REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE IBN KHALDOUN – TIARET

MEMOIRE

Présenté à:

FACULTÉ MATHEMATIQUES ET INFORMATIQUE
DÉPARTEMENT D'INFORMATIQUE

Pour l'obtention du diplôme de :

MASTER

Spécialité: Génie informatique

par:

MESLEM Narimene
HABIBI Fatima Zohra

Sur le thème

Optimisation de la prédiction dans un réseau biparti

Soutenu publiquement le .. / .. / 2022 à Tiaret devant le jury composé de :

Mr. BOUDAA Boudjema	Grade MCA	President
Mr. BERBER Elmehdi	Grade MAA	Examinateur
Mr. KHAROUBI Sahraoui	Grade MCA	Encadreur

2021-2022

Remerciment

« Le remerciement est plus beau que le déni de la belle »

Au terme de ce travail nous tenons a remercier en premier lieu le dieu qui nous a donné la force , le courage et la volonté d'achever cette réalisation et sans lesquelles notre travail n'aurait jamais été accompli.
Nos sincères remerciement a notre encadreur Mr KHAROUBI SAHRAOUI pour son apport considérable, ses précieuses orientations méthodologiques et ses encouragements.
Nos vifs remerciements vont également aux membres du jury Mr BOUDAA Boudjemaa ,Mr BERBER Elmehdi. Pour l'intérêt qui ont portés à notre recherche en acceptant à l'examiner et l'enrichir par leurs propositions.
Ce mémoire n'aurait jamais pu voir le jour sans le soutien actif de nos parents qui nous ont toujours encouragés moralement et matériellement.
Enfin on tient a exprimer vivement nos remerciement a toutes les personnes qui ont contribué de près ou de loin à sa réalisation ,car un projet ne peut pas être le fruit d'une seule personne

Dédicace

Je dédie ce travail :

A ma famille elle qui m'a doté d'une éducation digne, son amour a fait de moi ce que je suis aujourd'hui.

A mes Chères parents « MOHAMED et FATMA »pour tous leur sacrifices, leur amour, leur tendresse, leur soutien et leur priére tout au long de mes etudes.

A mes Chères sœurs Sihem, Fatima, Tamani pour leurs encouragements permanents, et leur soutien moral.

A mon chers frère Sid Ahmed « yarhamoho ALLAH ».

A mes Chères amies loubna, imene, sarah, fadhila , amina et a tout ceux avec qui j'ai passé des meilleurs moment.

A toute ma famille pour leur soutien tout au long de mon parcours universitaire.

Merci d''etre toujours l`a pour moi.

MESLEM Narimene

Dédicace

Je dédie ce travail:

A mes chers parents, Un sentiment particulier de gratitude pour tous leurs sacrifices, leur amour, leur tendresse, leur soutien et leurs prières tout au long de mes études,

A mes belles sœurs pour leurs encouragements permanents, et leur soutien moral, et sont très spéciales,

A mes chers frères Habib et Mohamed pour leur appui et leur encouragement, ne m'ont jamais quitté mon côté,

A toute ma famille pour leur soutien tout au long de mon parcours universitaire,

Merci à tout.

HABIBI Fatima Zohra

Résumé

Ce projet se situe dans le cadre du domaine de recherche d'information plus particulièrement les systèmes de recommandation (SR). Un système de recommandation est un outil de recherche d'information et de filtrage qui vise à proposer aux utilisateurs des items qui pourraient les intéresser. La plupart des solutions des SR se basent sur l'analyse des préférences des utilisateurs et leurs évaluations implicites ou explicites pour les items.

Les différentes évaluations (appelées aussi votes) sont souvent représentées sous forme d'une matrice utilisateurs x items. L'objectif de recommandation consiste à prévoir les évaluations manquantes dans cette matrice.

Nous nous intéressons dans ce projet à l'exploration des approches topologiques pour le calcul de recommandation afin de pallier à certains problèmes des méthodes classiques.

En effet, la matrice d'évaluation peut être vue comme une matrice d'adjacence d'un graphe biparti qui relie les deux ensembles utilisateurs et items. La problématique de recommandation se réduit alors à un problème de prédiction de liens dans un graphe biparti.

L'objectif de notre travail est de réaliser un système de recommandation où la première étape consiste à collecter et explorer des informations liées aux items. La deuxième étape consiste à exploiter les informations tirées de la première étape lors du calcul de prédiction de liens entre les utilisateurs et les items.

Table des figures

Liste des tableaux

Table des matières

Introduction générale

Les systèmes de recommandation (SR) sont un sujet de recherche populaire qui vise à aider les utilisateurs à trouver des articles qui pourront les intéresser en fournissant des suggestions qui correspondent étroitement à leurs intérêts.

Des différents algorithmes de recommandation ont été appliqués pour fournir un mécanisme automatique et intelligent permettant de filtrer l'excès d'informations disponibles pour les utilisateurs et de faire des recommandations personnalisées d'informations, de produits et de services lors d'une interaction en direct.

Une des approches les plus efficaces pour créer des systèmes de recommandation est le filtrage collaboratif (CF). Elle utilise les préférences connues sous forme de notes d'un groupe d'utilisateurs sur un ensemble d'items pour prédire leurs préférences inconnues à d'autres items et ainsi les recommander. Cette technique ainsi définie et utilisée souffre de plusieurs limites qui sont dues à plusieurs causes. D'un côté, le manque d'information relative aux préférences des utilisateurs (peu d'utilisateurs expriment explicitement leurs préférences) peut réduire la qualité de recommandation. De l'autre côté, l'incapacité de gérer l'arrivée d'un nouvel utilisateur et/ou un nouvel item puisqu'il n'a pas encore d'historique de préférences.

Pour surmonter ces faiblesses, la recherche dans ce domaine s'est orientée vers l'utilisation des graphes qui peuvent être une source d'information et de relations latentes entre les utilisateurs et les items. En effet, les utilisateurs et les items ainsi que la relation entre eux peuvent être présentés par un graphe biparti et le problème de recommandation sera traduit par un problème de prédiction de liens.

Chapitre 1

Filtrage D'information

1.1 Introduction

Généralement, l'accès à l'information sur Internet se fait de deux façons différentes. La première c'est une recherche active de l'information est accomplie par l'intermédiaire d'outils de recherche d'informations. Ces derniers ont pour objectif de fournir à l'utilisateur les documents qui vont satisfaire la requête de recherche formulée [31]. La deuxième c'est l'accès à l'information jugée pertinente qui se fait par des systèmes de filtrage d'information. À l'opposé des outils de recherche d'informations, le filtrage d'information ne requiert pas une formulation systématique du besoin informationnel de l'utilisateur. Ainsi, cette approche permet notamment de réduire la complexité de la recherche d'informations et facilite l'accès à l'information. Le système de filtrage d'informations achemine l'information aux uti1isateurs en se basant sur leurs profils. Ces derniers sont établis grâce à des techniques d'apprentissage des goûts des utilisateurs [31].

1.2 Description de filtrage d'information

Le « filtrage de l'information » est un processus qui consiste à extraire d'une masse importante les informations les plus pertinentes. Il s'agit donc de proposer à l'utilisateur un contenu susceptible de correspondre à ses besoins, après celui-ci ait définit ses centres d'intérêt.

C'est le profil de l'utilisateur, qui détermine les informations qui lui seront transmises, la modélisation de l'utilisateur est un élément essentiel du filtrage, elle est basée sur des approches techniques, les trois principales sont le modèle canonique, le modèle explicite et le modèle automatique.

Le filtrage intègre aussi les opérations d'exploitation et de présentation des résultats. Les informations à mettre à la disposition de l'utilisateur sont extraites de sources différentes et évoluent dans le temps.

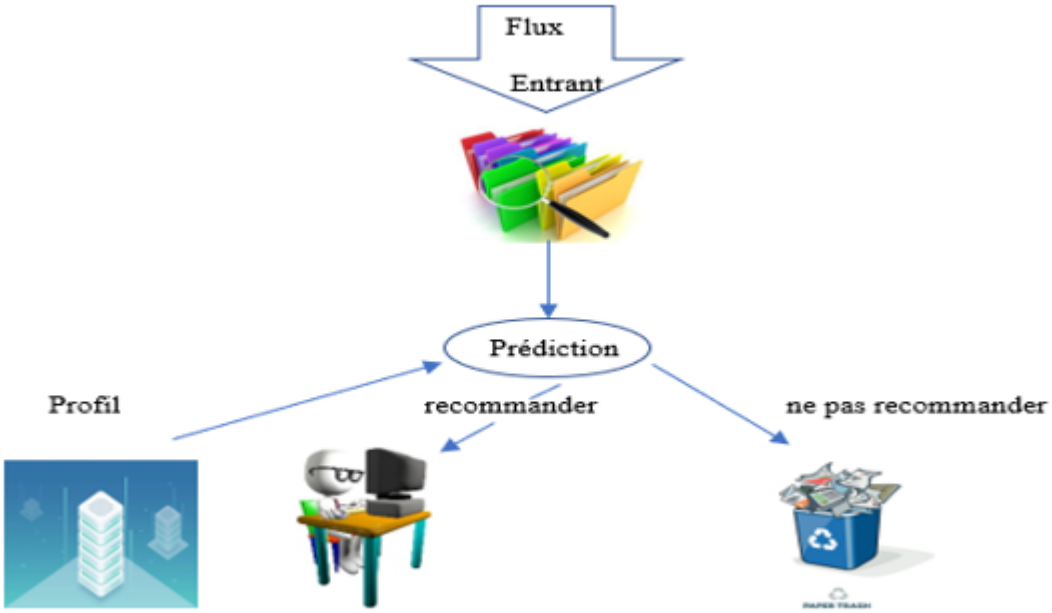

FIGURE 1.1 – Filtrage d'information

Le filtrage peut être vu comme la sélection d'informations pertinentes sur un flux entrant (voir Figure 1.1). Le système fait une « prédiction » quant à l'intérêt que présente l'information pour l'utilisateur. Cette prédiction s'appuie sur le « profil » de cet utilisateur et aboutit à une prise de décision : « recommander » ou « ne pas recommander » l'information[95].

Le problème de filtrage d'informations peut être formulé de la manière suivante [37]. Soit C un ensemble d'utilisateurs et S un ensemble de documents à recommander. Les deux ensembles peuvent être volumineux et contenant souvent des milliers de documents (ou utilisateurs). Soit U une fonction qui mesure l'utilité que représentera un document s à un utilisateur c. On cherche alors des documents s' de manière à maximiser la fonction d'utilité u. D'une manière plus formelle on peut écrire :

$$U : C \times S \rightarrow R$$
$$\forall c \in C, s'_c = arg_{\ s \in S} \max u(c, s)$$

L'utilité d'un document est souvent représentée par un vote ou une note soit donnée par l'utilisateur de manière explicite soit estimée par le système de manière implicite. Chaque utilisateur c de l'espace C est représenté par un profil, ce profil peut ne contenir que les votes de cet utilisateur dans les cas les plus simples, et peut être aussi plus complet contenant d'autres informations sur l'utilisateur, démographiques par exemple (sexe, âge, profession, situation familiale...).

Traditionnellement, les systèmes de filtrage d'informations ont été classés en cinq catégories : les systèmes à base de contenu, les systèmes de filtrage collaboratif, les

systèmes de filtrage hybrides, les système de filtrage social, les système de filtrage personnaliser et adaptatifs. Cette classification dépend de la manière avec laquelle l'utilité ou la pertinence éventuelle est calculée ou estimée.

1.3 Historique des systèmes de filtrage

La capacité des ordinateurs pour faire des recommandations à des utilisateurs a été reconnue assez tôt dans l'histoire de l'informatique. Un système bibliothécaire Grundy [32], était une première étape vers des systèmes de recommandation automatiques. Ce système était assez primitif. Il classait les utilisateurs en "stéréotypes" en se basant sur une courte interview, et utilisait ces stéréotypes pour produire des recommandations de livres. Ce travail constituait une première tentative intéressante dans le domaine des systèmes de recommandation. Cependant, son utilisation est restée très limitée.

Au début des années 1990, le filtrage collaboratif apparait comme une solution pour faire face à la surcharge d'information. L'année 1992 voit l'apparition du système de recommandation de documents Tapestry [33], ainsi que la création du laboratoire de recherche GroupLens, qui travaille explicitement sur le problème de la recommandation automatique dans le cadre des forums de news de Usenet. Tapestry avait pour but de recommander à des groupes d'utilisateurs des documents issus des newsgroups susceptibles de les intéresser. L'approche utilisée était de type "plus proches voisins" à partir de l'historique de l'utilisateur. On parle alors de filtrage collaboratif manuel, comme une réponse au besoin d'outils pour le filtrage de l'information énoncé à la même époque. La recommandation résulte d'une action collaborative des utilisateurs qui recommandent à d'autres utilisateurs des documents en leur attribuant des notes d'intérêt selon certains critères. Les systèmes de filtrage collaboratif automatiques apparaissent ensuite. GroupLens [34] utilise cette technique pour identifier les articles de Usenet susceptibles d'être intéressants pour un utilisateur donné. Les utilisateurs doivent seulement attribuer des notes ou effectuer d'autres opérations observables (par exemple, lire un article) ; le système combine alors ces données avec les notes ou les actions d'autres utilisateurs pour fournir des résultats personnalisés. Avec ces systèmes, les utilisateurs n'ont aucune connaissance directe des opinions des autres utilisateurs, ni des articles présents dans le système.

Au cours de ces dernières années, les systèmes de recommandation deviennent un sujet d'un intérêt croissant dans les domaines de l'interaction homme-machine, de l'apprentissage automatique ainsi que la recherche d'information. En 1995 apparaissent successivement Ringo [35], un système de recommandation de musique, basé sur les appréciations des utilisateurs et Bellcore [36], un système de recommandation de vidéos.

La même année, GroupLens crée la société Net Perceptions dont le premier client a été Amazon. De nos jours, les systèmes de recommandation sont devenus des composantes incontournables pour la plupart des sites du e-commerce.

1.4 Grandes familles de filtrage d'information

1.4.1 Filtrage basé contenu

Un système qui utilise le filtrage basé contenu exploite seulement les représentations des documents et les informations qui peuvent être dérivées de ces documents. Un tel type de filtrage pourrait par exemple utiliser la similarité des documents dans une matrice termes-documents pour déterminer la pertinence d'un document. Si un utilisateur exprime un intérêt pour un document, les documents similaires seront jugés potentiellement pertinents aussi.

Apports de filtrage basé contenu

Pour recommander des items en se basant sur le contenu, deux ensembles doivent être constitués : les profils des items et les profils des utilisateurs. La notion de contenu ne se rapporte donc pas uniquement au contenu des items, mais également aux attributs descriptifs des utilisateurs. Une approche basée contenu analyse un ensemble d'items précédemment notés ou consultés par un utilisateur, et construit un modèle ou un profil des intérêts de l'utilisateur sur la base des caractéristiques des items aimés ou détestés par celui-ci. En fonction de ses feedbacks, le profil de l'utilisateur est construit et souvent constitué d'un profil "positif" représentant les items qu'il a aimés et d'un profil "négatif" représentant les items qu'il a détestés.

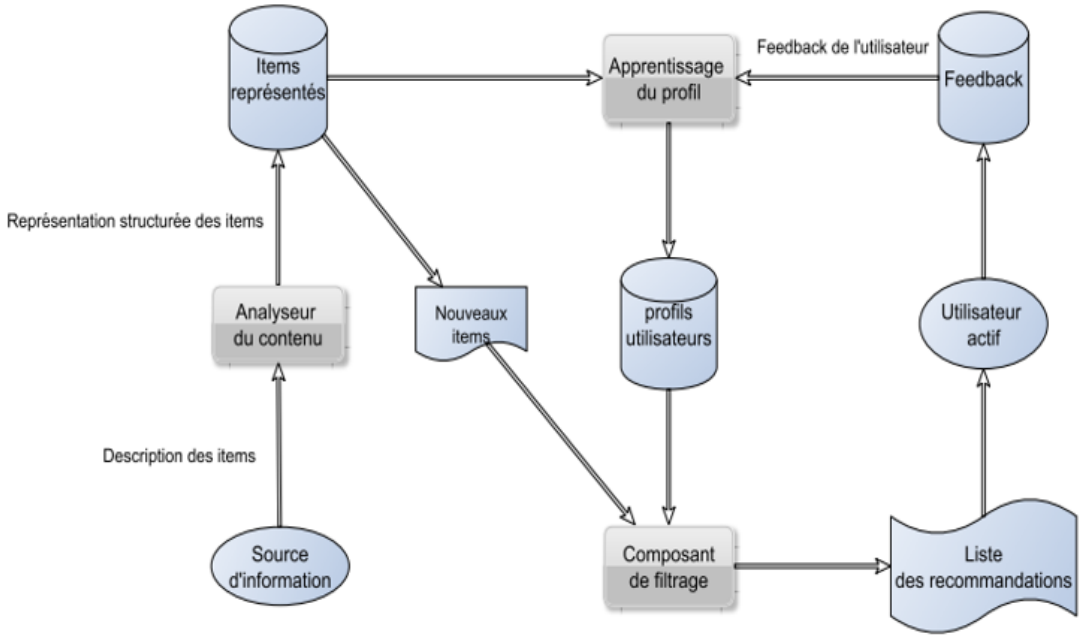

FIGURE 1.2 – Une architecture haut niveau d'un SR basé sur le contenu

Le processus de recommandation consiste donc essentiellement à comparer les attributs des items candidats avec les attributs du profil "positif" et "négatif" de l'utilisateur. De ce fait, les items qui seront recommandés à l'utilisateur sont les items qui

sont similaires à son profil "positif" et moins similaires à son profil "négatif ". Plus le profil de l'utilisateur construit reflète les préférences de l'utilisateur, plus le système de recommandation peut être efficace.

Un système de recommandation basé sur le contenu a besoin de techniques pour produire une représentation efficace des items et du profil de l'utilisateur pour pouvoir les comparer. Ainsi [43] proposent une architecture de haut niveau (figure 1.2) dans laquelle le processus de recommandation est réalisé en trois étapes, chacune étant gérée par un composant spécifique :

— **Analyseur du contenu**

Lorsque l'information n'est pas structurée (par exemple, un item représenté par un texte), ce module a pour but d'en réaliser le pré-traitement pour extraire l'information pertinente, la structurer et la représenter dans une forme cible appropriée (par exemple un vecteur de mots clés).

— **Apprentissage du profil**

Ce module collecte les données représentatives des préférences de l'utilisateur et généralise ces données, afin d'apprendre et de construire le profil de l'utilisateur. Des techniques d'apprentissage automatique [44] peuvent être utilisées pour cela. On peut citer à titre d'exemple les arbres de décisions, les réseaux de neurones et la classification naïve de Bayes. Ces techniques visent à inférer un profil de l'utilisateur en utilisant l'information sur les items qu'il a aimés ou n'a pas aimés.

— **Composant de filtrage**

Ce module filtre les items pertinents en faisant correspondre la représentation du profil utilisateur aux items candidats à la recommandation.

La pertinence de l'item est calculée en utilisant des métriques de similarité entre l'item considéré et le profil de l'utilisateur. Plus la similarité avec le profil "positif" est grande et plus la similarité avec le profil "négatif" est petite, plus l'item a des chances d'être recommandé.

Afin de construire et mettre à jour le profil de l'utilisateur actif, ses réactions aux items (notes) sont recueillies et enregistrées dans le composant Feedback. Ces notes d'intérêt sont exploitées au cours du processus d'apprentissage du modèle utile pour prédire la pertinence a priori d'un item que l'utilisateur n'a pas encore noté. Les utilisateurs peuvent aussi définir explicitement leurs domaines d'intérêt au préalable comme profil initial, mais ce cas est assez rare.

Limites de filtrage basé contenu

Pour conclure, le filtrage basé sur le contenu présente un certain nombre de limites. Tout d'abord, nous pouvons souligner la difficulté à indexer les documents multimédias.

En effet, le profil utilisateur peut prendre diverses formes, mais il demeure toujours composé par des mots. Ces derniers seront comparés aux mots qui composent le document. De ce fait, il est impossible d'indexer des documents multimédias. En outre, le fait que cette approche se base uniquement sur les profils thématiques pour caractériser les utilisateurs, pose également un problème. En effet, le filtrage basé sur le contenu n'intègre pas d'autres critères de pertinence que le profil thématique. Pourtant, il existe de nombreux autres aspects comme les informations démographiques (âge, position géographique, emploi, etc.), la qualité, la confiance, etc. De plus, l'effet dit d'« entonnoir» restreint le champ de vision des utilisateurs. En effet, le profil évolue toujours dans le sens d'une expression du besoin de plus en plus spécifique, qui ne laisse pas de place à des documents pourtant proches, mais dont la description thématique diffère. Par exemple, lorsqu'un nouvel axe de recherche apparaît dans un domaine, avec de nouveaux termes, ces derniers n'apparaissent pas dans le profil. Ainsi, les documents, portant sur cet axe de recherche, seront éliminés automatiquement par le système de filtrage. De ce fait, l'utilisateur n'aura donc jamais l'occasion d'exprimer un retour de pertinence positif sur ce nouvel axe de recherche.

Enfin, la masse critique est une autre limite du filtrage basé sur le contenu. En effet, il faut un certain nombre de documents pour que le système commence à performer [37][38].

1.4.2 Filtrage collaboratif

Plusieurs techniques ont été développées et un ensemble de modèles ont été proposés dans la littérature. Ce paragraphe fait le point sur l'état de l'art du filtrage d'information en général et plus précisément le filtrage d'information collaboratif.

Contexte et définitions

L'hypothèse principale sur laquelle repose le filtrage collaboratif est la bouche à oreille. Plus précisément, les gens à la recherche d'information devraient pouvoir bénéficier de ce que d'autres utilisateurs ont déjà trouvé et évalué. Par exemple, les personnes, qui veulent regarder un film ou lire un livre, demandent à leurs amis leurs opinions. Donc, dans le filtrage collaboratif, la sélection des documents à proposer à un utilisateur ne dépend plus des termes constituant le document (filtrage basé sur le contenu), mais des évaluations faites par les membres de son voisinage. Ainsi, si deux utilisateurs Alice et Bob ont évalué un certain nombre de documents de façon similaire, il y a de fortes chances qu'Alice aime ce que Bob aime, et inversement. Donc les documents que Alice a aimés peuvent être recommandés à Bob et inversement. De la sorte, cette approche résout une difficulté importante rencontrée par l'approche de filtrage basé sur le contenu, à savoir le traitement de documents multimédias [37].

La figure 1.3 illustre l'algorithme général d'un système de filtrage collaboratif. Typiquement, les étapes de cet algorithme sont les suivantes [37][38] :

- Collecter les appréciations de l'utilisateur sur les documents qu'i1 consulte ;
- Intégrer ces informations dans le profil de l'utilisateur ;
- Utiliser ce profil pour aider l'utilisateur dans ces prochaines recherches d'information.

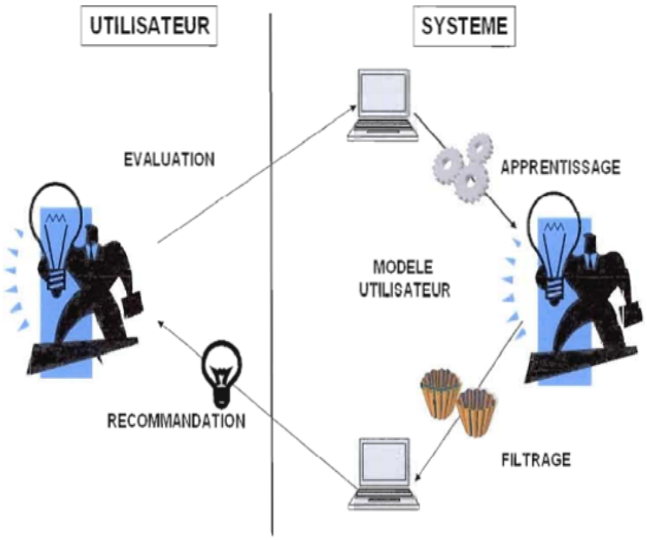

FIGURE 1.3 – Modèle général pour le filtrage collaboratif de l'information

À la base du filtrage collaboratif, on utilise les choix explicites ou explicites des utilisateurs pour des items. Donc plutôt que d'utiliser une matrice termes-documents comme la recherche d'information le fait et comme l'approche basée-contenu, le filtrage collaboratif utilise la matrice des votes utilisateurs-items, où un vote peut être soit explicite (ex. fournir une cote à un film), soit explicite (ex. acheter un DVD du film).

Le tableau illustre un exemple de matrice-utilisateurs-items. Les valeurs peuvent représenter des votes ou des comportements. Dans cet exemple, l'item I_3 n'a pas de vote pour l'utilisateur U_1. On peut tenter de l'estimer soit avec une approche item-item ou une approche utilisateur-utilisateur comme expliqué dans les sections qui suivent.

	I_1	I_2	I_3	I_4
U_1	5	1	?	2
U_2	4	1	0	3
U_3	4	2	1	2
U_4	1	4	3	2

TABLE 1.1 – Matrice Items-utilisateurs

Pour mieux définir ce type de filtrage on se réfère aux travaux de [45]. Ils décrivent plusieurs algorithmes conçus pour cette tâche, comprenaient des techniques basées sur

les coefficients de corrélation, le calcul de similarité basée sur les vecteurs et méthodes bayésiennes statistiques. Le filtrage collaboratif peut prendre plusieurs formes : Item-Item et Utilisateur-Utilisateur.

Architecture générale

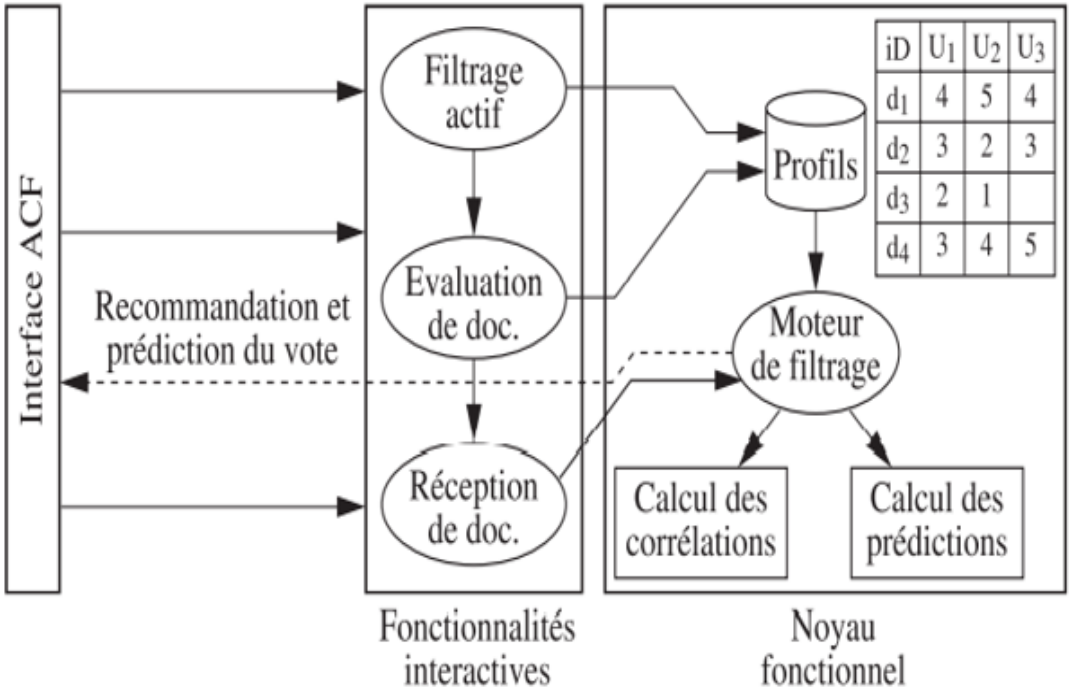

FIGURE 1.4 – Architecture générale d'un système de filtrage collaboratif

Approches d'un SFC

Les approches de filtrage collaboratif sont généralement classifiées en tant qu'algorithmes basés sur les modèles ou algorithmes basés sur la mémoire. Cependant, [46] propose un autre type de classification de ces algorithmes. Dans cette classification, il y a deux types d'algorithmes. D'une part, les algorithmes basés sur les utilisateurs prédisent l'évaluation d'un utilisateur donné sur un article donné en se basant sur l'information des évaluations pour des profils d'utilisateur similaires [45]. D'autre part, les algorithmes basés sur les articles fonctionnent selon le même principe, mais utilisent la similarité entre articles au lieu de la similarité entre utilisateurs [47].

— **Filtrage basé mémoire :** Les algorithmes basés mémoire [80][95][96] utilisent la totalité ou une partie des profils utilisateurs afin de générer une nouvelle prédiction. De tels algorithmes ont l'avantage d'être simples à mettre en œuvre et d'évoluer dynamiquement en fonction des profils utilisateurs. En effet, toute évolution d'un utilisateur se répercute directement dans le calcul de prédiction.

Cependant, ces algorithmes souffrent de deux inconvénients majeurs. D'une part, la forte complexité combinatoire empêche le passage à l'échelle pour un nombre important d'utilisateurs et de ressources. D'autre part, le faible nombre de ressources communément évaluées par les utilisateurs engendre des prédictions peu pertinentes [60].

— **Filtrage basé modèle :** le filtrage basé modèle apprend un modèle descriptif liant les utilisateurs, les documents et les votes d'un point de vue probabiliste, Pour estimer cette probabilité, [81] ont proposé l'utilisation de deux modèles : un modèle de clusters et un modèle réseaux bayesiens. Le modèle à base de clusters repose sur le principe que certains groupes ou types d'utilisateurs capturent un ensemble commun de préférences et de goûts. Dans le modèle à base de réseaux bayesiens proposé par [81] les nœuds correspondent aux documents. Les états pour chaque nœud correspondent aux valeurs d'évaluation possibles, le résultat est alors sous forme d'arbres de décision représentant chaque table de probabilité conditionnelle pour chaque nœud.

Limites de filtrage collaboratif

Pour conclure, le filtrage collaboratif présente un certain nombre de limites. En premier lieu, nous pouvons évoquer le problème lié à la taille de la population d'utilisateurs et de l'ensemble de documents à évaluer (le problème de la masse critique). En effet, il faut un certain nombre d'évaluations avant que le système puisse donner des résultats. De plus, cette approche souffre aussi du problème de démarrage à froid. En effet, les nouveaux utilisateurs commencent par un profil vide. Ainsi, une période d'apprentissage est nécessaire avant que le profil ne reflète concrètement les préférences de l'utilisateur. Pendant cette période, le système ne peut pas filtrer efficacement. Enfin, comme pour le filtrage basé sur le contenu, cette approche n'intègre pas d'autres aspects de pertinence capables d'améliorer le filtrage [37][38].

1.4.3 Filtrage hybride

L'approche de filtrage hybride repose sur l'idée de tirer profit des avantages des deux approches précédentes, en résolvant les problèmes qui leur sont liés. En fait, ces deux approches paraissent complémentaires. Les chercheurs du domaine estiment que le fait de combiner les deux méthodes pourrait être très bénéfique. D'où l'émergence de plusieurs techniques d'hybridation dont l'objectif consiste à combiner les deux approches (filtrage collaboratif et filtrage basé sur le contenu) de manière efficace [37][38].
Les récents travaux dans ce domaine visent à développer des algorithmes hybrides de plus en plus efficaces. Les auteures [38] décrit sept différents types de méthodes d'hybridation, présentées dans le tableau Cependant, selon [37], toutes ces méthodes se basent sur deux approches principales. La première approche repose sur des méthodes basées sur la mémoire pour générer les recommandations. La seconde approche utilise des méthodes basées sur les modèles pour déterminer les recommandations.

Méthode d'hybridation	Description
Pondérée	Les Résultats pondérés de plusieurs techniquesde recommandation sont combinés pourproduire une nouvelle recommandation.
Permutation	Le système permute entre les différentes techniques de recommandation selon le résultat de la recommandation.
Mixte	Les recommandations de plusieurs techniques sont présentées en même temps.
Combinaison	Différentes techniques de recommandation sont combinées en un unique algorithme de recommandation.
En cascade	Un système de recommandation raffine les résultats fournis par un autre système
Augmentation	Le résultat (« output ») d'une technique de recommandation est utilisé comme données en entrée («input ») pour l'autre technique.
Méta-niveau	Le modèle appris par une technique de recommandation est utilisé comme données en entrée (« input ») pour l'autre technique.

TABLE 1.2 −Méthodes d'hybridation

1.4.4 Filtrage social

Les données issues des réseaux sociaux représentent une véritable mine d'information pour un éventuel système de recommandation. Fonctionnant sur l'adage "Dis-moi qui tu fréquentes, je te dirai qui tu es", nous pouvons identifier un nouveau type de systèmes de recommandation basé sur la présence d'une communauté d'utilisateurs liés par des liens sociaux [48]. Sur les plateformes sociales, ces systèmes de recommandation permettent de recommander tout un ensemble d'informations. Les exemples incluent des utilisateurs à suivre, des publications précises, des éléments multimédias, des groupes (sous-communautés) à intégrer etc.

La principale caractéristique des réseaux sociaux étant l'existence d'un graphe de relations sociales, cette donnée est l'information principale au centre des diverses stratégies de recommandation. Les auteurs [49] présentent une étude qui compare diverses méthodes de prédiction de liens. La prédiction de liens consiste à analyser l'état du réseau à un instant t afin d'anticiper la création de nouveaux liens à un moment t + 1. Ces techniques sont souvent utilisées dans le contexte de la recommandation. Les méthodes présentées par [41] se basent sur les propriétés topologiques des réseaux pour le calcul des prédictions.

Parmi les mesures présentées, la mesure de Katz présente des résultats pertinents une fois exploitée sur des graphes issus du Web. Cette mesure, issue du monde de la sociologie [50], se base sur la connectivité entre deux nœuds du graphe social. Plus le

nombre de chemins entre deux nœuds est élevé et plus la longueur de ces chemins est courte, plus le score de Katz entre ces deux nœuds sera élevé. Concrètement, le score de Katz entre un nœud u et un nœud v s'exprime comme suit :

$$katz_{\beta}(u,v) = \sum_{l=1}^{\infty} \beta^{l} \times \left|P_{u,v}^{(l)}\right| = \sum_{p \in P_{u,v}} \beta^{|p|}$$

Où β représente un facteur de décroissance (β [0, 1]), $P_{u,v}$ représente l'ensemble de tous les chemins existant entre u et v et $\rho_{u,v}^{\langle l \rangle} \subseteq \rho_{u,v}$ l'ensemble de tous les chemins de longueur égale à l'existant entre u et v. Le facteur β est utilisé pour donner plus d'importance aux chemins courts (c.à.d aux nœuds proches dans le graphe) exploitant ainsi le phénomène de localité et d'homophilie.

Une autre mesure topologique est présentée dans [51], les auteurs proposent de combiner deux scores de classement de comptes basés sur deux sources différentes issues du même réseau (les invitations sur le réseau social, ainsi que le graphe social proprement dit). Ces données sont ensuite utilisées pour produire la liste triée des comptes les plus influents sur le réseau.

Certains travaux appliquent des méthodes de filtrage collaboratif ainsi que des méthodes basées sur le contenu dans un contexte social. L'étude [52] introduit une méthode pour classer des Tweets qui exploite les profils de préférences utilisateur, une mesure d'autorité du compte publiant le tweet ainsi que la qualité de la publication. La recherche [53] passe en revue une série de méthodes d'extraction de profils utilisateur sur le réseau Twitter. Il y est testé des méthodes basées sur le contenu publié par l'utilisateur, celui des comptes qu'il suit ainsi que celui de ses suiveurs. Un classement basé sur le score de TF-IDF est ensuite employé pour trouver les utilisateurs similaires. Une méthode similaire de recommandation est adoptée par [81].

Dans [54] les auteurs décrivent une approche qui garde trace des interactions passées par les utilisateurs pour le calcul en temps-réel des recommandations. Les techniques présentées par [54] et [53] fournissent des recommandations au niveau de granularité du tweet. Le passage à l'échelle est alors problématique étant donné le nombre important de tweets, l'aspect temps-réel de la plateforme et les fréquences élevées de publication.

Les approches citées précédemment ne prennent pas en considération la topologie du graphe dans le calcul de l'étude. L'auteur [46] présente une adaptation de l'algorithme Page Rank au monde du micro-blogging appelée TwitterRank. Cette approche prend en compte la structure des liens du graphe ainsi que l'autorité des utilisateurs sur des sujets (topics) donnés. Les topics utilisés par TwitterRank sont obtenus par l'application de la méthode LDA (Allocation de Dirichlet Latente) qui est une technique probabiliste permettant la caractérisation du contenu des utilisateurs. L'auteur [55] propose une méthode basée sur le contenu afin de fournir des recommandations de sujets (topics) en exploitant les liens implicites issus des conventions adoptées sur les plateformes de micro-blogging. Le chercheure [56] propose une technique de recommandation de hashtags personnalisée. Certains hashtags ayant un cycle de vie extrêmement court, les auteurs proposent de combiner le contenu des tweets avec une approche de filtrage

collaboratif sur une fenêtre temporelle d'un mois afin de recommander des hashtags pertinents aux utilisateurs.

Une approche présentée par[57] vise à maximiser la découverte de nouveau contenu lors de la tâche de recommandation. Ce problème s'apparente à un problème d'optimisation multi-objectif NP-difficile. Les auteurs proposent une approximation qui atteint un degré de propagation de contenu important.

Dans [59], les auteurs présentent le système de recommandation mis en production par Twitter pour sa fonction "Qui suivre ?" (Who to follow ?). Il se base sur le déploiement de l'algorithme SALSA [58] dans un environnement centralisé. SALSA fonctionne en créant un graphe biparti avec d'un côté le cercle de confiance d'un utilisateur (pouvant être calculé par exemple comme l'ensemble les comptes avec qui l'utilisateur interagit le plus ou bien à partir d'un algorithme de marche aléatoire) et de l'autre côté les comptes les plus suivis par ce cercle de confiance, considéré comme des autorités. Cette approche ne prend pas en considération le sujet (topic) sur lequel les comptes recommandés peuvent être une autorité.

1.4.5 Filtrage personnalisé et adaptatif

Depuis les années 60, le filtrage adaptatif a suscité un développement sans précédent. Ce développement du filtrage adaptatif est né de l'essor du traitement numérique, de la croissance soutenue de la puissance des processeurs de traitement qui permettent la mise en œuvre en temps réel et d'algorithmes de plus en plus complexes et qui vont à des cadences de plus en plus élevées.

Un filtre adaptatif est, par définition, un filtre numérique dont les coefficients estimés au sens d'un critère donné, s'adaptent aux variations des signaux reçus. Habituellement, un vecteur d'entrée et une réponse désirée sont utilisés pour définir un vecteur d'erreur qui contrôle alors l'évolution des paramètres du filtre adaptatif.

Dans la figure 1.5 on peut voir un schéma simplifié d'un filtre adaptatif, où d(n) représente le signal désiré (référence), y(n) : la sortie du filtre numérique (y(n)=x(n)*h(n)) et e(n) : le signal d'erreur.

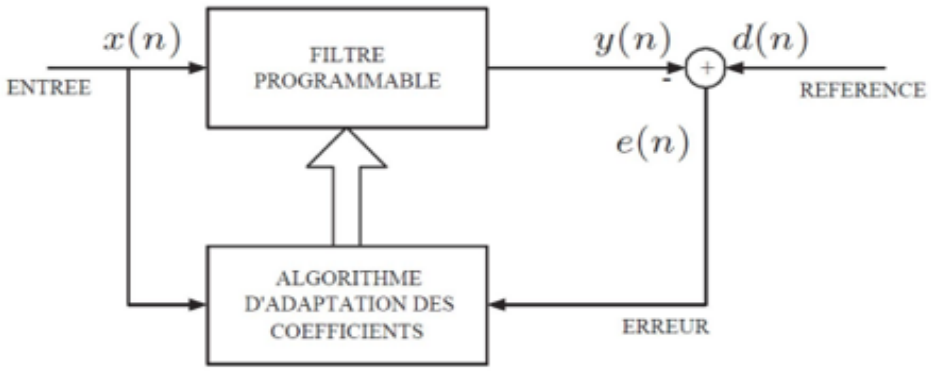

FIGURE 1.5 – Principe d'un filtre adaptatif

Comme le montre la Figure 1.5, un filtre adaptatif est un filtre numérique avec des coefficients qui sont déterminés et mis à jour par un algorithme adaptatif. L'algorithme adaptatif se comporte comme un opérateur humain qui a la capacité de s'adapter à un environnement changeant.

Les filtres adaptatifs peuvent être classés en fonction des choix qui sont faits sur les points suivants :

— Le critère d'optimisation,
— L'algorithme de mise à jour des coefficients,
— La structure du filtre programmable,
— Le type de signal traité, mono ou multidimensionnel.

1.5 Quelques systèmes de filtrage

1.5.1 Tapestry

Le concept du filtrage collaboratif a été lancé avec le projet Tapestry à Xerox Parc. La gestion des e-mails est sa motivation première [33].

Tapestry repose sur une « recommandation commentée » basé sur des annotations de qualité ou d'appréciation des documents faites par les utilisateurs. De cette manière, les documents sont filtrés en fonction de ces annotations [60] L'implication de l'utilisateur n'est pas limitée à fournir de simples jugements binaires d'acceptation ou de rejet [61]. Il donne la possibilité de faire des annotations en texte libre ou des appréciations dans le style « J'ai bien aimé » ou « Je déteste », ainsi les utilisateurs peuvent transmettre des jugements sur la valeur des documents qu'ils lisent. Les autres utilisateurs peuvent alors opérer des recherches parmi ces documents non seulement sur la base de leur contenu, mais également sur la base des jugements qu'ont portés d'autres utilisateurs à leur sujet. Tapestry a aussi introduit la prise en compte de la confiance dans la source de l'information.

Le système a souffert de deux problèmes [62]. Le premier est la taille de sa base d'utilisateurs. Puisque Tapestry est basée sur un système commercial de base de données, il ne peut être fourni librement. De plus, il n'a pas été conçu pour l'usage d'un grand nombre de personnes géographiquement distribuées. Ces deux facteurs se combinent pour limiter la population d'utilisateurs potentiels aux chercheurs à Xerox Parc. Cependant, cette population ne semblait pas assez grande pour constituer une masse critique d'utilisateurs et la grande majorité des documents passaient sans annotations. Ainsi le système souffrait d'un manque d'informations pour pouvoir fonctionner normalement.

Le deuxième problème avec Tapestry est le moyen par lequel les utilisateurs interagissent avec les filtres. Une interface commune exigeait des utilisateurs d'indiquer des requêtes en un langage dérivé de SQL. Cette forme d'interface a été un obstacle à l'exploration de nouveaux secteurs et a rendu difficile la visualisation de l'information disponible. Il n'en demeure pas moins que Tapestry fut un des premiers systèmes de filtrage existants [38].

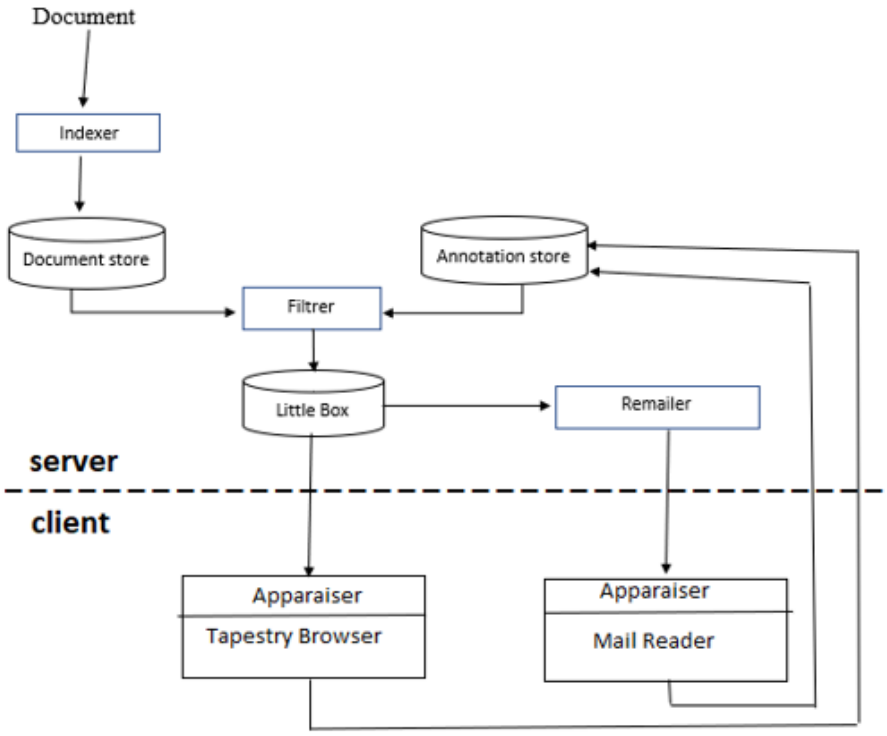

FIGURE 1.6 – Le Flux des Documents de Tapestry

1.5.2 GroupLens

GroupLens [34] [63], est un système expérimental de l'université du Minnesota, il est un des plus célèbres et solides dans ce domaine. Il est semblable dans son esprit à Tapestry : les lecteurs sont appelés à noter les articles qu'ils lisent sur une échelle numérique de cinq niveaux. Le système trouve alors des corrélations entre les différents utilisateurs et identifie des groupes d'utilisateurs dont les intérêts sont semblables. Ensuite, il emploie ces estimations pour prédire l'intérêt que porteront les utilisateurs à chaque article.

GroupLens prolonge Tapestry de deux manières [34] : d'abord, Tapestry est conçu pour partager des évaluations dans un même lieu. Avec GroupLens, les estimations sont réparties en plusieurs emplacements et son architecture est ouverte à la création de nouveaux clients de newsgroups. En second lieu, Tapestry ne supporte pas de requêtes globales. Les serveurs d'estimation qui ont été mis en place pour GroupLens prennent en considération les estimations globales de plusieurs experts, basées sur la corrélation de leurs estimations passées. Un lecteur n'a pas besoin de voir à l'avance les évaluations à employer et n'a pas besoin de savoir à qui les évaluations sont destinées réellement. Dans GroupLens, les estimations fournies sous un pseudonyme sont aussi utiles que celles qui sont signées. Pour son évaluation, la corrélation entre l'évaluation faite par le système et l'évaluation individuelle d'un utilisateur après la lecture d'un article, a été utilisée.

Cependant, en raison du grand nombre de différents documents, ce système dépend beaucoup du nombre de lecteurs et de leurs évaluations sur les mêmes documents [62]. De plus, il souffre d'un problème de démarrage à froid [63]. Beaucoup d'utilisateurs ont abandonné son utilisation ; ils avaient un grand nombre de documents à noter avant de commencer à recevoir des recommandations et donc à bénéficier du système (problème de motivation). En outre, les premiers utilisateurs ne recevaient pratiquement que des documents qu'ils avaient déjà lus et notés, en raison de la lenteur de l'apprentissage.

1.5.3 Le système de Malltz et Ehrlich

Ce système [62] est basé sur l'hypothèse que les utilisateurs recherchant l'information devraient pouvoir se servir de ce que d'autres ont déjà trouvé et évalué.

Une pratique courante chez les utilisateurs est d'utiliser l'e-mail pour envoyer des pointeurs sur des documents intéressants à des collègues ou des amis. Cependant, cette action requiert un effort relativement important de la part de l'expéditeur, et il arrive souvent que l'utilisateur n'envoie pas la référence à toutes les personnes qu'elle pourrait intéresser, ou qu'il oublie simplement de le faire.

Le système de Maltz et Ehrlich est présenté comme un substitut au mail dans ces situations. Il est intégré à un système de recherche d'information et permet à ses utilisateurs d'adresser des pointeurs aux personnes qu'ils jugent intéressées, sans avoir à interrompre leur session de recherche d'information. D'un autre côté, l'ensemble de ces échanges est stocké pour constituer une base de références [38].

1.5.4 CoCoFil

COCoFil1 (Community-Oriented Collaborative Filtering) est une plateforme de filtrage collaboratif autour de la notion de communauté. Elle comporte, outre l'espace destiné à l'utilisateur, un espace destiné à l'administration du système. Cet espace « administrateur » permet de gérer les divers paramètres du système, et d'intégrer des tableaux de bord destinés à surveiller l'état du système pendant son exploitation.

L'espace « utilisateur » est structuré selon les grandes familles d'activités que l'on peut y pratiquer. Ces activités sont conçues pour reprendre et étendre les fonctionnalités habituelles des outils courants.

La plateforme COCoFil comporte trois modules : Filtrage collaboratif, Paramétrage et Gestion de contact. Le module «Gestion de contact » assure l'identification dans la plateforme COCoFil, c'est-à-dire qu'il permet d'une part aux utilisateurs de saisir leurs informations personnelles, et d'autre part au système d'identifier l'utilisateur lors de son accès. Grâce à ce module, l'utilisateur peut en outre organiser son carnet d'adresses et échanger des recommandations avec d'autres utilisateurs dans le cadre du filtrage collaboratif actif.

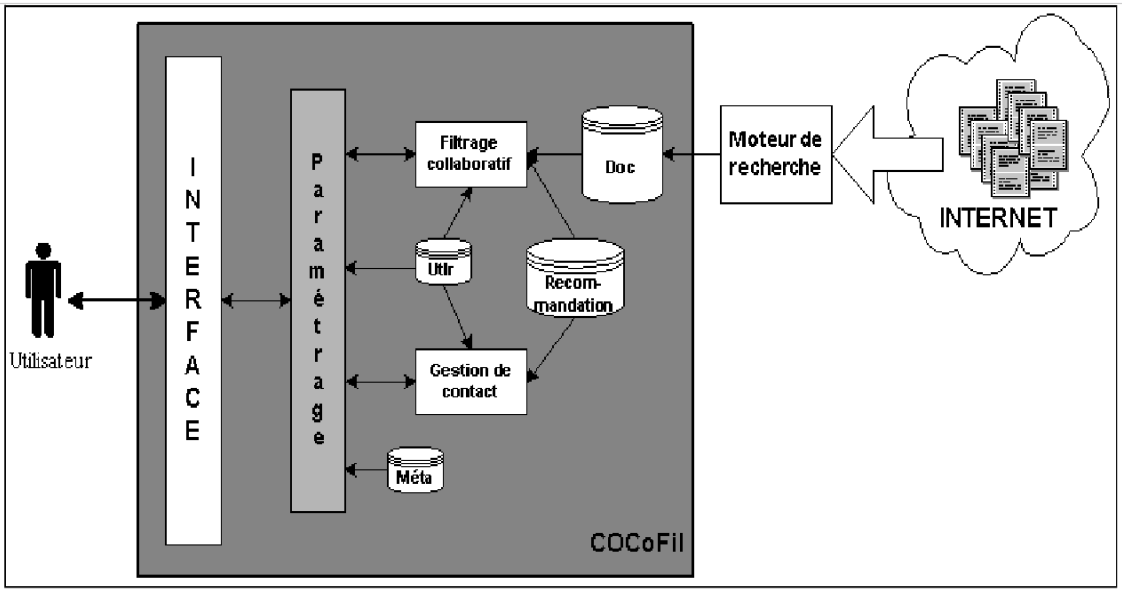

FIGURE 1.7 – Schéma des principales fonctionnalités de COCoFil

1.5.5 Amazon

Amazon est l'un des systèmes de vente en ligne des plus populaires. Il offre à ses utilisateurs la possibilité de personnalisation d'interface.

La compagnie d'Amazon.com gère une énorme matrice dans laquelle sont liés des millions de produits. Chaque inscription, chaque unité de cette matrice enregistrent en réalité le nombre d'acheteurs qui ont payé deux produits correspondants à cette unité. Ainsi, une recommandation proposée au clic de souris.

Amazon.com utilise l'algorithme « item-to-item collaborative filtering »[47]. Ce système commence par le calcul du degré de similarité entre articles en hors ligne « offline » construisant ainsi une table des similarités item-item. Cette étape est extrêmement gourmande en termes de temps de calcul. Ensuite, si l'utilisateur s'intéresse à un produit bien précis, le système lui recommande des produits similaires à celui si sur la base de la matrice des similarités des articles.

1.6 Conclusion

Les deux approches principales (filtrage basé sur le contenu et filtrage collaboratif) de filtrage d'information donnent des résultats très intéressants. Le filtrage collaboratif se propose de recommander aux utilisateurs certains articles qu'ils n'ont pas encore lus en se basant sur les opinions du groupe d'utilisateurs similaires. En revanche, le filtrage basé sur le contenu évalue si un texte est pertinent ou non pour un utilisateur, en fonction de son profil thématique. Néanmoins, ces deux aspects du profil semblent importants. Ainsi, une approche hybride combinant les goûts thématiques personnels

et les opinions des utilisateurs du groupe paraît la meilleure méthode pour fournir des recommandations.

Un autre aspect important est la collecte du degré d'appréciation des utilisateurs pour une ressource donnée. Afin d'obtenir un plus grand nombre de notes, une fonction d'´évaluation implicite basée sur la formule de Chan [64], en parallèle avec une fonction classique de collecte des évaluations (évaluation explicite), peut aussi être envisagée. Cette formule se charge d'estimer les évaluations que l'utilisateur est susceptible d'attribuer à une ressource donnée à partir de critères implicites. Cette méthode de collecte des préférences des utilisateurs permet de réduire le nombre d'évaluations manquantes et donc d'accroitre la qualité des prédictions du système de recommandation.

Chapitre 2

Interaction en graphe

2.1 Introduction

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 [1]. Les graphes sont partout, tout ce qui implique des relations (implicites ou explicites) peut-être modélisé sous forme de graphe, l'exploration de graphes est le processus de découverte, récupérer et analyser des modèles non triviaux dans des données en forme de graphique .

Dans ce chapitre, nous allons étudier les différents graphes. Dans un premier temps, nous allons nous intéresser à la définition de différents graphes, qu'ils soient simples, dirigés ou pondérés. À partir de là, nous présenterons trois des types les plus traitées dans les graphes. Ensuite, nous discuterons les caractéristiques des graphes (densité, voisinage, degré). On conclure par la détection de communauté dans les graphes et les approches, méthode les plus utilises dans ce dernier.

2.2 Concepts d'un graphe

Le graphe est un modèle mathématique abstrait d'un réseau

— Graphique Web, graphique social.

— Terminologie : graphe, sommet/nœud, arête.

La plupart des graphes obtenus lors de la modélisation des activités cités ci-avant exhibent des propriétés topologiques non-triviales mais similaires à d'autres graphes d'interactions observés dans d'autres contextes et d'autres domaines comme la biologie (ex. réseaux d'interactions entre protéines), les réseaux technologiques (ex. Internet, le web) [3], et les réseaux de transport (réseaux de routes, réseaux ferrés, réseau d'interconnexion entre aéroports) [4][15]. Nous désignons ces graphes, modélisant des systèmes réels, par le nom générique de graphes de terrain.

Un graphe est un ensemble de sommets (ou appelés nœuds) V (pour Vertices, en anglais) et d'arêtes notés E , $G = (V, E)$ où $V = \{ v_1, v_2, \ldots, v_n\}$ est l'ensemble de nœuds et $E = \{ e_1, e_2, \ldots, e_m\}$ est l'ensemble d'arêtes. L'arête $(v_i, v_j) \in E$ est dites incidente à v_i et v_j et ces nœuds sont appelés voisins ou adjacents. Un nœud $v$qui n'est adjacent à aucun autre nœud du graphe est appelé nœud isolé. On parle d'arête lorsque le graphe est non orienté et d'arcs lorsque le graphe est orienté. Un graphe est défini par son ordre qui représente son nombre de nœuds [6],[7]. La figure 2.1 présente un graphe d'ordre 4.

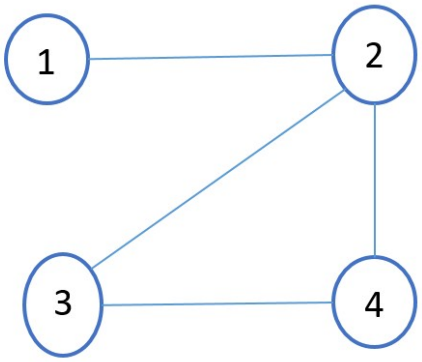

FIGURE 2.1 – Exemple d'un graphe d'ordre 4

A l'aide des graphes on peut compresser des graphiques sans perdre d'informations, trouver rapidement des structures complexes, reconnaître les communautés et les modèles sociaux, prédire si deux personnes deviendront amis, comprendre quels sont les nœuds importants, montrer comment le réseau va évoluer, aide à la visualisation de structures complexes, recherche de rôles, prédiction d'influence positive et négative [2].

2.3 Types de graphe

Il existe des graphes **orientés** (ou dirigés) et **non orientés** (non dirigés).

Un graphe **orienté** est un graphe pour lequel les arêtes sont orientées, ce qui n'est pas le cas pour un graphe non orienté.

Un graphe peut également être **pondéré** ou **value**, c'est-à-dire lorsqu'il existe une fonction

$W : e \in E \rightarrow R$ qui a chaque lien associe une valeur réelle. On note $G = (V, E, W)$ un graphe pondéré.

On appelle graphe **complet**, un graphe ou tous les sommets sont adjacents, c'est-a-dire si tout couple de sommets distincts est lié par une arête. Pour tout entier naturel n on note K_n le graphe complet d'ordre n .Le nombre d'arêtes du graphe complet K_n est égal à $\frac{n(n-1)}{2}$ On appelle clique un sous-graphe complet de G.

Un chemin du sommet s vers le sommet t dans un graphe orienté est une suite $(v_1, v_2, \ldots, v_k)$ de sommets telle que $v_0 = s$, $v_k = t$, $(v_{i-1}, v_i) \in E$, pour tout $1 \leq i \leq k$. Le terme k est appelé la longueur du chemin, et on dit que le sommet t est joignable à partir du sommet s. Le chemin est dit simple (ou élémentaire) si les v_i sont distinctes deux-a-deux (arêtes incidentes deux-a-deux).

La notion correspondante dans les graphes non orientés est celle de chaine. Dans un graphe non orienté, un cycle est une suite d'arêtes consécutives (chaine) dont les deux sommets extrémités sont identiques, c'est-à-dire tel que $v_0 = v_k$.

La distance géodésique entre deux sommets dans un graphe est définie par la longueur d'un plus court chemin entre ces deux sommets.

Un graphe est dit **connexe** si deux sommets quelconques peuvent être reliés par un chemin[8] .

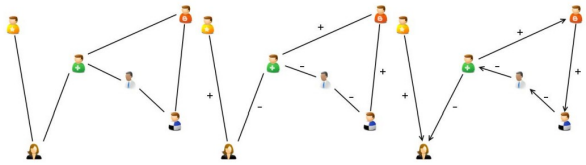

FIGURE 2.2 – Les trois différents types de graphes

Dans la figure 2.2 les graphes simples non orientés (à gauche), les graphes simples orientés (au centre) et les graphes pondérés orientés (à droite)

2.3.1 Graphes 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 : la 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 amis, familles, 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) [12].

Un graphe orienté G est la donnée d'un couple $G = (S, A)$ tel que :

- S est un ensemble fini de sommets

- A est un ensemble de couples ordonnés de sommets $(S_i, S_j) \in S^2$

Un couple (S_i, S_j) est appelé un arc, et est représenté graphiquement par $S_i \rightarrow S_j$, si est le sommet initial ou origine, et S_j le sommet terminal ou extrémité. L'arc $a = (S_i, S_j)$ est dit sortant en S_i et incident en S_j , et S_j est un successeur de S_i , tandis que S_i est un prédécesseur de S_j . L'ensemble des successeurs d'un sommet $S_i \in$ S *est noté* Succ(S_i) = $\{ S_j \in S, (S_i, S_j) \in A\}$. L'ensemble des prédécesseurs d'un sommet $S_i \in$ S *est noté* Pred(S_i) = $\{ S_j \in S, (S_j, S_i) \in A\}$ [9].

Par exemple :

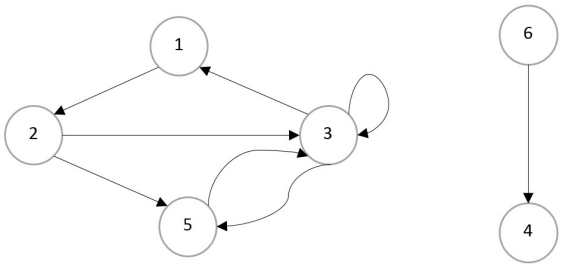

FIGURE 2.3 – Un graphe orienté

Le Graphe orienté sur la figure 2.3 : G = (S, A) avec S = 1, 2, 3, 4, 5, 6 et A = (1, 2),(2, 3),(2, 5),(3, 1),(3, 3),(3, 5),(5, 3),(6, 4).

2.3.2 Graphe valué

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 2.4 – Avis de Amazon sur les produits .

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 [12].

Un graphe valué $G = (S, A, v)$ est un graphe (S, A) (orienté ou non-orienté) muni d'une application $v : A \to R$. L'application v est appelée valuation du graphe. On peut étendre cette valuation en posant $\forall (x, y) \in S^2$, $v(x, y) = +\infty$ si $(x, y) \notin A$ [9].

2.3.3 Graphes et sous-graphes connexes

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) [12].

Un graphe non orienté est connexe si chaque sommet est accessible à partir de n'importe quel autre.

Autrement dit, si pour tout couple de sommets distincts $(s_i, s_j) \in S^2$, il existe une chaîne entre s_i et s_j .

Exemple : Par exemple, le graphe non orienté suivant :

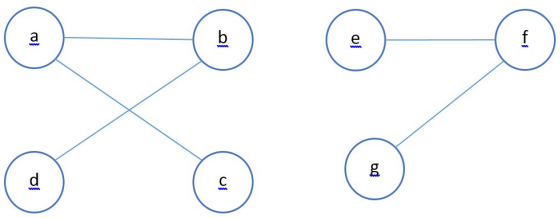

FIGURE 2.5 – Un graphe non orienté

n'est pas connexe car il n'existe pas de chaîne entre les sommets a et e. En revanche, le sous-graphe induit par les sommets $\{ a, b, c, d\}$ est connexe. Une composante connexe d'un graphe non-orienté G est un sous-graphe G_0 de G qui est connexe et maximal (c'est-à-dire qu'aucun autre sous-graphe connexe de G ne contient G_0). Un graphe est dit connexe si et seulement si il admet une unique composante connexe. Par exemple, le graphe précédent est composé de deux composantes connexes : la première est le sous-graphe induit par les sommets $\{ a, b, c, d\}$, et la seconde est le sous-graphe induit par les sommets $\{ e, f, g\}$ [9].

2.4 Caracteristiques des graphes

Nous présentons dans ce qui suit quelques propriétés des graphes.

— **Densité** : c'est le rapport entre le nombre d'arêtes observées et le nombre maximal d'arêtes possibles. Une densité égale à 0 veut dire que tous les nœuds sont

isolés et une densité égale à 1 veut dire qu'il existe un lien entre chaque paire de nœuds (graphe complet) [10] :

$$den(G) = \frac{2|E|}{|V| \cdot (|V| - 1)} \tag{2.1}$$

La densité du graphe de la figure 2.1 est :

$$den(G) = \frac{2 \times 4}{1 + (3 \times 2)} = 1.14$$

— **Voisinage** : dans un graphe $G = (V, E)$, le voisinage d'un nœud v_i est noté $\Gamma(v_i)$ tel que : $\Gamma(v_i) = \{v_j \in V | (v_i, v_j) \in E\}$. par exemple, les voisins du nœud 2 de la figure 2.1
sont les nœuds : 1, 3 et 4.

— **Le degré** : dans un graphe $G(V, E)$ le degré (v) d'un nœud v est donné par le nombre de ses voisins. Le degré moyen d'un graphe, noté $\bar{d}$ est défini par [11] :

$$\bar{d} = \frac{1}{|V|} \sum_{i \in v} d(i) \tag{2.2}$$

par exemple le degré du nœud 1 de la figure 2.1 est $d(1) = 1$.

2.5 Représentation d'un graphe

Il existe plusieurs structures permettant de représenter un graphe. L'une des structures les plus intuitives concerne la représentation matricielle. Un graphe peut être représenté par une matrice d'adjacence de taille $|V| \times |V|$ dont l'élément non-diagonal noté A_{ij} représente le nombre d'arêtes liant le nœud i au nœud j. La matrice d'adjacence est notée A. Dans un graphe simple (sans boucle), la diagonale de la matrice ne comprend que des zéros. Cette représentation permet d'avoir des informations sur la topologie du graphe et sur les relations entre paires de sommets. On peut par exemple, connaitre le nombre de chemins de longueur k entre deux nœuds i et j en élevant la matrice a la puissance k et en observant les éléments de la i^eme ligne et de la j^eme colonne de la matrice résultante.

Bien que la structure de matrice soit séduisante à utiliser, elle n'est pas pratique en programmation dans la mesure où elle demande un espace mémoire important même pour des machines modernes. Une représentation moins gourmande consiste à considérer une liste de voisins, nommée liste d'adjacence, notée $L_A = (l_{vi})_{i=1}^{n}$ ou l'élément l_{vi} est la liste des voisins du sommet v_i [8].

Matrice d'adjacence

Les graphes peuvent être définis par différentes matrices. Nous donnons ici la définition de la matrice d'adjacence.

La matrice d'adjacence A est une matrice carrée $n \times n$ pour un graphe $G = (V, E)$ non orienté, la matrice d'adjacence est symétrique où $A_i = 1$ si i et j sont voisins, 0 sinon

$$A_{i,j} = \begin{cases} 1, si\ i\ et\ j\ sont\ voisins \\ 0, sinon \end{cases}$$

La matrice d'adjacence du graphe donné à la figure 2.1 est :

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

2.6 Algorithmes de parcours d'un graphe

Un parcours de graphe est un chemin visitant chaque noeud du graphe. Un algorithme de parcours de graphe visite donc séquentiellement tous les noeuds du graphe.Il s'agit d'écrire un algorithme qui permet d'examiner les sommets une et une seule fois. la présence de circuits doit être prise en considération de façon à ne pas visiter plusieures fois le même sommet. Il faut donc marquer les sommets déja visités [14]. Le DFS (« Depth-First Search» , parcours en profondeur) ainsi que le BFS (« Breadth-First Search », parcours en largeur). Le DFS visite un noeud v puis choisit comme noeud à visiter par la suite l'un des voisins non visités de v. Si tous les voisins de v ont déjà étés visités, le DFS choisit l'un des voisins du noeud précédent, et ainsi de suite.

Le pseudo algorithme 1 présente le parcours DFS.

Algorithme 1 : DFS(G,s)

Données : Un graphe G et un noeud de départ s

```
1 début
2     Marque s comme visité
3     pour v ∈ voisins(s) faire
4         si v n'est pas marqué comme visité alors
5             DFS(G, v)
```

FIGURE 2.6 – Pseudo algorithme de DFS

Le BFS place d'abord le noeud d'origine dans une file . À chaque itération, le BFS va visiter le premier élément de la file puis placer tous ses voisins dans la file, s'ils n'y sont pas déjà.

Le pseudo algorithme 2 présente le parcours BFS.

Algorithme 2 : BFS(G,s)

Données : Un graphe G et un noeud de départ s

```
1 début
2     file ← (s)
3     tant que file ≠ () faire
4         v ← défile(file)
5         pour u ∈ voisins(v) faire
6             si u n'est pas marqué comme enfilé alors
7                 marque u comme enfilé
8                 enfile(file, u)
```

FIGURE 2.7 – Pseudo algorithme de BFS

Le BFS visite les noeuds par ordre de distance par rapport au noeud initial. Le DFS commence par créer un long chemin, puis revient sur ses pas quand il ne peut plus continuer. Un exemple de l'exécution de ces deux algorithmes est présenté Fig2.8[8].

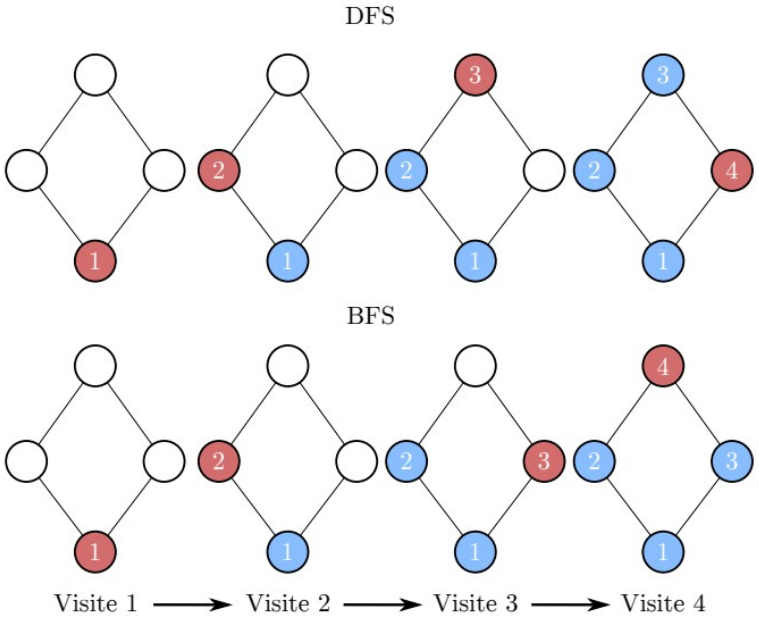

FIGURE 2.8 – État d'un graphe DFS et du BFS.

2.7 Détection de communautés dans un graphe

La détection de communautés est un domaine de recherche actif depuis ces vingt dernières années. De très nombreuses approches ont été mises en œuvre pour la détection de structures communautaires. Certaines méthodes considèrent le graphe dans son ensemble et effectuent une coupe pour trouver des communautés alors que d'autres privilégieront une approche nodale (c'est-à-dire, un partitionnement fondé sur les propriétés de nœuds voisins). L'ère du traitement de données massives, a également vu la naissance d'architectures parallèles et distribuées sur lesquelles se sont agrégés de nombreux algorithmes et des solutions à des problèmes multidisciplinaires.

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 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 [16] et ce sont les travaux de [17] 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 [18]. 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, [19] 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 2.9), 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.

L'objectif de la détection de communautés est de découvrir des groupes d'individus dans un réseau social. Ces groupes sont associés à des (clusters) dans le graphe.

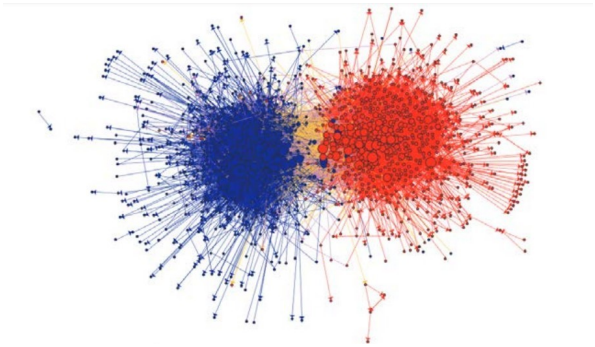

FIGURE 2.9 – Représentation de la structure du réseau de blogs

La ressemblance du problème d'identification de communautés avec beaucoup d'autres problèmes traités dans d'autres domaines, comme le clustering de données, le problème de calcul de cut dans des graphes ou encore les problèmes d'optimisation font qu'il existe une grande variété d'approches pour l'identification de communautés. Trois études de synthèse intéressantes mais non-exhaustives sont présentées dans [20],[22],[87]. Ici, nous proposons de classifier les approches existantes dans quatre classes non exclusives entre elles :

-Approches centrées groupes ou des nœuds sont regroupés en communautés en fonction de propriétés topologiques partagées.

-Approches centrées réseau ou la structure globale du réseau est examinée pour la décomposition du graphe en communautés.

-Approches centrées propagation qui appliquent souvent une procédure d'émergence de la structure communautaire par échange de messages entre nœuds voisins.

-Approches centrées graines ou la structure communautaire est construite autour d'un ensemble de nœuds choisis d'une manière informée.

2.7.1 Approches centrées groupes

Le principe consiste a confondre la définition d'une communauté avec un groupe de nœuds ayant certaines caractéristiques topologiques communes. L'exemple le plus trivial est d'assimiler une communauté a une clique maximale dans le graphe ou à une γ-dense quasi clique. Une clique est un sous-graphe complet. Une clique est maximale si on ne peut l'étendre en ajoutant de nouveaux nœuds. Une γ-dense quasi clique est un sous-graphe dont la densité est supérieur a un certain seuil $\gamma \in [0, 1]$. Or, le problème de calcul de cliques maximales est un problème NP- difficile, ce qui rends difficile d'envisager son utilisation dans le contexte de très grands graphes. Une autre concept utile, souvent employé dans le domaine de l'analyse des réseaux sociaux, est le concept de K-core. Un K-core est un sous-graphe connexe maximal dans lequel le degré de chaque nœud est supérieur ou égale a k. les graphes de terrain sont principalement des graphes très parcimonieux, de telle structures sont souvent minoritaire dans les graphes. Par contre de groupements denses de nœuds peuvent servir comme des graines pour la détection des communautés.

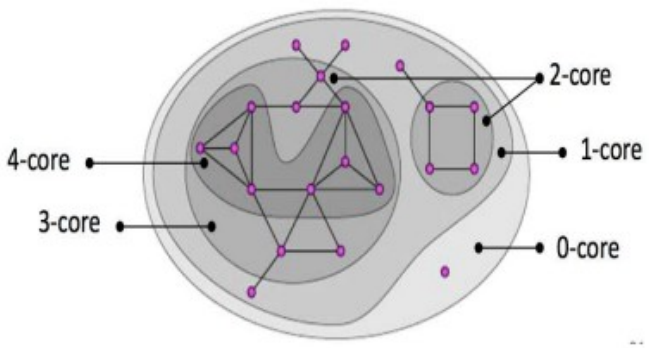

FIGURE 2.10 – Exemple de K-core dans un graphe

2.7.2 Approches centrées réseau

Les majeures parties des approches proposées dans la littérature s'appuient sur un schéma de calcul prenant en compte la connexion globale du graphe cible. Différentes approches ont été proposées. Nous reprenons dans la suite la classification proposée dans [22] des approches centrées réseau ou on distingue trois familles d'approches :

Approches de clustering

Un « cluster » est un ensemble de noeuds non vide. Une approche simple pour la détection de communautés consiste a transformer ce problème en problème classique de clustering de données [23]. Etant donnée n individus a regrouper en clusters, beaucoup d'algorithmes classiques calculent d'abord une matrice de similarité S de dimension $n \times n$ ou un élément S_{ij} exprime la similarité entre deux individus i et j selon une mesure de similarité donnée. Dans le cas d'un graphe G de n nœuds il est aussi possible de construire une matrice de similarité entre les nœuds du graphe en utilisant une mesure de similarité topologique entre les nœuds du graphe. Différentes mesures de similarité topologiques dyadiques peuvent être définies. Nous les classifions en trois grandes catégories :

– Les mesures basées sur le voisinage des nœuds, dites aussi mesures locales.

– Les mesures basées sur les chemins entre les nœuds, dites aussi mesures globales.

– Les mesures semi-locales.

Mesures basées sur les chemins, principalement on cite :

- **La proximité** $sim^{proxi}(x,y) = \frac{1}{dixt(x,y)}$: Plus la distance géodésique entre deux nœuds est petite plus la proximité des deux nœuds est grande. Or, rappelons qu'une caractéristique phare des graphes de terrain est la faible degré de séparation. Autrement dit, la distance moyenne entre chaque couple de nœuds est faible. Ce qui rends une telle mesure peu discriminant dans beaucoup de situations.

- **La mesure de Katz** : soit le $\sigma^l(x,y)$ l'ensemble de chemins de longueur l reliant deux nœuds x et y. La mesure de Katz proposée initialement dans [24] est définie par :

$$(sim)^{katz}(x,y) = \sum_{l=1}^{\infty} \beta \times \|\sigma^l(x,y)\|$$

ou $\beta << 1$ est un facteur qui va favoriser la prise en compte des chemins de longueurs courtes. Dans [25] on montre que si β est inférieur a la plus grande valeur propre de A_G alors le calcul de cette mesure pour chaque couple de nœuds converge pour les valeurs calculées par la formule matricielle suivante :

$$sim^{katz} = (I - \beta \times A_G)^{-1} - I$$

Ou I est la matrice identité. Le calcul de cette mesure est très couteuse pour les grands graphes. En pratique nous nous contentons d'une formule simplifiée comme nous le montrons lors de la discussion des mesures semi-locales.

2.7.3 Approches centrées propagation

Les approches centrés propagation exploitent la propriété de la densité des liens intra-communauté. En effet, en raison de la densité relative des communautés et des faibles liens intercommunautaire, on peut raisonnablement admettre qu'un signal émis par un nœud et retransmis par ses voisins a plus de chance de rester dans la communauté du nœud source, que de se propager aux autres communautés. Différents algorithmes exploitent cette propriété différemment. Par exemple, l'algorithme WalkTrap [26] calcule pour chaque nœud dans le graphe un vecteur qui donne la probabilité qu'un marcheur aléatoire arrive aux autres nœuds du réseau. Les vecteurs de probabilité ainsi calculés pour chaque nœud sont utilisés pour calculer des similarités entre les nœuds. D'autres algorithmes centrés propagation sont les algorithmes basés sur les techniques de propagation de labels [27],[86],[88],[89],[90],[91].

2.7.4 Approches centrées graines

Le schéma général d'une approche centrée graine est structuré en deux étapes :

- Déterminer un ensemble de nœuds ou groupes de nœuds dans le graphe qu'on désigne par des graines et qui constituent en quelque sorte les centres de communautés a retrouver.

- Appliquer une procédure d'expansion autour des graines afin d'identifier les communautés dans le réseau.

Différentes heuristiques de choix de graine ont été proposées. Une graine peut être composé d'un seul nœud sélectionné en utilisant les mesures classiques de centralité comme c'est fait dans [28]. Dans d'autres algorithmes la graine est composé d'un ensemble de nœuds qui ont une certaine connectivité [13].

Différentes stratégie d'expansion des graines sont aussi proposées. Dans beaucoup d'algorithmes on utilise les heuristiques développés pour l'identification de communautés locales [29],[93],[94]. Ces approches ne peuvent pas garantir de couvrir l'ensemble de nœuds d'un graphe dans la structure communautaire ainsi calculée. Dans [30]. une approche plus originale est proposée ou après la détection de graines, chaque nœud dans le graphe (graine ou non) calcule un vecteur de préférence d'appartenance aux communautés de chaque graine. L'appartenance communautaire des nœuds est le résultat d'un processus de vote local impliquant le nœud et ses voisins directs. Une étude comparative des approches centrées graines est présentée dans [92].

2.8 Conclusion

Dans ce chapitre nous avons introduit quelques notions de la théorie des graphes utiles pour la modélisation des réseaux sociaux. Cette modélisation(représentation par graphe) nous permettra d'analyser les graphes et détecter les structures communautaires existantes. La détection de communautés est utile dans plusieurs domaine : économique, santé, biologie...etc. Pour cela nous avons d'écrit et classé beaucoup de travaux qui traitent ce problème, qui peut être évalué par plusieurs métriques et sur différents types de data sets. Nous avons abordé les concepts fondamentaux de la théorie des graphes, utilisés ici pour représenter et manipuler les réseaux sociaux.

Chapitre 3

Prédiction des liens

3.1 Introduction

La tâche de prédiction de liens [65] consiste à prédire les parties d'arcs manquants. Supposons que le directeur du film Avatar soit absent du graphes de connaissances (GC), on souhaite le prédire, c-à-d l'identifier parmi tous les nœuds du GC. Le principe est de trouver des régularités dans les connaissances existantes et de les exploiter afin de classer les nœuds du GC. Plus haut est un nœud dans le classement, meilleure est la prédiction. La prédiction de liens a été introduite pour les réseaux sociaux avec un seul type d'arc [49] puis a ensuite été étendu à des données multi-relationnelles et appliquée aux GC [65]. Comparé à la classification supervisée, la prédiction de liens affronte plusieurs défis.

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 termes 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.

3.2 Graphe biparti

Définition

Un graphe est dit biparti si on peut partager son ensemble de sommets en deux parties A et B tels qu'il n'y ait aucune arête entre éléments de A et aucune arête entre éléments de B.

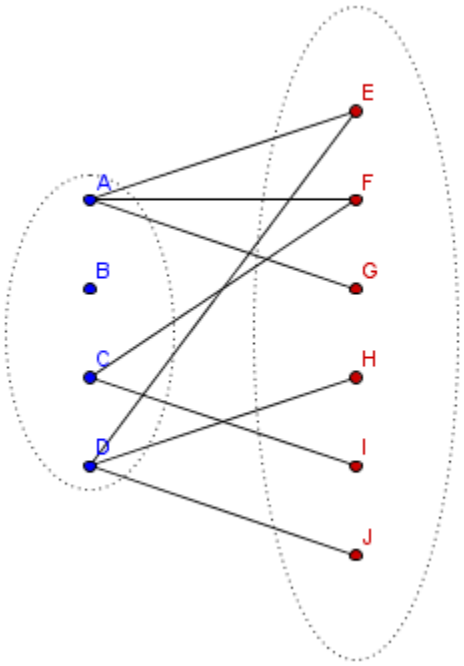

FIGURE 3.1 – Graphe biparti

Autrement dit, les graphes bipartis sont ceux que l'on peut colorer en utilisant au plus deux couleurs. Le théorème suivant, dû à König en 1916, caractérise les graphes bipartis :

Théorème : Un graphe est biparti si et seulement s'il ne contient pas de cycles de longueur impaire.

Rappelons que la longueur d'un cycle est égale au nombre d'arêtes qu'il contient. En particulier, d'après le théorème précédent, les arbres sont des graphes bipartis.

Pour les graphes bipartites Clients/Produits, le filtrage collaboratif [66] consiste à déterminer pour chaque utilisateur U l'ensemble des produits utilisés par les utilisateurs qui lui sont similaires puis à les lui recommander. Cette méthode a connu de nombreux succès [67]. selon [68] ont proposé une autre approche basée sur la notion de liens internes. Le principe des liens internes stipule que deux nœuds ayant au moins un voisin en commun pourront en acquérir davantage dans le futur tandis que deux nœuds qui n'en ont pas n'en auront jamais dans le futur. Cette dernière approche n'est malheureusement pas applicable aux graphes bipartites de publications car un lien interne ne peut exister qu'entre deux nœuds existants tandis que dans les graphes de publications, la création d'un lien se fait à travers une nouvelle publication (création d'un nœud de type "article").

Si le graphe biparti étudie les relations entre deux ensembles distincts, il ne prend pas en compte les relations à l'intérieur de ces deux ensembles. Par ailleurs, si les relations au sein d'un graphe biparti sont généralement non orientées, elles peuvent par contre être valuées. Ainsi, un auteur peut publier plusieurs fois dans une même revue.

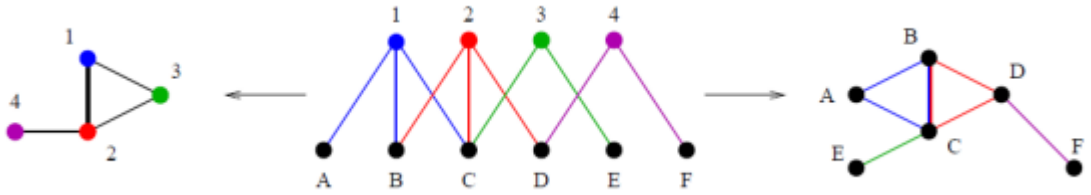

FIGURE 3.2 – Du graphe biparti aux graphes de co-occurrence

La transformation d'un graphe biparti - au centre - entre graphes de cooccurrence revient à multiplier la matrice de départ par sa transposée ou la transposée par la matrice de départ. Il n'est pas possible de retrouver le graphe d'origine à partir des matrices de co-occurence obtenues. Le double lien entre B et C sur le graphe de droite signale qu'ils ont deux liens communs dans le graphe de départ (vers les sommets 1 et 2)[69].

L'une des méthodes les plus courantes pour analyser ces graphes est de les transformer en deux graphes distincts de co-occurrence : le graphe acteur - événement est donc transformé en un graphe acteurs - acteurs (un lien entre deux acteurs indique qu'ils étaient tous deux présents à un même événement) et un graphe événement - événement (un lien entre deux événements indique que le même acteur a assisté aux deux). La figure montre comment s'effectue ces transformations. Les deux graphes obtenus peuvent alors être considérés comme deux graphes valués standards. Cette approche a longtemps été critiquée dans la mesure où la transformation implique une perte importante d'informations, point de vue remis récemment en question par Martin G. Everett et Stephen P. Borgatti [70] : les deux auteurs affirment en effet que l'analyse conjointe des deux graphes valués (acteurs - acteurs ; événements - événements) permet de trouver des résultats similaires et, dans la plupart des cas, de reconstituer le graphe de départ.

Point à souligner, certains auteurs ont affirmé que les réseaux bipartis étaient plus fiables que les réseaux « classiques » dans la mesure où la participation d'un individu à un événement pouvait être connue sans risque d'erreur, tandis que des biais importants existent quand les données portent sur des relations directes. Il n'est pas certain que cet argument soit totalement pertinent : ainsi, toute personne ayant étudié la participation de chercheurs à des colloques sait que la liste des participants ne correspond jamais à la liste des présents. De plus, et il est prudent de le rappeler, co-présence ne signifie pas interaction. Que deux auteurs publient dans une même revue ou utilisent à l'occasion un même mot clé est un indicateur faible d'un quelconque lien entre ces deux auteurs.

Cette partie s'intéresse uniquement aux méthodes d'analyse possibles sur un graphe biparti non transformé en abordant successivement les mesures possibles (globales et locales), la recherche de sous graphes fortement connexes, les adaptations des modèles petits-mondes et sans-échelle et enfin l'étude dynamique des graphes bipartis. Les enjeux posés par la visualisation des graphes bipartis seront abordés dans une synthèse ultérieure.

3.3 Prédiction des liens

La prédiction de liens est appliquée dans une grande varieté de domaines tels que l'analyse des liens, la bioinformatique, la recherche d'information[39], etc. Par exemple, on pourrait prévoir des futures amitiés lors de l'analyse des réseaux sociaux ou prédire les futurs co-auteurs dans un réseau de collaboration [40]. Formellement, la tâche de prédiction de liens peut être formulée comme suit [41] :

Etant donne un réseau social $G(V, E)$ ou V est l'ensemble des nœuds qui peuvent être de différents types (individus, organisations, entreprises, etc) et E est l'ensemble des arêtes les reliant a travers un type d'interdépendance (amitié, échange financier, proximité physique,etc). Une arête entre une paire de nœuds $(v_i, v_j) \in V$ represente une association qui a eu lieu a un moment donné t. La tâche est de prédire l'ensemble des liens potentiels qui peuvent être formés a l'instant $t + 1$.

La plupart des méthodes de prédiction de liens de l'état de l'art reposent sur deux groupes d'information du réseau qui peuvent être classés en information locale (basée sur les nœuds voisins) et information globale (basée sur les chemins dans le graphe).

Les approches reposant sur l'information locale utilisent les similarites locales qui caractérisent les nœuds dans le réseau. Ces dernières peuvent ' être les attributs essentiels a savoir le genre, l'age, les interets, ou des indices structurels basés uniquement sur la structure du réseau, par exemple les voisins communs entre deux nœuds. Cependant, les attributs des nœuds ne sont pas genéralement disponibles ou sont cachés [42]. Pour cette raison, la majorité des approches utilisent seulement les mesures reposant sur les similarites structurelles. Les approches globales utilisent des mesures basées sur l'ensemble des chemins entre les noeuds dans le réseau afin de déterminer ceux qui sont plus proches. L'intuition est que plus les nœuds sont proches dans le réseau, plus ils ont tendance a être liés ou a s'influencer dans l'avenir. L'avantage principal de ces deux types de mesures est qu'elles sont géneriques et par consequent peuvent être appliquées sur des graphes de différents domaines [39]. Ainsi, nous rappelons dans la suite de cette section certaines mesures de l'état de l'art basées sur les informations locales et globales.

3.3.1 Approches de prédiction de liens

Trois critères de classification :

Approche : Dyadiques / Structurelles

- Dyadique : Evaluer un score d'un lien entre deux nœuds$(v_i; v_j)$,
- Structurelle : Prédire l'évolution de sous-graphes (Prédiction de plusieurs liens en même temps).

Type d'attributs : topologiques / caractéristiques des Nœuds

- Approche topologique : Utiliser seulement le graphe du réseau.

— L'emploi des approches fondées sur l'analyse du contenu des nœuds nécessitent une expertise dans le domaine de l'application.

Prise en compte du temps : Oui / non

— Score d'un lien (u, v) est calculé par une fonction de similarité topologique,
— Deux familles de mesures de similarités topologiques :

*Mesures basées sur le voisinage des nœuds,

*Mesures basées sur les distances entre les nœuds.

3.3.2 Techniques de prédiction des liens

Dans un réseau social, il existe deux façons pour prédire l'évolution des liens , les approches non supervisée les approches basées sur l'apprentissage supervisé. Les approches non supervisés calculent une valeur de similarité, c'est un score attribué à chaque paire de noeuds non connectés (x, y), un score élevé indique une grande probabilité que x et y seront liés dans le futur et vice versa, après une liste des scores ordonnées est construite et les liens qui ont des grandes valeurs de similarité sont les plus susceptibles d'être liée.

Les approches basées sur l'apprentissage supervisé traitent ce problème comme un problème de classification binaire, par conséquent, nombreux modèles d'apprentissage et de probabilité peuvent être utilisé pour résoudre ce problème.

Les approche non supervisés

Ils existent beaucoup de méthodes de prédiction des liens non supervisés, simples et basiques, utilisent l'information de noeuds, la topologie et la théorie social pour calculer la similarité entre les paires de noeuds non connectés, les méthodes basé sur l'apprentissage supervisé sont les plus complexe, mais ils ont composé par des mesures de cette classe, nous allons présenté une vue systématique des ces mesures.

Mesures basées sur le contenu d'un nœud

Le calcul de la similarité entre les pairs de noeuds est une solution intuitive dans la tâche de la prédiction des liens. Il est basé sur une idée simple : les paires les plus similaires sont des noeuds ayant une grande vraisemblance et donc ce sont les plus susceptibles d'être reliée vice versa.

Cette hypothèse conforme au concept que les personnes tendent à créer des relations avec d'autres personnes qui sont similaires dans l'éducation, religions, les intérêts et localisation, ces caractéristiques peuvent être mesurées par une similarité attribuée à chaque pair de noeuds, une grande valeur de similarité entre deux noeuds indique qu'ils ont une grande probabilité d'être liée dans le futur.

Dans les réseaux sociaux réels, un noeud est généralement à un ou plusieurs attributs qui le caractérisent comme les profiles des utilisateurs dans les réseaux sociaux, nom d'un email dans les réseaux des emails, des publications dans les réseaux sociaux académiques, ces informations peuvent être exploitées directement pour calculer la similarité entre les pairs de noeuds. Dans la plus part des cas, les valeurs de ces attributs ayant une forme textuelle ce qu'il facilite le calcul de la similarité.

Bhattacharyya et Garg [97] ont remarqué par exemple qu'une personne dans un réseau social aime le football et une autre aime le soccer ou bien sport, malgré qu'ils n'ont aucune relation directe ils ont une similarité par ce qu'ils aiment le même contexte c'est le sport, en se basant sur cette idée, ils ont construit plusieurs modèles d'arbres de catégorisation pour étudier les mots-clés de profile des utilisateurs puis, ils ont définit des distances entre les mots clés pour déterminer la similarité entre les pairs d'utilisateurs. Leur observation la plus importante est que, sauf pour les amis directs, la similarité entre les utilisateurs sont approximativement la même, quelles que soient les paramètres topologiques de réseau. Ils montrent également que l'augmentation du nombre d'amis et les mots clés diminue la similarité entre une personne et leurs amis.

Anderson et Huttenlocher [98] utilisent principalement les intérêts des utilisateurs comme une mesure de similarité, ces intérêts sont présentés par des activités, par exemple éditer un article dans WIKIPEDIA, poser une question dans StackOverflow, commenter un statu dans Facebook, évaluer des produits d'un site e-commerce, évalué une application dans le PlayStore... tous ces actions sont présentées dans un vecteur de poids en calculant les nombres d'interactions par rapport aux interactions avec d'autres groupes, personnes etc. Une grande valeur indique que cette personne favorise par exemple des status d'une telle page, produits, d'autres utilisateurs...

En conclusion, ils existent des dizaines de méthodes qui utilisent comme référence les attributs et les activités des utilisateurs dans les réseaux sociaux, ces approches donnent des très bons résultats si nous pouvons capturer le maximum de ceux-ci, ce qui nous permettre de connaitre de plus en plus les comportements et les personnalités des internautes dans les réseaux sociaux.

3.4 Modélisation d'un système de recommandation en graphe biparti

La recommandation des produits appropriés aux clients est cruciale dans de nombreuses plateformes de e-commerce qui proposent un grand nombre de produits. Les systèmes de recommandation sont une solution favorite pour la réalisation de cette tâche. La majorité des recherches de ce domaine reposent sur des notes explicites que les utilisateurs attribuent aux produits, alors que la plupart du temps ces notes ne sont pas disponibles en quantité suffisante. Il est donc important que les systèmes de recommandation utilisent les données implicites que sont des flots de liens représentant les relations entre les utilisateurs et les produits, c'est-à-dire l'historique de navigation,

des achats et de diffusion. C'est ce type de données implicites que nous exploitons. Une approche populaire des systèmes de recommandation consiste, pour un entier N donné, à proposer les N produits les plus pertinents pour chaque utilisateur : on parle de recommandation top-N.

Pour ce faire, bon nombre de travaux reposent sur des informations telles que les caractéristiques des produits, les goûts et préférences antérieurs des utilisateurs et les relations de confiance entre ces derniers.

Cependant, ces systèmes n'utilisent qu'un ou deux types d'information simultanément, ce qui peut limiter leurs performances car l'intérêt qu'un utilisateur a pour un produit peut à la fois dépendre de plus de deux types d'information. Une extension du Session-based Temporal Graph (STG) introduit par Xiang et al [99], et qui est un graphe dynamique combinant les préférences à long et à court terme des utilisateurs, ce qui permet de mieux capturer la dynamique des préférences de ces derniers. STG ne tient pas compte des caractéristiques des produits et ne fait aucune différence de poids entre les arêtes les plus récentes et les arêtes les plus anciennes. Le nouveau graphe proposé, Time-weight content-based STG contourne les limites du STG en y intégrant un nouveau type de nœud pour les caractéristiques des produits et une pénalisation des arêtes les plus anciennes. Un système de recommandation basé sur l'utilisation de Link Stream Graph (LSG). Ce graphe est inspiré d'une représentation des flots de liens et a la particularité de considérer le temps de manière continue contrairement aux autres graphes de la littérature, qui soit ignore la dimension temporelle comme le graphe biparti classique (BIP), soit considère le temps de manière discontinue avec un découpage du temps en tranches comme STG.

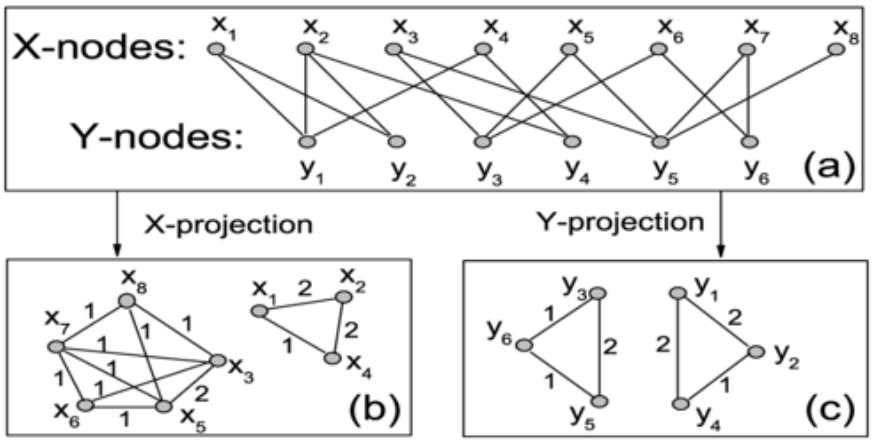

FIGURE 3.3 – Exemple de graphe biparti classique

Le graphe biparti classique (a) dans la figure 3.3 , sa projection sur la dimension X (b) et la dimension Y (c). Les poids des arêtes en (a) et (b), sont les nombres de voisins communs dans X et dans Y respectivement

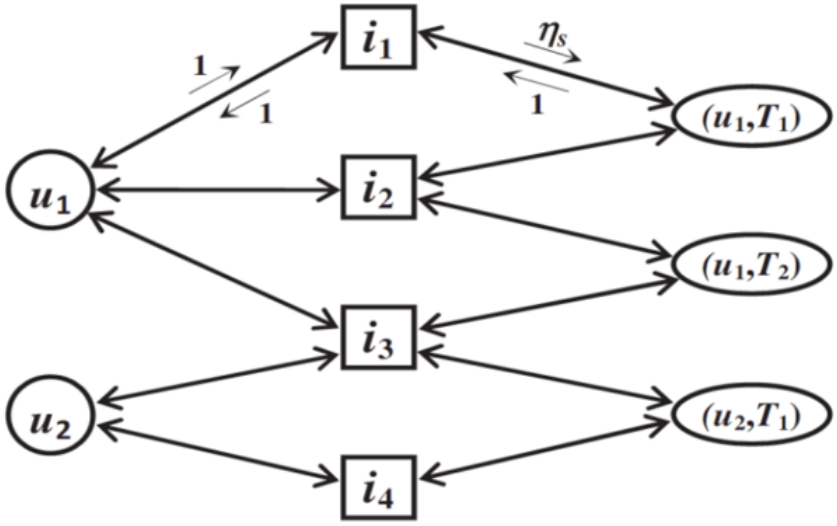

FIGURE 3.4 – Graphe Temporel basé sur la Session(STG)

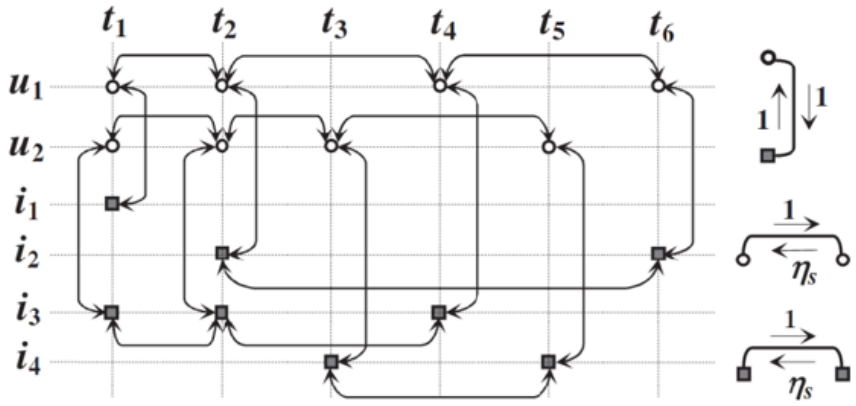

FIGURE 3.5 – Graphe de flux de liens

3.5 Modélisation avec un réseau bipartite pondéré

$$N = I \cup U$$

$$L = \{(i, u, R(i, j)), i \in I, u \in U, R(i, j) \in R^+\}$$

$$I \cap U = \emptyset$$

Un réseau pondéré fait référence à un réseau dont les liens sont quantifiés. Chaque lien a une valeur qui représente le degrés de la relation entre les deux nœuds. La topologie d'un réseau bipartite pondéré avec précision reflète l'architecture d'un système de recommandation. En particulier, l'approche de filtrage collaboratif qui est largement adopté dans les systèmes de recommandation est parfaitement modélisé par ce type de réseau. Les éléments (films,livres, pages web, etc.) représentent les nœuds du premier

ensemble, les utilisateurs représentent les nœuds du second ensemble, et la valeur de lien représente une valeur de classement.

3.6 Mesures de similarités

Dans le domaine des systèmes de recommandation, il existe deux principales familles de techniques pour réaliser des recommandations [61] [72] : une famille de techniques basées sur le filtrage collaboratif qui recherche des similarités de profils utilisateurs en se basant sur des notations (un nombre d'étoiles, la liste des achats passés, les lieux visités, etc.) et une famille de techniques qui se base sur des similarités de profils de contenus sur la base de descripteurs. Le papier [73] compare ces méthodes et affirme que les méthodes par filtrage collaboratif donnent généralement de meilleurs résultats que le filtrage de contenu, tout du moins lorsqu'il y a suffisamment de notations disponibles.

Cependant, lorsqu'un nouvel utilisateur ou un nouveau contenu survient dans le système, les méthodes par filtrage collaboratif ont des difficultés à fournir des recommandations efficaces aux nouveaux utilisateurs ou à recommander les nouveaux contenus, parce qu'aucun historique n'est disponible sur eux. Ce problème, appelé problème de démarrage à froid, est surmonté par les méthodes de filtrage de contenu. Cependant, ces techniques nécessitent des méta-données fiables : il s'avère que les profils d'internautes ou les fiches descriptives de lieux ne le sont pas, les préférences déclaratives quand elles existent sont souvent inconsistantes ; la collecte d'informations pertinentes de description de lieux pour attirer les visiteurs peut s'avérer coûteuse et complexe, voire impossible si l'on prend des commerces.

Nous considérons le pire cas où il n'y a pas de descripteurs sur les utilisateurs ni sur les lieux. Ainsi, nous rejetons les méthodes basées sur le contenu et nous focalisons sur les méthodes à filtrage collaboratif basées sur les actions spontanées des utilisateurs, comme leur notation sur les items, et dans notre cas, comme les check-ins dans les réseaux sociaux. Parmi ces actions spontanées, encore souvent sous-exploitées en recommandation, figurent aussi les actions sociales, comme le tissage de relations ou les interactions, sur les plateformes qui le permettent.

Quelques travaux ont pris en compte le graphe social en plus des relations utilisateurs-items pour les recommandations. L'auteur [74] montre que les données sociales apportent bien une amélioration des recommandations, en particulier dans un contexte de démarrage à froid, où peu d'information d'historique sont disponibles. Un clustering sur le graphe social semble capturer la détection de communautés de goûts cinématographiques puisqu'il améliore la recommandation de contenus peu populaires.

Selon [75] proposent une méthode pour factoriser la matrice de notation utilisateurs-items par la méthode de décomposition en valeurs singulières (SVD) en minimisant une fonction objectif. Si un terme appelé régularisation sociale est ajouté à la fonction, terme basé sur les amis directs de l'utilisateur dans le réseau social, la précision des recommandations est améliorée.

Dans [76], les auteurs proposent de combiner les matrices de similarités entre utilisateurs, dérivées des réseaux sociaux implicites et explicites.

Ils calculent deux matrices de similarités, une basée sur le réseau d'amis et une basée sur le réseau bipartite (utilisateur-item). Ces deux matrices sont combinées en une seule

matrice de similarité par une somme pondérée. Ils généralisent ensuite ce modèle pour incorporer plus de graphes. Leur algorithme donne de meilleures recommandations que les méthodes traditionnelles de filtrage collaboratif par voisinage.

Dans [77], les auteurs proposent un système de recommandation de groupes basé sur le réseau social d'amis, et basé sur le réseau de groupes reliant des utilisateurs à des groupes. Ils décrivent tout d'abord une manière de combiner le réseau social d'amis et le réseau de groupes en un seul graphe. Puis, ils proposent deux méthodes pour recommander des groupes : une basée sur la proximité dans le graphe en utilisant la mesure de Katz et une méthode modélisant les utilisateurs et les groupes selon des facteurs latents. Ces deux méthodes donnent de bons résultats, mais la méthode par mesure de Katz est la plus efficace en termes de temps de calcul et de qualité des recommandations.

3.6.1 Katz FSG

Un algorithme basé sur la mesure de Katz et considérant les informations de Fréquentations, Sociales et Géographiques : La méthode KatzFSG permet de recommander des lieux aux utilisateurs selon leurs check-ins et leur réseau social, en considérant le contexte géographique des lieux visités. Elle est composée de trois parties. La première se focalise sur la définition des différents graphes. La deuxième partie insiste sur la création du graphe géographique qui est un des principaux aspects de la méthode. La troisième partie décrit la méthode de fusion de graphes et comment celui-ci est utilisé pour induire des recommandations à l'aide d'une propagation de poids utilisant la méthode de centralité de Katz.

3.6.2 Définitions des trois graphes

Le graphe social S est graphe d'adjacence où les sommets représentent les utilisateurs et les arêtes les relations d'amitié. S est représenté par une matrice S, N×N symétrique (N étant le nombre d'utilisateurs).où S_ij vaut 1 si une relation d'amitié existe entre l'utilisateur u_i et l'utilisateur u_j , et vaut 0 sinon

Le graphe de fréquentation F se base sur les check-ins des utilisateurs dans les différents lieux. Dans le graphe bipartite, les nœuds sont soit des utilisateurs, soit des lieux. Les arêtes relient les utilisateurs avec les lieux dans lesquels ils ont fait un ou plusieurs check-in(s), elles sont pondérées en fonction du nombre de visites. La matrice F, qui représente ce graphe, est une matrice N×$M(M$ *étant le nombre de lieux*$), o\grave{u}\mathrm{F}_{ik}$ est le nombre de fois que l'utilisateur u_i a fréquenté le lieu l_k.

Le graphe géographique G relie les lieux entre eux. La matrice G, qui représente ce graphe, est une matrice M×M. Cette matrice est décrite dans la partie suivante.

Un Graphe Géographique basé sur le comportement de check-in

La construite de graphe en considérant le comportement de check-in de chaque utilisateur dans l'espace géographique. Il apparaît que le comportement de check-in des utilisateurs est fortement influencé par la proximité entre les lieux, comme vu dans [78].

Dans ce dernier, les auteurs construisent la densité de probabilité des check-ins suivant leur distance à un autre check-in et approximent les points obtenus par une courbe définie par la fonction $f(x) = ax^b$(fonction loi de puissance). f permet d'inférer la probabilité $\rho_r(d(l_i, l_j))$ de check-in en un lieu pour toute distance avec un autre check-in de l'utilisateur. Ils utilisent ensuite f pour calculer la probabilité qu'un check-in soit fait en un lieu suivant ses distances avec tous les check-ins effectués, par méthode naïve bayésienne (produit des probabilités liés à chaque distance avec les check-ins).

La distribution de probabilité représentée par f permet de relier chaque paire de lieux par une probabilité de check-in suivant leur distance mutuelle. Ainsi, pour chaque utilisateur, il est possible de créer un graphe géographique $G_u$dont les sommets sont les lieux et dont les arêtes sont pondérées par la probabilité de check-in calculée sur la distance mutuelle des lieux considérés.

Un tel graphe géographique G_u existe pour chaque utilisateur. Néanmoins, il apparaît que ces graphes ne varient que très peu d'un utilisateur à un autre. De plus, le graphe réalisé pour un utilisateur avec très peu de check-ins n'est pas vraiment pertinent car il n'y a pas assez de check-ins pour trouver la fonction de distribution réelle.

3.6.3 La mesure de Katz sur le graphe fusionné

Le graphe fisionné c'est les graphes S,F et G dans un unique graphe C. Ainsi, dans ce graphe, les nœuds sont soit des lieux soit des utilisateurs, et les arêtes peuvent relier des utilisateurs entre eux, des lieux entre eux ou des utilisateurs avec des lieux. La matrice C représente ce graphe unifié et est construite comme suit :

$$C = \begin{pmatrix} \alpha S & \lambda f \\ \lambda F^T & \gamma G \end{pmatrix}$$

Les coefficients α, λ, γ sont respectivement les degrés d'influence des matrices S,F,G dans la matrice C, avec $\alpha + \lambda + \gamma = 1$. Il est aussi à noter que F et G sont normalisés avant d'intégrer C. Ensuite propager un poids dans le graphe C par mesure de Katz est comme suit :

$$katz(c) = \beta C + \beta^2 C^2 + \beta^3 C^3 + \cdots, 0 \leq \beta \leq 1$$

β Est le poids qui est propagé à travers le graphe. Par cet algorithme, ce qui est plus intéressons est l'effet de la propagation de poids sur les relations utilisateurs-lieux. Ces relations sont représentées par le bloc katz (c) dans la matrice katz(c) Etant donné le coût important de ce calcul, une matrice de Katz tronquée est calculée telle que :

$$tKatz(c, k) = \Sigma_{i=1}^{k} (\beta^i C^i)$$

Finalement, pour chaque utilisateur u , les n lieux non visités ayant les meilleurs poids sur la ligne de l'utilisateur dans la matrice tKatz (c, k) sont sélectionnés. Ces lieux sont alors les recommandations à proposer à l'utilisateur correspondant.

3.7 Contribution et mise en ouvre

Les systèmes de recommandation (SR) sont un outil de recherche et de filtrage d'information qui visent à proposer aux utilisateurs des items qui pourraient les intéresser.

La technique des SR à laquelle nous nous intéressons dans notre travail est le filtrage collaboratif (FC). Elle est considérée comme l'une des techniques les plus utilisées dans les systèmes de recommandation, en raison de son efficacité [21][71] . Cette technique repose sur une hypothèse fondamentale stipulant que les utilisateurs ayant noté, aimé les mêmes items ou ayant des comportements similaires (acheter, regarder, consulter, ...), dans le passé, ils évalueront ou agiront sur d'autres items de la même manière, dans le futur. En effet, cette technique se base sur une matrice d'évaluations (notes) exprimées par les utilisateurs sur les items qui les ont intéressés.

Cependant, en pratique, un grand nombre de notes de cette matrice n'est pas disponible, en raison de la rareté inhérente aux données d'évaluation, ce qui est plus connu sous le nom du problème de parcimonie (sparsity, en anglais). Par conséquent, la qualité de la prédiction de notes baisse d'une manière considérable ce qui réduit la précision des algorithmes de FC.

En outre, les algorithmes de filtrage collaboratif tel qu'ils sont ne prennent pas en considération l'arrivée d'un nouvel utilisateur ou un nouvel item qui n'a pas d'historique de préférences ou de notes dans la matrice d'évaluations. Ceci est plus connu sous le nom du problème de démarrage à froid.

Ces deux problèmes limitent considérablement l'applicabilité des systèmes de recommandation en particulier dans les applications à grande échelle.

Pour surmonter ces problèmes, l'axe de la recommandation basée sur les graphes a été exploré afin de bénéficier des relations cachées et des corrélations qui existent entre les différents types de nœuds. Ces corrélations ne sont principalement pas basées sur la notion d'évaluation.

On se focalise sur la recommandation basée graphe. En effet, la matrice d'évaluation peut être vue comme une matrice d'adjacence d'un graphe biparti avec comme types de nœuds : les utilisateurs et les items. Le problème de recommandation se traduit alors en un problème de prédiction de liens [41] dans ce graphe biparti. En effet, la prédiction de liens fournit des méthodes pour estimer les connexions potentielles dans les graphes, ce qui a une importance théorique et pratique pour la recommandation. Notre objectif, à ce stade, sera d'adapter les techniques de prédiction de liens afin de prendre en compte les spécificités du graphe biparti.

3.7.1 Prédiction de lien d'item

Plusieurs recherches [82][83][84] abordent le problème de la prédiction des liens autour du nœud utilisateur en fonction des liens non pondérés de ses voisins. Or, le lien exprime une action binaire (achat, j'aime, signe positif), l'absence de lien exprime la

négation (non-achat, aversion, signe négatif). La prédiction d'un lien vers l'utilisateur actif dépend directement des liens de ses voisins. SibRank [85] repose sur deux types de liens : positif s'il y a accord entre l'utilisateur et la préférence ou négatif sinon. Pour déduire la similarité entre utilisateurs, la méthode explore la propagation multiplicative des signes de confiance/méfiance selon le principe les ennemis de mon ennemi et les amis de mon ami sont amis, alors que les ennemis de mon ami et les amis de mon ennemi sont ennemis . La fonction Srank permet d'estimer l'accord ou le désaccord entre l'utilisateur actif et les autres à travers les informations déduites de la structure du graphe. Dans ce travail nous présentons une nouvelle méthode de prédiction de lien basée sur l'approche item dans un réseau bipartite pondéré. Le poids d'un lien s'exprime sur une valeur de notation numérique plus ou moins précise qu'une valeur binaire. La méthode est basée sur la préservation des informations cachées (intérêt) grâce à une connectivité pondérée en item/utilisateurs à double projection.

Considérons un système de recommandation composé d'un ensemble d'item $\{I_1, I_2, \cdots I_N\}$ et un ensemble d'utilisateurs $\{U_1, U_2, \cdots U_M\}$ liés par des liens pondérés $\{(i, u, R(i, j)), i \in I, u \in U, R(i, j) \in R+\}$. Des interactions précédentes des utilisateurs avec les items (l'historique des notes), essayons de déduire un poids de lien inexistant. D'une autre manière, peut-on recommander un item planifié pertinent à un utilisateur ?. Cette décision revient à évaluer le poids du lien entre l'item et l'utilisateur, comme (I3, U1) (Figure 3.6). Pour faire une telle prédiction, nous cherchons à déterminer les parts d'éléments partagés entre les utilisateurs. Dans la première étape (étape vers l'avant), nous accumulons les quotas des utilisateurs vers les items et dans la seconde étape (backback step) les parts des items vers les utilisateurs, sachant que N est le nombre d'items,M est le nombre d'utilisateurs, et $R(i, j)$ est le poids du lien (i, j) (valeur de notation).

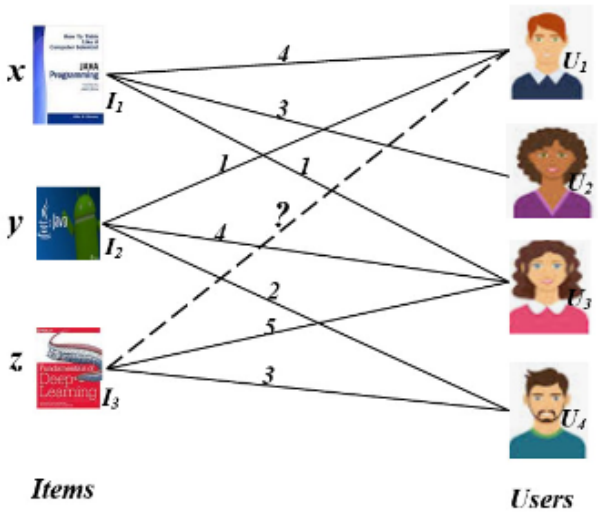

FIGURE 3.6 – Prédiction de lien pondérée

— Dans la première étape, chaque item a un cumul de notes

$$IC_i = \sum_{j=1}^{M} R\left(i,j\right) \tag{3.1}$$

— La part de l'utilisateur j

$$q_j = \sum_{j=1}^{M} \frac{R\left(i,j\right)}{IC_i} \tag{3.2}$$

— Dans la seconde étape, chaque utilisateur dispose d'un cumul de notes,

$$UC_j = \sum_{i=1}^{N} R\left(i,j\right) \tag{3.3}$$

— La part de l'item i ,

$$q(i) = \sum_{j=1}^{M} \frac{R\left(i,j\right)}{IC_j} \tag{3.4}$$

— Ensuite, la valeur de prédiction de lien de l'item t vers l'utilisateur s est donnée par la formule suivante :

$$\rho\left(t,s\right) = \sum_{j=1}^{M} \frac{R\left(t,j\right)}{\sum_{i=1}^{N} R(i,j)} * (\sum_{i=1}^{N} R(i,j)) * \frac{R\left(i,s\right)}{\sum_{j=1}^{M} R(i,j)} \tag{3.5}$$

— En remplaçant l'élément cumulé (3.1) et l'utilisateur cumulé (3.3), on obtient

$$\rho\left(t,s\right) = \sum_{j=1}^{M} \frac{R\left(t,j\right)}{UC_j} * (\sum_{i=1}^{N} R(i,j)) * \frac{R\left(i,s\right)}{IC_i} \tag{3.6}$$

La topologie bipartite offre un raisonnement bidirectionnel côté item et côté utilisateur, qui préserve les informations partagées entre les nœuds. La prise en compte des poids des liens précise le degré de satisfaction des utilisateurs. Cette méthode présente une prédiction basée sur les items, mais le même raisonnement basé sur l'utilisateur peut également être appliqué (Figure 3.7).

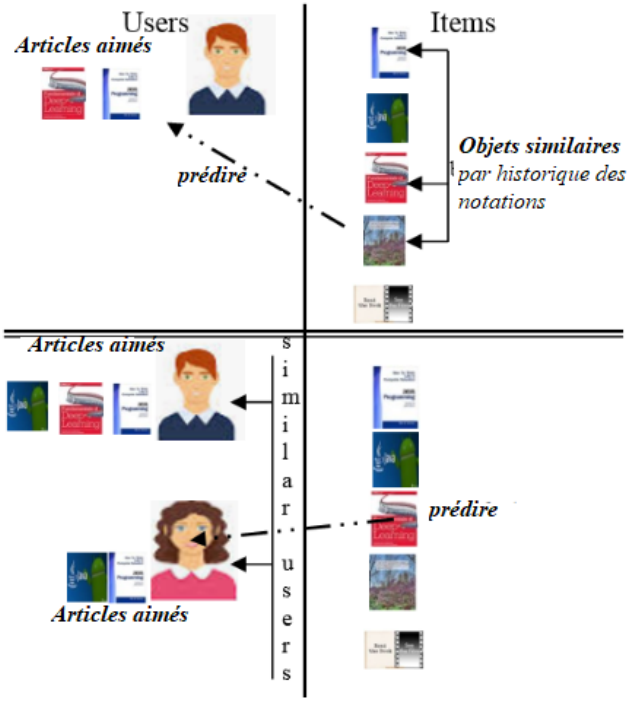

FIGURE 3.7 – Prédiction de lien item versus prédiction de lien utilisateur

L'exemple ci-dessous détaille la méthode proposée via les deux variantes à savoir la prédiction de lien item et la prédiction de lien utilisateur, en précisant les performances de prédiction entre elles. Sur la base des valeurs de lien présentées dans la figure (3.6)

a) Prédiction de lien d'item

Soit trois item I_1, I_2, I_3 avec des entrées initiales x, y, z respectivement.

Première étape (en avant) : déterminer la part de l'utilisateur,

$$\begin{cases} u_1 : \frac{4}{8}x + \frac{1}{7}y \\ u_2 : \frac{3}{8}x \\ u_3 : \frac{1}{8}x + \frac{4}{7}y + \frac{5}{8}z \\ u_4 : \frac{2}{7}y + \frac{3}{8}z \end{cases}$$

Deuxième étape (en arrière) : déterminer la part de l'item,

$$\begin{cases} I_1\text{: } \frac{4}{5}U_1 + \frac{3}{3}U_2 + \frac{1}{10}U_3 + \frac{0}{5}U_3 \\ I_2\text{: } \frac{1}{5}U_1 + \frac{0}{3}U_2 + \frac{4}{10}U_3 + \frac{2}{5}U_3 \\ I_3\text{: } \frac{0}{5}U_1 + \frac{0}{3}U_2 + \frac{5}{10}U_3 + \frac{3}{5}U_3 \end{cases}$$

Ce qui nous donne :

$$\begin{cases} I_1 : \frac{63}{80}x + \frac{6}{35}y + \frac{1}{16}z \\ I_2 : \frac{3}{20}x + \frac{13}{35}y + \frac{2}{5}z \\ I_3 : \frac{1}{16}x + \frac{16}{35}y + \frac{43}{80}z \end{cases}$$

Le tableau 3.1 montre les prédictions de lien d'item

	U_1	U_2	U_3	U_4
I_1	4	3	1	0.5304
I_2	1	0.45	4	2
I_3	0.7071	0.1875	5	3

TABLE 3.1 – Prédiction de lien d'item

Ainsi, la valeur de prédiction du lien
$(I_3, U_1) = 1/16\ (4) + 16/35\ (1) + 43/80\ (0) = 0{,}7071$
De plus, la valeur de prédiction du lien
$(I_1, U_4) = 63/80\ (0) + 6/35\ (2) + 1/16\ (3) = 0{,}5304$

Pour connaître l'erreur de prédiction, nous généralisons le calcul pour les valeurs de vote existantes (tableau 3.2) et appliquons la mesure d'erreur absolue moyenne MAE

La mesure de l'erreur absolue moyenne (MAE : Mean Absolute Error) c'est une mesure statistique qui s'appuie sur la moyenne des différences entre chaque note prédite et sa note réelle, formellement :

$$MAE = \frac{\sum_{(u,i)\in k} |r_{u,i} - \widehat{r}_{u,i}|}{|k|}$$

Où :

— $r_{u,i}$ est la vraie note donnée par u à i.
— $\widehat{r}_{u,i}$ la note prédite par le SR.
— k est l'ensemble des couples (user, item) pour lesquels la confrontation est effectuée.

	U_1	U_2	U_3	U_4	$MAEI_i$
I_1	3.3214	2.3625	1.7857	0.5304	0.7006
I_2	0.9714	0.45	3.6357	1.9429	0.1500
I_3	0.7071	0.1875	4.5786	2.5268	0.4473
	MAE (moyenne)				0.4326

TABLE 3.2 – MAE des prédictions basées sur les items

La mesure MAE permet de différencier la valeur de notation donnée par les utilisateurs et la valeur de prédiction générée par la formule (3.6). Pour les postes I2 et I3 :

$MAE(I_2) = 1/3(|10.9714| + |43.6357| + |21.9429|) = 0.1500$
$MAE(I_3) = 1/2(|54.5786| + |32.5268|) = 0.4473$

b) Prédiction du lien utilisateur
Première étape (en avant) : déterminer la part de l'item,

$$\begin{cases} I_1 : \frac{4}{5}x + \frac{3}{3}y + \frac{1}{10}z \\ I_2 : \frac{1}{5}x + \frac{4}{10}z + \frac{2}{5}f \\ I_3 : \frac{5}{10}z + \frac{3}{5}f \end{cases}$$

Deuxième étape (en arrière) : Détermination de la part de l'utilisateur,

$$\begin{cases} u_1 : \frac{3}{7}x + \frac{1}{2}y + \frac{3}{28}z + \frac{2}{35}f \\ u_2 : \frac{3}{10}x + \frac{3}{8}y + \frac{3}{80}z \\ u_3 : \frac{3}{14}x + \frac{1}{8}y + \frac{31}{56}z + \frac{169}{280}f \\ u_4 : \frac{2}{35}x + \frac{169}{250}z + \frac{19}{56}f \end{cases}$$

De même, le tableau 3.3 montre la MAE des prédictions basées sur l'utilisateur. Nous pouvons clairement voir que la base d'items.

	I_1	I_2	I_3	$MAEI_i$
U_1	3.3214	0.9714	0.7071	0.3536
U_2	2.3625	0.4500	0.1875	0.6375
U_3	1.7857	3.6357	4.5786	0.5238
U_4	0.5304	1.9429	2.5268	0.2652
	MAE (moyenne)			0.4450

TABLE 3.3 – MAE de prédictions basées sur l'utilisateur

les prédictions (MAE = 0,4326) sont plus précises que les prédictions basées sur l'utilisateur (MAE = 0,4450).

3.8 conclusion

Suite à l'importance de la prédiction des liens opérant sur les réseaux sociaux. Actuellement, de nombreux auteurs se sont intéressés à ce nouveau domaine. Récemment, beaucoup des travaux de recherche proposant davantage de techniques et de fonctions de prédiction des liens dans les réseaux sociaux sont en train de se faire.

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. [79] 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.

Chapitre 4

Implémentation

4.1 Introduction

Dans ce chapitre, nous décrivons les travaux applicatifs et les détails de mise en œuvre de notre travail. Nous allons décrire d'abord la base de données utilisée pour valider notre proposition. Par la suite, nous présentons les outils et langage d'implémentation. Nous détaillons, ensuite, l'implémentation de différentes étapes, et nous terminons par les résultats obtenus.

4.2 Ensemble de données-Data Set

4.2.1 Description de la base

Dans notre travail, nous avons utilisé la base de données des films MovieLens. Cette base est utilisée dans de nombreux projets de recherche liés aux systèmes de recommandation.

La base de données MovieLens est constituée de :

— 100,000 notes données sur une échelle de 1 à 5 (R).
— 943 utilisateurs (U) avec diverses informations : démographiques, personnels, ...
— Chaque utilisateur a noté au moins 20 films
— Cette base est partagée en 80% des données en une base d'apprentissage et 20% en une base de test.

4.2.2 Tailles du dataset

Pour effectuer nos expérimentations, nous avons choisi d'une manière aléatoire un échantillon de données de la base de données MovieLens. Cet échantillon est constitué de 940 notes (R) données par 100 utilisateurs (U) pour 100 films (I). Cet échantillon reste toutefois extensible.

Le tableau suivant contient les informations concernant notre échantillon de données.

	Nombre d'utilisateurs	Nombre d'items	Nombre de notes
MovieLens Dataset	100	100	940

TABLE 4.1 – Les valeurs correspondantes au Tailles du dataset

4.3 Mise en œuvre

Dans cette section, nous allons spécifier, dans un premier temps, les outils utilisés pour développer notre application. Dans un second temps, nous allons décrire notre application de recommandation.

4.3.1 Outils et langage

Langage JAVA

Nous avons choisi le langage java comme langage de programmation. Java est un langage de programmation développé par Sun Microsystems à partir de 1990 et officiellement présenté en 1995. À la genèse du projet, James Gosling, un des pères fondateurs du langage, avait en tête de combler diverses lacunes du langage C++. Java a ainsi conservé une syntaxe similaire à ce dernier tout en l'épurant.

FIGURE 4.1 – Le langage de programmation orienté objet JAVA

Reprenant en grande partie la syntaxe du langage C++, le langage de programmation informatique orienté objet Java permet de développer des applications client-

serveur. Les applications développées en Java peuvent fonctionner sur différents systèmes d'exploitations, comme Windows ou Mac OS. Des plugins ajoutés aux navigateurs peuvent toutefois être nécessaire. "La popularité de l'environnement d'exécution Java dans les navigateurs, et le fait que Java dans les navigateurs soit indépendant de l'OS".

L'Editeur netbeans

NetBeans IDE est un environnement de développement intégré gratuit et à code source ouvert destiné au développement d'applications sous Windows, Mac, Linux et Solaris.

L'environnement IDE simplifie le développement d'applications Web, d'entreprise, de bureau et mobiles utilisant les plates-formes Java et HTML5. Il offre également une assistance pour le développement d'applications PHP et C/C++.

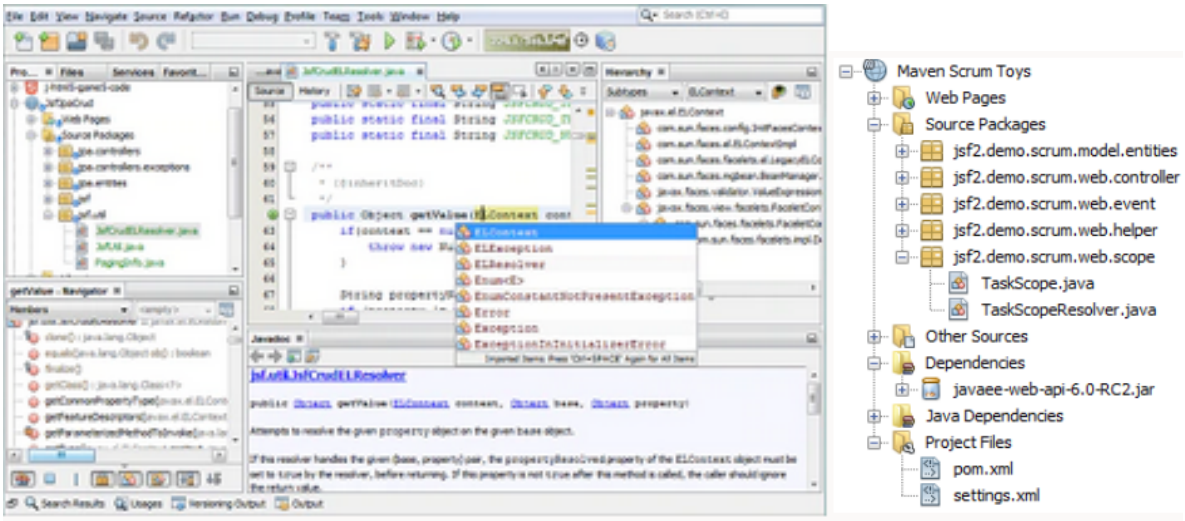

FIGURE 4.2 – Editeur NetBeans

NetBeans IDE propose des outils de première classe pour le développement d'applications Web, d'entreprise, de bureau et mobiles Java. Il est systématiquement le premier environnement IDE à prendre en charge les dernières versions de JDK, Java EE et JavaFX. Il fournit des aperçus intelligents pour vous aider à comprendre et à gérer vos applications, y compris une prise en charge complète des technologies populaires telles que Maven.

Avec ses fonctionnalités de développement d'applications de bout en bout, l'amélioration constante de Java Editor et ses améliorations en continu en termes de performances et de vitesse, NetBeans IDE établit la norme en matière de développement d'applications avec des technologies de pointe prêtes à l'emploi.

La base de données Oracle Database peut être enregistrée et accessible directement à partir de l'environnement IDE. Ce dernier prend en charge les connexions Thin JDBC et OCI avec la solution Oracle Database. Toutes les fonctionnalités d'accès aux données sont prêtes à l'emploi, telles que la possibilité de lire, créer, mettre à jour et supprimer des données directement dans l'environnement IDE, pris en charge par un éditeur SQL riche en fonctionnalités.

My SQL

Un serveur de bases de données stocke les données dans des tables séparées plutôt que de tout rassembler dans une seule table. Cela améliore la rapidité et la souplesse de l'ensemble. Les tables sont reliées par des relations définies, qui rendent possible la combinaison de données entre plusieurs tables durant une requête. Le SQL dans "MySQL" signifie "Structured Query Language" : le langage standard pour les traitements de bases de données.

Une requête en informatique, une requête est une interrogation d'une base de données. Elle peut comporter un certain nombre de critères pour préciser la demande. Il existe plusieurs langages de requêtes, qui sont spécifiques à la structure des bases de données. Le plus connu est le SQL, il est exploité dans les bases de données relationnelles (dont les informations sont enregistrées dans des tableaux à deux dimensions). OQL est le langage des bases de données orientées objet, WQuery celui des fichiers XML et Datalog celui des bases de données déductives.

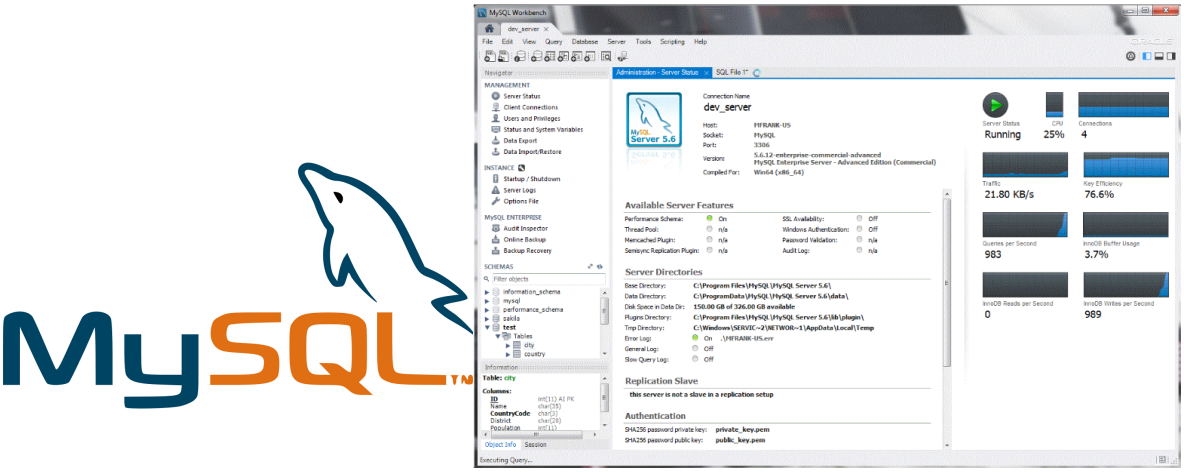

FIGURE 4.3 – SGBD My sql

Approche proposée

Notre approche proposée est de trouvé la prédiction de lien on calcule les voisins communs et la coefficient Jaccard en suite la mesure de l'erreur absolue moyenne.

En équivaut-on a aussi calculer la prédiction classique laquelle Vous nous avez demandé de calculer la similarité et la prédiction.

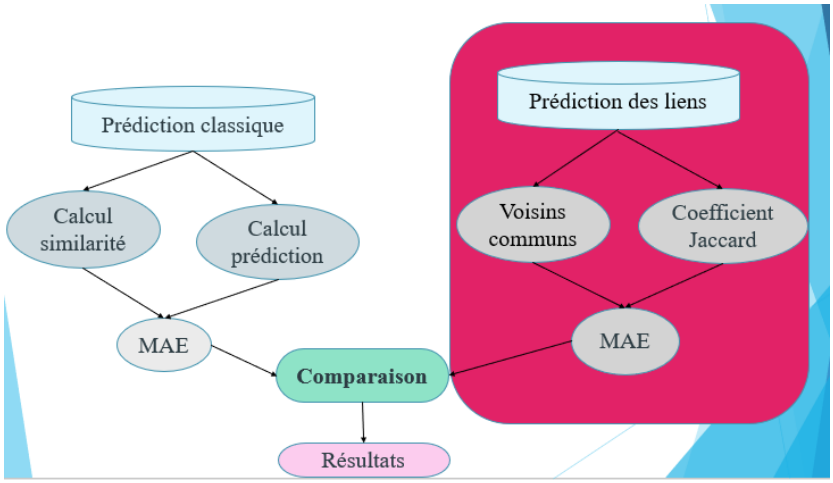

FIGURE 4.4 – Shéma sur l'approche proposée

4.3.2 Description de l'application

Dans cette section, nous allons présenter notre application réalisée à travers quelques interfaces graphiques et la sortie de chaque étape du travail.

Notre application est dotée d'une interface graphique facile à utiliser qui permet la mise en œuvre des principales visions théoriques étudiées ci-dessus.

L'interface d'accueil de notre application est donnée par la (figure 4.5).

FIGURE 4.5 – Interface d'accueil

A travers l'interface d'acceuil, avec le boutton "commencer" on pourra accéder au login. Ainsi, il y'a deux modes de login soit en mode admin, soit en mode user (figure 4.6)

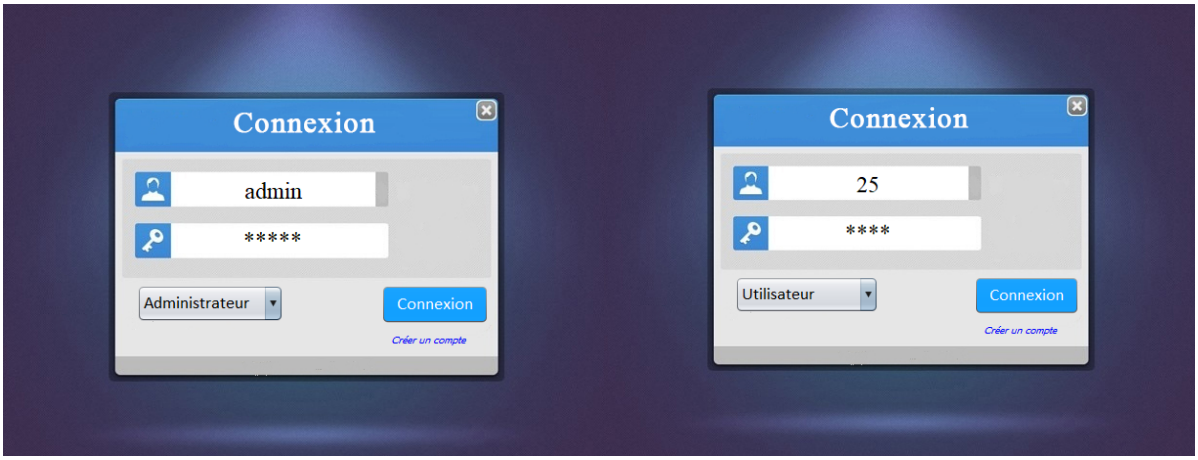

FIGURE 4.6 – Interface de connexion

Mode User

Un nouvau utilisateur peut créer un compte en remplissant ses informations personnelles.

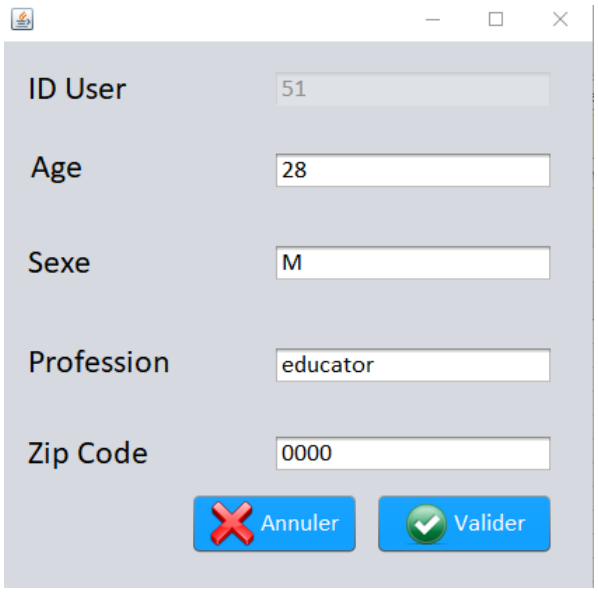

FIGURE 4.7 – Interface Ajouter User

L'accés a son compte lui permettre de le gérer.

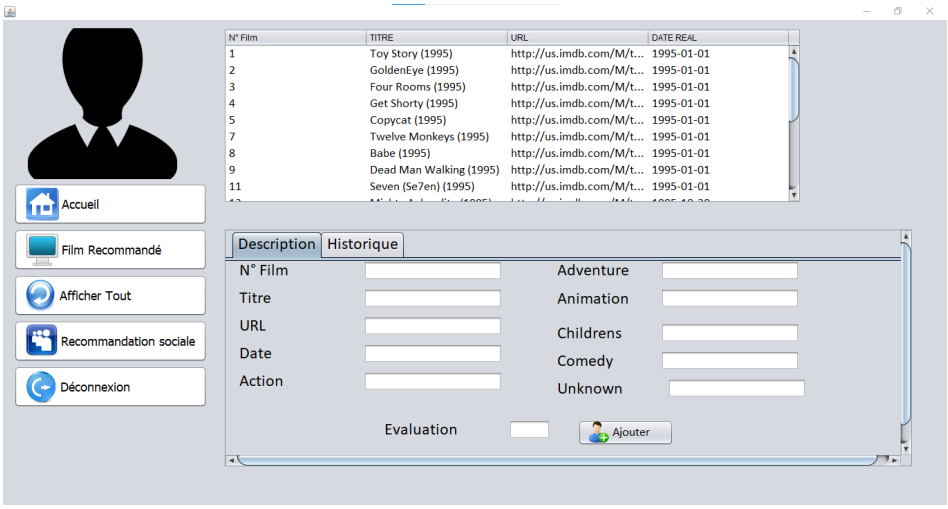

FIGURE 4.8 – Interface User

Mode Admin

A travers le login, l'admin pourra accéder à un menu sélectif des différents tailles de datsets montré dans la figure 4.9.

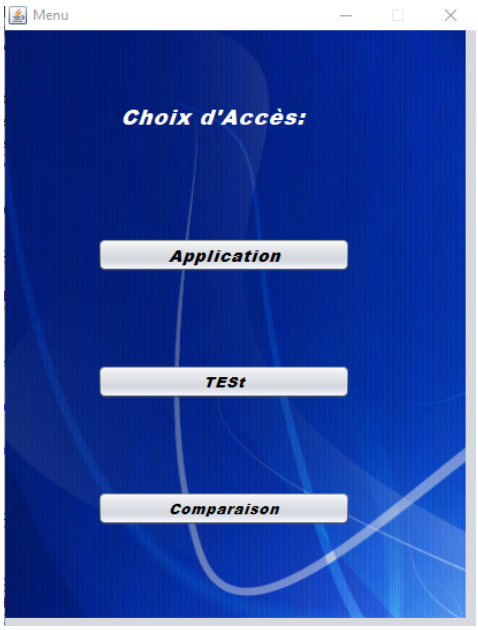

FIGURE 4.9 – Interface de menu

Tout d'abord le bouton "Application" nous menons à l'espace d'administrateur, qui comprend les différentes phases du processus de recommandation.

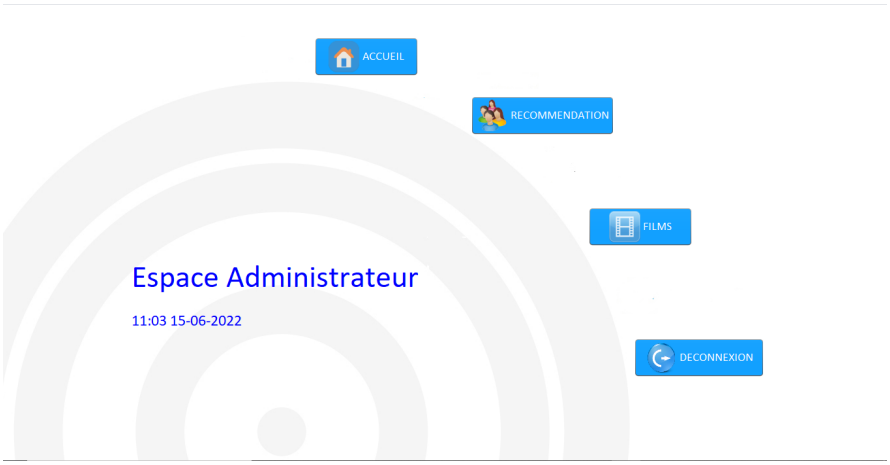

FIGURE 4.10 – Zone d'administrateur

Traitement d'items

Le boutton "Films" permet à l'admin de voir toute la liste des films comme il peut les ajouter, les modifier ou les supprimer.

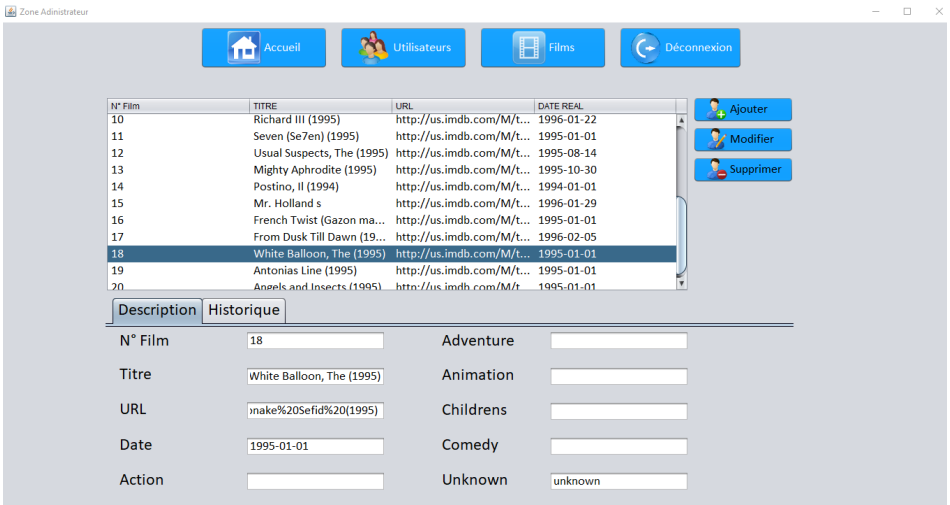

FIGURE 4.11 – Interface de la liste des films

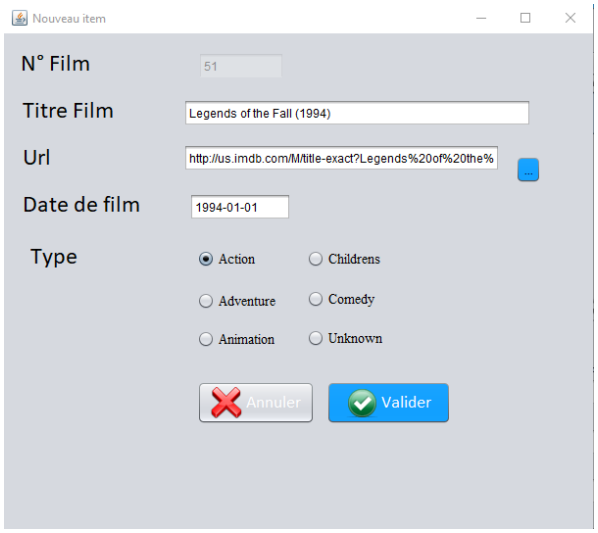

FIGURE 4.12 – Interface Ajouter Item

Ensuite, pour voir les calculs de prédiction classique on clique sur le boutton test et la fenêtre de la figure (4.13) s'affiche.

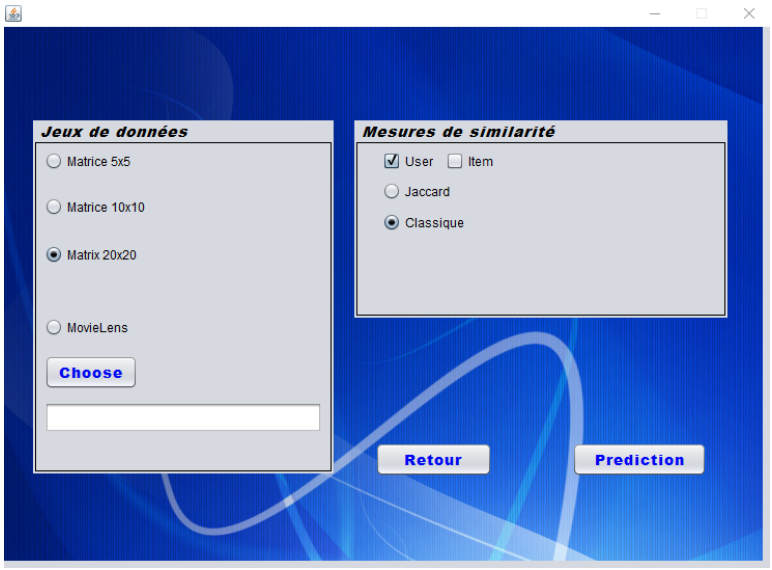

FIGURE 4.13 – Choix de la dimension du dataset

4.3.3 Fonction de prédiction classique

a) similarité

Il existe deux façon pour calculer la similarité : corrélation et cosinus, dans notre programme nous avons utilisé la fonction de similarité cosinus suivante :

$$cos(u,u^{'}) = \frac{\sum_{i \in I_{uu'}} r_{u,i} r_{u',i}}{\sqrt{\sum_{i \in I_u} {r_{u,i}}^2 \times \sum_{i \in I_{u'}} {r_{u',i}}^2}}$$

Où :

- u utilisateur
- i article
- $r_{u,i}$Note (rating) de l'utilisateur u pour l'item i.
- $I_{uu'}$ représente l'ensemble des items où on dispose des valeurs à la fois pour u et $u^{'}$

b) Prédiction

$$P_{a,i} = \bar{r}_a + \frac{\sum_{u=1}^{n} w(a,u)(r_{u,i} - \bar{r}_u)}{\sum_{u=1}^{n} |w(a,u)|}$$

Où :

- $\bar{r}_u$ Moyenne des notes attribuées par l'utilisateur u :

$$\bar{r}_u = \frac{1}{|I_u|} \sum_{i \in I_u} r_{u,i}$$

- a l'utilisateur considéré (actif).
- n nombre d'utilisateurs de la base dont le poids n'est pas nul, et ayant noté l'article i.
- w similarité cosinus.

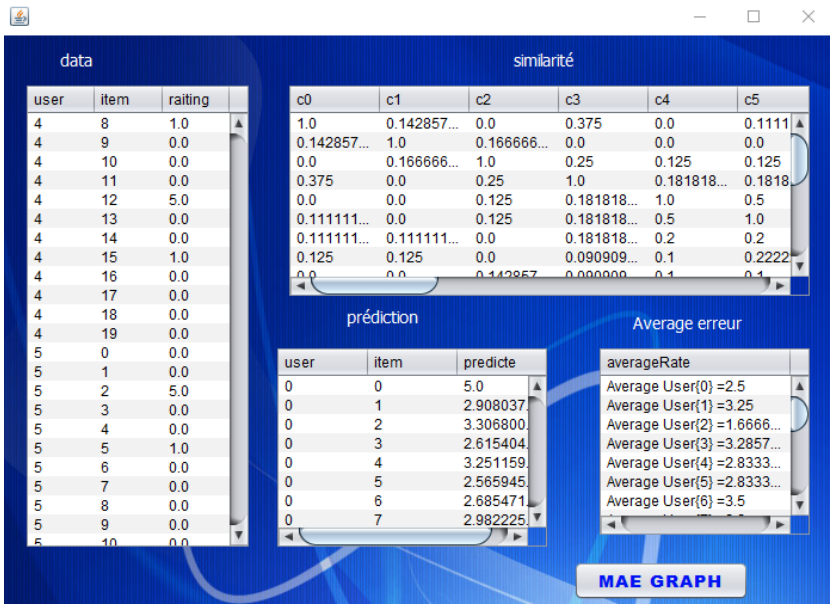

FIGURE 4.14 – Résultats de calcul de prédiction classique

c) MAE

La mesure de l'erreur absolue moyenne (MAE : Mean Absolute Error) formellement :

$$MAE = \frac{\sum_{(u,i)\in k} |r_{u,i} - \widehat{r}_{u,i}|}{|k|}$$

Où :

— $r_{u,i}$ est la vraie note donnée par u à i.
— $\widehat{r}_{u,i}$ la note prédite par le SR.
— k est l'ensemble des couples (user, item) pour lesquels la confrontation est effectuée.

Ensuite l'utilisateur passe au graphe MAE qui est afficher dans la figure (4.15) suivante :

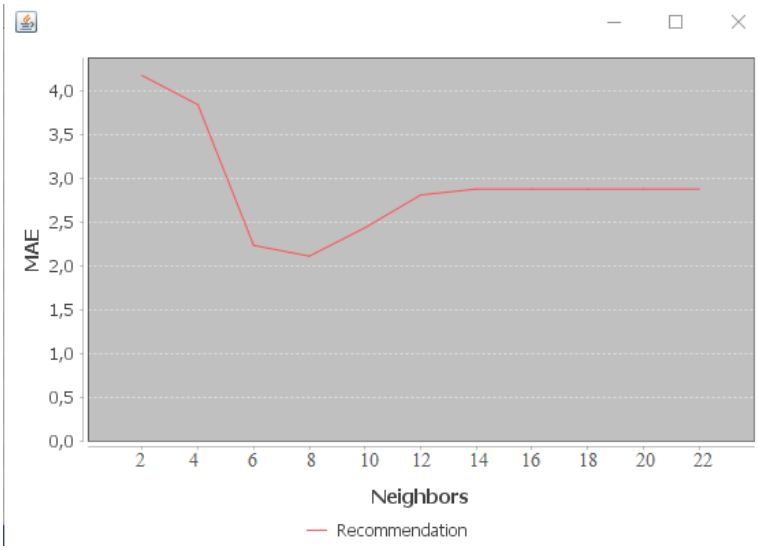

FIGURE 4.15 – Graphe MAE de prédiction classique

Il existe plusieur méthodes pour calculer la prédiction de lien, dans notre programme on a utilisé la Coefficient de Jaccard.

4.3.4 Coefficient de Jaccard

a) Voisins communs

Pour chaque nœud utilisateur cible u et chaque nœud item (u, i), nous allons calculer le nombre de voisins communs $CN_B(u, i)$:

$$CN_B(u,i) = |\left\{u' \in U | I'_u \cap I_U \neq \emptyset, i \in I'_u\right\}|$$

Où :

— U, est l'ensemble de tous les utilisateurs,
— I'_u, I_u sont les ensembles des items notés par les utilisateurs u' et u respectivement.

b) Jaccard

Pour chaque utilisateur cible u et chacun de ses items candidats i nous proposons de calculer le coefficient de Jaccard $JC_B(u,i)$,comme suit :

$$JC_B = \frac{(CN_B(u,i))}{|\{u' \in U | I_{u'} \cap I_u \neq \emptyset\}|} = \frac{|I_u \cap I_i|}{|I_u \cup I_i|}$$

Où :

- $CN_B(u,i)$ est le nombre de voisins communs entre u et i.
- U, est l'ensemble de tous les utilisateurs,
- I'_u, I_u, sont les ensembles des items notés par les utilisateurs u' et u respectivement.

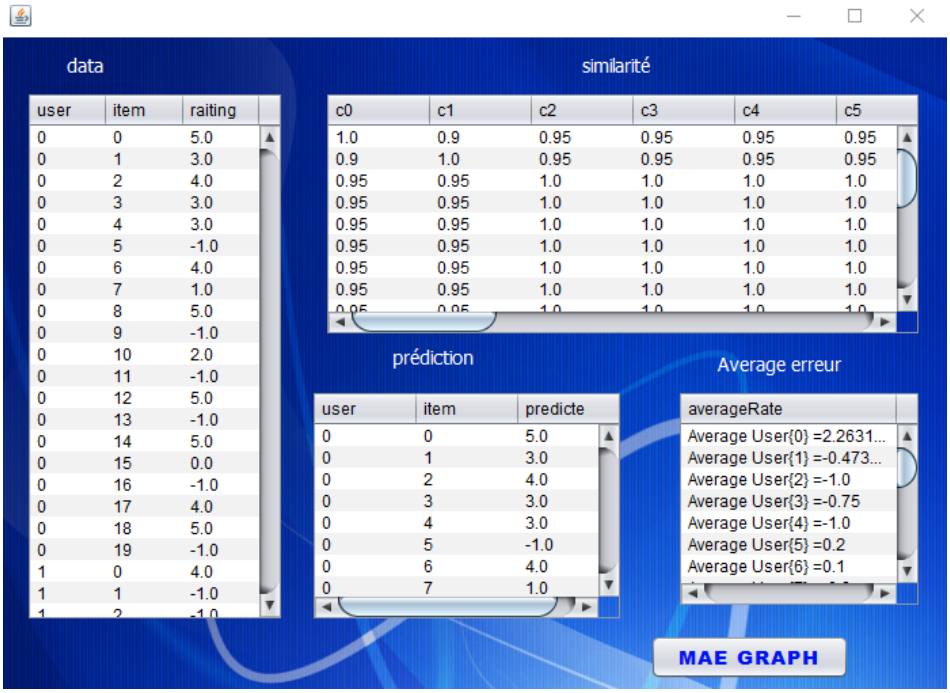

FIGURE 4.16 – Résultats de calcul Jaccard

Puis l'admin peut aussi faire les calcules on clique sur le boutton "Recommendation" et il charge le nombre de "user,items" de la base de données montrée par la figure (4.17).

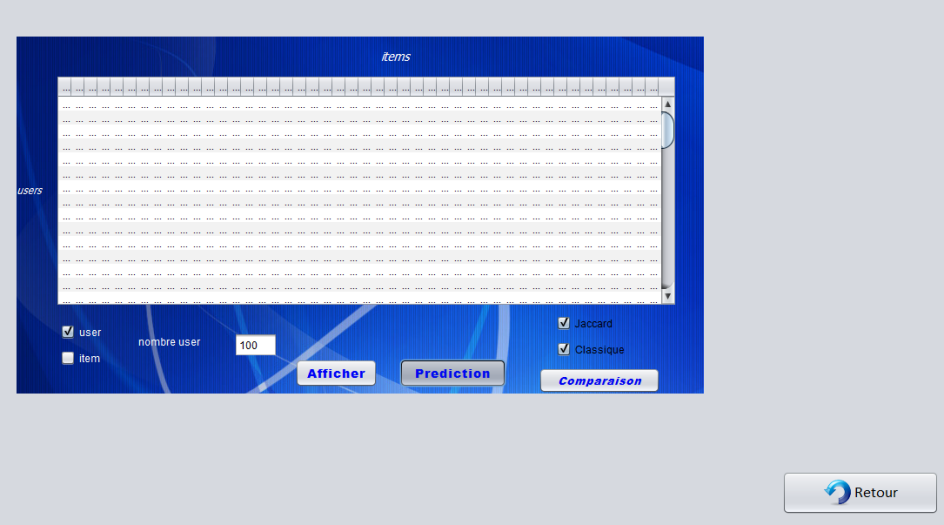

FIGURE 4.17 – Interface de Chargement de donnée

c) MAE

On cliquant sur le bouton "MAE Graphe", et le graphe s'affiche sur la figure (4.18) suivante :

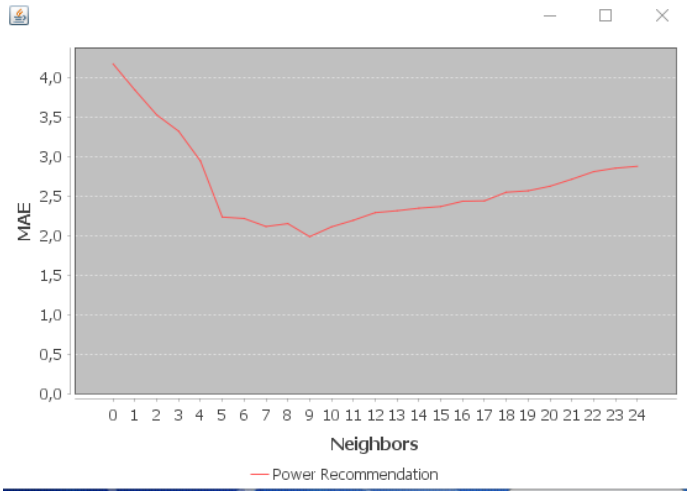

FIGURE 4.18 – Graphe MAE de prédiction de lien

Comparaison

le bouton comparaison dans l'interface de la figure(4.9) nous permet d'affiché les deux graphiques ensemble dans une seule interface, la figure(4.19).

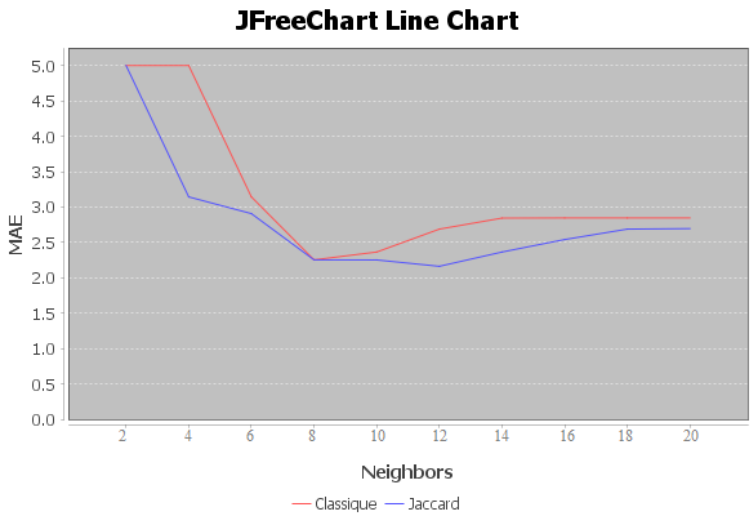

FIGURE 4.19 – Comparaison de MAE classique vs jaccard

4.4 Discussion

Après avoir visualiser les résultats avec la métrique MAE, nous avons constaté que les prédictions basées par l'approche liens (jaccard) sont satisfaisantes.

En effet, pour les valeurs de MAE de la prédiction de liens, on trouve que pour les 20 utilisateurs choisis, sa valeur minimale est de 1,98 et sa valeur maximale est de 4,16.

Pour les valeurs de MAE de prédiction classique, on trouve que pour les 20 utilisateurs choisis, sa valeur minimale est de 2,1 et sa valeur maximale est de 4,16.

Pour tout l'échantillon de données, Plus le nombre d'utilisateurs est élevé, plus la valeur de MAE se diminuer.

Ces résultats nous permettent de conclure que notre application de recommandation, nous donne la comparaison des deux graphiques, pour trouver a la fin que la prédiction des liens est la plus optimal.

Finalement, nous devons prendre en compte qu'il existe un nombre important des algorithmes de prédiction de liens qui sont basés sur le voisinage, mais pour des expérimentations réelles, plusieurs études ont montré qu'il n'existe pas une mesure de similarité absolue donne des bonnes prédictions pour n'importe quel réseau social.

4.5 Conclusion

Dans ce chapitre, nous avons présenté les outils de développement et les étapes d'implémentation et les différentes interfaces de notre application. Nous avons également présenté les résultats des calculs de similarité, prédiction, la moyenne d'erreur ainsi que les graphe MAE. Finalement, nous avons discuté les résultats obtenus.

Le but principal de cette implémentation est de faire la comparaison entre les graphes MAE de prédiction classique et MAE de prédiction des liens afin d'améliorer la qualité de recommandation.

Conclusion Générale

Les Systèmes de Recommandation sont des outils pertinents de recherche d'information et de filtrage qui vise à proposer aux utilisateurs des items qui pourraient les intéresser. La plupart des SR se basent sur l'analyse d'historique d'évaluation des items par les utilisateurs afin de prédire L'intérêt qu'un utilisateur peut porter à un item donné.

L'historique d'évaluation est souvent représenté sous forme d'une matrice $R : U \times I$ où un élément de cette matrice représente l'évaluation (note) qu'un utilisateur donne à un item. L'objectif De la recommandation est, alors, de prédire les valeurs manquantes dans cette matrice. Les techniques traditionnelles de système de recommandation souffrent de certains problèmes relatifs à cette matrice. Dans le cadre de ce projet, on s'intéresse à l'utilisation des approches de prédiction de liens pour le calcul de recommandation afin de faire face à la qualité médiocre des recommandations.

Nous nous intéressons, dans notre travail, à la prédiction de liens dans un graphe utilisateur-item en vue de la recommandation. Nous avons, dans un premier temps, exploré les corrélations existantes entre les différents items afin de construire un premier graphe mono-parti indépendant des utilisateurs et de leurs appréciations. Ceci nous a facilité l'intégration d'un nouvel item et d'un nouvel user comme ça nous a fourni des informations utiles qui ont guidé la phase de prédiction de liens.

Dans un second temps, nous avons procédé au calcul de la prédiction de liens entre les utilisateurs et les items. Nous avons adapté les métriques de proximité : Jaccard afin de prendre en considération les spécificités de notre graphe.

Les résultats que nous avons obtenus sont expérimentaux positifs et encourageants pour une approche symbolique au problème de la prédiction de liens.

Des perspectives d'amélioration de notre travail restent, toutefois, indispensables.

Bibliographie

[1] Luc-aurélien gauthier. Inférence de liens signés dans les réseaux sociaux, par apprentissage a partir d'interactions utilisateur. 2015. page 34.

[10] Matias. catherine. notes de cours : Analyse statistique de graphes. université pierre et marie et curie. 2015.

[11] Sigward. eric. introduction à la théorie des graphes. université paris. 2002.

[12] Luc-aurélien gauthier. inférence de liens signès dans les réseaux sociaux, par apprentissage à partir d'interactions utilisateur. 2015. page 49.

[13] Papadopoulos, s., vakali, a., and kompatsiaris, y. community detection in collaborative tagging systems. in community-built databases, pages 107–131. 2011.

[14] Ahcéne bonceur. representtion de graphe et programmation. 2012.

[15] Barthelemy, m. (2010). spatial networks. corr, abs/1010.0302.

[16] Wasserman, s. and faust, k. social network analysis : Methods and applications, volume 8. cambridge university press. 1994.

[17] Girvan, m. and newman, m. e. j. community structure in social and biological networks. proceedings of the national academy of sciences, 99(12) :7821–7826. 2002.

[18] Zachary, w. w. an information flow model for conflict and fission in small groups. journal of anthropological research, 33(4) :pp. 452–473. 1977.

[19] Adamic, l. a. and glance, n. 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. 2005.

[2] `https://hpi.de/fileadmin/user_upload/fachgebiete/mueller/courses/graphmining/GraphMining-01-Introduction.pdf`.
(derniere mise a jour 18/10/2016).

[20] Fortunato, s. community detection in graphs. physics reports, 486(3) :75–174. 2010.

[21] Bobadilla, j., ortega, f., hernando, a., et gutiérrez, a. «recommender systems survey». knowledge-based systems, 46, 109-132. 2013.

[22] Tang, l. and liu, h. community detection and mining in social media. synthesis lectures on data mining and knowledge discovery. morgan claypool publishers. 2010.

[23] Aggarwal, c. c. and reddy, c. k., editors data clustering : Algorithms and applications. crc press. 2014.

[24] Katz., l. a new status index derived from socimetric analysis. psychmetrika, 18(1), 18(1) :39–43. 1953.

[25] Fouss, f., pirotte, a., renders, j.-m., and sarens, m. random-walk computation of similarities between nodes of a graph with application to collaborative recommandation. ieee transactions on knowledge and data engineering, 19(3) :355–369. 2007.

[26] Pons, p. and latapy, m. computing communities in large networks using random walks. j. graph algorithms appl., 10(2) :191–218. 2006.

[27] Bajec, m. robust network community detection using balanced propagation. 2011.

[28] Khorasgani, r. r., chen, j., and zaiane, o. r. top leaders community detection approach in information networks. in 4th sna-kdd workshop on social network mining and analysis, washington d.c.,shah, d. and zaman, t. community detection in networks : The leader-follower algorithm. in workshop on networks across disciplines in theory and applications, nips. 2010.

[29] Clauset, a. finding local community structure in networks. physical review e. 2005.

[3] Tarissan, f., quoitin, b., mérindol, p., donnet, b., pansiot, j.-j., and latapy, m. towards a bipartite graph modeling of the internet topology. computer networks, 57(11) :2331–2347. 2013.

[30] Kanawati, r. licod : Leaders identification for community detection in complex networks. in socialcom/passat, pages 577–582. ieee. 2011.

[31] Belkin, n. j. and croft, w. b. information filtering and information retrieval : two sides of the same coin ? communications of the acm 35, 12, 29–38. 1992.

[32] Rich, e. user modeling via stereotypes*. cognitive science, 3(4) :329–354. 1979.

[33] Goldberg d., nichols d., oki b. m., terry d., «using collaborative filtering to weave an information tapestry», communications of the acm, vol. 35, no 12, p. 61-70. 1992.

[34] Resnick p., iacovou n., suchak m., bergstrom p., riedl j,,furuta, r. and neuwirth, c., «grouplens : An open architecture for collaborative filtering of netnews», proc. of the 1994 conference on computer supported collaborative work, eds. acm press, new york, p. 175-186. 1994.

[35] Shardanand u., maes p., «social information filtering : Algorithms for automating 'word of mouth'», proc. conf. human factors in computing systems, 1995.

[36] Hill, w., stead, l., rosenstein, m., and furnas, g. recommending and evaluating choices in a virtualcommunity of use. in proceedings of the sigchi conference on human factors in computing systems, pages 194–201. acm press/addison-wesley publi-shing co. 1995.

[37] Adomavicius g., tuzhilin a., «toward the next generation of recommender systems : A survey of the state-of-the-art and possible extensions», ieee transactions on knowledge and data engineering, vol. 17, no. 6, juin 2005.

[38] Berrut c., denos n., «filtrage collaboratif», assistance intelligente à la recherche d'informations, hermes - lavoisier, chapter 8, pp30, 2003.

[39] M. a. hasan, m. j. zaki. a survey of link prediction in social networks. in social network data analytics, pages 243–275. springer, 2011.

[4] Roth, c., kang, s. m., batty, m., and barthelemy, m. long-time limit of world subway networks. corr, abs/1105.5294. 2011.

[40] Newman, m. clustering and preferential attachment in growing networks. physical review letters, 64(025102).2001.

[41] D. liben-nowell, j. kleinberg. the link-prediction problem for social networks. j. am. soc. inf. sci. technol., 58(7) :1019–1031, 2007.

[42] L. lu, t. zhou. link prediction in complex networks : A survey. physica a, 390(6) :1150–1170, 2011.

[43] Lops, p., de gemmis, m., and semeraro, g. content-based recommender systems : State of the art and trends. in recommender systems handbook, pages 73–105. springer. 2011.

[44] 3] michalski, r. s., carbonell, j. g., and mitchell, t. m. machine learning : An artificial intelligence approach. springer science business media. 2013.

[45] Breese j., heckerman d., kadie k., «empirical analysis of predictive algorithms for collaborative filtering», proc. 14th conf. uncertainty in artificial intelligence, july 1998.

[46]] wang, y., stash, n., aroyo, l., hollink, l., and schreiber, g. using semantic relations for content-based recommender systems in cultural heritage. in proceedings of the 2009 international conference on ontology patterns-volume 516, pages 16–28. ceur-ws. org. 2009.

[47] Schafer, j. b., frankowski, d., herlocker, j., and sen, s. (2007). collaborative filtering recommender systems. in the adaptive web, pages 291–324. springer.

[48] Ricci, f., rokach, l., and shapira, b. introduction to recommender systems handbook. springer. 2011.

[49] Liben-nowell, d. et kleinberg, j. the link prediction problem for social networks. in proceedings of the 12th international conference on information and knowledge management. acm. 2003.

[50] Katz, l. a new status index derived from sociometric analysis. psychmetrika 18, 39. 1953. krebs, v. mapping networks of terrorist cells. j. amer. soc. in- form. sci. 27 24(3), 43–52. 2002.

[51] Budalakoti, s. and bekkerman, r. bimodal invitation-navigation fair bets model for authority identification in a social network. in proc. intl. world wide web conference (www), pages 709–718. 2012.

[52]] chen, k., chen, t., zheng, g., jin, o., yao, e., and yu, y. collaborative personalized tweet recommendation. in proc. intl. conf. on research and development in information retrieval (sigir), pages 661–670. 2012.

[53] Hannon, j., bennett, m., and smyth, b. re- commending twitter users to follow using content and collaborative filte-ring approaches. in proc. intl. conf. on recommender systems (rec-sys), pages 199–206. 2010.

[54] Diaz-aviles, e., drumond, l., gantner, z., schmidt-thieme, l., and nejdl, w. what is happening right now ... that interests me ? : online topic discovery and recommendation in twitter. in proc. intl. conf. on information and knowledge management (cikm), pages 1592–1596. 2012.

[55] Liang, h., xu, y., tjondronegoro, d., and christen, p. time-aware topic recommendation based on micro-blogs. in proc. intl. conf. on information and knowledge management (cikm), pages 1657–1661. 2012.

[56]] kywe, s. m., hoang, t.-a., lim, e.-p., and zhu, f. on recommending hashtags in twitter networks. in proc. intl. conf. on social informatics (socinfo), pages 337–350. 2012.

[57] Chaoji, v., ranu, s., rastogi, r., and bhatt, r. recommendations to boost content spread in social networks. in proc. intl. world wide web conference (www), pages 529–538. 2012.

[58] Lempel, r. and moran, s. salsa : The stochastic approach for link-structure analysis. acm transactions on information systems, 19(2) :131–160. 2001.

[59] Gupta, p., goel, a., lin, j., sharma, a., wang, d., and zadeh, r. wtf : the who to follow service at twitter. in proc. intl. world wide web conference (www), pages 505–514. 2013.

[6] Sigward. eric. introduction à la théorie des graphes. université paris. 2012.

[60] Lumineau n., «un tour d'horizon du filtrage collaboratif», travail réalisé dans le cadre de l'as personnalisation de l'information, laboratoire d'informatique de paris 6, 2002.

[61] Resnick p., iacovou n., suchak m., bergstrom p., riedl j, furuta, r. and neuwirth, c., «grouplens : An open architecture for collaborative filtering of netnews», proc. of the 1994 conference on computer supported collaborative work, eds. acm press, new york, 1994, p. 175-186. 1994.

[62] Maltz d., ehrlich e., «pointing the way : Active collaborative filtering», proc. of the sigchi conference on human factors in computing systems (chi'95), usa, 1995, p.202-209. 1995.

[63] Miller d., maltz j.l., herlocker l.r, gordan a., riedl j.a., konstan b.n., «grouplens : applying collaborative filtering to usenet news». 1997.

[64] Chan, p. a non-invasive learning approach to building user profiles. web usage analysis and user profiling. 1999.

[65] Nickel, m., k. murphy, v. tresp, et e. gabrilovich a review of relational machine learning for knowledge graphs. proc. ieee 104(1), 11–33. 2016.

[66] Rajaraman, a. et j. d. ullman mining of massive datasets. (2010-2011).

[67] Ricci, f., l. rokach, b. shapira, et p. b. kantor (eds.) recommender systems handbook. springer. 2011.

[68] Allali, o., c. magnien, et m. latapy link prediction in bipartite graphs using internal links and weighted projection. proceedings of the third international workshop on network science for communication networks (netsci- com). 2011.

[69] Oussama allali .structure et dynamique des graphes de terrain bipartis : liens internes et prédiction de liens. 2011. *Université Pierre et Marie Curie (UPMC).*

[7] Canu,mael. «détection de communautés orientée sommet pour des réseaux mobiles opportunistes sociaux». doctoral dissertation, université pierre et marie curie-paris vi. 2017.

[70] Martin g. everett et stephen p. borgatti : book ; analyzing social networks 30-apr-2013.

[71] Silveira, thiago, min zhang, xiao lin, yiqun liu, et shaoping ma. «how good your recommender system is ? a survey on evaluations in recommendation.» international journal of machine learning and cybernetics. 2019.

[72] Resnick, p. varian, h. r., recommender systems - introduction to the special section.. commun. acm 40(3), pp. 56-58. 1997.

[73] Candillier, l., jack, k., fessant, f. meyer, f., state-of-the-art recommender systems. collaborative and social information retrieval and access : Techniques for improved user modeling, pp. 1-22. 2009.

[74] Bothorel, c., analyse de réseaux sociaux et recommandation de contenus non populaires. revue des nouvelles technologies de l'information (rnti), vol. a.5. 2011.

[75] Ma, h. et al., recommender systems with social regularization. new york, s.n., pp. 287-296. 2011.

[76] Symeonidis, p., tiakas, e. manolopoulos, y., product recommendation and rating prediction based on. new york, usa, s.n., pp. 31-68. 2011.

[77] Vasuki, v., natarajan, n., lu, z. dhillon, i., affiliation recommendation using auxiliary networks. s.l., s.n., pp. 103-110. 2010.

[78] Ye, m., yin, p., lee, w.-c. lee, d.-l., exploiting geographical influence for collaborative point-of-interest recommendation. new york, ny, usa, acm, pp. 325-334. 2011.

[79] Tang, j., chang, s., aggarwal, c., and liu, h. 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. 2015.

[8] Jean-philippe attal. nouveaux algorithmes pour la détection de communautés disjointes et chevauchantes basés sur la propagation de labels et adaptés aux grands graphes. informatique [cs]. 2017.

[80] Resnick p., iacovou n., suchak m., bergstrom p., riedl j,, furuta, r. and neuwirth, c., «grouplens : An open architecture for collaborative filtering of netnews», proc. of the 1994 conference on computer supported collaborative work, eds. acm press, new york, p. 175-186. 1994.

[81] Breese j., heckerman d., kadie k., «empirical analysis of predictive algorithms for collaborative filtering», proc. 14th conf. uncertainty in artificial intelligence, july 1998.

[82] Page l, brin s, motwani r, winograd t. the pagerank citation ranking : Bringing order to the web. stanford29 university, stanford, ca 94305-9025, usa : Stanford infolab- digital libraries, technical report, 1999.

[83] Giannoulakis s, tsapatsoulis s. filtering instagram hashtags through crowdtagging and the hits algorithm. in :16 ieee transactions on computational social systems 6(3) : 592-603. doi : 10.1109/tcss.2019.2914080. 2019.

[84] Smith b, linden g. two decades of recommender systems at amazon. com. in : Ieee internet computing 2017 ;27 21(13) : 12-18. doi : 10.1109/mic. 72. 2017.

[85] Shams b, haratizadeh s. sibrank : Signed bipartite network analysis for neighbor-based collaborative ranking.29 physica a : Statistical mechanics and its applications 2016 ; 458 : 364-377. doi : 10.1016/j.physa. 2016.04.25.

[86] Cordasco, g. and gargano, l. label propagation algorithm : a semi-synchronous approach. ijsnm, 1(1) :3–26. 2012.

[87] Papadopoulos, s., kompatsiaris, y., vakali, a., and spyridonos, p. community detection in social media - performance and application considerations. data min. knowl. discov., 24(3) :515–554. 2012.

[88] Corlette, d. and iii, f. m. s. link prediction applied to an open large-scale online social network. in ht, pages 135–140. 2010.

[89] Gregor, s. finding overlapping communities in networks by label propagation. new journal of physic, 12 :103018. 2010.

[9] `https://perso.liris.cnrs.fr/samba-ndojh.ndiaye/fichiers/App_Graphes.pdf`.

[90] Raghavan, u. n., albert, r., and kumara, s. near linear time algorithm to detect community structures in large-scale networks. physical review e, 76 :1–12. 2007.

[91] Xie, j. and szymanski, b. k. community detection using a neighborhood strength driven label propagation algorithm. in procof ieee .network science workshop. 2011.

[92] Kanawati, r. seed-centric approaches for community detection in complex interaction networks : A comparative review.in complexity in social systems : from data to models, cergy. 2013.

[93] Chen, j., zaïane, o. r., and goebel, r. local community identification in social networks. in [memon and alhajj, 2009], pages 237–242. 2009.

[94] Ngonmang, b., tchuente, m., and viennet, e. local community identification in social networks. parallel processing letters, 22(1). 2012.

[95] Nakamura A., Abe N., «Collaborative Filteringusing weighted majority prediction algorithms», Proceedings of ICML98, pages395403. Morgan Kaufman Eds. 1998.

[96] Delgado J., Ishii N., «Memory-based weighted-majority prediction for recommender systems», In Proceedings of the ACM SIGIR-99,Recommender Systems Workshop, August 1999, UCBerkeley, pages 1-5, 1999.

[97] Wu F S Bhattacharyya P, Garg A. Analysis of user keyword similarity in online social networks. Social Network Analysis and Mining, 2011.

[98] Kleinberg J et al Anderson, Huttenlocher D. Effects of user similarity in social media. In Proceedings of the 5th ACM International Conference on Web Search and Data Mining (WSDM'12), 2012.

[99] Liang Xiang, Quan Yuan, Shiwan Zhao, Li Chen, Xiatian Zhang, Qing Yang, and Jimeng Sun. Temporal recommendation on graphs via long-and short-term preference fusion. In Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 723–732. ACM, 2010.