en particulier, la différence entre les accords portant sur des items appréciés de ceux portant sur des items non appréciés. L'hypothèse explorée est que les avis positifs ont une sémantique et donc une utilité différente par rapport aux avis négatifs. Nous proposons un modèle de filtrage collaboratif (basé sur la factorisation matricielle) qui exploite la polarité et apprend à pondérer l'influence des voisins en se basant sur des caractéristiques liées à cette polarité. Des expériences effectuées sur *Yahoo ! Music* et *Epinions* permettent d'évaluer la validité de ce modèle.

Sommaire

Introduction

Dans le chapitre précédent, nous avons présenté quelques travaux focalisés sur l'étude des liens signés (c'est-à-dire des liens pouvant être positifs ou négatifs) permettant par exemple de prendre en compte des relations antagonistes naturellement présentes (ennemi, méfiance, dissimilarité). L'information apportée par ces liens négatifs soulève cependant quelques difficultés d'ordre sémantique, notamment lorsqu'il s'agit de prédire la polarité de nouvelles relations observées [Leskovec et al., 2010a], comme nous avons pu le constater précédemment. Néanmoins, cette information exploitée directement s'est révélée pertinente dans des contextes comme la recommandation [Ma et al., 2009, Victor et al., 2011, Yang et al., 2012] ou la prédiction de liens (non signés) [Kunegis et al., 2013, Leskovec et al., 2010a].

Bien que ces travaux aient révélé un intérêt pour ces liens signés, leur collecte demeure difficile. En effet, alors que quelques services autorisent les annotations négatives, celles-ci restent minoritaires (de l'ordre de 15 à 25% sur des sites comme Slashdot, Epinions ou Wikipedia [Leskovec et al., 2010a]). Nous pouvons supposer que l'utilisateur ne perçoit pas l'intérêt d'expliciter de tels liens alors que cela ne lui permet pas de construire un meilleur profil. C'est pourquoi nous nous intéressons dans ce chapitre à d'autres moyens de collecter des informations signées entre les utilisateurs (de façon indirecte), en particulier dans le cas de la recommandation. Lorsque les utilisateurs évaluent des items, les notes négatives sont perçues comme utiles et pertinentes parce qu'elles permettent d'affiner le profil et d'éviter les mauvaises recommandations. Il est d'ailleurs bien plus naturel pour l'utilisateur de pénaliser un item, plutôt qu'un utilisateur dans un groupe social. Cette richesse de contenu signé à notre disposition nous permet alors d'étudier les relations entre les utilisateurs.

Il était jusque-là communément admis que les accords, qu'ils soient positifs ou négatifs, sont synonymes de similarité. Or, lorsqu'un utilisateur exprime un avis négatif à propos d'un item, il est possible que cela n'apporte pas la même information qu'un avis positif : il peut s'agir par exemple d'un item qu'il n'a pas apprécié ou d'un item qui ne correspond pas à ses goûts. L'hypothèse que nous explorons dans ce chapitre est que la nature des accords, qu'ils concernent des items appréciés (accords positifs) ou non appréciés (accords négatifs), détermine la nature des relations entre les utilisateurs.

Nous allons utiliser une combinaison d'informations globales et locales pour une tâche de filtrage collaboratif, à l'aide de méthodes de factorisation matricielle où utilisateurs et produits sont représentés dans un même espace latent. Comme nous avons pu le voir dans le premier chapitre, ces méthodes ont souvent montré d'excellentes qualités [Koren et al., 2009] et exploitent une information globale à l'ensemble des utilisateurs. L'hypothèse que nous explorons est que la sémantique des préférences est différente suivant leur polarité. Les méthodes de filtrage collaboratif ne font pas cette distinction et donnent la même importance et la même sémantique à toutes les préférences utilisateurs. Nous allons montrer dans ce travail que cette différence est fondée et proposons un modèle permettant de l'exploiter.

Nos contributions sont les suivantes :

— Nous étudions la sémantique des jugements positifs et négatifs afin de formaliser le principe consistant à distinguer la polarité des accords.

— Nous proposons un modèle permettant d'intégrer cette caractéristique en apprenant à pondérer l'importance des voisins non seulement selon leur similarité, mais en fonction de caractéristiques liées à la polarité des accords.
— Nous évaluons ce modèle sur deux corpus représentatifs : Yahoo ! KDD Cup 2011 et Epinions.

Le chapitre est organisé de la façon suivante. Nous commençons par présenter l'intuition qui se cache derrière notre hypothèse, dans la section 3.1. Nous illustrons cette idée à travers quelques exemples. Ensuite, nous présentons dans la section 3.2 un modèle de filtrage collaboratif qui exploite cette polarité et nous détaillons en section 3.2.2 quelques mesures caractérisant la polarité des accords entre utilisateurs. Enfin, les expériences conduites et leurs résultats qualitatifs et quantitatifs sont détaillés dans la section 3.3.

3.1 L'intuition

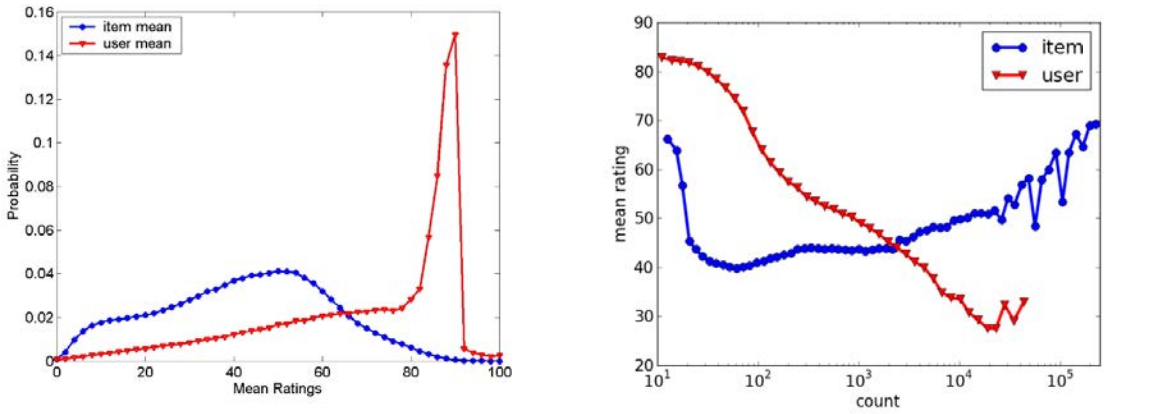
Dans ce chapitre, nous partons de l'hypothèse que les avis positifs et les avis négatifs n'ont pas la même sémantique et que les accords entre les utilisateurs, c'est-à-dire le fait de partager le même jugement, peuvent ne pas avoir le même sens selon qu'ils concernent des items appréciés ou non.

Nous supposons que les jugements peuvent être considérés de manière binaire (positifs/négatifs) soit directement lorsque l'utilisateur indique qu'il a apprécié un contenu ou non, soit indirectement quand nous disposons par exemple de jugements valués qu'il s'agit alors de transformer. Ainsi, l'historique des notes devient une source particulièrement pertinente quand il s'agit d'étudier les similarités entre utilisateurs puisqu'elle est naturellement plus abondante que les informations sociales.

Différence entre opinion positive et opinion négative pour l'utilisateur

L'hypothèse que nous faisons dans ce chapitre est que l'évaluation de la sémantique d'un jugement nécessite de prendre en compte le coût que représente l'évaluation d'un item pour l'utilisateur (en temps ou en argent par exemple). Nous pensons que lorsque le coût est important, l'utilisateur va d'abord choisir les items qu'il pense être plus susceptible d'apprécier. L'ensemble des items évalués (positivement et négativement) par l'utilisateur nous renseigne alors précisement sur ce qu'il recherche : cet ensemble restreint d'items correspond alors à *la zone thématique* de l'utilisateur. En revanche, si le coût est faible, l'utilisateur est susceptible d'étendre le panel des items évalués et de juger au-delà de sa *zone thématique*. L'acte est moins réfléchi, l'utilisateur évalue des items moins proches de ses centres d'intérêts : il élargit son champ de recherche. La conséquence est que le nombre potentiel d'items jugés négativement s'élargit également.

Nous supposons que les biais utilisateur découlent en partie du coût d'évaluation qui peut être différent selon les utilisateurs. Nous pensons également que ce coût peut dépendre du contexte et en particulier des items. Par exemple, le coût d'évaluation est plus élevé pour des produits chers d'un site de ventes à distance.



(a) Mean ratings distributions (b) Mean rating vs. the number of ratings

Figure 3.1 – *Caractéristiques des notes utilisateurs et items sur un corpus de musiques en ligne Yahoo ! Music. Illustration de [Dror et al., 2012].*

Au final, pour l'utilisateur, nous avons deux ensembles d'items : les items jugés positivement et les items jugés négativement.

Les items jugés positivement sont ceux qui correspondent le plus aux centres d'intérêts de l'utilisateur. Le coût d'évaluation n'a pas réellement d'impact sur la sémantique, mais plutôt sur la représentation de ces items dans l'ensemble des items évalués par un utilisateur. Lorsque l'évaluation représente un coût important, ces items sont sur-représentés car l'utilisateur fait davantage attention à ce qu'il évalue (le biais positivité). En revanche, quand le coût d'évaluation est faible, certaines données observées laissent penser que le biais positivité concerne les nouveaux utilisateurs qui évaluent en priorité les items qu'ils ont appréciés : on observe en effet sur la figure 3.1 (Yahoo ! Music) que plus les utilisateurs ont noté de morceaux de musique, plus la moyenne de leurs notes est faible (figure 3.1).

Les items jugés négativement sont ceux qui n'ont pas été appréciés par l'utilisateur. La sémantique peut dépendre alors du coût d'évaluation. Lorsque ce coût est élevé, un item jugé négativement a une sémantique forte puisque l'utilisateur pensait apprécier l'item. C'est alors un item qui fait partie de la zone thématique de l'utilisateur, mais qui n'est pas à son goût. En revanche, lorsque le coût est faible, les utilisateurs évaluent de manière plus ouverte et un item jugé négativement peut simplement indiquer que cet item ne correspond pas aux centres d'intérêts de l'utilisateur. Cette situation se retrouve plus régulièrement sur certaines plates-formes où des items sont suggérés en continu à l'utilisateur (radio en ligne ou fil de vidéos partagées par exemple).

Accords et désaccords

Lorsque deux utilisateurs partagent des jugements, ils peuvent partager des jugements positifs communs (ou accords positifs), des jugements négatifs communs (accords négatifs), ou encore des jugements contraires (désaccords). Alors que les accords positifs et les désaccords sont sans équivoque, notre intérêt se porte sur les accords négatifs dont la sémantique peut

dépendre des utilisateurs ou des items.

Exemple : Prenons le cas de quatre utilisateurs partageant sept musiques (figure 3.1) dans trois catégories musicales : la musique Rock 'n' roll, le Blues et la musique Pop-Rock partageant des caractéristiques avec les deux précédentes catégories.

Ce que nous constatons en premier lieu est que les couples utilisateurs (1,2) puis (3,4) partagent une grande partie de leur zone thématique. Les utilisateurs 1 et 2 partagent les mêmes goûts au sein de la musique Rock 'n' roll, mais sont en désaccord sur un item pop-rock. Ils ont néanmoins des goûts similaires, tout comme le couple (3,4). En revanche, ce que nous remarquons est que les utilisateurs 1 et 4 ne semblent pas avoir des goûts similaires puisqu'ils ne partagent aucune note parmi les items jugés positivement, représentatifs de leurs goûts. Pourtant, leurs seuls jugements communs sont des accords.

Cet exemple illustre le fait que des utilisateurs peuvent partager des accords négatifs communs sans pour autant avoir des goûts similaires. À l'inverse, des utilisateurs aux goûts similaires peuvent partager davantage de désaccords que des utilisateurs faiblement similaires (utilisateur 2 et 3 par exemple) qui ne noteront en commun que des items éloignés de leur zone thématique (donc de manière négative).

	Évaluation de musiques						
	Rock 'n' roll		Pop Rock			Blues	
	Track A	Track B	Track C	Track D	Track E	Track F	Track G
Utilisateur 1	+	+	-	-			
Utilisateur 2	+	+	+	-	-		
Utilisateur 3			-		+	+	+
Utilisateur 4			-	-	+	+	+

Table *3.1* *– Ensemble de notes communes pour quatre utilisateurs donnés sur un service d'évaluation de musique.*

Notre hypothèse

Notre hypothèse est que la similarité de goût entre les utilisateurs tient en majorité des accords positifs qui ne sont pas équivoques et dont la sémantique ne dépend pas de la disposition des utilisateurs à noter en dehors de leur zone thématique. Les accords négatifs n'apportent pas de preuves de similarité car les items concernés peuvent ne pas faire partie de la zone thématique des utilisateurs (lorsque le coût d'évaluation est faible). Les désaccords peuvent aussi avoir une sémantique moins forte puisque deux utilisateurs qui ne partagent pas les mêmes goûts (zones thématiques éloignées) ne seront pas (ou peu) en désaccord.

À partir de là, pour deux utilisateurs partageant suffisamment d'avis communs, nous définissons trois niveaux de similarité possibles :

— **Similarité *forte*** : leurs zones thématiques sont semblables, les utilisateurs partagent principalement des accords positifs.

— **Similarité *moyenne*** : leurs zones thématiques se chevauchent, mais ne se confondent pas. Les deux utilisateurs partagent beaucoup d'accords positifs (l'intersection de leur zone thématique), mais également des désaccords.

— **Similarité *faible*** : leurs zones thématiques ne s'intersectent pas. Les utilisateurs partagent principalement des jugements négatifs.

D'un point de vue empirique, il est possible de distinguer ces trois cas de figure. Pour cela, nous pouvons étudier la figure 3.2 représentant la proportion d'accords (parmi l'ensemble de leur notes communes) entre deux utilisateurs selon le nombre d'accords positifs et négatifs communs : une zone rouge indique que les utilisateurs ont majoritairement les mêmes avis sur les items et une zone bleue correspond à une zone où les utilisateurs ont un écart de notation très important. Ces statistiques ont été calculées sur le corpus Yahoo ! KDD utilisé dans la suite (section 3.3.1).

Nous distinguons trois zones correspondant aux trois états :

— **Similarité *forte*** au nombre d'accords négatifs très faible (*en bas* sur la figure 3.2) : les utilisateurs partagent beaucoup d'accords positifs (*à droite*), très peu d'accords négatifs et ne sont quasiment jamais en désaccord (zone rouge).

— **Similarité *moyenne*** au nombre d'accords négatifs et d'accords positifs équivalents (*au centre*) : les utilisateurs sont également en désaccord (zone verte, autant d'accords que de désaccords).

— **Similarité *faible*** au nombre d'accords négatifs élevé et nombre d'accords positifs faible (*en haut à gauche) de la figure 3.2* : les utilisateurs partagent beaucoup d'accords et peu de désaccords (zone rouge). La majorité de ces accords sont des jugements négatifs communs.

3.2 Le modèle

Afin de vérifier notre hypothèse sur des données réelles, notre travail s'appuie sur les techniques dérivées de la factorisation matricielle [Zhou et al., 2008] qui ont pour caractéristique principale de représenter utilisateurs et produits dans un espace latent commun. Comme nous avons pu le voir dans le premier chapitre, ces méthodes permettent de développer une grande variété de modèles en intégrant sous forme de contraintes des caractéristiques additionnelles des données. Par exemple, nous trouverons dans [Chen et al., 2011] 81 variantes de la factorisation matricielle appliquées aux données Yahoo !.

De nombreuses techniques de filtrage collaboratif reposent sur l'utilisation explicite du voisinage d'un utilisateur et/ou d'un item [Bobadilla et al., 2013], ce qui permet d'exploiter directement une information locale à un utilisateur. Dans notre cas, nous allons définir un modèle de factorisation matricielle prenant en compte l'information locale à travers la nature des accords entre les utilisateurs. Le modèle va notamment exploiter les représentations latentes des voisins afin de minimiser l'erreur de prédiction ; il va calculer un poids w_{uv} pour chaque couple d'utilisateurs dont la valeur va dépendre de caractéristiques prenant en compte

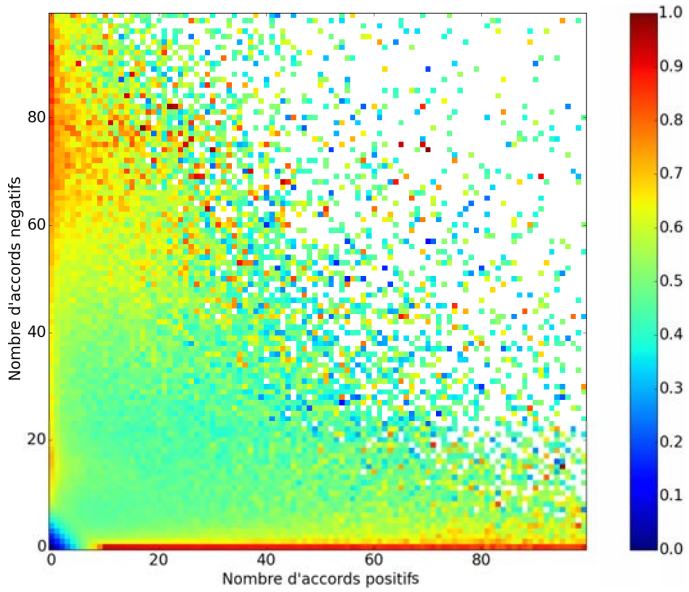


FIGURE *3.2 – Le pourcentage d'accords pour un couple d'utilisateurs donné, en fonction du nombre d'avis positifs et négatifs en commun. Une zone en rouge indique que les utilisateurs ne partagent que des accords et aucun désaccord. À l'inverse, une zone en bleu indique que les utilisateurs partagent uniquement des désaccords*

la polarité des accords. Nous cherchons à ce que le modèle pondère par lui-même l'importance de chaque voisin selon le signe des accords qu'ils partagent.

3.2.1 Représentation latente et voisinage

Avant de détailler notre approche, rappelons le contexte dans lequel s'inscrit notre modèle. Comme pour les principaux modèles utilisés dans [Chen et al., 2011], [Koren and Bell, 2011] ou encore [Yang et al., 2012], nous nous basons sur l'utilisation d'un espace latent $\mathbb{R}^n$ dans lequel sont représentés l'ensemble des utilisateurs et des produits (section 1.2) : la note prédite $\hat{r}_{ui}$ de l'utilisateur $u \in U$ pour le produit $i \in I$ est estimée à partir de vecteurs latents pour l'utilisateur (p_u) et le produit (q_i) :

$$\hat{r}_{ui} = \mu + b_i + b_u + {p_u}^T q_i \tag{3.1}$$

où μ est la moyenne des notes du corpus, b_i la moyenne des notes du produit i et b_u des notes de l'utilisateur u.

L'apprentissage de ce type de modèle se fait en général en minimisant l'erreur $e_{ui} = r_{ui} \quad \hat{r}_{ui}$ sur un ensemble de notes observées. Pour éviter tout phénomène de sur-apprentissage, nous ajoutons à la fonction objectif un terme de régularisation, ce qui nous donne en norme L2 :

$$min_{q_i,p_u \in \mathbb{R}^n} \sum_{(u,i)\in T} (r_{ui} - \mu - b_i - b_u - q_i^T p_u)^2 + \lambda(\|q_i\|^2 + \|p_u\|^2 + b_i^2 + b_u^2) \tag{3.2}$$

La somme étant prise sur toutes les notes utilisées pour l'apprentissage des représentations.

Exploiter le voisinage

Dans cette représentation, il est facile de voir que plus les utilisateurs donnent les mêmes notes sur les mêmes items, plus ils auront des représentations proches et que cette proximité dépend uniquement de la similarité des jugements entre utilisateurs.

Afin de prendre en compte le voisinage, nous pouvons modifier l'équation (3.1) en modifiant la note prédite en fonction de celles prédites par les voisins :

$$\hat{r}_{ui} = \mu + b_i + b_u + \sum_{v \in V_u \cup \{u\}} w_{uv} p_v^T q_i \tag{3.3}$$

où V_u est l'ensemble des voisins de u et w_{uv} une similarité entre u et v.

C'est ce type de formulation qu'exploitent par exemple [Yang et al., 2012,Bell and Koren, 2007]. Les approches diffèrent dans l'expression de w_{uv} et la détermination du voisinage. *Yang et al.* [Yang et al., 2012] supposent qu'un graphe social décrivant les relations entre les utilisateurs est disponible et que le poids w_{uv} est directement proportionnel à la similarité entre deux utilisateurs dans l'espace latent (p est le poids donné à la représentation de l'utilisateur cible) :

$$w_{uv} = \begin{cases} p & \text{si } v = u \\ (1-p) p_u \cdot p_v & \text{sinon} \end{cases} \tag{3.4}$$

Ce qui est décrit dans la section 1.2.2.

Hélas, le graphe social n'est pas toujours disponible. Dans ce cas, *Bell et al.* [Bell and Koren, 2007] exploitent un voisinage implicite où les voisins V d'un utilisateur sont ses k plus proches voisins calculés dans l'espace des notes, obtenus grâce à une fonction de similarité prenant en compte l'historique des notes ; w_{uv} est alors proportionnel à la similarité entre utilisateurs.

3.2.2 Modèles

Dans ce chapitre, nous utilisons également la formulation générale donnée dans l'équation (3.3) et exploitant un voisinage implicite. Contrairement aux approches les plus courantes, le poids w_{uv} n'est pas donné *a priori* par une fonction de similarité basée sur la représentation des nœuds, mais est appris [Koren, 2008]. Ce poids est une fonction des caractéristiques qui utilisent la polarité des jugements communs à deux utilisateurs.

Les poids sont exprimés sous la forme d'une fonction logistique qui dépend d'un ensemble de caractéristiques Φ qui caractérisent la relation entre les deux utilisateurs :

$$w_{uv} = \sigma(\tau \cdot \Phi) \tag{3.5}$$

où $\sigma(x) = (1 + e^{-x})^{-1}$ désigne la fonction logistique, τ est un vecteur de pondérations qui est appris et $\Phi = (1, \phi_1, \ldots, \phi_n)^T$ est l'ensemble des caractéristiques, la valeur «1» correspond à un terme de biais.

Nous notons J_u l'ensemble de tous les items notés par u. Parmi cet ensemble d'items, nous distinguons J_u^+ (respectivement J_u^-) l'ensemble des items ayant reçu une note positive (resp. négative) de l'utilisateur u. Pour deux utilisateurs u et v, nous notons D_{uv} = l'ensemble des items sur lesquels ils sont en désaccord et $A_{uv} = J_u \cap J_v$ l'ensemble des items sur lesquels ils sont en accord. Enfin, nous distinguons l'accord positif (ils aiment le même produit) de l'accord négatif (ils n'aiment pas le même produit). Ainsi, nous avons $A_{uv} = A_{uv}^+ \cup A_{uv}^-$ et, par exemple, $A_{uv}^+ = J_u^+ \cap J_v^+$.

Nous définissons les quatre caractéristiques suivantes :

Taux d'accord positif sur l'ensemble des accords Cette première caractéristique permet de caractériser parmi les utilisateurs similaires (avec des jugements en commun similaires) ceux qui ont une proportion importante d'accords positifs. Elle favorise les utilisateurs qui ont principalement aimé les mêmes produits. Formellement,

$$\phi_1(u, v) = \frac{|A_{uv}^+|}{|A_{uv}|} \tag{3.6}$$

Notons que cette mesure ne tient compte que des accords entre utilisateurs.

Taux d'accord positif sur l'ensemble des notes communes Cette caractéristique est similaire à la première à la différence qu'elle prend également en compte les désaccords. Ainsi, elle sera proche de 1 pour les utilisateurs qui sont peu souvent en désaccord et aiment les mêmes items.

$$\phi_2(u, v) = \frac{|A_{uv}^+|}{|A_{uv} \cup D_{uv}|} \tag{3.7}$$

Taux d'accord négatif sur l'ensemble des notes communes Cette caractéristique est équivalente à la précédente, sauf que nous considérons le cas des accords négatifs : la caractéristique est proche de 1 lorsque les utilisateurs sont du même avis à propos d'items qu'ils n'ont pas appréciés.

$$\phi_3(u, v) = \frac{|A_{uv}^-|}{|A_{uv} \cup D_{uv}|} \tag{3.8}$$

Notons que ces trois caractéristiques ne prennent pas en compte le nombre total d'items que les utilisateurs ont notés. Par exemple, deux utilisateurs partageant dix accords positifs (et rien d'autre) seront considérés similaires par la mesure $\phi_2(u, v)$, même si ces dix accords ne représentent qu'une infime partie des items qu'ils ont notés. Pour cela nous introduisons la caractéristique suivante.

Indice de Jaccard «positif» Cette caractéristique, combinée aux trois premières, permet de prendre en compte le nombre total de produits notés par les deux utilisateurs et ainsi d'évaluer l'importance des notes partagées. L'indice permet ainsi de nuancer les cas où deux utilisateurs seraient fortement en accord positif, mais que l'ensemble des notes communes aux deux utilisateurs ne serait pas significatif. L'indice de Jaccard correspond au rapport entre les notes communes et l'ensemble des notes données par les utilisateurs du couple. Nous utiliserons ici une version «positive» de cet indice calculé comme le rapport entre les accords positifs communs et l'ensemble des notes positives des deux utilisateurs.

$$\phi_4(u,v) = \frac{|A^+_{uv}|}{|J^+_u \cup J^+_v|} \tag{3.9}$$

Ces quatre caractéristiques ont été sélectionnées parmi un ensemble de caractéristiques candidates que nous avons définies et testées. Celles-ci nous permettent en particulier de distinguer les différentes zones observées sur la figure 3.2.

3.3 Les expériences

Dans cette section, nous présentons les données que nous utilisons pour l'ensemble des expériences. Nous définissons les différents protocoles utilisés pour nos modèles et discutons de l'étude empirique effectuée pour mesurer le potentiel de notre approche.

3.3.1 Jeux de données

Yahoo ! KDD Cup 2011

Ce corpus dispose d'une grande quantité de notes. Il s'agit d'une collecte d'une durée de 10 ans sur le site Yahoo ! Music, ce qui représente 262.810.175 notes et 1.000.090 utilisateurs. Construit pour la KDD Cup 2011, le corpus propose trois ensembles de notes pour l'apprentissage, la validation et le test des modèles. Les notes sont comprises entre 0 et 100. La moyenne de l'ensemble des notes est aux alentours de 48 sur 100. Nous avons binarisé les notes de la façon suivante : celles inférieures ou égales à 30 sur 100 sont considérées comme négatives, celles supérieures ou égales à 70 sont considérées positives. Les autres notes sont ignorées. Deux utilisateurs sont en accord si leur note est du même signe.

Epinions

Le corpus est composé de 132.000 utilisateurs et contient 1.560.144 articles et 13.668.319 notes. Les notes sont comprises entre 0 et 5, la moyenne est aux alentours de 4. Une note inférieure (strictement) à la moyenne est considérée comme négative, une note supérieure (strictement) à la moyenne sera positive [Yang et al., 2012], les autres sont ignorées. Ce corpus n'est pas décomposé en ensembles d'apprentissage et de test. Pour les évaluations, nous avons utilisé une validation croisée sur 3 échantillons (70% apprentissage, 30% test).

3.3.2 Algorithme d'apprentissage

Les modèles présentés en section 3.2.2 sont appris grâce à une descente de gradient stochastique. Cette approche est très utilisée dans la littérature, notamment dans [Chen et al., 2011, Koenigstein et al., 2011, Wu et al., 2011], car elle allie une facilité d'implémentation, une rapidité d'exécution et de bons résultats. Nous avons utilisé une descente alternée, c'est-à-dire en modifiant alternativement un ensemble de paramètres (utilisateurs/item/w_{uv}), les autres étant fixés. Nous parcourons donc chaque exemple d'apprentissage et mettons à jour successivement les vecteurs latents représentant les utilisateurs et les produits, puis, les paramètres w de notre fonction logistique.

La fonction de coût 3.2 est utilisée avec le modèle de recommandation suivant :

$$\hat{r}_{ui} = \mu + b_i + b_u + \left((1 - \frac{\sum_{v \in V_u} w_{uv}}{2k}) p_u + \frac{\sum_{v \in V_u} w_{uv} p_v}{2k} \right)^T \cdot q_i \tag{3.10}$$

En notant $e_{ui} = r_{ui} - \hat{r}_{ui}$, cela nous donne pour chaque mise à jour :

$$p_u \leftarrow p_u + 2\gamma \left(e_{ui} \left(1 - \frac{\sum_{v \in V} w_{uv}}{2k} \right) q_i - \lambda p_u \right) \tag{3.11}$$

$$q_i \leftarrow q_i + 2\gamma \left(e_{ui} \left((1 - \frac{\sum_{v \in V} w_{uv}}{2k}) p_u + \frac{\sum_{v \in V} w_{uv} p_v}{2k} \right) - \lambda q_i \right) \tag{3.12}$$

$$\tau \leftarrow \tau + \gamma_\tau \left(e_{ui} \frac{\sum_{v \in V} \phi_{uv} \sigma(\tau \cdot \phi_{uv})(1 - \sigma(\tau \cdot \phi_{uv}))}{2k} (p_v - p_u)^T q_i \right) \tag{3.13}$$

où γ et γ_τ sont les pas de gradient, et λ et λ_τ sont les coefficients de régularisation correspondants [1]. Notons que pour les modèles sans prise en compte du voisinage, w_{uv} est égal à zéro sauf lorsque $u = v$, et la mise à jour (3.13) n'est pas utilisée.

3.3.3 Calcul du graphe de similarité

Compte tenu de la quantité de données et de la taille de la matrice utilisateur×utilisateur utilisée dans les expériences pour calculer les voisinages, nous avons utilisé des heuristiques pour limiter le nombre de voisins considérés. Plus précisément, pour un utilisateur donné, nous gardons les 50 utilisateurs les plus en accord avec lui (en proportion des notes communes). Ensuite, nous ajoutons à notre présélection la même quantité d'utilisateurs choisis aléatoirement (dans l'ensemble des personnes ayant au moins un produit jugé en commun).

Ces 100 voisins sélectionnés pour chaque utilisateur permettent alors d'apprendre les poids de chaque caractéristique de la fonction w. Le premier ensemble de 50 voisins permet d'évaluer l'impact de la prise en compte du voisinage en terme de performances en prédiction,

1. nous avons un pas différent pour les poids des représentations latentes et pour les paramètres de la fonction logistique. Par exemple, pour les données Yahoo ! utilisées dans la suite, nous avons $\gamma = 3 \times 10^{-4}$, $\lambda = 1 \times 10^{-2}$, $\gamma_\tau = 1 \times 10^{-7}$ et $\lambda_\tau = 1 \times 10^{-2}$

alors que le second ensemble de 50 voisins pris au hasard nous permet d'assurer que les w soient correctement appris en évitant de sur-pondérer les voisins aux goûts similaires - cela force le modèle à apprendre des poids w faibles pour les utilisateurs aux goûts non similaires.

3.4 Les résultats

3.4.1 Évaluation qualitative

Évaluation indirecte

Dans un premier temps, nous voulons étudier la pertinence des caractéristiques proposées. Pour cela, nous introduisons une mesure pour évaluer si les utilisateurs ont des goûts similaires. Cette mesure d'affinité est basée sur leur voisinage commun et doit nécessairement être différente des caractéristiques que nous utilisons pour notre modèle. Idéalement, cette mesure devrait nous indiquer si les deux utilisateurs font partie d'un groupe cohérent d'utilisateurs qui ont une opinion positive des mêmes produits (d'une même zone thématique).

Dans la pratique, nous procédons de la façon suivante. Pour un couple d'utilisateurs u_1 et u_2, nous considérons comme voisinage commun l'ensemble V des utilisateurs qui ont noté au moins un item en commun avec chacun des deux utilisateurs. Si cet ensemble n'est pas vide, alors il y a au moins un utilisateur v_1 qui a noté un item en commun avec u_1 et un item en commun avec u_2. À partir de là, nous définissons notre mesure d'affinité entre les utilisateurs u_1 et u_2 comme la proportion de voisins dans V qui partagent un *accord positif* avec u_1 et un accord positif avec u_2 (éventuellement sur le même item). Lorsque cette mesure est nulle, cela signifie qu'il n'y a aucun voisin dans l'ensemble V qui partage une opinion positive commune avec u_1 et u_2. Inversement, une mesure à 1 indique que n'importe quel voisin $v \in V$ du couple partage un avis positif avec chacun des deux utilisateurs — ce qui quantifie cette notion de groupe cohérent.

Formellement, la mesure d'affinité peut être définie de la manière suivante :

$$\text{Aff}(u_1, u_2) = \frac{|\{v \in V | (A^+_{u_1,v} \neq \emptyset) \wedge (A^+_{u_2,v} \neq \emptyset)\}|}{|\{w \in V | (J^+_{u_1,w} \neq \emptyset) \wedge (J_{u_2,w} \neq \emptyset)\}|} \tag{3.14}$$

Nous avons sélectionné aléatoirement 25 000 000 couples d'utilisateurs sur un échantillon du corpus Yahoo ! pour lesquels nous avons calculé l'ensemble des caractéristiques décrites en section 3.2.2. Nous avons calculé notre mesure d'affinité en fonction des caractéristiques proposées. Nous décrivons ici les observations correspondantes pour deux d'entre elles : le taux d'accords positifs sur l'ensemble des notes communes (équation 3.7) et l'indice de Jaccard. Les figures 3.4 et 3.3 donnent en abscisse les quantiles d'une caractéristique (on considère 25 quantiles) et en ordonnée la proportion de voisins communs qui sont en accord positif au moins une fois avec chacun des deux utilisateurs du couple considéré. Elles montrent que la première caractéristique est plus fortement corrélée à la similarité de voisinage que l'indice de Jaccard qui ne prend pas en compte la polarité des jugements. Dans le cas de la première caractéristique, les couples au ϕ_2 faible partagent beaucoup moins de voisins avec qui ils

partagent des jugements positifs que ceux au ϕ_2 élevé (plus de 80%). Avec l'indice de Jaccard, la moyenne pour l'ensemble des quantiles ne varie pas énormément (entre 80% et 90%), ce qui est beaucoup moins informatif.

Nous pouvons observer une corrélation similaire entre les autres caractéristiques introduites en section 3.2.2 et la qualité du voisinage. Pour la caractéristique prenant en compte la quantité de jugements négatifs, la progression est inverse. Pour la caractéristique normalisée sur l'ensemble des accords, la corrélation est moins nette et moins marquée que celle observable sur 3.4.

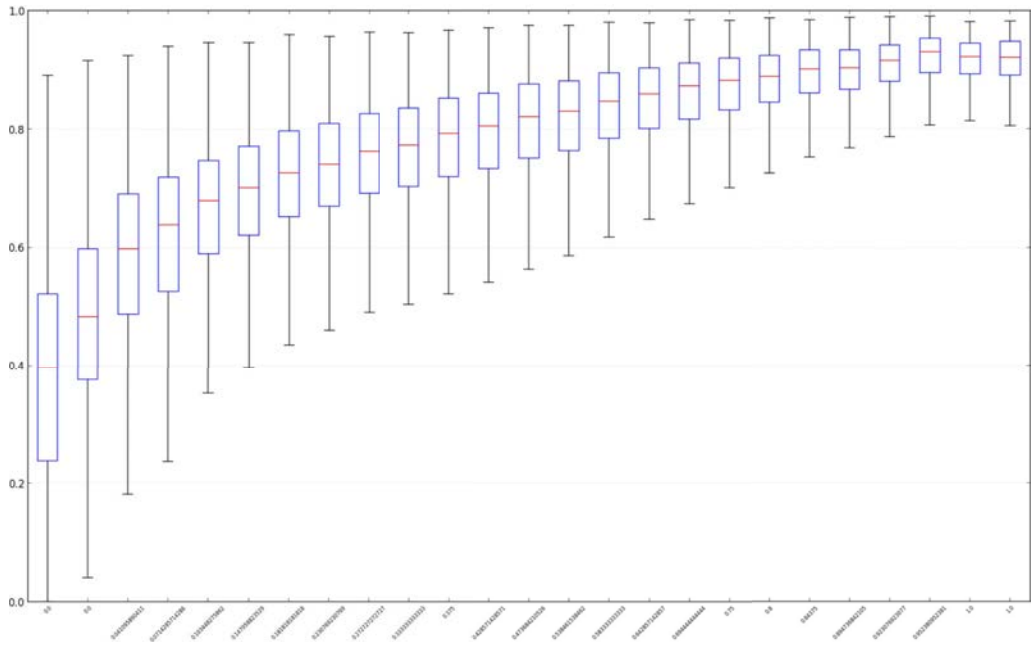


Figure ***3.3*** *– La mesure d'affinité en fonction du taux d'accord positif sur l'ensemble des notes communes. Chaque quantile représente 1 000 000 de couples d'utilisateurs.*

Sur la figure 3.5, nous avons reproduit le même type de graphique, mais avec en abscisse la valeur des poids w appris. Les valeurs élevées de w correspondant bien à une affinité élevée, avec le même effet de seuil déjà observé pour la caractéristique $\phi_2(u, v)$ sur la figure 3.3. Notre modèle semble donc avoir appris des coefficients pertinents et la nouvelle mesure w apprise semble donc plus discriminante pour prédire notre mesure d'affinité.

Analyse des coefficients appris

La table (3.2) donne les coefficients obtenus pour le vecteur de la régression logistique (colonne 2). Les coefficients mesurant les accords positifs sont supérieurs à zéro et tendent à faire augmenter la valeur de w, le coefficient correspondant à ϕ_2 étant de loin le plus élevé. Le coefficient correspondant aux accords négatifs tend à abaisser la valeur de w. Ainsi deux utilisateurs partageant uniquement des accords négatifs auraient un w proche de 0, alors que deux utilisateurs partageant principalement des accords positifs auraient un w proche

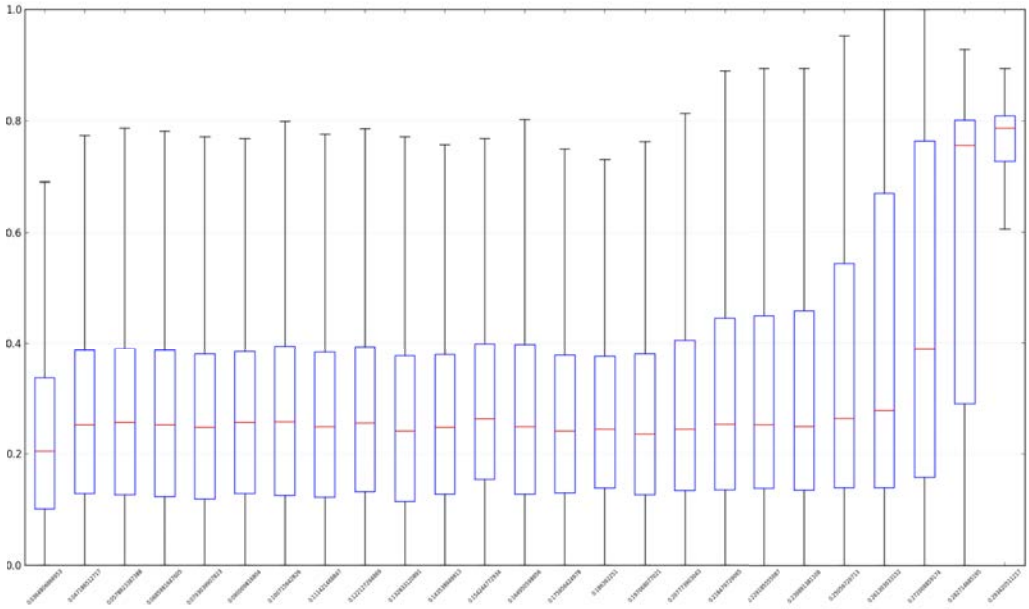


Figure *3.4* – *La mesure d'affinité en fonction de l'indice de Jaccard. Chaque quantile représente 1 000 000 de couples d'utilisateurs.*

de 1. Cette observation tend à valider l'hypothèse de différentiation des jugements positifs et négatifs pour la recommandation. La principale information apportée par ces résultats est que le taux d'accords négatifs sur l'ensemble des accords a un impact négatif sur la mesure de similarité w. Autrement dit, posséder des accords négatifs ne joue pas en faveur d'une plus grande similarité. Deux personnes ayant un taux élevé d'accords négatifs seront donc moins proches (au sens de notre mesure de similarité w apprise) que deux utilisateurs ayant uniquement des accords positifs.

Les colonnes suivantes de la table (3.2) donnent des exemples de vecteurs de caractéristiques calculés sur des couples d'utilisateurs. Les utilisateurs u et v_1 sont principalement en désaccord et w_{uv_1} est nul. Pour les deux autres couples (u, v_2) et (u, v_3), les utilisateurs sont principalement en accord sur les jugements qu'ils ont donnés. La différence se situe alors sur la proportion de jugements communs négatifs. Les utilisateurs u et v_3 partageant plus de produits qu'ils ont aimés et sont donc davantage similaires que u et v_2. Ce qui se traduit par une similarité w_{uv_2} inférieure.

3.4.2 Évaluation quantitative

Nous avons également évalué notre modèle dans le cadre de la prédiction de notes et de l'ordonnancement afin de mesurer l'amélioration que peut apporter la prise en compte de la polarité.

Pour évaluer nos modèles en prédiction de note, nous avons utilisé la mesure standard du domaine, la moyenne des erreurs quadratiques (RMSE) entre la note prédite et la note réelle. Pour l'évaluation en ordonnancement, nous utilisons la mesure de précision moyenne

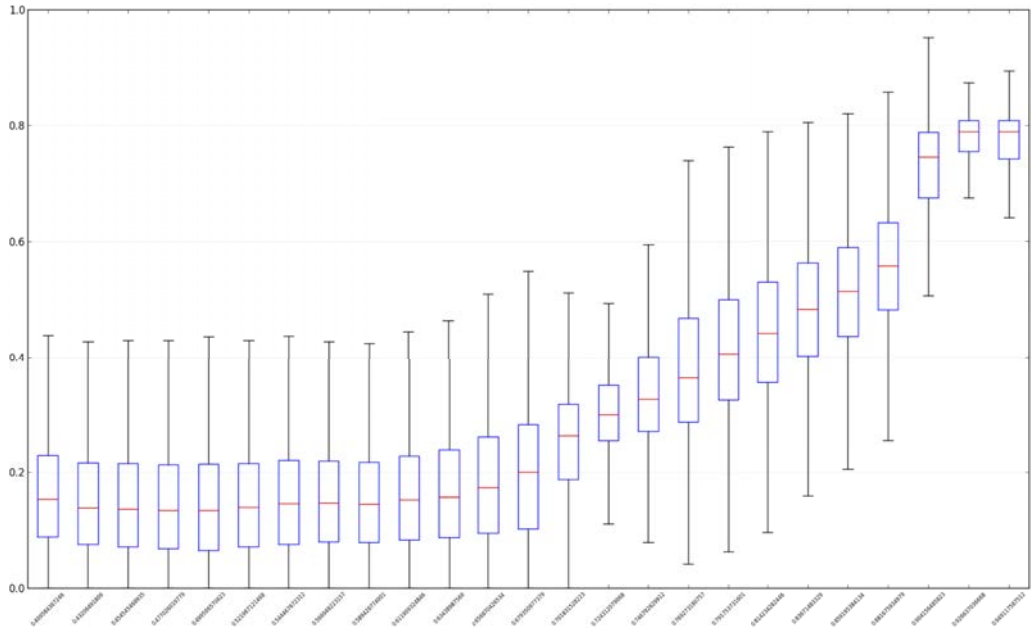


FIGURE ***3.5*** *– La mesure d'affinité selon la valeur de notre w appris*

	τ	Φ_{uv_1}	Φ_{uv_2}	Φ_{uv_3}
$\phi_1 : \lvert A_{uv}^+ \rvert / \lvert A_{uv} \rvert$	0.11	0.4	0.25	1.0
$\phi_2 : \lvert A_{uv}^+ \rvert / \lvert A_{uv} \cup D_{uv} \rvert$	6.74	0.11	0.75	0.75
$\phi_3 : \lvert A_{uv} \rvert / \lvert A_{uv} \cup D_{uv} \rvert$	-1.65	0.17	0.24	0.00
$\phi_4 : \lvert A_{uv}^+ \rvert / \lvert J_u^+ \cup J_v^+ \rvert$	0.55	0.12	0.08	0.19
1 : Biais	-5.20	1	1	1
w_{uv_i}		0.0	0.38	0.52

TABLE ***3.2*** *– Le vecteur τ des coefficients de la régression logistique. Exemples de poids w obtenus pour trois couples $(uv_1, uv_2, uv_3$ de plus en plus similaires)*

MAP. Elle correspond à la moyenne, sur l'ensemble des utilisateurs, de la précision moyenne de chaque produit de l'ensemble de test (voir section 1.3.1). Le modèle de référence est une factorisation matricielle correspondant aux équations (1) et (2).

Les résultats sont donnés dans le tableau 3.3. Comme nous pouvons l'observer, le modèle avec prise en compte du voisinage local et apprentissage des coefficients permet d'améliorer les résultats en ordonnancement et en prédiction de notes pour le corpus Yahoo ! et ceci, quel que soit le nombre de notes connues et la dimension D de l'espace latent. En effet, nous observons pour la MAP un gain de l'ordre de 3 à 8% et pour le RMSE, une amélioration de l'ordre de 1 à 2%. Notre mesure de similarité w apporte donc une information pertinente pour notre apprentissage.

Pour le corpus Epinions, nous obtenons un gain en ordonnancement, alors que nous perdons en performance dans le cadre de la prédiction de notes. Le voisinage construit ne

	Epinions			
	D = 25		D = 50	
	RMSE	MAP	RMSE	MAP
Modèle standard	**0.553**	0.128	**0.675**	0.125
Modèle avec Voisinage	0.608	**0.129**	0.798	**0.132**
	Yahoo ! — 10% des notes			
	D = 25		D = 50	
	RMSE	MAP	RMSE	MAP
Modèle standard	28.27	0.565	28.10	0.491
Modèle avec Voisinage	**27.60**	**0.589**	**27.55**	**0.535**
	Yahoo ! — 100% des notes			
	D = 25		D = 50	
	RMSE	MAP	RMSE	MAP
Modèle standard	26.02	0.556	26.18	0.505
Modèle avec Voisinage	**25.78**	**0.583**	**25.88**	**0.521**

Table 3.3 – *Résultats RMSE et MAP sur trois jeux de données : Epinions, un sous-corpus de Yahoo ! où l'on ne considère que 10% des notes les plus récentes et l'ensemble de corpus Yahoo !*

semble donc pas suffisamment informatif dans le cadre de la prédiction de notes, mais permet d'améliorer l'ordonnancement qui profite de l'ordre établi par les produits les plus notés par les voisins. La différence notable entre les deux corpus est que le nombre de notes est bien plus faible dans Epinions ; puisque l'utilisation du voisinage local apporte davantage de paramètres à apprendre, un nombre d'exemples d'apprentissage moins élevé pénalise donc de fait les performances en prédiction de notes.

Le fait que la mesure MAP soit améliorée est un indice supplémentaire de la validité de notre hypothèse selon laquelle les voisins les plus proches sont ceux qui partagent essentiellement des accords positifs.

Conclusion

Dans ce chapitre, nous nous sommes intéressés à l'utilisation de la polarité des jugements dans une tâche de filtrage collaboratif. Partant de l'hypothèse que les jugements positifs et négatifs ont des sémantiques différentes, nous avons proposé un modèle utilisant des informations locales de voisinage utilisateur qui exploite cette dissymétrie entre les deux polarités. En se basant sur des caractéristiques mesurant cette polarité, le modèle apprend à partir d'exemples à pondérer l'influence de ses voisins.

Des expériences réalisées sur deux jeux de données ont montré l'intérêt de l'apprentissage de la prise en compte de la polarité des opinions. Sur les deux jeux de données utilisés,

cette information permet d'augmenter la mesure MAP, plus représentative de la tâche de recommandation (c'est-à-dire suggérer en priorité les items les plus pertinents). Sur la mesure RMSE qui montre la qualité de l'approximation, les performances sont améliorées sur l'un des jeux de données et détériorées sur l'autre (celui au nombre de jugements beaucoup plus faible).

Finalement, notre travail ouvre des pistes quant à la prise en compte du voisinage dans les modèles de filtrage collaboratif basés sur un espace latent, en apprenant dans quelle mesure deux voisins sont similaires. Des extensions sont possibles, en particulier en prenant en compte le graphe social (quand celui-ci existe) et la polarité des relations entre utilisateurs.

4

Modèles utilisateur pour l'analyse de la sémantique des jugements communs

Résumé

Dans ce chapitre, nous poursuivons l'étude de la polarité des jugements pour mesurer la similarité entre les utilisateurs dans les systèmes de filtrage collaboratif. Nous proposons une mesure qui prend en compte les biais items (liés à la popularité) et les biais utilisateur (propension de l'utilisateur à noter de manière positive ou négative). Plus précisément, nous mesurons une *surprise* associée aux accords entre deux utilisateurs qui correspond à l'écart entre le nombre d'accords observés et l'espérance du nombre d'accords correspondant à différents modèles utilisateur et item. La validité de cette mesure est évaluée sur deux tâches (recommandation et prédiction de lien) et permet de distinguer trois types de relations entre utilisateurs.

Sommaire

Introduction

Le chapitre précédent nous a permis de mettre en évidence une différence de sémantique selon qu'un jugement identique entre deux utilisateurs est positif ou négatif. Ainsi, les accords qui concernent des jugements positifs sont davantage synonymes de similarité de goût (les utilisateurs aiment les mêmes items) que les jugements négatifs communs. Ceci s'avère particulièrement intéressant alors même que les principaux modèles exploitant les voisinages se fondent sur l'idée que la similarité est liée aux seuls accords, comme pour l'indice de Pearson ou le cosinus [Bellogín, 2013]. Dans ce chapitre, nous poursuivons l'analyse des accords en étudiant à quel point ces jugements communs sont le résultat du hasard ou non.

Les indicateurs du chapitre précédent prennent en compte le fait que le partage de jugements positifs est un signal fort de similarité entre les utilisateurs ; plus les accords positifs sont nombreux et plus cette similarité est élevée. Pourtant, si deux utilisateurs n'expriment que des jugements positifs sur des items populaires, la probabilité de partager *a priori* des accords positifs est forte, alors même que ces utilisateurs peuvent ne pas être similaires. Dans ce chapitre, nous posons l'hypothèse que pour mieux mesurer la similarité de deux utilisateurs, il est nécessaire de prendre en compte, pour chaque item sur lequel ils sont en accord, à la fois leur propension à noter positivement (biais utilisateur) et la probabilité que ledit item soit jugé positivement par un utilisateur (biais item). Comme nous en avons discuté dans le chapitre précédent, le biais utilisateur est lié au coût que représente implicitement l'évaluation d'un item, et le biais item est lié à sa popularité (les utilisateurs ont tendance a privilégier les items déjà populaires et largement évalués, et ce, indépendamment des coûts d'évaluation [Anderson, 2006]).

C'est dans cette optique que nous définissons, dans ce chapitre, de nouveaux indicateurs de manière à affiner ceux définis dans le chapitre précédent. Il s'agit de prendre en compte le biais utilisateur et le biais item afin de sélectionner les accords qui sont synonymes de liens forts. Plus précisément, dans ce chapitre nous proposons une mesure de similarité qui permet d'évaluer dans quelle mesure des accords sont informatifs ou non, de manière à identifier différents types de relations entre utilisateurs.

Nos contributions sont les suivantes :

— Nous proposons deux mesures qui indiquent, pour les accords et les désaccords, dans quelle proportion les accords sont dus au hasard ou à une véritable concordance des jugements ;

— Nous validons nos mesures en montrant la relation entre celles-ci et (1) la différence des notes données par 2 utilisateurs (Yahoo ! KDD Cup 2011) (2) la polarité de la relation entre utilisateurs (Epinions) ;

— Nous montrons l'existence de trois types de relations entre utilisateurs : similarité forte (ils partagent des jugements positifs communs), similarité faible (ils partagent des jugements communs mais principalement des jugements négatifs) et dissimilarité (leurs jugements communs sont dus au hasard).

4.1 L'intuition

Le précédent chapitre a permis de montrer que le signe d'un accord influe sur la concordance de notation entre les utilisateurs. Ainsi, pour étudier les similarités, il faut s'intéresser aux nombres d'accords positifs, d'accords négatifs et de désaccords. Comme nous avons pu le voir dans le premier chapitre, en particulier dans la section 1.3, il existe deux principaux biais au sein des corpus de recommandation : le biais positivité et le biais popularité. Ces biais sont facilement identifiables : nous pouvons observer sur de nombreux corpus (notamment via la figure 3.1) que les utilisateurs expriment généralement des jugements positifs (en particulier les nouveaux utilisateurs) et que les produits rares ou au contraire très populaires ont également des notes supérieures à la moyenne.

Dans ce chapitre, nous comparons les jugements communs observés entre deux utilisateurs (accords positifs et négatifs, désaccords) avec ceux que nous obtiendrions en considérant un modèle utilisateur dans lequel une note est déterminée par deux facteurs indépendants : l'utilisateur et l'item. Nous étudierons l'écart entre les accords positifs et négatifs observés et ceux attendus par ces modèles probabilistes de l'utilisateur. Par exemple, pour un accord positif sur un item donné, nous évaluons d'une part la probabilité que deux utilisateurs aient attribué une note positive sachant leur propension à juger positivement et, d'autre part, la probabilité d'observer cette même note positive sachant les notes reçues par l'item en question. Cela nous permet de différencier trois types de relations entre utilisateurs.

4.1.1 Probabilité d'occurrence et son impact sur la sémantique d'un jugement

Afin de mieux comprendre l'impact que les biais peuvent avoir sur la sémantique des différents accords, nous pouvons considérer l'exemple du corpus Epinions utilisé dans le chapitre précédent. Sur la figure 4.1 nous observons la répartition des notes selon leur valeur (entre 0 et 5). Sur ces données, les notes attribuées par les utilisateurs sont particulièrement élevées : 85% des notes sont supérieures ou égales à 4, la moyenne du corpus. Ainsi, le fait même que les notes négatives n'aient pas la même probabilité d'apparaître donne des sémantiques différentes aux accords positifs et aux accords négatifs.

Reprenons la répartition ci-dessus en considérant que les notes positives sont celles qui sont supérieures ou égales à 4 (les autres notes sont négatives) et étudions le cas de deux utilisateurs ayant noté un même item. En considérant que les individus notent de manière indépendante et non sous l'influence de groupes ou de voisins, nous avons alors :

— 72.25 % de chance que les deux utilisateurs aiment l'item (accord positif)

— 25.50 % de chance pour que l'un ait aimé et l'autre non (désaccord)

— 2.25 % de chance que les deux utilisateurs n'aiment pas l'item (accord négatif)

Les accords négatifs ont ainsi une probabilité d'occurrence bien moindre.

Ces différentes probabilités confirment l'intérêt de notre premier chapitre de différencier la polarité des accords. Nous pouvons illustrer cela à travers le cas de deux utilisateurs de ce

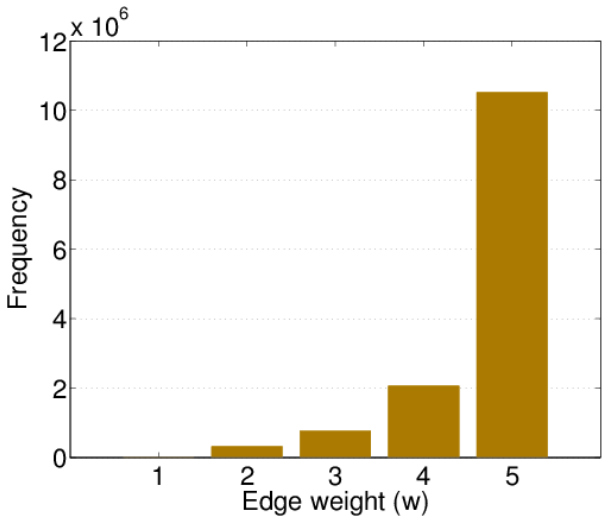


Figure 4.1 – *La distribution de notes attribuées par les utilisateurs dans le corpus Epinions [Massa and Avesani, 2006].*

corpus Epinions qui partageraient 75 accords négatifs et 5 accords positifs sur 100 items en commun :

— les utilisateurs partagent la même opinion négative sur 75 items, alors que cela devrait en concerner tout au plus 3 pour deux utilisateurs pris au hasard partageant 100 items.

— les utilisateurs ne partagent à l'inverse la même opinion positive que sur 5 items, alors que cela devrait concerner environ 72 items pour deux utilisateurs pris au hasard.

Si le premier élément peut être vu comme un indicateur fort de similarité (les utilisateurs partagent la même opinion), le second indicateur tempère néanmoins ce premier point puisque le nombre d'accords positifs est anormalement faible (5 accords positifs observés contre environ 72 attendus). En fait, la sur-représentation des notes positives fait que nous devrions observer davantage d'accords positifs que négatifs pour des utilisateurs similaires.

Dans ce chapitre, nous étudions les situations où les probabilités d'occurrence d'une note dépendent à la fois de chaque utilisateur et de chaque item. Nous mesurons une *surprise* positive (respectivement négative) mesurant l'écart entre les accords positifs (resp. négatifs) observés et ceux espérés par le modèle utilisateur, en supposant qu'une note est déterminée indépendamment par l'utilisateur et par l'item. Nous notons S^+ (respectivement S^-) le score positif (resp. négatif) associé à la surprise apportée par l'observation d'accords positifs (resp. négatifs). Pour chaque type d'accord étudié, nous pouvons nous attendre à trois situations différentes :

— ils sont plus nombreux que prévu : le score associé sera **positif**

— ils sont moins nombreux que prévu : le score associé sera **négatif**

— il y en a autant que prévu : le score associé sera **nul**

Nous allons définir formellement la notion de surprise conditionnellement à un ensemble de jugements communs, c'est-à-dire en considérant fixé le sous-ensemble d'items correspondants. Ce cadre simplificateur nous permet de définir des mesures simples de surprise. À

l'aide de quelques exemples, nous allons examiner les propriétés désirées de telles mesures. Nous formalisons cette notion de surprise dans la section 4.2.

Nous étudions le cas de deux utilisateurs ayant jugé 10 produits chacun et qui partagent 5 jugements communs.

Une surprise nulle

Dans le premier exemple, les deux utilisateurs ont jugé uniquement de manière positive, sur l'ensemble de leurs produits :

$$\begin{array}{lcccccccccccccc} u_1 : & + & + & + & + & + & + & + & + & + & + & & & & & \\ u_2 : & & & & & & + & + & + & + & + & + & + & + & + & + \end{array}$$

Nous nous attendons à ce que la surprise S^+ soit nulle, car quels qu'aient été les items communs considérés, le nombre d'accords positifs aurait été le même puisque les utilisateurs notent uniquement de manière positive.

Les cas moins triviaux dans lesquels la surprise devrait être nulle seront ceux où la proportion de notes positives/négatives de chaque utilisateur est identique sur l'ensemble de leurs notes et l'ensemble des notes communes. Par exemple :

$$\begin{array}{lcccccccccccccc} u_1 : & + & + & - & + & + & + & + & - & + & + & & & & & \\ u_2 : & & & & & & + & + & + & + & + & + & + & + & + & + \end{array}$$

La surprise S^+ devrait être nulle puisque l'utilisateur u_1 à 1 chance sur 5 de juger un item négativement, ce que nous observons sur l'ensemble des notes communes.

Une surprise positive

Dans notre second exemple, les utilisateurs ont cette fois attribué des jugements positifs et négatifs, et l'ensemble des items jugés en commun correspond exactement à l'ensemble des items appréciés par l'un et l'autre :

$$\begin{array}{lcccccccccccccc} u_1 : & - & - & - & - & - & + & + & + & + & + & & & & & \\ u_2 : & & & & & & + & + & + & + & + & - & - & - & - & - \end{array}$$

Nous considérons que cet exemple est très informatif, et illustre précisément ce que l'on attend de personnes qui noteraient de la même manière. Nous attendons une surprise S^+ positive associée aux accords positifs puisque le nombre d'accords positifs attendu sachant les distributions de notes de chaque utilisateur est de 2 ou 3 sur les 5 (sachant que chaque utilisateur a une proportion de notes positives/négatives de 50%).

Une surprise négative

Pour le dernier cas, les utilisateurs sont en désaccord sur l'ensemble de leurs items communs.

u_1 : $-\ -\ -\ -\ -\ +\ +\ +\ +\ +$
u_2 : $-\ -\ -\ -\ -\ +\ +\ +\ +\ +$

Ce cas est également très informatif, car les utilisateurs ne notent pas de la même manière. Le score positif S^+ associé à cette situation devrait être négatif.

Ces exemples soulignent l'importance de juger des accords en fonction de leur contexte, i.e. en fonction de la propension des utilisateurs à juger de telle ou telle manière. Tout écart devient alors une source d'information pour notre étude des similarités. De façon duale, nous pouvons aussi nous intéresser aux items. En effet, la popularité des items influe énormément sur le jugement des utilisateurs. Par exemple, un accord sur un item dont les notes seraient très majoritairement positives a toutes les chances d'être positif lui aussi. Nous définissons dans la suite un score de *surprise* qui prend en compte les biais utilisateur et item.

4.2 La surprise positive et la surprise négative

Dans cette section, nous proposons de nouvelles mesures de similarité entre utilisateurs qui prennent en compte la propension de l'utilisateur à juger positivement ainsi que la popularité de l'item. Le principe du score (mesurant une surprise) est de comparer pour deux utilisateurs donnés l'ensemble des accords observés avec leur nombre espéré suivant des modèles utilisateur/item que nous définissons par la suite.

4.2.1 Calcul du score

Nous définissons tout d'abord une variable appelée *gain item positif* g_i^+ (respectivement *négatif* g_i^-) dont la valeur mesure l'importance d'un accord positif (resp. négatif) sur un item donné. Les scores proposés utiliseront soit un gain constant (égal à 1) soit un gain dépendant des notes reçues par l'item. Dans ce dernier cas, un accord positif (resp. négatif) sur un item ayant majoritairement reçu des notes négatives (resp. positives) sera associé à un gain fort. À l'inverse, un accords positif (resp. négatif) sur un item ayant reçu majoritairement des notes positives (resp. négatives) sera associé à un gain faible.

Ce gain par item nous permet de définir un gain global $G^s(u, v)$ associé aux accords de type $s \in \{+, -\}$ pour un couple d'utilisateurs u et v partageant des jugements sur un ensemble I_{uv} d'items. Soit X_{ui}^s une variable aléatoire binaire qui indique si l'utilisateur u a noté l'item i avec une note de signe s ($X_{ui}^s = 1$) ou non ($X_{ui}^s = 0$). Un accord est alors observé entre u et v si $X_{ui}^s X_{vi}^s$ est égal à 1. De manière plus formelle, le gain global $G^s(u, v)$ pour les accords de type s entre les utilisateurs u et v est défini comme la variable aléatoire :

$$G^s(u, v) = \sum_{i \in I_{uv}} g_i^s X_{ui}^s X_{vi}^s \tag{4.1}$$

La loi de X^s_{ui} est définie par un modèle utilisateur. Nous en proposons quatre dans la suite.

Le score de surprise que nous proposons est défini comme la différence entre le gain observé $G^s_o(u,v)$ et un gain théorique $\mathbb{E}(G^s(u,v))$ où l'espérance est prise par rapport aux lois des variables X^s_{ui} et X^s_{vi}. Le gain observé $G^s_o(u,v)$ correspond à la somme des gains items sur lesquels les utilisateurs sont en accord de type s. L'idée est que si ces utilisateurs partagent un certain nombre de notes communes, le gain observé de signe s est important si, toutes choses égales par ailleurs : (1) les utilisateurs partagent beaucoup d'accords de signe s (2) ils sont en accord de signe s sur des items au gain s élevé.

Le gain observé $G^s_o(u,v)$ est calculé de la manière suivante :

$$G^s_o(u,v) = \sum_{i \in I^s_{uv}} g^s_i \tag{4.2}$$

avec I^s_{uv} l'ensemble des items communs pour lesquels les utilisateurs u et v ont tous deux donné une note de polarité s.

L'espérance de gain $\mathbb{E}(G^s(u,v))$ - correspondant au gain attendu (lorsque l'on n'observe pas les notes des utilisateurs) - est définie à partir d'un modèle utilisateur. Si nous supposons que les notes données par deux utilisateurs sont indépendantes entre elles, l'espérance se calcule de la manière suivante :

$$\begin{aligned} \mathbb{E}(G^s(u,v)) &= \sum_{i \in I_{uv}} g^s_i \times P(X^s_{ui}=1) \times P(X^s_{vi}=1) && (4.3)\\ &= \sum_{i \in I_{uv}} g^s_i \times P(s|u,i) \times P(s|v,i) && (4.4) \end{aligned}$$

Le gain observé et le gain espéré nous permettent de calculer le score de surprise. Dans tous nos modèles, nous faisons l'hypothèse que deux utilisateurs u et v notent de façon indépendante l'un de l'autre [2]. Ainsi, plus la différence est grande entre le gain observé et le gain espéré, plus forte est la certitude que les deux utilisateurs ont un accord de polarité s qui n'est pas dû au hasard. Ce score pouvant dépendre du nombre de jugements en commun, il est intéressant de considérer la version normalisée par la valeur maximale $G^s_{max}(u,v) = \sum_{I_{uv}} g^s_i$ qui peut être atteinte par le gain :

$$S^s_{uv} = \begin{cases} 0 & \text{si } G^s_{max} = 0 \\ \frac{G^s_o(u,v) - \mathbb{E}(G^s_{uv})}{G^s_{max}(u,v)} & \text{si } G^s_{max} \neq 0 \\ 1 & \text{sinon} \end{cases} \tag{4.5}$$

Si S est proche de 1, cela signifie que les accords du type s sont beaucoup plus présents que prévu. Si S est inférieur à zéro, cela signale un nombre d'accords de type s moins important

2. par ailleurs, les trois premiers modèles utilisateur font l'hypothèse que les notes d'un utilisateur sont indépendantes entre elles. Le quatrième modèle ne fait pas cette hypothèse d'indépendance.

que prévu. Enfin, lorsque S est égal (ou proche) à zéro, les accords entre les deux utilisateurs peuvent être le résultat du hasard.

Le calcul de l'espérance est le point clé puisqu'il dépend des différentes hypothèses faites sur les modèles utilisateur et item. Nous proposons ici quatre modèles pour définir notre espérance. Pour les trois premiers modèles, la popularité des items est prise en compte dans le calcul de l'espérance et le gain item fixé à 1 pour l'ensemble des items. Pour le quatrième modèle, le calcul de l'espérance prend en compte uniquement les modèles utilisateur, le gain item est alors une fonction de la popularité des items.

4.2.2 Modèle 1 : Popularité item (S_{po})

Dans ce premier modèle, nous supposons qu'un note est uniquement expliquée par l'item considéré et que $g_i^s = 1$. Autrement dit, nous avons $P(s|u,i) = P(s|i)$. Le calcul de l'espérance peut alors s'écrire de la manière suivante :

$$\mathbb{E}(G_{uv}^s) = \sum_{i \in I_{uv}} P(s|i) \times P(s|i) \tag{4.6}$$

Afin d'éviter le problème d'estimation de la probabilité $P(s|i)$ lorsque le nombre de notes reçues par l'item i est faible, nous utilisons un *a priori* bêta qui est la distribution conjuguée de la loi de Bernouilli.

Les paramètres de la loi bêta (α, β) sont estimés en considérant l'ensemble des notes. Nous voulons en effet qu'en l'absence de toute observation, la probabilité $P(+|i)$ soit égale à la probabilité $P(+)$ de noter positivement un item :

$$P(+|i) = \frac{\alpha}{\alpha + \beta} = \frac{\#\text{notes '+'}}{\#\text{notes}} = \frac{\sum_i n_i^+}{\sum_i n_i^+ + n_i^-} = r \tag{4.7}$$

où n_i^s est le nombre de notes de signe s attribuées à l'item i.

Ensuite, nous fixons nos paramètres α et β à l'aide d'un autre paramètre K, permettant de contrôler le poids des observations sur l'item par rapport aux observations sur chaque item. Soit :

$$\alpha = K \times r \text{ et } \beta = K \times (1 - r) \tag{4.8}$$

où K correspond à un nombre d'*observations* par défaut de chaque item.

Lorsque nous avons observé n_i^+ notes positives et n_i^- notes négatives pour l'item i, la probabilité *a posteriori* d'observer une nouvelle note positive est :

$$P(+|i, n_i^+, n_i^-) = \frac{\alpha + n_i^+}{\alpha + \beta + n_i^+ + n_i^-} \tag{4.9}$$

et symétriquement pour une note négative,

$$P(-|i, n_i^+, n_i^-) = \frac{\beta + n_i^-}{\alpha + \beta + n_i^+ + n_i^-} \tag{4.10}$$

4.2.3 Modèle 2 : Preuve sociale (S_{so})

La preuve sociale est un principe sociologique selon lequel les utilisateurs ont tendance à adopter le comportement des autres [Cialdini, 2012]. L'idée de ce modèle est de prendre en compte si un utilisateur note un item de la même manière que les autres ou si, au contraire, l'utilisateur note différemment. Nous supposons également pour ce modèle que la note de l'utilisateur dépend uniquement de l'item.

Formellement, un utilisateur donne une note de signe s si (1) il adopte le comportement des autres utilisateurs qui ont donné une note de signe s ou (2) il n'adopte pas le comportement des autres utilisateurs qui ont donné une note de signe $\neg s$. La variable aléatoire associée au cas où l'utilisateur adopte la note s des utilisateurs est notée a_s et $\neg a_s$ dans le cas contraire. La probabilité d'une note est donnée par :

$$P(s|u,i) = P(a_s|u) \times P(s|i) + P(\neg a_s|u) \times P(\neg s|i) \tag{4.11}$$

La probabilité $P(s|i)$ est estimée de la même manière que pour le précédent modèle. Nous utilisons le même *a priori* bêta pour le calcul de la probabilité $P(a_s|u)$. Par rapport au modèle 1, nous considérons ici la propension de l'utilisateur à se comporter comme la majorité des autres utilisateurs. En particulier, nous avons :

$$P(a_s|u) = \frac{\alpha + n_a^s}{\alpha + \beta + n_a^s + n_{\neg a^s}} \tag{4.12}$$

avec n_a^s le nombre d'utilisateurs qui sont en accord de type s avec l'utilisateur u et $n_{\neg a^s}$ le nombre d'utilisateurs qui sont en désaccord avec l'utilisateur u (somme sur tous les items).

4.2.4 Modèle 3 : Principe de cohérence (S_{co})

Dans ce modèle, nous supposons que la probabilité qu'un utilisateur u donne une note de signe s à un item particulier dépend à la fois de sa propension à noter s et de la distribution de notes reçues par cet item. Cette hypothèse suit le principe de cohérence selon lequel les notes respectent indépendamment le comportement de l'utilisateur et les notes des items. Autrement dit, nous avons :

$$P(s|u,i) \propto P(s|u) \times P(s|i) \tag{4.13}$$

Ces probabilités sont estimées comme précédemment (loi de Bernouilli).

4.2.5 Modèle 4 : Permutation aléatoire (S_{pa})

À la différence des précédents modèles, nous considérons ici que les notes données par un utilisateur ne sont pas indépendantes entre elles : nous supposons que le nombre exact de notes positives et négatives observées pour chaque utilisateur est important et doit être respecté. Pour illustrer cette importance, prenons l'exemple suivant :

u_1 : + + − + + + + − + +
u_2 : + + + + + + + + + +

La probabilité pour que les utilisateurs soient en désaccord est de 0.2, compte tenu des distributions de notes de chacun d'eux. En supposant l'indépendance des notes entre elles, nous aurions une probabilité $0.2 \times 0.2 \times 0.2$ d'avoir trois désaccords. Dans ce modèle, nous supposons que le nombre maximal de désaccords est de deux (sachant que u_1 n'a que deux notes négatives) et donc que la probabilité que u_1 et u_2 aient trois désaccords est nulle.

Afin d'écrire notre modèle de manière formelle, nous définissons V_u comme l'ensemble des utilisateurs [3] aux caractéristiques équivalentes à celles de u, c'est-à-dire qui ont le même nombre de notes positives et négatives sur l'ensemble des items I_u jugés par u :

$$V_u = \{\tilde{u} | (I_u = I_{\tilde{u}}) \wedge (n_u^+ = n_{\tilde{u}}^+) \wedge (n_u^- = n_{\tilde{u}}^-)\} \tag{4.14}$$

où n_u^s est le nombre de jugements de polarité s.

Cela revient à considérer un ensemble de jugements qui seraient générés par une permutation des jugements de l'utilisateur u parmi les items qu'il a jugés. Nous supposons ensuite que la probabilité de juger avec une certaine polarité l'ensemble des items I_u jugés par l'utilisateur u est uniforme, c'est-à-dire que n'importe quelle permutation a la même probabilité $1/|V_u|$.

Ceci nous permet de calculer l'espérance du gain pour le couple (u, v) comme le gain espéré pour un couple d'utilisateurs $(\tilde{u}, \tilde{v}) \in (V_u \times V_v)$. Cette espérance est définie de la manière suivante :

$$\begin{aligned} \mathbb{E}(G^s(u,v)) &= \sum_{\tilde{u} \in V_u} \sum_{\tilde{v} \in V_v} \sum_{i \in I^s_{\tilde{u}\tilde{v}}} p(\tilde{u}) p(\tilde{v}) g_i^s \\ &= \frac{1}{|V_u||V_v|} \sum_{\tilde{u} \in V_u} \sum_{\tilde{v} \in V_v} \sum_{i \in I^s_{\tilde{u}\tilde{v}}} g_i^s \end{aligned}$$

Le calcul de cette espérance est donné en annexe. Nous y utilisons un modèle combinatoire, et ce, de manière à calculer un nombre de partitions (via le multinôme de Newton) respectant les distributions de chaque utilisateur. Plus précisément, cela permet de calculer le nombre d'accords de polarité s :

$$\sum_{\tilde{u} \in V_u} \sum_{\tilde{v} \in V_v} |I_u^s \cap I_{\tilde{u}}^s| = \sum_{l=0}^{n} \binom{n}{l, n_u^s - l, n_v^s - l, n + l - n_u^s - n_v^s} \times \frac{\binom{n-1}{k-1}}{\binom{n}{k}} \tag{4.15}$$

avec $n = |I_u \cap I_v|$ le nombre de jugements communs. Soit au final :

$$\mathbb{E}(G_{uv}^s) = \frac{\sum_{l=0}^{n} \frac{(n-1)!}{(l-1)!(n_u^s - l)!(n_v^s - l)!(n+l-n_u^s-n_v^s)!(n-l+1)} \times \sum_{i \in I_{uv}} g_i^s}{\binom{n}{n_u^s} \times \binom{n}{n_v^s}} \tag{4.16}$$

3. ces utilisateurs ne correspondent pas nécessairement à des utilisateurs observés

Définition du gain item

Ce modèle ne tient pas compte de la popularité d'un item. Afin de pallier cela, nous définissons le gain associé à un accord de manière différente que dans les modèles précédents : pour un accord de polarité $s \in \{+, -\}$, le gain pour un item est égal à la probabilité de *ne pas observer* d'accord de type s pour cet item. Il est donc d'autant plus fort que l'item a reçu peu de notes de polarité s. En supposant que les deux utilisateurs notent de façon indépendante, le gain est donc :

$$g_i^s = 1 - p(s|i)p(s|i) \tag{4.17}$$

$$= 1 - \left(\frac{n_i^s}{n_i}\right)^2 \tag{4.18}$$

où u et v sont deux utilisateurs quelconques, u_i et v_i les notes qu'ils ont attribuées à l'item i, n_i^s le nombre de notes de signe s reçues par i et n_i le nombre total de notes reçues. On fait l'hypothèse que la probabilité $p(u_i = s) = \frac{n_i^s}{n_i}$ est uniforme et ne dépend que de l'item i.

4.3 Les expériences

Afin de mesurer la pertinence de nos mesures, nous les avons testées sur deux corpus, Yahoo ! KDD Cup 2011 [Dror et al., 2012] et Epinions [Massa and Avesani, 2006] présentés dans le chapitre précédent (section 3.3.1).

Nous avons exploité le corpus Yahoo ! KDD Cup 2011 pour évaluer la corrélation entre nos différents scores et la concordance des jugements. Nous avons usé de la même transformation que dans le chapitre précédent pour prendre en compte la polarité des notes, à savoir :

$$\textit{polarité} = \begin{cases} +1 & \text{si note} > 70 \\ -1 & \text{si note} < 30 \end{cases}$$

Nous exploitons le corpus Epinions pour mesurer la corrélation entre nos scores et la polarité des liens sociaux. Ce corpus contient en effet un graphe social signé dirigé dans lequel les utilisateurs ont indiqué leur confiance (+1) ou leur méfiance (-1) envers les autres utilisateurs. La transformation des notes correspond exactement à celle de chapitre précédent :

$$\textit{polarité} = \begin{cases} +1 & \text{si note} > 4 \\ -1 & \text{si note} < 4 \end{cases}$$

4.3.1 Mesure de la similarité

Afin d'évaluer la pertinence de nos caractéristiques nous avons choisi de nous intéresser à deux tâches : la recommandation et la prédiction de liens. Nous commençons par comparer nos quatre modèles en terme de performance en recommandation afin d'identifier les hypothèses (et donc le modèle) les plus prédictives pour évaluer la similarité entre deux utilisateurs.

Ensuite, nous étudions plus en détail le meilleur modèle sur les deux tâches de recommandation et de prédiction de liens en le comparant au modèle du chapitre précédent.

Pour comparer deux utilisateurs sur la tâche de recommandation, nous avons calculé pour chaque couple d'utilisateurs la moyenne des différences quadratiques entre leurs notes communes. Une valeur élevée signifiera par exemple que, en moyenne, les utilisateurs n'ont pas noté de la même façon. Formellement, cela nous donne pour chaque couple (u, v) la différence suivante :

$$RMSD_{uv} = \sqrt{\frac{\sum_{i \in I_{uv}} (r_{ui} - r_{vi})^2}{|I_u \cap I_v|}}$$

où r_{ui} est la note attribuée par l'utilisateur u à l'item i.

Pour la tâche de prédiction de liens, nous avons utilisé les données du corpus Epinions parmi lesquelles on trouve un graphe de confiance où les utilisateurs sont reliés entre eux par des liens positifs ou négatifs. Nous avons alors mesuré la proportion de liens négatifs observés en fonction de la valeur de nos scores.

4.3.2 Sélection des voisins

Pour le corpus Epinions, nous pouvons utiliser le graphe social pour sélectionner nos couples d'utilisateurs. En revanche, pour le corpus Yahoo ! KDD Cup 2011, nous devons sélectionner les couples parmi les utilisateurs partageant des items en communs.

Afin de limiter les calculs, nous avons respecté le même protocole que celui suivi dans le chapitre précédent, à savoir que pour chaque utilisateur u, nous sélectionnons 100 voisins, parmi lesquels nous avons : (1) les 50 utilisateurs partageant le plus d'items en commun avec u et (2) 50 utilisateurs tirés aléatoirement dans l'ensemble de ceux ayant au moins deux items communs avec l'utilisateur.

4.3.3 Comparaison des approches

Les quatre approches que nous proposons traduisent quatre hypothèses différentes pour les modèles utilisateur. Nous les comparons en terme de performance en prédiction de notes, et ce, à l'aide du test statistique d'analyse de la variance (ou ANOVA, *analysis of variance*). L'objectif de ce test consiste à évaluer à quel point des variables explicatives données (ou facteurs) influencent la distribution d'une variable continue à expliquer. Il s'agit dans notre cas de comparer les comportements de notation des utilisateurs (via la $RMSD$) en fonction du score positif S^+, du score négatif S^- et, afin d'étudier les interactions entre les variables, de la combinaison des deux $S^+ \times S^-$. Le modèle dont nous analysons la variance est une fonction linéaire des variables explicatives :

$$RMSD(u, v) = \alpha S^+(u, v) + \beta S^-(u, v) + \gamma S^+(u, v) S^-(u, v) + \mu \tag{4.19}$$

Les résultats sont donnés dans le tableau 4.1. Le coefficient de détermination R^2 mesure la variance expliquée par les facteurs. Sa valeur maximale 1 est obtenue si les variables prédisent

	R^2	Coefficients de la régression linéaire		
		S^+	S^-	$S^+ \times S^-$
Popularité item (S_{po})	0.19	-16.4	-3.89	27.00
Preuve sociale (S_{so})	0.23	14.72	-31.2	100.0
Principe de cohérence (S_{co})	0.44	16.30	-2.05	-3.08
Permutation aléatoire (S_{pa})	0.65	-32.18	-5.13	-63.5
Taux d'accords (P)	0.89	-74.2	-57.9	34.68

Table *4.1* – *Résultats du test statistique d'analyse de la variance de la RMSD entre les utilisateurs en fonction des scores de surprise positifs et négatifs. Corpus Yahoo !*

parfaitement le RMSD. Les coefficients indiqués dans le tableau sont ceux de la fonction 4.19 analysée. Ainsi, un facteur au coefficient positif a un impact négatif sur le RMSD (il l'augmente). Puis, lorsque le facteur $S^+ \times S^-$ a un coefficient positif, cela signifie que le RMSD augmente quand les deux scores S^+ et S^- sont de même signe.

Pour les trois premiers modèles, le facteur K de la loi *a priori* bêta est fixé à 1 (les différentes expériences ont montré que les performances ne variaient pas énormément en fonction de la valeur de ce K). Nous analysons dans la suite les résultats pour nos quatre modèles. Nous analysons également les résultats obtenus avec les scores ϕ_2 et ϕ_3 définis dans le précédent chapitre (notés ici P^+ et P^-).

Dans tous les cas, les trois facteurs (S^+, S^- et $S^+ \times S^-$) sont significatifs pour expliquer la différence de notation, ce qui n'est pas reporté dans le tableau 4.1 afin de le simplifier.

Popularité item (S_{po}) : Nous constatons que le modèle basé sur la popularité de l'item obtient les moins bons résultats, avec un coefficient R^2 de 0.19. Deux utilisateurs partageant des notes positives ou négatives (dans une moindre mesures) ont une notation plus similaire. En revanche, lorsqu'ils ont un S^+_{po} et un S^-_{po} de même signe, alors ils le sont moins. Pour ce modèle, les utilisateurs similaires sont alors ceux qui sont toujours d'accord de la même façon (accords de même signe).

Preuve sociale (S_{so}) : Ce modèle donne des résultats sensiblement similaires à ceux du modèle précédent puisque le R^2 vaut 0.23 (contre 0.19). Deux utilisateurs qui partagent davantage d'accords négatifs ($S^-_{so} > 0$) sont plus similaires que ceux partageant plus d'accords positifs ($S^+_{so} > 0$) ou ceux partageant davantage des deux ($S^+_{so} \times S^-_{so} > 0$). Lorsque les utilisateurs notent en fonction du comportement des autres, les utilisateurs à la notation la plus similaire sont ceux dont les jugements négatifs ne correspondent pas à leur comportement habituel.

Principe de cohérence (S_{co}) : Dans ce modèle où les notes sont déterminées indépendemment par l'utilisateur et par l'item, les résultats obtenus sont encore meilleurs, avec un coefficient R^2 à 0.44. Dans ce modèle, les utilisateurs qui notent de la même manière sont

ceux qui partagent plus d'accords négatifs ($S^-_{co} > 0$) ou dont les scores S^+_{co} et S^-_{co} sont de même signe. Les accords positifs (seuls) en plus grand nombre ne sont donc pas synonymes de similarité.

Permutations aléatoire (S_{pa}) : Nous observons que les résultats sont un peu meilleurs que le modèle précédent puisque le R^2 vaut 0.65. Ici, les utilisateurs qui partagent plus d'accords positifs et négatifs sont davantage similaires. Mais, ceux partageant uniquement des accords négatifs le seront dans une moindre mesure (S^-_{pa}). Les utilisateurs similaires sont avant tout ceux qui aiment les mêmes produits ($S^+_{pa} > 0$ et $S^+_{pa} \times S^-_{pa} > 0$).

Concernant les scores P^+ et P^- du précédent chapitre, ils permettent d'obtenir les meilleurs résultats de ce comparatif avec un R^2 à 0.89. Comme pour le précédent modèle, le score P^+ associé aux accords positifs est davantage synonyme de similarité (coefficient à -74.07) que le score P^- associé aux accords négatifs (-57.10). La combinaison des scores a un coefficient à 35.51 ce qui signifie qu'observer des scores de même signe n'est pas synonyme de similarité.

Pour la suite des expériences, nous nous focalisons sur les scores de surprises S^+ et S^- définis par le quatrième modèle *Permutations aléatoire*, qui sont les scores qui permettent le mieux d'expliquer la différence de notation entre deux utilisateurs.

4.3.4 Résultats

Pour analyser nos résultats, nous allons tout d'abord étudier les scores S^+ et S^- séparément, en les comparant aux indicateurs mis en place dans le chapitre précédent. Ensuite, nous nous intéressons à l'interaction de ces deux scores afin de mettre en évidence différents types de relations entre utilisateur. Nous avons évalué ces mesures grâce à deux échantillons de 6 000 000 de couples d'utilisateurs tirés aléatoirement, à la fois sur Yahoo ! Music et sur Epinions.

Concordance des jugements

Nous étudions tout d'abord nos mesures dans le contexte de la recommandation, à l'aide des figures 4.2 et 4.3. Ces quatre *boxplots* représentent la valeur de la RMSD en fonction de deux mesures de similarité. Le diagramme de gauche donne l'évolution de la RMSD en fonction de la mesure de similarité P^s, à savoir la proportion d'accords de signe s sur l'ensemble des accords. Le diagramme de droite montre l'évolution de la RMSD en fonction des scores S^s_{pa}. Chaque boîte représente un décile de 600 000 couples d'utilisateurs.

Score positif

Les deux *boxplots* de la figure 4.2 nous permettent d'observer dans les deux cas qu'une augmentation de nos mesures P^+ et S^+_{pa} est associée à une diminution de l'écart de notation. Pour S^+_{pa} en particulier, une valeur élevée implique une RMSD plus faible, c'est-à-dire qu'un

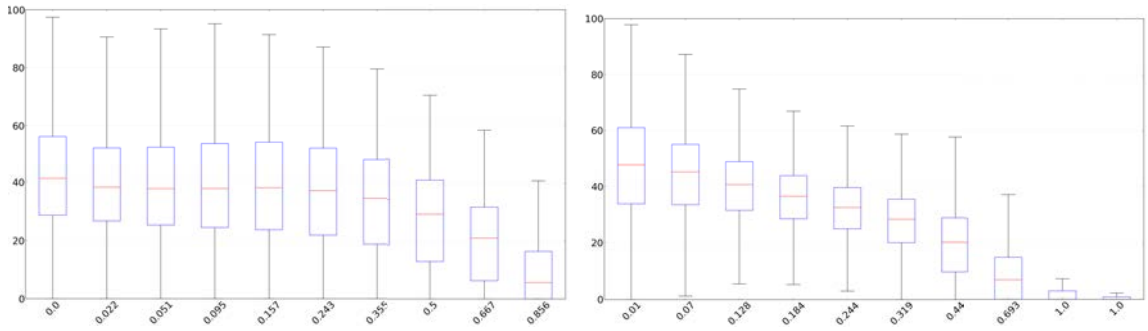


Figure 4.2 – *Évolution du RMSD en fonction de P^+ (à gauche) et S^+_{pa} (à droite). Corpus Yahoo*

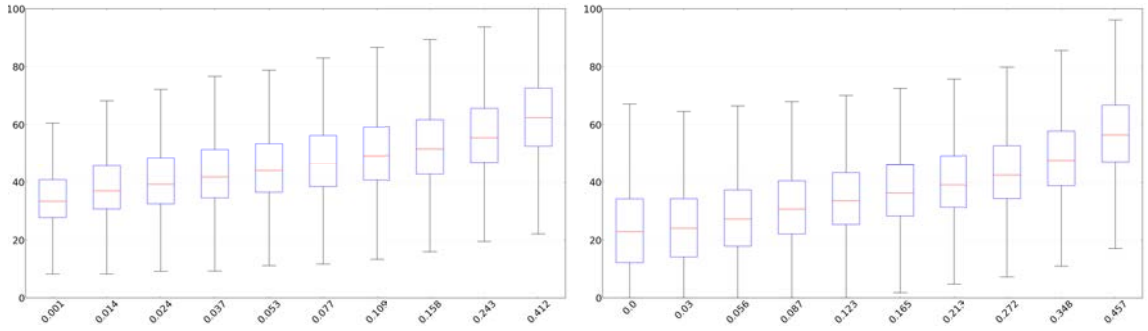


Figure 4.3 – *Évolution de RMSD en fonction de P (à gauche) et S_{pa} (à droite) . Corpus Yahoo*

nombre d'accords positifs supérieur au hasard réduit l'écart de notation entre les utilisateurs. Nous pouvons remarquer que la mesure *accords positifs sur notes communes* se comporte de la même manière, bien que la baisse de la RMSD ne soit marquée qu'à partir d'une valeur plus élevées et que la variance du RMSD soit plus importante, ce qui indique que S^+_{pa} est un meilleur indicateur.

Score négatif

Pour ces mesures sur les accords négatifs, le comportement est le symétrique du cas positif : une augmentation de valeur de P et S_{pa} s'accompagne d'un écart de notation plus important. Par exemple, un nombre d'accords négatifs supérieur aux attentes implique une moindre concordance des notes entre utilisateurs. Autrement dit, les personnes qui partagent plus d'accords négatifs que prévu ne notent pas de la même manière. La caractéristique *accords négatifs sur notes communes* mène à la même conclusion, mais là encore la tendance est moins nette.

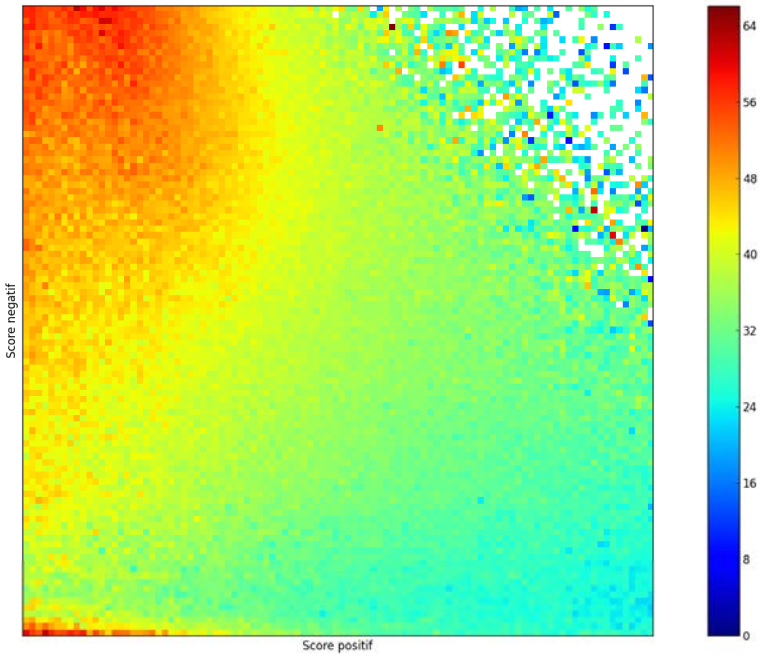


Figure 4.4 – *Heatmap représentant la valeur de RMSD en fonction du score positif S^+_{pa} et du score négatif S_{pa}.*

Interaction scores positifs et négatifs

L'objectif de notre approche est de pouvoir distinguer différents types de relations entre les utilisateurs, avec différents degrés de similarité. Considérés de manière indépendante, les scores positifs et négatifs S^+_{pa} et S_{pa} ne permettent pas de dissocier des groupes disctincts d'utilisateurs. C'est pourquoi nous allons nous intéresser ici à l'interaction entre les deux mesures. Ceci est aussi suggéré par le fort coefficient associé à $S^+_{pa} \times S_{pa}$ dans le tableau 4.1.

Pour cela, nous avons représenté cette interaction grâce à une heat-map (figure 4.4) qui montre le RMSD observé en fonction des scores positifs (abscisse) et négatifs (ordonnée). Nous avons centré les observations sur l'intervalle de scores $[0,1] \times [0,1]$ car la majorité des couples d'utilisateurs se situent dans cette zone. Trois types de relations peuvent être observés depuis cette représentation sur la figure 4.4 : (1) en bas à droite, (2) en haut à gauche et (3) en bas à gauche. Nous définissons ces trois relations de la manière suivante :

$$\begin{cases} \text{Similarité } \textit{forte} : & S^+_{pa} \text{ élevé} & S_{pa} \text{ faible (RMSD faible)} \\ \text{Similarité } \textit{faible} : & S^+_{pa} \text{ faible} & S_{pa} \text{ élevé (RMSD élevé)} \\ \text{Dissimilarité} : & S^+_{pa} \text{ faible} & S_{pa} \text{ faible (RMSD fort)} \end{cases}$$

Les personnes similaires (RMSD faible) se caractérisent par le fait que, sur l'ensemble des jugements communs, ils ont tendance à être plus en accord positif que prévu (S^+_{pa} fort), alors que pour les items jugés négativement, leurs accords sont conformes à ceux de notre modèle aléatoire ($S_{pa} = 0$) : leurs notes positives sont donc *"concentrées"* sur des sous-ensembles spécifiques d'items, alors que ce n'est pas le cas pour les notes négatives.

Les personnes faiblement similaires (RMSD élevé) ont tendance à être plus en accord négativement (S_{pa} fort) que positivement (S^+_{pa} faible). Intuitivement, cela correspond à des utilisateurs qui n'aiment pas les mêmes items, mais qui ont des zones thématiques proches (voir chapitre précédent). Sur la figure 4.4, ces utilisateurs se situent en haut à gauche ; le RMSD est élevé.

Les personnes dissimilaires ne partagent pas d'intérêts particuliers dans la mesure où les seuls accords qu'ils partagent peuvent être expliqués par leur façon de juger - scores positif S^+_{pa} et négatif S_{pa} faibles. Les notes communes sont généralement peu nombreuses et en majorité des désaccords. Sur la figure 4.4, ces utilisateurs sont dans la zone en bas à gauche où le RMSD moyen est élevé.

Le parallèle social

Nous avons également validé les scores proposés sur une tâche de prédiction de lien social sur le corpus Epinions où nous disposons de liens sociaux signés explicites. En particulier, nous cherchons à observer des corrélations entre d'une part les scores positif et négatif, et d'autre part le signe du lien social entre les utilisateurs. Les figures 4.5 et 4.6 montrent la proportion de liens de méfiance en fonction de ces scores. La proportion de liens de confiance peut être obtenue facilement puisque les proportions de liens de méfiance et de liens de confiance somment à 1.

Sur la figure 4.5, nous observons que la proportion de liens de méfiance entre les utilisateurs décroît en fonction du score P^+ (le taux d'accords positifs sur l'ensemble des accords). Ainsi, les utilisateurs qui partagent davantage d'accords positifs sont moins souvent reliés par un lien de méfiance, davantage par un lien de confiance. Concernant le score S^+_{pa}, une valeur proche de zéro implique un taux plus important de liens de méfiance parmi les utilisateurs. À l'inverse, les utilisateurs qui ont une surprise S^+_{pa} élevée, autrement dit les utilisateurs qui ont plus d'accords positifs que prévu, semblent être moins relié par des liens de méfiance. En revanche, lorsque le S^+_{pa} est inférieur à zéro, la proportion de liens de méfiance reste élevé mais se tasse quand le score S^+_{pa} diminue (cela concerne des utilisateurs qui ont généralement un S_{pa} plus important et un taux de désaccord plus faible).

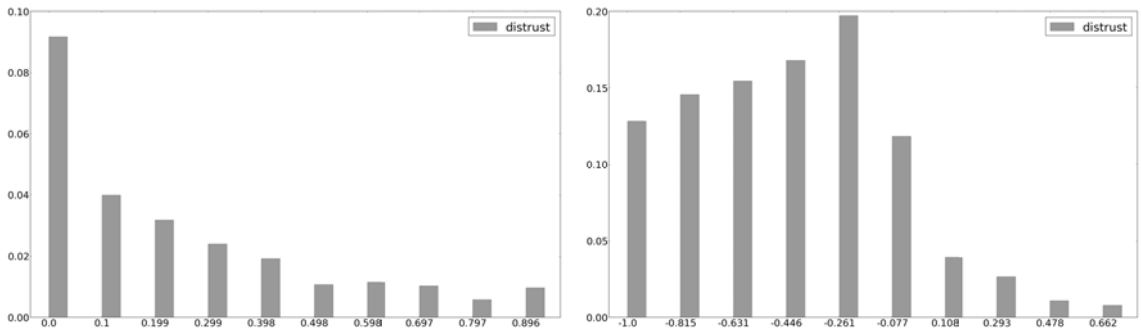


Figure 4.5 – *Répartition des liens de méfiance en fonction de P^+ (à gauche) et de S^+_{pa} (à droite). Corpus Epinions*

Les mesures associées aux accords négatifs semblent, elles aussi, particulièrement bien adaptées. Sur la figure 4.6, nous pouvons voir que le taux de liens de méfiance croît en fonction du score P . Les accords négatifs sont alors davantage synonymes de méfiance que les accords positifs. Nous observons également qu'un S_{pa} proche de zéro implique une proportion de liens sociaux négatifs faible (inférieur à 10%). Nous observons ensuite que la proportion de liens de méfiance augmente lorsque la magnitude du score négatif S_{pa} augmente, ce qui indique que toute déviation par rapport à ce qui est attendu au niveau des accords négatifs entraîne une plus grande probabilité d'avoir un lien négatif.

En conclusion, nos scores S^{+}_{pa} et S_{pa} semblent refléter les liens sociaux dans le corpus d'Epinions. Par rapport aux mesures P^{+} et P , elles permettent de définir des zones où la proportion de liens de méfiance est particulièrement importante.

Comme dans le cas de la recommandation, le fait d'inclure une correction pour le biais utilisateur et item, permet d'obtenir une relation plus claire entre la mesure choisie (RMSD ou proportion de liens de confiance) et notre score S^{s}_{pa} en comparaison avec le taux d'accords P^{s}.

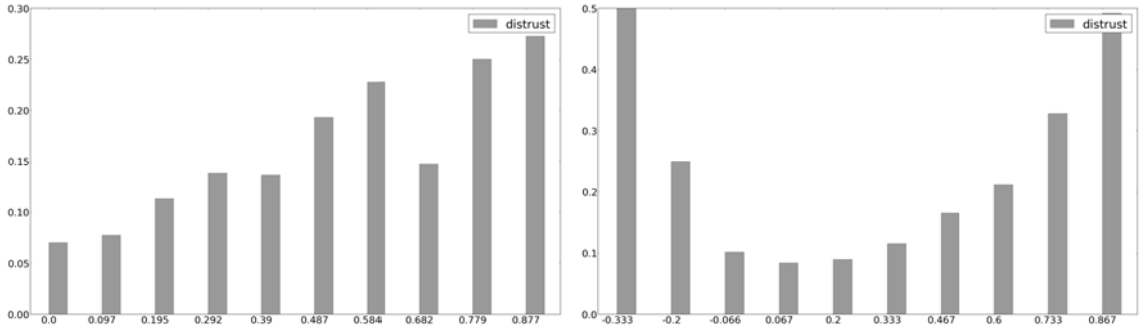


Figure *4.6* – *Répartition des liens de méfiance en fonction de P (à gauche) et de S_{pa} (à droite). Corpus Epinions*

Conclusion

Dans ce travail, nous avons poursuivi l'étude de la polarité des jugements du chapitre précédent en nous intéressant à une caractérisation plus fine des relations via la prise en compte des biais utilisateur et item. Le développement de telles métriques a des applications directes en recommandation - par exemple pour modifier les similarités des modèles de voisinage ou bien le critère à optimiser dans le cas des modèles basés sur un espace latent où les utilisateurs et items sont représentés (e.g. factorisation matricielle), ainsi que dans les réseaux sociaux, pour proposer des nouveaux liens entre utilisateurs.

Nous avons proposé deux scores, un score positif et un négatif, qui mesurent avec quel écart par rapport à l'aléatoire deux utilisateurs ont noté les mêmes items positivement ou négativement. Nous avons ensuite montré comment ces scores sont corrélés avec la différence de jugement des utilisateurs (filtrage collaboratif) et avec les liens de confiance dans un réseau social. En particulier, nous avons pu mettre au jour trois différents types de relation entre

les utilisateurs : les utilisateurs *fortement* similaires, les utilisateurs *faiblement* similaires et les utilisateurs dissimilaires. Les utilisateurs *fortement* similaires notent de la même manière et sont reliés par des liens sociaux positifs (confiance). Les utilisateurs *faiblement* similaires s'intéressent aux mêmes thèmes, mais n'aiment pas les mêmes items ; ils sont souvent reliés par des liens sociaux négatifs. Quant aux utilisateurs dissimilaires, leurs jugements communs ne sont dus qu'au hasard et ils ne sont généralement pas reliés dans le graphe social.

5

Prédiction de liens signés à partir des seuls jugements utilisateur

Résumé

L'information singée présente dans certains réseaux sociaux apporte une importante valeur ajoutée, et ce, pour de nombreuses tâches. Cette information demeure cependant relativement rare et délicate à obtenir. Certains travaux se sont alors intéressés à l'inférence de ces liens signés quand aucun lien explicite entre les utilisateurs n'est disponible. Ils utilisent pour cela d'autres sources d'information, par exemple les jugements sur le contenu crée par d'autres utilisateurs. Dans ce chapitre, nous étendons cette approche au cas où nous ne disposons d'aucune donnée sociale signée (directe ou indirecte). Nous proposons une approche qui repose uniquement sur les jugements que les utilisateurs partagent à propos d'items, ce qui permet de considérer de nombreux réseaux sociaux. Pour cela, nous exploitons les résultats des deux chapitres précédents, afin de prédire le signe d'un lien alors même qu'aucune donnée signée explicite n'est disponible dans le réseau social considéré.

Sommaire

Introduction

Comme nous avons pu le constater dans les précédents chapitres, les liens signés entre les utilisateurs sont difficiles à collecter. Il existe peu de sites web proposant ce type d'annotations et quand c'est le cas, elles sont peu utilisées (Epinions par exemple). Alors que les liens positifs sont perçus comme utiles pour les utilisateurs, les liens négatifs ont une connotation hostile et nous avons alors fait l'hypothèse que les utilisateurs n'en faisaient pas un large usage parce qu'ils les percevaient comme inconvenants, suggérant la tension et la provocation. Mais, puisque l'information apportée par ces liens négatifs permet d'améliorer les performances pour de nombreuses tâches classiques [Kunegis et al., 2013], la tâche consistant à inférer des liens signés depuis d'autres sources d'informations s'avère particulièrement attractive. Dans ce chapitre, nous exploitons les résultats obtenus jusque-là et nous les appliquons à la prédiction de liens signés dans des réseaux sans liens négatifs.

Pour intégrer nos précédentes observations, nous nous sommes inspirés des travaux récents [Tang et al., 2015] dans lesquels *Tang et al.* ont proposé d'apprendre un modèle de classification sur des données sociales positives en construisant un ensemble d'apprentissage où les exemples négatifs sont inférés à partir d'une autre source d'information. Comme source de données signées, ils ont exploité des interactions basées sur le contenu, c'est-à-dire les avis laissés par des utilisateurs à propos du contenu créé par d'autres utilisateurs. Ce type d'interactions s'observe notamment sur les sites de ventes en ligne, lorsqu'un utilisateur évalue la pertinence ou l'utilité d'un avis laissé sur un item (Amazon par exemple). Leur approche répond au problème de collecte de données négatives, car ce type d'interactions indirectes entre les utilisateurs est bien plus facile à collecter (les utilisateurs ont moins d'appréhension à évaluer un contenu plutôt qu'un utilisateur). En revanche, cette approche est limitée : elle suppose l'existence de telles interactions en nombre suffisant, alors que toutes les plates-formes ne possèdent pas de telles données et que tous les utilisateurs ne créent pas leur propre contenu - limitant ainsi le nombre d'interactions.

Dans notre travail, nous étudions le cas où nous ne possédons que deux sources d'information : un réseau social positif et l'historique de notes des utilisateurs à propos d'items. En nous basant sur les chapitres précédents, nous étendons le protocole proposé par [Tang et al., 2015] afin d'utiliser le seul historique de notes pour prédire la polarité des liens.

Nos contributions sont les suivantes :

— Nous adaptons le protocole proposé dans [Tang et al., 2015] pour construire un ensemble d'apprentissage signé à partir d'un réseau social classique et de l'historique de notes des utilisateurs

— Nous utilisons plusieurs ensembles de caractéristiques basées sur les données sociales : les interactions indirectes basées sur le contenu [Tang et al., 2015] et les caractéristiques définies dans les deux chapitres précédents

— Nous évaluons la tâche de prédiction de liens signés en comparant les performances sur trois corpus dérivés d'Epinions : (1) le réseau social est signé, (2) le réseau social non pas signé, mais avec des interactions signées indirectes entre les utilisateurs et enfin (3) le réseau social non signé et avec uniquement l'historique de notes des utilisateurs.

Ce chapitre est organisé de la façon suivante. Nous présentons la tâche et définissons le protocole de construction des données d'apprentissage à partir des différentes sources d'information à notre disposition, et ce, pour les trois cas de figure étudiés. Puis, nous présentons notre modèle de classification basé sur trois ensembles de caractéristiques que nous détaillons. Ensuite, nous évaluons nos modèles sur les données Epinions. Enfin, nous discutons de l'influence de chacune des caractéristiques présentées dans les chapitres précédents.

5.1 La prédiction de liens signés

Les deux chapitres précédents nous ont permis de mieux comprendre la sémantique des jugements communs entre les utilisateurs. Nous exploitons ici cette ressource riche et abondante pour inférer des liens sociaux négatifs lorsque ces liens ne sont pas disponibles explicitement dans un réseau social.

5.1.1 Sources d'informations signées

À la différence de la tâche classique de prédiction de liens (étudiée à la section 2.2.3), la tâche de prédiction de liens signés consiste à prédire le signe d'un lien déjà observé. Elle vient donc compléter la prédiction de liens non signés. C'est une tâche de classification supervisée à deux classes $\{+, -\}$ pour laquelle nous avons un ensemble de liens dont nous connaissons le signe (les données d'apprentissage) et un ensemble de liens pour lesquels nous cherchons à prédire le signe (les données de test).

Pour la construction des données d'apprentissage signées, nous distinguons trois cas de figure selon l'information disponible. Le cas n°1 est celui où nous disposons d'un réseau social signé. Le cas n°2 est celui étudié par [Tang et al., 2015] où les données signées sont inférées depuis les jugements d'un utilisateur sur le contenu crée par un autre. Enfin, le cas n°3 est celui que nous proposons et qui exploite l'historique des notes des utilisateurs à propos d'items.

Dans les trois cas, les liens signés de l'ensemble de test sont obtenus à partir du graphe social signé.

Cas n°1 : Données sociales signées

Le cas le plus simple est celui dans lequel les liens signés sont directement donnés par les utilisateurs. C'est le cas de quelques corpus comme celui d'Epinions étudié dans les chapitres précédents. Dans ce cas de figure, nous apprenons un modèle de classification directement à partir des données sociales signées à notre disposition. C'est notre premier cas d'étude et le cas traité dans [Leskovec et al., 2010b].

Cas n°2 : Interactions basées sur le contenu

Sachant que les données sociales du cas n°1 sont particulièrement rares, *Tang et al.* [Tang et al., 2015] ont récemment proposé un protocole afin d'apprendre ce modèle de classification à partir de données sociales uniquement positives, en inférant les liens négatifs d'apprentissage à partir d'autres sources d'information.

Nous décrivons ici le cas proposé par *Tang et al.* consistant à exploiter des interactions basées sur le contenu.

Certaines plates-formes invitent les utilisateurs à évaluer le contenu d'autres utilisateurs. C'est notamment le cas des sites de e-commerce autorisant les visiteurs à donner leur avis quant à la pertinence ou l'utilité d'un avis laissé sur un item donné.

Ces interactions sont destinées à évaluer le contenu lui-même, mais elles sont généralement corrélées aux avis des utilisateurs à propos des auteurs de ces contenus [Tang et al., 2015]. Ces interactions indirectes peuvent donc être utilisées comme données sociales signées. Le principal avantage est alors que ces interactions sont, en principe, beaucoup plus nombreuses puisqu'elles ne sont plus considérées comme inconvenantes ou provocantes - elles concernent le contenu et non l'auteur.

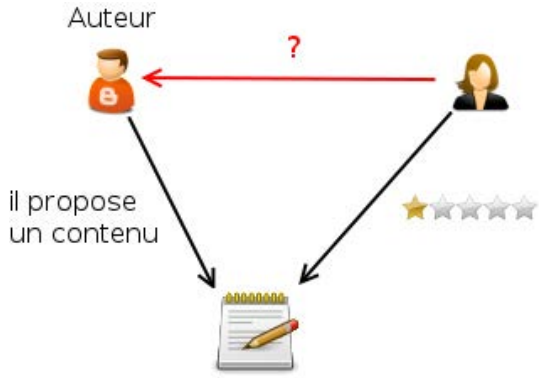


Figure 5.1 – *Exemple d'interaction indirecte entre deux utilisateurs, basée sur le contenu*

Pour construire les exemples négatifs de l'ensemble d'apprentissage, les auteurs proposent une règle simple (illustrée via la figure 5.1) : si un utilisateur u a attribué une note négative au contenu proposé par l'utilisateur v, alors un arc négatif de u vers v est ajouté au réseau social. Ensuite, si ce lien n'est pas en accord avec la théorie du statut (section 2.4.2), il est retiré. Enfin, tous les liens négatifs entre deux utilisateurs u et v reliés à un même utilisateur x tel que l'arc formé soit cohérent avec la théorie du statut sont ajoutés. Au final, le réseau social obtenu a des arcs positifs issus des déclarations des utilisateurs et des arcs négatifs inférés à partir des interactions à base de contenu.

Une première limite de cette approche réside dans le fait que les interactions concernent uniquement les auteurs de contenus. Ceci implique qu'on ne pourra exploiter des liens sociaux signés qu'en direction de ces auteurs. Ceci restreint notablement la quantité d'informations disponibles. En effet, la proportion d'acheteurs qui laisseront un commentaire sur un item ne

représente qu'une infime partie des utilisateurs.

Nous proposons d'exploiter une autre source d'information bien plus abondante : les jugements des utilisateurs étudiés dans les deux chapitres précédents, en adaptant le protocole précédent afin de tirer profit des divergences d'opinions entre les utilisateurs

Cas n°3 : Exploitation des jugements communs à propos d'items

Les jugements communs des utilisateurs sont une excellente source d'information négative. D'une part, ces données sont plus répandues. D'autre part, les utilisateurs partagent des items communs avec toutes sortes d'utilisateurs, même ceux avec qui ils n'ont pas de relation explicite.

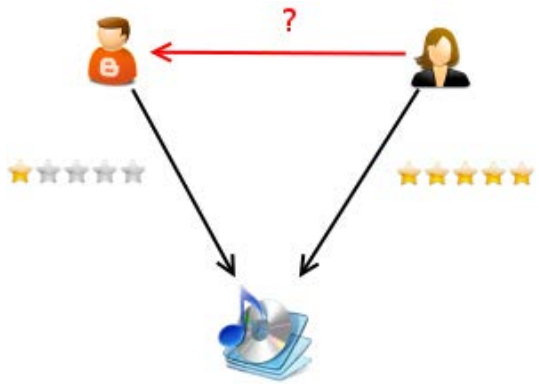


Figure 5.2 – *Exemple d'interaction indirecte entre deux utilisateurs, basée sur les jugements à propos d'items*

Nous utilisons le même protocole que précédemment, à la différence que la règle pour inférer un nouveau lien négatif exploite les différences de notation entre les utilisateurs (illustrée via la figure 5.2) : deux utilisateurs u et v dont l'écart de notation (en moyenne sur l'ensemble des items jugés en commun) est élevé seront reliés par un lien négatif. En particulier, nous mesurons la racine de la différence quadratique moyenne (*root mean square difference - RMSD*) entre leurs notes et si celle-ci dépasse un certain seuil, nous relions les utilisateurs par un lien négatif.

La RMSD se calcule de la façon suivante :

$$RMSD_{uv} = \sqrt{\frac{\sum_{i \in J_{uv}} (r_{ui} - r_{vi})^2}{|J_{uv}|}} \tag{5.1}$$

où r_{ui} est la note de l'utilisateur u à propos de l'item i et $J_{uv} = \{J_u \cap J_v\}$ l'ensemble des items jugés par u et par v.

Ce cas de figure, plus représentatif des données généralement à disposition, demeure néanmoins plus complexe à traiter que le cas de figure précédent puisque nous n'exploitons aucune interaction sociale signée directe ou indirecte, mais uniquement des données qui ne

sont pas directement corrélées avec une relation sociale. C'est donc à la fois le cas le plus accessible en terme de données disponibles et celui qui s'avère le plus difficile à traiter.

5.2 Le modèle de prédiction

Pour le modèle de classification, nous avons choisi de suivre les protocoles proposés par [Leskovec et al., 2010b] et [Tang et al., 2015] afin d'obtenir des résultats comparables en terme de performances. La prédiction du signe se fait par l'utilisation d'une régression logistique où la probabilité pour un lien d'être positif est donné par :

$$P(+|x) = \frac{1}{1 + e^{-(b_0 + \sum b_i x_i)}} \tag{5.2}$$

où $x = [x_1, x_2, ..., x_D]^T$ correspond à un ensemble de caractéristiques définissant chaque couple d'utilisateurs. Nous décrivons ces caractéristiques dans la suite.

Afin de maximiser les performances et de mieux modéliser les relations entre utilisateurs, nous combinons trois ensembles de caractéristiques : (1) des caractéristiques sociales issues des travaux de *Leskovec et al.* [Leskovec et al., 2010a, Leskovec et al., 2010b], (2) des caractéristiques sur les interactions basées sur le contenu [Tang et al., 2015] et enfin (3) des caractéristiques exploitant les notes des utilisateurs et issues des chapitres précédents.

Notons que ces caractéristiques sont calculées sur la base des données d'apprentissage. Ainsi, selon les trois cas de figures étudiés, ces caractéristiques n'auront pas nécessairement les mêmes valeurs pour les mêmes couples d'utilisateurs. Nous aurons l'occasion de revenir sur ce point lors de l'analyse des résultats.

Nous présentons chacun des ensembles de caractéristiques pour un couple d'utilisateurs u et v pour qui nous cherchons à prédire la polarité du lien.

5.2.1 Caractéristiques sociales

Les caractéristiques sociales exploitent uniquement l'information explicite des réseaux, c'est-à-dire les relations indiquées par les utilisateurs. Lorsque nous n'avons pas de liens signés explicites (cas 2 et 3), ces caractéristiques sont calculées à partir des relations signées inférées. Nous distinguerons deux ensembles de caractéristiques sociales : un ensemble de caractéristiques locales et un ensemble de caractéristiques basées sur le voisinage. Ces caractéristiques sont issues des travaux de *Leskovec et al.* [Leskovec et al., 2010a, Leskovec et al., 2010b] que nous avons présentés plus en détail dans la section 2.4.3.

Données locales

Le premier ensemble de caractéristiques, que nous noterons *Données locales* dans la suite, comptabilise le nombre de liens entrants et sortants observés chez chacun des deux utilisateurs u et v. En particulier, si nous cherchons à prédire un arc dirigé de u vers v, alors nous calculerons les caractéristiques suivantes :

— $d^+_{out}(u)$ et $d_{out}(u)$ les degrés sortants positif et négatif de u ;
— $d^+_{in}(v)$ et $d_{in}(v)$ les degrés entrants positif et négatif de l'utilisateur v
— $d_{out}(u)$ le degré sortant de u et $d_{in}(v)$ le degré entrant de v ainsi que $C(u, v)$ le nombre de voisins communs noté.

Ici, l'orientation a son importance. Pour un arc de v vers u, nous aurions calculé $d^+_{out}(v)$, $d^+_{in}(u)$, etc.

Données de voisinage

Afin de prendre en compte le voisinage, les caractéristiques que nous noterons *16 triades* comptabilisent le nombre de triades dans lesquelles les deux utilisateurs sont impliqués, à savoir le nombre de configurations distinctes (direction et signe) mettant en jeu un voisin commun à u et à v.

Il y a 16 types de triades possibles. Par exemple, il y a une caractéristique correspondant au nombre de voisins qui sont liés par un arc positif de u et un arc négatif de v. Puisque les arcs ont deux directions possibles et deux signes possibles, cela nous donne bien 16 configurations, illustrées dans la figure 5.3 ci-dessous.

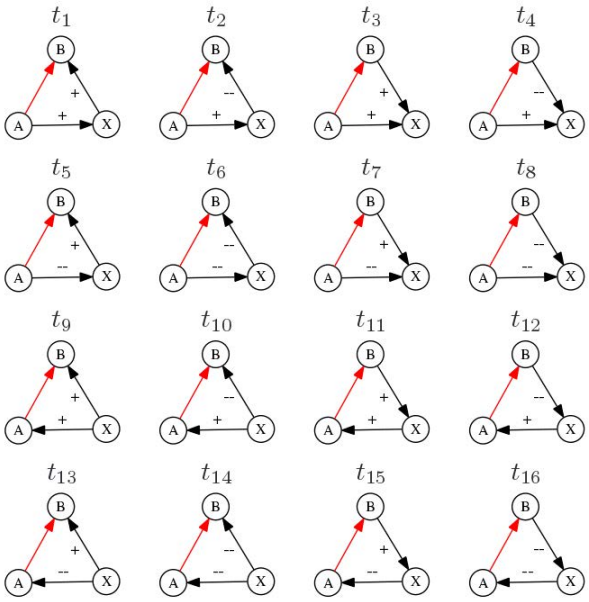


Figure 5.3 – *Les 16 configurations possibles de triades impliquant deux utilisateurs A et B. Illustration de Leskovec2010.*

Finalement, nous notons *All23* l'ensemble des caractéristiques sociales composées des deux ensembles précédents, *Données locales* et *16 triades* qui constituent l'ensemble des 23 caractéristiques définies dans [Leskovec et al., 2010a].

5.2.2 Caractéristiques basées sur les interactions indirectes

Comme nous l'avons évoqué précédemment et comme *Tang et al.* [Tang et al., 2015] l'ont souligné , les interactions indirectes basées sur le contenu sont très corrélées aux liens sociaux. Il est donc tout à fait logique de compléter les caractéristiques sociales avec de nouvelles caractéristiques basées sur le contenu même lorsque nous exploitons un graphe social déjà signé comme ensemble d'apprentissage. Pour cela, nous avons utilisé les caractéristiques proposées dans [Tang et al., 2015], à savoir :

— le nombre d'avis positifs et négatifs de u sur un contenu de v (et inversement, de v vers un contenu de u)

— Le coefficient de Jaccard pour les interactions positives (et négatives) entrantes (et sortantes) de u et de v

— Le nombre de contenus proposés par u et v

Ces caractéristiques seront notées *Interaction* dans la suite.

5.2.3 Caractéristiques basées sur les jugements communs

Ce troisième ensemble de caractéristiques exploite les deux chapitres précédents afin de représenter au mieux les jugements communs entre les utilisateurs. Notamment, nous exploitons la polarité des accords ainsi que les distributions de notes positives et négatives (pour les utilisateurs et les items).

Nous avons repris les similarités que nous avons définies, auxquelles nous avons ajouté quelques variantes. Nous décomposons ces scores en deux sous-groupes. D'un côté, nous aurons les scores prenant en compte la différence entre accords positifs et accords négatifs (l'ensemble est noté *Taux d'accord*) issus du troisième chapitre, de l'autre côté nous aurons les scores du quatrième chapitre prenant en compte les biais utilisateur et item (noté *Accords espérés*). À noter que nous différencions nos scores de surprise selon que l'on ait pris en compte la distribution de notes des utilisateurs sur l'ensemble de leurs items (S^s_{all}) (ce qui correspond aux scores du précédent chapitre) ou sur l'ensemble restreint des notes communes aux deux utilisateurs (S^s_{com}). L'intérêt de prendre en compte S^s_{com} est de compléter l'information capturée par S^s_{all} dans le cas où les items communs aux utilisateurs pourraient expliquer à eux seuls une différence de distribution entre l'ensemble de notes des utilisateurs et l'ensemble de leurs notes communes.

Nous résumons l'ensemble des caractéristiques dans le tableau 5.1.

5.3 L'évaluation

Pour évaluer la tâche de prédiction de liens signés, nous avons suivi le même protocole que [Leskovec et al., 2010b]. Nous avons construit un ensemble de données équilibré pour s'assurer d'avoir le même nombre d'instances des classes *positif* et *négatif*, protocole proposé initialement dans [Guha et al., 2004] et décrit dans la section 2.4.3. Puis, nous apprenons

	Caractéristiques
Taux d'accords	$P^+(u,v)$
	$P^-(u,v)$
	$\lvert J_u^+ \cap J_v^+ \rvert$
	$\lvert J_u^- \cap J_v^- \rvert$
	$\lvert J_u \cap J_v \rvert$
	$\lvert J_u^+ \rvert$
	$\lvert J_u^+ \rvert$
	$\lvert J_v^+ \rvert$
	$\lvert J_v^- \rvert$
Accords espérés	$S_{all}^+ = (G^+(u,v) - \mathbb{E}_{all}(G^+\lvert u,v))/G_{max}^+(u,v)$
	$S_{all}^- = (G_(^-u,v) - \mathbb{E}_{all}(G^-\lvert u,v))/G_{max}^-(u,v)$
	$S_{com}^+ = (G_(^+u,v) - \mathbb{E}_{com}(G^+\lvert u,v))/G_{max}^+(u,v)$
	$S_{com}^- = (G_(^-u,v) - \mathbb{E}_{com}(G^-\lvert u,v))/G_{max}^-(u,v)$
	$S_{all}^+ \times S_{all}^-$
	$S_{com}^+ \times S_{com}^-$
	$S_{all}^+ \times G_{max}^+$
	$S_{all}^- \times G_{max}^-$
	$S_{com}^+ \times G_{max}^+$
	$S_{com}^- \times G_{max}^-$

Table *5.1* *– Tableau récapitulatif des caractéristiques basées sur les notes des utilisateurs*

un classificateur (régression logistique) par validation croisée sur 10 ensembles. En revanche, nous avons choisi de garder tous les couples d'utilisateurs partageant au moins 10 voisins, quand [Leskovec et al., 2010b] n'a considéré que les couples d'utilisateurs avec un minimum de 25.

Données

Pour ces expériences, nous utilisons les données Epinions [Massa and Avesani, 2006] qui sont, à ce jour, les seules à notre disposition fournissant à la fois des données sociales signées, des interactions indirectes basées sur le contenu et des jugements utilisateurs. Plus précisément, nous avons réutilisé les interactions basées sur le contenu en ignorant l'auteur du contenu, et en supposant que les utilisateurs évaluent les items indépendamment de leur auteur.

Historique de notes

Nous avons décrit les statistiques concernant les notes dans les précédents chapitres. Mais cette fois, nous avons choisi de binariser en considérant 3 comme valeur neutre, soit :

$$\begin{cases} +1 & \text{pour une note supérieure à 3} \\ -1 & \text{pour une inférieure à 3} \end{cases} \tag{5.3}$$

Nous avons effectué des expériences avec une valeur neutre fixée à 4 et les résultats se sont avérés sensiblement inférieurs. Une des raisons à cela est que les utilisateurs ont tendance à noter en priorité les items qu'ils apprécient, ce qui mène à des évaluations majoritairement positives (le biais positivité). De telles évaluations seraient ignorées si la note neutre était fixée à 4. Le corpus Epinions ne contenant que peu de notes, il était nécessaire de préserver au maximum l'information présente dans ce corpus, ce qui nous a amené à utiliser le seuil défini en (5.3).

Interactions indirectes basées sur le contenu

Les données d'interaction sont traitées comme dans [Tang et al., 2015]. Un utilisateur u évalue l'utilité ou la pertinence du contenu d'un utilisateur v en donnant une note entre 0 et 5 . Nous appliquons la même transformation que précédemment et chaque interaction de u vers v est alors égal à $+1$ ou -1. Nous pourrons observer plusieurs interactions de la sorte de u vers v dans l'ensemble de données, par exemple lorsque v aura proposé plusieurs contenus annotés par u.

Un réseau social signé

Le corpus Epinions contient également un réseau social où les utilisateurs peuvent indiquer explicitement s'ils font confiance, ou non, en un autre utilisateur. La différence entre les liens sociaux et les interactions décrites ci-dessus est que les premiers sont dirigés vers un utilisateur particulier tandis que les seconds sont dirigés vers un contenu. Ces informations sont évidemment très corrélées [Tang et al., 2015].

Les statistiques clés du corpus Epinions sont résumées dans le tableau 5.2.

	Epinions
Tous les couples d'utilisateurs	
# d'utilisateurs	120 400
# de liens positifs	717 667
# de liens négatifs	123 705
Les couples d'utilisateurs partageant au moins 10 items	
# de liens positifs	279 276
# de liens négatifs	21 014

Table 5.2 – *Statistiques du jeu de données Epinions*

5.4 Les expériences

Nous rappelons que nous nous intéressons à trois cas de figure, correspondant à trois situations différentes que nous pouvons rencontrer en pratique :

— Cas 1 : le réseau social contient des liens signés explicites

— Cas 2 : le réseau social n'est pas signé, mais nous observons des interactions indirectes signées basées sur le contenu

— Cas 3 : le réseau social n'est pas signé et nous exploitons uniquement l'historique des notes utilisateurs

Pour les trois cas, nous nous évaluons sur un ensemble de liens signés explicites.

Notons que nos caractéristiques sont désavantagées par rapport à celles de [Tang et al., 2015] puisqu'il n'y a pas corrélation forte entre les jugements sur les items et les liens sociaux entre utilisateurs, à la différence des interactions sur le contenu.

5.4.1 Performance avec des liens signés explicites (cas n°1)

Les résultats sont indiqués dans le tableau 5.3 (première colonne).

Les caractéristiques sociales *All23* se comportent bien sur la tâche de prédiction avec une F-mesure de 0.89 ce qui est comparable aux résultats observés dans [Leskovec et al., 2010b].

Les caractéristiques basées sur les interactions indirectes et sur l'historique des notes donnent de moins bons résultats dans ce premier cas expérimental (0.81 et 0.78 respectivement). En revanche, elles permettent d'obtenir de meilleurs résultats quand elles sont ajoutées aux caractéristiques sociales ; 0.93 pour les caractéristiques basées sur les interactions et 0.90 pour celles basées sur l'historique de notes, au lieu de 0.89 avec les caractéristiques sociales seules. Ceci montre bien que ces données supplémentaires apportent un complément d'information, avec un léger avantage à celles basées sur les interactions du fait qu'elles sont davantage corrélées aux liens sociaux.

Nous remarquons par ailleurs que ces deux ensembles de caractéristiques (*Interactions* et *Notes utilisateurs*) se complètent bien l'un l'autre, avec une F-measure à 0.85. Cela montre donc tout l'intérêt de distinguer les deux ensembles issus pourtant des mêmes observations. Nous rappelons en effet que l'historique de notes des utilisateurs correspond ici aux avis concernant les contenus d'autres utilisateurs pour lesquels nous ignorons l'auteur.

Enfin, nous pouvons remarquer que si les caractéristiques basées sur les notes des utilisateurs permettent d'améliorer les performances des autres ensembles de caractéristiques pris un par un, ce n'est plus le cas une fois que nous combinons les trois ensembles de caractéristiques. Lorsque l'information d'interaction est présente, l'information apportée par les jugements communs ne semble pas utile puisqu'elles ne permettent pas d'améliorer les performances obtenues en combinant les ensembles *Interactions* et *All23* obtenant un score de 0.93.

5.4.2 Performance sans liens signés mais avec interactions indirectes (cas n°2)

Dans le tableau 5.3 (deuxième colonne) nous rapportons les résultats obtenus lorsque nous inférons les exemples négatifs d'apprentissage à partir des interactions basées sur le contenu.

	cas n°1	cas n°2	cas n°3
Caractéristiques sociales			
Données locales	0.85	0.56	0.35
16 triades	0.88	0.85	0.75
Total (*All23*)	0.89	0.87	0.79
Caractéristiques basées sur les interactions indirectes			
Interactions	0.81	0.50	-
Caractéristiques basées sur l'historique des notes			
Taux d'accords	0.75	0.65	0.67
Accords espérés	0.70	0.54	0.58
Total (*Notes utilisateurs*)	0.78	0.62	0.67
Combinaisons			
Sociales + Interactions	0.93	0.87	-
Sociales + Historique des notes	0.90	0.82	0.82
Historique des notes + Interactions	0.85	0.63	-
All			
Total	**0.93**	**0.88**	**0.82**

Table 5.3 – *Résultats en F-measure des trois cas considérés (réseau social signé, utilisation des interactions indirectes et utilisation de l'historique de notes) pour chaque ensemble de caractéristiques*

La première chose que nous observons est que les performances des caractéristiques *Données locales* s'effondrent, contrairement aux autres caractéristiques de voisinage qui gardent des performances sensiblement similaires. Cela montre que les informations à propos de la polarité des liens sont moins fiables qu'avec les données explicitées par les utilisateurs et que cela impacte davantage les caractéristiques qui en dépendent directement.

Nous notons également que les performances obtenues avec les caractéristiques issues des interactions sont également impactées par le changement de contexte. De plus, elles ne permettent pas de gain de performance une fois combinées avec les caractéristiques sociales. C'est également le cas des caractéristiques *Notes utilisateurs* qui obtiennent de moins bonnes performances comparées aux données sociales et qui se revèlent peu intéressantes en complément de ces dernières.

En revanche, les performances obtenues en combinant les trois ensembles sont légèrement meilleures ; ici, les deux ensembles de caractéristiques *Interactions* et *Notes utilisateurs* s'avèrent utiles pour compléter les caractéristiques sociales puisque nous obtenons au final une F-measure à 0.88 alors qu'aucune combinaison de deux ensembles de caractéristiques n'atteint une telle performance.

	Précision	Rappel	F-measure
Premier quartile	0.82	0.82	0.82
Second quartile	0.82	0.80	0.80
Troisième quartile	0.81	0.79	0.77

Table *5.4* *– Performances en prédiction obtenues selon la valeur du seuil RMSD dans le cas où les exemples négatifs d'apprentissage sont inférés à partir de l'historique de notes des utilisateurs*

5.4.3 Performance avec uniquement les jugements communs (cas n°3)

Nous analysons enfin le cas le plus intéressant où seuls les jugements des utilisateurs à propos d'items sont exploités pour inférer nos exemples négatifs d'apprentissage.

Nous pouvons commencer par analyser l'impact du seuil de RMSD à partir duquel nous considérons que les utilisateurs doivent être reliés par un lien négatif. Pour cela, nous avons appris notre modèle à partir de données d'apprentissage obtenues avec différents seuils (voir tableau 5.4), et ce, en utilisant les trois ensembles de caractéristiques à notre disposition. Nous pouvons observer que les meilleures performances sont obtenues en fixant le seuil au niveau du premier quartile, c'est-à-dire de manière à ce que 75% des couples soient reliés par un lien négatif.

Nous avons donc reporté l'ensemble des résultats obtenus avec le meilleur seuil (premier quartile) dans la troisième colonne du tableau 5.3. Il s'agit donc d'une borne supérieure obtenue sur le corpus Epinions. Notons que les caractéristiques basées sur les interactions ne peuvent plus être calculées dans ce cas, conformément à l'hypothèse de départ qui dit que seuls le réseau social non signé et les jugements utilisateurs sont à notre disposition.

Nous voyons (5.3) que dans ce cas, les performances obtenues par les caractéristiques basées sur l'historique de notes sont sensiblement supérieures à celles obtenues dans le cas précédent. De plus, les performances obtenues en les combinant avec les caractéristiques sociales sont exactement les mêmes que celles obtenues dans le cas précédent, à savoir une F-measure à 0.82, alors que les caractéristiques sociales obtiennent de moins bonnes performances seules, en comparaison des deux cas précédents. Ces deux résultats montrent qu'en l'absence d'interactions basées sur le contenu, exploiter les notes des utilisateurs est une bonne alternative pour prédire la polarité d'une relation sociale.

5.4.4 Étude de l'importance des caractéristiques

Afin d'avoir une vision plus qualitative sur l'importance des caractéristiques, nous pouvons regarder les coefficients appris par la régression logistique. Nous nous focalisons sur les coefficients appris dans le premier cas de figure, c'est-à-dire avec le réseau social signé, de manière à obtenir les coefficients les moins bruités possibles. De plus, nous avons appris une régression logistique pour chaque paire de caractéristiques (positive et négative) de notre ensemble résumées dans le tableau 5.1, puisque nos caractéristiques basées sur les notes sont

corrélées entre elles. Par exemple, nous avons calculé notre régression logistique avec P^+ et P comme seules caractéristiques ; cela nous donne une F-measure à 0.60 et les coefficients obtenus sont de 1.36 et 3.41 respectivement.

Les deux facteurs importants pour l'analyse de l'importance de nos caractéristiques sont les coefficients appris pour la régression logistique (les b_i de l'équation (5.2)) ainsi que le *odds ratio*. Un coefficient b_i inférieur à 0 signifie que la caractéristique associée est un signal de lien négatif. À l'inverse, un coefficient b_i positif est l'indicateur d'un lien positif. Le *odds ratio* (ou rapport des cotes) exprime quant à lui le degré de dépendance des caractéristiques avec la classe positive (lien positif). Il s'exprime sous la forme $o(+|x) = \frac{P(+|x)}{1 \quad P(+|x)}$ et s'interprète de la manière suivante : (1) une valeur à 1 indique que la polarité du lien n'est pas déterminée par la caractéristique considérée, (2) une valeur inférieure à 1 indique que les liens sont moins souvent positifs pour les utilisateurs avec une caractéristique élevée et inversement, (3) une valeur supérieure à 1 indique que les liens sont plus souvent positifs.

Afin d'interpréter plus facilement l'impact de chacune des caractéristiques, nous rapportons les valeurs du *odds ratio* par unité d'écart-type (4$^{\text{ème}}$ colonne du tableau 5.5). Cette normalisation facilite l'interprétation des résultats entre des caractéristiques réelles qui n'ont pas la même variation. Ce *odds ratio* normalisé s'exprime par :

$$\frac{o(+|x_i = \mu_i + \sigma_i)}{o(+|x_i = \mu_i)} = exp(\sigma_i b_i) \tag{5.4}$$

où $o(+|x_i = \mu_i)$ est le *odds ratio*, μ_i l'espérance pour la caractéristique x_i et σ_i son écart-type. Sur la figure 5.4, nous montrons le lien entre le *odds ratio* et la probabilité : une augmentation n'a pas la même signification selon la valeur absolue du *odds ratio*.

De la même manière, nous reportons les coefficients b_i appris pour la classe positive (équation (5.2)) normalisés par unité d'écart-type. Ces coefficients normalisés correspondent alors à l'effet *moyen* de chaque caractéristique sur le logarithme du *odds ratio*.

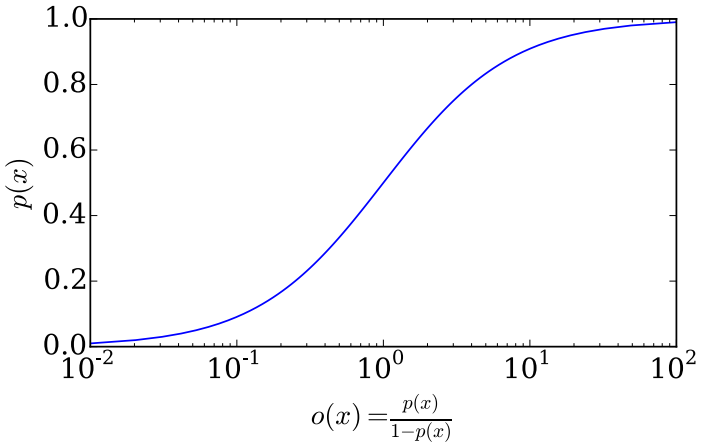


Figure 5.4 – *Lien entre le odds ratio et la probabilité.*

Nous pouvons faire les observations suivantes :

— Lorsque les utilisateurs sont en accord positif, ils ont généralement tendance à être reliés par un lien positif.

— Plus important, avoir un taux élevé d'accords négatifs est un bon indicateur d'une relation négative entre les deux utilisateurs. Cela est particulièrement vrai pour les caractéristiques basées sur les accords attendus, montrant ainsi que la prise en compte des biais utilisateur et item est importante.

Caractéristiques	Coefficients	Coefficients × Écart type	Odds ratio normalisé
$P^+(u,v)$	1.38	0.32	1.37
$P^-(u,v)$	-3.34	-0.14	0.86
S^+_{all}	1.65	0.41	1.50
S^-_{all}	-4.03	-0.16	0.85
S^+_{com}	1.60	0.41	1.50
S^-_{com}	-4.44	-0.17	0.84

Table *5.5* – *Les coefficients de la régression logistique pour la classe positive appris avec des données signées explicites.*

Conclusion

Prédire la polarité des relations entre les utilisateurs est important puisque cette donnée est une source précieuse d'informations pour les modèles travaillant avec les réseaux sociaux. Parce que de nombreux médias sociaux ne permettent pas aux utilisateurs d'exprimer des opinions négatives sur les autres, *Tang et al.* [Tang et al., 2015] ont proposé un protocole permettant d'exploiter les interactions indirectes entre les utilisateurs basées sur le contenu, une information à la fois signée et très corrélée aux liens sociaux. Plus précisément, ils proposent une manière de construire un ensemble d'apprentissage à partir de données uniquement positives, en inférant les exemples négatifs à partir des interactions indirectes.

Dans ce chapitre, nous proposons d'étendre ce protocole à une autre source d'information plus riche et plus abondante : l'historique de notes des utilisateurs. Cela nous permet alors de couvrir un plus large éventail de réseaux sociaux qui ne collectent ni les liens sociaux signés explicites, ni les interactions signées indirectes. En particulier, nous exploitons les résultats des deux précédents chapitres à propos des différences entre les accords positifs et négatifs afin de mieux modéliser les relations entre les utilisateurs. Nous nous intéressons alors à un cas plus complexe dans lequel les exemples négatifs d'apprentissage ne sont pas issus d'interactions entre utilisateurs.

Nous montrons sur l'ensemble de données Epinions que même si ces caractéristiques sont moins informatives que celles proposées dans [Leskovec et al., 2010b] et [Tang et al., 2015], elles sont porteuses d'information et constituent un signal extrêmement intéressant compte tenu de l'absence d'interactions signées entre les utilisateurs. De plus, il est possible qu'avec de plus grands jeux de données (par exemple Yahoo ! Music), les performances soient augmentées.

Ainsi, tirer profit des seuls jugements des utilisateurs se révèle pertinent lorsqu'aucune interaction signée n'est disponible, un cas particulièrement courant dans la majorité des réseaux sociaux actuels.

Troisième partie

Conclusion

Bilan et perspectives

Bilan

Dans ce manuscrit, nous nous sommes intéressés aux relations signées entre les individus. Alors que sur les principales plates-formes en ligne, les liens ont généralement une connotation positive (collaboration, amitié, partage), de nombreuses relations antagonistes entre utilisateurs existent à l'état *naturel* (méfiance, hostilité, désaccord). Cette source d'information signée a été le sujet de quelques travaux ces dernières années, par exemple [Leskovec et al., 2010a, Yang et al., 2012, Kunegis et al., 2010], et s'est avérée informative [Kunegis et al., 2013, Leskovec et al., 2010b] pour de nombreuses tâches classiques, notamment parce l'information apportée est d'une toute autre nature. Pourtant, ce sujet n'est pas simple à traiter puisque les données négatives sont d'une part difficiles à obtenir et d'autre part plus complexes à interpréter. Notre travail s'est alors porté sur l'utilisation et la définition de liens négatifs entre utilisateurs en utilisant d'autres sources d'information.

Notre hypothèse est que les liens sociaux négatifs sont trop complexes à collecter puisque l'utilisateur ne perçoit pas l'intérêt d'expliciter de tels liens : cela n'améliore pas ou peu son expérience et cela peut être vu comme une provocation. Alors que certaines plates-formes en ligne combattent les comportements malveillants, autoriser les relations négatives ne ferait donc que nourrir ces tensions. Ces relations sont de fait très peu observées dans les graphes de terrain. Sur d'autres plate-formes, les utilisateurs peuvent exprimer des avis sur le contenu ou le comportement d'autres utilisateurs. Ces interactions peuvent alors être collectées plus facilement (les utilisateurs ont moins de difficultés à exprimer un jugement sur un contenu par exemple), néanmoins elles demeurent aussi difficiles à interpréter et se limitent généralement aux auteurs de contenu.

C'est pour pallier le manque de données signées que nous avons proposé d'exploiter une autre source d'information plus abondante et riche en interactions entre les utilisateurs : l'historique de notes à propos d'items. Ce type de données est de plus en plus facile à obtenir depuis que de nombreuses plates-formes en ligne permettent de laisser un avis sur des items (sites de ventes en ligne, sites de partage de vidéos ou encore services de streaming de musiques). Certaines plates-formes encouragent même les retours des utilisateurs afin de leur offrir une recommandation plus personnalisée (Netflix, Apple Music). Plus l'historique de notes est fourni, plus les utilisateurs sont susceptibles de partager des avis communs. Nous avons alors exploité la polarité des jugements, c'est-à-dire la possibilité pour une opinion d'être positive ou négative, comme information signée pour l'étude des relations entre les utilisateurs.

Nous sommes partis de l'hypothèse que les jugements positifs et négatifs à propos d'un item n'ont pas la même sémantique, et que par conséquent, les accords entre les utilisateurs n'ont alors pas le même sens selon qu'ils concernent des items appréciés ou non. Nous avons proposé un modèle exploitant cette dissymétrie entre les deux polarités. En particulier nous avons défini des caractéristiques reflétant entre autres la proportion d'accords *positifs* et *négatifs* pour chaque couple d'utilisateurs et avons laissé le modèle apprendre le poids de chaque relation afin de pondérer l'importance des accords. La prise en compte de ce voisinage a permis d'augmenter les performances en prédiction de notes et en ordonnancement sur les deux jeux de données Epinions et Yahoo ! KDD Cup 2011.

Suite à ces résultats, nous avons affiné notre méthode en prenant en compte les modèles utilisateur et item. Puisque tous les utilisateurs ne notent pas de la même manière et que tous les items ne reçoivent pas le même nombre de notes, les accords n'ont pas la même probabilité d'occurrence. Un accord positif apportera une information faible s'il concerne un item populaire ou si les deux utilisateurs n'ont donné que des notes positives. Nous avons proposé deux scores, un score positif et un négatif, mesurant l'écart entre les accords positifs (ou négatifs) observés et ceux attendus en supposant qu'une note est expliquée par deux facteurs indépendants, l'utlisateur et l'item. Ces scores nous ont alors permis de montrer que les utilisateurs aux goûts similaires (qui aiment les mêmes produits) partagent plus d'accords positifs que prévu et sont généralement reliés par des liens sociaux positifs.

Ces différents résultats nous ont permis dans un troisième temps d'exploiter les différentes caractéristiques définies pour apprendre un modèle de classification pour l'inférence de la polarité des liens sociaux entre les utilisateurs. Puisque la classification nécessite des liens positifs et négatifs pour apprendre le modèle et que les liens négatifs sont si difficiles à collecter, nous avons proposé un protocole dans la continuité des travaux de *Tang et al.* [Tang et al., 2015] permettant d'exploiter l'historique des notes des utilisateurs pour inférer les exemples négatifs d'entraînement. Ce protocole permet alors d'être utilisé dans une plus grande variété de contextes où les utilisateurs sont reliés par de simples liens positifs, mais expriment leurs avis à propos de produits, de films ou encore de vidéos partagées. Les résultats se sont avérés prometteurs ; nous avons obtenu de bonnes performances pour notre tâche de prédiction du signe d'un lien, alors même qu'aucune donnée sociale signée n'a été utilisée lors de l'apprentissage.

Perspectives

Nos travaux ont permis de montrer qu'il existait une différence de sémantique entre les jugements positifs et négatifs, notamment au travers des accords entre les utilisateurs. Dans les travaux de recommandation, cette polarité n'est en général ni prise en compte pour l'étude des similarités entre les utilisateurs ni pour la construction des profils utilisateurs et items à partir de l'historique des notes. Ce travail ouvre donc quelques pistes pour les modèles classiques à base de facteurs latents où les jugements positifs et négatifs sont pour le moment considérés de manière similaire.

Pour les réseaux sociaux signés, l'exploitation des jugements utilisateurs s'est avérée pertinente pour inférer des relations signées dans le corpus Epinions. Néanmoins, le fait qu'Epinions soit le seul corpus à notre disposition, fournissant à la fois un historique de notes sur des items et un réseau social signé, limite fortement l'interprétation de nos résultats. Il nous semble nécessaire de valider ces hypothèses sur d'autres données.

Enfin, trouver de nouvelles sources d'informations indirectes entre les utilisateurs pourrait également s'avérer intéressant pour pallier le manque de données signées. Les données implicites ne sont pas rares sur les plates-formes en ligne : l'historique de navigation, de clics, de recherche, etc. De nombreuses données peuvent venir compléter celles à notre disposition. Parmi ces données, certaines pourraient fournir des informations négatives implicites.

Bibliographie

[Adamic and Adar, 2003] Adamic, L. A. and Adar, E. (2003). Friends and neighbors on the web. *Social networks*, 25(3) :211–230.

[Adamic and Glance, 2005] Adamic, L. A. and Glance, N. (2005). The political blogosphere and the 2004 u.s. election : Divided they blog. In *Proceedings of the 3rd International Workshop on Link Discovery*, LinkKDD '05, pages 36–43, New York, NY, USA. ACM.

[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 and possible extensions. *IEEE Trans. on Knowl. and Data Eng.*, 17(6) :734–749.

[Aggarwal, 2011] Aggarwal, C. C. (2011). *An introduction to social network data analytics*. Springer.

[Aggarwal and Philip, 2005] Aggarwal, C. C. and Philip, S. Y. (2005). Online analysis of community evolution in data streams. In *SDM*, pages 56–67. SIAM.

[Airoldi et al., 2006] Airoldi, E. M., Blei, D. M., Fienberg, S. E., Xing, E. P., and Jaakkola, T. (2006). Mixed membership stochastic block models for relational data with application to protein-protein interactions. In *Proceedings of the international biometrics society annual meeting*, page I5.

[Al Hasan et al., 2006] Al Hasan, M., Chaoji, V., Salem, S., and Zaki, M. (2006). Link prediction using supervised learning. In *SDM'06 : Workshop on Link Analysis, Counter-terrorism and Security*.

[Al Hasan and Zaki, 2011] Al Hasan, M. and Zaki, M. J. (2011). A survey of link prediction in social networks. In *Social network data analytics*, pages 243–275. Springer.

[Anderson, 2006] Anderson, C. (2006). *The Long Tail : Why the Future of Business Is Selling Less of More*. Hyperion.

[Baeza-Yates et al., 1999] Baeza-Yates, R., Ribeiro-Neto, B., et al. (1999). *Modern information retrieval*, volume 463. ACM press New York.

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

[Belkin et al., 2004] Belkin, M., Matveeva, I., and Niyogi, P. (2004). Regularization and semi-supervised learning on large graphs. In *Learning theory*, pages 624–638. Springer.

[Bell and Koren, 2007] Bell, R. M. and Koren, Y. (2007). Scalable collaborative filtering with jointly derived neighborhood interpolation weights. In *IEEE International Conference on Data Mining (ICDM*.

[Bellogín, 2013] Bellogín, Alejandro et de Vries, A. P. (2013). Understanding similarity metrics in neighbour-based recommender systems. In *Proceedings of the 2013 Conference on the Theory of Information Retrieval*, ICTIR '13, pages 13 :48–13 :55, New York, NY, USA. ACM.

[Bengio et al., 2006] Bengio, Y., Delalleau, O., and Le Roux, N. (2006). Label Propagation and Quadratic Criterion. In Chapelle, O., Schölkopf, B., and Zien, A., editors, *Semi-Supervised Learning*, pages 193–216. MIT Press.

[Bennett and Lanning, 2007] Bennett, J. and Lanning, S. (2007). The netflix prize. In *Proceedings of KDD cup and workshop*, volume 2007, page 35.

[Bhagat et al., 2011] Bhagat, S., Cormode, G., and Muthukrishnan, S. (2011). Node classification in social networks. In *Social network data analytics*, pages 115–148. Springer.

[Billsus and Pazzani, 1996] Billsus, D. and Pazzani, M. (1996). Learning probabilistic user models. In *Proceedings of the Workshop on Machine Learning for User Models, Sixth International Conference on User Modeling, Chia*. Springer.

[Blei et al., 2003] Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *the Journal of machine Learning research*, 3 :993–1022.

[Blondel et al., 2008] Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics : Theory and Experiment*, 2008(10) :P10008.

[Bobadilla et al., 2013] Bobadilla, J., Ortega, F., Hernando, A., and Gutiérrez, A. (2013). Recommender systems survey. *Knowledge-Based Systems*.

[Brandes et al., 2007] Brandes, U., Delling, D., Gaertler, M., Görke, R., Hoefer, M., Nikoloski, Z., and Wagner, D. (2007). On finding graph clusterings with maximum modularity. In *Graph-Theoretic Concepts in Computer Science*, pages 121–132. Springer.

[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.

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

[Canny, 2002] Canny, J. (2002). Collaborative filtering with privacy. In *Security and Privacy, 2002. Proceedings. 2002 IEEE Symposium on*, pages 45–57. IEEE.

[Chapelle et al., 2006] Chapelle, O., Schölkopf, B., Zien, A., et al. (2006). Semi-supervised learning.

[Chen et al., 2011] Chen, P.-l., Tsai, C.-t., Chen, Y.-n., Chou, K.-c., Li, C.-l., Tsai, C.-h., Wu, K.-w., Chou, Y.-c., Li, C.-y., Lin, W.-s., Yu, S.-h., Chiu, R.-b., Lin, C.-y., Wang, C.-c., Wang, P.-w., Su, W.-l., Wu, C.-h., Kuo, T.-t., Mckenzie, T. G., Chang, Y.-h., Ferng, C.-s., Ni, C.-m., Lin, H.-t., Lin, C.-j., and Lin, S.-d. (2011). A Linear Ensemble of Individual and Blended Models for Music Rating Prediction.

[Chiang et al., 2011] Chiang, K.-Y., Natarajan, N., Tewari, A., and Dhillon, I. S. (2011). Exploiting longer cycles for link prediction in signed networks. In *Proceedings of the 20th*

ACM international conference on Information and knowledge management, CIKM '11, pages 1157–1162, New York, NY, USA. ACM.

[Cialdini, 2012] Cialdini, R. (2012). *Influence et manipulation*. EDI8.

[Clauset et al., 2004] Clauset, A., Newman, M. E., and Moore, C. (2004). Finding community structure in very large networks. *Physical review E*, 70(6) :066111.

[Crandall et al., 2008] Crandall, D., Cosley, D., Huttenlocher, D., Kleinberg, J., and Suri, S. (2008). Feedback effects between similarity and social influence in online communities. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 160–168, New York, NY, USA. ACM.

[Cremonesi et al., 2010] Cremonesi, P., Koren, Y., and Turrin, R. (2010). Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 39–46. ACM.

[Davis, 1977] Davis, J. A. (1977). Clustering and structural balance in graphs. *Social networks. A developing paradigm*, pages 27–34.

[Deerwester et al., 1990] Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W., and Harshman, R. A. (1990). Indexing by latent semantic analysis. *JAsIs*, 41(6) :391–407.

[Delporte et al., 2013] Delporte, J., Karatzoglou, A., Matuszczyk, T., and Canu, S. (2013). Socially enabled preference learning from implicit feedback data. In Blockeel, H., Kersting, K., Nijssen, S., and ÅœelezñÃœ, F., editors, *Machine Learning and Knowledge Discovery in Databases*, volume 8189 of *Lecture Notes in Computer Science*, pages 145–160. Springer Berlin Heidelberg.

[Dror et al., 2012] Dror, G., Koenigstein, N., Koren, Y., and Weimer, M. (2012). The Yahoo! Music Dataset and KDD-Cup'11. *JMLR Workshop and Conference Proceedings*, 18 :3–18.

[Eckart and Young, 1936] Eckart, C. and Young, G. (1936). The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3) :211–218.

[Fortunato, 2010] Fortunato, S. (2010). Community detection in graphs. *Physics Reports*, 486(3) :75–174.

[Funk, 2006] Funk, S. (2006). Netflix Update : Try this at Home.

[Gauthier et al., 2014] Gauthier, L.-A., Piwowarski, B., and Gallinari, P. (2014). Filtrage collaboratif et intégration de la polarité des jugements. In *CORIA-CIFED 2014*.

[Gauthier et al., 2015a] Gauthier, L.-A., Piwowarski, B., and Gallinari, P. (2015a). Leveraging rating behavior to predict negative social ties. In *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*.

[Gauthier et al., 2015b] Gauthier, L.-A., Piwowarski, B., and Gallinari, P. (2015b). Polarité des jugements et des interactions pour le filtrage collaboratif et la prédiction de liens sociaux. In *CORIA 2015*.

[Girvan and Newman, 2002] Girvan, M. and Newman, M. E. J. (2002). Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12) :7821–7826.

[Goldberg et al., 1992] Goldberg, D., Nichols, D., Oki, B. M., and Terry, D. (1992). Using collaborative filtering to weave an information tapestry. *Commun. ACM*, 35(12) :61–70.

[Golub and Reinsch, 1970] Golub, G. H. and Reinsch, C. (1970). Singular value decomposition and least squares solutions. *Numerische mathematik*, 14(5) :403–420.

[Guha et al., 2004] Guha, R., Kumar, R., Raghavan, P., and Tomkins, A. (2004). Propagation of trust and distrust. In *Proceedings of the 13th international conference on World Wide Web*, WWW '04, pages 403–412, New York, NY, USA. ACM.

[Harary, 1953] Harary, F. (1953). On the notion of balance of a signed graph. *Michigan Math J*, 2 :143–146.

[Harary, 1955] Harary, F. (1955). On local balance and n balance in signed graphs. *Michigan Math J*, 3 :37–41.

[Heider, 1946] Heider, F. (1946). Attitudes and cognitive organization. *The Journal of psychology*, 21(1) :107–112.

[Herlocker et al., 1999] Herlocker, J. L., Konstan, J. A., Borchers, A., and Riedl, J. (1999). An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, pages 230–237, New York, NY, USA. ACM.

[Jamali and Ester, 2010] Jamali, M. and Ester, M. (2010). A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, RecSys '10, pages 135–142, New York, NY, USA. ACM.

[Kamvar et al., 2003] Kamvar, S. D., Schlosser, M. T., and Garcia-Molina, H. (2003). The Eigentrust algorithm for reputation management in P2P networks. *Proceedings of the twelfth international conference on World Wide Web - WWW '03*, page 640.

[Kashima and Abe, 2006] Kashima, H. and Abe, N. (2006). A parameterized probabilistic model of network evolution for supervised link prediction. In *Data Mining, 2006. ICDM'06. Sixth International Conference on*, pages 340–349. IEEE.

[Kerchove and Dooren, 2008] Kerchove, C. D. and Dooren, P. V. (2008). The PageTrust algorithm : How to rank web pages when negative links are allowed ? *the SIAM International Conference on Data Mining*, pages 346–352.

[Koenigstein et al., 2011] Koenigstein, N., Dror, G., and Koren, Y. (2011). Yahoo ! music recommendations : Modeling music ratings with temporal dynamics and item taxonomy. In *Proceedings of the Fifth ACM Conference on Recommender Systems*, RecSys '11, pages 165–172, New York, NY, USA. ACM.

[Koren, 2008] Koren, Y. (2008). Factorization meets the neighborhood : A multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 426–434, New York, NY, USA. ACM.

[Koren and Bell, 2011] Koren, Y. and Bell, R. (2011). Advances in collaborative filtering. In Ricci, F., Rokach, L., Shapira, B., and Kantor, P. B., editors, *Recommender Systems Handbook*, pages 145–186. Springer US.

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

[Krulwich, 1997] Krulwich, B. (1997). Lifestyle finder : Intelligent user profiling using large-scale demographic data. pages 37–45.

[Kunegis and Lommatzsch, 2009] Kunegis, J. and Lommatzsch, A. (2009). Learning spectral graph transformations for link prediction. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 561–568, New York, NY, USA. ACM.

[Kunegis et al., 2009] Kunegis, J., Lommatzsch, A., and Bauckhage, C. (2009). The slashdot zoo : mining a social network with negative edges. In *Proceedings of the 18th international conference on World wide web*, WWW '09, pages 741–750, New York, NY, USA. ACM.

[Kunegis et al., 2013] Kunegis, J., Preusse, J., and Schwagereit, F. (2013). What is the added value of negative links in online social networks ? In *Proceedings of the 22nd WWW conference*.

[Kunegis et al., 2010] Kunegis, J., Schmidt, S., Lommatzsch, A., and Lerner, J. (2010). Spectral analysis of signed graphs for clustering, prediction and visualization. In *Proc. SIAM Int. Conf. on Data Mining*, pages 559–570.

[Lang, 1995] Lang, K. (1995). Newsweeder : Learning to filter netnews. In *in Proceedings of the 12th International Machine Learning Conference (ML95*.

[Lees-Miller et al., 2008] Lees-Miller, J., Anderson, F., Hoehn, B., and Greiner, R. (2008). Does wikipedia information help netflix predictions ? In *Machine Learning and Applications, 2008. ICMLA '08. Seventh International Conference on*, pages 337–343.

[Leskovec and Horvitz, 2008] Leskovec, J. and Horvitz, E. (2008). Planetary-scale views on a large instant-messaging network. In *Proceedings of the 17th international conference on World Wide Web*, pages 915–924. ACM.

[Leskovec et al., 2010a] Leskovec, J., Huttenlocher, D., and Kleinberg, J. (2010a). Predicting positive and negative links in online social networks. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 641–650, New York, NY, USA. ACM.

[Leskovec et al., 2010b] Leskovec, J., Huttenlocher, D., and Kleinberg, J. (2010b). Signed networks in social media. In *Proceedings of the 28th international conference on Human factors in computing systems*, CHI '10, pages 1361–1370, New York, NY, USA. ACM.

[Liben-Nowell and Kleinberg, 2007] Liben-Nowell, D. and Kleinberg, J. (2007). The link-prediction problem for social networks. *J. Am. Soc. Inf. Sci. Technol.*, 58(7) :1019–1031.

[Ma et al., 2009] Ma, H., Lyu, M. R., and King, I. (2009). Learning to recommend with trust and distrust relationships. In *Proceedings of the third ACM conference on Recommender systems*, pages 189–196. ACM.

[Marlin et al., 2012] Marlin, B., Zemel, R. S., Roweis, S., and Slaney, M. (2012). Collaborative filtering and the missing at random assumption. *arXiv preprint arXiv :1206.5267*.

[Massa and Avesani, 2006] Massa, P. and Avesani, P. (2006). Trust-aware bootstrapping of recommender systems. In *ECAI 2006 Workshop on Recommender Systems*, pages 29–33+.

[McCallum and Nigam, 1998] McCallum, A. and Nigam, K. (1998). A comparison of event models for Naive Bayes text classification. In *AAAI Workshop on Learning for Text Categorization*.

[Milgram, 1967] Milgram, S. (1967). The small world problem. *Psychology today*, 2(1) :60–67.

[Mitzenmacher, 2001] Mitzenmacher, M. (2001). A brief history of lognormal and power law distributions. In *Proceedings of the Allerton Conference of Coomunication, Control, and Computing*, pages 182–191.

[Newman, 2001] Newman, M. E. (2001). Clustering and preferential attachment in growing networks. *Physical Review E*, 64(2) :025102.

[Newman, 2004] Newman, M. E. (2004). Fast algorithm for detecting community structure in networks. *Physical review E*, 69(6) :066133.

[Newman, 2005] Newman, M. E. (2005). A measure of betweenness centrality based on random walks. *Social networks*, 27(1) :39–54.

[Newman and Girvan, 2004] Newman, M. E. and Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical review E*, 69(2) :026113.

[Oard et al., 1998] Oard, D. W., Kim, J., et al. (1998). Implicit feedback for recommender systems. In *Proceedings of the AAAI workshop on recommender systems*, pages 81–83.

[Page et al., 1999] Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The pagerank citation ranking : Bringing order to the web. Technical Report 1999-66, Stanford InfoLab. Previous number = SIDL-WP-1999-0120.

[Paterek, 2007] Paterek, A. (2007). Improving regularized singular value decomposition for collaborative filtering. In *Proceedings of KDD cup and workshop*, volume 2007, pages 5–8.

[Pazzani, 1999] Pazzani, M. J. (1999). A framework for collaborative, content-based and demographic filtering. *Artif. Intell. Rev.*, 13(5-6) :393–408.

[Popescul and Ungar, 2003] Popescul, A. and Ungar, L. H. (2003). Statistical relational learning for link prediction. In *IJCAI workshop on learning statistical models from relational data*, volume 2003. Citeseer.

[Pradel et al., 2012] Pradel, B., Usunier, N., and Gallinari, P. (2012). Ranking With Non-Random Missing Ratings : Influence Of Popularity And Positivity on Evaluation Metrics. In *Proceedings of RecSys'12*.

[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 *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*, CSCW '94, pages 175–186, New York, NY, USA. ACM.

[Ricci et al., 2010] Ricci, F., Rokach, L., Shapira, B., and Kantor, P. B. (2010). *Recommender Systems Handbook*. Springer-Verlag New York, Inc., New York, NY, USA, 1st edition.

[Richardson et al., 2003] Richardson, M., Agrawal, R., and Domingos, P. (2003). Trust management for the semantic web. In *The Semantic Web-ISWC 2003*, pages 351–368. Springer.

[Salakhutdinov et al., 2007] Salakhutdinov, R., Mnih, A., and Hinton, G. (2007). Restricted boltzmann machines for collaborative filtering. In *Proceedings of the 24th international conference on Machine learning*, pages 791–798. ACM.

[Salton, 1989] Salton, G. (1989). *Automatic Text Processing : The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

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

[Sarwar et al., 2000] Sarwar, B. M., Karypis, G., Konstan, J. A., and Riedl, J. T. (2000). Application of dimensionality reduction in recommender system – a case study. In *IN ACM WEBKDD WORKSHOP*.

[Schafer et al., 2007] Schafer, J. B., Frankowski, D., Herlocker, J., and Sen, S. (2007). Collaborative filtering recommender systems. In *The adaptive web*, pages 291–324. Springer.

[Schafer et al., 2001] Schafer, J. B., Konstan, J. A., and Riedl, J. (2001). E-commerce recommendation applications. *Data Min. Knowl. Discov.*, 5(1-2) :115–153.

[Sebastiani, 2002] Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1) :1–47.

[Symeonidis et al., 2010] Symeonidis, P., Tiakas, E., and Manolopoulos, Y. (2010). Transitive node similarity for link prediction in social networks with positive and negative links. In *Proceedings of the fourth ACM conference on Recommender systems*, RecSys '10, pages 183–190, New York, NY, USA. ACM.

[Tang et al., 2015] Tang, J., Chang, S., Aggarwal, C., and Liu, H. (2015). Negative link prediction in social media. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, WSDM '15, pages 87–96, New York, NY, USA. ACM.

[Taskar et al., 2003] Taskar, B., Wong, M.-F., Abbeel, P., and Koller, D. (2003). Link prediction in relational data. In *Advances in neural information processing systems*, page None.

[Victor et al., 2011] Victor, P., Cornelis, C., and De Cock, M. (2011). Trust and distrust-based recommendations. In *Trust Networks for Recommender Systems*, pages 109–153. Springer.

[Wasserman and Faust, 1994] Wasserman, S. and Faust, K. (1994). *Social network analysis : Methods and applications*, volume 8. Cambridge university press.

[Wu et al., 2011] Wu, Y., Yan, Q., Bickson, D., Low, Y., and Yang, Q. (2011). Efficient multicore collaborative filtering. In *ACM KDD CUP workshop*.

[Yang et al., 2012] Yang, S.-H., Smola, A. J., Long, B., Zha, H., and Chang, Y. (2012). Friend or frenemy ? : Predicting signed ties in social networks. In *SIGIR*, New York, NY, USA. ACM.

[Zachary, 1977] Zachary, W. W. (1977). An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33(4) :pp. 452–473.

[Zhou et al., 2004] Zhou, D., Bousquet, O., Lal, T. N., Weston, J., and Schölkopf, B. (2004). Learning with local and global consistency. In *Advances in Neural Information Processing Systems 16*.

[Zhou et al., 2008] Zhou, Y., Wilkinson, D., Schreiber, R., and Pan, R. (2008). Large-scale parallel collaborative filtering for the netflix prize. In *Proc. 4th Int'l Conf. Algorithmic Aspects in Information and Management, LNCS 5034*, pages 337–348. Springer.

[Zhu, 2005] Zhu, X. (2005). Semi-supervised learning literature survey.

[Zhu and Ghahramani, 2002] Zhu, X. and Ghahramani, Z. (2002). Learning from labeled and unlabeled data with label propagation. Technical report, Citeseer.

A

Calcul de l'espérance du gain

Le gain espéré (section 4.2.5) considère toutes les permutations des notes sur un ensemble d'items. Pour le calculer de manière efficace, nous utilisons des techniques combinatoires que nous détaillons ici.

Nous commençons tout d'abord par calculer quel est le gain espéré avec un nombre k donné d'accords de polarité s. Il faut considérer toutes les façons de choisir k items parmi n, ce qui donne :

$$E(G^s(u,v)|k) = \frac{\binom{n-1}{k-1}}{\binom{n}{k}} \sum_{i \in I} g_i^s$$

Pour calculer l'espérance du gain $\mathbb{E}(G^s(u,v))$, il nous reste donc à calculer la probabilité d'avoir k accords de polarité s entre deux utilisateurs. Pour cela nous utilisons la formule du multinôme de Newton qui permet de calculer le nombre de partitions ordonnées. Ainsi, pour un couple (u,v) donné, sachant n le nombre de notes communes, α le nombre de notes de signe s de u et β le nombre de notes de signe s de v, alors le nombre de cas où nous observons avec k accords de signe s est :

$$\binom{n}{k, \alpha-k, \beta-k, n+k-\alpha-\beta} = \frac{n!}{k!(\alpha-k)!(\beta-k)!(n+k-\alpha-\beta)!} \tag{A.1}$$

Pour obtenir une probabilité, il faut calculer la somme sur tous les k de l'eq. (A.1), ce qui correspond aux nombres de façons de choisir α et β élements parmi n :

$$\binom{n}{\alpha} \times \binom{n}{\beta}$$

Finalement, le gain espéré dépend de toutes les combinaisons possibles, i.e. en considérant que k peut prendre ses valeurs dans $\{0, 1, ...n\}$ où n est le nombre de notes communes :

$$E(G^s(u,v)) = \frac{\sum_{k=max(0,\alpha+\beta-n)}^{k=min(\alpha,\beta)} \binom{n}{k,\alpha-k,\beta-k,n+k-\alpha-\beta} \times E(G^s(u,v)|k)}{\binom{n}{\alpha} \times \binom{n}{\beta}}$$