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

UNIVERSITE IBN KHALDOUN - TIARET

MEMOIRE

Présenté à :

FACULTÉ DES MATHEMATIQUES ET DE l'INFORMATIQUE
DÉPARTEMENT D'INFORMATIQUE

Pour l'obtention du diplôme de :

MASTER

Spécialité : Génie Informatique

Par :

Hadil Aboubaker

Mandi Nour-El-Yakine

Sur le thème

Système de recommandation à base de communautés

Soutenu publiquement le 19/09/2022 à Tiaret devant le jury composé de :

M. ALEM Abdelkader	MAA	Université Ibn-Khaldoun	Président
M. KOUADRIA Abderrahmane	MCB	Université Ibn-Khaldoun	Encadrant
M^{me}. LAAREDJ Zohra	MAA	Université Ibn-Khaldoun	Examinatrice

2021-2022

Remerciement

Notre parfaite gratitude et nos remerciements sont tout d'abord à ***Allah****, le clément et le miséricordieux, qui nous a donné la force, le courage et la volonté pour mener à bien ce modeste travail.*

C'est avec une profonde reconnaissance et une considération particulière, que nous remercions notre encadreur, ***Mr KOUADRIA Abderrahmane****, de nous avoir accordé sa confiance pour réaliser ce travail ainsi que pour son aide et son suivi permanent. On le remercie également pour sa patience, ses encouragements et ses précieux conseils.*

Nous remercions également les membres du jury qui ont bien voulu accepté de juger ce travail.

On tient à exprimer notre profond amour et immense gratitude à nos ***parents*** *pour leur contribution, leur soutien et leur patience.*

Nous adressons nos sincères et chaleureux remerciements aux personnes qui ont contribué de près ou de loin à l'élaboration de ce mémoire et nous ont aidé à le réaliser.

Merci

Dédicace

Avec l'expression de ma reconnaissance, je dédie ce modeste travail à ceux qui, quels que soient les termes embrassés, je n'arriverais jamais à leur exprimer mon amour sincère.

❖ *A l'homme, mon précieux offre du dieu, qui doit ma vie, ma réussite et tout mon respect ,Tu as toujours été à mes côtés pour me soutenir et m'encourager. Que ce travail traduit ma gratitude et mon affection : mon cher père* **Djilali**,.

❖ *A la femme qui a souffert sans me laisser souffrir, qui n'a jamais dit non à mes exigences et qui n'a épargné aucun effort pour me rendre heureuse : mon adorable mère* **Khaira**,.

❖ *A mes très chers frères Cherif Chamess Edin et Adel et mes belles sœurs Fatima, Hind, Siham et Nawal*

❖ *A mes très cher neveux et mes belle nièces " Khouloud , Jouri*

❖ *A mes belles « Asma & Fatima »*

❖ *Sans oublier mon binôme Aboubaker pour son soutien moral, sa patience et sa compréhension tout au long de ce projet.*

Résumé

Les systèmes de recommandation jouent un rôle important en suggérant des informations pertinentes aux utilisateurs. Dans ce travail, nous introduisons l'information démographique de l'utilisateur au niveau communautaire comme étant une nouvelle dimension pour les recommandations et nous présentons un système de recommandation utilisant des approches de filtrage collaboratif et de détection communautaire. Nous utilisons (1) un algorithme de détection de communautés pour découvrir les communautés des utilisateurs en analysant le graphe de similarité démographique utilisateur-utilisateur et (2) un filtrage collaboratif basé sur la matrice utilisateur-item pour la prédiction des notes. Notre approche améliore l'évolutivité, le problème de démarrage à froid et de manque de données du système de recommandation basé sur le filtrage collaboratif. Nous avons effectué des expériences sur l'ensemble de données de MovieLens, pour prédire la note du film et produire les meilleures recommandations pour les nouveaux utilisateurs (démarrage à froid). Les résultats sont comparés avec un système de filtrage collaboratif basé sur l'utilisateur.

Mots clés : Système de recommandation, Filtrage collaboratif, détection de communautés, méthode de Louvain.

Abstract

Recommender systems play an important role in suggesting relevant information to users. In this work, we introduce user demographic information at the community level as a new dimension for recommendations and present a recommender system using collaborative filtering and community detection approaches. We use (1) a community detection algorithm to discover user communities by analysing the user-user demographic similarity graph and (2) collaborative filtering based on the user-item matrix for score prediction. Our approach improves the scalability, cold start and data gap problem of the collaborative filtering-based recommender system. We performed experiments on the MovieLens dataset, to predict the movie rating and produce the best recommendations for new users (cold start). The results are compared with a user-based collaborative filtering system.

Keywords: Recommender system, Collaborative filtering, Community detection, Louvain method.

TABLES DES MATIERS

Liste des figures

Liste des tableaux

Liste des abréviations

AG	Algorithmes génétiques
CONGA	Cluster-Overlap Newman Girvan Algorithm
CV	Communauté virtuelle
FC	Filtrage Collaboratif
FCBU	Filtrage collaboratif basé utilisateur
FCI	Filtrage collaboratif inversé
GCE	Greedy Clique Expansion
GN	Girvan Newman
IF	Fréquence de l'item
ICF	Fréquence communautaire inverse
LPA	Label Propagation Algorithm
MAE	Mean Absolute Error
MPC	Méthode de percolation de clique
RMSE	Root Mean Squared Error
RSL	Réseaux Sociaux en Ligne
SITP	Service d'information touristique personnalisé
SLPA	Speaker listener label propagation Algorithm
SR	Système de Recommandation
SRS	Services de réseaux sociaux
WLPA	Weighted Label Propagation Algorithm

Introduction Générale

Introduction Générale

Avec le volume, la complexité et la dynamique croissants de l'information en ligne, la croissance explosive de l'information disponible sur Internet déroute souvent les utilisateurs. Les systèmes de recommandation (SRs) sont une solution clé et efficace pour surmonter le problème de surcharge d'information. Ces systèmes sont des outils de filtrage d'informations utiles pour guider les utilisateurs de manière personnalisée dans la découverte de produits ou de services pouvant provenir d'un large éventail d'options possibles. Les SRs jouent un rôle important dans les systèmes d'information pour améliorer les affaires et faciliter la prise de décision. En général, la liste de suggestions est basée sur les préférences de l'utilisateur, les caractéristiques des items, les interactions passées des utilisateurs avec les items et certaines informations supplémentaires telles que les données temporelles et spatiales. Les modèles de recommandation sont principalement classés en systèmes de filtrage collaboratif (FC), basés sur le contenu et hybrides basés sur les types de données d'entrée. Cependant, ces modèles ont leurs limites pour traiter les problèmes de démarrage à froid et de rareté des données ainsi que l'équilibre de la qualité des suggestions en fonction de différents critères.

Le système de recommandation est une partie importante de l'industrie. C'est un outil essentiel pour promouvoir les ventes et les services pour de nombreux sites Web en ligne et applications mobiles. Par exemple, 80% des vidéos visionnées sur Netflix proviennent du système de recommandation et 60 % des clics vidéo sur YouTube proviennent des suggestions de la page d'accueil [1]. En analysant son comportement d'utilisateur, le système de recommandation propose les items les plus appropriés (données, informations, biens, etc.). Ce système est une approche conçue pour traiter les problèmes d'un volume important et croissant d'informations et aide son utilisateur à atteindre son objectif plus rapidement dans le grand volume d'informations. Dans les SRs, nous essayons d'identifier et de suggérer l'item le plus approprié pour répondre aux préférences de l'utilisateur en devinant la pensée de l'utilisateur grâce aux informations que nous avons sur ses utilisateurs similaires et leurs opinions.

Les êtres humains ont tendance à s'associer avec des personnes ayant des goûts et des choix similaires. Cela conduit à la création de groupes d'intérêts communs appelés communautés, formés par un ensemble d'entités étroitement liées qui peuvent être des objets ou des individus. Ceux-ci peuvent être qualifiés de sous-groupes ou de clusters cohésifs. Le processus d'identification de ces

structures dans un réseau ou ensemble de données est connu sous le nom de détection de communauté et a été appliqué à divers domaines de recherche tels que les réseaux biologiques et de collaboration. Diverses approches de détection de communautés ont été proposées dans la littérature.

Ce projet présente l'idée de générer des recommandations individuelles basées sur la communauté pour l'utilisateur. La détection de communautés et le processus de filtrage des systèmes de recommandation ont des motifs similaires, c'est-à-dire pour trouver des connexions dans un ensemble d'items ou de personnes. Un système de recommandation basé sur l'utilisateur fonctionne pour filtrer et récupérer des informations auprès d'utilisateurs partageant les mêmes idées tandis qu'un système de recommandation basé sur le filtrage d'items découvre une similitude dans un ensemble d'items. Les techniques de détection de communautés sont analogues à la méthode de regroupement (clustering) qui nous donne un groupe d'utilisateurs similaires et donc des recommandations peuvent être générées pour cet ensemble d'utilisateurs similaires.

L'objectif de ce travail est de proposer une approche de recommandation basée sur la détection de communautés pour le système de recommandation collaboratif. La méthode de détection de communautés de Louvain a été appliquée pour découvrir les communautés dans l'ensemble de données de MovieLens. La méthode de génération de recommandations est basée sur l'idée d'utiliser le score d'item IF-ICF [2] (Frequency-Inverse Community Frequency) de chaque item dans la communauté de l'utilisateur cible. Les scores IF aident à trouver l'ensemble d'items qui sont uniques à une communauté particulière. Les valeurs de l'ICF sont inversement proportionnelles au nombre de communautés dans lesquelles un item a été évalué. ICF est utilisé pour calculer le caractère unique de l'item dans les communautés. Les scores IF-ICF des items sont en outre utilisés pour trouver les scores de prédiction des items non vus par l'utilisateur afin de présenter un ensemble de meilleures recommandations à l'utilisateur. Un prototype du système est développé en python et une analyse expérimentale a été réalisée pour le domaine du film.

Ce mémoire est organisé de la façon suivante :

Chapitre I : Ce chapitre présente un état de l'art sur les systèmes de recommandation collaboratif.

Chapitre II : Ce chapitre décrit les algorithmes de détection de communautés existantes. Ensuite, il présente un état de l'art sur les systèmes de recommandation à base de communautés.

Chapitre III : Dans ce chapitre, nous mettons la présentation et l'implémentation de notre solution proposée. Ensuite, nous discutons les résultats trouvés et les évaluations élaborées.

Chapitre I

Système de recommandation collaboratif

1.1 Introduction

Les systèmes de recommandation (SRs) sont devenus une partie intégrante de notre vie quotidienne en nous facilitant la prise de décision. L'utilisation de recommandations a cependant accru la demande d'explications suffisamment convaincantes pour aider les utilisateurs à faire confiance aux recommandations fournies. Les utilisateurs souhaitent que les recommandations soient compréhensibles et personnalisées en fonction de leurs besoins et préférences individuels.

Nous présentons au sein de ce chapitre, le contexte de notre travail, où nous définissons en premier lieu les systèmes de recommandation ; Ensuite, nous exposons différents domaines d'application des SR. Ainsi que nous expliquons les différents types de recommandation. Enfin nous détaillons les travaux de recommandations selon le type de données à recommander.

1.2 Système de Recommandation (SR)

Aujourd'hui, les systèmes de recommandation jouent un rôle essentiel dans notre vie quotidienne via de nombreux sites Web et applications tels que YouTube, Facebook, Netflix et Amazon. Ils sont apparus comme une solution au problème de surcharge d'information et ont reçu beaucoup d'attention et d'utilisation de la part de la communauté scientifique. Les systèmes de recommandation sont des outils de recherche et de filtrage d'informations qui aident les utilisateurs à découvrir des items pertinents et à faire de meilleurs choix lors de la recherche de produits ou de services tels que des films, des livres, des vacances ou des produits électroniques. L'objectif fondamental d'un système de recommandation est de fournir des suggestions personnalisées qui peuvent aider les utilisateurs dans le processus de prise de décision.

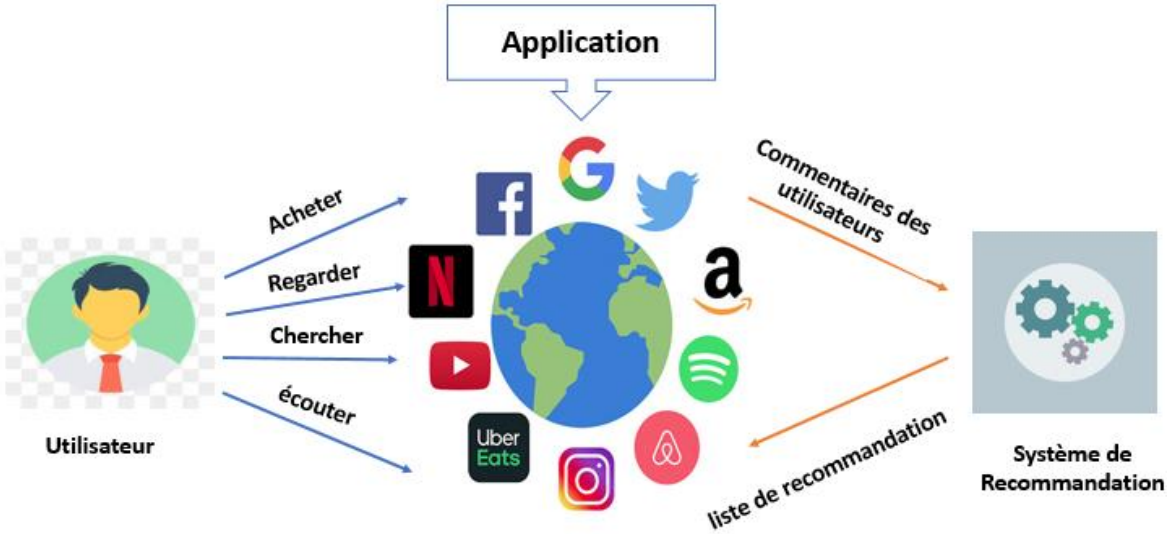


Figure 1.1 : *Schéma expliquant le fonctionnement d'un système de recommandation [3]*

Formellement, un système de recommandation est un outil de filtrage de données où il existe un ensemble d'utilisateurs *U*, un ensemble d'items *I* et une fonction d'utilité *h* qui dénote l'intérêt d'un utilisateur pour un produit. Ainsi, il est possible de voir les systèmes de recommandation comme une fonction à maximiser :

$$\forall u \in U, i'u = argmax\big(h(u,i)\big)\ i\epsilon \boldsymbol{I} \qquad (1.1)$$

I : représente un ensemble d'items pour *u* et la fonction d'utilité *h* est particulière au type d'approche utilisée. De plus, cette fonction varie selon le type d'information utilisé, implicite ou explicite. L'objectif est donc d'estimer la fonction d'utilité d'un item *i* pour un utilisateur *u*.

Certaines définitions sont devenues courantes dans le domaine de recommandation, l'utilisateur qui a effectué certaines actions et qui doit être recommandé, puis l'item à recommander, et enfin la note, qui représente à quel point l'utilisateur est intéressé par l'item. Ainsi, la collection (utilisateur, item, évaluation) est au cœur des systèmes de recommandation. Les notations peuvent prendre plusieurs formes. Il peut s'agir d'une valeur de : un à cinq (score de 1 à 5) ou de toute autre forme indiquant un niveau d'intérêt (aime, n'aime pas). Les données unaires sont un type de notation qui n'est pas directe. Par exemple, lorsqu'un utilisateur achète un item, il peut ne pas l'évaluer, mais

l'acheter est généralement une indication que l'utilisateur est intéressé par l'item (Amazon suppose que l'utilisateur a évalué 5 s'il a acheté l'item).

1.3 Domaines d'application des systèmes de recommandation

Le système de recommandation a été étendu et utilisé dans divers domaines de service. Dans cette section, nous avons l'intention d'analyser comment les modèles et technologies de recommandation pour divers systèmes de recommandation sont étudiés et utilisés en fonction des caractéristiques et de l'objectif du domaine de service réel. Sur la base des articles collectés pour analyse, les domaines de service dans lesquels le système de recommandation a été utilisé ont été classés en sept catégories principales : service de streaming, service de réseau social, service de tourisme, service de commerce électronique, service de santé, service d'éducation, service d'information académique. Les sept catégories principales sont divisées en fonction de la liste des services qu'utilisent un système de recommandation avec un nombre croissant d'utilisateurs ou une valeur commerciale croissante, et la liste des services qui apparaissent fréquemment lorsque le « système de recommandation » du moteur de recherche Google Scholar est recherché en tant que mot-clé. La figure 2 est un résumé visuel de la liste des services principalement utilisés dans le système de recommandation qui sera décrit dans cette section [4].

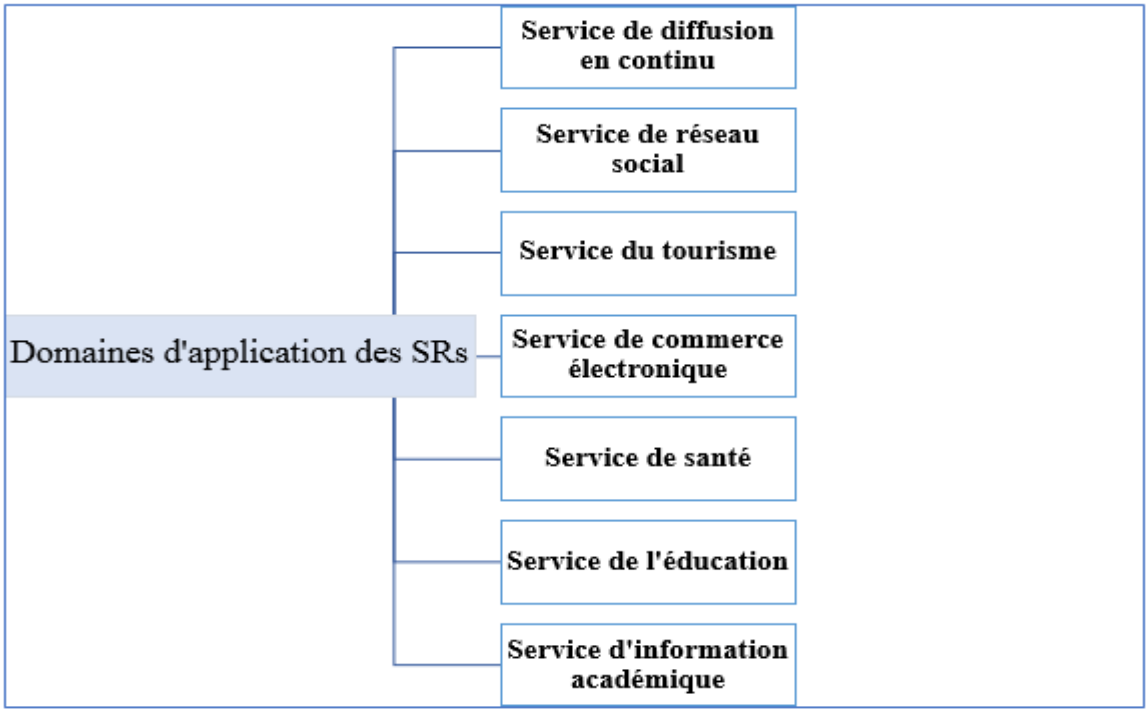


Figure 1.2 : *les Domaines d'application des SRs*

1.3.1 Service de diffusion en continu

Dans le passé, les contenus vidéo, tels que les films, étaient principalement consommés par les utilisateurs via la télévision ou les salles de cinéma. Récemment, une quantité importante de contenu vidéo est consommée via des plateformes de streaming tels que Netflix et YouTube. Le contenu audio passe également du téléchargement et de la consommation de fichiers sur l'appareil local d'un utilisateur à la consommation de contenu via des plateformes de streaming tels que Spotify. Les services de streaming liés au contenu multimédia ont été développés parallèlement au système de recommandation, car il est nécessaire de réduire les inquiétudes des utilisateurs quant au choix d'une grande quantité de contenu et de fournir un contenu adapté à chaque utilisateur. En général, dans le domaine du service de diffusion en continu, les données de préférence de l'utilisateur sont collectées en se concentrant sur les données d'historique d'utilisation du service de contenu multimédia de l'utilisateur, et après avoir mappé la préférence de l'utilisateur avec tout le contenu appartenant au service de diffusion en continu, les recommandations sont générées dans l'ordre du contenu le plus similaire à la préférence de l'utilisateur [5,6].

Jusqu'au début des années 2000, dans le domaine des services de streaming, le modèle de filtrage basé sur le contenu était largement utilisé dans les systèmes de recommandation. Cependant, à l'ère du Web 2.0, en raison de l'inefficacité de ne fournir qu'une partie du contenu aux utilisateurs dans un service de diffusion en continu qui présente de grandes quantités de données de contenu riches, «la recherche sur l'utilité de la méthode de filtrage collaboratif qui recommande des éléments en tenant compte de la similarité entre utilisateurs » a été principalement menée [7,8]. D'autre part, la plus grande caractéristique des services de streaming est que les informations sont classées par genre, artiste et distribution d'articles [9]. En raison de ces caractéristiques, les études utilisant des techniques de text mining ont été principalement utilisées dans l'étude de la recommandation de services de streaming. Dans l'étude d'Odić et al [10], Text Mining a été effectué sur les propriétés de la liste d'éléments de visualisation vidéo des utilisateurs afin de recommander le contenu souhaité par l'utilisateur. Dans Text Mining, l'ontologie a été utilisée non seulement pour trouver les éléments de film souhaités par les utilisateurs, mais également pour les éléments recommandés en tenant compte des informations de situation des utilisateurs.

Récemment, le nombre d'utilisateurs de services de streaming a augmenté de façon exponentielle. Par conséquent, pour assurer un service fluide, il est nécessaire de réduire la

surcharge de calcul de l'analyse des préférences de l'utilisateur. Afin de résoudre ce problème, des études ont été menées pour améliorer diverses techniques de recommandation et modèles de filtrage de recommandation pour les services de streaming. Barragáns-Martinez et al [11], ont utilisé l'algorithme de décomposition en valeurs singulières (SVD) pour réduire la surcharge de calcul lors du calcul des préférences de l'utilisateur afin d'atténuer les problèmes majeurs tels que les limitations d'évolutivité et la rareté de données. Grâce à cela, il a été proposé un modèle de recommandation hybride qui recommande des programmes de télévision avec une préférence d'utilisateur plus élevée.

1.3.2 Service de réseau social

Les services de réseaux sociaux en ligne (SRS) tels que Facebook, Instagram, Twitter et LinkedIn sont d'énormes échanges sociaux basés sur le numérique où les utilisateurs peuvent non seulement enregistrer leur vie quotidienne, leurs loisirs, leurs intérêts, etc., mais également fournir un champ d'interaction avec d'autres utilisateurs [12]. La forte augmentation de l'utilisation des SRS s'est également accompagnée d'une forte augmentation des données relatives aux utilisateurs.

Il est possible de collecter des informations sur le contenu que les utilisateurs enregistrent avec des messages via SRS. De plus, des données d'évaluation des utilisateurs peuvent être collectées ; en plus des données d'évaluation, celles-ci incluent divers types de données de rétroaction, tels que les goûts et les commentaires. Les données collectées ne sont pas seulement utilisées pour les recommandations au sein du SRS, mais sont également ouvertes à l'utilisation dans les systèmes de recommandation pour d'autres entreprises. En d'autres termes, diverses données collectées via SRS sont étroitement liées à l'avancement des systèmes de recommandation.

Étant donné que SRS est connecté à divers utilisateurs qui ne sont pas amis, les données d'autres utilisateurs similaires à l'utilisateur peuvent également être utilisées à des fins d'analyse pour produire des recommandations. Par conséquent, il est facile d'utiliser les modèles de filtrage collaboratif et de recommandation hybride [13].

1.3.3 Service du tourisme

À mesure que la demande de voyages a augmentée, les systèmes de recommandation ont commencé à être utilisés dans le domaine des services touristiques pour recommander des destinations touristiques, des recommandations d'itinéraire et des méthodes de transport. Comme le système de recommandation lié aux voyages utilise des données situationnelles, tels que les

données d'examen, les données de localisation, l'emplacement de l'utilisateur, l'heure et la météo, collectées via le service de réseau social. La recherche sur les systèmes de recommandation utilisant le SRS a augmenté dans le domaine des services touristiques.

Le SRS stocke les données d'enregistrement de l'utilisateur et l'emplacement de la publication téléchargée par l'utilisateur, et les services touristiques peuvent l'utiliser comme ensemble de données pour recommander des attractions et des itinéraires touristiques [14,15]. Le système de recommandation de voyage analyse ces données SRS et fournit des informations de voyage adaptées au goût de l'utilisateur, augmentant ainsi la satisfaction de ce dernier, la fidélité dans le tourisme et encourageant les utilisateurs à revenir dans les destinations touristiques après leur voyage. Kesorn et al. [14] ont proposé un cadre de service d'information touristique personnalisé (SITP) qui recommande des destinations touristiques personnalisées aux utilisateurs en fonction de leur analyse des données d'enregistrement sur Facebook. SITP utilise un modèle de recommandation hybride pour recommander des attractions touristiques aux utilisateurs, ainsi que les données d'enregistrement des amis Facebook des utilisateurs. Le modèle met à jour en permanence le profil de l'utilisateur en utilisant des données qui peuvent identifier les préférences que les utilisateurs génèrent lors de l'utilisation du service, tels que l'historique et les données d'évaluation.

1.3.4 Service de commerce électronique

Dans le passé, les items tels que les vêtements, la nourriture et les livres étaient principalement consommés par les utilisateurs via des magasins hors ligne. Cependant, ces dernières années, avec le développement des plateformes numériques tels que le web et les applications. La forme de consommation des items a changé à travers les plateformes de commerce électronique tels qu'Amazon, eBay et Alibaba. Le commerce électronique offre aux consommateurs de nombreux articles et diverses options dans l'environnement en ligne et offre aux vendeurs un moyen facile de vendre. En particulier, les consommateurs n'ont pas pu sortir en raison des mesures de confinement dues au COVID-19 et, par conséquent, ils n'ont pas pu utiliser les magasins hors ligne. Par conséquent, la consommation via les plateformes numériques a augmenté

de façon exponentielle. Les catégories d'articles vendus sur les plateformes numériques ont également commencé à se diversifier [16].

Le service de commerce électronique collecte des données relatives à divers utilisateurs pour l'expansion de l'entreprise et utilise activement ces données dans un système de recommandation. Le service prédit les préférences de l'utilisateur en analysant les informations utilisateur auxiliaires, tels que le sexe et le groupe d'âge.[17]. De plus, des recherches sont en cours pour recueillir des avis partagés sur des avis ou des articles qui reflètent les opinions subjectives des utilisateurs via la communauté virtuelle (CV) fournie par chaque commerce et utiliser le système de recommandation pour celui-ci [18]. Actuellement, les données de suivi sont générées en traçant les actions de la souris et du clavier lorsque l'utilisateur utilise le service, et ces données sont utilisées dans le système de recommandation. Les données de la souris et du clavier peuvent analyser les préférences des utilisateurs en suivant les interactions des utilisateurs avec les navigateurs et les applications pour déterminer les intentions d'achat [19].

Par conséquent, en utilisant les données collectées à partir du CV et du système de recommandation, il recommande l'item qui reflète la préférence de l'utilisateur. De plus, il est possible de recommander un article à un utilisateur en utilisant les informations de goût d'un groupe d'autres utilisateurs ayant des préférences similaires à l'utilisateur [20].

La caractéristique la plus importante du service de commerce électronique est que les consommateurs affichent généralement un modèle de dépenses qui complète les articles qu'ils ont précédemment préférés ou qu'ils ont déjà achetés. Par conséquent, trouver des articles similaires aux articles précédemment achetés par l'utilisateur est utile pour recommander des articles adaptés à l'utilisateur [21]. Raison pour laquelle, le modèle de recommandation de filtrage collaboratif et le modèle de recommandation hybride sont principalement utilisés dans le service [22, 23].

1.3.5 Service de santé

À mesure de l'accroissement de l'intérêt pour la santé, le nombre d'utilisateurs utilisant des appareils portables intelligents a commencé à augmenter à mesure que la technologie est devenue compatible avec les smartphones, et leur confort d'utilisation a également augmenté [24]. De tels dispositifs portables peuvent surveiller efficacement l'état biologique de l'utilisateur [24]. Smart Watch, un dispositif portable représentatif, mesure régulièrement les données corporelles de l'utilisateur [25], aide les utilisateurs ne possédant pas de connaissances médicales spécialisées à

prévenir les maladies et permet l'autodiagnostic. Ces appareils portables collectent une grande quantité de données biométriques de l'utilisateur pour aider à la recherche liée à la maladie ou au diagnostic approprié dans des situations corporelles spécifiques [26] et, en outre, il a été utile dans la recherche qui recommande un traitement [27]. Des études sur les systèmes de recommandation liés à la santé ont analysé la relation entre les schémas de symptômes des patients et les maladies pour fournir aux utilisateurs un aperçu des meilleures options de traitement [28,29]. Dans cette étude, le système de recommandation utilisé dans le domaine des services de santé a été subdivisé en domaines : système de recommandation de santé, e-santé, qui soutiennent le traitement professionnel en fonction de la finalité de l'application du système.

Dans le domaine des systèmes de recommandation de santé qui aident les utilisateurs à un traitement professionnel, l'objectif principal est de fournir des méthodes de traitement adaptées en fonction des symptômes des différents types de maladies et des stades de chaque maladie. À cette fin, le système de recommandation de santé analyse les informations du patient et les caractéristiques de la maladie, propose au patient, un diagnostic précis de la maladie et recommande un traitement approprié en fonction de la maladie diagnostiquée.

1.3.6 Service de l'éducation

À partir de la forme traditionnelle d'enseignement en classe ou en amphithéâtre, une nouvelle tendance éducative, appelée Smart Learning, s'est formée grâce à l'apprentissage en ligne, dans lequel l'apprentissage est effectué dans un environnement en ligne [30]. L'éducation intelligente a commencé à être progressivement utilisée dans l'éducation en raison de l'augmentation de la diffusion de divers appareils intelligents et du développement des réseaux sans fil. L'éducation intelligente peut accéder à de vastes ressources numériques et fournir de manière transparente un apprentissage personnalisé adapté aux besoins, aux objectifs, aux talents et aux intérêts des apprenants sans contraintes de temps et d'espace. De plus, la forme éducative s'est améliorée en reflétant la tendance d'apprentissage de l'ère numérique [31,32]. Par conséquent, le domaine des services éducatifs utilisant le système de recommandation fournit des ressources d'apprentissage en tenant compte du style d'apprentissage et du niveau de connaissances des

apprenants, offrant ainsi une expérience d'apprentissage efficace et efficiente. En d'autres termes, un contenu d'apprentissage personnalisé peut être fourni aux apprenants.

Dans l'étude de la recommandation de contenus d'apprentissage adaptés aux apprenants en mettant l'accent sur la similitude entre les apprenants et les objets d'apprentissage, le modèle de recommandation de filtrage basé sur le contenu a été principalement utilisé après avoir analysé les informations sur le profil de l'apprenant et les informations sur les objets d'apprentissage [33,34,35]. Une étude de Shu et al. [34] ont utilisé un modèle de recommandation de filtrage basé sur le contenu qui apprend les données textuelles des ressources d'apprentissage à l'aide de la technologie des réseaux neuronaux et fournit du matériel d'apprentissage à un niveau approprié aux apprenants en les combinant avec les préférences des apprenants.

Des études ont principalement été menées à l'aide du modèle de filtrage collaboratif, qui recommande un contenu d'apprentissage approprié en calculant les similitudes entre les activités d'apprentissage ou les apprenants [36,37,38,39]. De plus, en combinant le modèle basé sur les connaissances et la technique d'ontologie avec le modèle de filtrage collaboratif existant, il a été possible d'atténuer le problème de la rareté des données basée sur la similarité sémantique entre les apprenants et de générer des recommandations plus appropriées [36,39].

Dwivedi et al. [38] ont proposé un cadre de recommandation basé sur un modèle de filtrage collaboratif qui forme un groupe d'apprenants en reflétant les préférences individuelles des apprenants en e-learning et a fourni le contenu d'apprentissage le plus approprié aux apprenants en fonction de leur style d'apprentissage et de leur niveau de connaissances.

1.3.7 Service d'information académique

En raison de l'augmentation exponentielle de la quantité d'informations académiques, les chercheurs universitaires doivent consacrer beaucoup de temps et d'efforts à la recherche d'informations académiques dans le domaine lié à leur recherche. Dans le domaine des services d'information académique, des recherches sur les systèmes de recommandation ont été menées pour fournir des informations et des technologies qui peuvent être utiles aux universitaires lors de la conduite de recherches.

Un service représentatif auquel le système de recommandation dans le domaine de l'information académique est appliqué est la bibliothèque numérique, un système de collecte d'informations qui permet aux utilisateurs de rechercher et d'utiliser rapidement et facilement divers

documents numériques à travers le monde. En particulier, les bibliothèques numériques universitaires (BNU), un service qui soutient l'apprentissage, l'éducation et la recherche universitaires, utilisent également activement le système de recommandation [40,41]. Le système de recommandation n'a pas seulement été utilisé pour faciliter le processus d'accès à ces informations académiques, mais des recherches ont également été menées pour soutenir les données académiques liées à la recherche, la rédaction de la thèse et le processus de soumission [42,43]. En d'autres termes, l'objectif principal de la recherche sur les systèmes de recommandation dans le domaine de l'information académique est de recommander et de fournir des informations académiques adaptées à divers utilisateurs, y compris les communautés scientifiques, les instituts de recherche et les praticiens du développement, ainsi que de soutenir la recherche elle-même. D'autre part, les contenus les plus recommandés dans le domaine de l'information académique sont des documents de recherche académique composés principalement de textes. Par conséquent, un certain nombre d'études sur la recommandation d'informations académiques utilisant le modèle de recommandation de filtrage basé sur le contenu et le modèle de recommandation hybride utilisant des techniques de Text Mining ont été menées [44,45,46].

1.4 Classifications des SRs

En fonction de leur mode de fonctionnement général ou de la technique qu'ils utilisent [47], les systèmes de recommandation sont classés en différentes catégories, dont voici une liste de classifications :

1.4.1 Système de recommandation démographique :

Dans ce système de recommandation, les aspects démographiques des utilisateurs jouent un rôle dans la recommandation, tels que : l'âge, le sexe, l'occupation, la langue ou le pays. L'hypothèse est que la recommandation devrait varier en fonction du changement démographique. Ces solutions sont courantes dans le domaine du marketing. Cependant, il n'existe pas beaucoup de travail à leur sujet dans les systèmes de recommandation [48].

1.4.2 Recommandation basée sur les connaissances

Dans ces systèmes, les items recommandés sont basés sur la connaissance du domaine, la réponse aux questions, la façon dont certaines des fonctionnalités des items répondent aux besoins et aux préférences de l'utilisateur, ainsi que l'utilité de l'item pour l'utilisateur. Ces

recommandations de connaissances sont basées sur des cas [49], [50]. Une fonction de similarité détermine dans quelle mesure l'utilisateur a besoin de correspondre avec les recommandations.

1.4.3 Système basé sur les contraintes :

Il s'agit d'un autre type de système de recommandation, basé sur les connaissances. La principale différence entre les deux est la façon dont la solution est calculée. Dans la recommandation basée sur des cas, les items recommandés sont basés sur les métriques de similarité, tandis que les recommandations basées sur des contraintes exploitent principalement des bases de connaissances prédéfinies qui contiennent des règles explicites sur la manière de relier les exigences des clients aux fonctionnalités des items. Les systèmes basés sur la connaissance donnent généralement de meilleurs résultats au début de leur travail [51].

1.4.4 Recommandation basée sur la communauté

L'expression de bouche-à-oreille numérisé, devenue une source importante d'informations pour les consommateurs et les entreprises, permette aux consommateurs de partager facilement leurs opinions et leurs expériences sur la qualité de divers produits et vendeurs. Un système de recommandation basé sur la communauté est un système qui utilise le bouche-à-oreille numérisé pour créer une communauté d'individus qui partagent des opinions et des expériences personnelles, liées à leurs recommandations de produits et réputation des vendeurs. Ces systèmes présentent ou regroupent les opinions et les évaluations générées par les utilisateurs dans un format organisé. Les consommateurs consultent ces informations avant de prendre des décisions d'achat. Les travaux dans ce domaine ont été rendus possibles grâce à la généralisation des réseaux sociaux en ligne. Ce type de système de recommandation sera détaillé dans le chapitre suivant [52].

1.4.5 Système de recommandation contextuel :

Dans ce type de système, les résultats de la recommandation varient en fonction du contexte de l'utilisateur. Par exemple, dans le contexte temporel, les vêtements recommandés en été, varient totalement de ceux recommandés en hiver. Dans un contexte social, c'est qu'un film recommandé à l'utilisateur seul, peut varier de celui recommandé pour être vu en famille et le restaurant recommandé pour un jeudi soir entre amis, varierait d'un restaurant pour déjeuner en semaine avec des collègues [53].

1.5 Aperçu de l'architecture des SRs

Tous systèmes de recommandation contiennent un algorithme de recommandation. Cet algorithme est un filtrage collaboratif, un filtrage de contenu, un filtrage par cas, tout autre type de méthodes, ou un ensemble de celles-ci dans un modèle hybride. Les solutions collaboratives sont basées sur le contenu utilisant une méthode de similarité afin de trouver des items ou des utilisateurs similaires à recommander en fonction de leurs informations. Un système de recommandation a également besoin d'une source d'informations sur les utilisateurs, généralement dans la recommandation basée sur le filtrage collaboratif ; cette source est les évaluations des utilisateurs. Dans la recommandation basée sur le contenu, cette source peut être n'importe quelle action de l'utilisateur comme la navigation ou le fichier journal de l'utilisateur. Les réseaux sociaux sont devenus également une source d'informations sur les utilisateurs conduisant à des recommandations sociales. Pour évaluer un système de recommandation, la précision a un rôle principal, effectué généralement sur des ensembles de données disponibles (évaluation hors ligne). La satisfaction des utilisateurs bien que très importante est moins étudiée dans la littérature. Les composants des systèmes de recommandation sont présentés sur la figure1.3[54].

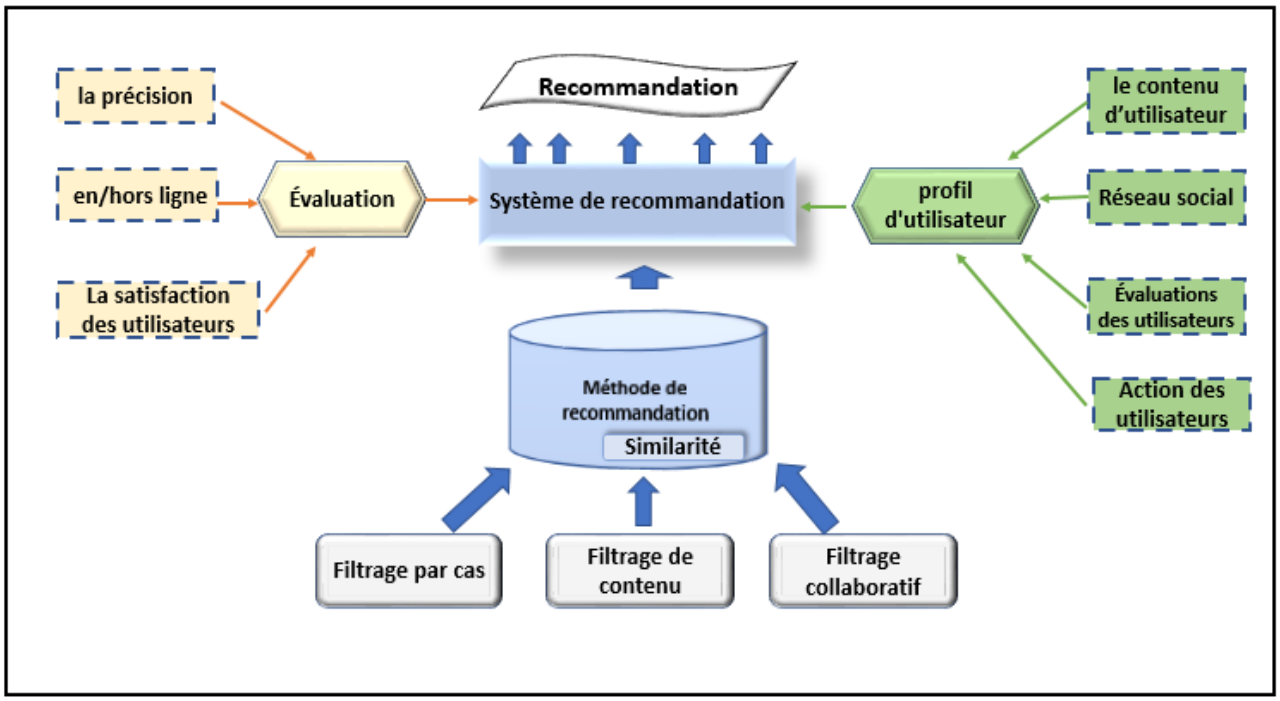


Figure 1.3 : *Composants des systèmes de recommandation*

1.6 Types d'approches de recommandation

Dans la littérature, les algorithmes de recommandation sont généralement classés en plusieurs types, selon la manière dont les recommandations sont formulées :

1.6.1 Approche basée sur le contenu

Le concept principal de cette approche est de recommander à un utilisateur des items similaires aux items qu'il a déjà consommés. Cela se réalise en analysant d'abord les descriptions des items que l'utilisateur a évalués, puis en créant un profil d'utilisateur [55]. Le profil doit représenter les intérêts de l'utilisateur pour les items. Ensuite, la recommandation est effectuée en faisant correspondre les attributs de l'utilisateur avec les attributs des items, ce qui donne un score d'intérêt pour cet item particulier pour un utilisateur. Une représentation précise de l'utilisateur peut entraîner un gain considérable.

Selon [56], l'ensemble du processus de recommandation basée sur le contenu peut être divisé en trois composants de base :

1. **Prétraitement et extraction de caractéristiques :** après avoir décidé quel contenu à considérer, l'étape qui suive, consiste à transformer toutes ces données en une représentation de documents textuels, c'est-à-dire à une représentation d'espace vectoriel basée sur des mots-clés. Généralement, nous le faisons avec un modèle de sac de mots, où chaque document ressemble à un sac contenant des mots sans aucun ordre. Cependant, avant de déterminer les sacs de mots, les données doivent être nettoyées en plusieurs étapes, telles que la suppression des mots vides, la radicalisation et la lemmatisation, et l'extraction de la phrase.
2. **Apprentissage basé sur le contenu des profils d'utilisateurs :** au sein de cette étape, les commentaires des utilisateurs (explicites ou implicites) sont utilisés en combinaison avec les informations de contenu des items pour construire les données d'apprentissage. Sur ces données d'apprentissage, un profil apprenant est construit et représente les préférences de chaque utilisateur.
3. **Filtrage et recommandation :** le modèle appris à l'étape précédente, prend toutes les entrées et génère la liste des recommandations pour chaque utilisateur.

L'approche basée sur le contenu, tente de résoudre les problèmes des algorithmes de filtrage collaboratif. Les principaux avantages de l'approche basée sur le contenu sont :

• Contrairement au filtrage collaboratif, cette méthode ne souffre pas du "problème de nouveaux items" du fait que les nouveaux items présentent des descriptions ainsi qu'une catégorisation.

• Le système basé sur le contenu est indépendant de l'utilisateur car il n'utilise pas les évaluations d'autres utilisateurs.

• Il est facile de fournir des explications, car le même contenu est utilisé pour expliquer les recommandations.

• Les représentations de contenu sont variées et ouvrent la possibilité d'utiliser différentes approches telles que : les techniques de traitement de texte, l'utilisation d'informations sémantiques, les inférences… etc.

Prenons l'exemple des systèmes de recommandation dans les films. Supposons que vous avez quatre films dans lesquels l'utilisateur commence par aimer seulement deux films au début. Pourtant, le 3ème film est similaire au 1er film en termes de genre, donc le système suggérera automatiquement le 3ème film. C'est quelque chose qui est automatiquement généré par un système de recommandation basé sur le contenu basé [57].

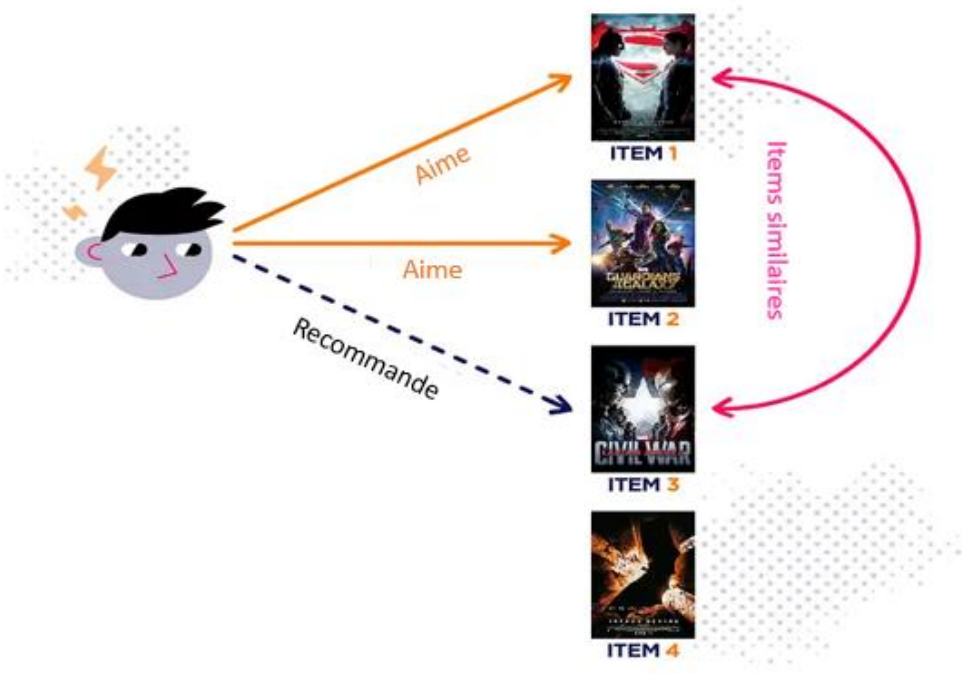


Figure1.4 *: Système de recommandation basée sur le contenu*

Les principaux inconvénients de l'approche basée sur le contenu sont :

- La dépendance aux caractéristiques associées au contenu : la méthode du filtrage basé sur le contenu n'utilise que les caractéristiques pour représenter le contenu des items (leur profil), ce qui rend cette technique non efficace et faible dans certains contextes avec certains types de données telles que les données multimédia, parce qu'il est très difficile d'extraire les caractéristiques à partir de ce type de données et par conséquent leur indexation est difficile. Un autre inconvénient de cette dépendance, c'est le cas où deux items distincts se représentent par les mêmes caractéristiques, ce qui rend leur différentiation impossible.
- L'incapacité du système à générer efficacement des recommandations qu'à partir d'un ensemble suffisant d'évaluations fournies par l'utilisateur (problème du nouvel utilisateur).
- L'incapacité de ces systèmes à intégrer des critères de pertinence des documents autres que les critères thématiques, malgré que plusieurs autres facteurs de pertinence puissent être utilisés comme l'adéquation entre le public visé par l'auteur et l'utilisateur, la fiabilité de la source d'information ou encore la qualité scientifique des faits présentés dans les articles.

1.6.2 Approche basée sur le filtrage collaboratif

Le filtrage collaboratif est un modèle de filtrage d'informations qui est apparu pour la première fois dans les années 1990 et est devenu un tremplin pour les recherches ultérieures sur les systèmes de recommandation [58,59]. Le filtrage collaboratif est un modèle qui construit la base de données des préférences d'un utilisateur en utilisant les données d'évaluation de l'utilisateur pour prédire les items qui correspondent au goût de l'utilisateur, puis l'utilise pour la recommandation [60]. Ce modèle peut être classé en filtrage collaboratif basé sur la mémoire et filtrage collaboratif basé sur un modèle [61]. Le filtrage collaboratif basé sur la mémoire peut être divisé en filtrage collaboratif basé sur l'utilisateur et filtrage collaboratif basé sur les items.

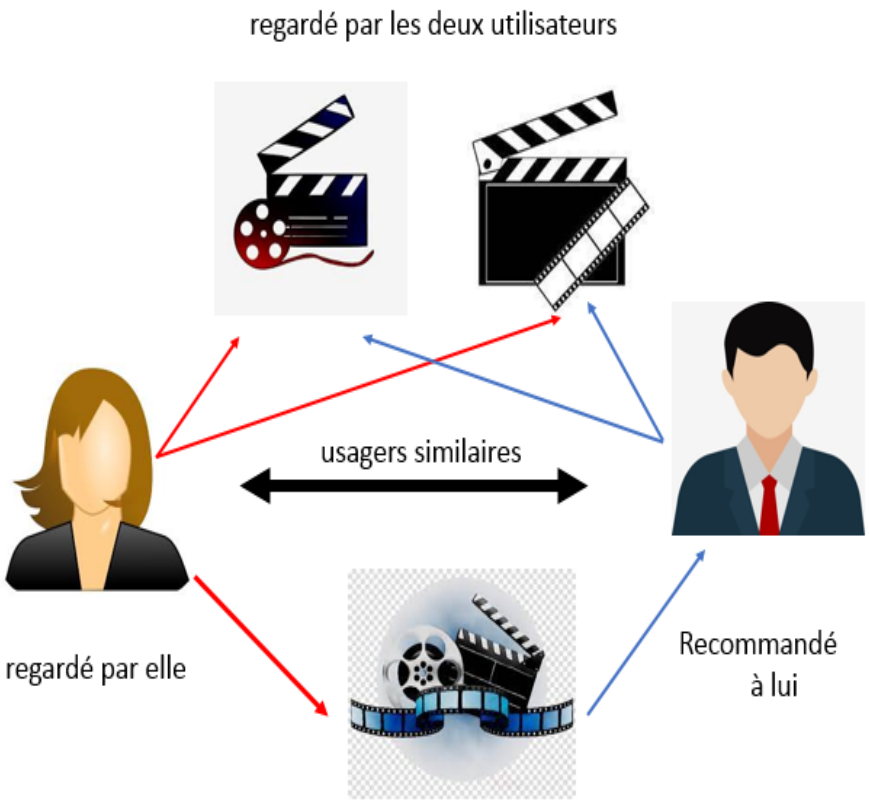


Figure1.5 : *Système de recommandation collaboratif [62]*

Tout algorithme de filtrage collaboratif a besoin d'une matrice de notes (appelée utilisateur-item) comme point de départ, puis vient deux tâches principales de recommandation :

La tâche de prédiction : pour les valeurs inconnues de (l'utilisateur, item), quelle note l'utilisateur attribuerait-il à cet item s'il lui était demandé de le faire ?

La tâche recommandation : pour un utilisateur, trouver la liste de n top items selon les préférences et les besoins de l'utilisateur.

Dans ce chapitre, ainsi que dans les suivants, nous ferons référence aux items par I et aux utilisateurs par *U*. Les items évalués ou achetés par l'utilisateur u sont I_u, alors que les utilisateurs qui ont évalué ou acheté un item i sont désignés par U_i. La matrice de notation est R_{UI} et la note de l'utilisateur u à l'item i est r_{ui}. Nous illustrons ci-dessous, dans le tableau I.1, un exemple d'une matrice d'évaluation pour trois utilisateurs et quatre films. Les valeurs marquées par « ? » indiquent l'absence d'avis par l'utilisateur.

	Item 1	Item 2	Item 3	Item 4
Utilisateu1	**4**	**2**	**4**	**?**
Utilisateu2	**?**	**5**	**5**	**4**
Utilisateu3	**2**	**5**	**5**	**?**

Tableau I.1 : Exemple de matrice de notation sur une échelle de 1 à 5.

r_u est un vecteur de toutes les notes fournies par l'utilisateur u et r_i est le vecteur de toutes les notes fournies à l'item i.

Les algorithmes de filtrage collaboratif sont regroupés en deux classes générales : à base de mémoire et à base de modèle :

1.6.2.1 Algorithmes basés sur la mémoire

Ces systèmes utilisent des techniques statistiques pour trouver un ensemble d'utilisateurs, appelés voisins, qui ont des modèles similaires de comportement d'évaluation avec l'utilisateur cible (c'est-à-dire qu'ils évaluent différents items de la même manière ou qu'ils ont tendance à acheter des ensembles d'items similaires), ou un ensemble d'items voisins, qui sont similaires aux items déjà notés de l'utilisateur. Ces voisinages peuvent être définis de deux manières [63] :

1. Filtrage collaboratif basé sur l'utilisateur

L'idée de base de cette méthode est de prédire les notes en fonction des préférences des voisins de l'utilisateur actif, qui ont des favoris similaires avec un utilisateur actif. Pour recommander un livre à un utilisateur au sein d'une application de recommandation de livres, le système de recommandation collaboratif basé sur l'utilisateur essaie de trouver d'autres utilisateurs qui présentent des goûts similaires (noter le même livre de manière similaire). Ensuite, seuls les livres les plus appréciés par les *k* utilisateurs les plus similaires serons recommandés. L'application la plus populaire de cette technique est l'algorithme *k* plus proche voisin basé sur l'utilisateur. Son principe de fonctionnement se résume en deux étapes [64] :

1. Déterminer les voisins de l'utilisateur actif en calculant sa similarité avec les autres utilisateurs de la base.
2. Calculer la prédiction de la note de l'utilisateur actif u_a pour un item $i \in I$ candidat à la recommandation, en analysant les notes de ses voisins sur ce même item.

La note prédite *pred* (u_a, i) de l'item *i* par l'utilisateur u_a dépend de la similarité entre cet utilisateur et ses voisins les plus proches notée par *sim* (u_a, u), l'évaluation de l'utilisateur *u* sur

l'item i notée par r_{ui}, et d'un facteur de normalisation. *pred (u, i)* est décrit par la formule suivante :

$$\text{pred}(u_a, i) = \overline{\mathcal{V}u_a} + \frac{\sum u \in voisins(\mathcal{U}_a) \cap \mathcal{U}_i \, \text{sim}(\mathcal{U}_a, \mathcal{U})(\mathcal{V}u_i - \overline{\mathcal{V}u})}{\sum u \in voisins(u_a) \cap \mathcal{U}_i \, |sim(u_a, u)|} \qquad (1.2)$$

2. Filtrage collaboratif basé sur les items

D'autre part, l'approche basée sur les items fournisse la prédiction de la note de l'utilisateur *u* pour un item candidat *i* est calculée à partir de ses notes pour les items voisins (similaires) de *i*. Son principe de fonctionnement est le suivant :

1. Pour l'item *i* candidat à la recommandation, on détermine les voisins les plus proches (les items similaires) en calculant sa similarité avec les autres items disponibles.
2. On calcule ensuite la prédiction de la note de l'utilisateur actif u_a pour l'item *i* à partir des notes que u_a a attribué aux voisins de l'item *i*.

La prédiction de la note de l'utilisateur actif u_a pour un item candidat à la recommandation i revient à calculer une moyenne pondérée de ses notes sur l'ensemble des items similaires à *i*. Chaque note r_{uj} est pondéré par la similarité de l'item *j* avec l'item *i*. Afin d'avoir une prédiction dans le même intervalle de valeurs que les votes, la prédiction est divisée par la somme des similarités. L'équation (1.3) donne la formule de calcul de la prédiction [65].

$$\text{pred}(u, i) = \frac{\sum_{i \in I_u} sim(i,j) \times r_{u,j}}{\sum_{i \in I_u} |sim(i,j)|} \qquad (1.3)$$

- **Fonctions de similarité**

Pour trouver des utilisateurs partageant les mêmes centres d'intérêts ou préférences, des fonctions de similarité *Sim (u, v)* sont calculées entre les vecteurs de notation des utilisateurs *u* et *v*, c'est-à-dire entre les lignes de la matrice de notation *R*. Diverses fonctions de similarité sont utilisées en pratique. Le choix d'une mesure de similarité appropriée est très important pour un système de recommandation, car différentes mesures de similarité fournissent des résultats

différents dans divers contextes d'information. Nous décrivons ci-dessous, certaines des métriques de mesure de similarité populaires qui sont utilisées dans le filtrage collaboratif [66].

- **Coefficient de corrélation de Pearson**

Ce coefficient calcule la corrélation statistique de Pearson entre deux vecteurs d'évaluation pour déterminer la similarité. S'il s'agit de calculer la similarité entre deux utilisateurs, la corrélation entre eux est mesurée à l'aide des deux lignes, appartenant aux deux utilisateurs, de la matrice d'évaluations. Les colonnes des items non évaluées par les deux utilisateurs sont ignorées. Seuls les items Co-évalués sont utilisés dans ce calcul. Ce coefficient se situe entre -1 et 1. Une similarité proche de -1, signifie une corrélation négative et inversement, une similarité proche de +1 signifie une corrélation positive. Il n'existe pas de corrélation entre les deux utilisateurs si la similarité est autour de 0. La similarité *sim (u, v)* entre les utilisateurs u et v est donnée par l'équation Eq (1.4) [67] .

$r(u, .)$ est la moyenne des évaluations de l'utilisateur *u*. *I* est l'ensemble des item Co-évalués par *u* et *v*.

$$\text{Sim}\,(u,v) = \frac{\sum \text{i}\in \text{I}_{\text{uv}}\ (\text{r}_{\text{u,i}} - \overline{\text{r}_{\text{u}}})(\text{r}_{\text{v,i}}\ - \overline{\text{r}_{\text{v}}})}{\sqrt{\sum \text{i}\in \text{I}_{\text{uv}}\ (\text{r}_{\text{u,i}} - \overline{\text{r}_{\text{u}}})^2}\ \sqrt{\sum \text{i}\in \text{I}_{\text{uv}} (\text{r}_{\text{v,i}} - \overline{\text{r}_{\text{v}}})^2}} \qquad (1.4)$$

- **Similarité basée sur le cosinus**

Dans la matrice d'évaluation, les lignes associées aux utilisateurs sont considérées comme des vecteurs d'évaluation. Ce type de mesure de similarité est calculé en utilisant l'angle cosinus entre deux vecteurs d'évaluation. Cet angle est mesuré dans un espace à *N* dimensions où *N* est le nombre d'items Co-évalués entre les deux utilisateurs. Cette similarité se situe entre 0 et 1, où le 0 signifie aucune similarité et 1 une forte similarité [68]. Cette similarité entre les utilisateurs est décrite par la formule :

$$\cos(u,v) = \frac{\sum k\epsilon I_{\text{u}}\cap I_{\text{v}}\ \ \text{r}_{\text{u}}\text{k}\,.\,\text{r}_{\text{v}}\text{k}}{\sqrt{\sum k\epsilon I_{\text{u}}\cap I_{\text{v}}\ r^2{}_{\text{u}}\text{k}}\ .\sqrt{\sum k\epsilon I_{\text{u}}\cap I_{\text{v}}\ r^2{}_{\text{v}}\text{k}}} \qquad (1.5)$$

- **Coefficient de Jaccard**

L'indice de Jaccard (ou coefficient de Jaccard, appelé « coefficient de communauté » dans la publication d'origine) est un indice statistique utilisé pour comparer la similarité et la diversité de l'ensemble des échantillons.

Coefficient de Jaccard : c'est le rapport entre le cardinal (la taille) de l'intersection des ensembles Considérés et le cardinal de l'union des ensembles. Il permet d'évaluer la similarité entre les ensembles. Soit deux ensembles *A* et *B*, l'indice est la suivante [69] :

$$\text{Jaccard}(A, B) = \frac{|A \cap B|}{|A \cup B|} \qquad (1.6)$$

I.6.2.2 Algorithmes basés sur des modèles :

Les systèmes de recommandation (SR) sont l'une des applications les plus visibles et les plus réussies de la technologie de l'intelligence artificielle dans la pratique, et les recommandations personnalisées, telles qu'elles sont fournies sur de nombreux sites de commerce électronique ou de médias modernes, peuvent avoir un impact substantiel sur différentes parties prenantes. Sur les sites de e-commerce par exemple, les choix des consommateurs peuvent être largement influencés par les recommandations, et ces choix sont souvent directement liés à la rentabilité de la plate-forme. Sur les sites Web d'actualités ou les médias sociaux, en revanche, les recommandations personnalisées peuvent déterminer dans une large mesure les informations que nous voyons, ce qui peut à son tour façonner non seulement nos propres croyances, décisions et actions, mais également les croyances d'une communauté d'utilisateurs ou toute une société. Des exemples typiques de méthodes de filtrage basées sur des modèles incluent les arbres de décision, les règles d'association, les réseaux bayésiens, les modèles sémantiques latents et le modèle de clustering. Généralement, les algorithmes basés sur la mémoire ont tendance à être faciles à mettre en œuvre et à produire une qualité de prédiction raisonnablement élevée. De plus, les résultats de la recommandation sont souvent faciles à expliquer. Cependant, les algorithmes basés sur la mémoire souffrent d'un grave problème d'évolutivité. Cela s'applique en particulier aux sites de commerce électronique modernes, car la croissance constante du nombre d'utilisateurs et d'articles ralentit leurs performances en ligne. Les algorithmes basés sur un modèle ont tendance à être plus rapides que les algorithmes basés sur la mémoire, car le temps nécessaire pour interroger le modèle est

généralement beaucoup plus court que le temps nécessaire pour interroger l'ensemble de données. Mais ses inconvénients montrent que de nombreux modèles théoriques sont complexes et ne correspondent parfois pas bien aux données réelles. De plus, il est plus difficile et sa prend beaucoup de temps de créer ou de mettre à jour des modèles pour des algorithmes basés sur des modèles, ce qui les rend inflexibles [70].

1.6.3 Système de recommandation hybride

Toutes les approches mentionnées ci-dessus présentent des forces et des faiblesses différentes, et chacune des méthodes peut être plus efficace dans différents cas. Par exemple, les systèmes de filtrage collaboratif dépendent des évaluations des utilisateurs. Les algorithmes basés sur le contenu reposent sur des descriptions d'items textuels et les méthodes basées sur les connaissances reposent sur des interactions avec les utilisateurs dans le contexte de bases de connaissances. Différentes stratégies d'hybridation peuvent être appliquées. Ces stratégies sont plus faciles à expliquer sur le système de recommandation hybride le plus fréquemment utilisé [71,72].

- une combinaison de méthodes collaboratives et basées sur le contenu :

- **Conception d'ensemble :** c’est une combinaison de méthodes de recommandation distinctes : dans ce cas, les systèmes collaboratifs et basés sur le contenu sont mis en œuvre séparément. Il existe deux stratégies de combinaison différentes : pondérée et modèle de commutation. Dans les hybridations pondérées, les sorties (notes) obtenues à partir de systèmes de recommandation individuels sont combinées à l'aide d'un ensemble de pondérations, par exemple en utilisant un schéma de vote ou une combinaison linéaire de notes. Les mécanismes de commutation sont souvent utilisés pour gérer le problème du démarrage à froid, dans lequel l'une des recommandations individuelles est choisie à un moment donné parce qu'elle est "meilleure" que les autres sur la base d'une métrique de qualité de recommandation. Par exemple, le système Daily Learner sélectionne le système de recommandation qui peut recommander avec un niveau de confiance plus élevé.
- **Conception monolithique :** Cette méthode d'hybridation est divisée pour présenter des stratégies de combinaison et de niveau méta. Dans les hybrides de combinaison de fonctionnalités, l'idée est de combiner les données d'entrée provenant de diverses sources (par exemple, le contenu et la collaboration) dans une représentation unifiée avant d'appliquer un algorithme prédictif. L'approche la plus populaire dans cette catégorie, consiste à ajouter une fonctionnalité collaborative aux modèles basés sur le contenu. Dans

un hybride de niveau méta, un système de recommandation est utilisé comme entrée d'un autre système. La méthode typique est appelée « collaboration via le contenu », où un système collaboratif est modifié pour utiliser les caractéristiques du contenu afin de déterminer les groupes de pairs.

- **Hybridations mixtes :** cette approche est la plus appropriée dans les domaines d'items complexes, et elle est souvent utilisée en combinaison avec des systèmes de recommandation basés sur les connaissances.

1.7 Problèmes et Difficultés d'un système recommandations

Construire un système de recommandations est une tâche qui s'avère être particulièrement complexe pour les entreprises qui n'ont ni les mêmes besoins, ni les mêmes objectifs. Malgré ces différences, certains problèmes sont récurrents à la majorité des systèmes de recommandations et c'est ce que nous allons voir dans ce chapitre.

- **Nouvel utilisateur :** Un nouvel utilisateur qui n'a pas encore accumulé suffisamment d'évaluations ne peut pas avoir de recommandations pertinentes.
- **Nouvel item :** Un item doit avoir suffisamment d'évaluations pour qu'il soit pris en considération dans le processus de recommandation.

1.7.1 Démarrage à froid : Les systèmes de filtrage collaboratif dépendent des évaluations des items par les utilisateurs. Ainsi, le démarrage à froid est un problème pour les nouveaux utilisateurs qui commencent à jouer avec le système, parce que le système ne dispose pas d'assez d'informations à leur sujet. Si le profil d'utilisateur est vide, il doit consacrer une somme d'efforts à l'aide du système, avant d'obtenir une récompense (les recommandations utiles). Ainsi, les nouveaux utilisateurs ne recevront pas de recommandations précises avant d'avoir évalué un certain nombre d'items.[72]

La cause du problème de démarrage à froid est liée au manque d'informations sur l'entité (film, livre, article, ...etc.) concernée qui est un problème très important et qui devrait être abordé. De nombreuses solutions ont été proposées pour résoudre ce problème.

1.7.2 Sérendipité : Vu que les systèmes de recommandation basés sur le contenu ne recommandent que les items correspondants au profil de l'utilisateur, ce dernier ne recevra que des recommandations similaires à celles qu'il a déjà rencontrées. Il n'aura aucune chance de recevoir des recommandations inattendues. Cela peut amener l'utilisateur à se lasser des recommandations

1.7.3 Sécurité ou crédibilité : Les systèmes de recommandation ne peuvent pas empêcher les actes de tromperie. Il est difficile de contrôler l'identité des utilisateurs et de pénaliser le comportement malveillant. Par conséquent, Il est indispensable d'avoir des moyens permettant à chaque utilisateur de décider en quels utilisateurs et en quels contenus avoir confiance.

1.7.4 Masse critique : Afin de former de meilleures communautés, le système exige un nombre suffisant d'évaluations en commun entre les utilisateurs pour les comparer entre eux. Malgré la taille énorme de l'ensemble des documents dans les systèmes, le nombre d'évaluations en commun entre les utilisateurs risque d'être faible.

1.7.5 Problème du mouton gris : Les utilisateurs d'un système de recommandation peuvent avoir des goûts particuliers et des préférences très inhabituelles par rapport aux autres. Ces utilisateurs sont à la frontière entre deux ou plusieurs clusters d'utilisateurs. Il leur est donc difficile de trouver des utilisateurs similaires et des recommandations pertinentes.

1.8 Conclusion

Dans ce chapitre, nous avons présenté un état de l'art sur les systèmes de recommandation qui sont devenus omniprésents ces dernières années dans de nombreux domaines. Nous nous sommes concentrés sur la technique de filtrage collaboratif qui est la technique de filtrage d'informations la plus populaire utilisée dans les systèmes de recommandation. Comparé à ses pairs (basé sur le contexte et basé sur les connaissances, etc.), le FC est choisi en raison de sa simplicité et de sa facilité de flexibilité dans le modèle tout en capturant les intérêts des utilisateurs au fil du temps. Dans le chapitre suivant, nous présentons les principales méthodes de détection de communautés, ainsi que quelques travaux sur les systèmes de recommandation à base de communautés.

Chapitre II

Les principales méthodes de détection de communautés

2.1 Introduction

La découverte de groupes cohésifs, de cliques et de communautés au sein d'un réseau est l'un des sujets les plus étudiés dans l'analyse des réseaux sociaux. Il a attiré de nombreux chercheurs en sociologie, biologie, informatique, physique, criminologie, etc. La détection de communautés vise à trouver des clusters sous forme de sous-graphes au sein d'un réseau donné. Une communauté est alors un cluster où de nombreuses arêtes relient des nœuds d'un même groupe et peu d'arêtes relient des nœuds de clusters différents.

Une approche générale de la détection de communautés consiste à considérer le réseau comme une vue statique dans laquelle tous les nœuds et liens du réseau sont maintenus inchangés tout au long de l'étude. Des études récentes se concentrent également sur l'évolution de la communauté puisque la plupart des réseaux sociaux ont tendance à évoluer avec le temps par l'ajout et la suppression de nœuds et de liens. En conséquence, les groupes à l'intérieur d'un réseau peuvent s'étendre ou se réduire et leurs membres peuvent passer d'un groupe à un autre au fil du temps. La plupart des études sur l'évolution des communautés utilisent des propriétés topologiques pour identifier les parties mises à jour du réseau et caractériser le type de changements tels que le rétrécissement, la croissance, la division et la fusion du réseau.

Ce chapitre fournit le contexte de la recherche liée à notre travail. Ce dernier contient un résumé des définitions existantes relatives aux réseaux sociaux, graphe et communauté. Ce chapitre traite également des algorithmes de détection de communautés existantes qui seront utilisés dans nos expériences au sein de ce projet. En outre, il donne un aperçu sur quelques travaux connexes qui se sont intéressés à l'application des méthodes de détection de communautés dans le cadre de la recommandation.

2.2 Réseaux Sociaux

Divers systèmes biologiques, sociaux et technologiques du monde réel peuvent être représentés par des réseaux. De tels réseaux sont composés de deux sous-structures principales : les nœuds et les liens. À savoir, les nœuds décrivent des entités dépendantes du domaine et relient les relations ou les interactions entre les entités. Un exemple d'interactions sociales peut être exprimé par des nœuds représentant des personnes et des liens représentant des relations sociales telles que des amitiés.

L'idée des réseaux sociaux a été introduite en 1954 par le sociologue J.A. Barnes [73]. Depuis, de nouvelles formes de réseaux sociaux ont émergé grâce au développement de la

technologie informatique à savoir les Réseaux Sociaux en Ligne (RSL) tels que Facebook, LinkedIn, YouTube et Google+. Ils fournissent des plateformes en ligne qui permettent la communication électronique entre les acteurs. Des millions d'utilisateurs sont réunis au sein de ces réseaux formant des communautés liées à différentes relations. Ils sont accessibles via des ordinateurs, des appareils mobiles, des tablettes, etc., et fournissent divers services tels que la gestion de profil, le chat, le partage de contenu, les commentaires, les forums et la messagerie instantanée. L'essor d'Internet au cours de la dernière décennie a stimulé le développement des RSL, ce qui en fait probablement l'acteur central du Web 2.0.

Deux concepts essentiels sont nécessaires pour construire un réseau social. Premièrement, il faut déterminer les entités sociales du réseau. Ces derniers peuvent être de divers types tels que des utilisateurs, des organisations, des entreprises, des substrats, des auteurs ou des criminels. Ils peuvent être nommés et avoir plusieurs propriétés qui leur sont jointes. Ensuite, il faut définir les liens sociaux selon certaines formes d'interdépendance tels que l'amitié, l'échange financier, la proximité physique, la connaissance, les relations de croyances, l'interaction chimique ou la copaternité.

Divers réseaux sociaux existent [74, 75, 76]. Les exemples les plus populaires seraient les réseaux sociaux d'amitiés entre personnes [77], les réseaux criminels [78, 79], ainsi que les réseaux de collaboration de co-auteurs [75]. D'autre part, de nombreuses informations peuvent être extraites des réseaux sociaux en ligne qui permettent la construction de réseaux tels que des amitiés (c'est-à-dire de Facebook) et des réseaux de collaboration de co-auteurs (c'est-à-dire d'ArXiv et Dblp) [75].

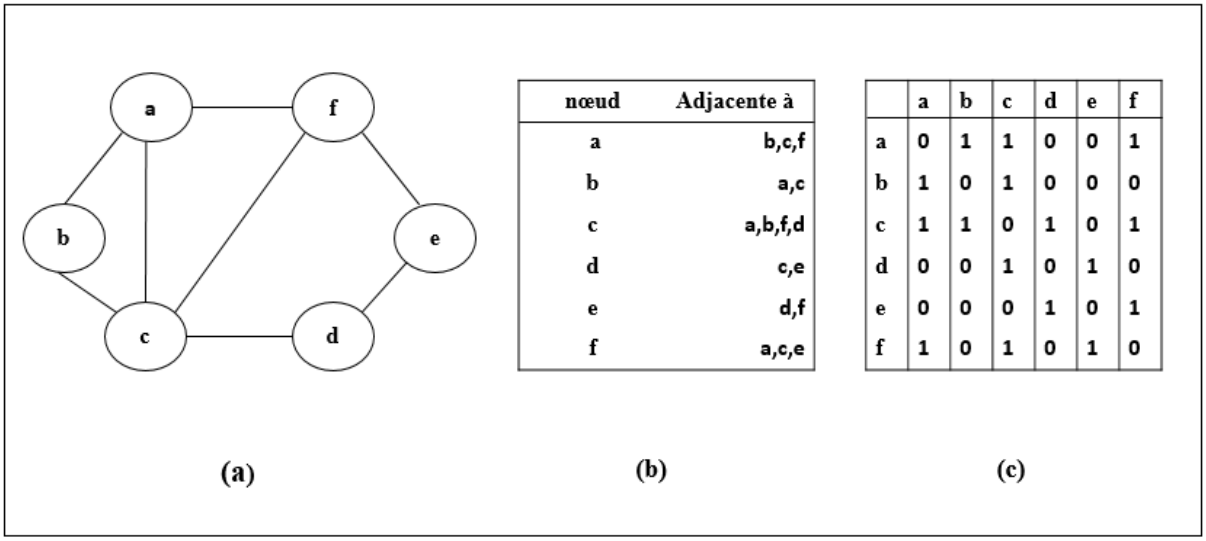
2.2.1 Représentation des réseaux sociaux

D'un point de vue historique, la première représentation des réseaux sociaux fut les sociogrammes développés par Jacob Moreno [80] pour apprendre les relations interpersonnelles. Il était dessiné par un ensemble de points représentant des personnes et des lignes codant leurs relations. Cette représentation a été formalisée mathématiquement dans les années 1950 et est devenue la représentation de base des sciences sociales et comportementales modernes, ainsi que de l'analyse des réseaux sociaux. Il est basé sur un concept classique de la théorie des graphes qui est la représentation du modèle de graphe.

A. Graphe :

Un graphe, dans sa compréhension intuitive, est un groupe de nœuds connectés les uns aux autres pour former soit un composant connecté, soit plusieurs composants disjoints.
De ce point de vue, un réseau de n acteurs et de m connexions est mathématiquement noté $G\,(_{V,\,E})$, où V est l'ensemble des nœuds $\{v_1, v_2, v_3, \ldots, v_n\}$, et E est l'ensemble des liens $\{e_1, e_2, e_3, \ldots, e_m\}$. Les nœuds décrivent les entités sociales et les liens décrivent les liens relationnels.
Schématiquement, un nœud (c'est-à-dire un acteur, un agent, un objet) peut être représenté en utilisant différentes couleurs, nuances, symboles et tailles pour spécifier différentes propriétés ou types. Il peut être étiqueté pour exprimer son nom, son type, son numéro, sa référence, etc. Pour indiquer différents types de relations, les acteurs sont liés par : des lignes pointillées, des lignes en gras, des multilignes, des couleurs différentes, des lignes non orientées (c'est-à-dire des réseaux de collaboration, réseaux d'amitié), les lignes dirigées (c. Ils peuvent être pondérés pour coder leur force, leur probabilité, leur fréquence, etc [81].

Les données graphiques peuvent être stockées à l'aide de matrices ou de listes. La représentation matricielle est plus pratique pour représenter des réseaux denses. A l'inverse, les données des réseaux sociaux sont rares surtout les réseaux à grande échelle, ainsi la représentation en liste est plus favorisée. Dans la figure 2.1 (a), un réseau social est représenté à l'aide d'un graphe à six nœuds reliés par huit arêtes. La liste d'adjacence (Figure 2.1 (b)) du premier donne une simple liste de nœuds et ceux qui leur sont liés/adjacents appelés « voisins ». En revanche, la matrice d'adjacence (Figure 2.1 (c)) est de taille 6 × 6 et contient des valeurs binaires indiquant la présence ou l'absence d'arêtes entre les nœuds. Dans le cas des graphes orientés, nous obtenons des matrices d'incidence et des listes d'incidence qui sont analogues aux premières sauf que les informations stockées spécifient l'incidence des nœuds et des arêtes.



nœud	Adjacente à
a	b,c,f
b	a,c
c	a,b,f,d
d	c,e
e	d,f
f	a,c,e

	a	b	c	d	e	f
a	0	1	1	0	0	1
b	1	0	1	0	0	0
c	1	1	0	1	0	1
d	0	0	1	0	1	0
e	0	0	0	1	0	1
f	1	0	1	0	1	0

Figure 2.1 : *(a) Réseau social représenté sous forme de graphe (les nœuds sont étiquetés par des lettres) (b) Liste d'adjacence (c) Matrice d'adjacence*

2.3 Détection des communautés dans les réseaux sociaux

Un réseau social peut être représenté par un graphe constitué d'un ensemble de nœuds et d'arêtes reliant ces nœuds. Les nœuds représentent les individus/entités, et les arêtes correspondent aux interactions entre eux. La tendance des personnes ayant des goûts, des choix et des préférences similaires à s'associer dans un réseau social conduit à la formation de clusters ou de communautés virtuelles. La détection de ces communautés peut être bénéfique pour de nombreuses applications telles que la recherche d'un domaine de recherche commun dans les réseaux de collaboration, la recherche d'un ensemble d'utilisateurs partageant les mêmes idées pour le marketing et les recommandations, et la recherche de réseaux d'interaction protéique dans les réseaux biologiques. Un grand nombre d'algorithmes de détection de communauté ont été proposés et appliqués à plusieurs domaines de la littérature [82].

2.3.1 Communauté

Une communauté peut être définie comme étant un groupe d'entités plus proches les unes des autres par rapport aux autres entités du jeu de données (voir figure 2.2). La communauté est formée d'individus tels que ceux au sein d'un groupe interagissent les uns avec les autres plus fréquemment qu'avec ceux en dehors du groupe. La proximité entre entités d'un groupe peut être mesurée par des mesures de similarité ou de distance entre entités. McPherson et al [83] ont déclaré que "la similarité engendre la connexion". Ils ont discuté de divers facteurs sociaux qui conduisent à des comportements similaires ou à l'homophilie dans les réseaux. Les communautés dans les

réseaux sociaux sont analogues aux clusters dans les réseaux. Un individu représenté par un nœud dans les graphes peut ne pas faire partie d'une simple communauté ou d'un groupe, il peut être un élément de nombreux groupes étroitement associés ou différents existant dans le réseau. Par exemple, une personne peut appartenir simultanément à un collège, une école, des amis et des groupes familiaux. Toutes ces communautés qui ont des nœuds communs sont appelées communautés qui se chevauchent. L'identification et l'analyse de la structure de la communauté ont été effectuées par de nombreux chercheurs appliquant les méthodologies de nombreuses formes de sciences. La qualité du clustering dans les réseaux est normalement jugée par le coefficient de clustering qui est une mesure de la tendance à laquelle les sommets d'un réseau ont tendance à se regrouper. Le coefficient de clustering global [84] et le coefficient de clustering local [85] sont deux types de coefficients de clustering discutés dans la littérature.

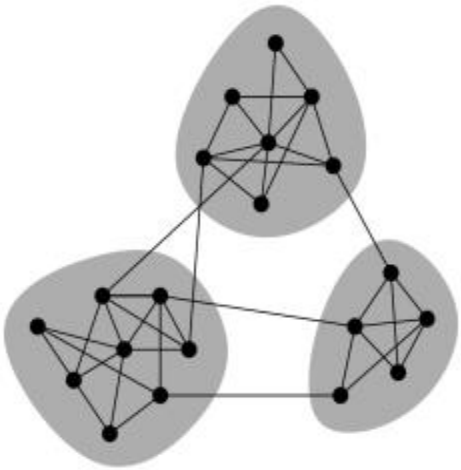


Figure2.2 : *Graphe structuré en trois communautés*.

La quantité de base à considérer est k_i, le degré d'un nœud générique i, qui en termes de matrice d'adjacence $A_{i,j}$ du réseau G est $k_i = \sum_j A_{i,j}$. (La matrice d'adjacence spécifie entièrement la topologie du réseau. Dans le cas le plus simple d'un réseau non pondéré et non orienté, elle est égale à 1 si i et j sont directement connectés ; elle est égale à zéro sinon.) Si nous considérons un

sous-graphe $V \subset G$, auquel appartient le nœud *i*, on peut scinder le degré total en deux contributions : $k_i(V) = k_i^{int}(V) + k_i^{out}(V)$

$k_i^{int}(V) = \sum_{j \in V} A_{i,j}$ Est le nombre d'arêtes reliant le nœud *i* aux autres nœuds appartenant à *V*.

$k_i^{out}(V) = \sum_{j \notin V} A_{i,j}$ est le nombre de connexions vers les nœuds du reste du réseau.

- **Définition de communautés au sens fort :** Le sous-graphe *V* est une communauté au sens *fort* si : $k_i^{int}(V) > k_i^{out}(V), \forall\, i \in V$

Dans une communauté forte, chaque nœud a plus de connexions au sein de la communauté qu'avec le reste du graphe.

- **Définition de la communauté dans un sens faible** : Le sous-graphe *V* est une communauté au sens faible si : $\sum_{i \in V} k_i^{int}(V) > \sum_{i \in V} k_i^{out}(V)$

Dans une communauté faible, la somme de tous les degrés dans V est supérieure à la somme de tous les degrés vers le reste du réseau.

2.4 Méthodes de regroupement d'éléments similaires

Les communautés sont les parties du graphique qui ont des connexions plus denses à l'intérieur et peu de connexions avec le reste du graphique [86]. Le but de l'apprentissage non supervisé est de regrouper des objets similaires sans aucune connaissance préalable à leur sujet. Dans le cas des réseaux, le problème de clustering fait référence au regroupement de nœuds en fonction de leur similarité calculée sur la base de caractéristiques topologiques et/ou d'autres caractéristiques du graphe. Le partitionnement et le clustering du réseau sont deux méthodes couramment utilisées dans la littérature pour trouver les groupes dans le graphe du réseau social.

2.4.1 Partitionnement de graphe

Le partitionnement de graphe est le processus de partitionnement d'un graphe en un nombre prédéfini de composants plus petits avec des propriétés spécifiques. Une propriété commune à minimiser est appelée taille de coupe. Une coupe est une partition de l'ensemble des sommets d'un graphe en deux sous-ensembles disjoints et la taille de la coupe est le nombre d'arêtes entre les composants. Une multi coupe est un ensemble d'arêtes dont la suppression divise le graphe en deux ou plusieurs composantes. Il est nécessaire de préciser le nombre de composantes que l'on souhaite obtenir en cas de partitionnement du graphe. La taille des composants doit également être spécifiée, sinon une solution probable mais non significative serait de mettre le sommet de degré minimum

dans un composant et le reste des sommets dans un autre. Etant donné que le nombre de communautés n'est généralement pas connu à l'avance, les méthodes de partitionnement de graphes ne sont pas adaptées pour détecter les communautés dans de tels cas [87].

2.4.2 Clustering

Le clustering est le processus de regroupement d'un ensemble d'éléments similaires dans des structures appelées clusters. Le regroupement du graphique du réseau social peut donner beaucoup d'informations sur les attributs cachés sous-jacents, les relations et les propriétés des participants ainsi que les interactions entre eux. Le clustering hiérarchique et la méthode de partitionnement du clustering sont les techniques de clustering couramment utilisées dans la littérature. Dans le clustering hiérarchique, une hiérarchie de clusters est formée. Le processus de création ou de nivellement de la hiérarchie peut être agglomérant ou diviseur. Dans les méthodes de regroupement agglomératif, une approche ascendante du regroupement est suivie. Un nœud particulier est cotisé ou aggloméré avec des nœuds similaires pour former un cluster ou une communauté. Cette agrégation est basée sur la similarité. Dans les approches de clustering de division, un grand cluster est divisé à plusieurs reprises en clusters plus petits. Les méthodes de partitionnement commencent par une partition initiale parmi le nombre de clusters prédéfini et la relocalisation des instances en les déplaçant entre les clusters, par exemple, le clustering K-means. Une évaluation exhaustive de toutes les partitions possibles est nécessaire pour atteindre l'optimalité globale dans le clustering basé sur les partitions. Cela prend du temps et est parfois irréalisable, cause pour laquelle les chercheurs utilisent des heuristiques gourmandes pour l'optimisation itérative dans les méthodes de partitionnement du clustering. La section suivante catégorise et discute des principaux algorithmes de détection de communauté [88].

2.5 Algorithmes pour la détection communautés

Un certain nombre d'algorithmes et de méthodes de détection de communautés ont été proposés et déployés pour l'identification des communautés dans la littérature. Il y a eu également des modifications et des révisions de nombreux procédés et algorithmes déjà proposés. Une enquête complète sur la détection de la communauté dans les graphiques a été réalisée par [89]. D'autres revues disponibles dans la littérature [90, 91, 92].

2.5.1 Détection de communautés basée sur le partitionnement de graphes

Des méthodes basées sur le partitionnement de graphes ont été utilisées dans la littérature pour diviser le graphe en composants de sorte qu'il y ait peu de connexions entre les composants. L'algorithme Kernighan-Line [93] pour le partitionnement de graphes était parmi les premières techniques pour diviser un graphe. Il partitionne les nœuds du graphe avec coût sur les arêtes en sous-ensembles de tailles données de manière à minimiser la somme des coûts sur toutes les arêtes coupées. Un inconvénient majeur de cet algorithme, c'est que le nombre de groupes doit être prédéfini. L'algorithme est cependant assez rapide avec un temps d'exécution qui dans le pire des cas est de $O(n^2)$. Newman [94] réduit la méthode du maximum de vraisemblance largement étudiée pour la détection de communauté à une recherche à travers un groupe de solutions candidates, dont chacune est elle-même une solution à un problème de partitionnement de graphe à coupe minimale. L'article [94] montre que les deux méthodes d'inférence communautaire les plus essentielles basées sur le modèle de bloc stochastique ou sa variante [95] corrigée en degré peuvent être mappées sur des versions du problème familier de partitionnement de graphe à coupe minimale. Ceci a été illustré en adaptant la méthode de partitionnement spectral laplacien [96,97] pour effectuer une inférence communautaire

2.5.2 Détection de communautés basée sur le clustering

La principale préoccupation de la détection de communauté est de détecter des clusters, des groupes ou des sous-groupes cohérents. La base d'un grand nombre d'algorithmes de détection de communauté est le clustering. Parmi les innovateurs des méthodes de détection des communautés, Girvan et Newman [74] qui ont joué un rôle majeur. Ils ont proposé un algorithme de division basé sur l'interdépendance des arêtes pour un graphe avec des arêtes non orientées et non pondérées. L'algorithme se focalise plus sur les arêtes entre les communautés et les communautés sont construites au fur et à mesure en supprimant ces arêtes du graphe d'origine. Trois mesures différentes pour le calcul de l'interdépendance des sommets d'un graphe ont été proposées dans Newman et Girvan [98].

La complexité temporelle dans le pire des cas de l'algorithme d'entre-arêtes est $O(m^2n)$ et est $O(n^3)$ pour les graphes creux, où m désigne le nombre d'arêtes et n le nombre de sommets. L'algorithme de Girvan Newman (GN) a été amélioré par de nombreux auteurs et appliqué à divers réseaux [99-106]. Chen et al [100] ont étendu l'algorithme GN pour partitionner les graphiques pondérés et l'ont utilisé pour identifier les modules fonctionnels dans le réseau protéomique de la levure. Rattigan et al [99] ont proposé les méthodes d'indexation pour réduire considérablement la

complexité de calcul de l'algorithme. GN. Pinney et al [102] ont également construit un algorithme qui utilise l'algorithme GN pour la décomposition des réseaux basé sur le concept théorique des graphes de centralité intermédiaire. Leur article a examiné l'utilité de la centralité de l'intermédiarité pour décomposer ces réseaux de diverses manières.

Radicchi [107] et al ont également proposé un algorithme basé sur l'algorithme GN introduisant une nouvelle définition de la communauté. Ils ont défini les communautés « fortes » et « faibles ». L'algorithme utilise un coefficient de regroupement des arêtes pour effectuer l'étape de suppression des arêtes séparatives de GN et a un temps d'exécution de $O(\frac{m^4}{n^2})$ et $O(n^2)$ pour les graphes creux. Moon et al [108] ont proposé et mis en œuvre la version parallèle de l'algorithme GN pour gérer des données à grande échelle. Ils ont utilisé le modèle MapReduce (Apache Hadoop) et GraphChi. Newman et Girvan ont d'abord défini une mesure connue sous le nom de "modularité" pour juger de la qualité des partitions ou des communautés formées [98]. La mesure de modularité qu'ils proposent a été largement acceptée et utilisée par les chercheurs pour évaluer la qualité des modules obtenus à partir des algorithmes de détection de communauté avec une grande modularité correspondant à une meilleure structure de communauté. La modularité a été définie comme $\sum_i e_{ii} - a_i^2$, où $\boldsymbol{e_{ii}}$ désigne la fraction des bords qui connectent les sommets dans la communauté $\boldsymbol{i}$, $\boldsymbol{e_{ij}}$ dénote la fraction des vertiges reliant les sommets dans deux communautés différentes $\boldsymbol{i}$ et $\boldsymbol{j}$ tandis que $a_i = \sum_i e_{ii}$ est la fraction des bords qui se connectent à sommets dans la communauté $\boldsymbol{i}$. La valeur Q = 1 indique un réseau avec une forte structure communautaire. L'optimisation de la fonction de modularité a reçu une grande attention dans la littérature. Le tableau 1 répertorie les méthodes de détection de communauté basées sur le clustering, y compris les algorithmes qui utilisent la modularité et l'optimisation de la modularité.

Newman [109] a travaillé pour maximiser la modularité afin que le processus d'agrégation de nœuds pour former des communautés conduise à un gain de modularité maximal. Ce changement de modularité lors de la jonction de deux communautés définies comme $\Delta Q = e_{ij} + e_{ji} - 2a_i a_j = 2(e_{ij} - a_i a_j)$ peut être calculé en temps constant et est donc plus rapide à exécuter par rapport à l'algorithme GN. Le temps d'exécution de l'algorithme est $O(n^2)$ pour les graphes creux et $O((m + n)\, n)$ pour les autres. Dans un travail récent, une version évolutive de cet algorithme a été implémentée en utilisant MapReduce par Chen et al [110]. Newman [96] a généralisé l'algorithme d'intermédiarité pour les réseaux pondérés. La modularité était désormais représentée comme :

$$Q = \frac{1}{2m}\sum_{i,j\epsilon v}\left(A_{ij} - \frac{d_i d_j}{2m}\right)\delta(c_i, c_j) \qquad (2.1)$$

Où $m = \frac{1}{2}\sum_{i,j} A_{ij}$ représente le nombre de bords entre les communautés c_i et c_j dans le graphique, tandis que k_i, k_j sont des degrés de vertes i et j tandis que δ (u, v) vaut 1 si $u = v$ et 0 sinon.

Auteur (algorithme)	Approcher	Paramètres	Disponibilité des codes
Newman and Girvan [108]	Division de clustering (utilisant la « modularité » comme indicateur de qualité)	Edge betweenness	https://github.com/kjahan/community
Newman [109,111,112]	Maximisation de la modularité	[109,111] : Modularité [112] : vecteur propre et valeur propre	[109] : http://web.ist.utl.pt/aplf/code/gcf003.html [111]: http://deim.urv.cat/~sergio.gomez/radatools.php#download [112]: http://deim.urv.cat/~sergio.gomez/radatools.php#download
Clauset e al [113]	Optimisation gourmande de la modularité	Arêtes, sommets, modularité	http://www.cs.unm.edu/~aaron/research/fastmodularity.htm
Blondel et al (Louvain Method) [114]	Clustering Hiérarchique	Nœuds, arêtes, Modularité	https://perso.uclouvain.be/vincent.blondel/research/louvain.html
Guimera et al [115] , Zhou et al [116]	Optimisation de la modularité à l'aide du recuit simulé	[115] : Nombre de liens, Probabilité de liaison, non. De modules, non.de partitions, Modularité	NO

		[116] : Nb d'arêtes, inter facteur et intra facteur, Modularité	
Duch et al [117]	Optimisation de la modularité à l'aide de l'optimisation Extrémal	Nombre de nœuds, liens, degré, modularité	http://deim.urv.cat/~sergio.gomez/radatools.php#description
Ye et al (AdClust) [118]	Clustering agglomératif	Sommets, Force, Modularité	NO
Wahl and Sheppard [119]	Clustering spectral flou hiérarchique	Modularité floue, Similitude Jaccard	NO
Falkowski et al (DENGRAPH) [98]	Clustering basé sur la densité	Fonction distance	NO
Dongen et al (MCL) [120]	Clustering Markovienne	Nombre de nœuds	http://www.micans.org/mcl/#source
Nikolaev et al [121]	Clustering basé sur la centralité d'entropie	Matrice de probabilité de transition pour le processus de Markov	NO
Steinhauser et al [122]	Clustering par consensus, marche Aléatoire...	Matrice de similarité, longueur des marches aléatoires	NO

Tableau 2.1 *: Détection de communautés basées sur le clustering [123]*

2.5.3 Détection de communautés basée sur la propagation d'étiquettes

La propagation d'étiquettes dans un réseau est la propagation d'une étiquette à divers nœuds existant dans le réseau. Chaque nœud atteint le label possédé par un nombre maximum de nœuds voisins. Cette section traite de certains algorithmes basés sur la propagation d'étiquettes pour découvrir des communautés. L'algorithme de propagation d'étiquettes (LPA : Label Propagation

Algorithm) a été proposé par Raghavan et al [124] dans lequel chaque nœud essaie initialement d'obtenir une étiquette à partir du nombre maximum d'étiquettes possédées par ses voisins. Le critère d'arrêt du processus était également le même, c'est-à-dire lorsque chaque nœud atteint une étiquette, que possèdent un nombre maximum de ses nœuds voisins. Chaque itération de l'algorithme prend $O(m)$ temps où m est le nombre d'arêtes. SLPA (speaker listener label propagation Algorithm) [125] est une extension de LPA qui pourrait analyser différents types de communautés telles que les communautés disjointes, les communautés qui se chevauchent et les communautés hiérarchiques dans les deux cas. Réseaux uni-partîtes et bi- partites. L'algorithme a un temps d'exécution linéaire de $O(Tm)$, où T est le nombre maximal d'itérations défini par l'utilisateur et m est le nombre d'arêtes. Sur la base de l'algorithme SLPA, Hu [126] a proposé un algorithme de propagation pondérée des étiquettes (WLPA : Weighted Label Propagation Algorithm). Il utilise la similarité entre deux des sommets d'un réseau sur la base des étiquettes des sommets obtenus lors de la propagation des étiquettes. La similarité de ces sommets est ensuite utilisée comme poids de l'arête dans la propagation de l'étiquette.

2.5.4 Détection de communautés basée sur des algorithmes génétiques (GA)

Les algorithmes génétiques (AG) sont des algorithmes de recherche heuristiques adaptatifs dont le but est de trouver la meilleure solution dans des circonstances données. Un algorithme génétique commence par un ensemble de solutions connues sous le nom de chromosomes et la fonction de fitness est calculée pour ces chromosomes. Si une solution avec une fitness maximale est obtenue, on s'arrête sinon avec une certaine probabilité des opérateurs de croisement et de mutation sont appliqués à l'ensemble actuel de solutions pour obtenir le nouvel ensemble de solutions. La détection de communauté peut être considérée comme un problème d'optimisation dans lequel une fonction objective qui capture l'intuition d'une communauté avec une meilleure connectivité interne que la connectivité externe est choisie pour être optimisée. Les AG ont été appliquées au processus de découverte et d'analyse de la communauté dans quelques travaux de recherche récents. Ceux-ci sont décrits brièvement dans cette section. Le tableau 2 répertorie les algorithmes disponibles dans la littérature pour la détection de la communauté basée sur GA [123].

Auteur (Algorithme)	Approcher	Paramètres	Disponibilité des codes
Pizzuti (GA-Net) [127]	Score communautaire en tant que fonction de fitness	Score communautaire	http://staff.icar.cnr.it/ piz zuti/codes.html
Pizzuti (MOGA-Net) [128]	Optimisation multi-objectifs	Score communautaire Communauté Fitness	http://staff.icar.cnr.it/ piz zuti/codes.html
Hafez et al [129]	Objectif unique, multi Optimisation objective	Nombre de gènes, mutation Opérateurs de croisement	No
Mazur et al [130]	Score communautaire et modularité comme fonctions de fitness	Fonctions de fitness	No
Liu et al [131]	Algorithme génétique et clustering	Taille de la population, Maximum numéro de génération, maximum no. De générations pour la fraction chromosomique la plus appropriée des moyeux minés, nombre de communautés	NO
Tasgin et al [132]	Optimisation de la modularité	Modularité, taille de la population, nombre de chromosomes	NO
Zadeh [133]	Algorithme culturel multi-population	BS_average, BSN	NO

Tableau 2.2 : *Détection de communautés basées sur les algorithmes génétiques*

2.6 Méthodes pour détecter les communautés qui se chevauchent

Une étude récente d'Amelio et al donne un examen complet des principaux algorithmes de détection de communautés qui se chevauchent et inclut les méthodes sur les réseaux dynamiques. Il existe une autre revue des méthodes pour découvrir les communautés qui se chevauchent faite par Xie et al [134]. La section suivante traite de certaines des méthodes de détection des communautés qui se chevauchent.

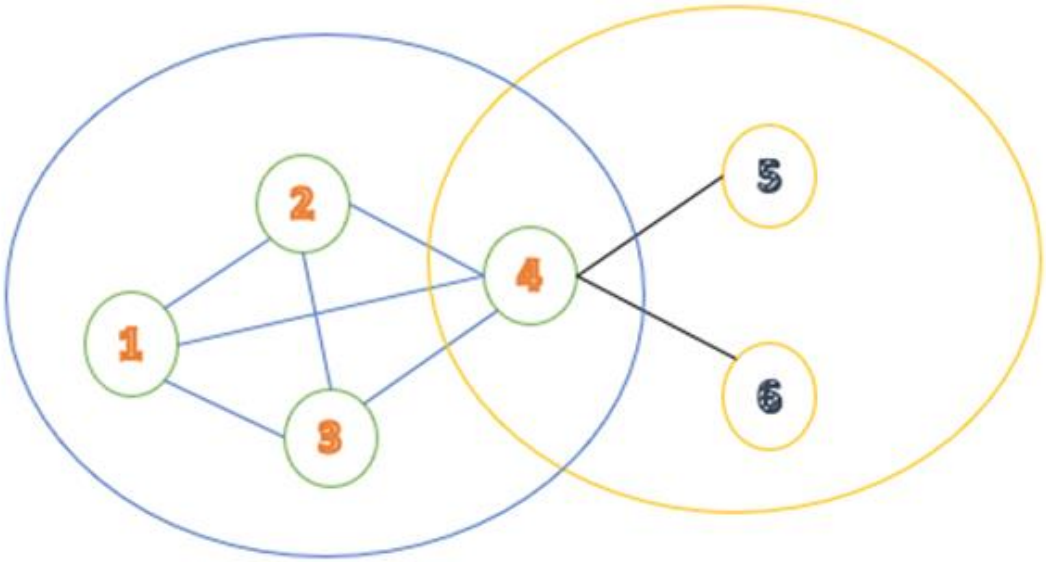


Figure 2.3 : *Détection des communautés chevauchantes*

2.6.1 Méthodes basées sur les cliques pour la détection de communautés qui se chevauchent

Une communauté peut être interprétée comme étant une union de petits sous-graphes complets (entièrement connectés) qui partagent des nœuds. Une k-clique est un sous-graphe entièrement connexe composé de k nœuds. Une communauté k-clique peut être définie comme l'union de toutes les k-cliques qui peuvent être atteintes les unes des autres à travers une série de k-cliques adjacentes. De nombreux chercheurs ont utilisé des cliques pour détecter les communautés qui se chevauchent. Les contributions importantes utilisant des cliques pour la détection de communautés qui se chevauchent sont résumées dans le tableau 3[123].

Auteur (algorithme)	Approcher	Paramètres	Disponibilité des codes
Palla et al (CPM) [135]	Méthode de percolation de clique	Nœuds, poids de seuil	http://igraph.wikidot.com/community-detection-in-r, http://www.cfinder.org/
Lancichinetti et al [136]	Fonctions de fitness	Fonctions de fitness	NO
Du et al (ComTector) [137]	Clustering basé sur les noyaux	Ensemble de tous les noyaux	No
Shen at al (EAGLE) [138]	Clustering hiérarchique agglomératif	Similarité entre deux communautés	No
Evans et al [138-140]	Graphique linéaire, graphique de clique	Liens, partition	NO
Lee et al (GCE) [141]	Expansion basée sur les cliques	Fonctions de fitness	https://sites.google.com/site/greedy
Gregory et al (CONGA, CONGO [142] Peacock algorithm [143])	Split betweenness	[103]: vertex, split betweenness [142]: Local betweenness Chemins courts [143]: rapport de edge betweenness and max. split betweenness	[103,142,143]: http://www.cs.bris.ac.uk/~steve /networks/

Tableau 2.3 : *Méthodes basées sur les cliques pour la détection de communautés qui se chevauchent*

La méthode de percolation de clique (MPC) a été proposée par Palla et al [134] pour détecter les communautés qui se chevauchent. La méthode trouve d'abord toutes les cliques du réseau et utilise l'algorithme d'Everett et al [144] pour identifier les communautés par l'analyse des composants de la matrice de chevauchement clique-clique. MPC a une durée d'exécution de$O(\exp(n))$. La méthode de percolation de clique proposée par [134] n'a pas pu découvrir la structure hiérarchique ainsi que l'attribut de chevauchement. Cette limitation a été surmontée grâce à la méthode proposée par Lancichinetti et al [135]. Il effectue une exploration locale afin de trouver la communauté pour chacun des nœuds. Dans ce processus, les nœuds peuvent être revisités un certain nombre de fois. L'objectif principal était de trouver des maximas locaux basés sur une fonction de fitness. Le logiciel CFinder [145] a été développé à l'aide de MPC pour la détection de communautés qui se chevauchent. Du et al [136] ont proposé ComTector (Community DeTector) pour la détection des communautés qui se chevauchent en utilisant des cliques maximales. Initialement, toutes les cliques maximales du réseau sont trouvées et forment les noyaux de la communauté potentielle. Ensuite, la technique agglomérative est utilisée de manière itérative pour ajouter les sommets restants à leurs noyaux les plus proches. Les clusters obtenus sont ajustés en fusionnant des paires de communautés fractionnaires afin d'optimiser la modularité du réseau. Le temps d'exécution de l'algorithme est ($C*T2$), où les communautés détectées sont notées par *C. T* est le nombre de triangles dans le réseau. EAGLE, un algorithme basé sur le clustering hiérarchique agglomératif a été proposé par Shen et al [137]. Dans la première étape, les cliques maximales sont découvertes et celles inférieures à un seuil sont rejetées. Les cliques maximales subordonnées sont négligées. La similitude se trouve entre ces communautés, et les communautés sont à plusieurs reprises fusionnées sur la base de cette similitude. Ceci est répété jusqu'à ce qu'il reste une communauté à la fin. Evans et al [138] ont proposé qu'en partitionnant les liens d'un réseau, les communautés qui se chevauchent puissent être découvertes. Dans le prolongement de ce travail, Evans et al [139] ont utilisé des graphiques linéaires pondérés. Dans un autre travail, Evans [140] a utilisé des graphes de cliques pour détecter les communautés qui se chevauchent dans les réseaux sociaux du monde réel. GCE (Greedy Clique Expansion) [141] identifie d'abord les cliques dans un réseau. Ces cliques agissent comme des graines pour l'expansion avec l'optimisation gourmande d'une fonction de fitness. Une communauté est créée en élargissant la graine sélectionnée et en effectuant son optimisation gourmande via la fonction de fitness proposée par [134]. CONGA (Cluster-Overlap Newman Girvan Algorithm) a été proposé par [103]. Cette méthode était basée

sur l'algorithme split-betweenness de Girvan-Newman. Le temps d'exécution de la méthode est $O(m3)$.

2.7 Détection de communautés dynamique

Les réseaux dynamiques sont des réseaux évoluant dans le temps. Dès lors, une nouvelle définition de communautés dynamiques s'impose : c'est une succession de communautés statiques [146]. L'évolution des communautés dynamiques peut se faire de diverses manières :

- La croissance et la contraction : correspondant à l'ajout et au retrait de nœuds d'une communauté existante.
- La naissance et la mort de communautés : des nouvelles communautés peuvent apparaître, et d'anciennes communautés peuvent disparaître avec l'évolution de réseau.
- La fusion et la division de communautés : Deux communautés - ou plus - peuvent en effet, se fusionner en une seule au cours du temps. De manière semblable, une communauté peut se diviser en deux ou plus en communautés, plus petites que celle dont elles sont issues.

2.7.1 Approches par détections statiques successives

Etant donné qu'elles ne font intervenir que des détections de communautés statiques suivies d'un post-traitement, ces méthodes sont considérées comme les plus simples. Cependant, toutes ces méthodes soufrent du problème de l'instabilité de la détection [147].

2.7.2 Approches par détections statiques informées successives

Il s'agit ici des approches qui utilisent toujours des instantanés, et effectuent une détection pour chacun d'entre eux. Cependant, pour résoudre le problème de l'instabilité des algorithmes, ces méthodes proposent de prendre en compte les résultats obtenus à l'étape t lors de la détection des communautés à l'étape t + 1. Ceci réduit l'instabilité, car, au cas où l'algorithme ne saurait lequel choisir entre deux découpages différents, il pourrait par exemple prendre le plus semblable au découpage précédent [148].

2.7.3 Approches travaillant sur des réseaux temporels

Ici, l'évolution du réseau n'est plus considérée comme une succession d'instantanés, mais comme une succession de modifications sur le réseau. Il s'agit donc de prendre en compte la ou les dernières modifications effectues sur le réseau, et de modifier les communautés existantes en conséquence. Il n'y a plus de problème d'instabilité ici, les communautés perdurant naturellement.

Toutefois, d'autres problèmes se posent, tel que l'aspect très local des modifications, qui peut entrainer une dérive vers des communautés qui ne sont plus valables à un moment donné, par rapport à l'état du réseau à cet instant (la détection de communauté n'étant jamais faite sur l'ensemble du réseau, mais uniquement par petites modifications locales successives) [149].

2.8 Système de recommandation basé sur la détection de communautés

Les systèmes de recommandation basés sur la communauté suggèrent des ressources, tels que des produits et des services, en fonction de groupes d'utilisateurs qui manifestent des préférences et des comportements similaires. Les informations extraites des services de réseaux sociaux promettent d'améliorer la précision des systèmes de recommandation dans divers domaines. Dans ce contexte, les techniques de détection de communauté nous aident à mieux comprendre le comportement collectif des utilisateurs en regroupant les utilisateurs similaires dans leurs intérêts, préférences et activités.

Tirer parti des communautés identifiées en analysant les grands réseaux sociaux dans le processus de recommandation est complexe et nécessite plusieurs étapes de traitement, tel que résumé au sein de la Figure 2.4 [150].

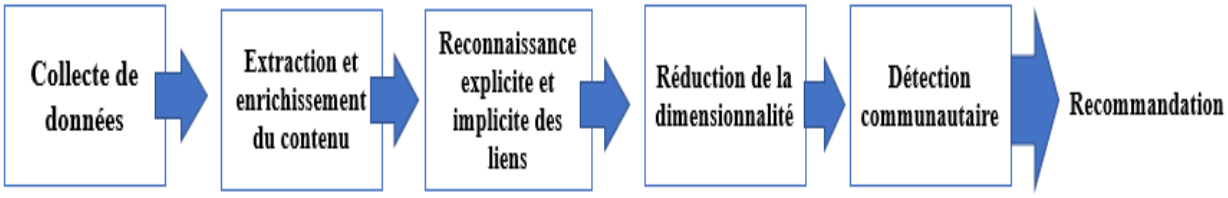


Figure 2.4 : *Étapes de la recommandation basée sur les communautés*

Après avoir collecté suffisamment d'informations pour représenter les relations sociales, les intérêts, les préférences et les activités des utilisateurs, l'étape de reconnaissance des liens exploite ces données pour formaliser les relations utilisateur-utilisateur et utilisateur-article. Le système analyse ces liens pour découvrir des communautés d'utilisateurs ayant des goûts similaires. La réduction de la dimensionnalité est souvent envisagée pour réduire la rareté des données. Les détails relatifs à chaque étape sont discutés dans les sections suivantes.

Collecte de données, extraction de contenu et enrichissement

La première étape interagit avec le service en ligne afin de collecter des informations pertinentes. Les données sont collectées en agrégeant toutes les interactions utilisateur-utilisateur

et utilisateur-item observées dans le réseau sur une période de temps. La sortie est un seul graphe G.

Afin de fournir des recommandations précises, SR collecte généralement des informations supplémentaires sur les utilisateurs en procédant à l'extraction et à l'enrichissement du contenu. Généralement, ces informations prennent la forme d'attributs représentant des caractéristiques démographiques (par exemple, le niveau de revenu, l'âge, l'emplacement géographique, la profession) et les intérêts pour lesquels une recommandation est recherchée (par exemple, les loisirs, les livres préférés et les goûts musicaux).

Dans les réseaux sociaux, les utilisateurs sont souvent caractérisés par leur historique d'activités en termes de contenu publié. Prenons un scénario, où l'utilisateur partage un article de presse sur la question des plastiques dans l'océan. Si nous sommes capables de dériver des concepts tels que l'activisme environnemental, la protection de la biodiversité et la durabilité ; nous pouvons mieux caractériser les intérêts et les objectifs de l'utilisateur. De la même manière, les clients des services de commerce électronique sont liés aux produits qu'ils ont évalués ou achetés dans le passé. La classification des clients est basée sur les caractéristiques des produits qu'ils ont achetés. Les caractéristiques communes sont les sujets, les balises et les concepts extraits des documents texte partagés et les genres musicaux attribués aux chansons écoutées. Il s'agit d'un exemple typique de modélisation d'utilisateurs, où des représentations explicites de certaines caractéristiques des utilisateurs sont acquises au cours de leurs activités normales sur la plateforme en ligne [151]. En règle générale, la sortie de l'extraction de contenu est un graphique attribué où chaque caractéristique extraite est représentée par un attribut de nœud. Des représentations équivalentes sont basées sur des matrices utilisateur-item qui peuvent représenter des activités d'utilisateur telles que des achats ou des évaluations.

Par exemple, Tchuente et al. [152] représentent explicitement des profils d'utilisateurs avec des vecteurs traditionnels dans un espace de grande dimension [151], où chaque dimension (ou attribut) dénote le niveau d'appartenance à une ou plusieurs communautés. Une taxonomie externe fournit une organisation hiérarchique des sujets associés à chaque communauté, de sorte que les profils ont différents niveaux de granularité, par exemple, Sports → Sports aériens → Parachutisme → Freeflying.

Lalwani et al. [153] ont développé un système de recommandation social utilisant la détection de communauté et le filtrage collaboratif comme technique d'interactions sociales au

niveau de la communauté. Le graphe social utilisateur-utilisateur a été analysé pour extraire la relation entre les amitiés en utilisant un algorithme de détection de communauté. La méthode de filtrage collaboratif a été utilisée pour la prédiction de notes, basée sur la matrice utilisateur-item. Le framework MapReduce a été développé pour présenter un système de recommandation social basé sur la communauté et un filtrage collaboratif évolutif. Le taux de couverture et l'évolutivité ont été améliorés en utilisant ces méthodes par rapport aux algorithmes traditionnels. De plus, le problème de démarrage à froid a été traité efficacement par cette approche, mais n'ont pas réussi à se concentrer sur le problème de rareté des données.

Parc et al. [154] ont développé un filtrage collaboratif inversé (FCI) pour fournir une meilleure recommandation, où le temps de traitement du FCI a été considérablement réduit en utilisant l'algorithme des k-plus proches voisins (KNN) comme filtrage glouton. Pour améliorer les performances de FCI, avant d'exécuter l'algorithme KNN, l'approche de pondération TF-IDF a été appliquée. Les résultats expérimentaux ont indiqué que la méthode FCI donne un meilleur temps de traitement et de prétraitement des requêtes avec une précision de haut niveau. La méthode FCI n'a pas été en mesure de prédire les notes des items non notés. Dans le pire des cas, l'algorithme était plus lent que l'index inversé, en raison des performances de l'algorithme gourmand, qui dépendait fortement de l'ensemble de données.

Shi et al. [155] ont développé l'approche basée sur les méta-chemins pour identifier la similarité des utilisateurs ou des items. Selon l'approche de régularisation double, l'intégration de divers types d'informations était flexible, là où cette approche a été considérée comme SimMF. Les informations d'attribut des items et des utilisateurs ont été largement utilisées pour améliorer la précision des recommandations. Les performances de recommandation doivent être améliorées en combinant les matrices de similarité avec une stratégie intelligente d'apprentissage des poids.

Lee et Tseng [156] ont amélioré les performances de prédiction en développant deux approches incorporant les informations contextuelles. Les notes des items ont été estimées en intégrant les informations contextuelles dans des techniques de calcul, où deux types de méthodes de filtrage collaboratif incluent le filtrage collaboratif basé sur un modèle et basé sur la mémoire. L'approche était suffisante, efficace et facile à mettre en œuvre, de plus, les performances de prédiction ont été améliorées en résolvant les problèmes d'incertitude des données à l'aide des informations de la procédure d'apprentissage. Lorsque le nombre d'utilisateurs et d'items augmente,

la méthode de filtrage collaboratif deviennent plus difficiles à résoudre, ce qui est une limitation de cette approche.

Sobhanam et al.[157] ont abordé le problème CS en utilisant les techniques de clustering et les règles d'association. La méthodologie proposée pour résoudre le problème de CS comprend la génération d'un profil d'utilisateur basé sur la taxonomie en fonction des informations obtenues auprès des utilisateurs. Les modèles fréquents sont extraits de l'ensemble de données transactionnel et sont construits à partir du profil d'utilisateur basé sur la taxonomie. Les règles d'association entre les sujets qui intéressent les utilisateurs peuvent être dérivées des modèles et ces règles nous permettent de découvrir les sujets qui apparaissent fréquemment ensemble. Les rubriques du profil développé sont normalisées. Maintenant, les profils d'utilisateurs qui contiennent le classement des utilisateurs pour les films peuvent être utilisés pour résoudre le problème du nouvel item.

Dans Hui et al [158], ont amélioré l'efficacité de la recommandation basée sur les réseaux sociaux, ils ont proposé deux modèles qui intègrent la régularisation de la communauté de chevauchement dans le cadre de factorisation matricielle. L'idée d'utiliser les communautés pour améliorer la précision de la prédiction des notes ne doit pas se limiter aux systèmes de recommandation basés sur les réseaux sociaux. Lorsque plus d'informations sont disponibles, il est également possible d'envisager des communautés d'items et d'intégrer la régularisation des communautés d'items dans leurs modèles. Des communautés d'items peuvent être obtenues via le regroupement d'items sur la base des caractéristiques des items ou du réseau bipartite utilisateur-item. De plus, les relations explicites (par exemple, les goûts similaires et les interactions fréquentes) peuvent également être prises en considération au lieu de ne considérer que les relations sociales implicites. En tenant compte des informations provenant à la fois des relations implicites et du réseau d'items, les systèmes de recommandation traditionnels sans réseau social de support peuvent bénéficier des modèles de recommandation sociale et l'idée du filtrage collaboratif social peut être appliquée dans un contexte beaucoup plus large.

Un système de recommandation sociale utilisant la détection de communauté basée sur la propagation d'étiquettes est proposé par Xinchang et al [159]. Dans ce travail, les auteurs ont appliqué la détection de communauté basée sur la propagation d'étiquettes et le filtrage collaboratif dans le système de recommandation de films.

Fatemi et al [160] ont proposé un nouveau système de recommandation sociale basé sur la communauté (CBSRS) qui utilise les nouveaux types de données disponibles dans les réseaux

sociaux. Il utilise les relations implicites entre les éléments dérivés des interactions directes des utilisateurs avec eux, et représente les graphes des réseaux sociaux des éléments. L'étude montre que le CBSRS est un réseau d'intérêt générique et qu'il l'utilise comme base pour détecter les communautés et fournir des recommandations plus précises et personnalisées. Il est également démontré qu'en raison de sa nature communautaire, le CBSRS proposé aborde cinq problèmes différents que les autres systèmes de recommandation n'ont pas réussi à résoudre. À savoir, il fournit une recommandation fortuite, surmonte les résultats de recommandation surspécialisés et répétitifs, recommande à un nouvel utilisateur et peut ainsi résoudre le problème du démarrage à froid. En effet, les nouveaux films peuvent être connectés au réseau générique d'intérêt après un seul examen. Le problème de manque de données associé aux recommandeurs est également abordé ici, car CBSRS peut presque toujours fournir des recommandations même avec peu de données disponibles. Enfin, il fournit des recommandations à des groupes ainsi qu'à des individus. Les articles recommandés sont classés en différentes catégories, tout en correspondant au mieux aux intérêts des utilisateurs, et en tenant compte de la popularité globale et de la qualité perçue des articles parmi tous les évaluateurs du réseau. Dans leurs expériences sur les données disponibles sur IMDb, ils ont montré des améliorations significatives par rapport aux approches existantes dans les systèmes de recommandation.

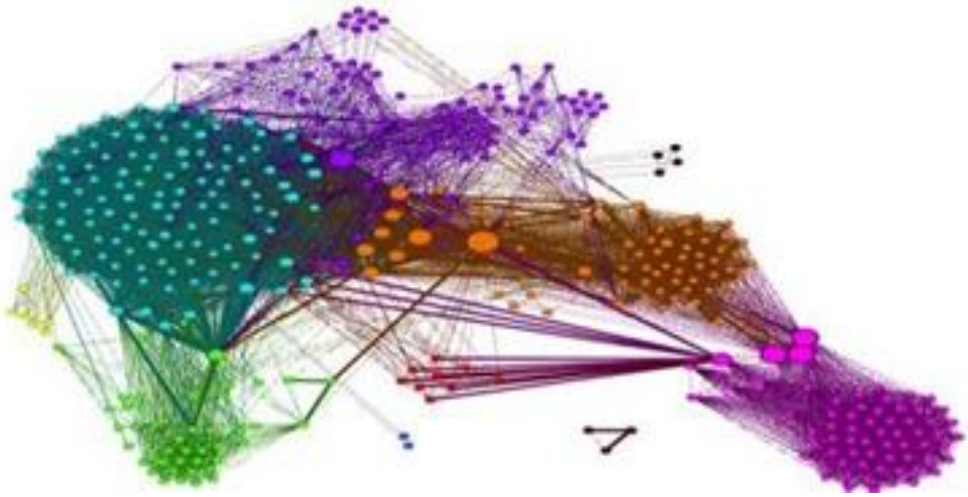


Figure 2.5 : *Distribution des genres pour toutes les communautés*

2.9 Conclusion

Nous avons présenté au sein de ce chapitre les concepts de base relatives aux réseaux sociaux, graphe et communautés. Nous avons également discuté les principaux algorithmes de détection de communautés, et on a terminé ce chapitre par un survol sur quelques travaux sur les systèmes de recommandation à base de détection de communautés.

Chapitre III

Vers une approche de recommandation collaborative à base de communautés

1.1 Introduction

Les systèmes de recommandation cherchent à prédire la « note » ou la « préférence » que l'utilisateur donnerait à un item sur la base d'ensembles de données de notation historiques. À l'ère de l'informatique, Internet a la capacité de refléter le réseau social, ce qui permet d'intégrer la détection de la communauté et des algorithmes de clustering pour améliorer leurs performances. Autrement dit, nous pouvons découvrir les informations cachées présentes dans le réseau social en utilisant des algorithmes de détection de communauté et utiliser la technique de regroupement pour révéler la préférence des utilisateurs sur la base de leur historique comportemental. Ces informations sont utilisées comme rétroaction pour notre système de recommandation afin d'améliorer la précision des prédictions. Dans ce chapitre nous détaillons notre proposition avec la présentation de leur architecture générale. Ensuite nous mettons en évidence le dataset utilisé dans l'expérimentation ainsi que la discussion des résultats obtenus.

2.2 Problématique et objectif

Dans le commerce électronique, l'opinion des utilisateurs sur les produits et les avis sont identifiés à l'aide de systèmes de recommandation. La technique de filtrage collaboratif est une technique couramment utilisée et devenu l'approche la plus populaire pour développer des systèmes de recommandation dans diverses applications commerciales. Pour donner des recommandations aux utilisateurs. Malheureusement, des problèmes tels que le problème du démarrage à froid (c'est-à-dire que de nouveaux utilisateurs ou items entrent dans le système et pour ceux-ci aucune information sur les préférences précédentes n'est disponible) et le problème de manque de données (c'est-à-dire le nombre d'items candidats à la recommandation est souvent énorme et les utilisateurs ne notent qu'un petit sous-ensemble des items disponibles) sont largement reconnus pour entraver l'efficacité des recommandations. Le but de ce travail est d'atténuer ces problèmes afin d'amélioré la qualité de recommandation et fournir des recommandations pertinentes. Pour ce faire nous proposons une approche de recommandation collaborative basée sur communautés.

3.3 Solution proposée

Ce travail présente la conception et le développement d'un système de recommandation collaboratif basé sur communautés. Nous appliquons la méthode de détection de communautés de Louvain pour découvrir les communautés des utilisateurs en analysant le graphe de similarité

démographique utilisateur-utilisateur. La méthode de génération de recommandations est basée sur l'idée d'utilisé le score IF-ICF (Item Frequency-Inverse Community Frequency) [2] de chaque item dans la communauté de l'utilisateur cible avec une mesure de similarité qui combine le coefficient de corrélation de Pearson avec le coefficient de Jaccard. Les scores IF aident à trouver l'ensemble d'items qui sont uniques à une communauté particulière. Les valeurs de l'ICF sont inversement proportionnelles au nombre de communautés dans lesquelles un item a été évalué. ICF est utilisé pour calculer le caractère unique de l'item dans les communautés. Les scores IF-ICF des items sont en outre utilisés pour trouver les scores de prédiction des items non vus par l'utilisateur afin de présenter un ensemble de meilleures recommandations à l'utilisateur. Un prototype du système est développé en python et une analyse expérimentale a été réalisée pour le domaine du film.

La figure 3.1 présente l'architecture générale de notre solution proposée.

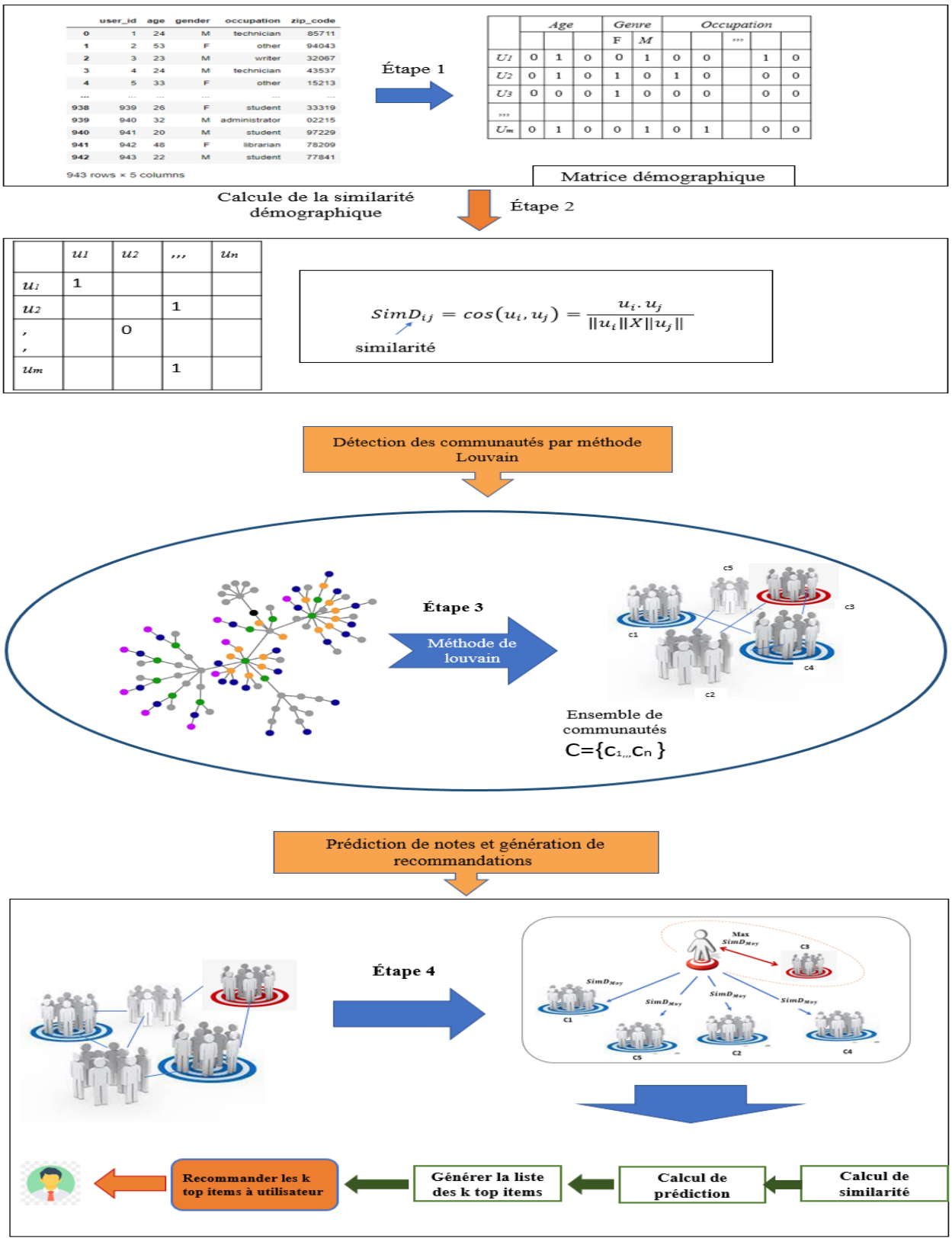


Figure 3.1 : *Architecture générale de l'approche proposée*

Les prochaines sections décriront en détail les quatre étapes qui constituent notre proposition.

3.3.1 Calcul de la similarité démographique des utilisateurs

Afin de créer des communautés d'utilisateurs, notre algorithme prend en entrée une matrice binaire qui contient l'information démographique des utilisateurs (tels que : l'âge, sexe et l'occupation) et utilise la métrique standard (similarité cosinus) pour calculer la similarité démographique de deux utilisateurs. Le résultat est un réseau pondéré où les nœuds représentent les utilisateurs et chaque bord pondéré représente la valeur de similarité des utilisateurs qu'il connecte.

Pour générer la matrice démographique binaire, nous construisons dans un premier temps un vecteur démographique pour chaque utilisateur qui est représenté par un vecteur binaire de 26 attributs (3 pour l'âge, 2 pour le sexe et 21 pour l'occupation) et nous effectuons un échantillonnage basé sur de différentes tranches d'âge des utilisateurs. Les tranches d'âge étudiées sont données dans le tableau 3.1. Nous divisons l'âge en trois catégories, lorsque l'âge est vectorisé à trois valeurs, des utilisateurs similaires peuvent être mieux regroupés et les résultats de la prédiction sont meilleurs.

000, si l'Age de l'utilisateur < 15
010, Si 15 ≤ l'Age de l'utilisateur ≤ 55
000, l'Age de l'utilisateur > 55

Tableau 3.1 : *Valeurs de caractéristique de l'âge de l'utilisateur u*.

10, si l'utilisateur u est male
01, si l'utilisateur u est femelle

Tableau 3.2 : *Valeurs de caractéristique du genre de l'utilisateur u*

1 si l'utilisateur prend sa profession
0 si l'utilisateur ne prend pas la profession

Tableau 3.3: *Valeurs de caractéristique de l'occupation de l'utilisateur u*

Soit u_i le vecteur démographique de l'utilisateur i et u_j le vecteur démographique de l'utilisateur j pour les items. La similarité $SimD_{ij}$ entre l'utilisateur i et l'utilisateur j peut être mesurée par la similarité en cosinus entre les vecteurs :

$$SimD_{ij} = cos(u_i, u_j) = \frac{u_i.u_j}{\|u_i\| X \|u_j\|} \qquad (3.1)$$

Les similarités peuvent être représentées dans un réseau (graphe), le réseau de similarité démographique des utilisateurs, qui relie chaque couple d'utilisateurs associés avec une arête

3.3.2 Détection des communautés

Cette étape a pour objectif de trouver les communautés des utilisateurs, en acceptant comme entrée le réseau de similarité démographique des utilisateurs qui a été construit à l'étape précédente. Pour identifier les communautés des utilisateurs, nous appliquons l'algorithme de détection de communautés Louvain [161] dans le graphe de similarité démographique utilisateur-utilisateur.

3.3.3 Affectation communautaire de l'utilisateur cible

Une fois la méthode de Louvain appliquée comme étape de prétraitement, le jeu de données sera divisé en communautés. Ensuite, la valeur de $SimD_{ij}(u_i,u_j)$ est trouvée pour l'utilisateur cible u_i par rapport à tous les autres utilisateurs u_j dans les communautés découvertes à l'aide de l'équation (3.1).

Après avoir trouvé les valeurs $SimD_{ij}$, la similarité moyenne $SimD_{Moy}$ est calculée pour chaque communauté. Ce qui est donnée par la formule ci-dessous :

$$SimD_{Moy} = \frac{\sum_{j=1}^{n} SimD(u_i,u_j)}{n} \qquad (3.2)$$

Où n est le nombre total d'utilisateurs dans la communauté C. La valeur maximale du $SimD$ représente une distance minimale entre l'utilisateur cible et les autres membres de la communauté. Par conséquent, l'utilisateur cible est affecté à une communauté avec laquelle il a un score moyen maximum de points communs. Ce processus est décrit dans la Figure 3.2.

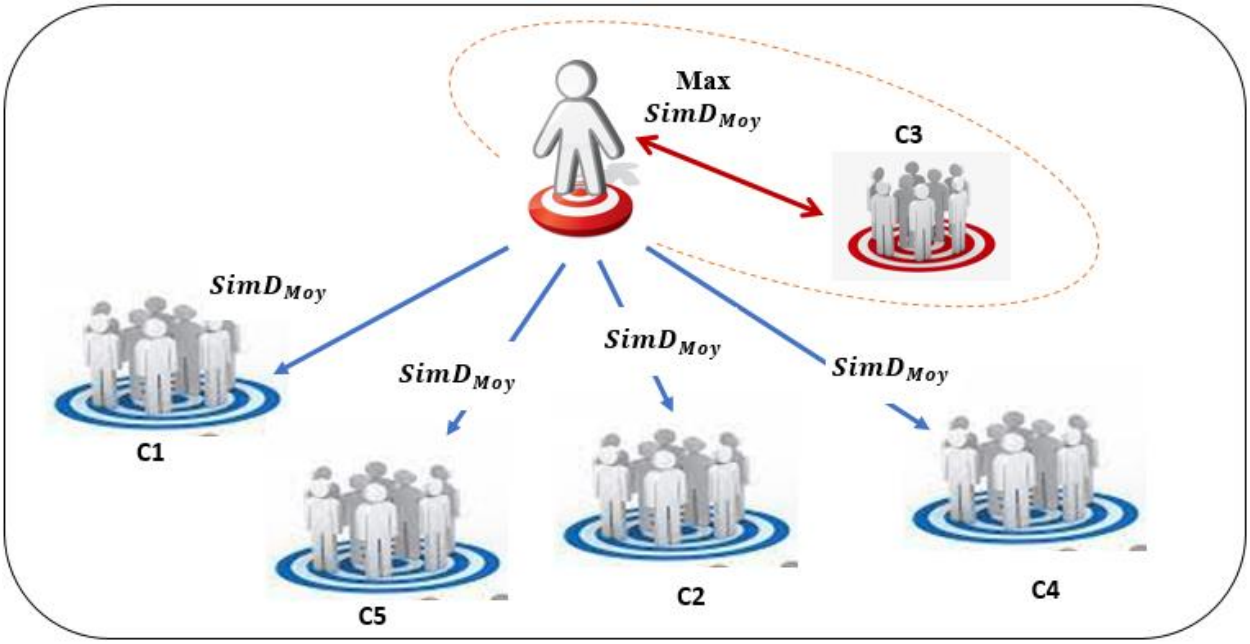


Figure 3.2 : *Affectation de l'utilisateur cible à la communauté.*

3.3.4 Prédiction de notes et génération de recommandations

Dans cette étape, les modèles appris des communautés seront exploités pour aider le système de recommandation à prédire les futures préférences des utilisateurs actifs en fonction de certaines catégories données par les communautés.

Tout d'abord, nous calculons l'importance d'un item comme étant une valeur de fréquence d'item-fréquence communautaire inverse (IF-ICF). En utilisant ces valeurs, dont l'objectif est de déterminer l'ensemble d'items qui sont uniques à la communauté à laquelle appartient l'utilisateur cible. Le score IF-ICF est proposé par Sharma et al [2].

IF-ICF est un modèle de pesée qui aide à trouver l'importance de chaque item appartenant à la communauté d'utilisateurs.

Fréquence de l'item (IF) : Elle découvre le nombre de fois où l'item a été évalué dans la communauté par rapport au nombre total d'évaluations de l'item dans la communauté. Il est donné par :

$$IF = \frac{Nombre\ de\ fois\ où\ l'item\ est\ noté\ dans\ la\ communauté}{Nombre\ total\ d'évaluations\ d'items\ dans\ la\ communauté} \qquad (3.3)$$

Fréquence communautaire inverse (ICF) : Elle est utilisée pour calculer le caractère unique de l'item dans les communautés. Si l'item a été largement évalué, il obtient un score ICF faible. ICF a été défini comme suit :

$$ICF = \log\left(\frac{Nb_c}{Nb_r}\right) \quad (3.4)$$

Où Nb_c : est le nombre total de communautés,

Nb_r : est le nombre total de communautés où l'item a été noté.

Calcul de la prédiction basée sur IF-ICF (score pour la génération de recommandations) : Après avoir obtenu les scores IF-ICF, nous utilisons la formule de prédiction la plus populaire pour un système de recommandation basé sur l'utilisateur pour trouver les scores de prédiction pour les items. Les valeurs IF-ICF sont ensuite utilisées dans la formule de prédiction en tant que facteur d'échelle similaire au travail effectué par Bedi et al. [162]. L'équation du score de prédiction est donnée par :

$$\hat{R}_{u_i} = \bar{r}_{u_i} + \text{IF.ICF} \sum_{U_j \in \text{Kn}_{u_i}} \text{sim}_g(u_i, v_j)(r_{u_j} - \bar{r}_{u_j}) \quad (3.5)$$

Où $\bar{r}_{u_i}$ et $\bar{r}_{u_j}$ désignent la moyenne des évaluations de l'utilisateur u et v.

Kn_{u_i}: *Kn* est la communauté d'utilisateurs cible (K plus proche voisins) de taille k où chaque utilisateur u a évalué l'item

r_{u_j} : désigne la note attribuée par l'utilisateur v (voisin) à un item i.

IF-ICF : Score de fréquence de l'item (IF) et de fréquence communautaire inverse (ICF) de l'item.

$sim_g(u_i, v_j)$*:* est une mesure de similarité qui combine le coefficient de corrélation de Pearson avec le coefficient de Jaccard.

$$sim_g(u_i, u_j) = sim_p(u_i, u_j) \ * \ sim_j(u_i, u_j) \quad (3.6)$$

Avec :

$$sim_p(u_i, u_j) = \frac{\sum_{a \in I}\left(r_{u_i,a} - \overline{R_{u_i}}\right)\left(r_{u_j,a} - \overline{R_{u_j}}\right)}{\sqrt{\sum_{k \in I}\left(r_{u_i,a} - \overline{R_{u_i}}\right)^2}\sqrt{\sum_{k \in I}\left(r_{u_j,a} - \overline{R_{u_j}}\right)^2}} \quad (3.7)$$

$$sim_j(u_i, u_j) = \frac{u_i \cap u_j}{u_i \cup u_j} \quad (3.8)$$

Où $sim_p(u_i, u_j)$ est la similarité entre les utilisateurs u_i et u_j, *I* est l'ensemble des items notés par les deux utilisateurs u_i et u_j, $r_{u_i,k}$ est la note de l'utilisateur u_i sur l'item *k*, $\overline{R_{u_i}}$ est la note moyenne de l'utilisateur u_i pour les *I* items.

Les scores de prédiction sont trouvés pour tous les items. Ces derniers sont ensuite disposés dans l'ordre décroissant de ces valeurs et les k premiers items sont présentés à l'utilisateur. L'utilisateur se voit donc présenter une liste de recommandations qui intègre à la fois les préférences de la communauté dans son ensemble et les choix individuels de l'utilisateur.

Algorithme 1

Entrée : Matrice démographique des utilisateurs, Matrice d'évaluations utilisateur-item

U= {u_1, u_2, ..., u_n} est l'ensemble des utilisateurs

I= {i_1, i_2, i_3, ..., i_n} est l'ensemble des items.

Sortie : Scores de prédiction pour les items recommandés

Calcul de la similarité démographique des utilisateurs

Pour tous les utilisateurs du système

Pour chaque utilisateur u_i ;

Pour tout autre utilisateur u_j ;

Trouver la similarité Cosinus, cos (u, v)

Avec $SimD = cos(u_i, u_j) = \frac{u_i . u_j}{\|u_i\| X \|u_j\|}$

Retourner la matrice *SimD* // *SimD :* poids des arêtes

Appliquer la méthode de Louvain sur la de similarité démographique

Retourner l'ensemble des communautés *C* ($C_1, C_2, ..., C_m$)

Pour l'utilisateur cible actuel *u*

Pour chaque communauté C

Calculer $SimD_{Moy}$;

$$SimD_{Moy} = \frac{\sum_{j=1}^{n} SimD(u_i, u_j)}{n}$$

Avec n est le nombre total d'utilisateurs dans la communauté.

Fin

Assigner l'utilisateur u à la communauté avec la moyenne maximale de *SimD* ;

Fin

Pour chaque item *i* (pas encore évalué par l'utilisateur 'u_i') ∈ C

Calculer IF.ICF et la note de prédiction où

$$IF = \frac{Nombre\ de\ fois\ où\ l'item\ est\ noté\ dans\ la\ communauté}{Nombre\ total\ d'évaluations\ d'items\ dans\ la\ communauté}$$

$$ICF = log\left(\frac{Nombre\ total\ de\ communautés}{nombre\ de\ communautés\ où\ l'item\ a\ été\ noté}\right)$$

Au sein de la communauté d'utilisateurs C,

Calculer le score de prédiction

$$\hat{R}_{u_i} = \bar{r}_{u_i} + \text{IF.ICF} \sum_{U_j \in \text{Kn}_{u_i}} \text{sim}_g(u_i, v_j)(r_{u_j} - \bar{r}_{u_j})$$

Fin

Trier les items dans l'ordre décroissant des scores de prédiction

Afin de générer une liste de recommandations.

Recommander les k premiers items à l'utilisateur.

3.4 Implémentation

1.4.1 Outils et environnement de développement

Cette partie est consacrée à la présentation des différents outils et langages utilisés afin de justifier nos choix techniques adoptés :

1.Langages de programmation

Nous avons utilisé le langage Python 3.10 (64-bit) pour l'implémentation de notre proposition.

Python est un langage de programmation puissant et facile à apprendre. Il dispose de structures de données de haut niveau et permet une approche simple mais efficace de la programmation orientée objet. Parce que sa syntaxe est élégante, que son typage est dynamique et qu'il est interprété, Python est un langage idéal pour l'écriture de scripts et le développement rapide d'applications dans de nombreux domaines et sur la plupart des plateformes. L'environnement python est riche en librairies.

Anaconda est un utilitaire pour Python offrant de nombreuse fonctionnalité. Il offre par exemple la possibilité d'installer des librairies et de les utiliser dans ses programmes.

Outre cette console, ANACONDA propose un logiciel servant à divers usages. Celui-ci permet principalement d'accéder aux autres logiciels installés dans la suite ANACONDA. Il est possible grâce à ce logiciel de contrôler les librairies utilisées par le programme, ainsi que de vérifier si celles-ci sont bien à jour. De plus, il offre à ses utilisateurs une grande documentation et un espace de partage communautaire.

La distribution libre Anaconda contient :

- Des bibliothèques d'apprentissage automatique comme TensorFlow, keras, sckit-learn et Theano.
- Des bibliothèques de visualisation comme Bokeh, Datashader, matplotlib et Holoviews.

Des éditeurs de texte : Jupyter Notebook, Pycharm, Spyder.

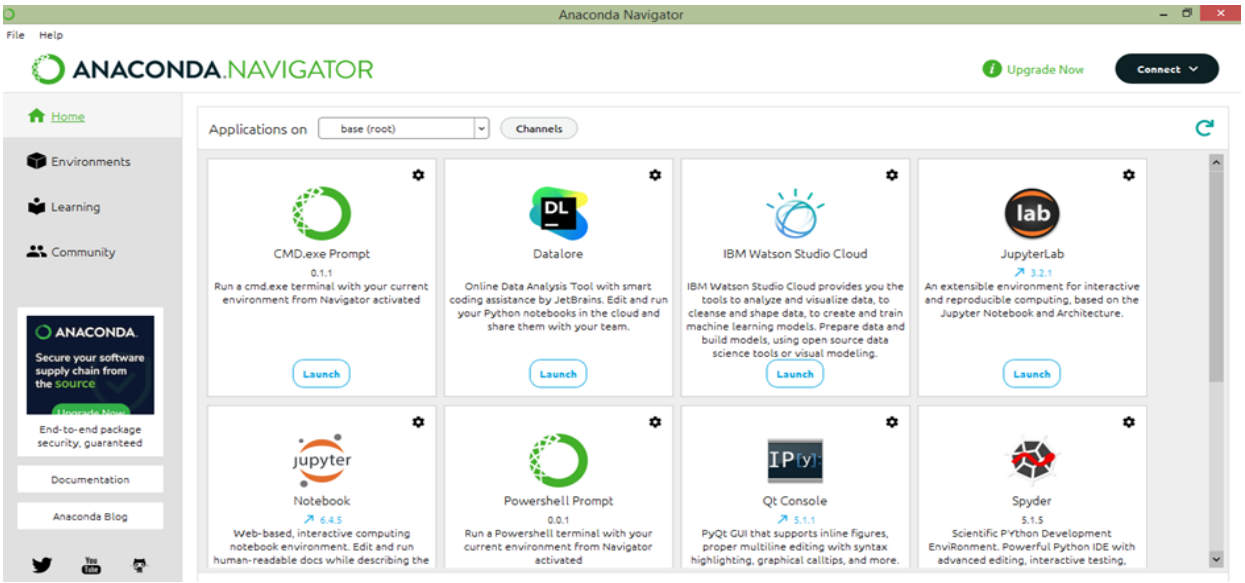


Figure 3.3 : Plateforme ANACONDA

Python est supporté par plusieurs éditeurs de code, nous allons citer les plus populaires :

IDLE C'est un environnement de développement intégré qui est simple et pratique pour les apprentis. IDLE est un éditeur de texte multifenêtre qui permet de créer, modifier et exécuter les scripts Python, mais son installation est un peu compliquée.

Spyder est un environnement de développement intégré comme IDLE créé pour la programmation scientifique. Il est codé en Python et pour Python ! Spyder est livré avec plusieurs fonctionnalités telles qu'un éditeur de code, une console interactive et finalement un explorateur de variables.

Jupyter notebook est une interface web développée pour la création de notebooks qui combinent du code et du texte narratif (titre, texte...) afin de permettre à l'utilisateur d'avoir un code réactif. Il permet aussi de faire des visualisations mathématiques grâces à ses différentes librairies

2. Les bibliothèques : Pour implémenter notre proposition nous aurons besoin de quelques bibliothèques, et pour les installer il faut écrire l'instruction (conda-install-c anaconda le_nom_de_la_bibliothéque).

Toutes les bibliothèques existantes sont données par le site : https://anaconda.org

Les packages utilisés sont :

Networkx : conda install -c anaconda networkx.

Pycairo : conda install -c anaconda pycairo.

Python-louvain conda install -c anaconda python-louvain.

Pyqt5 : conda install -c inso pyqt4.

Numpy : conda install -c anaconda Numpy.

Pandas : conda install -c anaconda Pandas.

3.5 Expérimentations

3.5.1 Jeu de données

Afin d'évaluer les performances de notre proposition, nous utilisons le jeu de données de MovieLens destiné à la recommandation des films. Ce dataset est très populaire et utilisé dans beaucoup d'études du domaine de filtrage collaboratif. Le site Web MovieLens est développé par le groupe de recherche GroupLens à l'université de Minnesota, États-Unis. On dispose sur ce site plusieurs jeux de données d'évaluations de films de tailles différentes. Nous utilisons dans cette expérimentation le dataset qui contient 100 000 évaluations de 1 à 5 étoiles, fournies par 943 utilisateurs sur 1682 films pendant une période allant de septembre 1997 à avril 1998. Ainsi, il existe 93,7% de données manquantes (matrice creuse). À partir de l'ensemble de données initial, cinq divisions distinctes pour l'apprentissage ainsi des données de test ont été générées (u1.base, u2.base, u3.base, u4.base, u5.base et u1.test, u2.test, u3.test, u4.test, u5.test). Pour chaque division de données, 80% de l'ensemble original a été inclus dans l'apprentissage et 20% d'entre eux ont été inclus dans les données de test.

3.5.2 Métriques d'évaluation

Lorsqu'un système de recommandation est développé, il est important d'être en mesure d'évaluer son fonctionnement et sa capacité à répondre aux objectifs qui lui ont été fixés. Parmi les mesures qui évaluent la qualité et l'efficacité des systèmes de recherche d'information et de recommandation, il existe :

- **Précision :** est définie par le ratio des items qui sont recommandés et qui sont pertinents sur le nombre total des recommandations (formule 3.9). Dans nos paramètres d'évaluation, nous considérons qu'un point d'intérêt est pertinent si l'utilisateur lui a attribué une note de 4 ou 5.

$$\boldsymbol{Precision} = \frac{|\{\boldsymbol{Recommandations\ Pertinentes}\} \cap \{\boldsymbol{recommandations\ émises}\}|}{|\{\boldsymbol{recommandations\ émises}\}|} \quad (3.9)$$

- **Erreur Absolue Moyenne (Mean Absolute Error MAE) :** est une métrique qui est régulièrement utilisée pour évaluer la précision d'une telle prédiction. En bref, elle représente la différence moyenne entre la valeur originale et celle prédite. MAE est calculé comme suit :

$$\boldsymbol{MAE} = \frac{\sum_{(u,i)\in \boldsymbol{Testset}} |\, r_{u,i} - pr_{u,i} \,|}{|\, \boldsymbol{Testset} \,|} \quad (3.10)$$

- **Erreur Quadratique Moyenne (Root Mean Squared Error RMSE) :**
  Peut être définie comme la moyenne des valeurs absolues des erreurs de prévision et est très appropriée lorsque le coût des erreurs de prévision est proportionnel à la taille absolue de l'erreur de prévision.

$$\boldsymbol{RMSE} = \sqrt{\frac{\sum_{(u,i)\in \mathrm{Testset}} (r_{u,i} - pr_{u,i})^2}{|\, \mathrm{Testset} \,|}} \quad (3.11)$$

Plus la MAE ou la RMSE est faible, meilleure est la précision du système.

3.6 Présentation de l'Application

Dans cette section, nous allons présenter quelques prises d'écrans de notre application

- **Lire Data :**

Au début de l'application, nous allons télécharger l'ensemble de données de MovieLens

```
Entrée [2]: data = pd.read_csv('user.csv',sep='|')

Entrée [3]: data
```

Out[3]:

	user_id	age	gender	occupation	zip_code
0	1	24	M	technician	85711
1	2	53	F	other	94043
2	3	23	M	writer	32067
3	4	24	M	technician	43537
4	5	33	F	other	15213
...	...	...	...	...	...
938	939	26	F	student	33319
939	940	32	M	administrator	02215
940	941	20	M	student	97229
941	942	48	F	librarian	78209
942	943	22	M	student	77841

Figure 3.4 : *l'ajout du data*

- **Code de préparation de la colonne âge**

```
Entrée [93]: #000 if the age of  < 15
             #010 if 15 ≤ the age of  ≤ 55
             #000 if the age of  > 55.
             data["Age"] = np.where((data["age"] < 15), "000",np.where((data["age"] > 55), "000","010"))
```

Figure 3.5 : *code de préparation de la colonne âge*

- **Code de préparation de la colonne gendre**

```
Entrée [95]: #1 0 if gender M
             #0 1 if gender F
             data_gender=pd.get_dummies(data['gender'])
```

Figure 3.6: *Code de préparation de la colonne gendre*

- **Code de préparation de la colonne occupation**

```
Entrée [96]: #1 If he takes his profession
             #0 If he does not take the profession
             new_data=pd.concat([data["Age"],data_gender,data_occupation],axis=1)
```

Figure 3.7 : *Code de préparation de la colonne occupation*

3.7 Résultats et analyses

Pour vérifier la faisabilité de notre approche proposée, nous comparons cette dernière avec quatre autres méthodes à savoir : DTD [163], DPSO [164], K-means et FCBU (Filtrage collaboratif basé utilisateur). Le nombre de cluster de K-means est attribué à trois cas et les individus du réseau sont divisés en 3, 6 et 10 clusters. Nous fixons la taille des k plus proche voisins de 10 à 50. Les deux tableaux 3.4 et 3.5 montrent les résultats expérimentaux de MAE et RMSE de notre approche et les quatre méthodes.

Nombre de voisins / Méthodes	10	20	30	40	50

DTD	0.7802	0.7893	0.7854	0.7848	0.7840
DPSO	08366	08366	08366	08366	08366
K-means	0.7595	0.7588	0.7592	0.7591	0.7585
SRBU	0.7755	0.7621	0.7605	0.7501	0.7592
Approche proposée	**0.7552**	**0.7432**	**0.7455**	**0.7417**	**0.7477**

Tableau 3.4*: Comparaison des résultats de différentes méthodes en termes de MAE*

Nombre de voisins / Méthodes	10	20	30	40	50
DTD	1.002	1.012	1.025	1.015	1.007
DPSO	1.043	1.043	1.043	1.043	1.043
K-means	0.9701	0.9655	0.9661	0.9668	0.9684
SRBU	0.9822	0.9695	0.9683	0.9671	0.9691
Approche proposée	**0.9625**	**0.9529**	**0.9542**	**0.9535**	**0.9533**

Tableau 3.5*: Comparaison des résultats de différentes méthodes en termes de RMSE*

Les figures 3.8 et 3.9 présentent les valeurs MAE et RMSE selon le nombre de voisins. Nous pouvons observer que notre méthode a plus d'efficacités que les autres algorithmes. Le résultat de DPSO est que chaque groupe ne contient qu'un seul utilisateur ; ainsi, la précision de la prédiction est indépendante du nombre de voisins. Les valeurs moyennes de MAE de DTD, DPSO, K-means, FCBU et l'approche proposée sont respectivement : 0,7847, 0.8366, 0.7590, 0.7614 et 0.7466. De plus, les valeurs moyennes de RMSE pour DTD, DPSO, K-means, FCBU et l'approche proposée sont de : 1.0122, 1.0430, 0.9673, 0.9712 et 0.9552. En outre, notre approche proposée est

plus efficace que l'algorithme K-means, du fait que celle-ci ne nécessite pas le nombre prédéfini de clusters.

Comme le montre la figure 3.8, lorsque k > = 20, la valeur de MAE de notre approche diminue évidemment et tend à être stable, ce qui est nettement meilleur que les quatre autres méthodes. La valeur MAE de notre algorithme atteint la valeur minimale à k = 40, ce qui signifie que la performance de l'algorithme est meilleure lorsque k plus proche voisin est 40.

La figure 3.9 illustre qu'à k= 40, notre approche a une augmentation de 1,36% en RMSE contre FCBU, de 1,33% contre K-means, de 8,95% contre DPSO et 6.15% par rapport à DTD.

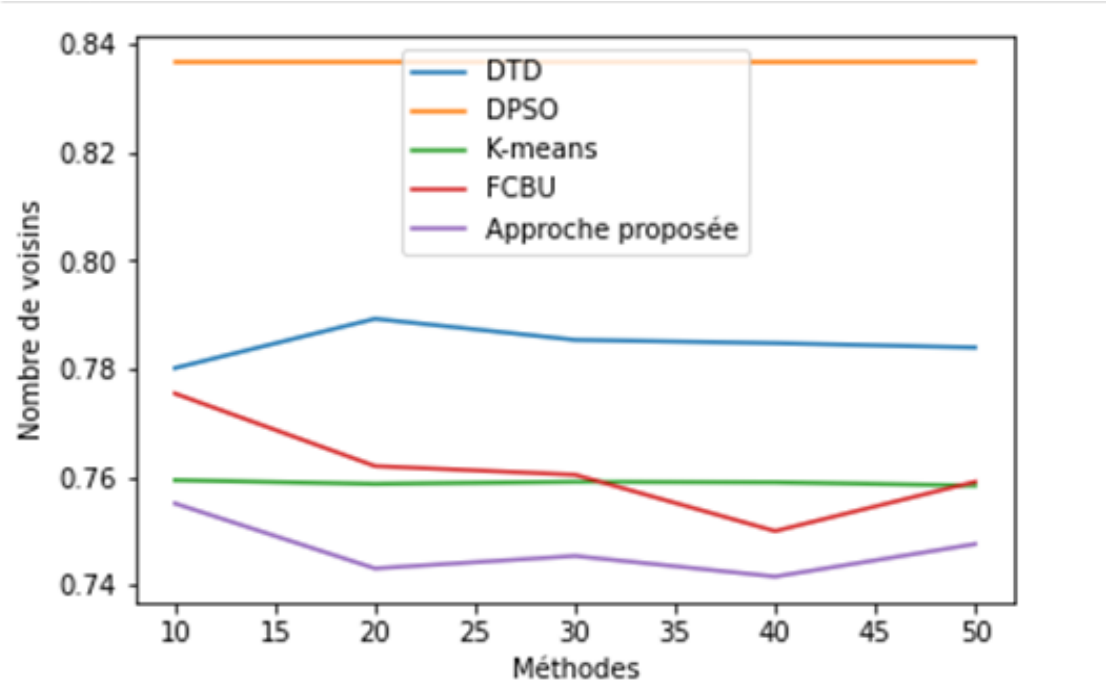


Figure 3.8: *Comparaison des résultats des différentes méthodes en termes de MAE*

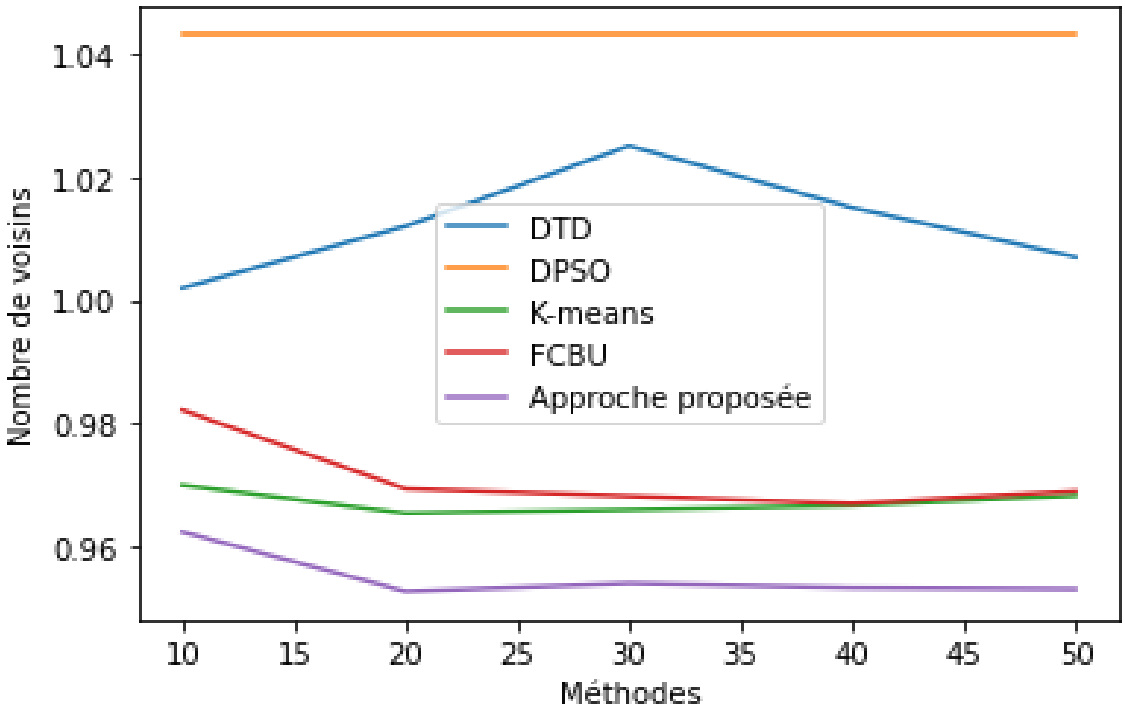


Figure 3.9 : *Comparaison des résultats des différentes méthodes en termes de RMSE*

3.8 Conclusion

Dans le cadre de ce chapitre, nous avons présenté notre proposition, les métriques d'évaluation, les outils de développement utilisés, ainsi que les résultats obtenus. Pour terminer, nous avons clôturé ce chapitre par une présentation de quelques prises d'écrans de notre application.

Conclusion Générale

Conclusion générale

Les systèmes de recommandation ont reconnu une grande importance dans les différentes applications web pour aider les utilisateurs à sélectionner les produits et les services qui correspondent à leur besoin. Actuellement, le filtrage collaboratif (FC) est considéré comme la technique de recommandation la plus réussite. Ces dernières années, plusieurs tentatives ont été entrepris pour incorporer des méthodes de détection de communautés au sein des systèmes de recommandation, et ce afin de rendre le processus de génération de recommandations plus précis en termes d'évaluation ou de prédiction de préférence et efficace en termes de complexité de calcul. Les méthodes de détection de communauté aident à déterminer des groupes similaires d'items ou de personnes. L'idée des deux techniques est identique pour trouver des connexions existant dans des groupes d'items ou de personnes. Au sein de ce travail nous avons proposé une approche de recommandation qui utilise cette notion afin de combiner la détection de communauté avec la technique FC. La méthode de détection de communautés est employée pour nous aider à identifier les personnes similaires à l'utilisateur cible qui sert finalement le voisinage potentiel de l'utilisateur. De plus, la fréquence d'item (IF-ICF) aide à déterminer l'importance des items existant dans la communauté de l'utilisateur. Les items avec des valeurs IF-ICF élevées le placent plus haut dans la liste de recommandations générée pour l'utilisateur. Dans chaque communauté, une mesure de corrélation d'utilisateur est adoptée pour trouver les voisins similaires les plus élevés avec l'utilisateur cible. Enfin, le filtrage collaboratif basé sur l'utilisateur est adopté dans chaque groupe. Des expériences comparatives démontrent que notre proposition obtient de meilleures performances de recommandation que les autres algorithmes comparés. Mais le travail reste encore à améliorer, nous souhaitons intégrer d'autres informations de réseau des items et d'autres caractéristiques des utilisateurs tels que les conditions d'utilisation, l'emplacement géographique et l'heure des évaluations des utilisateurs et des items dans le système de recommandation pour une amélioration supplémentaire des performances.

Bibliographie & Référence

1. Bastin J. Etude des systèmes de recommandations et mise en pratique des algorithmes. n.d. :86.
2. Sharma C, Bedi P. CCFRS – Community based Collaborative Filtering Recommender System. J Intell Fuzzy Syst 2017 ;32 :2987–95. https://doi.org/10.3233/JIFS-169242.
3. Ferreira D, Silva S, Abelha A, Machado J. Recommendation System Using Autoencoders. Appl Sci 2020 ;10 :5510. https://doi.org/10.3390/app10165510.
4. Ko H, Lee S, Park Y, Choi A. A Survey of Recommendation Systems: Recommendation Models, Techniques, and Application Fields. Electronics 2022 ;11 : 141. https://doi.org/10.3390/electronics11010141
5. Barragáns-Martínez, A.B.; Costa-Montenegro, E.; Burguillo, J.C.; Rey-López, M.; Mikic-Fonte, F.A.; Peleteiro, A. A Hybrid Content-Based and Item-Based Collaborative Filtering Approach to Recommend TV Programs Enhanced with Singular Value Decomposition. Inf. Sci. 2010, 180, 4290–4311. [CrossRef]
6. Gomez-Uribe, C.A.; Hunt, N. The Netflix Recommender System: Algorithms, Business Value, and Innovation. ACM Trans. Manage. Inf. Syst. 2016, 6, 1–19. [CrossRef]
7. Koren, Y. Factor in the Neighbors: Scalable and Accurate Collaborative Filtering. ACM Trans. Knowl. Discov. Data 2010, 4, 1–24. [CrossRef]
8. Bell, R.M.; Koren, Y. Lessons from the Netflix Prize Challenge. SIGKDD Explor. Newsl. 2007, 9, 75–79. [CrossRef]
9. McFee, B.; Barrington, L.; Lanckriet, G. Learning Content Similarity for Music Recommendation. IEEE Trans. Audio Speech Lang. Process. 2012, 20, 2207–2218. [CrossRef]
10. Odi´c, A.; Tkal˘ci˘c, M.; Tasi˘c, J.F.; Košir, A. Predicting and Detecting the Relevant Contextual Information in a Movie-Recommender System. Interact. Comput. 2013, 25, 74–90. [CrossRef]
11. Barragáns-Martínez, A.B.; Costa-Montenegro, E.; Burguillo, J.C.; Rey-López, M.; Mikic-Fonte, F.A.; Peleteiro, A. A Hybrid Content-Based and Item-Based Collaborative Filtering Approach to Recommend TV Programs Enhanced with Singular Value Decomposition. Inf. Sci. 2010, 180, 4290–4311. [CrossRef]
12. He, J.; Chu, W.W. A Social Network-Based Recommender System (SNRS). In Data Mining for Social Network Data; Memon, N., Xu, J.J., Hicks, D.L., Chen, H., Eds.; Annals of Information Systems; Springer: Boston, MA, USA, 2010; Volume 12, pp. 47–74, ISBN 978-1-4419-6286-7
13. Tsur, O.; Rappoport, A. What's in a Hashtag? Content Based Prediction of the Spread of Ideas in Microblogging Communities. In Proceedings of the Fifth ACM International

Conference on Web Search and Data Mining—WSDM ’12, New York, NY, USA, 8–12 February 2012; p. 643

14. Kesorn, K.; Juraphanthong, W.; Salaiwarakul, A. Personalized Attraction Recommendation System for Tourists Through Check-In Data. IEEE Access 2017, 5, 26703–26721. [CrossRef]

15. Sun, Y.; Fan, H.; Bakillah, M.; Zipf, A. Road-Based Travel Recommendation Using Geo-Tagged Images. Comput. Environ. Urban Syst. 2015, 53, 110–122. [CrossRef]

16. Galhotra, B.; Dewan, A. Impact of COVID-19 on Digital Platforms and Change in E-Commerce Shopping Trends. In Proceedings of the 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Coimbatore, India, 7–9 October 2020; pp. 861–866

17. Zhang, X.; Liu, H.; Chen, X.; Zhong, J.; Wang, D. A Novel Hybrid Deep Recommendation System to Differentiate User’s Preference and Item’s Attractiveness. Inf. Sci. 2020, 519, 306–316. [CrossRef]

18. Lu, Y.; Zhao, L.; Wang, B. From Virtual Community Members to C2C E-Commerce Buyers: Trust in Virtual Communities and Its Effect on Consumers’ Purchase Intention. Electron. Commer. Res. Appl. 2010, 9, 346–360. [CrossRef]

19. Sulikowski, P.; Zdziebko, T.; Hussain, O.; Wilbik, A. Fuzzy Approach to Purchase Intent Modeling Based on User Tracking for E-Commerce Recommenders. In Proceedings of the 2021 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Luxembourg, 11–14 July 2021; pp. 1–8.

20. Jiang, L.; Cheng, Y.; Yang, L.; Li, J.; Yan, H.; Wang, X. A trust-based collaborative filtering algorithm for E-commerce recommendation system. J. Ambient. Intell. Humaniz. Comput. 2019, 10, 3023–3034. [CrossRef]

21. Sarwar, B.; Karypis, G.; Konstan, J.; Reidl, J. Item-Based Collaborative Filtering Recommendation Algorithms. In Proceedings of the Tenth International Conference on World Wide Web—WWW ’01, Hong Kong, China, 1–5 May 2001; pp. 285–295

22. Gong, S. A Collaborative Filtering Recommendation Algorithm Based on User Clustering and Item Clustering. J. Softw. 2010, 5, 745–752. [CrossRef]

23. Hwangbo, H.; Kim, Y.S.; Cha, K.J. Recommendation System Development for Fashion Retail E-Commerce. Electron. Commer. Res. Appl. 2018, 28, 94–101. [CrossRef]

24. Tröster, G. The Agenda of Wearable Healthcare. Yearb. Med. Inf. 2005, 14, 125–138. [CrossRef]

25. Apple. Use Emergency SOS on Your Apple Watch. 2021. Available online: https://support.apple.com/en-us/HT206983 (accessed on 27 April 2021).

26. Mishra, T.; Wang, M.; Metwally, A.A.; Bogu, G.K.; Brooks, A.W.; Bahmani, A.; Alavi, A.; Celli, A.; Higgs, E.; Dagan-Rosenfeld, O.; et al. Pre-Symptomatic Detection of COVID-19 from Smartwatch Data. Nat. Biomed. Eng. 2020, 4, 1208–1220. [CrossRef] [PubMed]

27. Zheng, J.W.; Zhang, Z.B.; Wu, T.H.; Zhang, Y. A Wearable Mobihealth Care System Supporting Real-Time Diagnosis and Alarm. Med. Bio. Eng. Comput. 2007, 45, 877–885. [CrossRef] [PubMed]

28. Raghupathi, W.; Raghupathi, V. Big Data Analytics in Healthcare: Promise and Potential. Health Inf. Sci. Syst. 2014, 2, 3. [CrossRef] [PubMed]
29. Rehman, A.; Naz, S.; Razzak, I. Leveraging Big Data Analytics in Healthcare Enhancement: Trends, Challenges and Opportunities. Multimed. Syst. 2021, 1, 1–33. [CrossRef]
30. Tikhomirov, V.; Dneprovskaya, N.; Yankovskaya, E. Three Dimensions of Smart Education. In Smart Education and Smart e-Learning; Uskov, V.L., Howlett, R.J., Jain, L.C., Eds.; Smart Innovation, Systems and Technologies; Springer International Publishing: Cham, Switzerland, 2015; Volume 41, pp. 47–56, ISBN 978-3-319-19874-3.
31. Lin, J.; Pu, H.; Li, Y.; Lian, J. Intelligent Recommendation System for Course Selection in Smart Education. Procedia Comput. Sci. 2018, 129, 449–453. [CrossRef]
32. Zhu, Z.-T.; Yu, M.-H.; Riezebos, P. A Research Framework of Smart Education. Smart Learn. Environ. 2016, 3, 4. [CrossRef]
33. Wan, S.; Niu, Z. An E-Learning Recommendation Approach Based on the Self-Organization of Learning Resource. Knowl. -Based Syst. 2018, 160, 71–87. [CrossRef]
34. Shu, J.; Shen, X.; Liu, H.; Yi, B.; Zhang, Z. A Content-Based Recommendation Algorithm for Learning Resources. Multimed. Syst. 2018, 24, 163–173. [CrossRef]
35. Chen, Y.; Li, X.; Liu, J.; Ying, Z. Recommendation System for Adaptive Learning. Appl. Psychol. Meas. 2018, 42, 24–41. [CrossRef]
36. Tarus, J.K.; Niu, Z.; Yousif, A. A Hybrid Knowledge-Based Recommender System for e-Learning Based on Ontology and Sequential Pattern Mining. Future Gener. Comput. Syst. 2017, 72, 37–48. [CrossRef]
37. Klašnja-Mili´cevi´c, A.; Vesin, B.; Ivanovi´c, M.; Budimac, Z. E-Learning Personalization Based on Hybrid Recommendation Strategy and Learning Style Identification. Comput. Educ. 2011, 56, 885–899. [CrossRef]
38. Dwivedi, P.; Bharadwaj, K.K. E-Learning Recommender System for a Group of Learners Based on the Unified Learner Profile Approach. Expert Syst. 2015, 32, 264–276. [CrossRef]
39. Wu, D.; Lu, J.; Zhang, G. A Fuzzy Tree Matching-Based Personalized E-Learning Recommender System. IEEE Trans. Fuzzy Syst. 2015, 23, 2412–2426. [CrossRef]
40. Serrano-Guerrero, J.; Herrera-Viedma, E.; Olivas, J.A.; Cerezo, A.; Romero, F.P. A Google Wave-Based Fuzzy Recommender System to Disseminate Information in University Digital Libraries 2.0. Inf. Sci. 2011, 181, 1503–1516. [CrossRef]
41. Tejeda-Lorente, Á.; Porcel, C.; Peis, E.; Sanz, R.; Herrera-Viedma, E. A Quality Based Recommender System to Disseminate Information in a University Digital Library. Inf. Sci. 2014, 261, 52–69. [CrossRef]
42. He, Q.; Pei, J.; Kifer, D.; Mitra, P.; Giles, L. Context-aware citation recommendation. In Proceedings of the 19th International Conference on World Wide Web—WWW ’10, Raleigh, NC, USA, 26–30 April 2010. [CrossRef]
43. Wang, D.; Liang, Y.; Xu, D.; Feng, X.; Guan, R. A Content-Based Recommender System for Computer Science Publications. Knowl. -Based Syst. 2018, 157, 1–9. [CrossRef]

44. Park, N.; Roman, R.; Lee, S.; Chung, J.E. User Acceptance of a Digital Library System in Developing Countries: An Application of the Technology Acceptance Model. Int. J. Inf. Manag. 2009, 29, 196–209. [CrossRef]
45. Jeong, H. An Investigation of User Perceptions and Behavioral Intentions towards the E-Library. Libr. Collect. Acquis. Tech. Serv. 2011, 35, 45–60. [CrossRef]
46. Achakulvisut, T.; Acuna, D.E.; Ruangrong, T.; Kording, K. Science Concierge: A Fast Content-Based Recommendation System for Scientific Publications. PLoS ONE 2016, 11, e0158423. [CrossRef]
47. F. Ricci, L. Rokach, et B. Shapira, Introduction to recommender systems handbook. Springer, 2011.
48. R. Mihalcea et A. Csomai, « Wikify! linking documents to encyclopedic knowledge », in Proceedings of the sixteenth ACM conference on Conference on information and knowledge management, 2007, p. 233–242.
49. Ò. Celma, « Foafing the music: Bridging the semantic gap in music recommendation », in The Semantic Web-ISWC 2006, Springer, 2006, p. 927–934.
50. B. Sheth et P. Maes, « Evolving agents for personalized information filtering », in Artificial Intelligence for Applications, 1993. Proceedings., Ninth Conference on, 1993, p. 345–352.
51. Picot-Clémente R. Une architecture générique de Systèmes de recommandation de combinaison d'items: application au domaine du tourisme n.d.:146.
52. http://dspace.univ-tlemcen.dz/bitstream/112/8177/1/Systeme-de-recommandation-desservices-web-semantiques.pdf
53. M. Baldauf, S. Dustdar, et F. Rosenberg, « A survey on context-aware systems », Int. J. Ad Hoc Ubiquitous Comput., vol. 2, no 4, p. 263–277, 2007.
54. Quba RCA. On enhancing recommender systems by utilizing general social networks combined with users goals and contextual awareness n.d.:128.
55. D. Mladenic, « Text-learning and related intelligent agents: a survey », IEEE Intell. Syst., vol. 14, no 4, p. 44–54, 1999.
56. F. Ricci, L. Rokach, et B. Shapira, Introduction to recommender systems handbook. Springer, 2011.
57. Collaborative Filtering In Recommender Systems: Learn All You Need To Know | 2021. https://www.iteratorshq.com/blog/collaborative-filtering-in-recommender-systems/ (accessed September 20, 2022).
58. Goldberg, D.; Nichols, D.; Oki, B.M.; Terry, D. Using Collaborative Filtering to Weave an Information Tapestry. Commun. ACM 1992, 35, 61–70. [CrossRef]
59. Resnick, P.; Iacovou, N.; Suchak, M.; Bergstrom, P.; Riedl, J. GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work, Chapel Hill, NC, USA, 22–26 October 1994; pp. 175–186.
60. Im, I.; Hars, A. Does a One-Size Recommendation System Fit All? The Effectiveness of Collaborative Filtering Based Recommendation Systems across Different Domains and Search Modes. ACM Trans. Inf. Syst. 2007, 26, 4. [CrossRef]

61. Park, S.-H.; Han, S.P. Empirical Analysis of the Impact of Product Diversity on Long-Term Performance of Recommender Systems. In Proceedings of the 14th Annual International Conference on Electronic Commerce—ICEC ’12, Singapore, 7–8 August 2012; pp. 280–281.

62. Systèmes de recommandation – Olcya n.d. https://www.olcya.com/2018/09/22/2018-09-22-systemes-de-recommandation/ (accessed September 20, 2022).

63. Christian Desrosiers and George Karypis. A survey of collaborative filtering techniques. adv. In Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors, Recommender Systems Handbook, pages 107–144., 2011. 35,36,44

64. Filtrage collaboratif article par article – Acervo Lima n.d. https://fr.acervolima.com/filtrage-collaboratif-article-a-article/ (accessed September 21, 2022).

65. Hasnaa C. La recommandation de publication dans un réseau social n.d. :76.

66. Bisson, 2000

67. Quba RCA. On enhancing recommender systems by utilizing general social networks combined with users’ goals and contextual awareness n.d.:130.

68. Tadlaoui M. Système de recommandation de ressources pédagogiques fondé sur les liens sociaux : Formalisation et évaluation n.d. :132

69. Paul Jaccard, « *Distribution de la flore alpine dans le bassin des Dranses et dans quelques régions voisines* », Bulletin de la Société vaudoise des sciences naturelles, vol. 37, 1901, p. 241-272.

70. Zaïer Z. Modèle multi-agents pour le filtrage collaboratif de l’information n.d. :158.

71. BURKE: Hybrid Web Recommender Systems. pages 377–408, 2007.

72. Burke R., (2002): Hybrid recommender systems: Survey and experiments. In User Modeling and User Adapted Interaction, 12(4), pp. 331-370.

73. Barnes, J.A. (1954) Class and Committees in a Norwegian Island Parish. Human Relations, 7, 39-58. http://dx.doi.org/10.1177/001872675400700102

74. Girvan M, Newman M. Community structure in social and biological networks. Proceedings of the national academy of sciences 2002, 99 (12):7821-7826. doi:10.1073/pnas.122653799.

75. Newman, M.E.J. (2001) Scientific Collaboration Networks (I): Network Construction and Fundamental Results. Physical Review, 64, Article ID: 016131.

76. L. Traud, E. D. Kelsic, P. J. Mucha, et M. A. Porter. Community structure in online ccollegiate social networks. 2011. URL http://arxiv.org/abs/ 0809.0690v1. (Cité page 17.)

77. Joseph, D. L., & Newman, D. A. (2010). Emotional intelligence: An integrative meta-analysis and cascading model. Journal of Applied Psychology, 95(1), 54–78. https://doi.org/10.1037/a0017286 ,
Anil et al., 2015 ; Geffre, Deckro et Knighton, 2009

78. Anil, A., Kumar, D., Sharma, S., Singha, R., Sarmah, R., Bhattacharya, N., & Singh, S. R. (2015). Link prediction using social network analysis over heterogeneous terrorist network.

In Proceedings of the international conference on smart city socialcom sustaincom (p. 267-272).

79. Geffre, J., Deckro, R. F., & Knighton, S. A. (2009). Determining critical members of layered operational terrorist networks. The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology, 6 (2), 97-109.

80. Jacob Moreno (1933).

81. Chouchani N. Une approche de détection des communautés d'intérêt dans les réseaux sociaux : application à la génération d'IHM personnalisées n.d. :152

82. Brachotte, Gilles, et Alexander Frame. L'usage de Twitter par les candidats. Perspectives internationales lors des élections du Parlement européen en mai 2014. EMS Editions, 2018

83. McPherson M, Lovin LS, Cook JM. Birds of a feather: Homophily in Social Networks. Annual review of sociology 2001:415 - 444. doi: 10.1146/annurev.soc.27.1.415.

84. Luce RD, Perry AD. A method of matrix analysis of group structure. Psychometrika 1949, 14 (2):95 –116. doi:10.1007/BF02289146.

85. Watts DJ, Strogatz SH. Collective dynamics of 'small-world'networks. Nature 1998, 393 (6684):440-442. doi:10.1038/30918.

86. Fortunato S. Community detection in graphs. Physics Reports 2010, 486 (3):75-174. doi: 10.1016/j.physrep.2009.11.002.

87. Boutin F. Filtrage, partitionnement et visualisation multi-échelles de graphes d'interactions à partir d'un focus n.d. :166.

88. [Saporta 1900] [Lebart et al. 200], [kohonene 1982]

89. Coscia M, Giannotti F, Pedreschi D. A classification for community discovery methods in complex networks. Statistical Analysis and Data Mining: The ASA Data Science Journal 2011, 4 (5):512- 546. doi:10.1002/sam.10133.

90. Fortunato S, Castellano C. Community structure in graphs. In: Computational Complexity: Springer; 2012, 490-512.doi:10.1007/978-1-4614-1800-9_33.

91. Plantié M, Crampes M. Survey on social community detection. In: Social Media Retrieval: Springer; 2013, 65-85.doi:10.1007/978-1-4471-4555-4_4.

92. Kernighan BW, Lin S. An efficient heuristic procedure for partitioning graphs. Bell system technical journal 1970, 49 (2):291-307. doi:10.1002/j.1538-7305. 1970.tb01770. x.

93. Newman M. Community detection and graph partitioning. EPL (Europhysics Letters) 2013, 103 (2):28003. doi:10.1209/0295-5075/103/28003.

94. Karrer B, Newman M. Stochastic blockmodels and community structure in networks. Physical Review E 2011, 83 (1):016107. doi:10.1103/PhysRevE.83.016107.

95. Fiedler M. Algebraic connectivity of graphs. Czechoslovak mathematical journal 1973, 23 (2):298- 305.

96. Pothen A, Simon HD, Liou K-P. Partitioning sparse matrices with eigenvectors of graphs. SIAM journal on matrix analysis and applications 1990, 11 (3):430-452. doi:10.1137/0611030.

97. Newman M, Girvan M. Finding and evaluating community structure in networks. Physical review E 2004, 69 (2):026113. doi:10.1103/PhysRevE.69.026113.

98. Rattigan MJ, Maier M, Jensen D. Graph clustering with network structure indices. In: Proceedings of the 24th International conference on Machine learning (ICML): ACM; 2007: 783- 790.doi:10.1145/1273496.1273595.

99. Chen J, Yuan B. Detecting functional modules in the yeast protein-protein interaction network. Bioinformatics 2006, 22 (18):2283-2290. doi:10.1093/bioinformatics/btl370.

100. Holme P, Huss M, Jeong H. Subnetwork hierarchies of biochemical pathways. Bioinformatics 2003, 19 (4):532-538.

101. Pinney JW, Westhead DR. Betweenness-based decomposition methods for social and biological networks. Interdisciplinary Statistics and Bioinformatics 2006:87-90.

102. Gregory S. An algorithm to find overlapping community structure in networks. In: Knowledge discovery in databases: PKDD Springer; 2007, 91-102.doi:10.1007/978-3-540-74976-9_12.

103. Guimera R, Danon L, Diaz-Guilera A, Giralt F, Arenas A. Self-similar community structure in a network of human interactions. Physical review E 2003, 68 (6):065103. doi:10.1103/PhysRevE.68.065103.

104. Arenas A, Danon L, Diaz-Guilera A, Gleiser PM, Guimera R. Community analysis in social networks. The European Physical Journal B-Condensed Matter and Complex Systems 2004, 38 (2):373-380. doi:10.1140/epjb/e2004-00130-1.

105. Tyler JR, Wilkinson DM, Huberman BA. E-mail as spectroscopy: Automated discovery of community structure within organizations. The Information Society 2005, 21 (2):143-153.

106. Radicchi F, Castellano C, Cecconi F, Loreto V, Parisi D. Defining and identifying communities in networks. Proceedings of the National Academy of Sciences of the United States of America 2004, 101 (9):2658-2663. doi:10.1073/pnas.0400054101.

107. Moon S, Lee J-G, Kang M, Choy M, Lee J-w. Parallel community detection on large graphs with MapReduce and GraphChi. Data & Knowledge Engineering 2015, Article in Press. doi: 10.1016/j.datak.2015.05.001.

108. Newman M. Fast algorithm for detecting community structure in networks. Physical review E 2004, 69 (6):066133. doi:10.1103/PhysRevE.69.066133.

109. Chen Y, Huang C, Zhai K. Scalable community detection algorithm with MapReduce. Commun. ACM 2009, 53:359-366. doi:10.1147/JRD.2013.2251982.

110. Newman M. Analysis of weighted networks. Physical Review E 2004, 70 (5):056131. doi:10.1103/PhysRevE.70.056131.

111. Newman M. Modularity and community structure in networks. Proceedings of the National Academy of Sciences 2006, 103 (23):8577-8582. doi:10.1073/pnas.0601602103.

112. Clauset A, Newman ME, Moore C. Finding community structure in very large networks. Physical review E 2004, 70 (6):066111. doi:10.1103/PhysRevE.70.066111.

113. Blondel VD, Guillaume JL, Lambiotte R, Lefebvre E. Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment 2008. doi:10.1088/1742- 5468/2008/10/P10008.

114. Guimera R, Sales-Pardo M, Amaral LAN. Modularity from fluctuations in random graphs and complex networks. Physical Review E 2004, 70 (2):025101. doi:10.1103/PhysRevE.70.025101

115. Zhou Z, Wang W, Wang L. Community Detection Based on an Improved Modularity. Pattern Recognition 2012:638-645. doi:10.1007/978-3-642-33506-8_78.

116. Duch J, Arenas A. Community detection in complex networks using extremal optimization. Physical review E 2005, 72 (2):027104. doi:10.1103/PhysRevE.72.027104.

117. Ye Z, Hu S, Yu J. Adaptive clustering algorithm for community detection in complex networks. Physical Review E 2008, 78 (4):046115. doi:10.1103/PhysRevE.78.046115.

118. Wahl S, Sheppard J. Hierarchical Fuzzy Spectral Clustering in Social Networks Using Spectral Characterization. In: The Twenty-Eighth International Flairs Conference; 2015: 305-310

119. Falkowski T, Barth A, Spiliopoulou M. DENGRAPH: A density-based community detection algorithm. In: IEEE/WIC/ACM International Conference on Web Intelligence (WI); 2007: 112- 115.doi:10.1109/WI.2007.74.

120. Dongen SV. Graph Clustering by Flow Simulation, PhD thesis, University of Utrecht. 2000.

121. Nikolaev AG, Razib R, Kucheriya A. On efficient use of entropy centrality for social network analysis and community detection. Social Networks 2015, 40:154-162. doi: 10.1016/j.socnet.2014.10.002.

122. Steinhaeuser K, Chawla NV. Identifying and evaluating community structure in complex networks. Pattern Recognition Letters 2010, 31 (5):413-421. doi: 10.1016/j.patrec.2009.11.001.

123. Bedi P, Sharma C. Community detection in social networks: Community detection in social networks. Wiley Interdiscip Rev Data Min Knowl Discov 2016; 6:115–35. https://doi.org/10.1002/widm.1178

124. Raghavan UN, Albert R, Kumara S. Near linear time algorithm to detect community structures in large-scale networks. Physical Review E 2007, 76 (3):036106. doi:10.1103/PhysRevE.76.036106.

125. Xie J, Szymanski BK. Towards linear time overlapping community detection in social networks. In: Advances in Knowledge Discovery and Data Mining: Springer; 2012, 25-36

126. Hu W. Finding Statistically Significant Communities in Networks with Weighted Label Propagation. Social Networking 2013, 2:138-146 doi:10.4236/sn.2013.23012.

127. Pizzuti C. GA-Net: A genetic algorithm for community detection in social networks. In: Parallel Problem Solving from Nature–PPSN X: Springer; 2008, 1081-1090.doi:10.1007/978-3-540- 87700-4_107.

128. Pizzuti C. A multiobjective genetic algorithm to find communities in complex networks. IEEE Transactions on Evolutionary Computation 2012, 16 (3):418-430. doi:10.1109/TEVC.2011.2161090.

129. Hafez AI, Ghali NI, Hassanien AE, Fahmy AA. Genetic algorithms for community detection in social networks. In: 12th International Conference on Intelligent Systems Design and Applications (ISDA): IEEE; 2012: 460 465. doi:10. 1109/ISDA. 2012. 6416582.

130. Mazur P, Zmarzlowski K, Orlowski AJ. A Genetic Algorithms Approach to Community Detection. Acta Physica Polonica Series A- General Physics 2010, 117(4).

131. Liu X, Li D, Wang S, Tao Z. Effective algorithm for detecting community structure in complex networks based on GA and clustering. In: International Conference on Computational Science (ICCS 07): Springer; 2007: 657-664.doi:10.1007/978-3-540-72586-2_95.

132. Tasgin M, Herdagdelen A, Bingol H. Community detection in complex networks using genetic algorithms. arXiv preprint arXiv: 0711.0491 2007.

133. Zadeh PM, Kobti Z. A Multi-Population Cultural Algorithm for Community Detection in Social Networks. Procedia Computer Science 2015, 52:342-349. doi: 10.1016/j.procs.2015.05.105.

134. Xie J, Kelley S, Szymanski BK. Overlapping community detection in networks: The state-of-theart and comparative study. ACM Computing Surveys (csur) 2013, 45 (4):1-35. doi:10.1145/2501654.2501657.

135. Palla G, Derenyi I, Farhas I, Vicsek T. Uncovering the overlapping community structure of complex networks in nature and society. Nature 2005, 435:814–818. doi:10.1038/nature03607.

136. Lancichinetti A, Fortunato S, Kertész J. Detecting the overlapping and hierarchical community structure in complex networks. New Journal of Physics 2009, 11 (3):033015. doi:10.1088/1367- 2630/11/3/033015.

137. Du N, Wu B, Pei X, Wang B, Xu L. Community detection in large-scale social networks. In: Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis: ACM;2007: 16-25.doi:10. 1145/1348549. 1348552.

138. Shen H, Cheng X, Cai K, Hu M-B. Detect overlapping and hierarchical community structure in networks. Physica A: Statistical Mechanics and its Applications 2009, 388 (8):1706-1712. doi: 10.1016/j.physa.2008.12.021.

139. Evans T, Lambiotte R. Line graphs, link partitions, and overlapping communities. Physical Review E 2009, 80 (1). doi:10.1103/PhysRevE.80.016105.

140. Evans T, Lambiotte R. Line graphs of weighted networks for overlapping communities. The European Physical Journal B 2010, 77 (2):265-272. doi:10.1140/epjb/e2010-00261-8.
Evans TS. Clique Graph and Overlapping Communities. Journal of Statistical Mechanics: Theory and Experiment 2010, 12. doi:10.1088/1742-5468/2010/12/P12037.

141. Lee C, Reid F, McDaid A, Hurley N. Detecting highly overlapping community structure by greedy clique expansion. arXiv preprint arXiv:1002.1827 2010.

142. Gregory S. A fast algorithm to find overlapping communities in networks. In: ECML PKDD: European Conference on Machine Learning and Knowledge Discovery in Databases - Part I: Springer; 2008: 408-423

143. Gregory S. Finding overlapping communities using disjoint community detection algorithms. In: Complex networks: Springer; 2009, 47-61.doi:10.1007/978-3-642-01206-8_5.

144. Everett MG, Borgatti SP. Analyzing Clique Overlap. Connnections 1998, 21 (1):49-61.

145. Adamcsek Bz, Palla G, Farkas IsJ, Dere'nyi I, Vicsek Ts. CFinder: locating cliques and overlapping modules in biological networks. Bioinformatics 2006, 22 (8):1021-1023.

146. Rachid MAO. Approche d'optimisation pour le suivi de l'évolution de la structure communautaire des réseaux dynamiques n.d. :112.

147. Mohamed Moctar A, Sarr I. Détection de communautés statiques et dynamiques. Revue Intelligence Artificielle 2016 ;30 :469–96. https://doi.org/10.3166/ria.30.469-496.

148. R. CAZABET, Détection des communautés dynamiques dans des réseaux temporels, THÈSE DE DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE,2013.

149. T. Aynaud, Détection de communautés dans les réseaux dynamiques, THÈSE DE DOCTORAT DE L'UNIVERSITÉ PIERRE ET MARIE CURIE ,2011

150. Gasparetti F, Sansonetti G, Micarelli A. Community detection in social recommender systems: a survey. Appl Intell 2021; 51:3975–95. https://doi.org/10.1007/s10489-020-01962-3.

151. Gauch S, Speretta M, Chandramouli A, Micarelli A (2007) User profiles for personalized information access. In: Brusilovsky P, Kobsa A, Nejdl W (eds) The adaptive web. Springer, Berlin, pp 54–89

152. Tchuente D, Canut M-F, Baptiste-Jessel N, Peninou A, Sedes F (2012) A community-based algorithm for deriving users' profiles from egocentrics networks. In: Proceedings of 2012 international conference on advances in social networks analysis and mining (ASONAM 2012), ASONAM '12. IEEE Computer Society, Washington, DC, pp 266–273

153. Lalwani D, Somayajulu DV, Krishna PR (2015) A community driven social recommendation system. In: 2015 IEEE international conference on Big Data (Big Data). IEEE, pp 821–826

154. Park Y, Park S, Jung W, Lee SG (2015) Reversed cf: A fast collaborative fltering algorithm using a k-nearest neighbor graph. Expert Syst Appl 42(8):4022–4028

155. Shi C, Liu J, Zhuang F, Philip SY, Wu B (2016) Integrating heterogeneous information via exible regularization framework for recommendation. Knowl Inf Syst 49(3):835–859

156. Lee WP, Tseng GY (2016) Incorporating contextual information and collaborative filtering methods for multimedia recommendation in a mobile environment. Multimed Tools Appl 75(24):16719–16739

157. Sobhanam H, Mariappan AK (2013) Addressing cold start problem in recommender systems using association rules and clustering technique. In: 2013 international conference on computer communication and informatics. IEEE, pp 1–5

158. Hui et al

159. Xinchang et al

160. Fatemi, M.; Tokarchuk, L. A Community Based Social Recommender System for Individuals & Groups. In 2013 International Conference on Social Computing; IEEE: Alexandria, VA, USA, 2013; pp 351–356. https://doi.org/10.1109/SocialCom.2013.55.

161. Louvain “BLONDEL, Vincent D, GUILLAUME, Jean-Loup, LAMBOITTE, Renaud, et al. Fast unfolding of communities in large networks. Journal of statistical mechanics: theory ans experiment,2008, vol 2008, no 10, p.P10008.

162. P. Bedi and S. Chawla, Agent based information retrieval system using information scent, Journal of Artificial Intelligence **3**(4) (2010), 220–238.

163. Wu J, Jiao Y (2014) Clustering dynamics of complex discretetime networks and its application in community detection. Chaos 24(3):268–4

164. PizzutiC (2008) GA-Net: a genetic algorithm for community detection in social networks. Springer-Verlag, New York, pp 1081–1090