N° d'ordre: 95

Année: 2014

# UNIVERSITE ABDELMALEK ESSAADI
# ECOLE NATIONALE DES SCIENCES APPLIQUEES
# TANGER

Centre d'Etudes Doctorales: « Sciences et Techniques de l'Ingénieur»
Formation Doctorale: «Sciences et Techniques de l'Ingénieur»

## THESE DE DOCTORAT

Présentée
Pour l'obtention du

**DOCTORAT EN SCIENCES ET TECHNIQUES DE L'INGENIEUR**

Par :

**NAJMA HAMZAOUI**

Discipline : Informatique
Spécialité : Système de Recommandation

Titre : **Nouvelles Techniques de Recommandation et de Détection des Communautés**

**Soutenue le 26 Juillet 2014 devant le Jury :**

| | | |
|---|---|---|
| ***Pr. Ahmed EL MOUSSAOUI*** | Présidence de l'UAE | *Président* |
| ***Pr. Kamal Eddine EL KADIRI*** | ENSA de Tétouan | *Rapporteur* |
| ***Pr. Saiida LAZAAR*** | ENSA de Tanger | *Rapporteur* |
| ***Pr. Chakir El AMRANI*** | FST de Tanger | *Rapporteur* |
| ***Pr. Youssef MERIOUH*** | ENCG de Tanger | *Examinateur* |
| ***Pr. Abdelouahid LYHYAOUI*** | ENSA de Tanger | *Directeur de thèse* |
| ***Pr. Abdelfettah SEDQUI*** | ENSA de Tanger | *Co-encadrant* |

*Structure de recherche accréditée d'accueil:*
*Laboratoire des Technologies Innovantes de l'Ecole Nationale des Sciences Appliquées de Tanger*

# REMERCIEMENTS

Ces quatre années de thèse représentent pour moi une expérience enrichissante sur le plan scientifique avec des échanges intéressants et constructifs et des discussions riches avec différents collègues de travail. L'aide et le soutien de ces personnes aide à améliorer mon savoir-faire, mon savoir-être et mes connaissances.

Mes remerciements et ma gratitude vont à mon directeur de thèse, Monsieur Abdelouahid Lyhyaoui, Professeur à l'ENSA de Tanger. Grâce à ses bonnes directives, et son expérience dans la recherche, j'ai pu avancer, tout en m'accordant une grande liberté de travail. Je remercie également mon encadrant Monsieur Abdelfettah Sedqui, Professeur à l'ENSA de Tanger, de ses capacités scientifiques qui m'ont permis de développer mes aptitudes pour la recherche, je lui en suis très reconnaissante.

J'adresse aussi mes très sincères remerciements à Monsieur le directeur de l'Ecole Nationale des Sciences Appliquées de Tanger, le Professeur Abderrahmane SBIHI, de me faire l'honneur de s'intéresser à ce travail et d'avoir présidé le jury.

Un grand merci aux rapporteurs, qui ont consacré de leur temps pour lire ma thèse et l'évaluer. Leurs questions et remarques vont grandement participées à la finition de ce manuscrit.

Je remercie tous les membres du jury qui ont accepté d'évaluer mon travail.

Merci aussi très chaleureusement à l'ensemble de l'équipe du laboratoire des Technologies Innovantes, plus particulièrement à Wafae Baida pour son implication dans mes recherches et mes collègues Ikram, Amina, Maha, Jihane, Asma, Hayat, Souad, Wafae, Abdelatif, Ismail qui ont su m'épauler.

Enfin, je remercie profondément mes parents, toute ma famille et mes amis pour leur aide tout au long de ces quatre années.

Enfin, je tiens à remercier toutes les personnes que je n'aurais pas encore citées ici, qui ont contribué de près ou de loin à cette thèse.

# RESUME

Si la technologie est un prolongement naturel du comportement des utilisateurs, l'adaptation des différentes solutions techniques devrait permettre idéalement de simplifier les activités humaines dans leurs formes originales. Le comportement naturel humain d'une personne consiste à s'inspirer des expériences d'autres personnes. Ce type d'induction constitue l'essence de l'intelligence collective de la communauté afin de satisfaire le besoin de l'utilisateur. L'intelligence collective et la sensibilité au contexte, sont deux technologies utilisées dans les systèmes intelligents. La première permet d'apprendre et de dériver de nouvelles informations à partir de la composition d'expériences de leurs utilisateurs. La seconde rend ces systèmes capables de raisonner sur leur connaissance abstraite sur ce qui se passe.

Tous les agents intelligents comme les systèmes de recommandation peuvent obtenir des informations personnalisées. En effet, ce sont des systèmes qui ont pour objectif d'aider les utilisateurs à trouver des items intéressants, prévoir l'information pertinente qui répondra à leur satisfaction et leurs besoins réels grâce à un processus de recueil, de filtrage et de recommandation.

Avec l'énorme masse d'information circulant dans le Web, il est de plus en plus difficile de trouver rapidement et efficacement les informations nécessaires et utiles. Cependant, avec l'apparition des systèmes de recommandation au cours des années 90, la réduction de la surcharge d'information est devenue facile.

L'idée de départ lors de développement du système de recommandation, était d'observer simplement que l'utilisateur a tendance à s'appuyer sur les recommandations des autres utilisateurs pour la prise de la décision. Le filtrage collaboratif est considéré comme la technique de recommandation la plus réussie. En effet, il est le plus utilisé dans les systèmes de recommandation pour le e-commerce. Cette technique permet de recommander un élément à un utilisateur en fonction des profils des utilisateurs qui lui sont les plus proches.

De nos jours, la dernière génération des méthodes de Filtrage Collaboratif nécessite encore des améliorations supplémentaires pour rendre la recommandation plus efficace et plus précise. La plupart des algorithmes de filtrage collaboratif existants souffrent encore du problème de la rareté, l'évolutivité et le démarrage à froid.

Dans cette thèse, nous proposons des solutions pour ces problèmes rencontrés dans le filtrage collaboratif via quatre méthodes. Le filtrage collaboratif multicritère basé -item (MCCR, Multi-Criteria Collaborative Recommender) et sa nouvelle formulation théorique pour améliorer la recommandation et résoudre le problème de la rareté. La méthode de regroupement (STGM, Straight Through Grouping Model) pour obtenir les communautés, résoudre le problème de l'évolutivité et le problème de démarrage à froid. La méthode de recommandation d'une liste d'items sans calcul de prédiction basée sur la co-dissimilarité et l'arbre couvrant de poids minimum (RMCS, Recommendation model based on Co-dissimilarity and Spanning Tree) basée sur la théorie des graphes. Finalement, la méthode proposée pour la classification des mesures des similarités (Logical Actions of trees for the comparison of Classification Methods, LATCCM) utilisées dans les systèmes de recommandation pour améliorer leurs performances.

# TABLE DES MATIERES

# LISTE DES FIGURES

# LISTE DES TABLEAUX

# LISTE DES ABREVIATIONS

Pour des raisons de lisibilité, la signification d'une abréviation ou d'un acronyme n'est souvent rappelée qu'a sa première apparition dans le texte d'un chapitre. Par ailleurs, puisque nous utilisons toujours l'abréviation la plus usuelle, il est fréquent que ce soit le terme anglais qui soit employé, dans tel cas nous présentons une traduction.

| | |
|---|---|
| RS | Recommender System |
| CF | Collaborative Filtering |
| Rating | Note, vote, évaluation ou score |
| BEA | Bond Energy Algorithm |
| TG | La Technologie de Groupe |
| ACM RecSys | Association of Computing Machinery Recommender systems |
| MCCR | Multi-Criteria Collaborative Recommender |
| STGM | Straight Through Groping Method |
| RMCST | Recomendation Method based on Co-dissimilarity and Spanning Tree |
| LATCCM | Logical Actions of Trees for the Comparison of Classification Methods |
| TF-IDF | Term Frequency- Inverse Document Frequency |
| MEV | Modèle d'Espace Vectoriel |
| MIT | Massachsetts Institue of Technology |
| FMN | La Factorisation en Matrices non Négatives |
| DVS | La Décomposition en Valeurs Singulières |
| ACP | L'Analyse en Composantes Principales |
| EM | Coefficient d'Energie Maximum |
| QAP | Quadratic Affectation Problem |

# INTRODUCTION GENERALE

Grâce à l'Internet, les ressources de l'information sont devenues un moyen de communication, d'échange et d'aide à la décision. Ces sources peuvent influencer le choix de l'utilisateur par des technologies de l'information et des outils qui aident à la décision.

L'émergence d'Internet en tant que canal de distribution des sources de l'information a suscité l'intérêt des chercheurs au cours de ces dernières années, en raison de la facilité d'accès à une quantité énorme d'informations provenant de ces multiples sources. Certains ont même suggéré qu'Internet causerait un changement de paradigme en marketing [1]. Traditionnellement, les sources de l'information impersonnelles peuvent seulement fournir de l'information non personnalisée aux utilisateurs [2,3,4]. Mais cela est remis en question avec la venue des nouvelles technologies de l'information et des outils d'aide à la décision.

Actuellement, l'intelligence artificielle remplace l'intelligence de l'homme. Tous les agents intelligents comme les systèmes de recommandation (SR) peuvent obtenir des informations personnalisées. Leur but est de réduire la surcharge de l'information grâce à un processus de recueil, filtrage et recommandation de l'information d'une manière proactive. Par exemple, les systèmes de recommandation tentent de prédire quels sont les produits les plus adaptés aux utilisateurs, en se basant sur leurs préférences collectées de plusieurs façons, afin d'effectuer la tache de recommandation. La recommandation doit prévoir les objets pertinents qui répondent à la satisfaction des utilisateurs plutôt que de proposer des suggestions en rapport avec une politique commerciale. Nous pouvons rencontrer cette situation dans les applications de commerce électronique. A titre d'exemple, pour favoriser l'introduction de nouveaux items vestimentaires de partenaires d'Amazon, le système suggère de fausses recommandations [5]. Ces suggestions des items par les systèmes de recommandation à un utilisateur peuvent être des propositions d'items à acheter, de nouvelles à lire, de la musique à écouter, ou des films à voir, des livres à lire. Le mot «item» est le terme général utilisé pour dénoter ce que le système de recommandation recommande aux utilisateurs.

Le développement du e-commerce a conduit au progrès des SR, un nouveau type de source qui a contribué à l'émergence d'un nouveau champ de recherche qui s'intéresse aux SR dont les premiers articles fondateurs sont apparus au milieu des années 90. Les SR peuvent être utilisés pour fournir une information personnalisée orientée surtout vers les utilisateurs qui n'ont pas suffisamment de compétences pour évaluer la quantité immense d'items [6]. Ces dernières années sont révélatrices de l'utilisation des SR, car ils ont un rôle important dans les applications du m-commerce et les sites web tels que Allocine, Netflix, Amazon, Ebay,

Last.fm, Youtube, IMDb, Yahoo. Par exemple, pour améliorer son SR, Netflix a décerné un prix d'un million de dollars à l'équipe qui a réussi cette mission [7]. Les SR sont devenus un domaine de recherche indépendant, qui est relativement nouveau comparé aux recherches sur les systèmes d'information classiques. Nous pouvons par exemple, citer ACM Recommender System[1], crée en 2007 qui est devenu rapidement le principal événement annuel dans la recherche sur la recommandation.

De nombreuses entreprises sont entrain de développer et déployer des systèmes de recommandation. Dans le cadre des services, les SR ont prouvé qu'ils sont un bon moyen pour faire face au problème de surcharge cognitive. L'utilisateur peut avoir une recommandation pertinente par le système en fonction de connaissances variées (feedbacks d'utilisateurs, profil de l'utilisateur, le contexte, les items à recommander) et par l'action de l'utilisateur qui peut être enregistrée d'une manière implicite ou explicite. Ces actions génèrent de nouvelles recommandations pour la prochaine interaction avec le système.

En effet, l'idée de départ lors de développement du système de recommandation, était simplement d'observer que l'utilisateur avait tendance à s'appuyer sur les recommandations des autres utilisateurs pour la prise de la décision [8,9]. Actuellement, les moteurs de recommandation reposent sur trois paradigmes [10,11,12,106] : ceux basés sur le contenu, ceux basés sur le filtrage collaboratif, et ceux basés sur l'hybride qui est la combinaison des deux premiers paradigmes[13]. Les méthodes de filtrage collaboratif peuvent être classifiées comme des approches à base de mémoire ou à base de modèle [21]. Elles ont démontré leur efficacité que ça soit au niveau recherche, comme dans la pratique.

De nos jours, la dernière génération des méthodes de Filtrage Collaboratif, a besoin encore d'améliorations supplémentaires pour rendre la recommandation plus efficace et plus précise, afin de satisfaire l'utilisateur. Cette génération nécessite des algorithmes robustes et évolutifs surtout que les utilisateurs et les items sont en augmentation continue. La plupart des algorithmes de filtrage collaboratif existants souffrent encore des problèmes de la rareté, l'évolutivité et le démarrage à froid.

Le problème de la rareté apparaît lorsque le nombre de notes déjà obtenu pour un système de recommandation est généralement très faible par rapport au nombre de notes qui doivent être prédites. Le problème du passage à l'échelle (évolutivité ou scalabiliy) des algorithmes de

[1] http://recsys.acm.org/

Filtrage Collaboratif se manifeste lorsque les utilisateurs et items sont en augmentation continue. Quand des items n'ont pas encore été notés par des utilisateurs, les systèmes collaboratifs ne peuvent pas les recommander, cela engendre le problème de démarrage à froid.

De plus, le succès du système de recommandation dépend de la disponibilité d'une masse critique d'utilisateurs et des items. Par exemple, il peut y avoir plusieurs items qui n'ont été notés que par quelques personnes et ne seraient recommandés que très rarement, même si leur notes étaient élevées.

D'autre part, pour les utilisateurs dont les goûts sont inhabituels par rapport au reste de la population, il n'y aura pas d'autres utilisateurs qui sont particulièrement similaires aboutissant à des recommandations pauvres [106].

Une façon de surmonter le problème de la rareté des notes consiste à utiliser des informations de profil utilisateur lors du calcul de similarité, c.à.d, deux utilisateurs peuvent être considérés comme similaires, non seulement si ils ont évalué les mêmes items de la même façon, mais aussi s'ils appartiennent au même segment contexte. Par exemple, [167] utilise le sexe, l'âge, l'indicatif régional, et de l'information sur l'emploi des utilisateurs dans une application pour la recommandation des restaurants. Cette extension des techniques traditionnelles de filtrage collaboratif est parfois appelé "filtrage démographique" [167]. Ces systèmes utilisent les caractéristiques d'utilisateurs, et peuvent aussi utiliser les caractéristiques des items [33].

Toutefois, la plupart de ces approches ne sont pas sensibles au contexte. Les données utilisées dans la plupart de ces approches sont les scores et sont considérées comme statiques, c.à.d, les caractéristiques de l'item ou l'utilisateur et les changements dans l'intérêt de l'utilisateur ne sont pas pris en considération, et les scores produits à des moments différents sont pondérés de la même manière.

En effet, la problématique du filtrage collaboratif est très générale. Dans notre méthodologie de travail de cette thèse, nous essayons de résoudre les problèmes précités en répondant aux quatre questions suivantes:

P**remière question**: Est-ce qu'on peut renforcer l'algorithme de filtrage collaboratif par d'autres critères pour améliorer sa précision?

Cette suggestion apporterait de plus des nouvelles connaissances. On n'est plus dans l'espace user- item. Dans un premier temps, nous cherchons par ceci à renforcer le filtrage collaboratif classique, moyennant un modèle multicritère afin de proposer une recommandation plus précise. Ce modèle sera évoqué dans la première partie du troisième chapitre.

Dans un second temps, nous nous sommes intéressés à l'approche à base de modèle dans notre stratégie de prédiction. Parmi les méthodes les plus utilisées du filtrage collaboratif basé sur les modèles, se trouve celui utilisant les modèles de classification. La classification non supervisée (clustering) sur des données matricielles, formées d'utilisateurs et d'items, permet de former des blocs pertinents et significatifs. Ainsi, La classification croisée (co-clustering) permet de former simultanément des clusters d'utilisateurs et d'items, de façon à ce que les utilisateurs considérés comme similaires génèrent les mêmes scores.
Pour résoudre le problème de la rareté, une approche différente pour traiter les matrices creuses de notation a été utilisée dans [81], [168], où une technique de réduction de dimension, a été utilisée pour réduire la dimension de matrices.

En dépit de la rareté, le défi le plus important dans le FC est l'évolutivité. De nombreux chercheurs ont montré que l'utilisation de la technique de co-clustering est plus robuste pour résoudre le problème de l'évolutivité, et elle est un moyen viable pour augmenter l'évolutivité tout en conservant une bonne qualité de recommandation [68,69,77].
Une classe récente de modèles réussis de filtrage collaboratif est basée sur la factorisation matricielle. De nombreuses méthodes ont étudié l'usage des méthodes de factorisation pour le co-clustering [81, 82, 83, 84, 85, 86, 87, 28, 29, 30, 88]. Les approches de FC basées sur un modèle tentent de fournir des résultats plus précis que les systèmes à base de mémoire.
Les méthodes à base de modèle (*Model based*) construisent un modèle de prédiction, souvent probabiliste, en se basant sur une partie de données, certaines d'entre celles utilisent des techniques de clustering. Les modèles basés sur le clustering ont une meilleure évolutivité (scalability) que les méthodes classiques de FC, parce qu'ils font des prédictions dans des clusters, plutôt que sur l'ensemble de la base des données [89,90,91,92]. La plupart des applications utilisent diverses formes de techniques de génération des clusters.
**Deuxième question** : Est-ce que nous pouvons trouver un algorithme de clustering plus robuste et plus évolutif, par rapport aux existants, pour traiter les matrices creuses?
Le filtrage collaboratif, qui constitue une des solutions techniques implémentant l'intelligence collective, est déjà utilisé avec succès dans plusieurs systèmes de

recommandation, par exemple dans les boutiques web qui suggèrent les items à acheter en fonction de nos préférences et des personnes ayant le même goût.

Une classe récente de modèles réussis dans les méthodes de factorisation pour le co-clustering fournit juste des approximations de la matrice d'origine. Tandis que notre solution que nous voulons proposer pour résoudre le problème de l'évolutivité et de la rareté, et le démarrage à froid est fondée sur un bi-clustering naturel qui permet de former simultanément des clusters d'utilisateurs et d'items, de façon à ce que les utilisateurs considérés comme similaires génèrent des scores identiques. La réponse détaillée à cette question sera traitée dans la deuxième partie du troisième chapitre.

De plus, dans les systèmes de recommandation, les différents travaux portent sur des systèmes proposant la recommandation d'un seul item et rarement qu'ils recommandent une liste d'items.

**Troisième question** : pouvons-nous offrir une recommandation d'un ensemble d'items sans calcul de prédiction ?

La troisième idée avancée dans cette thèse est que, dans certains types d'applications, il peut être préférable, qu'au lieu de proposer un seul item ou une liste d'items satisfaisant quelques critères du profil, proposer une combinaison d'items, qui, lorsqu'ils sont associés les uns aux autres, correspondent à la satisfaction complète de l'utilisateur. Ainsi, notre approche consiste à rechercher des combinaisons d'items à proposer sans calcul de notes de prédiction, en se basant sur la théorie de graphes. La réponse à cette question sera traitée dans la troisième partie du troisième chapitre. Afin d'évaluer la qualité de notre méthode de recommandations, nous avons utilisé le jeu de données MovieLens[2] fournie par l'équipe de recherche Américaine GroupLens.

Enfin, nous nous intéressons à la comparaison des mesures de similarité. Le choix de cette mesure est très important, malheureusement, dans la majorité des cas, il s'agit d'un choix arbitraire.

**Quatrième question** : Est ce que nous pouvons proposer une classification de ces mesures de similarité pour faciliter le choix ?

Une solution pour répondre à cette question est présentée dans la quatrième partie du troisième chapitre.

Le reste de la thèse est organisé comme suit :

[2] http://www.grouplens.org/node/73

Le premier chapitre, présente plus formellement les deux approches les plus communément utilisées dans le SR : le filtrage thématique (à base du contenu) et le filtrage collaboratif. Nous nous concentrons sur le filtrage collaboratif qui est le plus en vogue aujourd'hui, ainsi que les techniques utilisées.

Le deuxième chapitre, présente le clustering et la théorie des graphes, et leur utilisation pour les systèmes de recommandation. Le but principal de ce chapitre est d'introduire les fondements nécessaires pour les chapitres suivants.

Le troisième chapitre est consacré à la présentation de nos approches pour la recommandation. Nous proposons des solutions pour les problèmes rencontrés dans le filtrage collaboratif, sous forme de quatre méthodes. Le filtrage collaboratif multicritère basé sur l'item (MCCR, Multi-Criteria Collaborative Recommender) et sa nouvelle formulation théorique pour améliorer la recommandation et résoudre le problème de la rareté. La méthode de regroupement (STGM, Straight Through Grouping Model) pour obtenir les communautés, résoudre le problème de l'évolutivité et le problème de démarrage à froid. La méthode de recommandation d'une liste d'items sans calcul de prédiction basée sur la co-dissimilarité et l'arbre couvrant de poids minimum (RMCS, Recommendation model based on Co-dissimilarity and Spanning Tree) basée sur la théorie des graphes, et finalement la méthode proposée pour la classification des mesures des similarités (Logical Actions of trees for the comparison of Classification Methods, LATCCM) utilisées dans les systèmes de recommandation pour améliorer ses performances. Les résultats expérimentaux seront abordés dans le quatrième chapitre. Finalement, nous présentons une conclusion générale qui résume nos travaux et qui met en évidence les axes de recherche à développer.

# *CHAPITRE I*

# LES SYSTEMES DE RECOMMANDATION

## 1.1. Introduction

Ce chapitre rassemble l'état de l'art de différentes méthodes et techniques utilisées dans les système de recommandation. Ces systèmes traitent le problème de la surcharge cognitive (ou surcharge d'informations). Trois types d'approches sont principalement utilisés : le filtrage basé sur le contenu, le filtrage collaboratif, et le filtrage hybride. Nous nous intéressons plus formellement aux deux approches les plus communément utilisées : le filtrage basé sur le contenu et le filtrage collaboratif. La première approche basée sur le contenu recherche des items en se basant sur ses caractéristiques et le profil de l'utilisateur. La deuxième approche par filtrage collaboratif recherche des items en se basant sur les choix d'autres utilisateurs dans le système. Les deux classes d'algorithmes de filtrage collaboratif, sont les algorithmes basés sur la mémoire et les algorithmes basés sur le modèle. Ce chapitre permet, en outre, d'identifier certaines limitations du filtrage basé sur le contenu et du filtrage collaboratif.

L'émergence et le développement du commerce électronique a conduit au progrès des systèmes de recommandation, un domaine de recherche en plein essor. Ces derniers permettent aux entreprises de filtrer l'information, puis de recommander de manière proactive des produits à leurs clients en fonction de leurs préférences. Recommander des produits et des services peut renforcer la relation entre l'acheteur et le vendeur, et donc augmenter les bénéfices [14]. Les systèmes de recommandation doivent veiller à accroître la satisfaction des utilisateurs. Ces dernières années sont révélatrices de l'utilisation des systèmes de recommandation sur le Web à travers l'intelligence collective, la sensibilisation au contexte et le social computing [15, 16, 17]. Les articles [18,19] présentent une classification détaillée des systèmes de recommandation pour le commerce électronique, et élucident la façon dont ils peuvent être utilisés pour fournir un service personnalisé fidélisant le client. Actuellement, les moteurs de recommandation [10] reposent sur ces deux paradigmes.

## 1.2. Les systèmes de recommandation basés sur le contenu

L'objectif des SR à base du contenu est de cibler des objets pertinents issus d'un large espace de sources possibles d'une façon personnalisée pour les utilisateurs. Son principe consiste à recommander les items similaires à ceux préférés par l'utilisateur dans le passé. Ces systèmes à base du contenu considèrent les caractéristiques des items afin de les corréler au profil des utilisateurs. En effet, chaque utilisateur possède un profil le décrivant à travers ses centres d'intérêts. Dans le but de recommander de nouveaux items intéressants, les SR à

base de contenu essayent de faire correspondre les attributs des items avec les préférences et les intérêts de l'utilisateur. Pour un nouvel item, le système compare l'item avec le profil de l'utilisateur afin de prédire le score que pourrait porter l'utilisateur sur l'item. Les items sont alors recommandés en fonction de leur proximité aux utilisateurs.

La recherche dédiée aux SR inclut différents domaines, notamment, la recherche d'information (RI), et l'intelligence artificielle [20]. En recherche d'information, les utilisateurs expriment leurs besoins en donnant une requête, alors que dans le système de filtrage d'information, le besoin est exprimé par le profil de l'utilisateur.

Tandis qu'en intelligence artificielle, la recommandation est fondée sur un modèle appris à l'aide des techniques d'apprentissage en exploitant les préférences passées des utilisateurs qui constituent leur profils. Tout simplement, les profils reflètent les intérêts à long terme de l'utilisateur et ils sont représentés par des vecteurs de mots-clefs. [21] propose un SR à base de modèle qui permet de faire des prédictions par construction d'un modèle. Ce système a pour objectif de prévoir l'information qui répond à la satisfaction et les besoins réels sans déranger l'utilisateur. Cela implique l'application des systèmes d'apprentissage qui vont apprendre le profil d'utilisateur sans exiger à le fournir et à catégoriser les nouvelles informations en se basant sur celles déjà stockées. Etant donné un nouveau item, le modèle prédictif fourni par les méthodes des systèmes d'apprentissage sera capable de prédire le degré d'intérêt que peut porter l'utilisateur pour l'item.

Pour cette classe de SR à base de contenu, nous ne pouvons pas oublier les techniques de représentation des items et les algorithmes de recommandation utilisés. Les items sont représentés par un ensemble de caractéristiques, par exemple, les descriptions des items dans la plupart des systèmes de filtrage à base de contenu sont des caractéristiques textuelles contrairement aux données structurées, il n'y a pas d'attribut avec les données bien définies.

A cause de l'ambiguïté du langage, la construction d'un profil utilisateur par analyse de caractéristiques textuelles engendre de nombreuses complications. Les profils basés sur des mots-clefs traditionnels ne sont pas capables de capturer la sémantique des intérêts des utilisateurs, car ils sont essentiellement générés par une opération de correspondance de chaînes. Alors, si une correspondance est trouvée à la fois dans le profil et dans le document, le document est considéré comme approprié. Cette correspondance de chaine souffre des problèmes de polysémie et de synonymie. La gestion de ces deux problèmes nécessite le développement de techniques d'analyse sémantique. La polysémie rend pertinents de

mauvais documents et la synonymie ne permet pas au système d'identifier toutes les informations pertinentes.

Mais, la plupart des SR utilisent de simples modèles de recherche, comme la correspondance de mots clefs ou le modèle d'espace vectoriel (MEV) avec la pondération basique du terme le plus communément utilisé (Terme Frequency- Inverse Document Frequency, TF-IDF) basés sur des observations empiriques sur le texte [22].

Dans ces modèles, chaque document est représenté par un vecteur de dimension N, où chaque dimension correspond à un terme de l'ensemble du vocabulaire d'une collection de documents. Formellement, tout document est représenté par un vecteur poids sur ces termes, où chaque poids indique le degré d'association entre le document et le terme.

De l'analyse des principaux systèmes développés pendant ces 20 dernières années, le plus important à retenir est qu'il est nécessaire qu'un nombre suffisant de preuves d'intérêt des utilisateurs soit disponible pour que la représentation, à la fois des items et des profils par des mots clefs, donne des résultats précis.

La plupart des SR à base de contenu sont conçus comme des classificateurs de textes construits à partir d'un ensemble de documents d'apprentissage qui sont soit, des exemples positifs, soit des exemples négatifs des intérêts de l'utilisateur. Par exemple, "Personal Web Watcher", [23], apprend les intérêts des utilisateurs à partir des pages web qu'ils visitent et à partir des documents qui ont un lien hypermédia avec les pages visitées. Il traite les documents visités comme des exemples positifs d'intérêts pour l'utilisateur et des documents non visités comme des exemples négatifs.

Les approches basées sur les mots clefs souffrent des limites lorsque des caractéristiques plus complexes sont nécessaires, d'où le besoin d'avoir des stratégies de représentation plus avancées, pour que les SR à base de contenu prennent en compte l'information susceptible d'être pertinente pour l'utilisateur et la sémantique associée aux mots.

En conclusion, pour les méthodes de filtrage à base de contenu, celles qui incorporent la connaissance linguistique, et/ou spécifique, offrent des meilleures prestations que les méthodes traditionnelles.

### 1.3. Les Systèmes de recommandation basés sur l'approche collaborative

Le terme *Collaborative Filtering* est défini comme une technique utilisant les comportements connus d'une population pour prévoir les agissements futurs d'un individu à partir de l'observation de son attitude dans un contexte donné. Un premier exemple personnalisé, "Tapestry "a été mis en place chez Xerox en 1992 [24]. Deux ans plus tard, Paul Resnick du MIT (Massachsetts Institue of Technology) et ses collaborateurs de l'université de Minnesota ont proposé l'architecture *Grouplens* pour recommander des articles dans les newsgroups [25]. La librairie *Amazon* a popularisé le filtrage collaboratif avec sa fonction «les utilisateurs qui ont aimé ce livre ont aussi aimé tel autre livre». En 1998, Brin et Page ont publié leur algorithme PageRank et lance *Google*. A la même année chez Microsoft, John S. Bresse et ses collaborateurs présentent une comparaison détaillée des divers algorithmes de filtrage collaboratif [21].

Durant les années 2000, les algorithmes de FC étaient basés sur les réseaux bayésiens ou les réseaux de neurones avec une approche basée sur l'utilisateur. En 2003, *Amazon* dépose un brevet introduisant le filtrage collaboratif basé sur l'item [27]. Ce type d'algorithme a été également publié la même année et de façon indépendante par la communauté *Grouplens*. En 2006, la compagnie *Netflix* annonce son challenge avec une récompense très attrayante, rendent ainsi disponible un ensemble de données réelles et volumineuses pour évaluer les SR [26]. En 2009, *Netflix* a décerné un prix d'un million de dollars à l'équipe qui a réussie à améliorer les performances de son SR [26]. Une classe récente de FC est développée, basée sur la factorisation matricielle et sur le contexte [30,31, 32, 33].

Sans avoir besoin d'information exogène sur les items et les utilisateurs comme dans le filtrage à base de contenu, le FC se base sur des schémas de notation pour produire des recommandations d'items à des utilisateurs donnés. La plupart des méthodes traditionnelles de FC utilisent l'évaluation d'un item par l'utilisateur sur un item lors du calcul de la similarité. D'autres travaux [34, 35, 36, 37, 38, 167] montrent l'importance de l'information démographique et calculent la similarité des utilisateurs à partir de leurs évaluations et leurs informations démographiques. Aussi, des articles ont mis l'accent sur l'utilisation du temps pour les SR : dans [39,40] les auteurs ont montré que l'année de production d'un film affecte significativement les préférences des utilisateurs cibles. Loren Terveen et al [41] définissent les préférences des utilisateurs en utilisant leurs histoires personnelles. Kazunari Sugiyama a exploré un type de FC en fonction du temps avec une analyse détaillée de l'historique de navigation de l'utilisateur en un jour [42]. Yanchang Zhao et al ont proposé une fonction de

pénalisation des scores en les considérants comme une série chronologique [43]. Le caractère récent des évaluations a été étudié aussi dans [44,45]. De nombreuses applications de FC ont été mises en service pendant une longue période, accumulant plusieurs évaluations d'utilisateurs, dont certains sont très anciennes. Cependant, ce caractère récent des notes n'a pas été utilisé jusqu'à présent dans un modèle convexe pour ajuster la prédiction qui utilise la similarité basée sur les notes des items, et celle basée sur les attributs d'items à prévoir automatiquement la préférence d'un utilisateur.

Actuellement, Web-catch-up TV a révolutionné les habitudes car il offre aux utilisateurs la possibilité de regarder des programmes en temps et lieu préféré, en utilisant une variété de dispositifs. Avec l'offre croissante de contenu de télévision, il y a un besoin émergeant des solutions de recommandation personnalisée, qui aident les utilisateurs à choisir des programmes d'intérêt. [46] a développé une série d'approches de recommandation à partir des modèles de l'observation des utilisateurs d'un fournisseur de service de rattrapage de télévision à l'échelle nationale Australienne.

L'approche de FC s'appuie sur l'hypothèse que les gens, lors de la recherche d'information, devraient se servir des notes d'autres utilisateurs, à la différence des approches du filtrage à base de contenu, qui utilisent juste les items précédemment notés par un seul utilisateur. Cette approche vient résoudre certains problèmes de l'approche à base de contenu. Ainsi, il devient possible de traiter n'importe quelle forme du contenu grâce aux retours des autres utilisateurs, et de diffuser des items avec des contenus différents non nécessairement similaires à ceux déjà reçus, tant que les autres utilisateurs manifestent leurs intérêts pour ces différents items. Pour ce faire, pour chaque utilisateur, un ensemble de plus proches voisins doit être identifié, et la décision de proposer ou non un item à un utilisateur, dépendra des appréciations de son voisinage. De plus, les recommandations collaboratives sont basées sur la qualité des items évalués par les utilisateurs, au lieu de s'appuyer sur le contenu qui peut être un mauvais indicateur de qualité.

Afin de prédire l'intérêt d'un utilisateur pour un item, des connaissances sur l'utilisateur ou sur l'item doivent être assimilées par le système de recommandation. Ces informations sont regroupées dans une matrice appelée matrice d'usage.

Une définition formelle de la recommandation a été introduite par [47] :

Soit C l'ensemble de tous les utilisateurs et P l'ensemble de tous les items qui peuvent être recommandés. Soit U un ensemble ordonné et $u: C \times P \rightarrow U$ une fonction mesurant le score

de l'utilisateur $c \in C$ sur l'item $p \in P$. Pour chaque utilisateur c, le système de recommandation sélectionne l'item $p' \in P$ qui maximise le score ou l'utilité de c.

$$\forall c \in C, p'_c = arg\max_{p \in P} u(p, c) \qquad (1)$$

Le score ou l'utilité d'un utilisateur c pour un item p, noté *u(p,c)* est généralement représenté par une note . Un exemple de matrice d'usage qui regroupe ces notes des utilisateurs sur des items est représenté dans la table 1.

| | $P_1$ | $P_2$ | $P_3$ | $P_4$ | $P_5$ | $P_6$ |
|---|---|---|---|---|---|---|
| $c_1$ | | 3 | | 2 | | 2 |
| $c_2$ | 1 | | 1 | | 3 | |
| $c_3$ | | 5 | 4 | 1 | 4 | 5 |

Table 1: Matrice d'usage de trois utilisateurs et six items

La construction de la matrice d'usage se fait soit à partir du filtrage collaboratif passif, qui repose sur l'analyse des comportements des utilisateurs, par exemple les pages web visitées sur une période de temps prédéfinie [48], soit à partir du filtrage collaboratif actif, qui repose sur les données déclarées par les utilisateurs telles que les notes [49].
La recommandation peut être faite par l'exploitation de la matrice d'usage de deux façons différentes. La première approche est connue sous le terme de recommandation basée sur les utilisateurs (*user-based*) [51], calcule les similarités entre les utilisateurs $c_i$ et $c_j$ à l'aide de leurs profils. La deuxième approche, une recommandation basée sur les items (*item-based*) [49,51], qui calcule les similarités entre items $p_i$ et $p_j$ selon les mesures attribuées par les utilisateurs. Une autre méthode combinant les deux approches est proposée dans [50].

Les méthodes collaboratives peuvent être groupées en deux classes générales [21]: La première se base sur le voisinage (basées sur la mémoire, memory based) [21,25,100], il s'agit de comparer chaque recommandation pour l'utilisateur courant à l'ensemble de la base de données. La deuxième classe utilise les méthodes à base de modèle, qui construisent un modèle de prédiction, souvent probabiliste, sur une partie de la base de données.

### 1.3.1. Recommandation basée sur le voisinage

Les SR basés sur le voisinage(ou à base de mémoire) se fondent sur l'avis de personnes partageant les mêmes idées pour donner une évaluation sur un item p.

Ainsi, dans le FC basé sur le voisinage, les notes sont directement utilisées pour prédire les scores des nouveaux items.

Dans ce qui suit, Nous introduisons les deux approches basées sur le voisinage entre item et celui entre utilisateur du FC, ainsi que leurs étapes, y compris le calcul de la similarité, et la phase de prédiction

#### 1.3.1.1. Voisinage entre utilisateur

Dans ce cas, le score est généré en utilisant les notes attribuées aux items. Ces notes sont données par d'autres utilisateurs, appelés voisins qui ont des habitudes de notations similaires à l'utilisateur c en question. Donc, l'approche basée sur le voisinage estime la similarité entre les utilisateurs ayant les mêmes comportements seuls ceux ayant notés l'item p peuvent être utilisés dans la prédiction.

Dans ce qui suit, nous allons détailler le reste des étapes à savoir le calcul de la similarité et la prédiction.

**a) Calcul de la similarité entre utilisateurs ou items:**

Le calcul de la similarité entre utilisateurs ou items consiste à mesurer la similitude entre les lignes ou les colonnes de la matrice d'usage. Le choix de la mesure utilisée dépend généralement de la nature des éléments, dont les composants sont des notes. Il faut dire qu'il existe plusieurs méthodes de mesure de similarité, mais nous allons parler des plus utilisées ou les plus populaires : la similarité Cosinus [21,51] et la similarité de Pearson [25,100], définies respectivement comme suit

$$\underset{cosinus}{S}(a,b) = \frac{\sum_{x \in E_a \cap E_b} u(a,x) \times u(b,x)}{\sqrt{\sum_{x \in E_a \cap E_b} u(a,x)^2 \sum_{x \in E_a \cap E_b} u(b,x)^2}} \qquad (2)$$

Avec a, b sont deux utilisateurs ou deux items, $E_a$ est l'ensemble des items mesurés par l'utilisateur a et $E_b$ l'ensemble des items mesurés par l'utilisateur b.

$$\underset{pearson}{S}(a,b) = \frac{\sum_{x \in E_a \cap E_b} (u(a,x) - \bar{u}_a) \times (u(b,x) - \bar{u}_b)}{\sqrt{\sum_{x \in E_a \cap E_b} (u(a,x) - \bar{u}_a)^2 \sum_{x \in E_a \cap E_b} (u(b,x) - \bar{u}_b)^2}} \qquad (3)$$

Où $\bar{u}_a$ (respectivement $\bar{u}_b$) représente la moyenne des valeurs contenues dans le vecteur a (respectivement b).

En revanche, si les éléments contiennent uniquement des données binaires, la distance de Jaccard peut être utilisée [10] :

$$\underset{Jaccard}{S}(a,b) = \frac{|E_a \cap E_b|}{|E_a \cup E_b|} \qquad (4)$$

Il s'agit du rapport entre la cardinalité de l'intersection des ensembles considérés et la cardinalité de l'union des ensembles.

**b) Prédiction**

La prédiction consiste à calculer l'intérêt qu'un utilisateur pourrait porter à un item ou plusieurs items encore non mesurés.

Le principe consiste d'abord à rechercher les utilisateurs possédant les mêmes comportements que l'utilisateur courant. Dès lors, les recommandations sont prédites en fonction des mesures de ces utilisateurs les plus proches.

On peut prédire la note $u(p,c)$ de l'utilisateur $c$ pour l'item $p$ par la moyenne des notes de ces voisins, mais le problème avec cette méthode est qu'elle ne prend pas en compte le fait que les voisins peuvent avoir des niveaux différents de similarité S. En effet, une solution pour prédire la note de l'utilisateur, est de pondérer la contribution de chaque voisin par sa similarité à $c$.

De telle sorte que la note prédite devient :

$$u(p,c) = \frac{\sum_{\{c_i \in E_p\}} S(c,c_i) \times u(p,c_i)}{\sum_{\{c_i \in E_p\}} S(c,c_i)} \qquad (5)$$

Avec $E_p$ de C l'ensemble de tous les utilisateurs ayant mesuré l'item $p$

Néanmoins, un problème majeur du FC est la notation des utilisateurs. En effet, si un utilisateur considère que la perfection n'existe pas, il n'affectera jamais la note maximale à un item et donc repartir ses notes de 1 à 4 (si les notes possibles sont de 1 à 5). À l'inverse, un utilisateur différent peut, s'il n'aime pas noter trop sévèrement, repartir les notes qu'il attribue de 2 à 5. Pour pallier ce problème, la moyenne des notes de l'utilisateur $c_i$ est introduite à la formule de la note prédite :

$$u(p,c) = \bar{u}_c + \frac{\sum_{\{c_i \in E_p\}} S(c,c_i) \times (u(p,c_i) - \bar{u}_{c_i})}{\sum_{\{c_i \in E_p\}} S(c,c_i)} \qquad (6)$$

Où $\bar{u}_c$(respectivement $\bar{u}_{c_i}$) représente la moyenne des notes de l'utilisateur $c$ (respectivement $c_i$)

#### 1.3.1.2. Voisinage entre Items

Dans ce cas, le score généré par l'utilisateur c sur l'item p est calculé en se basant sur les notes des items similaires à p [27,51, 52]. Les scores générés par les utilisateurs sur ces derniers sont similaires. Ces approches Item-based rassemblent les items dont les scores sont identiques. L'intérêt pour les approches de FC basées sur les items est plus récent que celui des approches de FC sur les utilisateurs [51,53]. *Amazon* [28] a mis en avant cette approche avec un système construisant une matrice de relation entre les items en se basant sur les achats.

Le calcul cette fois-ci de la note d'un utilisateur $c$ pour un item $p$, est formalisé comme suit :

$$u(p,c) = \frac{\sum_{\{p_j \in E_c\}} S(p,p_j) \times u(p_j,c)}{\sum_{\{p_j \in E_c\}} S(p,p_j)} \qquad (7)$$

Avec $E_c$ de P l'ensemble de tous les items mesurés par l'utilisateur $c$.

Pour pallier les différences d'utilisations des mesures, la moyenne des notes de chaque utilisateur est introduite dans la formule suivante:

$$u(p,c) = \bar{u}_p + \frac{\sum_{\{p_j \in E_c\}} S(p,p_j) \times (u(p_j,c) - \bar{u}_{p_j})}{\sum_{\{p_j \in E_c\}} S(p,p_j)} \qquad (8)$$

Où $\bar{u}_p$(respectivement $\bar{u}_{p_j}$) représente la moyenne des notes $p$ (respectivement $p_j$)

### 1.3.2. Recommandation basée sur un modèle

Les approches basées sur un modèle mettent en œuvre des méthodes issues de l'apprentissage automatique comme les modèles bayésiens ou les méthodes de clustering. Ces méthodes sont généralement performantes mais ont un coût de conception et de fonctionnement plus important que les méthodes basées sur la mémoire [56,55]. Néanmoins, dans le cas de données dispersées, ces méthodes semblent plus efficaces. Pour le lecteur intéressé, une description précise des approches basées sur les modèles est proposée par Su et Khoshgoftaar [56].

A la différence des systèmes basés sur le voisinage qui utilisent les notes stockées pour le calcul de la prédiction, les approches basées sur un modèle utilisent ces notes pour construire

un modèle prédictif par apprentissage. L'idée générale est de modéliser les interactions utilisateur-item avec des facteurs représentant des caractéristiques latentes des utilisateurs et items dans le système, comme des classes d'utilisateurs et d'items. Ce modèle est ensuite conçu à partir des données disponibles, et utilisé plus tard pour prédire les notes des utilisateurs pour de nouveaux items. Les approches basées sur un modèle sont nombreuses, elles incluent les méthodes de clustering [21], l'analyse de la sémantique latente [57], les machines de Boltzmann [58], les machines à support vectoriel [58], et la décomposition en Valeur singulière [60,61, 62].

Les approches basées sur les modèles comme les méthodes de clustering ont été étudiées pour remédier aux insuffisances des algorithmes de FC à base de mémoire [21,60].Les méthodes existantes de clustering les plus classiques pour le FC peuvent être classées en trois catégories : Les méthodes de partitionnement, les méthodes basées sur la densité et les méthodes hiérarchiques [61,62]. Une méthode de partitionnement couramment utilisée est l'algorithme *k-means* proposée par [63] qui a deux avantages : l'efficacité relative et la mise en œuvre facile. Les méthodes de clustering basées sur la densité recherchent généralement des classes denses d'objets séparés par des zones creuzes et elles sont bien connues comme méthodes de classification fondées sur la densité [64,65].

Les méthodes hiérarchiques décrites dans [66], créent une décomposition hiérarchique de l'ensemble des objets de données en utilisant quelques critères.

Dans la plupart des situations, le clustering est une étape intermédiaire et son résultat est utilisé pour le calcul de l'évaluation. Les méthodes de clustering pour le FC peuvent être appliquées de différentes façons à savoir le mono-clustering, le bi-clustering, le co-clustering. Dans le cas de mono-clustering [67, 68, 69] les données sont partitionnées en classes utilisant l'algorithme de FC à base de mémoire et la corrélation de *pearson* comme mesure de similarité.

Dans le cas de l'approche bi-clustering, le filtrage implique à la fois le clustering des utilisateurs et celui des items simultanément. [70] propose un partitionnement simultané pour le filtrage collaboratif en temps réel. Les auteurs du [71,72] ont utilisé aussi une méthode de co-clustering, mais en introduisant une analyse de la dualité entre les utilisateurs et les items, avec une proposition d'une nouvelle mesure de similarité. [73] classe les utilisateurs et les items séparément, utilisant les variations des moyens et l'échantillonnage de *Gibs*. [74] applique le clustering des utilisateurs en se basant sur les items qu'ils évaluent et le clustering des items en se basant sur les utilisateurs qui les ont notés. Les utilisateurs peuvent être réorganisés en fonction du nombre d'items qu'ils évaluent et les items peuvent être regroupés

de la même façon. Chaque utilisateur est affecté à une classe avec un degré d'appartenance proportionnelle à la similarité entre l'utilisateur et la moyenne de la classe. Un modèle de mélange flexible (MMF) regroupe les utilisateurs et les items en même temps, permettant à chaque utilisateur et item d'être dans plusieurs classes et modélisant séparément les classes des utilisateurs et des items [75]. Les résultats expérimentaux montrent que l'algorithme MMF a une meilleure précision que l'algorithme de FC basé sur la corrélation de Pearson [76]. En dépit de la rareté, le défi le plus important de FC est l'évolutivité. De nombreux chercheurs ont trouvé que l'utilisation de la technique de co-clustering est plus robuste pour résoudre ce problème, et elle est un moyen viable pour augmenter l'évolutivité tout en conservant une bonne qualité de recommandation [68,69,77]. Ainsi, lorsque la base de données est grande,[78,79] compressent les données d'abord en construisant un modèle de clustering, les recommandations sont ensuite générées en utilisant une approche efficace basée sur les plus proches voisins. Un résumé des travaux sur le FC basés sur le clustering peuvent être consultés dans [80].

Une classe récente de modèles réussis de filtrage collaboratif est basée sur la factorisation matricielle. De nombreuses méthodes ont étudié l'usage des méthodes de factorisation pour le co-clustering, comme est le cas des méthodes SVD, NMF,Tri-NMF, PMF, Non linear PMF, Bayesian PMF, et NPCA [81, 82, 83, 84, 85, 86, 87, 28, 29, 30, 88].

Les modèles de clustering ont une meilleure évolutivité que les méthodes classiques de FC, parce qu'ils font des prédictions dans des petites classes, plutôt que sur l'ensemble de la base des clients [89,90,91,92]. Cependant, le calcul de clustering complexe et coûteux est géré hors ligne. Toutefois, la qualité de recommandation est généralement faible, il est possible de l'améliorer en utilisant plusieurs segments fins [27]. Etant donné que le clustering optimal sur les grands ensembles de données n'est pas possible, la plupart des applications utilisent diverses formes de techniques de génération des classes, en particulier ceux avec une forte dimensionnalité, dans ce cas l'échantillonnage ou la réduction de dimensionnalité se voit nécessaire.

Les SR basés sur des modèles tentent à fournir des résultats plus précis que les systèmes basés sur le voisinage. Cependant, la grande partie des travaux de recherche et des systèmes commerciaux (par exemple, *Amazon* [27], *TiVo* [93] et *Netflix* [94] sont basés sur le voisinage. Actuellement, il existe beaucoup plus de systèmes de recommandation basés sur le voisinage, car ils sont considérés comme plus faciles et intuitifs à manipuler. Tout d'abord, ils fournissent naturellement des explications plus intuitives du raisonnement derrière les

recommandations, ce qui améliore l'expérience utilisateur. Enfin, ils peuvent immédiatement délivrer des recommandations à l'utilisateur en se basant sur le retour qu'il vient de fournir.

### 1.4. Les limitations des types du système de recommandation

Parmi les méthodes présentées, nous remarquons que les SR souffrent de certaines limitations. Ainsi, les systèmes à base de contenu présentent certains problèmes, entre autre, la difficulté d'indexation des documents multimédia. Le filtrage à base de contenu s'appuie sur un profil qui décrit le besoin de l'utilisateur du point de vue thématique. Ce profil peut prendre divers formats et il repose toujours sur des termes qui seront comparés aux termes qui indexent le document. De ce fait, la difficulté d'indexer des documents multimédia ou non est un goulet d'étranglement pour cette approche. L'incapacité à traiter d'autres critères de pertinence que les critères strictement thématiques, pose également un problème. Il existe plusieurs facteurs de pertinence comme la qualité scientifique des faits présentés, la fiabilité de sources d'informations, le degré de précision des faits présentés, public visé, etc... C'est-à-dire qu'il y a une analyse limitée du contenu. Les techniques à base de contenu ont une limite sur le nombre et le type de caractéristiques qui sont associées aux objets à recommander. La connaissance du domaine est souvent nécessaire, aucune recommandation basée sur le contenu ne peut fournir de suggestions convenables, si l'analyse de contenu ne contient pas d'information pour discriminer les items que l'utilisateur refuse.

L'effet dit «entonnoir» restreint le champ de vision des utilisateurs. En effet, le profil évolue toujours dans le sens d'une expression de besoins de plus en plus spécifique lors de la mise en place d'un filtrage thématique. Ainsi, l'utilisateur ne reçoit que les recommandations relatives à ses préférences, une fois que son profil devient stable, parce que ce dernier évolue naturellement par restriction progressive sur les thèmes recherchés. Par conséquent, il ne peut pas découvrir de nouveaux domaines pouvant potentiellement l'intéresser. Par exemple, lorsqu'un nouvel axe de recherche surgit dans un domaine, avec de nouveaux termes pour décrire les nouveaux concepts, ces termes n'apparaissent pas dans le profil, ce qui élimine automatiquement les documents par filtrage, l'utilisateur n'aura jamais l'occasion d'exprimer un retour de pertinence positif en vers ce nouvel axe de recherche, à moins d'en avoir connaissance et de modifier son profil manuellement en ajoutant les termes pertinents. Cet inconvénient est appelé problème de «sur-spécialisation» ou «heureux hasard» ou «entonnoir».

Le paradigme du filtrage collaboratif apporte précisément une réponse à ces problèmes, en s'appuyant sur l'avis d'une communauté d'utilisateurs. Les trois limitations du système à

base de contenu (difficulté d'indexation, l'incapacité à traiter d'autres critères, effet entonnoir) n'apparaissent pas dans le filtrage collaboratif.
En réponse au problème d'indexation, la sélection ne s'appuie plus sur le contenu des documents, mais sur les opinions que les utilisateurs émettent sur les documents. Un autre avantage de la recommandation basée sur les opinions, c'est qu'elle reflète les autres facteurs de pertinence utiles aux utilisateurs. En effet, lorsqu'un utilisateur émet une opinion positive sur un document, il affirme non seulement que le document traite bien un sujet qui l'intéresse, mais aussi que ce document est de bonne qualité et qu'il lui convient à lui personnellement (public visé). Ainsi, le problème de l'incapacité à traiter d'autres facteurs est également résolu. La qualité de l'information est connue via des évaluations d'utilisateurs.
Enfin, l'effet entonnoir est lui aussi éliminé, du fait que les documents entrants ne sont pas filtrés en fonction du contenu. A l'inverse, le FC n'est pas soumis à l'effet entonnoir, car les utilisateurs peuvent tirer profit des mesures d'intérêt des autres utilisateurs et recevoir les recommandations pour lesquelles les utilisateurs le plus proches ont émis un intérêt. Alors, le système peut suggérer des documents sans rapport explicite avec les thèmes déjà évoqués.
Cependant, les systèmes collaboratifs ont leurs propres limites. Un problème du FC est celui du démarrage à froid, Actuellement, il existe trois types de démarrage à froid : le système débutant, le nouvel utilisateur, et le nouvel item.

Le problème du « système débutant » survient lorsque la matrice d'usage est vide. Les méthodes de filtrage collaboratif ne peuvent fonctionner que s'il existe des informations dans cette matrice d'usage. La solution consiste soit à trouver des variables descriptives des items afin d'organiser ces derniers entre eux et inciter les utilisateurs à les parcourir, remplissant ainsi la matrice d'usage, soit à collecter des données externes en fonction du domaine applicatif.
Afin de formuler des recommandations précises pour un utilisateur, le système de FC doit d'abord apprendre les préférences de l'utilisateur à partir de ces scores. Le deuxième type de démarrage à froid concerne le nouvel utilisateur. Plusieurs solutions existent : lui soumettre un questionnaire sur les items, ou faire de la recommandation éditoriale afin de l'inciter à parcourir les items et ainsi enrichir le système. Pour éviter cette tâche fastidieuse pour l'utilisateur, certains auteurs proposent d'associer le nouvel utilisateur à un « stéréotype » en exploitant par exemple une source d'informations démographiques externe comme les pages web personnelles des internautes. Le troisième type est celui du nouveau item: dans le cas du

filtrage collaboratif, un item n'ayant reçu aucune note, ou n'ayant jamais été acheté ne peut être recommandé.

De nouveaux items sont ajoutés régulièrement à des systèmes de recommandation. Les systèmes collaboratifs reposent uniquement sur les préférences des utilisateurs pour faire des recommandations. Par conséquent, jusqu'à ce que le nouvel élément soit évalué par un nombre important d'utilisateurs, le système de recommandation ne serait pas en mesure de recommander. Il s'agit alors de le rendre visible aux utilisateurs afin d'obtenir un certain nombre de mesures d'intérêt (typiquement le cas des « fausses » recommandations).

Le système de FC exige une base de données substantielle et plusieurs évaluations de l'utilisateur avant d'être utilisable. Dans tout système de recommandation, le nombre de notes déjà obtenu est généralement très faible par rapport au nombre de notes qui doivent être prédites, ce problème est connu sous l'appellation de la rareté « sparcity». De plus, le succès du système de recommandation collaboratif dépend de la disponibilité d'une masse critique d'utilisateurs et d'items.

### 1.5. Synthèse de la classification des approches de filtrage collaboratif.

Nous présentons dans la table 2, une synthèse des approches de FC [95,96,97,10,170] :

| | Les techniques de FC | |
|---|---|---|
| | Approche à base de mémoire | Approche à base de modèle |
| Avantages | - Simpliste<br>- Performance<br>- Réactif | - Raisonnement prédictif<br>- Moindre complexité |
| Inconvénients | - Complexité combinatoire | - Non dynamique |

Table 2: Synthèse des techniques de FC

- Les algorithmes basés sur la mémoire offrent l'avantage d'être réactifs, en intégrant dynamiquement des nouveaux utilisateurs ou items. Si ces méthodes fonctionnent bien sur des exemples de tailles réduites, il est souvent difficile de passer à des situations proposant un grand nombre d'items ou d'utilisateurs, à cause de la complexité combinatoire des algorithmes utilisés.
- Les algorithmes basés sur un modèle offrent une valeur ajoutée au-delà de la seule fonction de prédiction. En effet, ils mettent en lumière certaines corrélations dans les

données, proposant ainsi un raisonnement intuitif pour les recommandations, une autre manière d'aborder le problème du filtrage collaboratif consiste à classifier les utilisateurs et les items en groupes. Pour chaque groupe d'utilisateurs, il s'agit d'estimer la probabilité qu'un item soit choisi. Ces approches souffrent bien souvent du problème de convergence lié à l'initialisation des clusters et fournissent dans certains cas des recommandations de mauvaise qualité. Les algorithmes basés sur un modèle minimisent le problème de la complexité combinatoire. Cependant, ces méthodes ne sont pas dynamiques et elles réagissent mal à l'insertion de nouveaux contenus dans la base de données.

- Les algorithmes basés aussi bien sur la mémoire et sur le modèle offrent une alternative combinant les avantages des deux approches.

### 1.6. Domaines d'applications

Il existe de nombreux systèmes collaboratifs développés autant dans le monde industriel que dans le monde académique. Le système Grundy [98] était le premier système de recommandation utilisant les stéréotypes en tant que mécanisme pour la construction de modèles. Plus tard, le système Tapestry [24] s'est appuyé sur chaque client pour identifier les clients partageant les mêmes idées. GroupLens [47], Video Recommender [99], et Ringo [100] utilisent également des algorithmes de filtrage collaboratif. Nageswara et Talwar [11] proposent une classification des systèmes de recommandation en six catégories suivant la fonctionnalité à laquelle ils répondent :

- Content-based filtering systems : utilisant les données sur les items et le profil de l'utilisateur courant.
- Collaborative filtering systems : utilisant des données sur un ensemble de comportements D'utilisateurs interagissant avec un item.
- Demographic filtering systems : utilisant des données démographiques telles que l'âge, le sexe, le niveau social, etc. permettant de segmenter des populations les rapprochant de certains items.
- Knowledge-based recommender systems : utilisant de la connaissance fonctionnelle pour générer des recommandations
- Utility-based recommender systems : utilisant une fonction d'utilité sur les items pour aider à la recommandation.
- Hybrid recommender systems : utilisant plusieurs approches pour minimiser les inconvénients de certaines méthodes.

Montaner et al. [101] produisent une taxonomie et classifient les systèmes de recommandation existants en plusieurs catégories :

- Les divertissements (entertainment) : films, musiques, etc.
- Les contenus (content) : actualités personnalisées, pages Web, applications de e-learning, antispams, etc. ;
- Le commerce électronique : livres, appareils photos, ordinateurs, etc.
- Les services (services) : voyages, expertises, locations, etc.

Ricci et al. [10] présentent une classification des domaines de recommandations existants en fonction de plusieurs critères d'évaluation subjectifs dont le risque d'impact sur le client suite à une mauvaise recommandation. Il constate par exemple, que les sites d'assurance vie, de tourisme et de recherche d'emplois ont davantage de risque que les sites de commerce électronique, d'actualités, de films ou de musiques.

Les systèmes de recommandation sont vitaux pour les sites de commerce en ligne, dont les exemples les plus frappants sont *Amazon*, *NetFlix*, *Pandora* et *Strands*. Les systèmes de recommandation touchent principalement aujourd'hui quatre domaines commerciaux en ligne : les **films**[3], la **musique**[4], les **livres**[5] et la **publicité**[6]. La recherche dans le domaine des systèmes de recommandation en **m-commerce**[7] s'est d'ailleurs accélérée ces dernières années. Dans ce cadre, les applications sont nombreuses et variées, nous pouvons mentionner par exemple le tourisme [102] ou la recommandation dans le domaine de la restauration [103]. Le m-commerce ouvre des perspectives de recherche autour de la mobilité, de la capacité de calcul limitée, des capacités de transmission, de la taille de l'écran, etc.

La table 3 présente une liste non exhaustive d'exemples de systèmes de recommandation commerciaux et académiques, leur domaine d'application et la technique de filtrage utilisée.

| | | Systèmes collaboratifs | | |
|---|---|---|---|---|
| Système | Domaine | Thématique | Collaboratif | Hybride |
| *Adaptive Place [104]* | Restaurants | | ✓ | |
| *Amazon [27]* | Livres, films, etc. | | | ✓ |

[3] http://www.netflix.com
[4] http://www.last.fm
[5] http://www.amazon.com
[6] http://www.facebook.com
[7] . Les applications en m-commerce ne couvrent pas seulement les applications du e-commerce, mais également les nouvelles applications qui peuvent être exécutées à tout moment, de n'importe quel endroit via les mobiles ou les tablettes [113].

| | | | | |
|---|---|---|---|---|
| *Eigenstate [105]* | Académique | | ✓ | |
| *Fab [*106*]* | Livres | | | ✓ |
| *InfoFinder [107]* | Actualités | ✓ | | |
| *Last.fm [*108*]* | Musique | | | ✓ |
| *LIBRA [*109*]* | Livres | | | ✓ |
| *Google News [*110*]* | Actualités | | ✓ | |
| *GroupLens [47]* | Actualités | ✓ | | |
| *MovieLens [*111*]* | Films | | ✓ | |
| *MYCIN [*112*]* | Prescriptions | | ✓ | |
| *Netflix [*26*]* | Films | | | ✓ |
| *Org. Structure [*114*]* | Appareils photos | | ✓ | |
| *Pandora [*115*]* | Musique | | ✓ | |
| *RecTree [*69*]* | Images | | | ✓ |
| *Ringo [*116*]* | Musique | | ✓ | |
| Tapestry [24] | Images | | ✓ | |
| SASY [117] | Vacances | | ✓ | |
| Top Case [118] | Vacances | | ✓ | |
| TrustWalker [119] | Académique | ✓ | | |
| IMDb [120] | Films | | | ✓ |
| Ebay [121] | Tout article confondu | | | ✓ |
| Alibaba [122] | Tout article confondu | | | ✓ |
| Google Play [123] | Application mobile, musique, films, etc | | | ✓ |
| iTunes [124] | Application mobile, musique, films, etc | | | ✓ |

Table 3 : Classification des systèmes collaboratifs commerciaux et académiques

*CHAPITRE II*

# CLUSTERING ET GRAPHES POUR LES SYSTEMES DE RECOMMANDATION

## 1.1 Introduction

La première partie de ce chapitre décrit l'importance du clustering, la définition de la similarité entre éléments qui constitue la base de toute méthode de clustering ainsi que les méthodes du clustreing les plus utilisées en présentant ses applications dans les systèmes de recommandation. Parmi les méthodes les plus utilisées dans les systèmes de recommandation basés sur un modèle, se trouve les modèles de classification. Nous faisons une analogie entre ces méthodes de classification et celles basées sur la factorisation matricielle.

La deuxième partie présente une vue générale sur les graphes et ses applications sur les systèmes de recommandation. De nombreuses applications Web reposent sur l'utilisation de graphes possédant une forte structure de communautés, afin de proposer aux utilisateurs des contenus personnalisés.

## 2.1. Le clustering

Le clustering consiste à structurer un ensemble d'objets en différents groupes(ou clusters) en fonction d'une certaine notion de ressemblance. Les objets qui sont considérés comme similaires sont ainsi associés au même cluster alors que ceux qui sont considérés comme différents sont associés à des clusters distincts. Cet axe de recherche sur le clustring est étudié depuis de nombreuses années dans différentes communautés : machine learning, data mining, pattern recognition, statistiques, etc [80].

Selon l'application envisagée pour le clustering, les considérations ne sont pas les mêmes, et la définition même de l'intérêt d'une solution peut varier. Par exemple, lorsque l'objectif est de réduire la taille d'un jeu de données pour un traitement plus efficace, alors une solution est jugée optimale lorsqu'elle minimise la perte d'information liée à une telle compression des données. Si le clustering est utilisé sur des bases de données de clients dans le but de faire émerger des groupes de clients aux comportements différents, alors le critère de qualité d'un résultat est principalement basé sur une séparation importante des groupes formés. Dans d'autres cas au contraire, considérer des clusters qui se chevauchent peut se révéler plus intéressant. Enfin, il est également possible de définir l'intérêt d'un clustering par une notion de densité des clusters produits. Les méthodes basées sur de tels critères sont alors en général mieux adaptées pour cibler des clusters de forme quelconque, alors que ces derniers seront plus efficaces si l'objectif est de fournir une représentation compréhensible des clusters identifiés.

Il est ainsi admis dans la communauté qui travaille sur le clustering qu'aucun critère ni aucune méthode ne sont intrinsèquement meilleurs que d'autres sur l'ensemble des problématiques et des difficultés envisageables. Par contre, certains critères et certaines méthodes sont plus appropriées que d'autres dans certains champs d'application et face à certains complications.

### 2.1.1. Notions de similarité

L'objectif du clustering est de regrouper un ensemble de données de la manière la plus naturelle possible. Cette volonté de regrouper naturellement est bien sûr ambiguë et le plus souvent formalisée par l'objectif de définir des groupes d'objets tels que la similarité entre objets d'un même groupe soit maximale et que la similarité entre objets de groupes différents soit minimale.

Le problème est alors de définir cette notion de similarité entre objets. Typiquement, la similarité ente objets est estimée par une fonction calculant la distance entre ces objets. Une fois cette fonction distance est définie, la tache de clustering consiste alors à réduire au maximum la distance entre membres d'un même cluster tout en augmentant au maximum la distance entre clusters.

Donc, l'apprentissage non supervisé nous contraint alors à disposer d'une distance définie sur le langage de description des objets. Ainsi, deux objets proches selon cette distance seront considérés comme similaires, et au contraire, deux objets séparés par une large distance seront considérés comme différents.

Le choix de cette mesure de ressemblance entre objets est très important. Malheureusement, trop souvent, il s'agit d'un choix arbitraire, sensible à la représentation des objets, et qui traite tous les attributs de la même manière.

Une solution pour palier à cette limitation est celle de la prise de la connaissance d'un expert, qui identifiera certains attributs, considérés comme plus pertinents que d'autres pour le problème considéré, et leur attribuera un poids plus important lors du calcul des distances entre objets. Cependant, cette solution devient très difficile à mettre en œuvre lorsque le nombre d'attributs décrivant les données est trop grand, ou lorsqu'il n y a pas d'expert humain.

### 2.1.2. Les méthodes de clustering

On distingue classiquement deux grandes familles de méthodes en clustering : Les méthodes hiérarchiques et les méthodes par partition. Dans le premier cas, une hiérarchie de clusters est formée de telle manière que plus on descend dans la hiérarchie, plus les clusters sont spécifiques a un certain nombre d'objets considérés comme similaires. Au contraire, dans le second cas, le résultat fourni est une partition de l'espace des objets, c'est-à-dire que chaque objet est associe à un unique cluster.

Dans le cas du clustering hiérarchique, deux grandes méthodes se distinguent :

Les méthodes du Clustering Ascendant Hiérarchique (CAH), démarrent avec autant de clusters que d'objets puis fusionnent successivement l'ensemble des clusters selon un certain critère jusqu'à ce que tous les objets soient finalement regroupés dans un unique cluster stocké à la racine de la hiérarchie.

Les méthodes du Clustering Descendant Hiérarchique (CDH), démarrent avec un unique cluster regroupant l'ensemble des objets, puis divisent successivement les clusters selon un certain critère jusqu'à ce que tous les objets se retrouvent dans des clusters différents stockés aux feuilles de la hiérarchie.

On retrouve donc à ce niveau la différence classique en apprentissage entre généralisation et spécialisation : les méthodes ascendantes démarrent avec une solution tout à fait spécifique aux données, qui est ensuite généralisée à chaque étape, alors que les méthodes descendantes démarrent avec une solution complètement générale, qui est ensuite spécialisée à chaque étape.

Dans le cas du clustering par partition, plusieurs méthodes se distinguent fortement :

- Le clustering statistique est basé sur l'hypothèse que les données ont été générées en suivant une certaine loi de distribution, le but étant alors de trouver les paramètres de cette distribution, ainsi que les paramètres cachés déterminant l'appartenance des objets aux différentes composantes de cette loi [125]; le clustering basé sur l'algorithme k-moyennes, méthodes très souvent utilisée, en est un cas particulier [126].
- Le clustering stochastique consiste à parcourir l'espace des partitions possibles selon certaines heuristiques, et à sélectionner celle qui optimise un critère donné [127],
- Le clustering basé sur la densité a pour but d'identifier dans l'espace les zones de forte densité entourées par des zones de faible densité pour la formation des clusters [128];

- Comme son nom l'indique, le clustering basé sur les grilles utilise une grille pour partitionner l'espace de description des objets en différentes cellules, puis identifie les ensembles de cellules denses connectées pour former les clusters [129] ;
- Le clustering basé sur les graphes consiste à former le graphe connectant les objets entre eux et dont la somme des valeurs des arcs, correspondant aux distances entre les objets, est minimale, puis à supprimer les arcs de valeurs maximales pour former les clusters [130];
- Enfin, la base du clustering spectral consiste à projeter itérativement les objets dans des sous-espaces de variance maximum, puis à utiliser une méthode de partitionnement dans de tels sous-espaces pour séparer les données [131].

Différentes méthodes hybrides ont également été proposées qui mélangent les caractéristiques de méthodes hiérarchiques et des méthodes par partition, cherchant ainsi à bénéficier des atouts de chaque méthode. De nombreux travaux ont montré que l'usage des méthodes de factorisation pour le clustering donne un meilleur résultat que l'algorithme k-means [132, 28,29,88].

Etant donné un ensemble de données $X = \{x_1, \dots, x_n\}$, l'objectif du clustering est de partitionner l'ensemble de données en k clusters $\{C_1, \dots, C_k\}$, selon certains principes. Par exemple, l'algorithme classique k-means atteint ce but en minimisant la fonction coût suivante

$$J = \sum_{k} \sum_{x_i \in C_k} \|x_i - f_k\|^2 \qquad (9)$$

Où $f_k$ est le centre du cluster $C_k$ .

Soient F et G deux matrices tel que $F = [f_1, f_2, \dots, f_K] \in \mathbb{R}^{m \times k}$ où la matrice F est de dimension $m \times k$ et $G = (g_{ij}) \in \mathbb{R}^{n \times k}$ où G de dimension $n \times k$.

$$g_{ij} = \begin{cases} 1 & \text{si } x_i \in C_j \\ 0 & \text{sinon} \end{cases}.$$

La fonction J sous la forme matricielle est :

$$J = \|X - FG^T\|_F^2 \qquad (10)$$

où $\|.\|_F$ désigne la norme matricielle de Frobenius. Donc le but de k-means est de trouver G en minimisant J, cela peut être réalisé à l'aide des techniques de factorisation [133].

Parmi toutes les méthodes de factorisation, la plus utilisée pour le clustering est la Factorisation en Matrices non Négatives (FMN).

La FMN est une méthode générale de décomposition matricielle, introduite par Lee et Seung [134]. Elle permet d'approximer toute matrice X de taille (m×n) et dont les éléments sont tous positifs, grâce à une décomposition de la forme $X \approx FG$, où F et G sont des matrices de dimension (m×k) et (k×n). La matrice X contient les vecteurs réels de dimension n, la matrice F contient les vecteurs correspondants dans un espace de dimension $k < n$, et la matrice de passage G contient les vecteurs de base. Cette méthode impose des contraintes de positivité ce qui la différencie notamment d'autres méthodes de séparation de sources, telle que la Décomposition en Valeurs Singulières (DVS) sur les coefficients des matrices utilisées, et cherche à minimiser une fonction objective.

Lorsque la matrice de données est positive, la décomposition matricielle est restreinte à des valeurs nulles et positives seulement.

Déterminer les matrices F et G revient à minimiser la distance entre la matrice initiale X et le produit F.G ; plus précisément, il faut minimiser la norme de Frobenius $J = \|X - FG^T\|_F^2$ sous les contraintes de non-négativité. C'est un problème d'optimisation non trivial, que Lee propose de résoudre en initialisant F et G aléatoirement, puis en alternant les mises à jour suivantes :

$$F_{ik} \leftarrow F_{ik}\frac{(XG)_{ik}}{(FG^TG)_{ik}} \qquad G_{jk} \leftarrow G_{jk}\frac{(X^TF)_{jk}}{(GF^TF)_{jk}} \qquad (11)$$

La FMN a été appliquée avec succès en plusieurs domaines, notamment en reconnaissance des visages [134], en classification de documents textuels, et en filtrage collaboratif [135]. Elle a connu un grand développement dans le domaine de clustering où plusieurs extensions de l'algorithme classique ont été réalisées.

De nombreuses expériences ont montré que la Factorisation en matrices non négatives (FMN) a été appliquée avec succès au clustering [134, 136] où une matrice X est décomposée en un produit de deux matrices de facteurs, dont l'une correspond à des centres de cluster (prototypes) et l'autre composé des vecteurs indicateur de cluster.

Soit une matrice $X \in \mathbb{R}^{m \times n}$ décomposé en F et G tel que $X \approx F.G$, où les matrices $F \in \mathbb{R}^{m \times K}$ et $G \in \mathbb{R}^{n \times K}$ sont positives. La dimension K correspond au nombre de clusters [87]. La décomposition matricielle est interprétée comme suit :

- Lorsque les éléments à regrouper sont sur les colonnes de X, i.e., X est formée de n vecteurs de dimension m, les colonnes de F sont considérées comme des vecteurs de base (centre de clusters) et chaque ligne de G contient la mesure avec laquelle chaque vecteur de base est utilisé pour reconstruire X.
- Dans le cas contraire, si les éléments à regrouper sont représentés dans les lignes de la matrice X, alors G est la matrice qui contient les centres de clusters et F est celle qui permet de retrouver X à partir de G

Une autre manière d'interpréter les résultats de FMN appliquée au clustering est de considérer une normalisation de la matrice de mesure afin de donner la probabilité à posteriori pour qu'un élément appartienne à un cluster donné [87]. Dans la suite, on présente un résumé des principales méthodes de factorisation appliquées pour le clustering.

- Décomposition en Valeurs Singulières (DVS)

La méthode classique de factorisation est l'analyse en composantes principales (ACP), qui utilise la décomposition en valeurs singulières, elle permet d'avoir des valeurs de signe positif et négatif. Cette méthode nous permet de décomposer la matrice initiale X en une matrice orthogonale U d'ordre$(m \times m)$, une matrice orthogonale V d'ordre$(n \times n)$, et une matrice "pseudo-diagonale" $\sum$ de dimension $(m \times n)$ telles que : $X = U\Sigma V^T$. Elle est utilisée pour estimer les entrées manquantes (les articles non votés) dans les systèmes de recommandation.

- NMF Orthogonale

[132] montre que lorsqu'une contrainte d'orthogonalité est imposée sur un facteur de la décomposition, elle permet de mieux interpréter les résultats du clustering. Les bases orthogonales obtenues à partir de NMF sont généralement plus appropriées pour le clustering, car ils ont tendance à mieux déterminer les centroïdes, tandis que la méthode classique de NFM tend à construire un convexe contenant les données.

- Tri-Factorisation

Considérons la décomposition de X en trois matrices suivantes $X \approx FSG^T$ ; dans le cas du co-clustering de la matrice X, [87,88] affirme que cette décomposition en trois matrices permet d'avoir une meilleure précision du co-clustering. F donne un clustering en ligne et G en colonne. Plus précisément, nous sommes amenés à minimiser

$$\|x_i - f_k\|^2 \quad \text{avec} \quad F^T.F = I \text{ et } G^T.G = I \qquad (12)$$

L'actualisation des trois matrices se fait comme suit :

$$G_{jk} \leftarrow G_{jk}\sqrt{\frac{(X^TFS)_{jk}}{(GG^TX^TFS)_{jk}}} \quad F_{ik} \leftarrow F_{ik}\sqrt{\frac{(XGS^T)_{ik}}{(FF^TXGS^T)_{ik}}} \quad S_{ik} \leftarrow F_{ik}\sqrt{\frac{(F^TXG)_{ik}}{(F^TFSG^TG)_{ik}}} \qquad (13)$$

Dans ce qui suit, nous présentons quelques méthodes des systèmes de recommandation qui utilisent des techniques du clustering.

### 2.1.3. Les systèmes de recommandation basés sur le clustering

Parmi les méthodes les plus utilisées dans les systèmes de recommandation basées sur les modèles, se trouve celles utilisant les modèles de classification [28, 88]. Les méthodes dites de clustering permettent de limiter le nombre d'utilisateurs considérés dans le calcul de la prédiction. Lorsque la corrélation porte sur les utilisateurs, on parle de « communautés d'intérêts ». Lorsqu'il s'agit de rapprocher les items en fonction des usages, c à d des items appréciés ou achetés par plusieurs utilisateurs, on parle de « clusters d'items ». Les prédictions sont ensuite inférées depuis ces modèles. Le temps de traitement est diminué et les résultats sont potentiellement plus pertinents puisque les observations portent sur un groupe d'utilisateurs ayant un comportement semblable [61]. Nous verrons dans cette section qu'il est possible de réaliser du clustering basé sur les corrélations non seulement entre utilisateurs, mais également entre items [51]. Par exemple, Shani et al. [140] utilisent les listes de films favoris des internautes sur leur blog *MySpace*[8] pour réaliser des recommandations basées sur les corrélations entre items, en appliquant sur ces listes des méthodes de clustering basées sur les co-occurences de films.

Dans ce qui suit, nous approfondirons particulièrement les algorithmes basés sur des modèles qui s'appuient sur des techniques du clustering.

- **K-Means**

[8] http://www.myspace.com

La méthode des k-means [108] consiste dans un premier temps à choisir aléatoirement k centres dans l'espace de représentation utilisateurs/items. Chaque point de l'espace correspond à un utilisateur dont les coordonnées sont les mesures. Ensuite, chaque utilisateur est positionné dans le cluster de centre le plus proche. Une fois les groupes d'utilisateurs formés, la position des centres est recalculée et l'opération réitérée jusqu'à l'obtention d'un état stable. La complexité algorithmique est en $O(k * n * t)$ où k est le nombre de clusters, n le nombre d'utilisateurs et t le nombre d'itérations. La prédiction fonctionne pour les algorithmes basés sur la mémoire. En utilisant ces méthodes de clustering, la prédiction ne porte plus sur l'ensemble des utilisateurs mais sur des clusters d'utilisateurs appartenant au même groupe que l'utilisateur courant ci. D'autres algorithmes proches des k-means existent. Par exemple, Firefly [100] consiste à sélectionner uniquement les profils utilisateurs dont la valeur de corrélation de Pearson [25,100] par rapport à l'utilisateur courant si est supérieure à un seuil fixé empiriquement.

La méthode des k-means est utilisée dans de nombreux systèmes de recommandation, cependant, il est très difficile de connaître le nombre k de centres appropriés. D'autre part, cet algorithme est coûteux en temps de calcul et peut présenter des problèmes de convergence. En effet, les clusters générés sont très dépendants de la phase d'initialisation de l'algorithme et il est souvent rare de tomber sur un optimum global minimisant les distances intra-groupes et maximisant les distances inter-groupes. D'une façon générale, il est même fréquent que l'algorithme ne converge pas. Par ailleurs, les résultats sont non reproductibles, c'est-à-dire que si l'on lance deux fois de suite l'algorithme avec des paramètres identiques, les résultats s'avèrent différents. Bien que le k-means représente la méthode de clustering la plus populaire, il existe des algorithmes concurrents comme *Repeated Clustering*, *Gibbs Sampling* [73] ou *RecTree* [141]. Seule la constitution des clusters varie dans ces méthodes, la phase de prédiction des mesures étant souvent similaire.

- **Repeated Clustering**

L'algorithme de « clustering répété » (Repeated clustering), [73] consiste à effectuer des affinements successifs des groupes d'utilisateurs. Ainsi, un premier regroupement d'utilisateurs par rapport aux items mesurés permet de définir des classes d'utilisateurs et/ou des classes d'items. Le processus est réitéré sur les classes issues du premier clustering. Cette méthode présente un problème de généralisation. Au cours des itérations successives, des utilisateurs aux profils et/ou aux comportements différents risquent d'être regroupés. L'algorithme *Gibbs Sampling* [73] est une méthode d'estimation de paramètres pour un modèle statistique, l'algorithme se déroule en deux phases :

1. Étape d'attribution : l'utilisateur se voit attribuer une classe avec une probabilité.
2. Étape d'estimation : le système estime la classe à laquelle appartient l'utilisateur pour un item donné. L'algorithme converge, mais il est extrêmement coûteux en temps de calcul.

- **RecTree**

Le clustering hiérarchique (*RecTree*) [141] cherche à fractionner l'ensemble des utilisateurs en cliques. Celles-ci sont hiérarchisées sous forme d'un arbre. L'algorithme commence par associer à la racine la répartition des mesures de tous les utilisateurs C. Pour construire l'arbre, on cherche à maximiser les similarités entre les utilisateurs d'un même clique et à minimiser celles entre les utilisateurs de deux cliques différentes. Ainsi, plus on descend dans l'arbre et plus les clusters sont spécifiques à certains groupes d'utilisateurs similaires. Plus on parcourt l'arbre en profondeur, plus les utilisateurs partagent une mesure semblable sur un item. Cet algorithme présente les mêmes problèmes de convergence que l'algorithme k-means. Il s'avère toutefois intéressant car il permet facilement de subdiviser les communautés d'utilisateurs. Il est potentiellement moins coûteux en temps de calcul que les k-means et il possède moins de complexité combinatoire [52].

Les performances du FC basé sur le clustering sont moins bonnes que celles fournis par les approches basées sur la mémoire. En effet, les communautés d'utilisateurs créées par le clustering sont rarement homogènes et certains utilisateurs appartenant à deux clusters différents peuvent apporter une information pertinente. En revanche, lors de l'utilisation de données volumineuses, le clustering permet de franchir le changement d'échelle pour appliquer par la suite une approche basée sur la mémoire. Une autre classe de technique qui nous permet d'aboutir à des clusters se base sur la théorie des graphes[142,88].

Dans ce qui suit, un aperçu sera donné sur les principaux aspects de la théorie des graphes, en particulier, l'arbre de recouvrement de poids minimal d'un graphe (MST).

### 2.2. Les graphes

La notion de graphe est relativement récente puisqu'elle n'est apparue formellement qu'au cours du XXe siècle, cependant, elle est devenue aujourd'hui indispensable dans de nombreux domaines, notamment en informatique fondamentale et appliquée, en optimisation, en complexité algorithmique, etc. L'étude des graphes et de leurs applications est donc l'occasion d'aborder des questions très diverses, dont les applications sont nombreuses. C'est ainsi qu'on développera par exemple les méthodes d'ordonnancement de taches à partir des chemins optimaux dans les graphes, ou encore, des propriétés de réseaux de communication quant à la connectivité des graphes. Historiquement, les graphes ont été en fait considérés,

bien avant d'établir leur cadre théorique, avec des problèmes connus comme celui des ponts de Königsberg [143]. L'émergence de superordinateurs, des réseaux et la nécessité d'une organisation efficace des calculs parallèles et distribués sur des vastes fichiers a déterminé la nécessité de renforcer la tendance pour l'utilisation des graphes comme l'outil le plus efficace pour l'automatisation de la programmation. Les graphes constituent, en effet, un sujet d'étude fertile, cette partie donne un bref historique et les principaux aspects de la théorie des graphes.

### 2.2.1. Un bref historique de la théorie des graphes

Tout le monde s'accorde à considérer que la théorie des graphes est née en 1736 avec la communication d'Euler (1707-1783) dans laquelle il proposait une solution au célèbre problème des ponts de Königsberg. Le problème posé était le suivant : deux îles A et D sur la rivière Pregel à Königsberg (alors capitale de la Prusse de l'Est, aujourd'hui rebaptisée Kaliningrad) étaient reliées entre elles ainsi qu'aux rivages B et C à l'aide de sept ponts désignés par des lettres minuscules) comme le montre la figure 1.

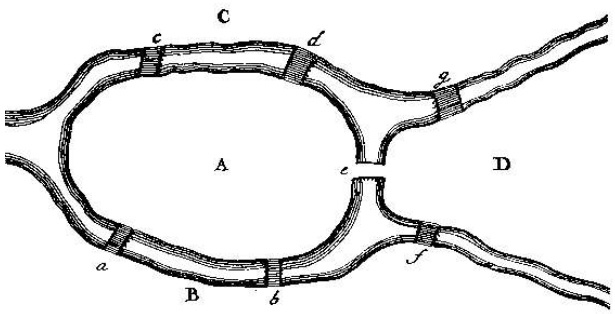


Figure 1: La rivière de l'île Pregel et de Kneiphof [143]

L'obstacle posé consistait, à partir d'une terre quelconque A, B, C, ou D, traverser chacun des ponts une fois et une seule et revenir à son point de départ (sans traverser la rivière à la nage). Euler représenta cette situation à l'aide d'un " dessin" où les sommets représentent les terres et les arêtes, les ponts comme le montre la figure 2.

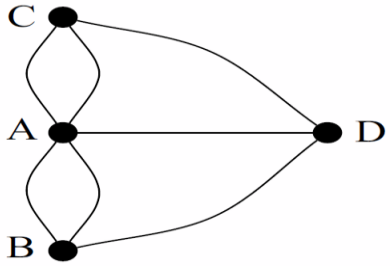


Figure 2: Graphe associé au problème des ponts de Königsberg[143].

Comme nous le montrerons ultérieurement, Euler démontra que ce problème n'a pas de solution et il est identique à celui consistant à tracer une figure géométrique sans lever le crayon et sans repasser plusieurs fois sur une même arête. Pendant tout un centenaire, la recherche dans ce domaine n'a pas connu de nouveautés, cependant, en 1847, Kirchhoff (1824-1887) développa la théorie des arbres pour l'appliquer à l'analyse de circuits électriques. Une décennie plus tard, Cayley (1821-1895) découvrit la notion d'arbre alors qu'il essayait d'énumérer les isomères saturés des hydrocarbures de type CnH2n+2. A cette époque, deux autres problèmes d'importance pour la théorie des graphes furent également proposés et partiellement résolus.

Le premier est la conjecture des quatre couleurs qui affirme que quatre couleurs suffisent pour colorier n'importe quelle carte plane telle que les "pays" ayant une frontière commune soient de couleurs différentes. C'est sans doute Möbius (1790-1868) qui présenta le premier ce problème dans l'un de ses cours en 1840. Environ dix ans après, de Morgan (1806-1871) essaya de résoudre ce problème. Les lettres de Morgan à ces divers collègues mathématiciens constituent les premières références à la conjecture des quatre couleurs. Le problème devient célèbre après sa publication, par Cayley en 1879. Ce problème est resté très longtemps sans solution, il fallut attendre jusqu'en 1976 pour qu'Appel et Haken [144] prouvent ce théorème en réduisant le problème à un nombre fini de situations particulières et en trouvant une solution pour chacune d'entre elles à l'aide d'un ordinateur.

Le second problème est dû à Sir Hamilton (1805-1865). En 1859, il inventa un casse-tête qu'il vendit pour 25 guinées à un fabricant de jouet de Dublin. Ce jeu consiste en un dodécaèdre régulier en bois (un polyèdre à 12 faces et 20 sommets), chaque face étant un pentagone régulier.

Trois arêtes sont donc issues de chaque sommet. Un clou est fiché sur chaque sommet marqué du nom de vingt grandes villes mondiales. Le casse-tête consiste à enrouler une ficelle passant une fois et une seule fois par chacune des villes (sommets). Bien que la solution de ce problème soit aisée à obtenir, personne n'a encore trouvé de condition nécessaire et suffisante de l'existence d'un tel chemin (appelé chemin Hamiltonien) dans un graphe quelconque.

Cette période fertile fut suivie d'un demi-siècle de relative inactivité : les années vingt virent la résurgence de l'intérêt pour les graphes. L'un des pionniers de cette période fut Konig à qui l'on doit le premier ouvrage consacré entièrement à la théorie des graphes (Konig, 1936) [145]. Il est sans doute à l'origine de l'utilisation du terme " graphe" pour désigner ce qui était préalablement considéré comme un ensemble de "points et de flèches". A partir de 1946, la

théorie des graphes a connu un développement intense sous l'impulsion de chercheurs motivés par la résolution de problèmes concrets. Parmi ceux-ci, citons de manière privilégiée Kuhn (1955), Ford et Fulkerson (1956) et Roy (1959). Parallèlement, un important effort de synthèse a été opéré en particulier par Claude Berge, son ouvrage " théorie des graphes et ses applications" publié en 1958 (Berge, 1958) marque sans doute l'avènement de l'ère moderne de la théorie des graphes par l'introduction d'une théorie des graphes unifié et abstraite rassemblant de nombreux résultats épars dans la littérature. Depuis, cette théorie a pris sa place, en subissant de très nombreux développement essentiellement dus à l'apparition des calculateurs, au sein d'un ensemble plus vaste d'outils et de méthodes généralement regroupés sous l'appellation " recherche opérationnelle" ou " mathématiques discrètes".

### 2.2.2. Théorie des Graphes : Définitions

Nous introduisons dans cette section quelques définitions fondamentales utilisées dans notre recherche, certaines issues de [146].

**Définition**

Soient X =un ensemble de sommets et A= {(x,y) : x$\in$X, y$\in$*X*}=ensembles des arêtes.

On appelle graphe G = (X,A) la donnée d'un ensemble X dont les éléments sont appelés sommets et d'une partie de A symétrique ( (x,y) $\in$ A , (y, x)$\in$ A) dont les éléments sont appelés arêtes. Le nombre de sommets est appelé ordre du graphe.

**Définition**

En présence d'une arrête a = (x,y) qui peut être notée simplement xy, on dit que x et y sont les extrémités de a, que a est incidente en x et en y, et que y est un successeur ou voisin de x (et vice versa).

**Définition**

On dit qu'un graphe est sans boucle si A ne contient pas d'arrête de la forme (x,x), c'est-à-dire joignant un sommet à lui-même.

**Définition**

Un graphe ne possédant pas de boucle ni d'arêtes parallèles (deux arêtes distinctes joignant la même paire de sommets) est appelé graphe simple ou 1-graphe.

Remarque : Graphiquement, les sommets peuvent être représentés par des points et l'arête a = (x,y) par un trait reliant x=(a,y). On notera que la disposition des points et la longueur ou la forme (rectiligne ou incurvée) des traits n'a aucune importance. Seule l'incidence des différentes arêtes et sommets compte.

**Définition**

On appelle graphe orienté ou digraphe G = (X,A) la donnée d'un ensemble X dont les éléments sont appelés sommets et d'une partie A dont les éléments sont appelés arcs ou arêtes.

**Définition**

En présence d'un arc a = (x,y) qui peut être noté simplement xy, on dit que x est l'origine (ou extrémité initiale) et y l'extrémité (terminale) de a, que a est sortant en x et incident en y, et que y est un successeur de x tandis que x est un prédécesseur de y. On dit aussi que x et y sont adjacents.

**Définition**

Un graphe non orienté obtenu à partir d'un graphe orienté *G* en éliminant l'orientation sur les arcs est dénoté graphe correspondant.

**Définition**

Un graphe orienté est simple si les sommets identifiant un arc sont distincts et s'il n'existe pas deux arcs identifiés par la même paire ordonnée de sommets

### 2.2.3. Modes de représentation d'un graphe

L'essor de la théorie des graphes est essentiellement dû à l'avènement de puissants calculateurs. Il est donc légitime de s'intéresser à la manière de représenter les graphes au sein d'un ordinateur. Plusieurs modes de représentation peuvent être envisagés selon la nature des traitements que l'on souhaite appliquer au graphe considéré.

- **Listes de succession:**

Un graphe peut être représenté à l'aide d'un dictionnaire ; il s'agit d'une table à simple entrée ou chaque ligne correspond à un sommet et comporte la liste des successeurs ou des prédécesseurs de ce sommet**.**

- **Matrice d'adjacence**

Les outils classiques d'algèbre linéaire peuvent également être utilisés pour coder les graphes. La première idée consiste à considérer chaque arc comme un lien entre deux sommets.

**Définition**

Considérons un graphe G = (X, A) comportant n sommets. La matrice d'adjacence de G est égale à la matrice $U = (u_{ij})$ de dimension n x n telle que

$$u_{ij} = \begin{cases} 1 & \text{si } (i,j) \in A \text{ (c'est à dire } (i,j) \text{ est une arête)} \\ 0 & \text{sinon} \end{cases}$$

Une telle matrice, ne contenant que des "0" et des "1" est appelée, de manière générale, une matrice booléenne.

Un graphe orienté quelconque à une matrice d'adjacence quelconque, alors qu'un graphe non orienté possède une matrice d'adjacence symétrique. L'absence de boucle se traduit par une diagonale nulle.

Ce mode de représentation engendre des matrices très creuses (i.e. comprenant beaucoup de zéros). Cependant la recherche de chemins ou de chaînes s'effectue aisément avec une telle représentation. De plus, la matrice d'adjacence possède quelques propriétés qui peuvent être exploitées. Considérons un graphe G et sa matrice d'adjacence associée U :

- la somme des éléments de la $i^{ème}$ ligne de U est égale au degré sortant du sommet $x_i$ de G.
- la somme des éléments de la $j^{ème}$ colonne de U est égale au degré entrant du sommet $x_j$ de G.

- **Matrice d'incidence**:

La seconde idée permettant une représentation matricielle d'un graphe exploite la relation d'incidence entre arêtes et sommets.

**Définition**

Considérons un graphe orienté sans boucle G = (X, A) comportant n sommets $x_1, x_2, \ldots, x_n$ et m arêtes $a_1, a_2, \ldots, a_m$. On appelle matrice d'incidence (aux arcs) de G la matrice $M = (m_{ij})$ de dimension n x m telle que :

$$m_{ij} = \begin{cases} 1 & si\ x_i\ est\ l'\text{extrémité}\ intiale\ de\ a_j \\ -1 & si\ x_i\ est\ l'\text{extrémité}\ terminale\ de\ a_j \\ 0 & si\ x_i\ n'est\ pas\ une\ \text{extrémité}\ de\ a_j \end{cases}$$

Pour un graphe non orienté sans boucle, la matrice d'incidence (aux arêtes) est définie par:

$$m_{ij} = \begin{cases} 1\ si\ x_i\ est\ une\ \text{extrémité}\ de\ a_j \\ 0 \qquad\qquad sinon \end{cases}$$

### 2.2.4. Etude de la connexité

- **Chaines et cycles, élémentaires et simples**

**Définition**

Une chaîne est une séquence finie et alternée de sommets et d'arêtes, débutant et finissant par des sommets, telle que chaque arête est incidente avec les sommets qui l'encadre dans la séquence. Le premier et le dernier sommet sont appelés (sommets) extrémités de la chaîne.

La longueur de la chaîne est égale au nombre d'arêtes qui la composent.

Si aucun des sommets composant la séquence n'apparait plus d'une fois, la chaîne est dite chaine élémentaire.

Si aucune des arêtes composant la séquence n'apparaît plus d'une fois, la chaîne est dite chaîne simple.

Un cycle est une chaîne dont les extrémités coïncident.

Un cycle élémentaire (tel que l'on ne rencontre pas deux fois le même sommet en le parcourant) est un cycle minimal pour l'inclusion, c'est-à-dire ne contenant strictement aucun autre cycle.

Un circuit est un chemin dont les extrémités coïncident.

- **Chemins et circuits, élémentaires et simples**

Toutes les définitions précédentes, s'appliquant au cas des graphes non orientés, peuvent être transposées au cas des graphes orientés.

**Définition**

Un chemin est une séquence finie et alternée de sommets et d'arcs, débutant et finissant par des sommets, telle que chaque arc est sortant d'un sommet et incident au sommet suivant dans la séquence (cela correspond à la notion de chaîne "orienté").

Si aucun des sommets composant la séquence n'apparaît plus d'une fois, le chemin est dit chemin élémentaire.

Si aucune des arêtes composant la séquence n'apparaît plus d'une fois, le chemin est dit chemin simple.

Un circuit est un chemin dont les extrémités coïncident.

En parcourant un circuit élémentaire, on ne rencontre pas deux fois le même sommet.

- **Graphes et sous-graphes connexes**

De manière intuitive, la notion de connexité est triviale. Un graphe est connexe si l'on peut atteindre n'importe quel sommet à partir d'un sommet quelconque en parcourant différentes arêtes. De manière plus formelle, on a :

**Définition**

Un graphe G est connexe s'il existe au moins une chaîne entre une paire quelconque de sommets de G.

### 2.2.5. Arbres et arborescences

- **Définitions et propriétés**

**Définition**

Un arbre est un graphe connexe sans cycles.

Un graphe sans cycle qui n'est pas connexe est appelé une forêt (chaque composante connexe est un arbre).

Par définition même, un arbre est donc un graphe simple. On constate également que $\top$ = (X,T) est un arbre si et seulement s'il existe une chaîne et une seule entre deux sommets quelconques.

Etant donné un graphe quelconque G = (X,A), un arbre de G est un graphe partiel connexe et sans cycles. Si ce graphe partiel inclut tous les sommets du graphe G, l'arbre est appelé arbre maximum ou arbre couvrant. Une forêt de G est un graphe partiel sans cycle de G (non nécessairement connexe). Une forêt maximale de G est une forêt de G maximale pour l'inclusion (l'ajout d'une seule arête supplémentaire du graphe à cette foret crée un cycle).

**Définition**

Un graphe G est une arborescence s'il existe un sommet R appelé racine de G tel que, pour tout sommet S de G, il existe un chemin et un seul de R vers S.

La notion d'arborescence couvrante se définit comme celle d'arbre couvrant, mais elle est plus délicate car il faut trouver une racine (qui n'existe pas toujours).

- **Arbres couvrants de poids minimum**

Considérons le problème qui consiste à relier n villes par un réseau câblé de la manière la plus économique possible. On suppose connue la longueur $l_{ij} = l(a_{ij})$ la longueur de câble nécessaire pour relier les villes i et j. Le réseau doit évidemment être connexe et il ne doit pas admettre de cycles pour être de coût minimal ; c'est donc un arbre et ce doit être l'arbre maximum le plus économique.

Le problème à résoudre se pose donc dans les termes suivants :

**Définition**

Soit un graphe non orienté G, connexe, pondéré par une fonction positive $l$ attachée aux arêtes. Soit un arbre couvrant T = (X,B) définit comme graphe partiel de G avec un ensemble d'arêtes B. Son poids (ou coût) total est :

$$l(T) = \sum_{a \in B} l(a) \qquad (14)$$

On dit que T est un arbre couvrant de poids minimal de G si l(T) est minimal parmi les poids de tous les arbres couvrants possibles de G.

On peut montrer que si toutes les arêtes sont de " poids" différents, l'arbre couvrant de poids

minimal est unique. Plusieurs algorithmes ont été proposés pour résoudre ce problème [147].

Dans ce qui suit nous allons présenter quelques algorithmes qui utilisent les graphes dans les systèmes de recommandations.

### 2.2.6. Le filtrage collaboratif basé sur les graphes

De nombreuses applications Web reposent sur l'utilisation de graphes possédant une forte structure de communautés, afin de proposer aux utilisateurs des contenus personnalisés. Par exemple, Facebook recommande à ses utilisateurs de nouveaux contacts, et Amazon indique à ses clients des articles susceptibles de les intéresser.

Ces algorithmes de recommandation s'appuient sur une notion de distance entre les utilisateurs, qui permet de représenter l'influence qu'ils exercent les uns sur les autres. Par exemple, le filtrage collaboratif Horting [171] est une approche basée sur un graphe de relation de similarité (arcs) entre les utilisateurs (noeuds). La notion d'influence se décline sous la contrainte de Horting qui impose de ne considérer que les utilisateurs ayant un grand nombre de mesures communes. Ainsi que la contrainte de prédictabilité ; l'ajout à la notion de Horting une information sur le degré de ressemblance entre deux utilisateurs en se basant sur la distance de Manhattan. Le parcours du graphe permet de filtrer les utilisateurs proches et ceux ayant un nombre de mesures important. La notion de prédictabilité est plus contraignante que le concept de proximité car le système a besoin d'un échantillon suffisamment important de ressources communément mesurées.

L'utilisation des graphes est plus répandue dans le traitement de l'information, l'organisation des données, la modélisation de nombreux types de relations et de la dynamique des processus dans différents systèmes sociaux. Typiquement, le système de recommandation peut se représenter comme un graphe biparti, contient deux ensembles des sommets : ensembles des utilisateurs et autres des ressources. Il y a des approches visant à améliorer les recommandations par l'utilisation des graphes. Par exemple, il y a ceux qui ont considéré le problème de prédiction de liens comme un problème des machines d'apprentissage [172], ils ont montré que la prise en compte de la nature bipartite du graphe peut améliorer les performances des modèles de prévision, cela est obtenu par la projection du graphe bipartie à un graphe unimodal et par l'introduction de nouvelles variantes de mesures topologiques pour mesurer la probabilité de deux nœuds à être connectés. Il y a d'autres dans [173] qui ont proposé une approche pour le lissage des votes. Il s'agit d'un algorithme basé sur un graphe

des ressources. Alors que chaque vote donné par un utilisateur à un ensemble des ressources doit être suffisamment souple. Donc, un coefficient de Smoothness est calculé en se basant sur un graphe des ressources tout en respectant la structure intrinsèque des ressources. Cette méthode peut explorer l'information géométrique des données d'un élément et de faire usage de ces informations pour produire de meilleures recommandations. Une autre méthode présentée par [174] dont l'article utilise l'agrégation des graphes de préférence pour la prédiction de votes collaboratifs. Le principe de cet approche est basé sur l'idée de former un graphe de préférence pour un utilisateur cible en se basant sur les valeurs de votes donnée à un ensemble des ressources pour arriver à construire un graphe de préférences, à partir les graphes de préférences des utilisateurs tout en minimisant les nombres des back-edge dans le graphe global de préférences.

Une tendance récente importante des systèmes traitent du domaine de la sensibilité au contexte et plus généralement à la situation. Entre outre, ces systèmes, crées en employant les arbres conceptuels, dans lesquels les nœuds portent les concepts sémantiques pour chaque dimension de contexte et les arcs définissent les types de ces dimensions. On obtient une composition de plusieurs triplets de contexte (dimension, relation, valeur) en une structure plus expressive. Chaque concept dans l'arbre provient d'une taxonomie correspondante à la dimension, et peut modéliser une partie de la réalité sur un quelconque niveau d'abstraction.

Le but de ce chapitre est de présenter principalement l'importance du clustering et les graphes, ainsi que leurs méthodes les plus utilisées en présentant ses applications dans les systèmes de recommandation.

Dans le chapitre suivant, nous proposons de nouvelles méthodes de recommandation basées sur des techniques de clustering et des graphes, qui donnent une forte structure de communautés et une meilleure recommandation avec et sans calcul de valeurs manquantes.

# *CHAPITRE III*

# NOS CONTRIBUTIONS POUR LES SYSTEMES DE RECOMMANDATION

## Introduction

Ce chapitre présente nos apports sur l'algorithme de filtrage collaboratif. La première partie présente notre nouvel algorithme de filtrage collaboratif, qui intègre une combinaison entre la similarité en se basant aussi bien sur les notes d'items que sur leurs attributs, dans un modèle convexe. Les paramètres de pondération de ce modèle est une fonction de temps qui tient compte de la contribution de ces deux similarités, qui fait décroitre progressivement l'influence des anciennes notes et résoudre le problème de la rareté. Le calcul de la similarité est précédé par la formation du voisinage de l'item cible, la première méthode utilisée est celle de l'approche simple de regroupement k-means, pour avoir des items dans différents groupes. Cette méthode a connu des limites à cause du problème de sparcity. Pour pallier à cette limitation et augmenter la précision de notre modèle, nous avons opté pour une approche systémique issue de la Technologie du Groupe. Cette approche offre des communautés à partir de l'amélioration de l'algorithme BEA. C'est une nouvelle façon d'identifier le voisinage et de résoudre le problème de l'évolutivité permettant par la suite de faire la recommandation. Ensuite, un deuxième type de filtrage collaboratif est présenté, basé cette fois sur la théorie des graphes pour fournir une liste des meilleurs items au lieu de la recommandation d'un seul item, sans calcul de prédiction. Enfin, une méthode pour la classification des mesures des similarités utilisées dans les systèmes de recommandation est présentée.

### 3.1. L'algorithme de Filtrage Collaboratif multicritères

L'intérêt pour les approches basées sur les items est plus récent que celui pour les approches basées sur les utilisateurs. L'approche du filtrage collaboratif basée sur les items donne de bons résultats aussi bien au niveau de la recherche qu'au niveau pratique. Le calcul de la prédiction de la note de l'utilisateur pour l'item cible, est la moyenne des notes déjà faites sur l'item cible pondérées par la similarité des items avec ce dernier.

Toutefois, dans la plupart de ces approches, les données utilisées sont les notes et sont considérées comme statiques, c'est-à-dire, les notes produites à des moments différents sont pondérés d'une façon uniforme. Aussi, seules les notes sont utilisées dans le calcul de la similarité entre items. Dans tout système de recommandation, le nombre de notes déjà obtenu est généralement très faible par rapport au nombre de notes qui doivent être prédites, ce qui est connu sous le nom de problème de sparcity ou la rareté. Le succès et la performance du système de recommandation collaboratif dépend de la disponibilité d'une masse critique des

notes d'items. Une façon de surmonter ce problème de la rareté des notes consiste à utiliser les attributs d'items lors du calcul de la similarité, c.-à-d., deux items peuvent être considérés comme similaires non seulement si ils ont été évalués par les mêmes utilisateurs de la même façon, mais aussi si ils ont les mêmes attributs.

Notre objectif est d'améliorer la performance de l'approche du filtrage collaboratif basée sur les items. Dans cet esprit, nous proposons d'enrichir son processus de la recommandation par d'autres critères.

Tout d'abord, nous regroupons les items par l'algorithme de partitionnement k-means pour former le voisinage de l'item actif, puis dans la phase de prédiction, nous utilisons une combinaison convexe entre la similarité basée sur les notes des items et celle basée sur leurs attributs.

Nous remplaçons la mesure de la similarité $S(p, p_j)$ basée sur les notes des items utilisée dans la formule 8 de la génération de la recommandation de l'item cible $p$ pour l'utilisateur cible $c$ qu'est définie précédemment par la formule 15 :

$$\mathrm{S}(p, p_j) = \lambda\, \mathrm{sim}_{\mathrm{rating}}(p, p_j) + (1-\lambda)\mathrm{sim}_{\mathrm{Attribute}}(p, p_j) \qquad (15)$$

Où $\mathrm{sim}_{\mathrm{rating}}$ désigne la similitude des notes entre les items p et $p_{\mathrm{j}}$, $\mathrm{sim}_{\mathrm{Attribute}}$ dénotes la similitude d'information des attributs des items p et $p_{\mathrm{j}}$ .

Ces deux mesures de similarité sont pondérées par un paramètre $\lambda$ qui est entre 0 et 1. Lorsque $\lambda = 0$, on se trouve dans le cas de la similarité basée sur les attributs des items. Alors que quand $\lambda = 1$, c'est dans le cas de la similarité basée sur les notes [33].

Evidemment, La similarité $\mathrm{sim}_{\mathrm{rating}}$ qui se base sur les notes des items joue un rôle principal dans l'algorithme de filtrage collaboratif, toutefois, lorsque le nombre d'utilisateurs qui co-évaluent les items $p_i et\, p_j$ par les utilisateurs est trop peu, la $\mathrm{sim}_{\mathrm{rating}}$ n'est pas précise. L'ajout des autres données sur les items consiste à rendre la similarité S plus précise.

Pour tenir compte de l'avantage de ces deux mesures de similarité et que les notes produites à des moments différents ne soient pas pondérées également lors de la phase de prédiction de la préférence de l'utilisateur, nous estimons que le paramètre $\lambda$ dépend du temps, comme il est décrit dans la formule 16 :

$$\lambda = \mathrm{e}^{-\alpha.\Delta \mathrm{t}} \text{ avec } \alpha = 1/T_0 \quad \text{et} \quad \Delta t = t - t_i \qquad (16)$$

λ désigne la fonction exponentielle, $T_0$ désigne le paramètre qui contrôle le taux de décroissance spécifique de l'utilisateur et t, $t_i$ désignent le moment où l'évaluation de l'utilisateur courant et de l'utilisateur i, est générée.

Ce paramètre λ ajuste la contribution de la similarité basée sur les notes et celle basée sur les attributs des items pour prévoir automatiquement la préférence de l'utilisateur.

Lorsque le moment de la génération de l'évaluation de l'utilisateur est ancien, λ tend vers 0, ce qui donne l'importance à la similarité basée sur les attributs. Dans le cas où le moment de la génération de l'évaluation est récent, λ tend vers 1, ce qui donne l'importance à la similarité basée sur les notes. Le choix de la fonction exponentielle nous permet aussi de réduire l'influence des anciennes notes.

Intuitivement, nous pouvons supposer que les notes récentes correspondent au dernier intérêt d'utilisateurs, un item qui a été évalué récemment devrait avoir un impact plus important sur la prédiction de leur comportement futur qu'un élément qui a été évalué il y a longtemps. Donc, nous soutenons que plus les données sont récentes, plus elles contribuent à bien prédire les items.

D'après l'équation (16), α définit la vitesse de décroissance de la courbe de la fonction exponentielle et il est inversement proportionnel à $T_0$. Le choix de $\alpha = 1/T_0$ permet de décider le taux de décroissance de l'intérêt de l'utilisateur.
Le taux de décroissance des notes est déterminé par la façon dont l'intérêt d'utilisateur sur les items change. Nous analysons le paramètre $T_0$ pour définir le taux de décroissance du poids attribué à chaque note. Nous faisons varier le paramètre $T_0$ comme il est exprimé par l'équation (16), les valeurs attribuées sont 50, 100 et 200. Les résultats sont illustrés dans la figure (3).

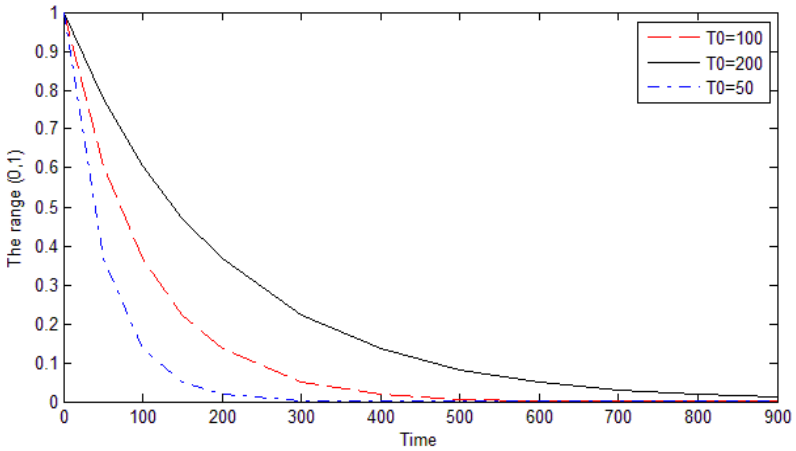


Figure 3: Les courbes de la fonction de temps en utilisant différents $\boldsymbol{T_0}$

Lorsque le paramètre $T_0$ est petit et l'intérêt de l'utilisateur change fréquemment, la vitesse de décroissance de la courbe est rapide, nous pouvons dire que les notes récentes contribuent mieux dans la prédiction. Contrairement, si $T_0$ est grand et l'intérêt de l'utilisateur persiste, la vitesse de décroissance de la courbe devient lente, dans ce cas nous pouvons dire que les anciennes notes peuvent aider à améliorer la précision de la prédiction de la préférence future de l'utilisateur.

Le $T_0$ contrôle le taux de décroissance, et par conséquent nous donne une information sur l'importance de données historiques.

Donc, nous devrions sélectionner le $T_0$ approprié pour les items afin que ceux notés récemment peuvent mieux contribuer dans la recommandation. Puisque l'intérêt de l'utilisateur est sensible au temps, et son avis sur un item peut changer fréquemment, il n'est pas possible de fournir une valeur correspondance du $T_0$ pour chaque utilisateur et chaque item. Nous supposons que le même utilisateur a les mêmes préférences et changements d'intérêt pour les items similaires. Nous utilisons donc l'approche simple de regroupement k-means pour discriminer entre les différents items.

Ensuite, pour chaque groupe d'items, nous sélectionnons automatiquement la valeur du paramètre $T_0$. En d'autres termes, à chaque fois, nous prenons un item noté par un utilisateur à partir d'un cluster d'items qui sera utilisé pour mesurer à quel point le comportement d'utilisateur sur l'item, peut être expliqué. Pour cela, nous introduisons l'erreur absolue moyenne (Mean Absolute Error, MAE) qui calcule l'écart absolu moyen de recommandations, de leurs vraies valeurs spécifiées de l'utilisateur, son expression est comme suit:

$$MAE = \frac{\sum_{i=1}^{n} |u_i - q_i|}{n} \quad (17)$$

Où n représente le nombre des notes d'utilisateur dans le cluster. $q_i$ est la valeur réelle de la note de le $i^{ème}$ item. $u_i$ est la prédiction de la note pour le $i^{ème}$ item

Pour chaque utilisateur, la prédiction $u_i$ pour chaque item est paramétrée par $T_0$. Nous utilisons donc MAE($T_0$) à souligner que le MAE est paramétré par $T_0$.

En minimisant la valeur de MAE, nous trouverons le paramètre optimal $T_0$. Formellement, le problème d'optimisation est indiqué comme suit:

$$\text{argmin MAE}(T_0) \quad (18)$$

Tout d'abord, nous réalisons une exploration du paramètre dans un intervalle, ensuite, nous analysons toutes les valeurs possibles et nous sélectionnons la valeur optimale attribuée au paramètre $T_0$ qui minimise le MAE. A titre illustratif, la figure 4 présente cette valeur optimale.

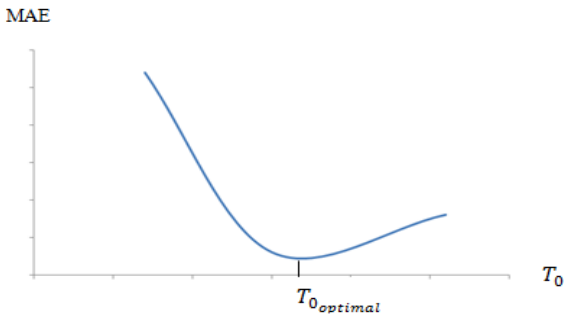


Figure 4: Le paramètre $\boldsymbol{T_0}$ qui minimise MAE

Par cette méthode, nous pouvons avoir approximativement la valeur optimale personnalisé de $T_0$ pour chaque utilisateur.

En résumant, notre algorithme de FC basé sur les items se déroule en ces étapes:

1. Un regroupement des items par l'algorithme de partitionnement k-means
2. La sélection du centre du cluster proche de l'item actif
3. La formation de voisinage de l'item actif dans le cluster par la stratégie de sélection des plus proches items en utilisant la mesure de la similarité Pearson Correlation 1.3, entre l'item actif et ces derniers.
4. La génération de recommandation par la formule 8 en intégrant la nouvelle formule 15 de la similarité améliorée que nous venons de la décrire

Nous avons jusqu'à ici présenté notre approche de filtrage collaboratif qui intègre à la fois les évaluations sur les items et leurs caractéristiques dans un modèle convexe, et dont le paramètre de pondération est une fonction de temps qui tient compte de l'avantage de contribution des mesures de similarité de ces données et qui fait décroitre progressivement l'importance des anciennes données ; autrement dit les évaluations récentes de l'utilisateur reflètent et contribuent mieux à avoir les préférences futures de l'utilisateur.

La première étape de notre méthode proposée [179] est d'utiliser le regroupement par l'approche simple de regroupement k-means pour former le voisinage à partir des groupes. Cette méthode a pu résoudre le problème de l'évolutivité, mais elle présente des limitations à cause du problème de la rareté.

Pour augmenter la précision de notre algorithme de FC, dans la partie qui suit, nous allons présenter une nouvelle version améliorée de l'algorithme de maximum d'énergie (Bond Energy Algorithm, BEA) [179]. Cette approche donne une nouvelle façon d'identifier le voisinage et surmonter le problème de la rareté.

### 3.2. Le co-clustering pour l'obtention des communautés

Notre objectif est de proposer une méthode de regroupement basée sur BEA pour notre algorithme de filtrage collaboratif, qui permet de réaliser une classification simultanée des utilisateurs et des items, permettant d'extraire, tous les clusters, correspondants aux différentes communautés. Ces clusters connaîtront une forte association entre les utilisateurs et les items, assurant une énergie maximale et minimisant le problème de la rareté. Tout d'abord, nous présentons l'algorithme BEA, ainsi notre nouvelle version améliorée de l'algorithme BEA [179].

#### 3.2.1. **L'algorithme de maximum d'énergie** :

La taille grandissante des réseaux nous oblige à chercher des méthodes capables de faciliter leur gestion. Ce besoin implique la recherche des méthodes du co-clustering pour structurer ces réseaux sous formes de groupes ayant des caractéristiques communes.

De nos jours, la classification croisée ou co-clustering est utilisée dans des domaines différents. Dans le domaine industriel, elle est nommée la Technologie de Groupe. C'est un concept basé sur l'identification et l'exploitation des ressemblances ou la similarité entre les produits et les processus de conception et de fabrication en vue de rationaliser la production

et de diminuer les coûts industriels [137]. Dans ce sens, nous allons présenter l'algorithme de co-classification, BEA, qui est adapté pour un co-clustering d'utilisateurs et d'items. C'est une généralisation des algorithmes de co-classification.

Le principe de l'algorithme BEA est de réaliser une co-classification sur une matrice creuse afin d'identifier les groupes d'objets moyennant des permutations des colonnes et des lignes de la matrice d'usage. Il cherche aussi à afficher et découvrir les associations et les interrelations qui existent entre les groupes. L'algorithme se base sur la liaison entre un élément de la matrice d'usage A et ces quatre proches voisins, L'élément en question est $U_{22}$ tel qu'il est illustré dans la table 4. D'après McCormick [138], ces liaisons peuvent être considérées comme une énergie.

| | $c_1$ | $c_2$ | $c_3$ | | ⋯ | $c_M$ |
|---|---|---|---|---|---|---|
| $p_1$ | $U_{11}$ | $U_{12}$ | $U_{13}$ | ⋯ | ⋯ | $U_{1M}$ |
| $p_2$ | $U_{21}$ | $U_{22}$ | $U_{23}$ | ⋯ | ⋯ | $U_{2M}$ |
| $p_3$ | $U_{31}$ | $U_{32}$ | $U_{33}$ | ⋯ | ⋯ | $U_{2M}$ |
| : | : | : | : | : | : | : |
| $p_N$ | $U_{N1}$ | $U_{N2}$ | $U_{N2}$ | ⋯ | ⋯ | $U_{NM}$ |

Table 4: Energie de liaison d'un élément par ses quatre proches voisins

En tenant compte de l'énergie frontalière calculée, la permutation des lignes et des colonnes se fait à fin de rassembler les éléments de la matrice pour créer des regroupements qui maximisent l'énergie. La permutation est basée sur la valeur du coefficient d'Energie Maximum (EM), comme suit :

$$Energie\ (U_{22}) = U_{22} * (U_{12} + U_{21} + U_{23} + U_{32}) \quad avec\ U_{ij} = \{1:5\} \qquad (19)$$

D'une manière générale, la Mesure d'Efficacité, (ME) d'une matrice A est la somme des forces de liaison de la matrice, où la force de liaison entre deux plus proches éléments est définie comme leur produit. La ME, est alors donnée par (avec la convention que $U_{0,j} = U_{M+1,j} = U_{i,0} = U_{i,N+1} = 0$) :

$$ME(A) = \sum_{i=1}^{M}\sum_{j=1}^{N} U_{ij}(U_{i,j+1} + U_{i,j-1} + U_{i+1,j} + U_{i-1,j}) \qquad (20)$$

où A est une matrice non négative de dimension NxM.

Maximiser la ME par les permutations des lignes et des colonnes sert à créer des énergies solides de liaison en entraînant les grands éléments d'une matrice d'être ensemble. La ME définit par l'équation 20 présente des avantages théoriques et computationnels très importants : D'une part, la ME est applicable à des réseaux de toute taille et forme, la seule exigence est que les éléments du tableau doivent être non négatifs. D'autre part, les liaisons verticales (horizontales) ne sont pas affectées par la permutation des colonnes (lignes). Par conséquent, l'optimisation de la ME peut être obtenue exactement par une recherche de la permutation optimale de la colonne (ligne). La contribution de la ME pour n'importe quelle colonne (ou ligne) est uniquement influencée par les deux colonnes adjacentes (ou lignes).

Comme indiqué précédemment, l'algorithme de maximum d'énergie vise à maximiser la somme de l'énergie de liaison sur toutes ces permutations des lignes et des colonnes de la matrice d'entrée A. Autrement dit, cherchons à maximiser

$$\max\left\{\sum_{i=1}^{M}\sum_{j=1}^{N} U_{ij}\left(U_{i,j+1}+U_{i,j-1}+U_{i+1,j}+U_{i-1,j}\right)\right\} \qquad (21)$$

où la maximisation est prise sur N!xM! matrices possibles, qui peuvent être obtenus à partir de la matrice d'entrée, par permutations des lignes et par colonnes. Ce problème, comme indiqué précédemment dans [139], se réduit à deux optimisations distinctes ; la première pour les lignes et la deuxième pour les colonnes. Etant donné que les problèmes sont-équivalents, seul le premier doit être discuté.

Soit II = {r (1), r(2), ...,r (N)} la permutation des N colonnes de la matrice A, produisant la nouvelle matrice $B = [b_{i,j}] = [U_{i,r(j)}]$. La maximisation de la somme de la liaison des lignes est donnée par (avec la convention $b_{i,0} = b_{i,N+1} = 0$)

$$\max_{r}\left\{\sum_{i=1}^{M}\sum_{j=1}^{N} b_{ij}\left(b_{i,j-1}+b_{i,j+1}\right)\right\} \qquad (22)$$

où II va sur tous les N! permutations possibles.

L'algorithme BEA, décrit dans [139] est le suivant :

| |
|---|
| Etape 1: Placer l'une des colonnes arbitrairement. Régler i = 1.<br>Etape 2: Placer individuellement chacun des restants N-i colonnes dans chacune des i +1 positions possibles (à gauche et à droite des colonnes i déjà placés), et de calculer la contribution de chaque colonne pour la ME. Placer la colonne qui donne la plus grande contribution supplémentaire à la ME dans son meilleur emplacement. Incrémenter i de 1 et répéter jusqu'à ce que i = N.<br>Etape 3: Lorsque toutes les colonnes sont placées, répétez la même procédure sur les lignes. |

Table 5: L'algorithme BEA

Cet algorithme a plusieurs caractéristiques importantes:

- Le temps de calcul ne dépend que de la taille de la matrice. Pour une matrice de dimension MxN, le nombre d'opérations est de $[M^2.N + N^2.M]/2$ .
- Le classement final obtenu à l'aide de l'algorithme est indépendant de l'ordre dans lequel les lignes (colonnes) sont présentées, mais, il ne dépend que de la ligne initiale (colonne). Les regroupements finaux, sont insensibles à l'initialisation et leurs ME associées sont numériquement proches.
- L'algorithme donnera à partir de la matrice d'entrée, une matrice de sortie sous forme de blocs purs (où certains blocs sans intersection, ou bien en forme de blocs de checkerboard (Figure 6). Dans le cas de la forme checkerboard, les blocs d'éléments non nuls de la matrice sur la diagonale principale ne représentent que des groupes primaires de ligne et de colonne des variables et des blocs non diagonaux indiquent les relations entre les groupes.

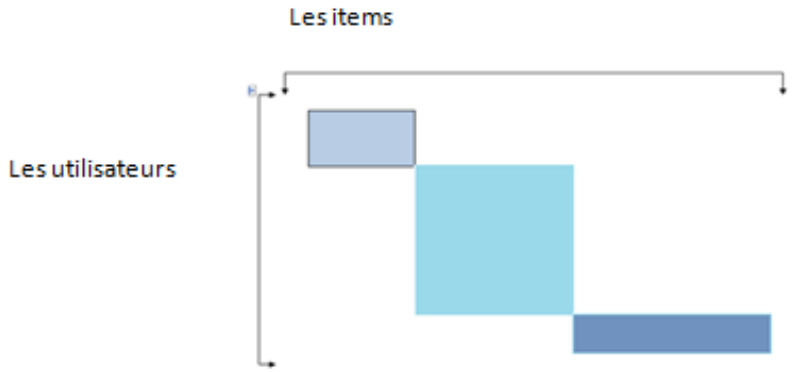


Figure 5: La forme du bloc diagonal

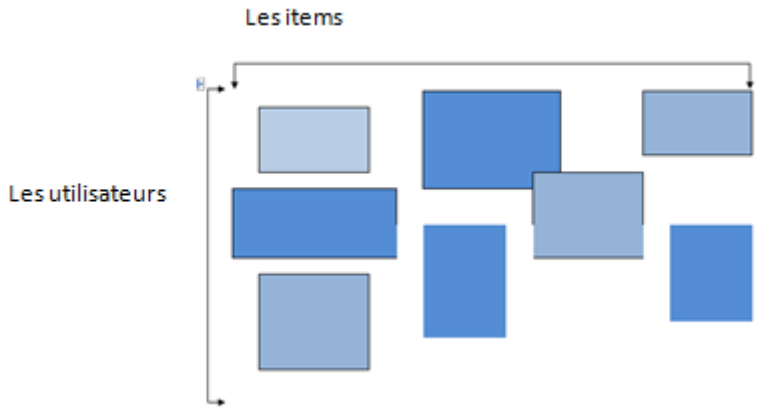


Figure 6: La forme checkerboard des blocs.

- Dans le cas de la forme checkerboard, les blocs d'éléments non nuls de la matrice sur la diagonale principale ne représentent que des groupes primaires de ligne et de colonne des variables et des blocs non diagonaux indiquent les relations entre les groupes.

### 3.2.2. La version de BEA proposée :

Notre système de recommandation sera basé sur ce concept, et permettra d'obtenir des groupes d'utilisateurs ayant une certaine ressemblance, ou ayant des votes similaires sur les items. L'algorithme BEA permet de réaliser la classification croisée (co-clustering) sur des matrices creuses donnant ainsi des classes homogènes.

Reste à discuter le problème de la détection et l'extraction de ces communautés. BEA donne des blocs détectables visuellement et ne fournit pas une extraction automatique, spécialement, pour les grandes bases de données, ce qui implique la recherche d'une nouvelle méthode pour effectuer une extraction précise et automatique après la formation de blocs naturels par BEA [139].

Notre solution proposée pour la détection automatique de ces blocs suit les étapes suivantes : Après l'application de la BEA sur la matrice d'incidence Utilisateurs x Items de dimension (N x M), nous proposons de faire une seconde étape d'une deuxième réorganisation basée sur le poids calculé pour chaque ligne (utilisateur $c_i$) et chaque colonne (item $p_j$)

$$D(c_i) = \sum_{j=1}^{M} U_{ij} * w_i \quad \text{avec } w_i = 2^{M-j} \qquad (23)$$

On considère que ce poids est associé à chaque objet, que ce soit l'utilisateur ou l'item, il sera utilisé sur tous les utilisateurs et tous les éléments qui sont représentés par leurs poids.

L'utilité de ce poids apparaît lorsque nous décidons de les réarranger, dans l'ordre décroissant. En effet, le tri de ces poids de lignes et de colonnes, essaye de transformer la matrice de résultat à partir de BEA, sous une forme diagonale. Nous projetons par la suite, l'ensemble de poids d'utilisateurs sur un axe et celui d'items sur un autre axe, de manière à différencier les catégories d'utilisateurs et d'items.

Ensuite, l'approche proposée tente de rechercher de partitions des utilisateurs et des items, en se basant sur l'Étendue Mobile de ces poids. Le calcul de l'Étendue Mobile d'utilisateurs/d'items, est exprimé par la différence de poids des deux éléments ordonnés successivement. Sur cette base, nous pouvons facilement détecter les clusters d'utilisateurs ou items.

En résumant, les étapes de notre méthode sont comme suit :

| Notre méthode permet d'extraire les blocs issus de l'algorithme de BEA |
|---|
| Etape 1 : Calculer le poids pour chaque ligne dans la formule (23). |
| Etape 2 : Trier les poids de ces lignes (les utilisateurs) par ordre décroissant. |
| Etape 3 : Faire une réorganisation des lignes en se basant sur les poids triés |
| Etape 4 : Calculer l'étendue mobile pour chaque poids des deux lignes successives. |
| Etape 5 : Projeter l'ensemble de poids d'utilisateurs sur un axe pour trouver Les clusters. |
| Etape 6 : Découper les clusters selon la plus grande valeur de l'étendue |

Table 6: Les étapes de l'extraction des blocs issus à partir BEA

Jusqu'à présent, nous avons présenté la nouvelle méthode d'extraction diagonale des blocs issus de l'algorithme BEA. Dans le cas où nous obtiendrons les communautés non diagonales, nous appliquerons une méthode qui utilise les graphes pour l'extraction de régions denses dans les contextes bruités [150]. Après avoir obtenu les blocs non diagonaux par cette approche, nous pouvons proposer une solution pour le problème du démarrage à froid, basée sur le concept d'utilisateurs clés ou d'items clés, qui sera traitée dans la section suivante.

### 3.2.3. Utilisateurs clés et le problème de démarrage à froid

L'objectif de notre recherche consiste à aider le nouvel utilisateur du système à trouver l'information qui l'intéresse. Le comportement naturel humain consiste à se renseigner auprès

d'autres personnes ayant une expérience. Parmi les utilisateurs de notre système de recommandation, nous distinguons ceux qui sont considérés comme des utilisateurs clés ; il s'agit d'un groupe d'utilisateurs capable de représenter une solution au problème de démarrage à froid et guider un nouvel utilisateur vers l'information pertinente. L'intérêt d'un tel soutien apparaît clairement dans le cas où l'utilisateur n'a pas encore rencontré des items et à propos desquels il ne dispose encore d'aucune expérience.

Les principales caractéristiques d'un utilisateur clé sont :

- un nombre maximum d'items évalués.
- Appartenir à plusieurs communautés.

Ces utilisateurs clés sont facilement détectables à partir de la matrice finale obtenue après l'application de BEA, à titre illustratif, la figure 7 présente deux utilisateurs clés comme solution pour le démarrage à froid.

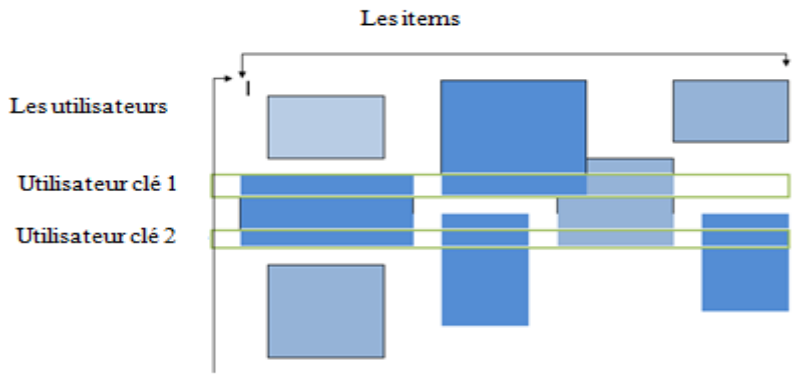


Figure 7: Les utilisateurs clés

En conclusion, notre solution proposée semble plus efficace dans le cas de données dispersées et donnent une co-classification naturelle. Elle est considérée comme une extension de l'algorithme de co-clustering BEA issue de TG. Dans un premier lieu, elle permet d'obtenir des blocs avec une forte association entre les utilisateurs et les items, assurant une énergie maximale. Cela a pour but surmonter le problème de l'évolutivité et de minimiser le problème de la rareté dans ces blocs. Dans un second lieu, elle permet de faire une classification avec un mécanisme d'adaptation de poids lors de l'extraction de blocs.

Dans la section suivante, notre troisième apport correspond à proposer une nouvelle méthode pour un autre type de recommandation. Il s'agit de recommander une liste d'items à l'utilisateur au lieu de lui recommander un seul item. Cette méthode est basée sur un algorithme de la théorie des graphes et sans calcul de prédiction [177].

### 3.3. La recommandation d'une liste d'items

Pour recommander une liste d'items à un utilisateur actif, nous proposons une méthode de recommandation basée sur la théorie des graphes. Cette dernière prédit les préférences de l'utilisateur sans calculer la prédiction par notes. Pour extraire une liste de suggestions, les algorithmes utilisent la notion de la similarité, cette dernière a pour but de donner une valeur à la ressemblance entre deux objets. En se basant sur la matrice d'usage, nous avons deux matrices de similarité, entre les items et les utilisateurs respectivement. Ensuite, pour classifier les utilisateurs ou les items sous forme d'un arbre recouvrant de poids minimum, pour chacune des deux matrices représentées par un graphe connexe pondéré, nous sommes basés sur l'algorithme de kruskal [147]. Finalement, la méthode proposée se base sur ces deux arbres pour recommander une liste des meilleurs items. En effet, à partir de la matrice des votes, nous calculons la matrice de similarité entre les utilisateurs ainsi qu'entre les items, nous obtenons deux matrices carrées d'ordre N et M respectivement. Ces deux matrices ayant l'importance au niveau de la détermination des familles des items et des utilisateurs similaires. Pour ce faire, nous calculons d'abord la similarité Cosine entre deux utilisateurs $c_i, c_j$ ou entre deux items $p_i, p_j$ en utilisant la formule décrite dans l'équation 2 du deuxième chapitre. Ensuite, une transformation en dissimilarité est réalisée comme suit:

$$\mathrm{dis}(c_i, c_j) = 1 - \mathrm{cosine}(c_i, c_j) \qquad (24)$$

Les deux matrices de dissimilarité se basent sur la formule (24). Nous réalisons une représentation graphique pour les deux matrices en prenant comme poids des arêtes les valeurs des coefficients de dissimilarité. De plus, notre approche est basée sur la dissimilarité entre les utilisateurs ainsi entre les ressources, ce qui implique l'utilisation de la notion de co-dissimilarité. Nous utiliserons ces graphes pour extraire deux arbres de poids minimaux correspondant aux utilisateurs et aux items. À partir des graphes correspondant à ces deux matrices de dissimilarité d'utilisateurs et d'items, nous construisons deux arbres recouvrant des poids minimaux en utilisant l'algorithme de Kruskal [147] décrit dans la table 6.

| |
|---|
| L'algorithme de Kruskal permet de trouver un arbre de recouvrement de valeur minimale d'un graphe G=(X,U)<br>Etape1 : Trier les arêtes de G par valeur croissante, poser $T = \emptyset$<br>Etape2 : Pour chaque arête (x,y) et par valeur croissante faire :<br>Si $T \cup \{(x,y)\}$ est sans cycle alors<br>Ajouter (x,y) à T : $T = T \cup (x,y)$ |

Table 7: L'algorithme Kruskal [147]

L'arbre représenté sera obtenu par élimination des arêtes ayant un poids élevé. L'arbre des utilisateurs ou des items garde les arêtes ayant les valeurs minimales des coefficients de dissimilarité.

### 3.3.1. Arbre des utilisateurs

L'arbre $\mathrm{Ar}_{\mathrm{users}}$ représentant la matrice de dissimilarité des utilisateurs est le suivant :

Soit G (X,T) l'arbre des utilisateurs, où X représente l'ensemble des sommets d'utilisateurs et T l'ensemble des arêtes ayant comme poids le coefficient de dissimilarité entre les utilisateurs.

Cet arbre représente l'ensemble des utilisateurs classifiés en fonction de dissimilarité minimale.

Soit C l'ensemble de n utilisateurs $\{C_1, C_2, \ldots\ldots, C_n\}$, à travers $\mathrm{Ar}_{\mathrm{users}}$ , pour chaque utilisateur $C_i$ de C. Nous cherchons $S_{C_i}$ : l'ensemble des utilisateurs voisins qui sont similaires à l'utilisateur $C_i$ . Pour chaque utilisateur $C_j$ de l'ensemble $S_{C_j}$, nous déterminons un ensemble $E_{C_j}$ des items ayant des votes élevés. Enfin, nous allons avoir un ensemble $E_{S_{C_i}}$ regroupant tous les items des utilisateurs de l'ensemble $S_{C_i}$ décrit dans la formule suivante :

$$E_{S_{C_i}} = \coprod_{\mathrm{j}=1}^{\mathrm{n}} \mathrm{E}_{\mathrm{C_j}} \qquad (25)$$

### 3.3.2. Arbre des items

Par même principe pour l'arbre des items $\mathrm{Ar}_{\mathrm{items}}$ , nous allons passer d'un graphe simple où les sommets sont les items et les arêtes représentent les valeurs de coefficient de dissimilarité vers un arbre couvrant de poids minimal.

Soit P l'ensemble de m items $\{P_1, P_2, \ldots\ldots, P_m\}$ de notre système de recommandation, pour chaque utilisateur $C_i$ , nous cherchons $E'_{C_i}$ l'ensemble des items de P qui sont similaires aux éléments de l'ensemble $E_{C_i}$ à partir l'arbre d'items $\mathrm{Ar}_{\mathrm{items}}$ .

### 3.3.3. Résultat de la recommandation

L'intersection des ensembles $E'_{C_i}$ et $E_{S_{C_i}}$ donne un nouvel ensemble des items similaires à ceux de l'ensemble $E_{C_i}$, ces derniers sont recommandés par les utilisateurs de $S_{C_i}$, qui sont considérés comme les plus similaires à $C_i$.

En prenant un exemple explicatif de l'application de cette méthode de recommandation à l'utilisateur $C_5$.

Une fois nous avons les deux arbres recouvrant de poids minimum présentés dans la figure 8, nous passerons à la phase de recommandation d'une liste d'items selon notre démarche proposée.

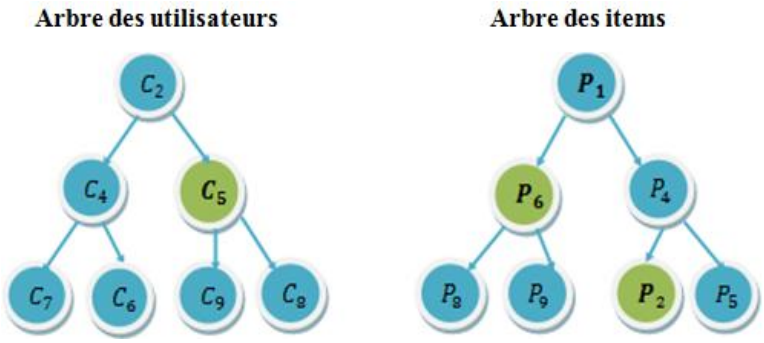


Figure 8: Les arbres des items et des utilisateurs

En prenant $P = \{P_1, P_2, P_4, P_5, P_6, P_8, P_9\}$ et $C = \{C_2, C_4, C_5, C_6, C_7, C_8, C_9\}$.

La figure 9 illustre cette démarche, l'utilisateur $C_5$ a donné une note élevée aux items $\{P_2, P_6\}$ ce qui construit l'ensemble $E_{C_5}$. $S_{C_5} = \{C_2, C_8, C_9\}$, l'ensemble des utilisateurs voisins qui sont similaires à un utilisateur donné $C_5$.

Nous déterminons $E_{S_{C_5}} = \{E_{c_2}, E_{c_8}, E_{c_9}\} = \{P_1, P_2, P_6, P_8, P_9\}$ , l'ensemble d'items mesurés par $S_{C_5}$ l'ensemble et qui est déterminé à travers l'arbre des utilisateurs $Ar_{users}$ .

Et nous déterminons aussi $E'_{c_5} = \{P_1, P_4, P_8, P_9\}$ l'ensemble des items appartenant à P et qui sont similaires aux éléments de $E_{C_5}$, à déterminer à partir l'arbre d'items $Ar_{items}$

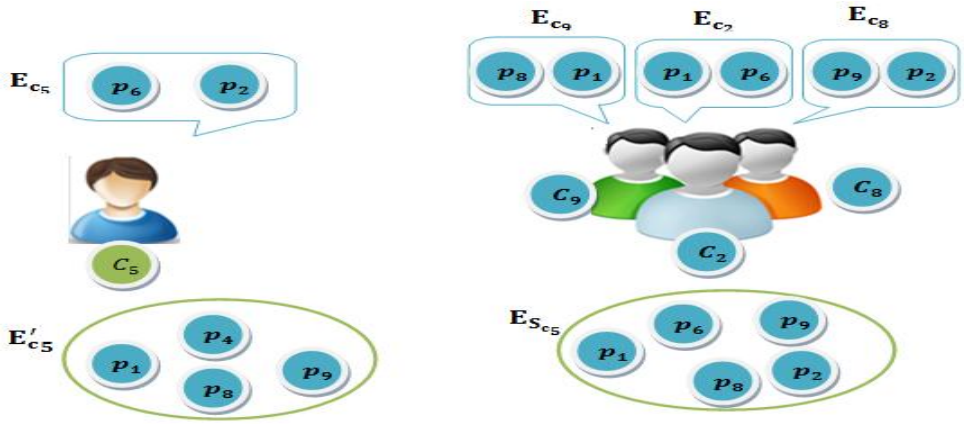


Figure 9 : Le schéma explicatif de la procédure de la recommandation d'une liste des items

L'intersection des ensembles $E'_{C_5}$ et $E_{S_{C_5}}$ donne l'ensemble $\{P_1, P_8, P_9\}$ des items qui sont similaires à ceux de l'ensemble $E_{C_5}$ (l'ensemble des items ayant des notes élevées par $C_5$),

ainsi qu'ils sont déjà recommandées par les utilisateurs qui sont considérés comme les plus proches et les plus similaires à $C_5$, comme il est illustré dans la figure 10.

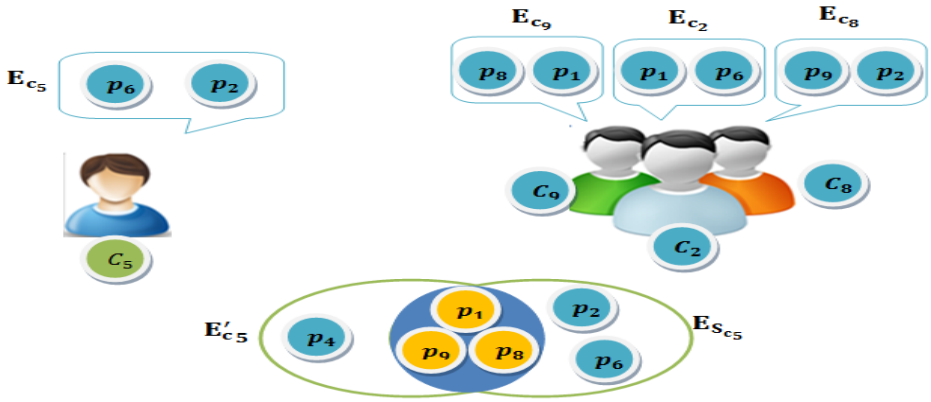


Figure 10: Le schéma explicatif de la procédure de la recommandation d'une liste des items

Dans le cas d'une intersection vide, nous allons classifier les items de l'union des ensembles $E'_C$et $E_{S_C}$ en se basant sur la valeur moyenne des votes des items, et nous recommandons ceux ayant les moyennes élevées.

En résumant, les étapes de notre procédure sont présentées dans la table suivante :

1. Construire deux arbres de recouvrement de poids minimum de dissimilarité minimale des utilisateurs et des items par l'algorithme de Kruskal
2. Pour chaque utilisateur $C_i$, déterminer l'ensemble $E_{S_{C_i}}$ décrit par la formule 25, à partir l'arbre d'utilisateurs $\mathrm{Ar}_{\mathrm{users}}$
3. Pour chaque utilisateur $C_i$, nous cherchons $E'_{C_i}$ l'ensemble des items de P qui sont similaires aux éléments de l'ensemble $E_{C_i}$ à partir l'arbre d'items $\mathrm{Ar}_{\mathrm{items}}$
4. L'intersection des ensembles $E_{S_{C_i}}$ et $E'_{C_i}$ constitue la liste à recommander.

Table 8: Les étapes de la procédure de la recommandation d'une liste d'items

Dans l'article [177], nous avons présenté le filtrage collaboratif pour ce type de recommandation qui s'appuie sur les graphes et qui a pour but de recommander une liste d'items sans calcul de prédiction.

Finalement, il nous semble intéressant de comparer les indices de similarité utilisés dans les systèmes de recommandation, surtout que nous avons remarqué que le choix de ces indices

est arbitraire et il n'y a pas une étude qui a été faite pour les comparer comme nous le proposons dans la section suivante. Dans [178], nous proposons une nouvelle procédure de comparaison des indices de similarité appliqués dans le domaine de l'industrie utilisables aussi dans les systèmes de recommandation. Le but de cette procédure est de classer ces indices de similarité en familles.

### 3.4. La comparaison des indices de similarité

Notre comparaison sera fondée sur les structures de dendrogrammes des indices de similarité obtenus à partir de leurs matrices de ressemblance et par conséquent le résultat sera un dendrogramme regroupant ces indices en familles. Le but est de connaitre les indices de similarité les plus proches.

Différentes approches qui ont été proposées, comparent les clusters, en reposant sur plusieurs indicateurs de performance [175]. En revanche, pour comparer deux classifications, notre procédure s'appuie sur une distance entre structure dendrogrammes. Il s'agit de trouver les dendrogrammes considérés comme des hypergraphes par la classification hiérarchique, et de les comparer à partir de la distance Marczewski-Steinhaus [151].

Beaucoup des travaux ont été consacrés à la présentation et à la définition des différents indices qui nous paraissent importants dans notre propos, comparent des classes résultantes directement. Quand on dispose de deux partitions effectuées sur les mêmes données, il faut savoir si elles sont en accord ou bien si elles diffèrent significativement. Une manière d'aborder ce problème consiste à calculer un indice de concordance entre partitions et à définir une valeur critique à partir de laquelle on conclura que les deux partitions sont ou non concordantes. La plupart des indices sont présentés en formulations relationnelles en utilisant les formules de passages proposées par Kendall [152] et Marcotorchino [153]. A l'indice bien connu de Rand et celui corrigé par Hubert [154], une version asymétrique de Rand [155] a été proposée et utilisée pour la comparaison de partitions emboîtées, avec des nombres différents de classes. Deux autres indices inspirés de test de Mac Nemar et de l'indice de Jaccard. L'indice de corrélation vectorielle introduit par P. Robert et Y. Escoufier [156] qui se révèle identique au coefficient de S. Janson et J. Vegelius [157], le coefficient kappa de Cohen [158], l'indice de redondance proposé Stewart et Love [159], ainsi que l'indice de Popping [160].

La proposition d'une comparaison qui s'appuie sur les dendrogrammes et non pas sur la comparaison des classes résultantes directement, a pour but de fournir un outil rationnel et

efficace pour le regroupement des différentes méthodes de classification afin de les situer les unes par rapport aux autres. Cette proposition sera une démarche générale incluant le choix de méthodes les plus proches et celui des familles de remplacement.

Notre procédure de comparaison des méthodes de classification consiste à réaliser un dendrogramme issue de la matrice des distances entre toutes les arborescences de ces méthodes de classification. Pour ce faire, nous utilisons un algorithme de classification ascendante hiérarchique en tenant compte du critère du lien moyen. Ensuite, à partir de l'arbre de classification de ces méthodes et en procédant à des coupures dans l'arbre, nous obtenons un certain nombre de familles. Le choix du seuil de découpage dépend des critères retenus (nombre de méthodes de classification de remplacement, les partitions les plus proches d'une partition donnée, indice de Ward) qui conditionnent nos objectifs visés. Cette méthode présente l'avantage de fournir une vue globale sur des familles cohérentes de méthodes.

Dans ce qui suit, nous allons présenter le cadre théorique de la distance entre arborescences [151] avant la présentation d'un exemple explicatif.

### 3.4.1. Cadre théorique :

- **Hypergraphes générés par arborescences**

Selon [151], Les arbres traités sont un cas particulier d'hypergraphes générés par arborescence dont la famille des nœuds possède des propriétés spéciales.

Soit X=$\{x_1, x_1, x_2, x_3, \dots, x_n\}$ l'ensemble des sommets terminaux d'une arborescence.

$d^-(x_i) = 1$ , $d^+(x_i) = 0$ pour tout élément de X où $d^-(x_i)$ et $d^+(x_i)$ représentent les demis-degrés intérieurs et extérieurs du nœud $x_i$ respectivement. Soit $\mathcal{A}$ la classe de toutes les arborescences avec X l'ensemble des sommets terminaux. Soit $A \in \mathcal{A}$ représentée par l'hypergraphe $(X, E_A)$ où la classe des arrêtes $E_A$ est définie comme suit: chaque $v \notin$X (c.à.d. chaque nœud non terminal dans l'arborescence génère $d^+(v) - 1$ arêtes dans $E_A$ . Une telle arête consiste en ces éléments de X qui sont des nœuds terminaux de la sous arborescence générée par $v$ et qui est obtenue en considérant $v$ comme une racine c'est-à-dire en supposant que $d^-(v) = 0$.

Notre méthode de construction de l'hypergraphe $H_A$ conduit à l'insertion suivante :

Proposition1 :

(i) Si $H_A = (X, E_A)$ est l'hypergraphe généré par une arborescence $A \in \mathcal{A}$ comme décrit ci-dessus, alors $|E_A| = n - 1$

(ii) L'hypergraphe $H_A$ généré par $A \in \mathcal{A}$ est non simple si au moins un des nœuds $v \in A \; tel \; que \; d^+(v) \geq 2$. Par définition, un hypergraphe est simple si toutes ses arrêtes sont distinctes.

- **Distances entre arborescences**

Soit $|E_A| = n$ ; où $|.|$ est le cardinal de l'ensemble X. Soit $E^*$ la classe de tous les sous ensembles de X, et $\mu(E)$ la mesure de E sur $E^*$. Considérons $\mu(E) < \infty \; \forall \, E \in E^*$ . La distance de *Marczewski-Steinhaus* [151] entre deux ensembles $E_1$ et $E_2$ de $E^*$ est :

$$\sigma_\mu(E_1, E_2) = \begin{cases} \dfrac{\rho(E_1, E_2)}{\mu(E_1 \cup E_2)} & si \; E_1 \cup E_2 > 0 \\ 0 & si \; E_1 \cup E_2 = 0 \end{cases} \qquad (26)$$

Avec $\rho(E_1, E_2) = \mu(E_1 \Delta E_2)$, $\Delta$ est la différence symétrique.

Notons que $0 \leq \sigma_\mu(E_1, E_2) \leq 1$, en particulier si nous considérons que $\mu_c(E) = |E|$ et posons ensuite $e_1 = |E_1|$, et $e_2 = |E_2|$ et $d = |E_1 \cap E_2|$.

$$\sigma_{\mu c}(E_1, E_2) = \frac{e_1 + e_2 - 2d}{e_1 + e_2 - d} \qquad (27)$$

Nous avons aussi : $0 \leq \sigma_{\mu c}(E_1, E_2) \leq 1$

Considérons $A_1$ et $A_2$, deux éléments de $\mathcal{A}$ représentés par les hypergraphes $H_{A_1} = (X, E_{A_1})$ et $H_{A_2} = (X, E_{A_2})$ respectivement. La distance entre ces hypergraphes tient en considération l'étape spécifique de la construction des arêtes. La distance entre arborescences est donnée par la formule suivante :

$$d(A_1, A_2) = \frac{1}{n-1} \min_{p \in P} \sum_{i=1}^{n-1} \sigma_\mu \left( E_{A_1}^i, E_{A_2}^{p_i} \right) \qquad (28)$$

Où $p_i$ est le $i^{ème}$ élément de la permutation p des n-1 entiers. P est l'ensemble de toutes les permutations, $\sigma_\mu(.,.)$ est donnée ci-dessus. $E^{i}_{A_1} \in E_{A_1}$ et $E^{p_i}_{A_2} \in E_{A_2}$ i=1, n-1.

Les faits suivants sont impliqués par la définition précédente :

(i) (A,d) est un espace métrique.

(ii) $d(A_1, A_2) \leq 1$, $A_1$ et $A_2 \in \mathcal{A}$, la distance d(.,.)< 1 si on utilise $\sigma_{\mu c}(.,.)$ de la formule 27 au lieu de $\sigma_\mu(.,.)$ de la formule 26.

- Les étapes de notre procédure de la comparaison des indices de similarité sont :

1. Construire les dendrogrammes corresponds à chaque indice de similarité à partir la matrice de ressemblance obtenu par la classification hiérarchique (CAH) sur la matrice de données.
2. Comparer les dendrogrammes 2 à 2 par la distance de Marczewski-Steinhaus de la formule (28).
3. Construire la matrice de distances entre tous les dendrogrammes.
4. Obtenir le meta-dendrogramme final de tous les indices de similarité

- **Exemple de distance entre arborescences :**

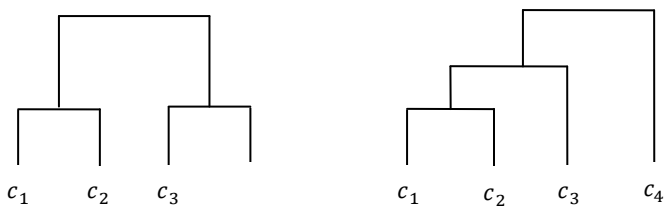


Figure 11 : Arbres d'assemblage

En prenant X={$c_1$ , $c_1$ ,$c_2$ ,$c_3$ ,$c_4$ } l'ensemble des composants d'un produit, $A_1$et $A_2$ deux arbres d'assemblages possibles (figure 11). Nous calculons la distance proposée entre les deux gammes $A_1$ et $A_2$ de A, basée sur l'ensemble des composants de X. $|X| = 4$ .

Pour cela, nous cherchons les ensembles $E_{A_1}$ et $E_{A_2}$ des sous-arbres correspondant aux etapes intermediares de la constitution du produit. Ces étapes sont les arêtes des hypergraphes

$H_{A1} = (X, E_{A1})$ et $H_{A2} = (X, E_{A2})$. D'après la proposition 1, le nombre d'étapes intermédiaires pour la constitution du produit est $|E_{A_1}|=|E_{A_2}|=|X|-1=3$.

Pour simplifier les notations, nous posons $c_i = i$ :

$E_{A_1} = \{\{1,2\},\{3,4\},\{1,2,3,4\}\}$ avec $\mathrm{E}^1_{\mathrm{A}_1} = \{1,2\}$, $\mathrm{E}^2_{\mathrm{A}_1} = \{3,4\}$ et $\mathrm{E}^3_{\mathrm{A}_1} = \{1,2,3,4\}$.

$E_{A_2} = \{\{1,2\},\{1,2,3\},\{1,2,3,4\}\}$

La distance $d(A_1, A_2)$ donnée par la formule donnée par ***Marczewski-Steinhaus*** se calcule entre les composantes de $E_{A_1}$ et les composantes des permutations de $E_{A_2}$. Pour cela, nous cherchons l'ensemble P des permutations p de $E_{A_2}$. P est décrit comme suit :

$\{\{1,2\},\{1,2,3\},\{1,2,3,4\}\}$ avec $\mathrm{E}^1_{\mathrm{A}_2} = \{1,2\}$, $\mathrm{E}^2_{\mathrm{A}_2} = \{1,2,3\}$ et $\mathrm{E}^3_{\mathrm{A}_2} = \{1,2,3,4\}$

$\{\{1,2\},\{1,2,3,4\},\{1,2,3\}\}$ avec $\mathrm{E}^1_{\mathrm{A}_2} = \{1,2\}$, $\mathrm{E}^2_{\mathrm{A}_2} = \{1,2,3,4\}$ et $\mathrm{E}^3_{\mathrm{A}_2} = \{1,2,3\}$

La même démarche est appliquée pour chercher $\mathrm{E}^{\mathrm{p_i}}_{\mathrm{A}_2}$ des permutations restantes :

$\{\{1,2,3\},\{1,2\},\{1,2,3,4\}\}$

$\{\{1,2,3\},\{1,2,3,4\},\{1,2\}\}$

$\{\{1,2,3,4\},\{1,2\},\{1,2,3\}\}$

$\{\{1,2,3,4\},\{1,2,3\},\{1,2\}\}$

Nous calculons la distance entre $E_{A_1}$ et les permutations de $E_{A_2}$. Le minimum des valeurs obtenues nous assure la distance entre les arbres $A_1$ et $A_2$. Dans le cas traité ci-dessus, $d(A_1, A_2)$=0.25.

# *CHAPITRE IV*

# RESULTATS EXPERIMENTAUX

## Introduction

Les travaux que nous avons réalisés se basent en particulier sur les systèmes de recommandation, mais certains présentant des méthodes génériques peuvent être utilisées dans plusieurs domaines. Pour valider nos résultats, nous présentons la base et la structure des données utilisées, les métriques d'évaluation ainsi que la série d'expériences effectuée pour examiner la performance de nos nouveaux méthodes de recommandation qui ont été implémentés avec les algorithmes cités auparavant ; à savoir l'algorithme de filtrage collaboratif multicritères MCCR, la méthode de regroupement STGM, la méthode de recommandation d'une liste d'items RMCS. Enfin, la méthode de comparaison des indices de similarité LATCCM. Les résultats semblent très prometteurs.

### 4.1. La base de données

Afin d'évaluer la performance d'un algorithme de filtrage collaboratif, nous utilisons le jeu de données MovieLens[9] fournie par l'équipe américaine de recherche GroupLens[10] de l'Université du Minnesota. MovieLens[11] est un site web de recommandation de films (http://MovieLens.umn.edu/) à travers lequel les utilisateurs évaluent d'abord un sous-ensemble de films qu'ils ont déjà vu. L'application capture les notes de l'utilisateur pour les films et fournit une recommandation formée d'une liste de films. Le jeu de données MovieLens a été largement utilisé par la communauté scientifique pour évaluer et comparer les algorithmes de filtrage collaboratif [162]. Il présente en effet l'avantage de reposer sur des votes réels et fournit de ce fait un bon support de validation.

La base de données historique se compose de 100 000 votes (U) de 943 utilisateurs (C), et 1682 films (P), chaque utilisateur a au moins 20 évaluations, ainsi que ses caractéristiques. La matrice de votes présente une dispersion de 6,30 %, c'est-à-dire qu'il y a 93,70 % de données manquantes, considérées comme des non-votes. Aujourd'hui, le site a plus de 45 000 utilisateurs qui ont exprimé des opinions sur les 6600 films différents. Les notes sont sur une échelle Likert avec des valeurs entières comprises entre 1 et 5, avec 1 et 2 représentant les notes négatives, 3, 4 et 5 représentant les avis positifs.

Le jeu de données a été divisé en cinq ensembles d'apprentissage et cinq ensembles de test appelés respectivement « base » et « test ». Le fichier U.data correspond au jeu de données complet. U[1-5].base sont les cinq ensembles d'apprentissage et U[1-5].test sont

[9] http://www.grouplens.org/node/73
[10] http://www.grouplens.org
[11] http://www.movielens.org/login

les cinq ensembles de validation générés. Les ensembles d'apprentissage et de test contiennent respectivement 80 % et 20% des votes globaux.

Les items de l'ensemble de données *MovieLens* correspondent à des films. Les caractéristiques des items sont sous forme de descriptions sur les films : titre, catégorie, sujet, auteurs, et le temps de la publication [19, 32]. Chaque film lui correspond une ligne de caractéristiques sous la forme suivante :

Identifiant de film | Titre de film | date de diffusion | date sortie de vidéo |Mise URL | inconnu | Action | Aventure | Animation |Enfants | Comédie | Crime | Documentaire | Drame |Fantasy | Film-Noir | Horreur | Musique | Mystère | Romance | Science-fiction | Thriller | Guerre | Western.

Les identifiants de films sont ceux utilisés dans l' ensemble de données principale. Le titre du film est une chaîne. Les dates de publication sont de la forme jj - mmm - aaaa, par exemple, 14 - Jan- 1967. L'URL IMDb est un lien menant à la page de base de données du film correspondant. Les 19 derniers champs correspondent aux genres de films qui peuvent appartenir à plus d'un genre en même temps.

Les 943 utilisateurs du jeu de données sont décrits par trois variables descriptives qualitatives nominales (sexe, métier, code postal) et une variable quantitative (âge) :

• Les utilisateurs sont représentés par 670 hommes et 273 femmes;

• Les métiers les plus représentatifs sont les étudiants 196, les autres 105, les professeurs 95, et les agents de l'administration 79.

• La variable descriptive code postal apporte peu d'information et n'est pas sélectionnée

• La répartition des âges se distribue entre 7 et 73 ans, avec une médiane de 31 ans et une moyenne de 34 ans.

### 4.2. La Structure de données

Dans cette section, nous fournissons des détails sur la structure de données de *Movielens* dont nous avons besoin pour les algorithmes de CF proposés.

### 4.2.1. Les données d'évaluation des items

La représentation des données pour le filtrage collaboratif est basée sur la construction d'une matrice U de votes de dimension (M × N) item /utilisateur, qui est présentée dans la table 7. U(i,j) dans la $i^{ème}$ ligne et la $j^{ème}$ colonne de la matrice représente la note de l'item $p_i$ évalué par l'utilisateur $c_j$.

Les items sont énumérées de 1 à 1682, et les utilisateurs de 1 à 943, tandis que l'évaluation d'un utilisateur sur un item prend des valeurs comprises entre 1 et 5.

| | $c_1$ | $c_2$ | … | $c_N$ |
|---|---|---|---|---|
| $p_1$ | $U_{11}$ | $U_{12}$ | … | $U_{1N}$ |
| $p_2$ | $U_{21}$ | $U_{22}$ | … | $U_{2N}$ |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| $p_M$ | $U_{M1}$ | $U_{M2}$ | … | $U_{MN}$ |

Table 9: La matrice de l'évaluation des utilisateurs sur les items

### 4.2.2. Les données de temps de l'évaluation des items

La table 10, illustre la matrice T de dimension M× N de données relatives au moment de réalisation de l'évaluation de l'item. L'élément $T_{i,j}$ dans la $i^{ème}$ ligne et la $j^{ème}$ colonne de la matrice T, correspond au moment d'évaluation de l'item i pour l'utilisateur j. Nous avons donc :

| | $c_1$ | $c_2$ | … | $c_N$ |
|---|---|---|---|---|
| $p_1$ | $T_{11}$ | $T_{12}$ | … | $T_{1N}$ |
| $p_2$ | $T_{21}$ | $T_{22}$ | … | $T_{2N}$ |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| $p_M$ | $T_{N1}$ | $T_{N2}$ | … | $T_{MN}$ |

Table 10: La matrice du temps de l'évaluation des utilisateurs sur les items

### 4.2.3. Les caractéristiques des items

Les variables descriptives des items peuvent être présentées de différentes manières. Tout d'abord, en se basant sur les données descriptives des items telles que le genre d'un film. Ensuite, elles peuvent être renseignées par les utilisateurs sous forme textuelle. Il peut s'agir par exemple de données structurées (systèmes des tags), mais également de données non structurées telles que des textes descriptifs, des critiques journalistiques ou encore des commentaires sur des forums en ligne. Les mots sont souvent pris comme descripteur et subissent généralement des traitements linguistiques. Une fois les variables

descriptives des utilisateurs ou items sont collectées, des mesures de similarités peuvent être mises en place.

Dans notre cas, nous avons besoin aussi d'extraire les attributs (genre du film par exemple) de l'élément existant dans les données de *MovieLens* pour construire des vecteurs caractéristiques des 19 caractéristiques de l'item dans une table de vecteurs d'items à partir de leurs renseignements , spécialement les 19 derniers champs qui sont les genres des films afin de les utiliser dans notre modèle de recommandation.

### 4.3. L'évaluation des systèmes de recommandation

Beaucoup de mesures ont été proposées pour évaluer la performance d'un algorithme de filtrage collaboratif. La mesure la plus utilisée est l'erreur absolue moyenne, MAE.

Nous nous concentrerons par la suite sur l'explication de cette mesure. Par ailleurs, il semble important de contrôler avant toute chose la dispersion du jeu de données d'entrée avant d'évaluer le système de recommandation [95].

#### 4.3.1. Sparcity

La dispersion (sparcity) [106] d'un jeu de données représente le ratio de remplissage de la matrice d'usage, celle-ci est définie comme suit :

$$Sparcity = 1 - \frac{|U|}{|P| \times |C|} \qquad (29)$$

Où |U|, |P|, |C| correspondent aux cardinalités des mesures renseignées, des items et aux utilisateurs respectivement.

Les caractéristiques du jeu de données Movielens ; il a été collectée à l'aide d'une plateforme par des pionniers du domaine, le groupe de recherche Grouplens de l'université de Minnesota ; elle présente une dispersion de 6,30 %, c'est-à-dire qu'il y a 93,70 % de données manquantes, considérées comme des non-votes.

#### 4.3.2. L'erreur Absolue Moyenne

Nous définissons l'erreur absolue moyenne (MAE, Mean Absolute Error) [165, 25] comme une mesure de l'écart de prévision de notes de recommandations avec les vraies valeurs spécifiées par les utilisateurs. Supposant que l'ensemble de valeurs réelles des notes spécifiées de l'utilisateur est $\{q_1, \ldots, q_n\}$ et l'ensemble de prédiction des évaluations émis par l'algorithme de recommandation est $\{u_1, u_2, \ldots, u_n\}$, formellement,

$$MAE = \frac{\sum_{i=1}^{n} |u_i - q_i|}{n} \quad (30)$$

Où n représente le nombre total des prédictions calculées pour tous les utilisateurs. Selon cette mesure, un meilleur algorithme CF doit avoir un MAE minimal. D'autres mesures similaires, telles que la racine carrée de la moyenne des différences au carré (Root Mean Squared Error, RMSE) et la sensibilité ROC sont parfois utilisées. Cependant, nous utilisons le MAE afin d'évaluer la qualité de prédiction.

### 4.3.3. Précision et Rappel

Dans le cas de recommandation d'une liste d'items, nous ne sommes pas basés sur une heuristique de prédiction, mais plutôt, nous proposons une méthode graphique ou structurelle basée sur des ensembles, ce qui implique l'utilisation des métriques d'évaluation comme la précision et rappel.

Lorsqu'on cherche à prédire si un utilisateur est intéressé ou non par un item, quatre possibilités sont offertes par la matrice de confusion.

| Item | Pertinent | Non pertinent |
|---|---|---|
| Recommandé | Vrai Positif ($vp$) | Faux Positif ($fp$) |
| Non recommandé | Faux Négatif ($fn$) | Vrai Négatif ($vn$) |

Table 11: Matrice de confusion de la recommandation d'un item à un utilisateur

La précision [69] correspond au pourcentage ou au nombre des items suggérés et s'avérant véritablement pertinentes pour l'utilisateur. Par exemple, si l'on considère une liste des Top-N recommandations, la précision correspond à la proportion d'items véritablement consommés, appréciés ou achetés par l'utilisateur courant. Il est calculé en utilisant l'expression suivante :

$$\text{precision} = \frac{vp}{vp + fp} \quad (31)$$

Le rappel (recall) [166] mesure le nombre de recommandations pertinentes émises au regard du nombre total de recommandations pertinentes. Concrètement, on énumère le

nombre d'items dont la mesure associée est non nulle et se retrouvant parmi les items suggérés, il est calculé par la formule ci-dessous :

$$\text{rappel} = \frac{vp}{vp + fn} \qquad (32)$$

Si la précision est faible, l'utilisateur sera insatisfait, car il devra perdre du temps à lire des items qui ne l'intéressent pas. Si le rappel est faible, l'utilisateur n'aura pas accès à un item qu'il souhaiterait avoir.

Un système de recommandation parfait doit avoir une précision et un rappel près de la valeur 1, mais ces deux exigences sont souvent contradictoires et une très forte précision ne peut être obtenue qu'au prix d'un rappel faible et vice-versa.

### 4.4. Les résultats de l'algorithme de filtrage collaboratif multicritère MCCR

Nous abordons les trois questions suivantes:

- Comment le paramètre $T_0$ peut influencer sur la précision de la prédiction ?

$T_0$ est un paramètre important pour notre nouvel algorithme, désigne le taux de décroissance de l'importance des données anciennes. C'est à-dire, $T_0$ décide sur l'importance des données historiques. Différentes valeurs de $T_0$ sont testées pour examiner l'impact de ce paramètre sur la performance de l'algorithme proposé.

- Comment se situe notre algorithme par rapport aux algorithmes de filtrage collaboratif existants ?

Notre approche est comparée à l'algorithme classique basé item, avec l'algorithme de pondération de temps. Ces deux algorithmes utilisent la corrélation de Pearson.

- Est-ce que l'application d'une approche de regroupement sur notre méthode améliore la performance ?

Nous avons appliqué la méthode de regroupement k-means pour former le voisinage d'un élément cible sans utiliser la base de données entière pour voir si cela augmentera la précision de notre algorithme ou non.

Dans une première phase, nous testons: l'impact du paramètre $T_0$

Le paramètre $T_0$ contrôle le taux de décroissance, par conséquent nous donne une information sur l'importance de données historiques. Nous avons choisi l'échantillon d'items de taille 300, le voisinage d'item a été fixé à 30 et nous avons fait varier le

paramètre $T_0$ en lui effectuer les valeurs 10, 20, 50, 100 et 200. Les résultats sont donnés dans le tableau 10.

| Nombre d'items | $T_0$ | MAE |
|---|---|---|
| 300 | 10 | 0.1381 |
| | 20 | 0.1135 |
| | 50 | 0,1153 |
| | 100 | 0,1146 |
| | 200 | 0.1161 |
| | auto | **0.1113** |

Table 12: MAE pour des différentes valeurs $\mathbf{T_0}$

Nous avons varié le $T_0$ par différentes valeurs et nous avons observé qu'il y a des variations dans les valeurs de MAE. L'attribution d'une valeur fixe $T_0$ pour tous les items du système est inappropriée. Les résultats expérimentaux montrent que la valeur de paramètre $T_0$ « auto » calculée automatiquement par l'algorithme proposé, a une amélioration légère de MAE.

Dans une seconde phase, nous faisons la comparaison de notre algorithme avec les algorithmes de filtrage collaboratif existants.

Nous comparons notre nouvel algorithme MCCR avec l'algorithme classique basé sur l'item et l'algorithme de pondération de temps. Comme compte tenu de la MAE pour mesurer la performance de l'algorithme de prédiction en faisant la variation du nombre de voisinage [10,15,20,25,30,35,40], notre algorithme est capable de surpasser la précision de la prédiction, les résultats sont présentés dans la figure 12.

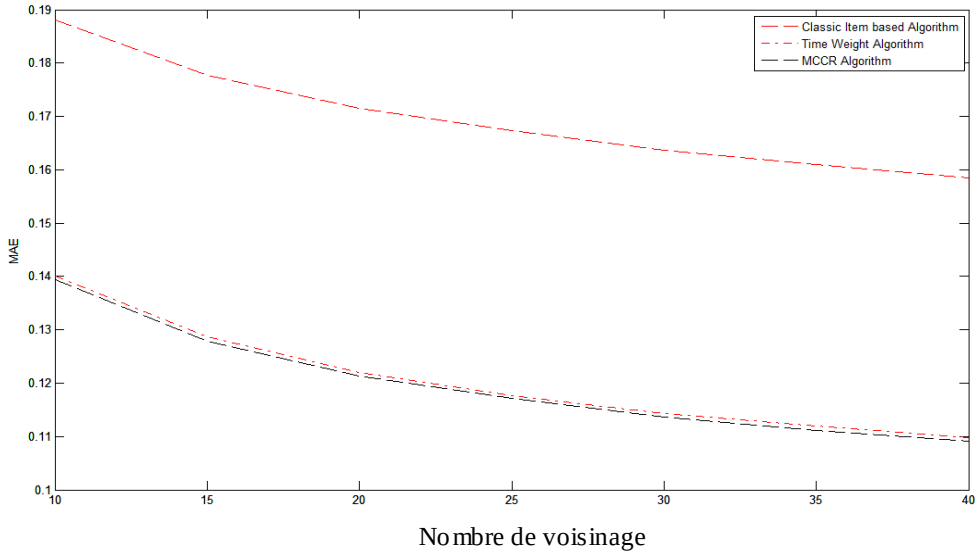


Figure 12: La comparaison de la précision des algorithmes de recommandation

La troisième phase: l'application de regroupement sur l'algorithme de FC

Dans ce qui suit, nous choisissons la méthode k-means comme algorithme de clustering. Le k-means a un paramètre explorable qui est k. Nous appliquons le regroupement sur l'algorithme item-based avec les valeurs [2,3,4,5] et sur le même intervalle précédent du voisinage. Nous allons prendre celui qui minimise le MAE.

| | Le clustering k-means pour l'algorithme de filtrage collaboratif | | | | | | |
|---|---|---|---|---|---|---|---|
| | 10 | 15 | 20 | 25 | 30 | 35 | 40 |
| k=2 | 0.1864 | 0.1766 | 0.1710 | 0.1668 | 0.1646 | 0.1624 | 0.1652 |
| k=3 | 0.1896 | 0.2852 | 0.1994 | 0.1651 | 0.1616 | 0.1595 | 0.1578 |
| k=4 | 0.1843 | 0.1736 | 0.1674 | 0.1637 | 0.1631 | 0.1577 | 0.1547 |
| k=5 | 0.1841 | 0.1736 | 0.1712 | 0.1674 | 0.1596 | 0.1605 | 0.1541 |

Table 13: Le MAE pour le filtrage collaboratif à base item avec les différents k

Comme indiqué à la table 13, nous avons calculé le MAE en explorant k et le nombre du voisinage. Nous avons pris le k qui minimise le MAE pour comparer l'algorithme de FC à base item avec et sans regroupement comme il est montré dans la figure 13.

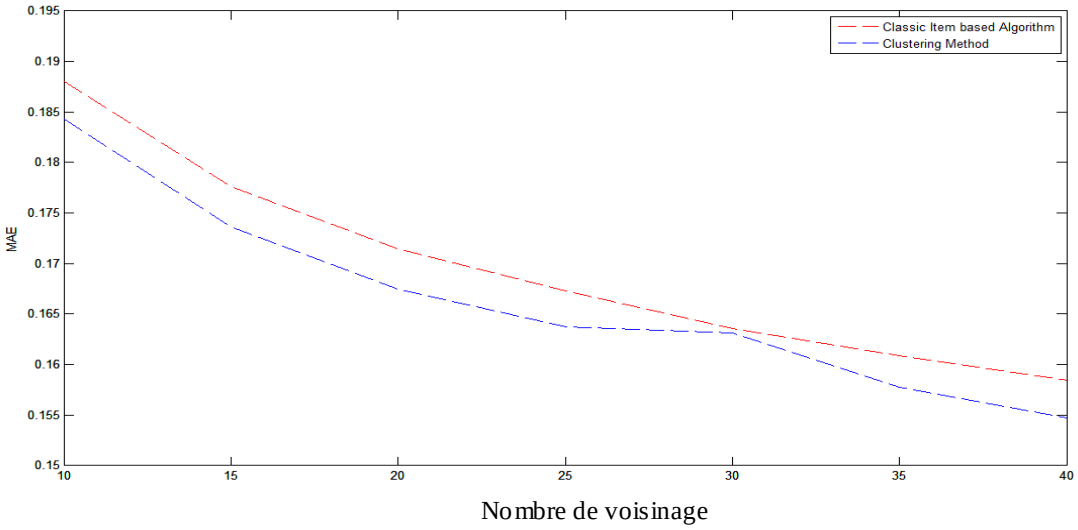


Figure 13: La précision de l'algorithme de FC à base item avec et sans clustering

Une fois le k approprié trouvé, nous remarquons dans la figure 13 que le regroupement a un effet positif sur l'amélioration de la prédiction de l'algorithme de FC à base item

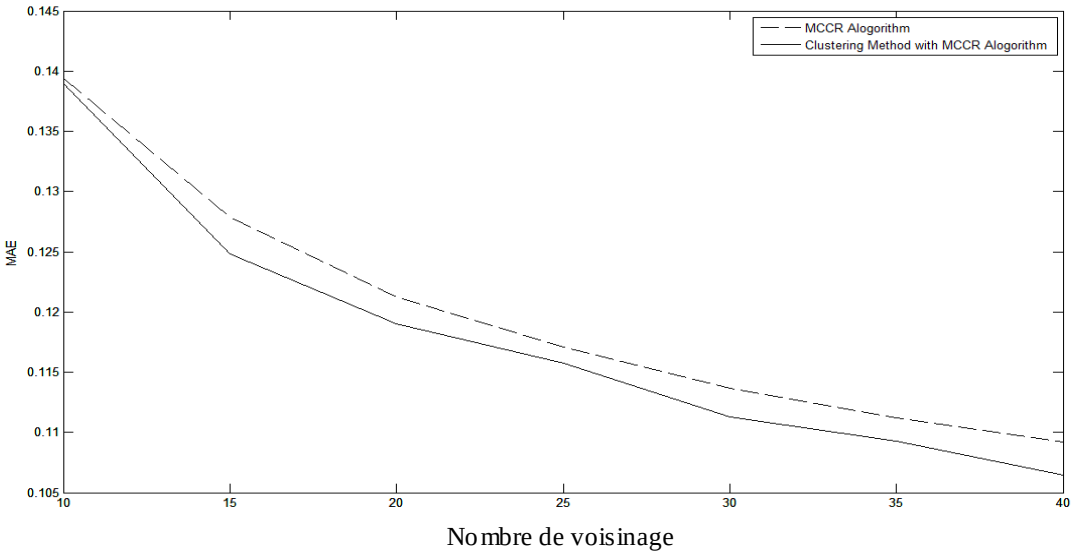


Figure 14: La comparaison de la précision du l'algorithme MCCR avec et sans clustering

Nous observons la même chose dans la figure 14 pour la comparaison de notre algorithme MCCR et le MCCR avec regroupement. Nous avons trouvé que la variation du nombre du voisinage et le regroupement a un effet significatif sur la qualité de la prédiction. En effet, le MAE diminue en fonction du nombre de voisinage d'items, plus sa valeur est faible, moins l'erreur est importante. Il est clair que la performance de

l'algorithme proposé s'améliore avec l'augmentation du nombre d'items similaires, ce qui conduit à une bonne précision dans la recommandation.

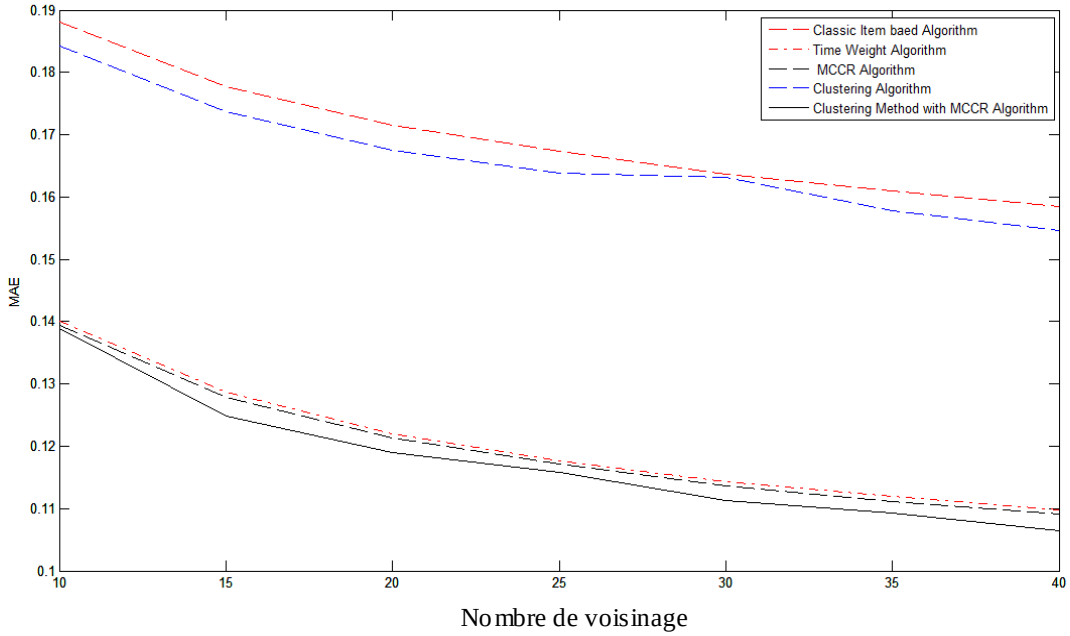


Figure 15: La comparaison de la précision des algorithmes de recommandation

Dans la figure 15, nous utilisons le MAE pour mesurer la performance de l'algorithme de prédiction. Afin de vérifier l'efficacité de notre algorithme avec et sans regroupement, et en variant le nombre de voisinage, notre algorithme est comparé avec d'autres algorithmes existants : l'algorithme classique basé sur l'item sans et avec regroupement et l'algorithme de pondération du temps. Ces algorithmes offrent des bons résultats mais, nous remarquons que la précision de notre algorithme avec regroupement est meilleure par rapport à ces derniers. Les résultats expérimentaux montrent qu'il y a un effet positif sur la précision de la recommandation en raison de la prise en compte du nombre de voisinage et du regroupement.

Le nombre de voisinage a été sélectionné par la technique de best-n-neighbors [108] qui consiste à choisir les « n » meilleures corrélations pour un n donné. Cette technique se comporte raisonnablement assez bien, et ne limite pas la coverage (le pourcentage d'items utilisé dans la prédiction), cependant, le choix d'un grand n entraîne trop de bruit pour les items qui ont des corrélations élevées et le choix d'un plus petit n peut causer de mauvaises prévisions pour les utilisateurs qui n'ont pas de corrélations élevées, bien que dans nos expérimentations, cet effet ne s'est pas produit jusqu'à ce que la taille du voisinage a été réduit à moins de 15. Cet effet peut être vu dans la figure 15.

L'approche de regroupement k-means est la plus utilisée dans les systèmes de recommandation à cause de la simplicité de sa mise en œuvre et sa performance reconnue pour l'évolutivité et la précision de la recommandation [80]. Toutefois, lorsque la rareté augmente, cela influence négativement sur les algorithmes de regroupement et la précision de la recommandation [108]. C'est pour cette raison que nous avons proposé une solution à ce problème en utilisant un autre type de clustering, (co-clustering) et qui est présenté dans la partie suivante.

Nous avons montré que notre algorithme MCCR a pour objectif de rendre la recommandation plus précise. Tout d'abord, une nouvelle mesure de prédiction multicritères a été incluse pour obtenir de meilleures performances de prédiction. Cette phase de prédiction est précédée par le clustering en utilisant k-Means pour mieux traiter le problème de l'évolutivité et pour éviter la complexité de calcul. Cela a un impact positif sur la prédiction d'un item pour un utilisateur cible, mais, nous avons remarqué que la rareté influence négativement sur le clustering, pour cela, nous considérons que l'utilisation des nouvelles techniques de co-clustering dans le processus de recommandation à savoir BEA pourra pallier ce problème.

### 4.5. Les résultats de la méthode de regroupement STGM

BEA est une méthode de co-clustering qui peut utiliser une matrice binaire ou avec des valeurs quelconques. L'utilité de BEA vient par sa capacité de produire une matrice de blocs diagonaux qui sont considérés utiles pour les problèmes qui se décomposent en sous problèmes. BEA peut aider à identifier donc un certain nombre de blocs sans ou avec interaction, tel que le montre l'exemple de la matrice résultat du BEA dans la figure 16. Les blocs non diagonaux indiquent les interrelations entre les groupes comme dans la Figure 17.

L'algorithme donnera à partir de la matrice d'entrée (un aperçue d'une partie de la matrice d'incidence illustré par la Figure 15), une matrice de sortie d'une forme de blocs diagonaux (voir Figure 16).

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 5 | 4 | 5 | 0 | 3 | 5 | 0 | 5 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 5 | 0 | 3 | 4 | 0 | 0 | 5 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 |
| 0 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | 0 | 0 |
| 5 | 1 | 5 | 4 | 0 | 0 | 0 | 5 | 0 | 4 |
| 0 | 0 | 5 | 0 | 5 | 5 | 0 | 0 | 0 | 4 |
| 4 | 4 | 5 | 3 | 5 | 0 | 0 | 5 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 5 | 0 | 0 | 0 | 2 | 0 | 0 | 0 |
| 5 | 0 | 5 | 0 | 5 | 0 | 0 | 0 | 0 | 0 |
| 4 | 1 | 0 | 3 | 5 | 0 | 0 | 0 | 0 | 4 |
| 3 | 4 | 0 | 0 | 5 | 0 | 0 | 0 | 0 | 4 |
| 4 | 4 | 5 | 0 | 4 | 0 | 4 | 4 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 4 | 0 |
| 3 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 4 |
| 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 0 |
| 3 | 0 | 5 | 0 | 5 | 0 | 5 | 0 | 0 | 0 |

Figure 16: Une partie de la matrice d'incidence

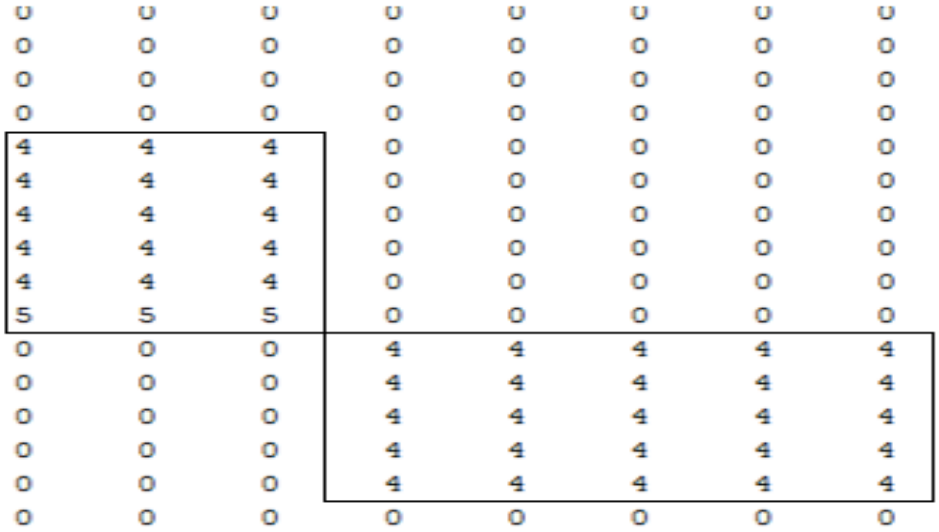

Figure 17: La matrice de sortie de blocs diagonaux

En appliquant BEA, le but est alors de rassembler les utilisateurs ou (items) ayant donné des votes similaires à un ensemble d'items. Nous trouvons que l'algorithme d'énergie maximum répond à ce besoin. Après la permutation des lignes et des colonnes par BEA et en se basant sur la valeur maximale d'énergie, nous obtenons une matrice contenant des blocs de valeurs élevés (blocs positifs), ce qui est le cas ici, ou bien des blocs de faibles valeurs (blocs négatifs).

Pour le cas des blocs non diagonaux, nous pouvons remarquer dans la figure 18 que les utilisateurs appartenant à des groupes différents peuvent avoir des valeurs élevées pour les mêmes items.

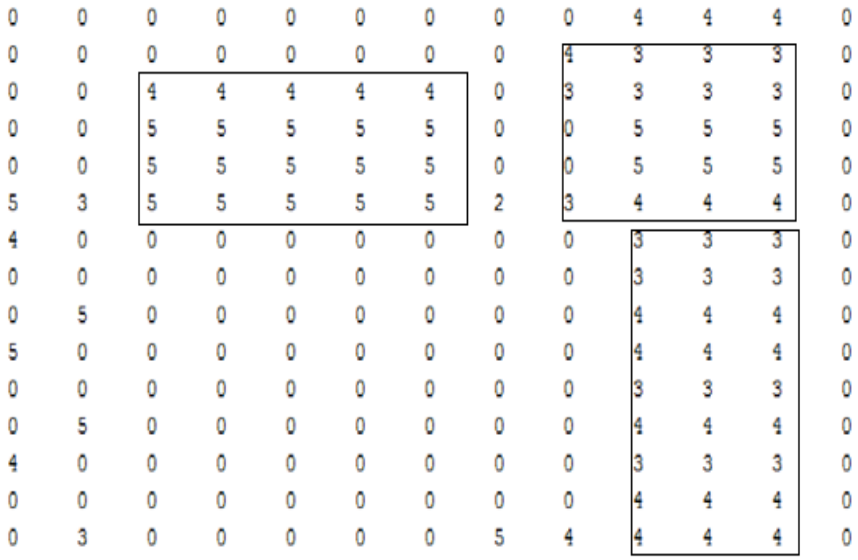

Figure 18: La fome checkerboard du bloc

L'algorithme BEA est plus précis, plus robuste comme il est demontré dans le domaine de la bioinformatique et offre des meilleurs résultats [163]. Le nombre de blocs est generé automatiquement sans aucune nécessité de specifier leur nombre, alors que dans l'algorithme de partitionnement k-means, un ensemble prédéfini de paramètres d'entrée s'impose : le nombre de clusters, la position initiale de centres des classes et la métrique de distance sont généralement fournis en entrée.

Une fois BEA appliqué sur les données, nous avons trouvé les blocs, mais reste à faire leur extraction. Pour cela, nous avons fait une seconde réorganisation de la matrice résultante basée sur le poids calculé pour chaque ligne (utilisateur) et chaque colonne (item) précité dans la formule 23, et nous avons effectué un tri décroissant sur ces poids, cela permet d'obtenir les résultats présentés dans les figures 19 et 20.

Comme indiqué dans la figure 19 représentant les utilisateurs en fonction de leurs poids, nous pouvons remarquer une série claire de clusters définis par intervalles comprend les utilisateurs.

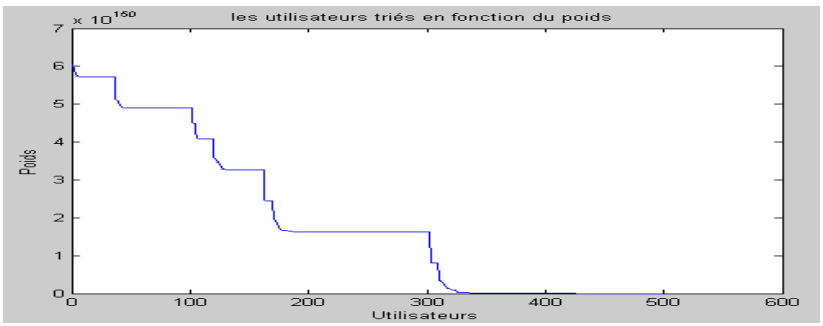


Figure 19: Les utilisateurs triés par le poids calculé

D’après la figure 20, la même procédure est appliquée aux colonnes (items) en calculant le poids et le tri sur les valeurs de ce dernier. Nous remarquons que les clusters se chevauchent et les données ne sont que faiblement séparées.

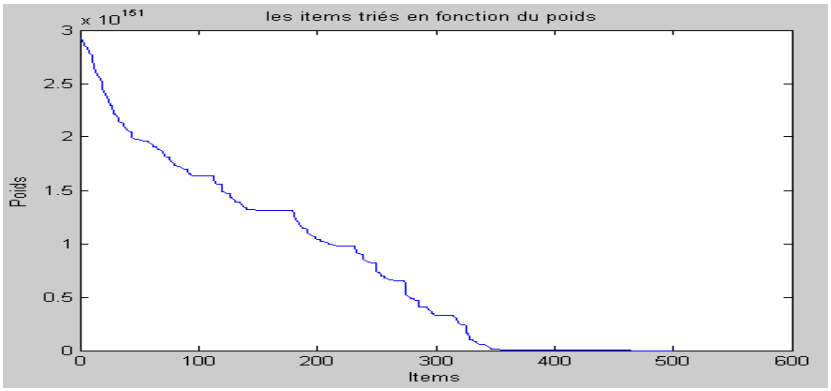


Figure 20: Les items triés par le poids calculé

Notre méthode de partitionnement d’un ensemble d’utilisateurs ou un autre des items de N valeurs et M valeurs respectivement sur une dimension consiste à diviser cet ensemble en coupant entre ces valeurs consécutives séparées par les plus grande distances. Ces valeurs sont rangées par ordre décroissant. Avec les valeurs de chaque poids, nous avons calculé l’étendue mobile pour les utilisateurs et les items triés. Cela donne lieu à deux courbes 21 et 22.

Cette technique est alors tout à fait efficace lorsque des séparations importantes entre les données existent et s’il n’y pas de bruit dans les données. C’est le cas de la figure 20, ou un ensemble d’utilisateurs sont bien séparés. Par contre la présence de bruit dans les données risque de mettre cette méthode en défaut à cause de l’apparition d’objets bruités dans la zone de séparation des groupes.

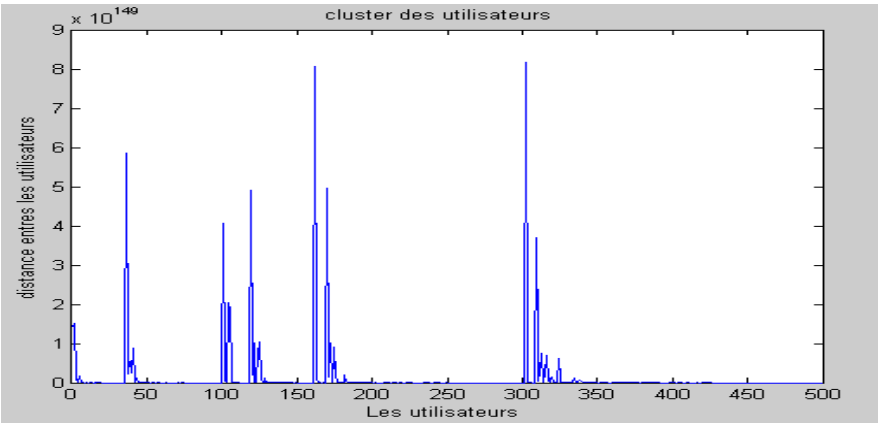


Figure 21: L’étendue mobile des utilisateurs

De la même manière, si les données ne sont que faiblement séparées lors de leurs projections sur une dimension, alors la méthode peut également échouer à cibler la meilleure zone de séparation. C'est le cas de la figure 21, même s'il est plus probable que les valeurs consécutives séparées par la plus grande distance se situent dans les zones de séparation des groupes plutôt qu'en leurs centres. Une solution pour pallier cette limitation consiste à rechercher la distance entre deux objets contenus dans un certain voisinage au lieu de chercher la distance maximale entre objets directement voisins. Cela améliore la probabilité de cibler des objets proches de zones de séparation des groupes.

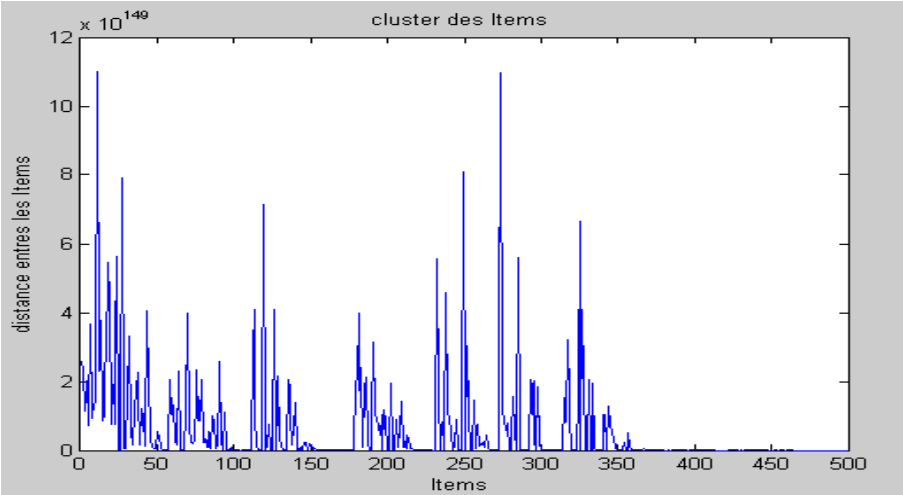


Figure 22: L'étendue mobile des items

Enfin, on peut considérer différents nombres de groupes possibles lors des découpages sur une dimension. Le découpage se fait selon la plus grande valeur de l'étendue glissante et ainsi de suite. On arrête le découpage jusqu'un critère assurant une inertie inter-classe maximale et une inertie intra-classe minimale.

La solution proposée, STGM est considérée comme une extension de l'algorithme BEA. Nous avons adapté ce dernier pour un co-clustering naturel d'utilisateurs et d'items. En effet, nous avons appliqué une fonction de mesure et une étendue glissante sur la matrice d'utilisateurs et d'items triés, fournie par l'algorithme BEA pour trouver des blocs diagonaux formés d'utilisateurs et d'items. Nous avons ainsi obtenu deux projections correspondantes pour trouver les clusters d'utilisateurs et ceux d'items. La méthode a été testée avec succès sur les données MovieLens. Cette approche systémique se comporte bien avec certains problèmes souvent rencontrés dans les systèmes de recommandation:

la rareté, l'évolutivité et le démarrage à froid. Le nombre de blocs est automatiquement créé sans aucune nécessité de spécifier leur nombre, alors que pour les autres méthodes de partitionnement, ce nombre doit être fixe au début.

D'une part, sur la base des résultats obtenus, nous remarquons que la méthode est prometteuse pour réaliser de meilleures classifications croisées sur la matrice d'utilisateurs et d'items. Toutefois, il faut signaler que nous trouvons des difficultés au niveau de la détermination des blocs non diagonaux. Dans le cas d'une matrice de données de taille moyenne, nous pouvons détecter les blocs visuellement, cependant, cette opération s'avère difficile dans le cas d'une grande base de données. Ce problème a été résolu par l'article [150], basé sur l'extraction des ensembles avec contexte bruité. Après l'extraction éventuelle des blocs non diagonaux, nous pouvons résoudre le problème de démarrage à froid.

Dans ce qui suit, nous présentons les résultats de la troisième méthode de recommandation d'une liste d'items sans calcul de prédiction.

## 4.6. Les résultats de la recommandation d'une liste d'items (RMCS)

- Paramètres d'évaluation de l'algorithme de recommandation

Pour évaluer notre algorithme, le calcul de la précision et du rappel nécessite avoir une base de test pour mesurer la performance de notre méthode RMCS. Pour cela, nous proposons la solution suivante :

Pour chaque utilisateur de l'ensemble de test, fournir une sous partie des mesures. Par exemple, blanchir un intervalle des mesures. Ainsi, on cherche à prédire les notes de l'utilisateur pour les items blanchis et on mesure l'erreur commise par l'algorithme lors de la prédiction de la mesure par rapport à la vraie mesure auparavant blanchie. De cette manière, on peut détecter l'ensemble des items pertinents recommandés ainsi que l'ensemble des données pertinentes et non recommandées.

- Pourcentage d'élimination des données

En variant l'intervalle blanchis comme le montre la table14, nous observons un effet sur les valeurs de précision et rappel. Comme résultat, nous trouvons que lorsqu'on élimine 25% de données, on obtient de bons résultats pour la précision et le rappel, car les 75% des données restantes contiennent une grande information autour de l'utilisateur qui aide

notre approche à effectuer de bonne recommandation. Par contre, lorsqu'on élimine 75% des informations, on obtient des résultats peu satisfaisants.

- Voisinage

Notre approche est basé sur le voisinage item/item et user/user pour effectuer la recommandation. En variant ce paramètre, nous trouvons que : lorsqu'on augmente le voisinage, une grande liste des items sera retournée. Cela augmente le rappel et diminue la précision et vice versa. Si l'utilisateur n'a pas besoin d'une liste complète de tous les éléments potentiels, mais juste d'une liste d'items pertinents, dans ce cas, seule la précision peut être appropriée. Mais, si la tâche est de trouver tous les éléments pertinents, le rappel devient important, donc cela dépend de notre besoin.

| **Pourcentage d'élimination** | **Niveau du voisinage** | **Précision moyenne** | **Rappel moyen** |
|---|---|---|---|
| **25%** | 1 | 0.9604 | 0.2501 |
| | 2 | 0.7610 | 0.4812 |
| | 3 | 0.5809 | 0.6324 |
| **50%** | 1 | 0.3416 | 0.1435 |
| | 2 | 0.2700 | 0.1863 |
| | 3 | 0.2031 | 0.2500 |
| **75%** | 1 | 0.1781 | 0.1166 |
| | 2 | 0.1592 | 0.0265 |
| | 3 | 0.1373 | 0.0085 |

Table 14: Précision/rappel

En prenant le cas de la figure 23 qui présente la courbe obtenue pour notre approche pour le voisinage de 3 avec 25% de données éliminées.

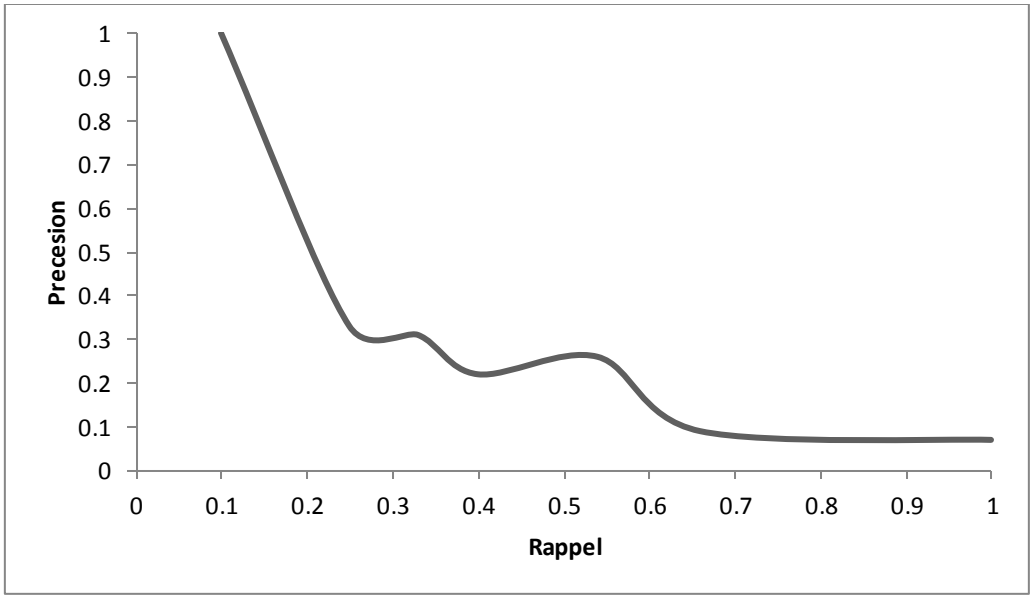


Figure 23: La courbe de la précision/rappel

Nous remarquons que pour des valeurs faibles de la précision, le rappel devient important et vice versa. Nous pouvons sélectionner la valeur optimale pour notre système ça dépend notre besoin.

Si la précision est faible, l'utilisateur sera insatisfait, car il devra perdre du temps à lire des items qui ne l'intéressent pas. Si le rappel est faible, l'utilisateur n'aura pas accès à un item qu'il souhaiterait avoir. Un système de recommandation parfait doit avoir une précision et un rappel près de la valeur 1, mais ces deux exigences sont souvent contradictoires et une très forte précision ne peut être obtenue qu'au prix d'un rappel faible et vice-versa.

Les utilisateurs font confiance au système de recommandation lorsque celui-ci recommande des items qu'ils apprécient. La satisfaction de l'utilisateur diminue quand un nombre significatif d'erreurs est produit par le système. Ils favorisent ainsi la mesure de précision sur celle du rappel. Ils soulignent que pour tout système de recommandation commercial, le plus important est d'éviter les faux positifs. Ainsi, le niveau de satisfaction des utilisateurs peut être facilement établi.

Dans ce sens, nous avons établi cette méthode pour avoir une recommandation personnalisée. Tout d'abord, cette méthode capte et capitalise sur les préférences des utilisateurs afin de les guider dans leurs choix. Ensuite, l'utilisateur lui-même est avantagé par un gain de temps et une découverte d'items souvent cachés auxquels il n'aurait pas pensé.

Dans la section suivante, nous comparons les indices de similarité utilisés dans le cadre du domaine industriel et du domaine du système de recommandation.

## 4.7. Les résultats de la comparaison des coefficients de similarité

Nous allons appliquer notre méthode de comparaison sur des indices appliqués dans le domaine de l'industrie mais qui sont utilisables dans les systèmes de recommandation.

Nous avons utilisé la base de données de [175] présentée dans la (table 15).

| | | Les produits | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Machines | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| | 1 | 0 | 1 | 1 | | | | | 1 | 1 | | 1 | | 1 | 1 | | 1 | 1 | | 1 | |
| | 2 | | | 1 | 1 | | 1 | 1 | | | | | | | 1 | | | | 1 | | 1 |
| | 3 | | 1 | | | | | | 1 | 1 | | 1 | | 1 | 1 | | 1 | 1 | | 1 | |
| | 4 | | | 1 | 1 | | 1 | 1 | | | 1 | | | | | | | | 1 | | 1 |
| | 5 | 1 | | | | 1 | 1 | | | | 1 | | 1 | | | 1 | | 1 | | | |
| | 6 | 1 | | | | 1 | | | | 1 | 1 | | 1 | | | 1 | | | | | 1 |
| | 7 | | | 1 | 1 | | 1 | 1 | | | | 1 | 1 | | | | | | 1 | | 1 |
| | 8 | | | 1 | 1 | | 1 | 1 | | | | | | | | | | | 1 | | 1 |

Table 15: La matrice de huit Machines

On dispose de 8 machines dont on doit identifier les groupes afin de créer des cellules de production. Chaque cellule contiendra un nombre de machines qui traite une famille de produits. On se base ici sur les 20 indices de similarité comparés dans [175] pour classer les machines de la matrice en familles. La figure 24 donne tous les possibles dendrogrammes de classification CAH. Notre méthode basée sur ces structures, calcule les distances entre ces dendrogrammes 2 à 2. En utilisant la CAH, nous obtenons un méta-dendrogramme classifiant les 20 méthodes.

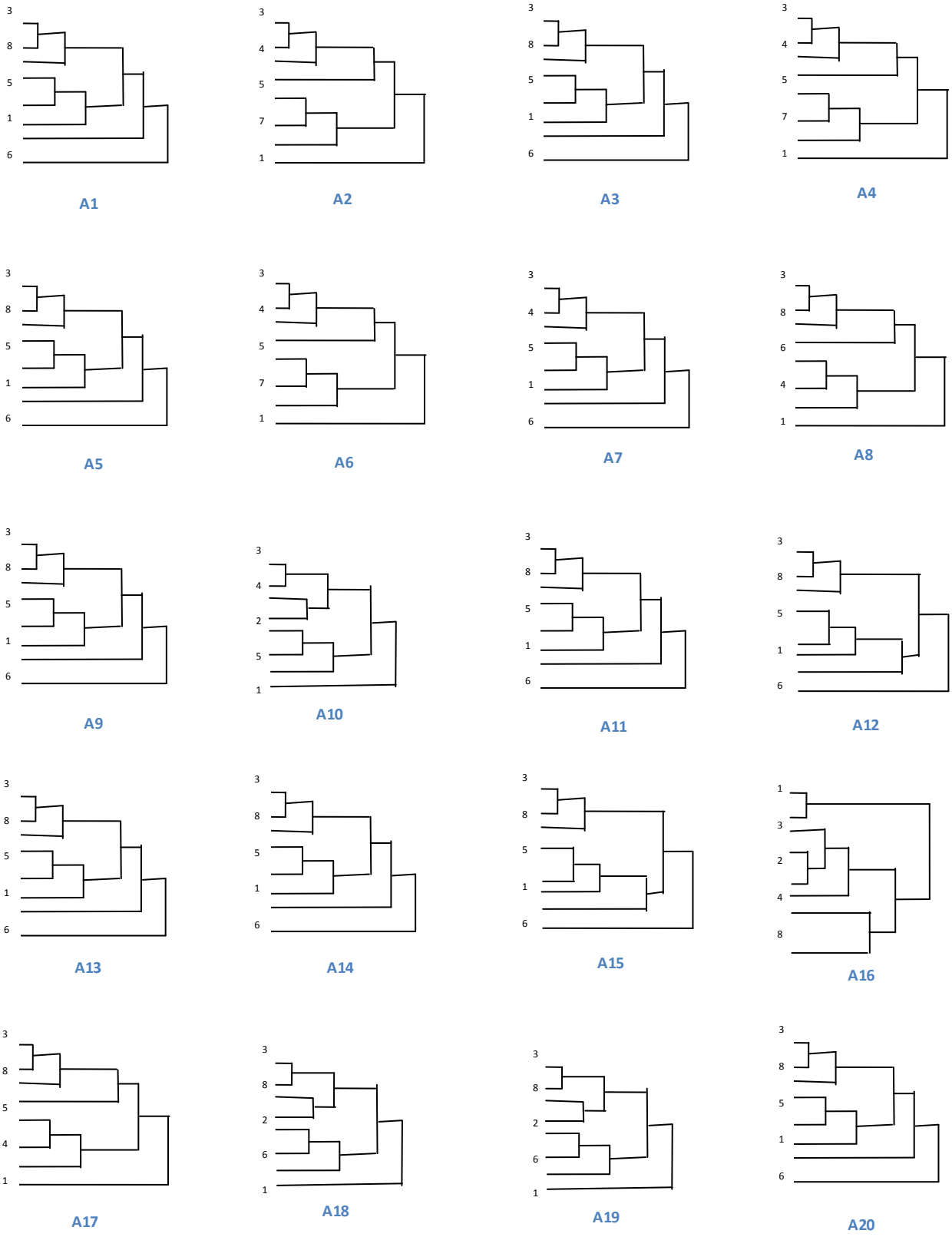


Figure 24: La classification de 20 indices de similarité

**Table 16: Matrice des distances entre les arbres des indices de coefficients**

| | $A_1$ | $A_2$ | $A_3$ | $A_4$ | $A_5$ | $A_6$ | $A_7$ | $A_8$ | $A_9$ | $A_{10}$ | $A_{11}$ | $A_{12}$ | $A_{13}$ | $A_{14}$ | $A_{15}$ | $A_{16}$ | $A_{17}$ | $A_{18}$ | $A_{19}$ | $A_{20}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $A_1$ | 0 | 0.3810 | 0 | 0.3810 | 0.0357 | 0.3095 | 0.3146 | 0.3478 | 0.0357 | 0.3748 | 0.0357 | 0.0816 | 0.0357 | 0.0357 | 0.0816 | 0.5722 | 0.1097 | 0.3387 | 0.3388 | 0 |
| $A_2$ | | 0 | 0.3810 | 0 | 0.3810 | 0.1071 | 0.0816 | 0.5000 | 0.3810 | 0.1857 | 0.3809 | 0.3963 | 0.3810 | 0.3810 | 0.3963 | 0.5607 | 0.3238 | 0.4976 | 0.4976 | 0.3810 |
| $A_3$ | | | | 0.3810 | 0.0357 | 0.3095 | 0.3146 | 0.3478 | 0.0357 | 0.3748 | 0.0357 | 0.0816 | 0.0357 | 0.0357 | 0.0816 | 0.5723 | 0.1097 | 0.3388 | 0.3388 | 0 |
| $A_4$ | | | | | 0.3810 | 0.1071 | 0.0816 | 0.5000 | 0.3809 | 0.1857 | 0.3809 | 0.3963 | 0.3810 | 0.3810 | 0.3963 | 0.5607 | 0.3238 | 0.4976 | 0.4976 | 0.3810 |
| $A_5$ | | | | | 0 | 0.3095 | 0.3121 | 0.3197 | 0 | 0.3476 | 0 | 0.1097 | 0 | 0 | 0.1097 | 0.5697 | 0.0816 | 0.3048 | 0.3048 | 0.0357 |
| $A_6$ | | | | | | 0 | 0.1862 | 0.5000 | 0.3095 | 0.2571 | 0.3095 | 0.3172 | 0.3095 | 0.3095 | 0.3172 | 0.5607 | 0.2524 | 0.4677 | 0.4677 | 0.3095 |
| $A_7$ | | | | | | | 0 | 0.4578 | 0.3120 | 0.1811 | 0.3120 | 0.3810 | 0.3121 | 0.3121 | 0.3810 | 0.5723 | 0.3044 | 0.4701 | 0.4701 | 0.3146 |
| $A_8$ | | | | | | | | 0 | 0.3197 | 0.4000 | 0.3197 | 0.3963 | 0.3197 | 0.3197 | 0.3963 | 0.5488 | 0.2952 | 0.2214 | 0.2214 | 0.3478 |
| $A_9$ | | | | | | | | | 0 | 0.3476 | 0 | 0.1097 | 0 | 0 | 0.1097 | 0.5697 | 0.0816 | 0.3048 | 0.3048 | 0.0357 |
| $A_{10}$ | | | | | | | | | | 0 | 0.3476 | 0.4024 | 0.3476 | 0.3476 | 0.4024 | 0.5440 | 0.3310 | 0.4571 | 0.4571 | 0.3748 |
| $A_{11}$ | | | | | | | | | | | 0 | 0.1097 | 0 | 0 | 0.1097 | 0.5697 | 0.0816 | 0.3048 | 0.3048 | 0.0357 |
| $A_{12}$ | | | | | | | | | | | | 0 | 0.1097 | 0.1097 | 0 | 0.5607 | 0.1786 | 0.3833 | 0.3833 | 0.0816 |
| $A_{13}$ | | | | | | | | | | | | | 0 | 0 | 0.1096 | 0.5697 | 0.0816 | 0.3048 | 0.3048 | 0.0357 |
| $A_{14}$ | | | | | | | | | | | | | | 0 | 0.1096 | 0.5697 | 0.0816 | 0.3048 | 0.3048 | 0.0357 |
| $A_{15}$ | | | | | | | | | | | | | | | 0 | 0.5607 | 0.1786 | 0.3833 | 0.3833 | 0.0816 |
| $A_{16}$ | | | | | | | | | | | | | | | | 0 | 0.5488 | 0.5440 | 0.5440 | 0.5723 |
| $A_{17}$ | | | | | | | | | | | | | | | | | 0 | 0.3143 | 0.3143 | 0.1097 |
| $A_{18}$ | | | | | | | | | | | | | | | | | | 0 | 0 | 0.3388 |
| $A_{19}$ | | | | | | | | | | | | | | | | | | | 0 | 0.3388 |
| $A_{20}$ | | | | | | | | | | | | | | | | | | | | 0 |

### 4.7.1. Calcul des distances

Pour pouvoir comparer ces indices de similarité, nous allons comparer les structures de dendrogrammes obtenues ci-dessus en utilisant la distance de *Marczewski – Steinhaus* modifiée [151].

Par exemple, pour les deux premiers dendrogrammes de la figure 24, les ensembles des étapes intermédiaires pour la classification $E_{A_1}$et $E_{A_2}$sont donnés. Ils sont impliqués directement dans le calcul de la distance « d » entre $A_1$et $A_2$ comme décrit dans l'exemple du premier paragraphe.

$|E_{A_1}|=|E_{A_2}|=|X|-1=7$.

$E_{A_1} = \{\{3,8\},\{1,6\},\{3,8,5\},\{1,6,2\},\{3,8,5,1,6,2\},\{3,8,5,1,6,2,7\},\{3,8,5,1,6,2,7,4\}\}$

$E_{A_2} = \{\{3,4\},\{1,6\},\{3,4,5\},\{1,6,8\},\{3,4,5,7\},\{3,4,5,7,1,6,8\},\{3,4,5,7,1,6,8,2\}\}$

$d(A_1,A_2)=0.3810$

Les distances calculées entre les différents arbres sont données au tableau16.

### 4.7.2. Exploitation

A partir de la matrice des distances, et en appliquant l'algorithme de classification hiérarchique ascendante, on obtient un méta-dendrogramme (figure 25). A partir de ce dernier et en procédant à des coupures selon les objectifs fixés, nous acquérons un certain nombre de familles des indices.

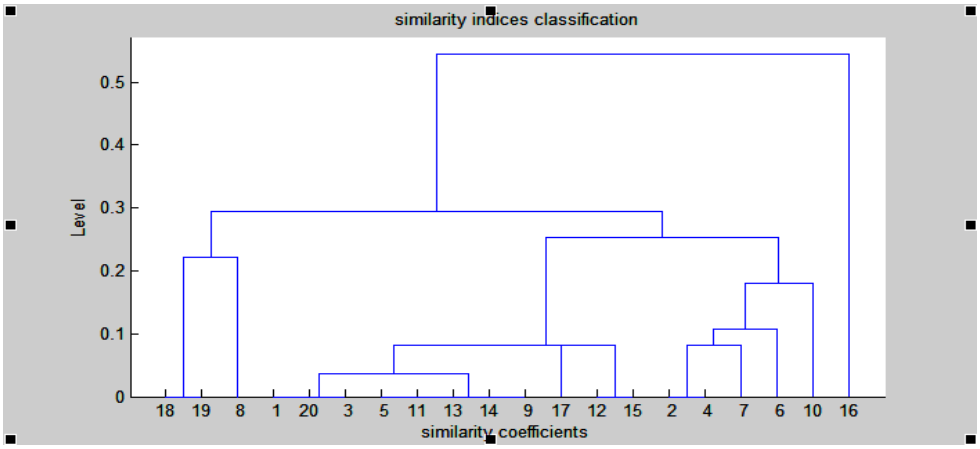


Figure 25 : La classification des indices de similarité

Pour la recherche de familles d'indices, il est nécessaire de couper dans l'arbre de classification à un niveau approprié. On va choisir $0.221 < \alpha < 0.252$ , assurant le minimum d'inertie intra classes.

Pour ce niveau, on obtient ainsi les 4 familles d'indices de la figure 25.

| | |
|---|---|
| Famille $F_1$ | $F_1 = \{18,19,8\}$ |
| Famille $F_2$ | $F_2 = \{1,20,3,5,11,13,14,9,17,12,15\}$ |
| Famille $F_3$ | $F_3 = \{2,4,7,6,10\}$ |
| Famille $F_4$ | $F_4 = \{16\}$ |

Table 17 : Familles cohérentes des indices

En se basant sur [175], nous obtenons les mêmes résultats. Nous pouvons dire que la famille F2 contient les indices les plus performants comme Jaccard, Sorenson, Kulczynski et Sokal and Sneath 2. Par contre, la famille F3 est celle des indices inefficaces à savoir : Hamann, Simple matching, Rogers and Tanimoto.

Dans ce travail, nous avons présenté une procédure qui a pour but fournir un outil rationnel et efficace pour le regroupement et la comparaison de différentes méthodes de classifications.

Nous pouvons aussi situer les méthodes de la littérature les unes aux autres en se basant sur la comparaison des dendrogrammes de ces méthodes. Ceci nous aide à garder les meilleures méthodes pour des problèmes donnés.

Si l'on considère que nous avons plusieurs partitions correspondantes à plusieurs méthodes pour un problème donné, nous pouvons utiliser notre méthode pour trouver la partition consensus. Elle correspond dans notre cas à un dendrogramme consensus réalisé à partir de tous les dendrogrammes trouvés (travail en cours)

# Conclusion générale

Le filtrage collaboratif, qui constitue une des solutions techniques implémentant l'intelligence collective, est déjà utilisé avec succès dans plusieurs systèmes de recommandation. Par exemple, dans les boutiques web qui suggèrent les items à acheter en fonction de nos préférences et celles des personnes ayant le même goût. Actuellement, le FC est considéré comme la technique de recommandation la plus réussite. Cependant, le filtrage collaboratif présente certaines limitations, comme le cas du problème de la rareté, où le nombre de notation spécifiée par l'utilisateur est généralement très faible par rapport au nombre de notations attendues et le problème d'évolutivité, surtout que les utilisateurs et les items sont en augmentation permanente, nécessitant des algorithmes robustes et évolutifs. Un troisième problème à surmonter est celui du problème de démarrage à froid qui surgit lorsque des nouveaux items apparaissent et ne peuvent être point recommandés puisqu'ils n'ont reçue encore aucune note.

Nos contributions au niveau de cette thèse présentent des solutions pour ces défaillances très souvent rencontrées dans le filtrage collaboratif et peuvent être résumées comme suit :

Premièrement, nous avons proposé le filtrage collaboratif multicritères basé-item (MCCR) pour résoudre le problème de la rareté et rendre la recommandation plus précise. En effet, nous avons enrichi son processus de recommandation par d'autres critères que les notes, en prenant en compte les données caractéristiques des items au niveau du calcul de la similarité des items avec l'item cible. Nous avons opté pour l'utilisation des notes des items et leurs caractéristiques lors du calcul de similarité, de sorte que ces notes ne soient pas statiques. Pour ce faire, nous avons utilisé une combinaison convexe de deux mesures de similarité, la première se base sur les notations des items et la deuxième se base sur leurs attributs. La pondération de chacune de ces deux mesures de similarité dépend du temps. Cette combinaison convexe introduit le facteur temps qui correspond au moment de l'évaluation de l'item par l'utilisateur. Ce facteur s'avère très important, il permet de réduire l'influence des anciennes notes et contribue dans l'amélioration des performances du FC. Les résultats obtenus sont significatifs.

Deuxièmement, nous avons tenté de répondre au besoin d'amélioration de FC en ce qui concerne la rareté et l'évolutivité en utilisant des nouvelles techniques de classification croisée. L'utilisation de cette technique est plus robuste pour résoudre ce problème, et c'est un moyen viable pour augmenter l'évolutivité tout en conservant une bonne qualité de recommandation. Pour cela, nous avons proposé la méthode de regroupement STGM basée sur l'algorithme BEA, cet algorithme forme des blocs avec une forte énergie maximale sur des donnes éparses. Nous l'avons adapté pour un meilleur co-partitionnement de matrices d'utilisateurs et d'items pour les systèmes de données creuses. Cette méthode puissante donne un groupement efficace et précis et les résultats sont remarquables. Les communautés sont automatiquement crées sans aucune nécessité de spécifier leur nombre, alors qu'il est exigé dans les autres méthodes de partitionnement. Cette méthode donne aussi une solution au problème de démarrage à froid basée sur les items ou les utilisateurs clés.

Troisièmement, vu l'importance de graphes et leurs utilisations dans de nombreuses applications Web, on a pu utiliser cet outil pour trouver des structures de communautés. Ces dernières servent à leurs tours de proposer aux utilisateurs des contenus personnalisés. Dans certains types d'applications, il peut être préférable, de proposer une combinaison d'items au lieu d'un seul. Ainsi, nous avons proposé l'approche RMCS qui consiste à rechercher une liste d'items sans calcul de prédiction en utilisant les graphes. La méthode a été testée avec succès sur la base de données MovieLens.

Quatrièmement, les algorithmes de recommandation s'appuient sur différentes mesures de similarité. Il est à noter que le choix de ces mesures est d'une importance extrême. Dans ce contexte, nous avons proposé la méthode de comparaison des mesures de similarité LATCCM basée sur la structure hiérarchique qui donne une procédure rationnel et efficace pour le regroupement permettant d'identifier les classes semblables.

Si dans ces travaux de recherche les problèmes de la rareté, de l'évolutivité, et de démarrage à froid ont été appréhendés, d'autres problèmes persistent et restent encore ouvertes. Ceci sera notre préoccupation dans les perspectives de recherche résumées dans les axes suivants

Pour la méthode de regroupement STGM:

- Continuer le travail au niveau de l'extraction de blocs non diagonaux puis passer à la phase de recommandation après l'obtention de ces derniers.

Pour la procédure de comparaison des indices de similarité LATCCM:

- Améliorer la distance entre arbres en établissant une relation d'ordre ou de niveau sur les classes de dendrogrammes de ces mesures de similarité.
- Appliquer la méthode dans différents contextes dont le but est de la généraliser pour comparer les méthodes de classifications pour n'importe quelles données partitionnées.

Analyse probabiliste de recommandation en déterminant :

- La probabilité de mauvaise recommandation pour la méthode RMCS proposée.
- Lien entre mauvais classement d'un utilisateur ou d'un item et la dissimilarité
- Lien entre la probabilité d'une bonne recommandation et le degré de similitude des classes.

# BIBLIOGRAPHIE


[1] Hoffman, D.L, Novak. A New Marketing Paradigm for Electronic Commerce, Information Society,Vol.13, N1, p.43-55, 1997

[2] Andreasen, A.R. Attitudes and Custmer behavior, A decision Model, Perspectives in Consumer Behavior, Glenview, Il:Scott, Foresman and Company, 1968

[3] Lutz, R. J Reilly, P.J. An Exploration of the Effects of Perceived Social and Performance Risk on Consumer Information Acquisition, Advances in Consumer Research, Urbana, IL: Association for Consumer Research, Vol.1, p.393-405, 1974

[4] Mitra, K. Reiss et Cappella, L.M. An Examination of Perceived Risk, Information Search and Behavioral Intentions in Search, Experience and Credence services , The Journal of services Marketing,Vol.13, N3, p.208-228, 1999

[5] N. Wingfield and J. Pereira. Amazon Uses Faux Suggestions to Promote New Clothing Store,Wall Street Journal, 2002

[6] P. Resnick and H. R. Varian. Recommender Systems. Communications of the ACM, 1997

[7] Koren, Y. Bell, R.Volinsky. Matrix factorization techniques for recommender systems. IEEE Computer, 2009

[8] Mahmood, T.Ricci . Improving recommender systems with adaptive conversational strategies. Hypertext, 2009

[9] McSherry, F, Mironov. Differentially private recommender systems: building privacy into the net. ACM international conference on Knowledge discovery and data mining, 2009

[10] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor. Recommender Systems Handbook. Springer, 2011

[11] K. Nageswara Rao and V.G.Talwar. Application Domain and Functional Classification of Recommender Systems a Survey. Desidoc journal of library and information technology, 2008

[12] E. R.Omiecinski. Alternative Interest Measures for Mining Associations in Databases. IEEE Trans. on Knowl. and Data Eng, 2003

[13] Burke. The Adaptive Web. chapter Hybrid Web Recommender Systems, Springer-Verlag, 2007

[14] Y. Zhang and J. R. Jiao. An Associative Classification-Based Recommendation System for Personalization in B2C e-commerce Applications. Expert Syst. Appl, 2007

[15] Katrien Verbert, Nikos Manouselis, Xavier Ochoa, Martin Wolpers, Hendrik Drachsler, Ivana Bosnic, Erik Duval. Context-Aware Recommender Systems for Learning: A Survey and Future Challenges,IEEE Transactions on Learning Technologies, Vol.5, N4, p.318-335, Fourth Quarter 2012

[16] Jiliang Tang, Huiji Gao, Xia Hu and Huan Liu. Context-aware review helpfulness rating prediction, RecSys, 2013

[17] Negar, Hariri DePaul - Query-Driven Context Aware Recommendation, Proceedings of the 7th ACM conference on Recommender systems , 2013

[18] J. B. Schafer, A. J. Konstan, and J. Riedl. E-Commerce Recommendation Applications Data Mining and Knowledge Discovery, 2001

[19] RuLong Zhu, SongJie Gong. Analyzing of Collaborative Filtering Using Clustering Technology, in Procs of ISECS International Colloquium on Computing, Communication, Control and Management, 2009

[20] Baeza-Yates, R. Ribeiro-Neto, B. Modern Information Retrieval. Addison-Wesley.1999

[21] J. Breese, D. Heckerman, and C. Kadie, Empirical analysis of predictive algorithms for collaborative filtering, in Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, 1998

[22] G. Salton . Automatic Text Processing. Addison-Wesley, 1989

[23] D. Mladenic. Machine learning used by PersonalWebWatcher. Workshop on Machine Learning and Intelligent Agents, 1999

[24] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. Using Collaborative Filtering to Weave an Information Tapestry. Commun. ACM, 1992

[25] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. GroupLens: an Open Architecture for Collaborative Filtering of Netnews. In Proceedings of the ACM conference on computer supported cooperative work,1994

[26] R. M. Bell and Y. Koren. Lessons From the Netflix Prize Challenge. SIGKDD Explor. Newsl, 2007

[27] G. Linden, B. Smith, and J. York. Amazon.com Recommendations: Item-to-Item Collaborative Filtering. IEEE Internet Computing, 2003

[28] Li Pu, Boi Faltings: Understanding and improving relational matrix factorization in recommender systems, Proceedings of the 7th ACM Conference on Recommender Systems, 2013

[29] Oluwasanmi Koyejo, Sreangsu Acharyya, and Joydeep Ghosh, Retargeted matrix factorization for collaborative filtering. RecSys, ACM, 2013

[30] Jason Weston, Ron J. Weiss, Hector Yee, Nonlinear latent factorization by embedding multiple user interests, Proceedings of ACM Recommender Systems, 2013

[31] Paolo Cremonesi, Yehuda Koren , Roberto Turrin, Performance of recommender algorithms on top-n recommendation tasks, Proceedings of the 7th ACM conference on Recommender systems, 2013

[32] YaE Dai, SongJie Gong. Personalized Recommendation Algorithm using User Demography Information, IEEE Computer Society Press, 2009

[33] SongJie Gong, XiaoYan Shi. A Collaborative Recommender Combining Item Rating Similarity and Item Attribute Similarity, IEEE Computer Society Press, 2008

[34] Long Yun, Yan Yang, Jing wang, Ge Zhu. Improving Rating Estimation in Recommender Using Demographic Data and Expert Opinions,Software Engineering and Service Science, 2011

[35] Marius Kaminskas, Francesco Ricci, Markus Schedl**.** Location-aware Music recommendation Using Auto-Tagging and Hybrid Matching, Proceedings of the 7th ACM conference on Recommender systems, 2013

[36] Bo Hu, Martin Ester. Spatial Topic Modeling in Online Social Media for Location Recommendation. Proceedings of the 7th ACM conference on Recommender systems, 2013

[37] Huiji Gao, Jiliang Tang, Xia Hu, and Huan Liu. Computer, Exploring Temporal Effects for Location Recommendation on Location-Based Social Networks. Proceedings of ACM Recommender Systems, 2013

[38] T. Y. Tang, P. Winoto, and K. C. C. Chan. On the temporal analysis for improved hybrid recommendations. In Web Intelligence. Proceedings. IEEE/WIC International, 2003

[39] L. Terveen, J. McMackin, B. Amento, and W. Hill. Specifying preferences based on user history. In Conference on Human Factors in Computing Systems, Minneapolis, Minnesota, USA, 2002

[40] K. Sugiyama, K. Hatano, and M. Yoshikawa. Adaptive web search based on user profile constructed without any effort from users. In Proceedings of the 13th International Conference on World Wide Web, 2004

[41] Y. Zhao, C. Zhang, and S. Zhang. A recent-biased dimension reduction technique for time series data. In Advances in Knowledge Discovery and Data Mining: 9th Pacific-Asia Conference, Lecture Notes in Computer Science, 2005

[42] Y Ding, Xue Li, Time Weight Collaborative Filtering, Proceedings of the 14th ACM International Conference on Information and knowledge management, 2005

[43] K. Ali and W. v. Stam. Tivo. Making show recommendations using a distributed collaborative filtering architecture. In Conference on Knowledge Discovery in Data, Seattle, USA, 2004

[44] M. Xu, S. Berkovsky, S. Ardon, S. Triukose, A. Mahanti and I. Koprinska. Catch-up TV Recommendations: show old favourites and find new ones, Proc. ACM Recommender Systems, 2013

[45] G. Adomavicius and A. Tuzhilin. Toward the Next Generation of Recommender Systems : A survey of the State-of-the-Art and Possible Extensions. IEEE Transactions On Knowledge and Data Engineering, 2005

[46] A. S. Das, M. Datar, A. Garg, and S. Rajaram. Google News Personalization : Scalable Online Collaborative Filtering. In Proceedings of the 16th international conference on World Wide Web, ACM, 2007

[47] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl. GroupLens : Applying Collaborative Filtering to Usenet News. Commun. ACM, 1997

[48] M. F. Hornik, P.Tamayo, Extending Recommender Systems for Disjoint User/Item Sets: The Conference Recommendation Problem , Vol. 24, N.8, IEEE Transaction on Knowledge and Data Engineering, 2012

[49] M. Deshpande and G. Karypis. Item Based Top-N Recommendation Algorithms. ACM Transactions on Information Systems, 2004

[50] http://www.last.fm

[51] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl. Item-Based Collaborative Filtering Recommendation Algorithms. Proceedings of the 10th international conference on World Wide Web , 2001

[52] S. Castagnos. Modélisation de Comportements et Apprentissage Stochastique non Supervisé de

Stratégies d'Interactions Sociales au Sein de Systèmes Temps Réels de Recherche et d'Accès à l'Information. Thèse, IAEM-Lorraine, 2008

[53] L. Candillier, F. Meyer, and M. Boullé. Comparing State-of-the-Art Collaborative Filtering Systems. In Proceedings of the 5th international conference on Machine Learning and Data Mining in Pattern Recognition, Springer-Verlag, 2007

[54] X. Su and T. M. Khoshgoftaar. A Survey of Collaborative Filtering Techniques. Adv. In in Proceedings of Artif Intell, 2009

[55] Hofmann, T. Collaborative filtering via Gaussian probabilistic latent semantic analysis. on Research and Development in Information Retrieval, 2003

[56] Salakhutdinov, R. Mnih, A. Hinton. Restricted boltzmann machines for collaborative filtering. In Proceedings of 24th International Conference on Machine Learning, 2007

[57] Grcar, M. Fortuna, B. Mladenic, M.Grobelnik. K-NN Versus SVM in the collaborative filtering framework. Data Science and Classification, 2006

[58] Bell, R. Koren, Y.Volinsky and al. Modeling relationships at multiple scales to improve accuracy of large recommender systems. In Proceedings of Conf. on Knowledge Discovery and Data Mining , 2007

[59] Koren, Y. Factorization meets the neighborhood, a multifaceted collaborative filtering model. In Proceedings of Conf. on Knowledge Discovery and Data Mining , 2008

[60] C. Basu, H. Hirsh, and W. Cohen. Recommendation as classification: using social and content-based information in recommendation. In Proceedings of Conference on Artificial Intelligence, USA 1998.

[61] J. Han and M. Kamber. Data Mining: Concepts and Techniques, 2001

[62] X. Su, M. Kubat, M. A. Tapia, and C. Hu. Query size estimation using clustering techniques. In Proceedings of Conference on Tools with Artificial Intelligence, 2005

[63] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In Symposium on Math, 1967

[64] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of Conference on Knowledge Discovery and Data Mining, 1996

[65] M. Ankerst, M. Breunig, H.P. Kriegel and J. Sander. OPTICS: ordering points to identify the clustering structure. In Proceedings of ACM SIGMOD Conference, 1999

[66] T. Zhang, R. Ramakrishnan, and M. Livny. BIRCH: An efficient data clustering method for very large databases. ACM SIGMOD Conference, vol. 25, p.103–114, 1996

[67] M. O'Connor and J. Herlocker. Clustering items for collaborative filtering. In Proceedings of the ACM SIGIR Workshop on Recommender Systems, 1999.

[68] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Riedl. Recommender systems for large-scale E-commerce: scalable neighborhood formation using clustering. In Proceedings of the International Conference on Computer and Information Technology, 2002

[69] S. H. S. Chee, J. Han, and K. Wang. RecTree: An efficient collaborative filtering method. In Proceedings of the Conference on Data Warehousing and Knowledge Discovery, 2001

[70] T. George, S. Merugu. A scalable collaborative filtering framework based on co-clustering. In Proceedings of the IEEE ICDM conference 2005

[71] Panagiotis Symeonidis, Alexandros Nanopoulos, Apostolos Papadopoulos, Yannis Manolopoulos. Nearest-Biclusters Collaborative Filtering. WEBKDD 2006

[72] Panagiotis Symeonidis, Alexandros Nanopoulos, Apostolos N. Papadopoulos, Yannis Manolopoulos. Nearest- biclusters collaboartive filtering based on constant and coherent values. Inf Retrieval 2007

[73] L. H. Ungar and D. P. Foster. Clustering methods for collaborative filtering. AAAI Press, 1998.

[74] S. Geman and D. Geman, Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images, Transactions on Pattern Analysis and Machine Intelligence, Vol. 6, N6, p.721–741, 1984

[75] L. Si and R. Jin, Flexible mixture model for collaborative filtering. In Proceedings of the 20th International Conference on Machine Learning, Vol. 2, p.704–711, 2003

[76] T. Hofmann and J. Puzicha. Latent class models for collaborative filtering. In Proceedings of the Conference on Artificial Intelligence, p.688–693, 1999

[77] J. Kelleher and D. Bridge. Rectree centroid: An accurate, scalable collaborative recommender. In Proceedings of the Fourteenth Irish conference on artificial Intelligence and Cognitive Science, 2003

[78] Rashid, A.M. Lam, S.K., Karypis, G.,Riedl, J,ClustKNN: A Highly Scalable Hybrid Model and Memory-Based CF Algorithm. WEBKDD, 2006

[79] AM. Rashid,, S. K. Lam, A. LaPitz, G. Karypis and J. Riedl. Towards a Scalable kNN CF Algorithm:Exploring Effective Applications of Clustering. LNCS Vol 4811, p.147-166,

Advances in Proceedings of the Web Mining and Web Usage Analysis ,2007

[80] RuLong Zhu, SongJie Gong. Analyzing of Collaborative Filtering Using Clustering Technology. In Proceedings of the Colloquium on Computing, Communication, Control, and Management, 2009

[81] D. Billsus and M.J. Pazzani; learning collaborative information fillters. In Proceedings of the Conference on Machine Learning, 1998

[82] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. Neural information processing Systems 2008

[83] N. D. Lawrence and R. Urtasun. Non-linear matrix factorization with gaussian processes. In Proceedings of the Conference on Machine Learning, 2009

[84] R. Salakhutdinov and A. Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. In Proceedings of the Conference on Machine Learning, 2008

[85] K. Yu, S. Zhu, J. Lafferty, and Y.Gong. Fast nonparametric matrix factorization for Large Scale Collaborative filtering. In Proceedings of the SIGIR conference on Research and development in information retrieval, 2009

[86] D.D. Lee, Seung H. S., Learning the parts of objects by non-negative matrix factorization , Nature, Vol.401, p.788-791, 1999

[87] J. Yoo, S. Choi. Orthogonal nonnegative matrix tri-factorization for co-clustering: Multiplicative updates on Stiefel manifolds, Information Processing and Management, 2010

[88] L. Shi. Trading-off Among Accuracy, Similarity, Diversity, and Long-tail: A Graph-based Recommendation Approach. ACM Recommender Systems, 2013

[89] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Riedl, Analysis of recommendation algorithms for E-commerce, ACM E-Commerce, 2000

[90] G.R. Xue, C. Lin, Q. Yang, et al. Scalable collaborative filtering using cluster-based smoothing. In Proceedings of the ACM SIGIR Conference, 2005

[91] A. Ng and M. Jordan, Pegasus. A policy search method for large MDPs and POMDPs. In Proceedings of the Conference on Uncertainty in Artificial Intelligence, 2000

[92] Y. Zhuang, W. Chin, Y. Juan and C. Lin. A fast parallel SGD for matrix factorization in shared memory systems,Proc. ACM Recommender Systems, 2013

[93] Ali, K.. van Stam, W. TiVo: Making Show Recommendations Using a Distributed Collaborative Filtering Architecture. In Proceedings of the ACM Conference on Knowledge

Discovery and Data Mining, 2004

[94] www.netflix.com

[95] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich. Recommender Systems An Introduction. Cambridge University Press, 2011

[96] D. Poirier , Isabelle Tellier et Patrick Gallinari. Des Textes Communautaires à la Recommandation. Thèse, Orléans, Paris 6, 2011

[97] Thomas PITON. Une Méthodologie de Recommandations Produits Fondée sur l'Actionnabilité et l'Intérêt Économique des Clients Application à la Gestion de la Relation Client du groupe VM Matériaux.Thèse à l'École polytechnique de l'Université de Nantes, 2011

[98] E. Rich. Readings in Intelligent User Interfaces. chapter User Modeling via Stereotypes. Morgan Kaufmann Publishers Inc, 1998

[99] W. Hill, L. Stead, M. Rosenstein, and G. Furnas. Recommending and Evaluating Choices in a Virtual Community of Use. ACM Press/Addison-Wesley Publishing Co, 1995

[100] U. Shardanand and P. Maes. Social Information Filtering. Algorithms for Automating Wordof Mouth. ACM Press/Addison-Wesley Publishing Co, 1995

[101] M. Montaner, B. López, and J. L. De La Rosa. A Taxonomy of Recommender Agents on the Internet. Artif. Intell. Rev, 2003

[102] Z.Wan. Personalized Tourism Information System in Mobile Commerce. Management of e-Commerce and e-Government, 2009

[103] M. Hosseini-Pozveh, M. A. Nematbakhsh, and N. Movahhedinia. A Multidimensional Approach for Context-Aware Recommendation in Mobile Commerce. Informal publication, 2009

[104] C. A. Thompson, M. H. Goker, and P. Langley. A Personalized System for Conversational Recommendations. Journal of Artificial Intelligence Research, 2004

[105] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins. Eigentaste. A Constant Time Collaborative Filtering Algorithm. Inf. Retr, 2001

[106] M. Balabanovic and Y. Shoham. Fab, Content-Based. Collaborative Recommendation. Communication of the ACM, 1997

[107] B. Krulwich and C. Burkey. The InfoFinder Agent: Learning User Interests Through Heuristic Phrase Extraction. IEEE Expert, 1997

[108] J. L. Herlocker, Konstan, A. Borchers, and J. Riedl. An Algorithmic Framework for Performing Collaborative Filtering. ACM SIGIR conference on Research and development in information retrieval, 1999

[109] M. Bilgic. Explaining Recommendations, atisfaction vs. Promotion. Beyond Personalization, 2005

[110] C. H. Cai, A. W. C. Fu, C. H. Cheng, and W. W. Kwong. Mining Association Rules with Weighted Items. In Proceedings of the International Symposium on Database Engineering and Applications, IEEE Computer Society, 1998

[111] J. L. Herlocker, J. A. Konstan, and J. Riedl. Explaining Collaborative Filtering Recommendations, 2000

[112] B. G. Buchanan and E.H. Shortliffe. Rule Based Expert Systems, The Mycin Experiments of the Stanford Heuristic Programming Project, Addison-Wesley Longman Publishing Co, 1985

[113] W.C. Hu. Adaptive Web Browsing Using Web Mining Technologies for Internet Enabled Mobile Handheld Devices, Emerging Trends and Challenges in Information Technology Management, 2006

[114] P. Pu and L. Chen. Trust Building with Explanation Interfaces. In proceedings of the 11th International Conference on Intelligent User Interfaces, ACM, 2006

[115] J. Bu, S. Tan, C. Chen, C. Wang, H. Wu, L. Zhang, and X. He. Music Recommendation by UnifiedHypergraph : Combining Social Media Information and Music Content. In Proceedings of the International Conference on Multimedia, ACM, 2010

[116] R. Burke. Knowledge-Based Recommender Systems, In Encyclopedia of Library and Information Systems, 2000

[117] M. Czarkowski. A Scrutable Adaptive Hypertext. In Proceedings of the Fourth Workshop on Empirical Evaluation of Adaptive Systems, held at the 10th International Conference on User Modeling, p.384–387, Springer, 2005

[118] D. McSherry. Explanation in Recommender Systems. Artif. Intell. Rev, 2005

[119] M. Jamali and M. Ester. TrustWalker , a Random Walk Model for Combining Trust-Based and Item-Based Recommendation. In Proceedings of the 15th ACM International Conference on Knowledge Discovery and Data mining, ACM, 2009

[120] www.imdb.com

[121] www.ebay.com

[122] www.alibaba.com

[123] https://play.google.com

[124] http://www.apple.com

[125] L.Ye, and M.Spetsakis .Clustering on unobserved data using mixture of Gaussians. Technical report, York University, 2003

[126] E. Diday, J. Lemaire, J. Pouget, and F.Testu. Un algorithme de type nuées dynamiques, Revue Modulad, 1983

[127] Berkhin, P. 2002. Survey of clustering data mining techniques. technical report, Accrue Software, 2002

[128] M. Ester, H.P. kriegel, J.Sander, and X. Xu. A density based algorithm for discovering clusters in large spatial databases with noise. In 2nd International Conference on knowledge Discovery and data Mining, 1996

[129] P. Brezellec, Didier, G. Gizmo. Un algorithme de grille cherchant des clusters homogenes. 3ème Conference Francophone sur l'apprentissage automatique, 2001

[130] Pettie, S. Ramachandran, V. An optimal minimum spaning tree algorithm. In automata, languages and programming, 2000

[131] Verma, D. and M. Meila. A comparaison of spectral clustering algorithms. Technical report, University of washington, 2003

[132] C. Ding, T. Li, W. Peng, and H. Park. Orthogonal Nonnegative Matrix Tri-factorizations for Clustering. Proc. of SIGKDD, 2006

[133] P. O. Hoyer. Non-negative matrix factorization with sparseness constraints. Machine Learning Research , 2004

[134] D.D.Lee, H. S. Seung. Learning the parts of objects by non-negative matrix factorization, Nature, Vol. 401, p.788-791, 1999

[135] J.F. Pessiot, V. Truong, N. Usunier, M.R. Amini et P. Gallinari. Factorisation en matrices non négatives pour le filtrage collaboratif . Actes de CORIA, 2006.

[136] F. Shahnaz, M. Berry, P. Pauca, R.Plemmons . Document clustering using nonnegative matrix factorization. Information Processing and Management, 2006

[137] Walter Jean Luc, Mutel Bernard. Study of group technology implementation: Application in the Holeg international firm, thèse, Université de Mulhouse, France

[138] W.T. Mccormick, S. B. Deutsch, J. J. Martin and P. J. Schweitzer. Identification of Data Structures and Relationships by Matrix reorderingTechniques. Research, Institute for Defense Analyses, Arlington, Va., 1969

[139] William T. McCormick Jr., Paul J. Schweitzer and Thomas W. White. Problem Decomposition and Data Reorganization by a Clustering Technique. Operations Research, Vol.20, No. 5, p. 993-1009, 1972

[140] G. Shani, M. Chickering, and C. Meek. Mining Recommendations from the Web. In Proceedings of the ACM conference on Recommender systems, RecSys, 2008

[141] S. Castagnos and A. Boyer. A Client/Server User-Based Collaborative Filtering Algorithm: Model and Implementation. In Proceeding of 17th European Conference on Artificial Intelligence The Netherlands, IOS Press, 2006

[142] F. Garcin, C. Dimitrakakis and B. Faltings. Personalized news recommendation with context trees. Proceeding of ACM Recommender Systems , 2013

[143] L. Euler, Solutio Problematis Ad geometriam Situs Pertinentis, Commenrarii Academiae Scientiarum Imperialis Petropolitanae , 1736

[144] K. Appel et W. Haken. Every planar map is 4-colorable. Bulletin of the AMS, Volume 82, p.711-712, 1976

[145] N. Biggs, E. Lloyd et R. Wilson, Graph Theory, ClarendonPress Oxford, 1976

[146] M. Minoux et G. Bartnik, Graphes, algorithmes, logiciels, Dunod Informatique, Bordas Paris, 1986

[147] Joseph kruskal. KRUSKAL Algorithm's. In proceedings of the American Mathematical society, 1956

[148] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu. A framework for projected clustering of high dimensional data streams. In VLDB, 2004

[149] Q. Li and M. Zhou. Research and design of an efficient collaborative filtering predication algorithm. In Parallel and Distributed Computing. Applications and Technologies, 2003

[150] Karima Mouhoubi, Lucas Latocart, Celine Rouveirol. L'extraction de motifs ensemblistes dans les contextes bruités, dans la Conférence Francophone sur l'Apprentissage Automatique , 2012

[151] M. Karonski and Z. Palka . On MarZeweski-Steinhaus Distance between Hypergraphs,

Expositione Mathematicae, 1977

[152] Kendall M.G, Stuart A. The Advanced Theory of Statistics, Griffin, Londre, 1961

[153] F. Marcotorchino. Utilisation des Comparaisons par Paires en Statistique des Contingences. Etude du Centre Scientifique IBM France, 1984

[154] L. Hubert, P. Arabie. Comparing Partitions. Journal of Classification, Vol. 2, p.193-198, 1985

[155] Chavent, M., et al., Critère de Rand Asymétrique, in Proceedings SFC, 8èmes rencontres de la Société Francophone de Classification, Pointe à Pitre, 2001

[156] Robert P., Escoufier, Y. A Unifying Tool for Linear Multivariate Statistical Methods, the RV-coefficient. Appl. Statist., Vol. 25, p.257-265, 1976

[157] S.Janson , J.Vegelius. The J-index as a Measure of Association for Nominal Scale Response Agreement. Applied psychological measurement, Vol. 16, p. 243-250, 1982

[158] J.Cohen . A Coefficient of Agreement for Nominal Scales. Educ. Psychol. Meas., Vol.20, p.27-46, 1960

[159] D.Stewart , W.Love , A General Canonical Correlation index, Psychological Bulletin, Vol.70, p. 160- 163, 1968

[160] Popping, R. Traces of agreement. On the Dot- Product As a Coefficient of Agreement. Quality and Quantity, Vol.17, N1, p.1-18, 1983

[161] Boujault A., Favrel J., Baptiste P. et al. , Ingénierie des systèmes flexibles d'assemblage, Compte-rendu de fin de recherche MRT sur l'ingénierie des systèmes flexibles d'assemblage, Institut Productique Besancon, 1991.

[162] B. Miller, I. Albert, S. Lam, J. Konstan, J. Rield. Movielens unplugged: experiences with an occasionally connected recommender systems. In:Proc. Internat. Conf. Intelligent User Interfaces, 2002

[163] Ryosuke LA Watanabe, Enrique Morett and Edgar E Vallejo. Inferring modules of functionally interacting proteins using the Bond Energy Al Algorithm. BMC Bioinformatics , 2008

[164] http://recsys.acm.org/

[165] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating Collaborative Filtering Recommender Systems. ACM Trans. Inf. Syst., 2004

[166] B. N. Miller, J. A. Konstan, and J. Riedl. PocketLens .Toward a Personal Recommender

System. ACM Trans. Inf. Syst., 2004

[167] M. Pazzani. A Framework for Collaborative, Content-Based and Demographic Filtering. Artificial Intelligence Rev., 1999

[168] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Application of Dimensionality Reduction in Recommender Systems . Proc. ACM WebKDD Workshop, 2000

[169] Z. Huang, H. Chen, and D. Zeng. Applying Associative Retrieval Techniques to Alleviate the Sparsity Problem in Collaborative Filtering. ACM Trans. Information Systems, Vol. 22, N1, p.116- 142, 2004

[170] Romain Picot Clémente, Une architecture générique de Systèmes de recommandation de combinaison d'items. Application au domaine du tourisme, Thèse, l'Université de Bourgogne, 2011

[171] Aggarwal, J. L.Wolf, K.L.Wu, and P. S. Yu. Horting Hatches an Egg , A New Graph-Theoretic Approach to Collaborative Filtering. In In Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge discovery and data mining, ACM Press, 1999.

[172] N.Benchettara, R.Kanawati, C. Rouveirol. Supervised Machine Learning applied to Link Prediction in Bipartite Social Networks. Advances in Social Networks Analysis and Mining , 2010

[173] Fei Wang ,Sheng Ma, Liuzhong Yang, Tao Li. Recommendation on Item Graphs. Sixth IEEE International Conference on Data Mining, 2006

[174] S. Maunendra, Desarkar Sudeshna, Sarkar Pabitra Mitra . Aggregating Preference Graphs for Collaborative Rating Prediction. In RecSys,Proceedings of the fourth ACM Conference on Recommender systems , 2010

[175] Yasuda, Y. Yin et K., Similarity coefficient methods applied to the cell formation problem: a comparative investigation. Computers & Industrial Engineering , 2005

[176] Najma Hamzaoui, Abdelfettah Sedqui, Abdelouahid Lyhyaoui, Multi-Criteria Collaborative Recommender, International Journal of Computational Linguistics Research Vol. 3 N 3 September 2012

[177] Ouafae Baida, Najma Hamzaoui, Abdelfettah Sedqui, Abdelouahid Lyhyaoui, Recomendation based on Co-clustering Algorithm, Co-dissimilarity and Spanning Tree, International Journal of Computational Linguistics Research Vol. 3 N3 September 2012

[178] Najma Hamzaoui, Maha Akbib, Wafae Baida, Abdelfettah Sedqui, Abdelouahid Lyhyaoui,

Logical Actions of trees for the comparison of Classification Methods, International Journal of Advanced Research in Computer and Communication Engineering,Vol. 2, Issue 6, April 2013

[179] Najma Hamzaoui, Wafae Baida, Abdelfettah Sedqui, Abdelouahid Lyhyaoui, Straight Through Grouping for Recommender System, International Journal of Application or Innovation in Engineering & Management, Vol. 2, Issue 9, September 2013

# SUMMARY

If technology is a natural extension of the user behavior, the adaptation to different technical solutions should ideally allow to simplify human activities in their original forms. The natural human behavior of a person is to learn from the experiences of others. This type of induction is the essence of the collective intelligence of the community to meet the need of the user. Collective intelligence and sensitivity to context, are two technologies used in intelligent systems. The first allows you to learn and derive new information from the composition of the experiences of their users. The second makes these systems capable of reasoning about their abstract knowledge about what is happening.

All intelligent agents as recommender systems can obtain personalized information. Indeed, these are systems that are designed to help users to find interesting items, provide relevant information that meets their satisfaction and their real needs through a process of collecting, filtering and recommendation.

With the huge amount of information circulating on the Web, it is increasingly difficult to quickly and efficiently find necessary and useful information. However, with the emergence of recommender systems over the 90 years, reducing information overload has become easier.

The basic idea when developing the system recommendation was simply to observe that the user tends to rely on recommendations from other users in making the decision. Collaborative filtering is regarded as the most successful technique recommendation. Indeed, it is the most used in recommender systems for e-commerce. This technique allows to recommend items to a user based on the user profiles that are closest to him.

Nowadays, the latest generation of Collaborative Filtering methods requires further improvements to make the most efficient and accurate recommendation. Most existing collaborative filtering algorithms still suffer from the problem of scarcity, scalability and cold start.

In this thesis, we propose solutions for these problems in collaborative filtering via four methods. The Multi-item based collaborative filtering (MROC, Collaborative Multi-Criteria Recommender) and new theoretical formulation to improve the recommendation and solve the problem of scarcity. The clustering method (STEM, Straight Through Grouping Model) to get communities, solve the problem of scalability and cold start. The method of recommendation from a list of items without prediction calculation based on the co-dissimilarity and the minimum spanning tree (RMCS, Recommendation model based on co-dissimilarity and Spanning Tree) based on graph theory. Finally, the proposed measures for

the classification of similarities (Logical Shares of trees for the comparison of Classification Methods, LATCCM) used in recommender systems to improve their performance method.

**Keywords:**

Recommender System, Collaborative Filtering, K-PPV Methods, Clustering Techniques, Graph Theory