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

**UNIVERSITE IBN KHALDOUN – TIARET**

# MEMOIRE

Présenté à :

FACULTÉ MATHEMATIQUES ET INFORMATIQUE
DÉPARTEMENT D'INFORMATIQUE

Pour l'obtention du diplôme de :

MASTER

Spécialité : Réseaux et Télécommunications

Par :

❖ ***SAADI YAZID***
❖ ***OULDRABAH MOHAMED AMINE***

Sur le thème

## *Mise en cluster évolutive pour la prédiction de note basée sur le filtrage collaboratif*

Soutenu publiquement le .. / .. / 2020 à Tiaret devant le jury composé de :

| | | | |
|---|---|---|---|
| Mr kharoubi Sahraoui | Grade | Université MCB | Président |
| Mr Kouadria Abderrahmane | Grade | Université MAA | Encadreur |
| Mr Allem Abdelkader | Grade | Université MAA | Examinateur |

2020-2021

# *Remerciement*

*Avec l'aide d'Allah le tout puissant nous avons pu achever ce modeste travail malgré les difficultés rencontrées avec l'événement qu'a vécu le monde : la pandémie du covid- 19 et ses circonstances,*

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

*En premier lieu, nous tenons à remercier M. KOUADRIA Abderrahmane, enseignent à l'université Ibn-khaldoun-Tiaret. En tant qu'encadreur, il était disponible, à notre écoute et il nous a consacré tout son temps. Nous le remercions de nous avoir bien orienté à trouver les solutions adéquates pour faire avancer ce travail. Enfin nous avouons que sans lui, ce mémoire n'aurait jamais vu le jour.*

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

*Nous remercions nos chers parents pour leurs contributions, leurs soutiens et leurs patiences.*

*Merci*

# *Résumé*

Les systèmes de recommandation (SR) sont utilisés pour aider les gens à réduire le temps qu'ils passent à découvrir les articles qui les intéressent. Le filtrage collaboratif (FC) est l'une des techniques les plus efficaces en SR. La rareté des données rend le résultat de la recommandation inexact.

Dans ce travail, nous utilisons une méthode de classification évolutive, où le but est de regrouper les utilisateurs qui possèdent un intérêt similaire dans le même cluster et de recommander des items aux utilisateurs qu'ils pourraient aimer.

Au préalable, la distance d'attribut utilisateur est calculée, selon le modèle de clustering de réseau construit, où les états des utilisateurs évoluent avec le temps. Les états des utilisateurs seraient stables après une certaine période d'itération. À la lumière des états stables des utilisateurs, ces derniers sont regroupés en plusieurs groupes (cluster).

Deuxièmement, nous remplissons la matrice de notation en fonction du temps de notation et du genre de films. Troisièmement, le filtrage collaboratif basé sur l'utilisateur est adopté dans chaque cluster. Les similarités entre les utilisateurs uniquement dans le même cluster sont calculées avec la matrice remplie.

Enfin, l'évaluation (note) cible est calculée en fonction de l'ensemble voisin des utilisateurs, et les N items les plus intéressés sont recommandés à l'utilisateur cible. Grâce aux expériences et par rapport aux algorithmes de recommandation existants, l'algorithme adapté améliore la précision de la recommandation et résout efficacement les problèmes de manque de données et de démarrage à froid.

# *Abstract*

Recommendation systems (SR) are used to help people reduce the time they spend discovering the articles that interest them. Collaborative filtering (FC) is one of the most effective techniques in SR. The scarcity of data makes the result of the recommendation inaccurate.

In this work, we use an evolutionary classification method, where the goal is to group together users who have a similar interest in the same cluster and recommend items to users they might like.

Beforehand, the user attribute distance is calculated, based on the constructed network-clustering model, where user states change over time. User states would be stable after a certain iteration period. In light of the stable states of the users, they are grouped into several groups (cluster).

Second, we fill in the rating matrix based on rating time and genre of movies. Third, user-based collaborative filtering is adopted in each cluster. Similarities between users only in the same cluster are calculated with the matrix filled

Finally, the target assessment (score) is calculated as a function of the neighboring set of users, and the N most interested items are recommended to the target user. Based on experiences and compared to existing recommendation algorithms, the adapted algorithm improves the accuracy of the recommendation and effectively resolves the issues of lack of data and cold start.

# TABLE DES MATIERES

Remerciement ..... I
Résumé ..... II
Abstract ..... III
Liste des figures ..... VIII
Liste des tableaux ..... X
Liste des abréviations ..... XI
Introduction Générale ..... 1
**Chapitre I : Les systèmes de recommandation personnalisée ..... 3**
1. Introduction : ..... 4
2. Définition et fonctionnement des systèmes de recommandations : ..... 4
3. Historique : ..... 5
4. Types de données d'entrée de système de recommandation : ..... 6
4.1. Item : ..... 6
4.2. Utilisateur : ..... 6
4.3. Transaction : ..... 6
5. Domaines d'application des systemes de recommandation : ..... 7
6.Les approches de recommendation : ..... 7
6.1. L'approche basée sur le contenu : ..... 8
6.1.1.Définition : ..... 8
6.1.2.Principe général : ..... 9
6.1.2.1. Les modèles vectoriels : ..... 9
6.1.2.2. Le processus de la recommandation basé contenu : ..... 11
6.1.3.Les points forts et les limites des approches basées sur le contenu : ..... 12
6.1.4.Exemple sur le filtrage basé contenu : ..... 13
6.2. Le filtrage collaboratif : ..... 15
6.2.1.Algorithme basé sur la memoire : ..... 16
6.2.1.1. Basé utilisateur : ..... 16
6.2.1.2. Basé item : ..... 17
6.2.2.Algorithme basé sur un modèle : ..... 17
6.2.3.Les points forts et les limites du filtrage collaboratif : ..... 18

6.2.4.Exemple sur le filtrage collaboratif :.....19
6.3. Le filtrage hybride : .....21
6.3.1.Définition et principe : .....21
6.3.2.Les taxonomies de burke (classification) : .....22
6.3.3.Exemple hybride : .....23
6.4. Autres systèmes de recommendation :.....23
6.4.1. Le filtrage démographique :.....23
6.4.2. Le filtrage à base d'utilité : .....24
6.4.3. Le filtrage à base de connaissance : .....24
6.4.4. La recommandation basée tags :.....24
7. Conclusion : .....24

**CHAPITRE II : Les méthodes de clustering.....26**
1. Introduction :.....27
2. Définition : .....27
3. Les exigences de clustering : .....28
4. Les trois étapes principales du clustering : .....29
5. Approche collaborative pour le clustering : .....29
6. Les méthodes de clustering :.....30
6.1. Hierarchique :.....31
6.1.1.Principe : .....32
6.1.2.L'approche par agglomeration (ascendante) : .....33
6.1.2.1. Le principe général de la cah : .....33
6.1.2.2. Algorithme : .....34
6.1.3. L'approche par division (descendante) :.....34
6.1.3.1. Algorithme : .....35
6.2. Non hierarchique :.....36
6.2.1.Le clustering par partitionnement : .....36
6.2.1.1. Méthodes basées sur les k-centroïdes :.....37
6.2.1.1.1. Méthode des centres mobiles : .....37
6.2.1.1.1.1. Méthode des k-moyennes : .....38

6.2.1.2. k-medoids :.....40
6.3. Le clustering par densité :.....40
6.4. Autres méthodes :.....42
6.4.1.Grille :.....42
6.4.2. Probabiliste :.....43
6.4.3. Le clustering basé sur les graphes :.....44
6.4.4. Le clustering stochastique :.....44
7. Comparaison entre les méthodes de clustering :.....45
8. Mesure de similarité :.....46
8.1. Calcul de la similarité :.....46
8.2. Fonctions de distance :.....46
9. Les limites et les points forts :.....47
10. Conclusion :.....48

**CHAPITRE III : Proposition et évaluation.....49**
1.Introduction :.....50
2.Présentation des systèmes de recommandation à base de clustering :.....50
3.Notre système de recommandation basé sur clustering :.....51
3.1.Le modèle dynamique :.....51
3.2.Mesures de similarité :.....52
4.Solution proposée :.....54
4.1.Générer des clusters utilisateurs basé sur la distance d'attribut utilisateur :.....54
4.2.Téchnique de remplissage basée sur le temps de notation et le genre de film :.....57
4.3.Recommandation :.....59
4.4.L'algorithme :.....60
5.Dataset :.....62
6.Métrique d'evaluation :.....63
7.Outils utilisés :.....64
7.1.Python :.....64
7.2.Caractéristiques du langage :.....65
8.Présentation de l'application :.....65

9.Résultats et analyses :.....67
9.1.Comparaison des résultats de la matrice remplie :.....67
9.2.Comparaison des résultats d'un indice de similarité amelioré : .....69
9.3.Comparaison des résultats du regroupement dynamique des utilisateurs evolutifs : .....71
10.Conclusion : .....73
Conclusion générale .....74
Bibliographie et Référence .....76

# *Liste des figures*

Figure 1.1: Un système de recommandation basé sur le contenu. ....9
Figure 1.2: Architecture d'un système de recommandation basé sur le contenu utilisant l'apprentissage profond Suglia et al. [2017]. ....11
Figure 1.3: Un système de recommandation collaboratif. ....16
Figure 1.4: Vos recommandations » sur le site Amazon.com. ....21
Figure 1.5: Panier de recommandations sur le site Amazon.com....21
Figure 1.6: Le système de recommandation hybride....23
Figure 2.1: Illustration de regroupement en clusters ....28
Figure 2.2: Les deux types de clustering non-hiérarchique/ hiérarchique ....31
Figure 2.3: Principe général du clustering hiérarchique....32
Figure 3.1: Graphe statistique de la situation d'âge de l'utilisateur sur l'ensemble de données MovieLens....56
Figure 3.2: Distribution des notes dans la base de données MovieLens....62
Figure 3.3: L'ajout du data ....66
Figure 3.4: Code de préparation de la colonne âge ....66
Figure 3.5: Code de préparation de la colonne sex....66
Figure 3.6: Code de préparation de la colonne occupation....67
Figure 3.7: Résultat final de la préparation du data....67
Figure 3.8: Comparaison des performances entre la matrice remplie et la matrice de notation d'origine par la mesure RMSE ....68
Figure 3.9: Comparaison des performances entre la matrice remplie et la matrice de notation d'origine par la mesure MAE ....69
Figure 3.10: Comparaison entre les performances de différentes similitudes par la mesure MAE ....70
Figure 3.11: Comparaison entre les performances de différentes similitudes par la mesure RMSE ....70
Figure 3.12: Comparaison entre l'algorithme réalisé et des algorithmes existants par la mesure MAE ....71

Figure 3.13: Comparaison entre l'algorithme réalisé et des algorithmes existants par la mesure RMSE .....72
Figure 3.14: Comparaison entre l'algorithme réalisé et l'algorithme k-means et PR-ACMF par la mesure MAE .....72
Figure 3.15: Comparaison entre l'algorithme réalisé et l'algorithme k-means et PR-ACMF par la mesure RMSE .....73

# *Liste des tableaux*

Tableau 1.1 : Exemple d'une matrice de notes.....5
Tableau 1.2 : Extrait du catalogue de livres.....13
Tableau 1.3 : Extrait du profil de l'utilisatrice Ouldrabah.....14
Tableau 1.4 : Coïncidence entre les caractéristiques des livres et les préférences de Ouldrabah indiquées dans son profil. .....14
Tableau 2.1: Etude comparative entre les méthodes de clustering.....45
Tableau 3.1: Attributs des utilisateurs.....55

# *Liste des abréviations*

**BEOS:** Blue Eyed Operating System

**CHA:** Classification Hiérarchique Ascendante

**DBCLASD:** Distribution-Based Clustering Algorithm for Clustering LArge Spatial Datasets

**DBSCAN:** Density-Based Spatial Clustering of Applications with Noise

**DIANA :** DIvisive ANAlysis

**EM :** Espérance-Maximisation

**FC:** Filtrage Collaboratif

**GDBSCAN:** Generalized Density-Based Spatial Clustering of Applications with Noise

**IDF:** Inverse Document Frequency

**LSTM**: Long Short-Term Memory

**MacOs:** Macintosh Operating System

**MAE:** Mean Absolute Error

**MS-DOS:** Microsoft Disk Operating System

**OS**: Operating System

**RMSE:** Root Mean Squared Error

**SR :** Système de Recommandation

**TF-IDF :** Term Frequency- Inverse Document Frequency

# Introduction Générale

# *Introduction Générale*

Parmi les domaines de recherche principaux relatifs à la problématique de surcharge et de richesse d'informations disponibles nous citons le domaine de la recherche d'information, dont le principe général est d'élaborer des méthodes et des algorithmes afin de rechercher des ressources (par exemple, des pages web, des films, des produits) en fonction de requêtes formulées par des utilisateurs. Il n'est cependant pas toujours évident pour un utilisateur de savoir comment exprimer sa demande. De plus, sa requête correspond généralement à une quantité importante de ressources et il est difficile de savoir quels résultats lui présenter en premier, d'autant plus que d'un utilisateur à l'autre, l'ordre de priorité peut changer.

Un autre domaine de recherche relatif à cette problématique est le domaine des systèmes de recommandation (SR). Ces systèmes sont capables de fournir des recommandations adaptées aux préférences et aux besoins des utilisateurs. Ils se sont avérés être très satisfaisants pour aider les utilisateurs à accéder aux ressources désirées dans un temps limité, les SR sont devenus de plus en plus populaires et sont aujourd'hui un composant principal pour beaucoup d'applications dans différents domaines. Un avantage très conséquent des systèmes de recommandation est que l'utilisateur n'a pas besoin de formuler de requêtes. Sa seule requête est implicite et elle peut se traduire par : "Quelles sont les ressources qui correspondent à mes préférences, mes besoins et mes contraintes ? ".

Le filtrage collaboratif (FC) est l'une des techniques SR les plus fréquemment utilisées [1,2]. L'algorithme doit calculer la similarité entre l'utilisateur cible (ou item) et tous les utilisateurs (ou items). Avec l'augmentation des utilisateurs et des items, la rareté des données et la faible expansibilité sont les facteurs clés qui affectent la qualité du système de recommandation. Afin de résoudre ces problèmes et de trouver avec précision les utilisateurs voisins, une technologie de clustering est utilisée pour optimiser l'algorithme de filtrage collaboratif. Les approches basées sur le clustering offrent une alternative aux méthodes basées sur des modèles. Au lieu de décomposer la matrice de notation en matrices de petits rangs, ces approches ont réduit l'espace de recherche en regroupant des utilisateurs ou des items similaires [3]. Il existe des algorithmes de clustering non hiérarchiques bien connus, à savoir le k-means et la méthode floue c-means. Le clustering comprend principalement les étapes suivantes : pré-traitement des données ; la fonction de distance pour mesurer la similarité entre les objets ; obtenir le cluster.

La technologie de clustering peut améliorer l'évolutivité du filtrage collaboratif [4]. La technologie de remplissage matriciel a été largement utilisée dans de nombreuses tâches d'exploration de données. Dans le système de recommandation, le nombre d'utilisateurs et d'items est très important. Cependant, chaque utilisateur évalue seulement quelques items, et chaque item n'est noté que par très peu d'utilisateurs. En conséquence, les données de notation sont extrêmement rares, ce qui entraîne une erreur lors du calcul des similarités entre les utilisateurs et les items, ce qui affecte gravement la précision de la prédiction [5]. Le fait de remplir les notes inconnues est le moyen le plus direct de densifier les données éparses et d'améliorer ainsi les performances du clustering. Il est crucial de sélectionner un algorithme de remplissage approprié [6]. De nombreux chercheurs ont profité de l'utilisation de l'approche de remplissage des données. Cette approche reste un moyen simple et efficace pour améliorer l'effet de calcul de la similarité. Dans ce travail, nous adoptons un modèle de clustering évolutif dynamique basé sur la distance des attributs utilisateur par rapport aux utilisateurs de cluster. Nous remplissons la matrice de note (utilisateur-item) avec les notes prédites par le temps de notation et les items à contenu multiple. De plus, nous utilisons la matrice remplie pour calculer la similarité des utilisateurs qui sont dans le même cluster. Ensuite, les k plus proches voisins sont générés pour prédire les notes. Enfin, les items $N\text{-}top$ sont recommandés à l'utilisateur cible.

Le manuscrit est organisé de la façon suivante :

**Chapitre I : les systèmes de recommandation personnalisée**

Nous présentons au sein de ce chapitre un état de l'art sur les systèmes de recommandation, tels que leurs principes, fonctionnements et caractéristiques, ainsi que nous énonçons les différents types de filtrage.

**Chapitre II : Les méthodes de Clustering**

Le deuxième chapitre concerne les principales méthodes de clustering. Il présente les origines, le principe de fonctionnement, l'architecture générale de ces méthodes, ainsi que leurs avantages et inconvénients. Il présente aussi les systèmes de recommandation à base de clustering.

**Chapitre III** : **Proposition et évaluation**

Dans ce chapitre, nous mettons notre proposition qui essaie de répondre à la problématique posée et d'améliorer le traitement et le résultat de système de recommandation ce qui permet à moyen terme d'accroître la satisfaction de l'utilisateur, et de garantir sa fidélisation

# Chapitre I

# Les Systèmes de Recommandation Personnalisée

## 1. Introduction :

Nous sommes confrontés souvent à faire des choix, tel que se vêtir ? Quel film regarder ? Quel article acheter ? Que visiter en période de voyage ?

La taille de ces domaines de décision est très souvent grande. Par exemple, Netflix disposait en 2019 de plus de 50,000 films dans sa base de données et 8,8 millions de nouveaux abonnés [7], et ce nombre ne cesse de croitre au fil des années. La liste des possibilités qui s'offrent à nous est donc en général de très grande taille. L'évaluation de ces possibilités pour trouver ce qui nous convient le plus est une tâche difficile et peut consommer beaucoup de notre temps.

Delà, Un système de recommandation personnalisé a pour objectif d'apprendre les goûts et les préférences d'un utilisateur donné afin de lui recommander par la suite les items susceptibles de l'intéresser. En fait, c'est un système de traitement de l'information qui exploite les différents types de données pour établir des recommandations à partir de sélection dans leur catalogue.

Ce chapitre décrit de façon détaillée le contexte de notre travail. Nous présentons la théorie des systèmes de recommandation, ses origines, les concepts de base et notions liées, sa formalisation, les différentes approches et problèmes liés aux méthodes classiques et nous terminons par la présentation des approches hybrides.

## 2. Définition et fonctionnement des systèmes de recommandations :

Le Système est capable de fournir des recommandations personnalisées ou permettant de guider l'utilisateur vers des ressources intéressantes ou utiles au sein d'un espace de données important.

Mais il existe une définition générale [8] qui les définit comme suit :

"Des systèmes capable de fournir des recommandations personnalisées permettant de guider l'utilisateur vers des ressources intéressantes et utiles au sein d'un espace de données important".

L'ensemble des triplets (utilisateur, item, note) forme ce que l'on appelle la matrice des notes. Les paires (utilisateur, item) où l'utilisateur n'a pas donné de note pour l'item sont des valeurs non connues dans la matrice. Le tableau 1.1 illustre un exemple d'une matrice de notes pour 4 utilisateurs et 4 films. Les valeurs marquées " ? " indiquent que l'utilisateur n'a pas donné d'avis [9]. Un système de recommandation se focalise sur deux tâches. La première tâche est la prédiction : étant donné un utilisateur et un item, quelle serait la préférence de l'utilisateur pour

cet item ? En d'autres termes, le système doit prédire la valeur des notes marquées " ? ". La deuxième tâche est la recommandation : étant donné

| | La Bataille d'Alger | l'inspecteur Tahar | Titanic | Carnaval fi dachra |
|---|---|---|---|---|
| Yazid | 4 | 3 | 2 | 4 |
| Amine | ? | 4 | 5 | 5 |
| Abderrahmane | 2 | 2 | 4 | ? |
| hamza | 3 | ? | 5 | 2 |

*Tableau 1.1 : Exemple d'une matrice de notes*

Un utilisateur, quelle liste ordonnée de n recommandations peut-on lui suggérer ? On parle alors de liste Top-n. A noter que la liste des Top-n recommandations n'est pas forcément la liste des n items avec les plus hautes valeurs de prédiction. La prédiction des notes n'est pas le seul critère utilisé pour produire une liste de recommandations. En effet, un algorithme de recommandation peut utiliser d'autres critères, tels que le contexte.

## 3. Historique :

Cela fait presque 25 ans que la recherche s'intéresse à la façon de recommander automatiquement des items aux utilisateurs. Une grande variété de méthodes, d'algorithmes et d'outils ont été proposés (Adomavicius et Tuzhilin, 2005 ; Su et Khoshgoftaar, 2009) [10]. Le domaine de recherche sur les systèmes de recommandation est apparu au début des années 90 et depuis, notamment avec l'intégration des réseaux sociaux, de l'apprentissage artificiel et du big data, ce domaine est en constante évolution. Bridge et al. (2006) et Yang et al. (2014) [11] expliquent que les systèmes de recommandation se basent sur plusieurs domaines de recherche tels que la recherche d'information, la modélisation de l'utilisateur, l'apprentissage artificiel, les sciences cognitives et les interactions humain-machine. Parmi les premiers systèmes de recommandation qui sont apparu dans les années 90, nous retrouvons Tapestry (Goldberg et al., 1992)[12] pour la recommandation des messages des newsgroups, GroupLens (Resnick et al.,

1994)[13] pour la recommandation des articles Usenet et Ringo (Shardanand et Maes, 1995) pour la recommandation de la music[14].

## 4. Types de données d'entrées de système de recommandation :

Les systèmes de recommandation utilisent trois types de données d'entrée pour générer des recommandations dans tout type de système de recommandation [15]. La sortie d'un système de recommandation peut prendre la forme d'une liste de recommandations ou d'un score de prédiction [16]. Les types de données d'entrée des systèmes de recommandation sont décrits dans les sous-sections de cette section.

### 4.1. Item :

Les éléments sont des objets qui sont recommandés aux utilisateurs par les systèmes de recommandation. Les éléments utilisés dans les systèmes de recommandation peuvent être divisés en deux catégories [15].

- Item à faible complexité et coût
- Item à complexité et coût élevés

### 4.2. Utilisateur :

La personne qui obtient des recommandations du système est appelée en tant qu'utilisateur du système. Étant donné que les systèmes de recommandation sont des systèmes personnalisés, les informations utilisateur sont l'une des informations les plus importantes que les recommandeurs utilisent. Les informations utilisateur peuvent inclure les informations de préférence de l'utilisateur relatives aux articles et elles peuvent également inclure les informations de relation entre les utilisateurs telles que ; niveaux de confiance des utilisateurs [15].

### 4.3. Transaction :

Les transactions peuvent être définies comme les interactions entre les utilisateurs et le système de recommandation tels que ; les informations de notation relatives à un élément fournies par l'utilisateur [15]. Il existe quatre types de transactions utilisées dans les systèmes de recommandation [17] :

- Évaluations numériques (telles que : 1-5 étoiles et échelle 1-10)

- Notes ordinales (telles que : "tout à fait d'accord, d'accord, neutre, en désaccord, fortement en désaccord")
- Notes binaires (telles que «bon, mauvais» et «comme, je n'aime pas»)
- Évaluations unaires (définit si l'utilisateur a observé / acheté l'article ou non)

## 5. Domaines d'application des systèmes de recommandation :

Les domaines d'application utilisés dans les systèmes de recommandation sont principalement regroupés en quatre catégories principales [15, 18, 19] :

- Divertissement (comme les films, les programmes télévisés et la musique)
- Contenu (tels que : journaux, documents et pages Web)
- Commerce électronique (tels que : livres, appareils photo et PC)
- Service (tel que : voyage, consultation et location de maison)

## 6. Les approches de recommendation :

Un nombre important de travaux de recherche ont traité la problématique de la recommandation au cours des dernières années. Ces travaux sont issus de plusieurs domaines comme le Machine Learning, les statistiques et surtout la recherche d'information. Les techniques de recommandation peuvent être classées de différentes manières. Parfois plusieurs termes sont utilisés pour désigner une même méthode ou approche. L'objectif ici est de s'appuyer sur les classifications les plus connues sur lesquelles nous basons notre étude.

Plusieurs facteurs entrent en considération afin de catégoriser les systèmes de recommandation.

1. La connaissance de l'utilisateur (c.-à-d. son profil en fonction de ses gouts).
2. Le positionnement d'un utilisateur par rapport aux autres (la notion de classes ou réseaux d'utilisateurs).
3. La connaissance des items à recommander.
4. La connaissance des différentes classes d'items à recommander.

De ces facteurs sont produits divers type de recommandations dont les plus utilisés sont le filtrage basé sur le contenu et le filtrage collaboratif [20]. Ce document présente dans un premier temps ces deux approches ainsi que leurs hybridations, ainsi que d'autres systèmes

### 6.1. L'approche basée sur le contenu :

#### 6.1.1. Définition :

Les systèmes de recommandation basés sur le contenu s'appuient sur des évaluations effectuées par un utilisateur sur un ensemble de documents ou items. L'objectif est alors de comprendre les motivations l'ayant conduit à juger comme pertinent ou non un item donné. Le système peut alors proposer à l'utilisateur un choix parmi de nouveaux items jugés proches des items qu'il a précédemment apprécies. Ces méthodes font donc appels aux facteurs 1, 3. Et 4. Présentés en section 6. Notons que le filtrage par le contenu peut s'appuyer sur les appréciations à la fois positives mais également négatives d'un utilisateur. Salton and Buckley ont montré en effet dans le cadre du filtrage de descripteurs que l'absence d'un mot est à prendre en compte au même titre que sa présence [21]. L'objectif de ces derniers étant de proposer une mesure de pondération de mots afin de faire émerger les plus représentatifs d'un domaine. Leur protocole expérimental est le suivant. Ils disposent d'un certain nombre de requêtes d'utilisateurs et d'un certain nombre de corpus de domaines différents. Alors, un classement de la proximité des requêtes utilisateurs avec les documents des différents corpus est obtenu en utilisant leurs différentes mesures proposées. Les résultats ont montré que l'utilisation de la mesure bien connue de l'IDF (inverse document fréquence), pondérant les termes en fonction de leur absence dans un certain nombre de documents, améliorait significativement la qualité du classement obtenu.

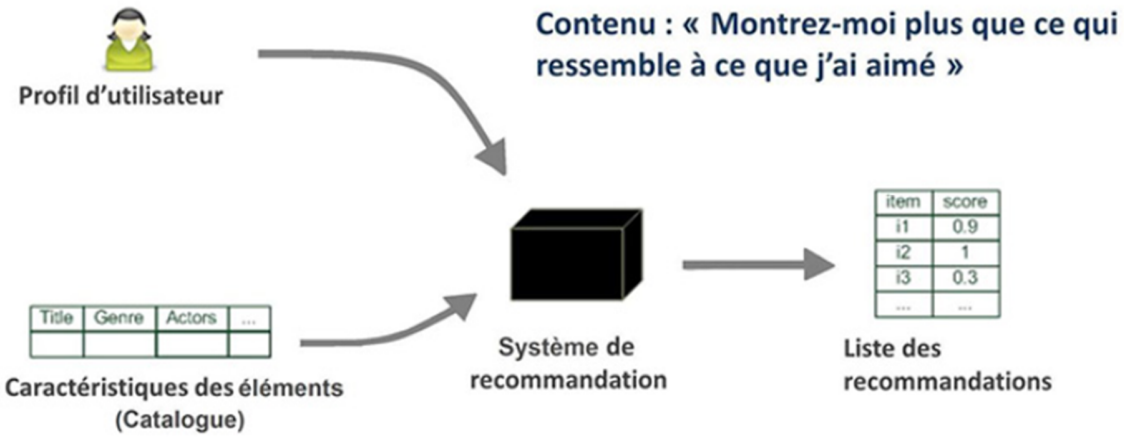

*Figure 1.1 : Un système de recommandation basé sur le contenu*

### 6.1.2. Principe général :

Le filtrage basé sur le contenu est la méthode la plus populaire pour recommander des articles caractérisés par du contenu textuel [22], par exemple les revues et les articles scientifiques. Le domaine de recherche des systèmes basés sur les contenus est très proche de la Recherche d'Information [23]. En effet ces systèmes s'appuient sur un modèle d'espace vectoriel, les profils des utilisateurs et des articles étant représentés par des vecteurs de termes pondérés de type TF-IDF [24]. Ces systèmes ont pour limite qu'ils n'arrivent pas à percevoir le sens des textes, ce qui crée une perte de l'information sémantique liée au profil de l'utilisateur et de l'article. Les technologies liées au web sémantique (ontologie, thésaurus, base de connaissance) sont des solutions qui ont été proposées pour répondre à cette problématique. Dans la suite de cette section nous allons présenter, les modèles vectoriels et le processus de la recommandation basé contenu

#### 6.1.2.1. Les modèles vectoriels :

Les modèles vectoriels et une de leurs extensions, l'analyse sémantique latente, sont des méthodes qui utilisent des termes pour représenter les documents sous forme vectorielle dans des espaces à plusieurs dimensions. Ils ont été utilisés dans les systèmes de recommandation Web, [25], [26], [27], [28]. Ces systèmes s'appuient sur des retours implicites des utilisateurs pour comprendre leurs préférences, pages visitées, pages mis en favori ou sur des retours explicites des utilisateurs avec des systèmes « j'aime », « je n'aime pas ». Les documents et les profils des utilisateurs sont représentés par *N* mots-clés qui permettront par la suite de produire des recommandations en comparant ces deux familles de vecteurs. Ces techniques ont aussi été employées dans la recommandation de textes liés à l'actualité [29]. Au-delà du texte, certains systèmes de recommandation sur le contenu utilisent des « tags » pour qualifier les articles et les utilisateurs [30]. Les méthodes de pondération BM25 [31] ont servi à pondérer les vecteurs obtenus à partir des tags. Depuis l'augmentation exponentielle du nombre de sources textuelles sur le Web, plus particulièrement avec l'arrivée des bases encyclopédiques (Wikipedia, Freebase), les nouvelles techniques de représentation vectorielles liées au plongement lexical (Word2vec) [32], ont aussi été utilisées pour représenter les articles et les utilisateurs [33]. Ces

techniques ont montré des résultats encourageants et ont laissé présager de futures exploitations pour les systèmes de recommandation basés sur le contenu. L'usage de l'apprentissage profond est d'autant plus intéressant qu'il est capable d'extraire directement des caractéristiques des articles [34]. L'apprentissage profond permet au système de recommandation de travailler directement avec la donnée. Ceci est très utile quand la donnée peut être caractérisée par du son, des images, des vidéos, en plus du texte. Les réseaux neuronaux convolutifs ont été utilisés pour extraire des caractéristiques visuelles d'images pour apprendre une distance entre des articles [35]. La distance était personnalisée en fonction d'un utilisateur, et le but était de maximiser la probabilité qu'il existe une relation entre un utilisateur et un article. Un réseau de neurones convolutif a aussi été appliqué aux recommandations musicales [36]. Le but ici était de résoudre le problème du démarrage à froid, c'est-à-dire produire des recommandations quand il y a peu de données d'écoute, réussir à recommander les musiques peu populaires ou nouvelles. Mais il est connu qu'il existe un écart sémantique entre le signal audio et les préférences d'un utilisateur. Ce problème peut être résolu en utilisant des données textuelles comme la biographie d'un artiste pour entrainer le réseau de neurones [37]. Les informations textuelles ont même la faculté d'améliorer les recommandations [38]. Une architecture a été proposée pour prédire la probabilité qu'un utilisateur apprécie un article [39]. Ce travail utilise une forme de réseau de neurones récurrents, Long short-term memory (LSTM). Il est composé de deux modules qui génèrent deux types de plongement, un pour les utilisateurs et un pour les articles. Le plongement pour les articles est appris à partir du texte de description fourni avec l'article. Cette architecture a été étendue en ajoutant différents modules qui prennent en compte d'autres types d'informations fournies avec les articles (p.ex. les genres) [40], elle est schématisée à la figure 1.2. Le modèle présenté donne de meilleurs résultats que les systèmes de recommandation basés sur le filtrage collaboratif et les méthodes de plongement de mots sur les jeux de données Movielens et DBbook.

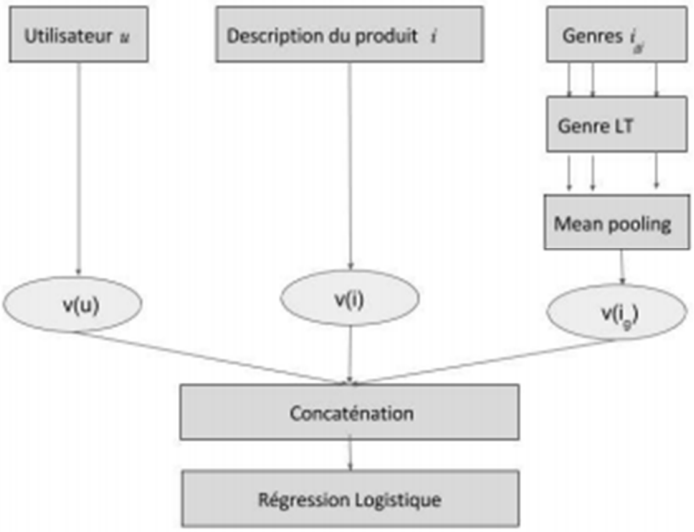


*Figure 1.2 : Architecture d'un système de recommandation basé sur le contenu utilisant l'apprentissage profond Suglia et al. [2017].*

#### 6.1.2.2. le processus de la recommandation basé contenu :

Un système de recommandation basé sur le contenu a besoin de techniques pour produire une représentation efficace des items et du profil de l'utilisateur pour pouvoir les comparer. Ainsi [41] proposent une architecture de haut niveau dans laquelle le processus de recommandation est réalisé en trois étapes, chacune étant gérée par un composant spécifique :

- **Analyseur du contenu** : Lorsque l'information n'est pas structurée (par exemple, un item représenté par un texte), ce module a pour but d'en réaliser le prétraitement pour extraire l'information pertinente, la structurer et la représenter dans une forme cible appropriée (par exemple un vecteur de mots clés).
- **Apprentissage du profil** : Ce module collecte les données représentatives des préférences de l'utilisateur et généralise ces données, afin d'apprendre et de construire le profil de l'utilisateur. Des techniques d'apprentissage automatique [42] peuvent être utilisées pour cela. On peut citer à titre d'exemple les arbres de décisions, les réseaux de

neurones et la classification naïve de Bayes. Ces techniques visent à inférer un profil de l'utilisateur en utilisant l'information sur les items qu'il a aimés ou n'a pas aimés.

- **Composant de filtrage :** Ce module filtre les items pertinents en faisant correspondre la représentation du profil utilisateur aux items candidats à la recommandation. La pertinence de l'item est calculée en utilisant des métriques de similarité entre l'item considéré et le profil de l'utilisateur. Plus la similarité avec le profil "positif" est grande et plus la similarité avec le profil "négatif" est petite, plus l'item a des chances d'être recommandé.

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

### 6.1.3. Les points forts et les limites des approches basées sur le contenu :

Les approches basées sur le contenu présentent plusieurs avantages et inconvénients. Les points forts des approches basées sur le contenu sont :

- **Utilisateur indépendant** : pour construire un système de personnalisation, l'approche basée sur le contenu ne nécessite pas d'index de similarité entre les utilisateurs [43]. La liste de recommandations peut être générée en analysant les attributs des articles et le profil des utilisateurs.
- **Comportement transparent** : ce système vous fait prendre conscience des attributs des articles sur la base desquels une recommandation a été faite.

Cependant les approches de recommandation basées sur le contenu présentent aussi de nombreux inconvénients :

- **Intégration d'un nouvel utilisateur non immédiate** : un utilisateur doit évaluer un certain nombre d'items avant que le système ne puisse interpréter ses préférences et lui fournir des recommandations pertinentes [44]. Ce problème est connu dans la littérature sous le nom du problème de démarrage à froid pour les utilisateurs (user cold Start).

- **Diversité et nouveauté insuffisantes :** ce système peut poser le problème de la surspécialisation.
- **Analyse de contenu bornée** : si les éléments ne disposent pas d'assez d'informations sur les attributs [45,46], donner une liste de recommandations plus précise sera une tâche difficile pour le système de recommandation basé sur le contenu.

### 6.1.4. Exemple sur le filtrage basé contenu :

Considérons un catalogue de film, tel que l'extrait présenté dans le tableau 1.2, ainsi que l'extrait du profil de l'utilisateur « *Ouldrabah* », basé sur ses précédents achats de films et sur les informations qu'elle a données concernant ses préférences, tel qu'illustré par le tableau 1.3. Nous souhaitons proposer à « *Ouldrabah* » des films susceptibles de lui plaire. Il s'agit de faire coïncider le tableau 1.2 et le tableau 1.3 et de définir quels films correspondent le mieux aux préférences de « *Ouldrabah* ». Le tableau 1.4 résume la coïncidence entre les deux autres tableaux (oui : les informations coïncident ; non : les informations ne coïncident pas). Ainsi à partir du tableau 1.4, intuitivement, les films «La Bataille d'Alger » et «Les Vacances de l'inspecteur Tahar» pourraient être recommandés à « *Ouldrabah* » (puisque les caractéristiques de ces deux films coïncident le mieux avec les préférences de «*Ouldrabah* »).

| Titre | Genre | Réalisateur | Prix | Mots-clés |
|---|---|---|---|---|
| La Bataille d'Alger | Guerre | Gillo Pontecorvo | 19,50 | Historique, Guerre, Algérie, France, Bataille, … |
| Les Vacances de l'inspecteur Tahar | Policier | Moussa Haddad | 23,20 | Inspecteur, Police, Détective, Tunisie, Vacance, … |
| Carnaval fi dachra | Comedie | Mohamed Oukassi | 8,50 | Athmane Aliwat, Elections, Commune, Le maire, … |

*Tableau 1.2 : Extrait du catalogue de livres.*

| | | |
|---|---|---|
| Identifiant | | 1 |
| Informations personnelles | Prénom | Ouldrabah |
| | Sexe | Homme |
| Films | Genres | Guerre, Policier |
| | Réalisateur | Gillo Pontecorvo, Ahmed Rachdi |
| | Titres | Lotfi, Hassan Taxi, Deux Femmes, … |
| | Mots-clés | Détective, Comedie, Guerre, … |
| | Prix moyen | 20 |
| Livres | Genres | Thriller, Policier |
| | Auteurs | S. King, M. Connelly |

*Tableau 1.3 : Extrait du profil de l'utilisatrice Ouldrabah*

| Préférences utilisateur | La Bataille d'Alger | Les Vacances de l'inspecteur Tahar | Carnaval fi dachra |
|---|---|---|---|
| Genre | Oui | Oui | Non |
| Prix | Oui | Oui | Oui |
| Réalisateur | Oui | Non | Non |
| Mots-clés | Oui | Oui | Non |

*Tableau 1.4 : Coïncidence entre les caractéristiques des livres et les préférences de Ouldrabah indiquées dans son profil.*

### 6.2. Le filtrage collaboratif :

La première méthode de recommandation basée sur les usages que nous présentons, Le filtrage collaboratif (FC) a pour principe d'exploiter les évaluations faites par des utilisateurs sur certains contenus (films), afin de recommander ces mêmes films à d'autres utilisateurs, et sans qu'il soit nécessaire d'analyser le contenu des films.

Tous les utilisateurs du système de filtrage collaboratif peuvent tirer profit des évaluations des autres en recevant des recommandations pour lesquelles les utilisateurs les plus proches ont émis un jugement de valeur favorable, et cela sans que le système dispose d'un processus d'extraction du contenu des documents. Grace à son indépendance vis-à-vis de la représentation des données, cette technique peut s'appliquer dans les contextes où le contenu est soit indisponible, soit difficile à analyser, et en particulier elle peut s'utiliser pour tout type de données : texte, image, audio et vidéo. De plus, l'utilisateur est capable de découvrir divers domaines intéressants, car le principe du filtrage collaboratif ne se fonde absolument pas sur la dimension thématique des profils, et n'est pas soumis à l'effet « entonnoir ». Un autre avantage du filtrage collaboratif est que les jugements de valeur des utilisateurs intègrent non seulement la dimension thématique mais aussi d'autres facteurs relatifs à la qualité des films tels que la diversité, la nouveauté, etc.

Le FC souffre de plusieurs gros problèmes. Le problème principal étant le démarrage à froid : c'est le fait qu'un utilisateur doit voter sur beaucoup d'objet avant d'obtenir les recommandations.

Les algorithmes utilisés dans les recommandeurs de filtrage collaboratif sont regroupés sous deux catégories principales

- Algorithmes basés sur l'heuristique (basés sur la mémoire)
- Algorithmes basés sur un modèle

La principale différence entre les approches basées sur l'heuristique et les approches basées sur le modèle est :

Les approches basées sur des modèles utilisent des modèles qui sont appris à l'aide d'algorithmes statistiques et d'apprentissage automatique, mais les approches basées sur l'heuristique utilisent des règles heuristiques pour calculer les scores de satisfaction des éléments selon les utilisateurs

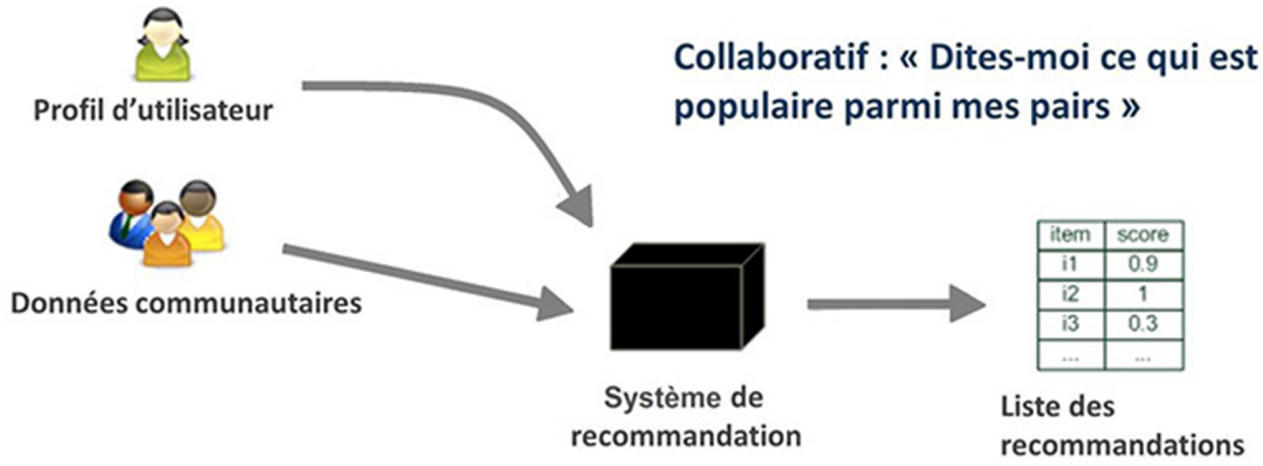


*Figure 1.3 Un système de recommandation collaboratif*

### 6.2.1. Algorithme basé sur la mémoire :

Elles sont aussi appelées méthodes basées sur des heuristiques [47], ou basées sur les voisins [48]. Ces méthodes utilisent la matrice de notations des utilisateurs stockée en mémoire pour proposer des recommandations. Le type de voisinage devise ces méthodes en deux catégories : basées sur les utilisateurs (*user-based*) ou basées sur les items (*item-based*) [49]. L'obtention des voisins, d'un utilisateur ou d'un item, se fait en utilisant des mesures de similarité pour calculer la distance entre deux utilisateurs ou deux items. La similarité peut être calculée selon différentes formules, les plus utilisées sont celles du coefficient de Pearson [50,51] et la méthode du vecteur cosinus [52,53].

#### 6.2.1.1. Basé utilisateur :

Pour les méthodes basées sur les utilisateurs, l'intérêt d'un utilisateur $u$ pour un item $i$ est calculé en se basant sur les évaluations de ses voisins similaires.

L'idée principale est d'identifier les utilisateurs qui ont des valeurs de notation similaires et leur suggérer une liste des meilleurs articles en fonction de leurs préférences (Hahsler, 2014). Il existe une grande variété de paramètres pour comparer la similarité entre les vecteurs ou trouver le voisin le plus proche (dans notre cas, les utilisateurs). Les plus populaires sont la similarité des cosinus ou Pearson Corrélation [54,55].

Son principe de fonctionnement se résume ainsi :

- Déterminer les voisins de l'utilisateur courant en calculant sa similarité avec les autres Utilisateurs de la base.
- Calculer la prédiction du vote de l'utilisateur courant $u_a$ pour un item $i \in /I$ $u_a$ candidat à la recommandation, en analysant les votes de ses voisins sur ce même item.

#### 6.2.1.2. Basé item :

Dans la section ci-dessus, l'algorithme était basé sur les utilisateurs et les étapes pour identifier les recommandations ont d'abord été d'identifier les utilisateurs similaires en termes d'achat des mêmes éléments, puis de recommander une nouvelle utilisation des éléments que les autres utilisateurs ont acquis. Maintenant, l'approche est le contraire. Nous commençons à chercher des utilisateurs similaires en fonction des achats et des préférences, en d'autres termes, nous essayons de découvrir à quel point un film est similaire à un autre film. Les principales idées calculent la similarité entre les deux item $P_J$ et $P_I$, en séparant les utilisateurs qui ont déjà regardé et évalué les deux films, puis en utilisant l'une des techniques pour calculer la similarité entre les éléments, par exemple, la similarité basée sur le cosinus, la similarité basée sur la corrélation ou la similarité du cosinus ajusté [56].

Son principe de fonctionnement se résume ainsi :

- Pour tout item $i$ de $I$, déterminer les voisins les plus proches en calculant sa similarité avec les autres items.
- Calculer la prédiction du vote de l'utilisateur courant $u_a$ pour l'item i en analysant les votes de ce dernier sur les voisins de $i$.

### 6.2.2. Algorithme basé sur un modèle :

Les méthodes basées sur un modèle ont été intégrées aux systèmes de recommandation pour remédier aux problèmes des méthodes basées sur la mémoire, dont notamment : la non robustesse au manque de données ainsi que le non passage à l'échelle [57] [58]. Pour faire face à ces deux problèmes, les méthodes basées sur un modèle utilisent notamment les techniques de réduction de dimensionnalité ou le clustering dans le but d'écarter les utilisateurs ou les items non représentatifs.

Ainsi l'espace de représentation utilisateur-item est plus réduit et le taux de données manquantes est moins important comparé à l'espace de représentation original. Les voisins peuvent ainsi être calculés dans cet espace réduit, ce qui permet de garantir le passage à Dans le cadre des méthodes basées sur un modèle, le processus de (FC) consiste à construire des modèles (généralement en hors ligne "offline") en exploitant les données collectées sur l'utilisateur et/ou sur l'item. Les modèles construits sont par la suite utilisés pour générer les prédictions qui sont proposées à l'utilisateur actif lors de son interaction avec le système.
Le processus de construction du modèle est basé sur les techniques d'apprentissage automatique, telles que : le clustering, les réseaux bayésiens, les arbres de décision, etc.

### 6.2.3. Les points forts et les limites du filtrage collaboratif :

Les méthodes de filtrage collaboratif présentent plusieurs avantages dont les plus importants sont :

- **Effet de surprise (serendipity) :** l'effet de surprise que peut recevoir l'utilisateur en recevant une recommandation pertinente qu'il n'aurait pas trouvée seul est souvent souhaitable. Les algorithmes basés sur le filtrage collaboratif permettent généralement de faire des recommandations à effet de surprise. Par exemple, si un utilisateur u est proche d'un utilisateur $v$ du fait qu'il ne regarde que des comédies, et si $v$ apprécie un film d'un autre genre, ce film peut être recommandé à u du fait de sa proximité avec $v$
- **Non nécessitée de la connaissance du domaine** : les systèmes de recommandation basés sur le filtrage collaboratif ne requièrent aucune connaissance sur les items. Ces méthodes peuvent recommander des items sans avoir besoin de comprendre leurs sens ni disposer de leurs attributs. La recommandation est basée uniquement sur les notes données aux items.

Cependant, l'utilisation des techniques de filtrage collaboratif peut entrainer plusieurs problèmes :

- **Le démarrage à froid :** concerne à la fois les nouveaux utilisateurs et les nouveaux items qui sont introduits dans le système. Un nouvel utilisateur qui n'a noté aucun item ne peut pas recevoir de recommandation puisque le système ne connaît pas ses goûts. Ce

problème est connu sous le nom de problème du démarrage à froid pour les utilisateurs (user cold start). Une solution à ce problème est de lui demander explicitement de noter un certain nombre d'items. D'autres solutions consistent à recommander au départ les items les plus populaires ou même des recommandations aléatoires. Ce problème du démarrage à froid se pose aussi lors de l'ajout d'un nouvel item. Celui-ci ne peut pas être recommandé avant d'avoir été noté par un certain nombre d'utilisateurs.

- **La parcimonie (sparsity) :** Le nombre d'items candidats à la recommandation est souvent énorme et les utilisateurs ne notent qu'un petit sous-ensemble des items disponibles. De ce fait, la matrice des notes est une matrice creuse avec un taux de valeurs manquantes pouvant atteindre *95%* du total des valeurs [59]. Les systèmes de filtrage collaboratif ont des difficultés dans ce cas, le nombre de notes à prédire étant largement supérieur aux nombres de notes déjà connues. Le problème de la parcimonie peut être réduit en utilisant les approches par modèles qui réduisent la dimension de la matrice des notes.
- **Le problème du mouton gris (gray sheep) :** Les utilisateurs qui ont des goûts étranges (qui varient de la norme ou qui sortent du commun) n'auront pas beaucoup d'utilisateurs voisins. Il sera donc difficile de faire des recommandations pertinentes pour ce genre d'utilisateurs [Ghazanfar and Prügel-Bennett, 2014]

### 6.2.4. Exemple sur le filtrage collaboratif :

Amazon.com utilise des algorithmes de recommandation pour personnaliser la boutique en ligne pour chaque client. Le magasin change radicalement en fonction des intérêts des clients, montrant des produits de programmation à un ingénieur logiciel ou des jouets pour bébés aux jeunes parents.

L'algorithme d'Amazon.com est basé sur le filtrage collaboratif appliqué aux éléments (Item-based collaborative filtering ou Item-to-Item). Le calcul en temps réel de cet algorithme s'adapte à la fois au nombre de clients et au nombre de produits dans le catalogue.

Amazon.com utilise les recommandations comme un outil de marketing ciblé. Cliquer sur le lien « Vos recommandations » conduit les clients dans une zone où ils peuvent filtrer leurs recommandations par ligne de produit et/ou par domaine, évaluer les produits recommandés,

évaluer leurs achats antérieurs, et comprendre pourquoi les produits sont recommandés (voir la figure 1.4 : ces recommandations pourraient correspondre à l'utilisatrice « *Ouldrabah* », rencontrée dans les différents exemples précédents). De plus, comme illustré sur la figure 1.5, le panier de recommandations propose aux clients des suggestions de produits sur la base des produits dans leur panier. La fonction est similaire aux achats impulsifs dans une ligne de caisse du supermarché, sauf que sur Amazon.com, les achats impulsifs sont ciblés pour chaque client. En effet, Amazon.com utilise des algorithmes de recommandation pour personnaliser son site web en fonction des intérêts de chaque client. Ces algorithmes sont basés sur le filtrage collaboratif Item-to-Item. Le calcul en temps réel s'adapte au nombre de clients et au nombre de produits dans le catalogue. Il fait coïncider chacun des produits achetés et notés par l'utilisateur avec des produits similaires, puis combine ces produits similaires dans une liste de recommandations. Pour déterminer la correspondance la plus proche d'un produit donné, l'algorithme construit une matrice de produits similaires en trouvant les produits que les clients ont tendance à acheter ensemble. Plus précisément, pour construire cette matrice, l'algorithme trouve des produits similaires à chacun des achats et des évaluations de l'utilisateur, agrège ces produits, puis recommande les produits les plus populaires ou les plus corrélés. Ce calcul est très rapide, car il dépend seulement du nombre de produits que l'utilisateur a achetés ou notés.

*Figure 1.4 : Vos recommandations » sur le site Amazon.com.*

*Figure 1.5: Panier de recommandations sur le site Amazon.com.*

### 6.3. Le filtrage hybride :

#### 6.3.1. Définition et Principe :

Un système de recommandation est dit hybride quand il combine deux ou plusieurs approches de recommandation différentes. La recommandation basée sur le contenu et la recommandation collaborative ont souvent été considérées comme complémentaires [47]. Les approches basées sur le contenu ont l'avantage de pouvoir recommander les nouveaux items non encore évalués par un utilisateur, alors que le filtrage collaboratif ne peut recommander un item que s'il a été noté par un certain nombre d'utilisateurs auparavant. Les approches basées sur le contenu nécessitent de disposer des attributs des items, en plus d'une étape d'analyse pour pouvoir les extraire et les représenter, alors que le filtrage collaboratif ne requiert pas d'accès au contenu des items pour pouvoir faire de la recommandation. Il s'appuie uniquement sur la matrice des notes d'utilisateurs pour les différents items. L'hybridation de ces deux techniques, afin de traiter les insuffisances de chaque technique utilisée seule et profiter de leurs points forts, a fait l'objet de plusieurs travaux de recherche. Le système FAB [60] est un des premiers systèmes de recommandation hybrides. Il combine le filtrage collaboratif et une approche basée sur le contenu afin de traiter à la fois le problème du démarrage à froid pour les items et la surspécialisation. Dans ce système, deux critères doivent être satisfaits pour recommander un item : son contenu doit être similaire au profil de l'utilisateur, et il doit être apprécié par les

voisins les plus proches. Il existe plusieurs manières de faire de l'hybridation et aucun consensus n'a été défini par la communauté des chercheurs.

### 6.3.2. Les taxonomies de BURKE (classification) :

Toutefois, [Burke] a identifié sept manières différentes de faire l'hybridation :

- **Pondérée (Weighted) :** les scores fournis par les recommandeurs sont agrégés à l'aide d'une combinaison linéaire ou d'un schéma de vote.
- **Par sélection (Switching) :** le système bascule entre les deux techniques de recommandation en fonction de la situation.
- **Mixte (Mixed)** les recommandations de plusieurs recommandeurs sont disponibles et sont présentées ensemble en même temps au moyen de certaines stratégies de classement ou de combinaison.
- **Par combinaison des propriétés (Feature combination) :** les fonctionnalités utilisées par différents recommandeurs sont intégrées et combinées dans une source de données unique, exploitée par un seul recommandeur.
- **Par augmentation de propriétés (Feature augmentation) :** la sortie d'un programme de recommandation est utilisée en tant que fonction d'entrée supplémentaire pour un autre groupe de recommandation.
- **En cascade** : la recommandation est effectuée comme un processus séquentiel de manière à ce qu'un des recommandeurs affine les recommandations données par l'autre.
- **En définissant un niveau méta** : le modèle généré par l'un des recommandeurs est utilisé comme entrée pour un autre recommandeur. Comme indiqué dans [23] : «Cela diffère de l'augmentation des caractéristiques : dans un hybride de l'augmentation, nous utilisons un modèle appris pour générer des caractéristiques à saisir dans un deuxième algorithme ; dans un hybride de méta-niveau, le modèle entier devient l'entrée. ”

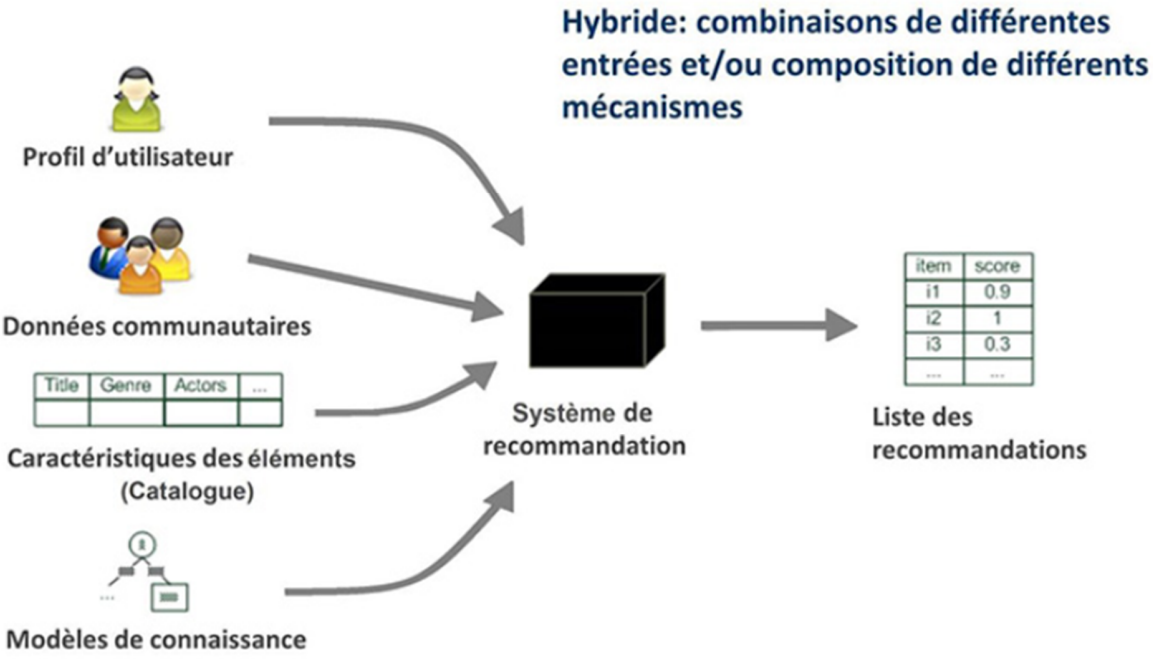


*Figure 1.6: Le système de recommandation hybride*

### 6.3.3. Exemple hybride :

Netflix est aussi un bon exemple de système hybride. Il propose des recommandations en comparant les habitudes de visionnage des films d'utilisateurs similaires (recommandation sociale) et suggère aussi des films qui partagent des caractéristiques avec des films que l'utilisateur a noté positivement (recommandation objet).

## 6.4. Autres systèmes de recommendation :

En plus des trois catégories des systèmes de recommandation présentées précédemment et admises par la majorité des chercheurs de ce domaine, Burke [61] a ajouté trois autres types de systèmes de recommandation : (1) le filtrage démographique, (2) le filtrage à base d'utilité, et (3) le filtrage à base de connaissance.

### 6.4.1. Le filtrage démographique :

Le filtrage démographique est justifié par le fait que les individus ayant certains attributs personnels communs (sexe, âge, pays, etc.) auront également des préférences communes. La

recommandation démographique classifie les utilisateurs selon les attributs de leur profil personnel, et propose des recommandations basées sur des catégories démographiques [62,63].

### 6.4.2. Le filtrage à base d'utilité :

Le filtrage à base d'utilité propose des suggestions basées sur le calcul de l'utilité de chaque objet pour l'utilisateur. De ce fait, une fonction d'utilité doit être créée pour l'utilisateur (Burke, 2002).

### 6.4.3. Le filtrage à base de connaissance :

Le filtrage à base de connaissance suggère des objets basés sur des inférences sur les besoins et les préférences des utilisateurs. Une représentation des connaissances (par exemple : par règles) sur la façon dont un élément satisfait un besoin d'un utilisateur en particulier est nécessaire (Burke, 2002). Pour améliorer la qualité des recommandations, des métadonnées comme des informations sur le contenu des items ont été utilisées comme connaissances additionnelles.

La prochaine section présente une nouvelle approche utilisée dans les systèmes de recommandation. Cette approche exploite la notion d'annotation collaborative ou folksonomie. L'annotation collaborative est employée comme une approche utilisée dans l'analyse automatique des préférences des utilisateurs et les recommandations.

### 6.4.4. La recommandation basée tags :

Dans ces dernières années, une nouvelle génération des systèmes de recommandation boostée par les aspects du web sémantique a émergé. Ces systèmes exploitent des outils tels que les taxonomies, les ontologies, les réseaux sociaux et les annotations. Dans cette section, nous analysons une nouvelle approche qui améliore la compréhension des apprenants. Cette approche introduit les annotations (les tags) dans le processus de recommandation. Nous décrivons d'abord les systèmes d'annotation collaborative (collaborative tagging système), ensuite nous présentons quelques systèmes de recommandation, basés sur le tagging collaboratif, qui ont été proposés dans la littérature.

## 7. Conclusion :

Dans ce chapitre, nous avons présenté une classification en deux approches principales : la recommandation basée sur le contenu et la recommandation basée sur la collaboration. Cette classification simple nous a permis le regroupement d'autres approches présentées parfois séparément.

Toutes ces approches présentent néanmoins des caractéristiques complémentaires et par conséquent un grand nombre de travaux se sont intéressés aux approches hybrides qui combinent plusieurs approches et qui permettent de profiter des avantages et fournir des recommandations plus précises.

# Chapitre II

# Les méthodes de Clustering

## 1. Introduction :

Nous détaillerons au sein de ce chapitre les méthodes de « clustering » ce qui représente l'une des techniques statistiques largement utilisées dans le data mining. Il est dans un cadre d'apprentissage non supervisé, qui tente d'obtenir des informations sans aucune connaissance préalable, ce qui n'est pas le cas de l'apprentissage supervisé.

La question principale autour de laquelle s'articulera le travail du clustering est de synthétiser et de résumer d'énormes quantités de données en les divisant en petits groupes homogènes de sorte que les données (observations) à l'intérieur du même groupe soient plus similaires les unes aux autres que les observations qui appartiennent à d'autres groupes. Cette définition suppose qu'il existe une mesure de qualité de regroupement bien définie qui quantifie le degré d'homogénéité des grappes obtenues. Bien qu'il n'y ait pas de consensus sur ce qu'est une bonne mesure de qualité, elle peut plutôt varier d'une application à l'autre et d'un ensemble de données à l'autre.

Les groupes sont formés en utilisant uniquement les propriétés de chaque élément de l'ensemble de données.

En général, les ensembles de données utilisés pour le regroupement contiennent des points (c'est-à-dire des observations) dans Rn, mais, bien qu'ils puissent être composés d'autres types d'informations telles que des données catégorielles ou des valeurs non numériques, la représentation typique de chaque élément ou modèle est un vecteur. Les types de données suivants pour le regroupement sont pris en compte :

- Variables qualitatives.
- Variables quantitatives (continues)
- Variables nominales et ordinales

## 2. Définition :

Le clustering aussi connu sous nom (Segmentation) est un regroupement en classes homogènes consistent à représenter un nuage des points d'un espace quelconque en un ensemble de groupes appelé Cluster de telle sorte que les données du même cluster aient des caractéristiques similaires, et celles appartenant à des clusters distincts soient dissimilaires [64].

Ce type de méthodes vise à répondre au problème de : diminution de la dimension de l'espace d'entrée, ou pour le groupement des objets en plusieurs catégories (clusters) [65]

«Cluster» est donc une collection d'objets qui sont «similaires» entre eux et qui sont «dissemblables » par rapport aux objets appartenant à d'autres groupes. On peut voir cette définition clairement graphiquement dans l'exemple suivant :

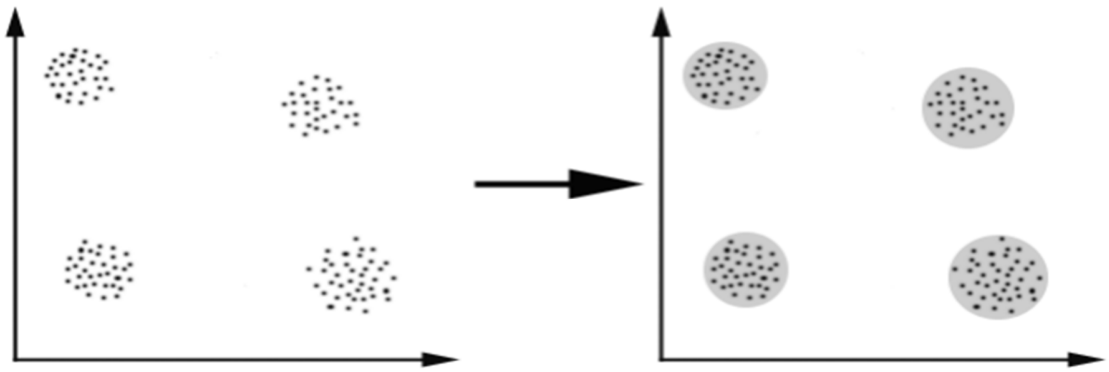

*Figure 2.1 : illustration de regroupement en clusters*

## 3. Les exigences de clustering :

Les principales exigences qu'un algorithme de clustering doit répondre sont les suivantes :

- Evolutivité des clusters
- Traiter les différents types d'attributs
- Découvrir les clusters de forme arbitraire
- Exigences minimales pour la connaissance du domaine afin de déterminer les paramètres d'entrée.
- Capacité de composer avec le bruit et les valeurs manquantes traitées les dimensionnalités élevées. l'intelligibilité et la convivialité

## 4. Les trois étapes principales du clustering :

Le processus de clustering se divise en trois étapes majeures [66] :

- **La préparation des données :**

L'étape de la préparation des données consiste à sélectionner et/ou pondérer ces variables, voire à créer de nouvelles variables, afin de mieux discriminer entre eux les objets à traiter non-supervisé [67].

- **L'algorithme de clustering :**

L'algorithme doit alors être choisi de manière à ce que ses caractéristiques répondent convenablement à ces deux questions :

- Quelle est la nature (qualitative et quantitative) des données ?
- Quelle est la nature des clusters attendus (nombre, forme, densité, etc.) ?

Les critères de décision peuvent être : la quantité de données à traiter, la nature de ces données, la forme des clusters souhaites ou encore le type de schéma attendu (pseudo-partition, partition stricte, dendrogramme … etc.)

- **L'exploitation des résultats de l'algorithme :**

Deux situations sont possibles : soit la tâche de clustering s'inscrit dans un traitement global d'apprentissage, soit les clusters génères par clustering constituent un résultat final.

Dans le premier cas, l'analyse des clusters obtenus (mesures statistiques de qualité) peut aider à orienter le traitement suivant. Une description des clusters n'est pas nécessaire dans cette situation.

En revanche, dans le cas où le clustering constitue à lui seul un processus global de découverte de classes, l'exploitation des clusters pour une application donnée passe par une description de ces derniers. Lorsque les objets se présentent sous la forme d'une matrice de (dis) similarité, il existe peu de méthodes pour décrire les classes (medoides, k objets représentatifs et mesures de cohésion). Lorsque les objets sont décrits par un ensemble de variables, on peut avoir recours à des méthodes de description des classes (descriptions conceptuelles) [68].

## 5. Approche collaborative pour le clustering :

De très nombreuses méthodes de classification non supervisées existent et de nouvelles méthodes ou des améliorations de méthodes existantes sont proposées régulièrement par la communauté de la fouille de données.

La communauté scientifique s'est intéressée ces dernières années à l'utilisation conjointe de plusieurs algorithmes pour traiter les mêmes données. De nombreuses méthodes ont été proposées permettant la mise en œuvre d'approches utilisant plusieurs algorithmes et tirant parti des informations contenues dans les déférents résultats produits. Chacune de ces approches utilise les informations fournies par les méthodes de clustering d'une manière spécifique.
Dans ce cadre, nous allons étudier plus particulièrement le clustering collaboratif.

## 6. les méthodes de clustering :

La classification non supervisée (clustering en anglais) consiste à segmenter un ensemble de vecteurs non labellisés (la classe n'est pas fournie) en groupes (clusters) qui possèdent les propriétés suivantes [69] :

- Homogénéité dans les groupes. les données appartenant à un même cluster doivent être les plus similaires possibles.
- Hétérogénéité entre groupe. les données appartenant à différents clusters doivent être les plus dissemblables possibles.

Il existe une trentaine de méthodes et leurs variations incluent dans différentes familles. Mais peuvent être regroupées en deux grandes catégories [70] :

- Méthodes hiérarchiques :
  - ✓ Agglomératives (ascendantes)
  - ✓ Divisives (descendantes)
- Méthodes non hiérarchiques :
  - ✓ Par partitionnement
  - ✓ Basées sur la densité
  - ✓ Basées sur les grilles

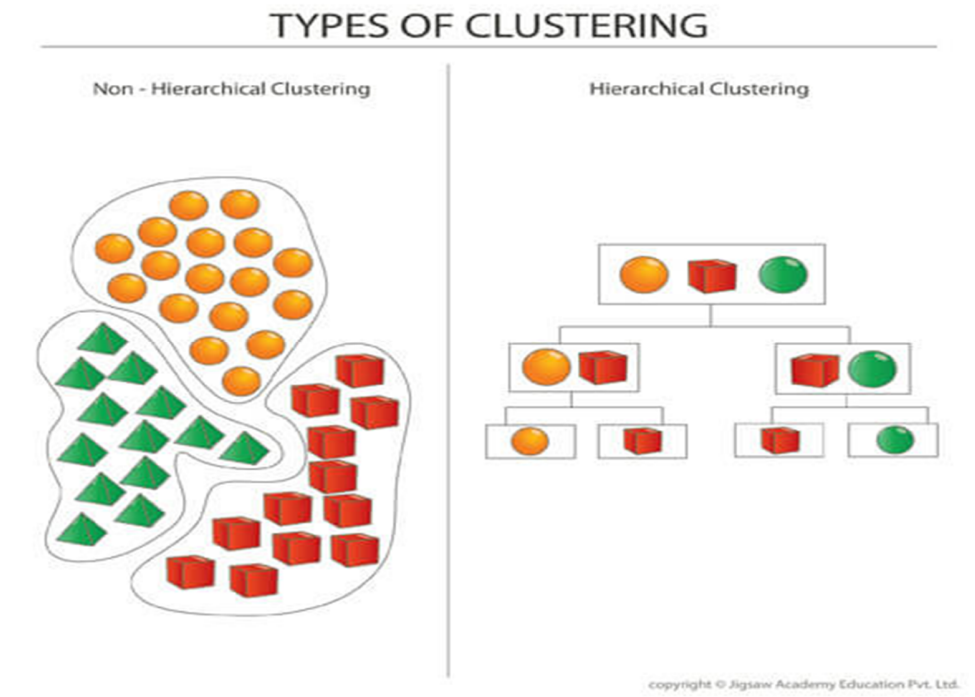


*Figure 2.2 les deux types de clustering non-hiérarchique/ hiérarchique*

Dans le premier cas, on décompose l'ensemble d'individus en une arborescence de groupes.

Dans le 2ème, on décompose l'ensemble d'individus en K groupes,

Les méthodes peuvent être séparées comme suite :

### 6.1. Hiérarchique :

La majorité des méthodes proposent un résultat sous la forme d'une structure plate, c'est à dire sans lien entre les clusters. Il est cependant naturel pour certaines applications de représenter le résultat sous la forme d'une hiérarchie de clusters. Le processus basique des méthodes hiérarchiques a été donné par [Johnson, 1967] [Lance & Williams, 1967],

Ces méthodes construisent une hiérarchie de clusters. Chaque nœud contient ses clusters enfants, et les nœuds frères partitionnent les objets contenus dans leurs parents. Ce type d'approche permet d'explorer les données à différents niveaux de granularité [69].

### 6.1.1. Principe :

Son principe est simple, initialement chaque individu forme une classe, soit *N* classes, donc on cherche à réduire ce nombre de classe newnbrclss < *n* itérativement de sorte que dans chaque étape on fusionne deux classes ensemble (Les deux classes choisies pour être fusionnées sont celles qui sont les plus "proches" en fonction de leur dissimilarité) ou ajouter un nouveau élément à une classe (un élément appartient à une classe s'il est plus proche de cette classe que de toutes les autres) La valeur de dissimilarité est appelée indice d'agrégation. Qui commence dans la première itération faible, et croîtra d'itération en itération [71].

Deux grands types d'approches de clustering hiérarchique existent : les approches par agglomération « ou La classification hiérarchique ascendante (CHA) » et les approches par division « ou La classification hiérarchique descendantes ».

Dans les approches par agglomération, l'algorithme part des objets et ceux-ci sont ensuite regroupés jusqu'à obtenir un cluster unique contenant tous les objets. Les approches divisives partent elles, de l'ensemble des données, et les divisent en clusters qui sont ensuite divisés à leur tour de manière récursive [72].

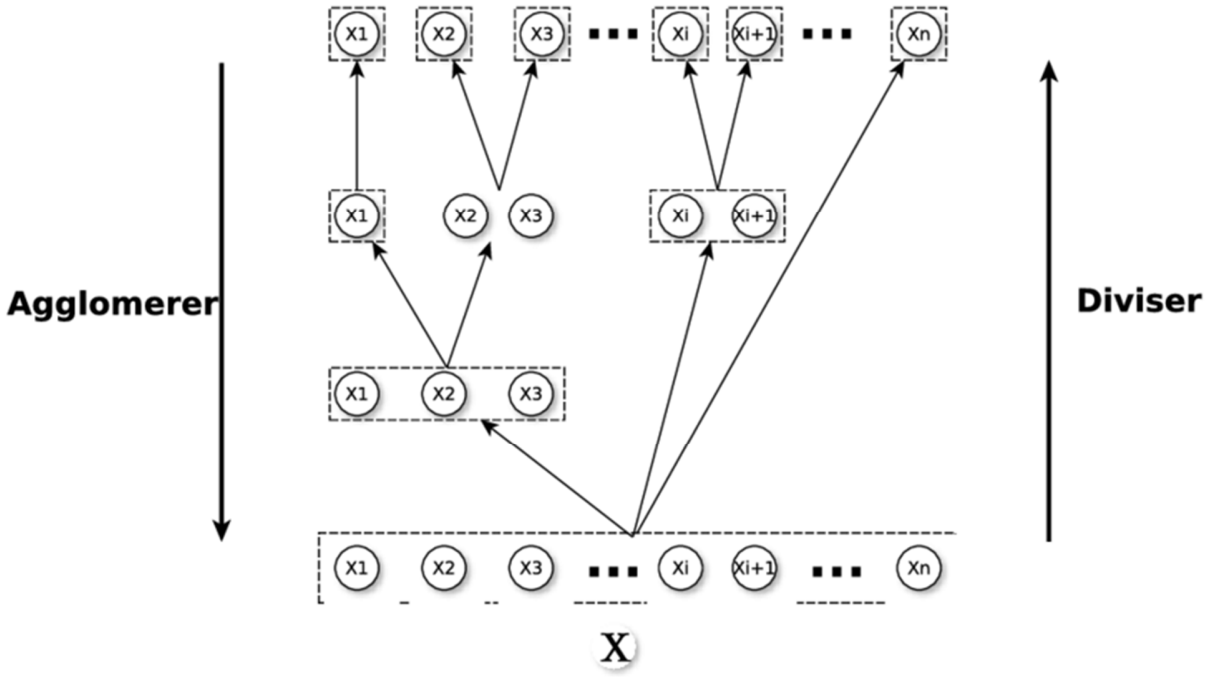


*Figure 2.3 : Principe général du clustering hiérarchique*

### 6.1.2. L'approche par agglomération (ascendante) :

La CHA permet de construire une hiérarchie entière des objets sous la forme d'un "arbre" dans un ordre ascendant. On commence en considérant chaque individu comme une classe et on essaye de fusionner deux ou plusieurs classes appropriées (selon une similarité) pour former une nouvelle classe. Le processus est itéré jusqu'à ce que tous les individus se trouvent dans une même classe. Cette classification génère un arbre que l'on peut couper à différents niveaux pour obtenir un nombre des classes plus ou moins grand. Différentes mesures de distance interclasses peuvent être utilisées , la distance euclidienne, la distance inferieur (qui favorise la création de classes de faible inertie) ou la distance supérieure (qui favorise la création de classes d'inertie plus importante) etc. le cas de la classification ascendante hiérarchique, à partir des éléments, on forme des petites classes ne comprenant que des individus très semblables, puis à partir de celle-ci, on construit des classes de moins en moins homogènes, jusqu'à obtenir la classe tout entière.
Le clustering hiérarchique ascendant ne nécessite pas de prédéfinir un nombre de clusters [73].
Cependant, si un clustering plat des données est nécessaire, il est possible d'effectuer une coupe dans la hiérarchie au niveau qui propose le nombre de clusters demandé. Il est également possible de définir un niveau de similarité entre les clusters au-delà duquel on ne souhaite plus fusionner les clusters. Un certain nombre d'heuristiques existent pour choisir à quel niveau de la hiérarchie il est judicieux d'effectuer une coupe pour obtenir le meilleur partitionnement des données.

#### 6.1.2.1. Le principe général de la CAH :

On suppose que les distances entre tous les objets, deux à deux, ont été calculées suivant l'une des formules (citées dans le sous-titre 6.1.2). On procède alors par étapes successives, chacune d'elles consistant à réunir les deux objets les plus proches. A la fin de chaque étape on recalcule les distances entre le groupe nouvellement créé et le reste des objets. Cela permet de réitérer le processus jusqu'à ce que tous les objets aient été réunis dans un seul groupe. Lorsque cela est achevé on dresse un arbre hiérarchique dont les nœuds représentent les fusions successives, la hauteur de ces nœuds étant égale à la valeur de la distance entre les deux objets, ou groupes, fusionnés. Le niveau des nœuds a donc ainsi une signification concrète ; on dit dans ce cas qu'on obtient une hiérarchie indicé [74].

- **Hiérarchie totale de parties d'un ensemble E :**

*HE* est un sous-ensemble de l'ensemble $\rho$(E) des parties de *E* ayant les propriétés suivantes :

- L'ensemble *E* est un élément de *HE*

- (i) ∈ *HE* pour tout élément de *E*

- h ∩ □′ ∈ {Ø, h, □′} pour h et □′ deux parties quelconque de *HE* autrement dit h et □′ sont soit disjointes, soit incluse l'une dans l'autre.

- **Hiérarchie de partie indicée :**

C'est une hiérarchie de parties *H* à laquelle est associée une échelle d'indices qui satisfait la propriété suivante [74] :

A tout *H* élément de *E* est associé un nombre $v(h) \geq 0$ tel que si $h \subset$ □′ alors $v(h) <$ (□′).

#### 6.1.2.2. Algorithme :

***Algorithme** de **clustering hiérarchique ascendant***

***Entrée :***

- *Un ensemble d'objet X= {$x_1$,$x_{2...}x_n$}*
- *Une matrice de similarité **S.***

***Sortie :***

- *Une hiérarchie **H.***

***Début***

***1. Initialisation***

***1.1.** Chaque individu est placé dans son propre cluster,*

***2. Construction de la hiérarchie***

***2.1.** Répéter*

***2.1.1.** Sélection dans **S** des deux clusters les plus proches $c_i$ et $c_j$*

***2.1.2.** Fusion de $c_i$, et $c_j$, en un cluster Cg plus général.*

***2.1.3**. Mise à jour de S en calculant la similarité entre $c_g$ et les clusters existants*

***2.2.** Jusqu'à la fusion des **2** derniers clusters*

***Fin***

### 6.1.3. L'approche par division (DESCENDANTE) :

Appelée aussi approche divisive, Le clustering hiérarchique descendant est très peu utilisé à cause de la difficulté du choix du critère de division, elle démarre d'un unique cluster contenant l'ensemble des objets [75] [76]. Ce cluster est divisé successivement de manière à ce que les clusters résultant soient les plus dissimilaires possibles. Le processus de clustering se termine lorsque chaque cluster sera réduit à un singleton représentant une feuille de l'arborescence. La difficulté de définir le critère de séparation d'un cluster constitue l'inconvénient majeur de ce type d'approche et c'est la raison pour laquelle il n'existe que peu d'algorithmes divisifs. En effet, pour un cluster composé de n objets, il existe $(2^n - 1)$ possibilités de division.

Elles se fondent en général sur deux processus :

- chercher une classe à scinder.
- choisir un mode d'affectation des objets aux sous-classes.

#### 6.1.3.1. Algorithme :

Les algorithmes de clustering hiérarchiques descendants différent par le critère permettant le choix du cluster à diviser. Ce choix dépend de la nature des variables qui définissent les objets à traiter [77]. Deux approches des plus connues sont : les approches monothétiques et les approches conceptuelles [77].

- **Les approches monothétiques :** ce sont les approches qui construisent simultanément une partition d'objets et une caractérisation de chaque cluster. Elles procèdent par division des clusters selon un assemblage d'éléments qui exprime une condition nécessaire à l'appartenance à un groupe [78]. Nous pouvons trouver plus de détails et des exemples de ce type de méthodes dans [79] [78] notamment sur l'algorithme DIVCLUS-T dans [80].
- **Les approches conceptuelles :** dans le clustering hiérarchique ascendant ce type d'approche est généralisé. L'objectif consiste à compacter des observations sous forme de classes hiérarchiques symboliquement caractérisables à l'aide d'un ensemble de propriétés, à la fois homogènes et contrastées [81] [82].

DIANA (DIvisive ANAlysis) [Kaufman & Rousseeuw, 1990] est l'un des algorithmes les plus utilisés dans ce type d'approche. Cet algorithme permet d'éviter l'exploration de toutes les

possibilités de séparation en commençant par déterminer l'objet le plus dissimilaire (atypique) d'un cluster et de lui associer les objets les plus proches de façon à définir deux clusters.

***Algorithme : DIANA***

***Entrée =***

- *Un ensemble d'objet X= $\{x_1,x_2...,x_n\}$*
- *Une matrice de similarité S.*

***Sortie :***

- *Une hiérarchie H.*

***Début***

***1. Initialisation***

*1.1. Initialiser le cluster racine C= $\{(x_1, x_2..., x_n\}\}$.*

*1.2- H=C*

*1.3. Sélectionner le cluster c E C ayant le diamètre maximal.*

***2. Construction de la hiérarchie***

*2.1- Déterminer l'objet $x^*$ le plus atypique (dissimilaire) du cluster c tel que :*

$$x^* = arg\,min \frac{1}{|c|-1} \sum_{j \neq i} s(x_i\ , x_{j)}$$

*2-2. Construire le cluster $c^*$={x*}.*

*2.3. Pour chaque objet $x_i \notin$ c calculer :*

$$s_i = [Moy\ (s(x_i\ , x_j))]_{x_i \in c \setminus c^+} - [Moy\ (s(x_i\ , x_j))]_{x_i \in c^+}$$

*2.4-. Déterminer $x_k$ tel que :*

$$x_k = arg\,min(s_j)$$

*2.5. Si $x_k$ <0 alors $c^*=c^*\ U\{x_k\}$.*

*2.6. Répéter les étapes 2.1. 2.2 Et 2.3 jusqu'é ce que $x_k$>0*

*2.7. Remplacer c par c\c* et $c^*$ dans C.*

*2.8. Ajouter c\c*et c* clans H.*

*2.9. Répéter les étapes de 1.3 é 2.8 jusqu'é ce l'ensemble C soit réduit é un singleton.*

*2.10. Retourner H.*

***Fin***

**6.2. Non hiérarchique :**

**6.2.1. Le clustering par partitionnement :**

Apparu au milieu du 20ème siècle, le partitionnement de données appartient au domaine du machine learning et plus précisément à la classification automatique non supervisée. On se place donc dans le cadre où le nombre de groupes et les caractéristiques communes de ces groupes au sein de l'échantillon ne sont pas connus par l'expérimentateur [83].

Le clustering par partitionnement Cherche à décomposer directement la base de données dans un ensemble de clusters disjoints. Plus spécifiquement, ils essaient de déterminer un nombre de partitions qui optimisent certains critères (fonction objective). La fonction objective peut mettre en valeur la structure locale ou globale des données et son optimisation est une procédure itérative.

Nous distinguons deux types d'approches de classification par partition :

- ✓ les k-centroïdes
- ✓ les k-médoïdes

Ces deux stratégies se différencient par le type des représentants des grappes.

#### 6.2.1.1. Méthodes basées sur les k-centroïdes :

Nous introduisons en premier la méthode des centres mobiles, qui est la première, selon notre connaissance, à avoir utilisé ce principe. Ensuite, nous présentons la plus connue de ce type de méthodes : la méthode k-moyennes.

##### 6.2.1.1.1. Méthode des centres mobiles :

Elle a été développée par Forgy en 1965 [77] [84].Son principe est simple, il consiste à construire une partition en k classes en tirant aléatoirement k éléments de l'ensemble des données E. Ces points sont définis comme les représentants des classes. Les autres éléments sont alloués un à un à la classe du plus proche centre, ainsi k classes sont construites. Ensuite, les centroïdes (Définition 6-2-1-1) des classes sont calculés et définis comme les nouveaux représentants des classes auxquelles ils appartiennent. Un processus de réattribution des éléments est appliqué à nouveau mais selon les nouveaux représentants, dits "centres". Cette dernière étape est répétée soit jusqu'à ce que les classes deviennent stables, soit un nombre de fois défini par l'utilisateur [85].

Les détails de cet algorithme sont décrits ci-dessous :

***Procédure Centres Mobiles*** *:*

- ***Variables*** *:*

*1. Ensemble E de taille n :*
*2. d dissimilarité, k nombre de classes :*

- ***Début :***

*1. Choisir aléatoirement k centres de E :* $x_1^*, x_2^*, x_3^*, \dots, x_k^*$
*2 Construire la partition initiale* $p_0 = \{c_1, c_2, c_3, \dots., c_k\}$*ou chaque élément* $x_i \in E$ *est affecté à la classe du centre le plus proche, tel que :*

***Pour*** *i de 0 à n affecter* $x_i$ *a la classe du centroide le plus proche :*

$$p_m = \{x_i \in E\ d(x_i, x_j^*) = min_{j=1\dots k} d(x_i, x_j^*)\}$$

***Fin pour***

*3. Calculer les centroides des classes résultantes.*
*4. Déterminer ces centroides comme les nouveaux centres des classes*
*5. Refaire l'étape 2 selon les nouveaux centres ;*
*6. Retourner le résultat. // L'algorithme peut être arrêté après un nombre d'itérations déterminé par l'utilisateur ou après stabilité des classes.*

***Fin***

La méthode des centres mobiles a connue plusieurs améliorations, la plus commune est la méthode des k-moyennes.

#### 6.2.1.1.1.1. Méthode des K-moyennes :

k-means dans le but de maximiser la séparabilité et la compacité des groupes et ceci en choisissant l'objet le plus mal classé comme étant le nouveau centre du groupe. L'utilisation de cette technique peut se faire soit par intégration directe dans la phase d'initialisation du classique k-means ou bien dans une approche incrémental comme dans le cas du global k-means. Nos expériences montrent que cette technique produit des groupes qui sont similaire à celle du global k-means en un temps très réduit [86].

- **k-means :**

K-means définit par McQueen [87] est un des plus simples algorithmes de classification automatique des données. L'idée principale et de choisir aléatoirement un ensemble de centres fixé a priori et de chercher itérativement la partition optimale. Chaque individu est affecté au centre le plus proche, après l'affectation de toutes les données la moyenne de chaque groupe est calculé, elle constitue les nouveaux représentants des groupes, lorsqu'ont abouti à un état stationnaire (aucune donnée ne change de groupe) l'algorithme est arrêté.

***Algorithme****. 1 : K-means*

***Entrée***

*Ensemble de N données, noté par x*

*Nombre de groupes souhaité, noté par k*

***Sortie***

*Une partition de K groupes* $\{c_1, c_2, c_3, \dots, c_k\}$

***Début***

*1. Initialisation aléatoire des centres* $c_k$

***Répéter***

*2. Affectation : générer une nouvelle partition en assignant chaque objet au groupe dont le centre est le plus proche :*

$$x_i \in c_k si \nvdash j \; |x_i - u_k| = min|x_i - u_j| \qquad 1$$

*Avec* $u_k$ *le centre de la classe K ;*

*3. Représentation : Calculer les centres associe à la nouvelle partition ;*

$$u_k = \frac{1}{N} \sum_{x_i \in c_k} x_i \qquad 2$$

***Jusqu'à** convergence de l'algorithme vers une partition stable ;*

***Fin.***

- **Global k-means :**

Est une solution au problème d'initialisation du k-means, elle est fondé sur les données et vise à atteindre une solution globalement optimale. Elle consiste à effectuer un clustering incrémental et à ajouter dynamiquement un nouveau centre suivi par l'application du k-means jusqu'à la convergence. Les centres sont choisis un par un de la façon suivante : le premier centre est le centre de gravité de l'ensemble des données (résultat de l'application du k-means avec *k=1*), les autres centres sont tirés de l'ensemble de données ou chaque donnée est une candidate pour devenir un centre, cette dernière sera testée avec le reste de l'ensemble, le meilleur candidat est celui qui minimise la fonction objectif , l'algorithme suivant permet d'illustrer le principe [88] :

***Algorithme 2** : Global k-means*

***Entrée***

*Ensemble de N données, notés par x :*

*Nombre de groupes souhaiter, noté par k ;*

***Sortie***

*Une partition de K groupes* $\{c_1, c_2, c_3, \dots, c_k\}$

***Début***

*1.* $c_i$ *= Centre de gravité de l'ensemble des données ;*

***Répéter***

*2. Initialiser les centres i-1 par le résultat de l'étape précédente ;*

*3. Trouver l'$i^{eme}$ centre :*

***Pour cheque*** *donnée x* ***faire***

*3.1. Considéré x comme étant 1e $i^{eme}$ centre*

*3.2. Affecter les données aux plus proche centre ;*

*3.3. Calculer l'erreur quadratique pour*

*$c_i$=x :*

***Fin faire***

*3.4. Garder 1e centre $c_i$== x qui minimise l'erreur quadratique ;*

*4. Appliquer le k_means jusqu'à la convergence :*

***Jusqu'à*** *obtenir une partition en k groupes*

***Fin***

#### 6.2.1.2. k-medoids :

L'algorithme k-medoids développé par Kaufman et Rousseeuw [Kaufman & Rousseeuw, 1990] est une adaptation de l'algorithme k-means. Au lieu de calculer le centre de gravité de chaque cluster, un élément représentatif, appelé « médoid », est choisi dans chaque cluster à chaque itération. Le médoid de chaque cluster est calculé en trouvant un objet xi à l'intérieur du cluster qui minimise la somme suivante : $\sum j \in c_i d(x_i, x_j)$. [2.1]

Où $C_i$ est le cluster contenant l'objet i et $d\ (i, j)$ la distance entre les objets $x_i$ et $x_j$.

***Algorithm : k-medoids***

***Entrée :***

- *Un ensemble d'objet X= {$x_1$, $x_2$ ...,$x_n$}*
- ***K :*** *le nombre de clusters.*

***Sortie :***

- *C= {$c_1, c_2, c_3, \dots, c_k$} : un ensemble de clusters.*

***Début :***

*1. Choisir aléatoirement les k premiers médoids.*

*2. Associer chaque médoid a un cluster.*

*3. Affecter chaque objet au cluster représenté par le médoid le plus proche a l'objet*

*4. Recalculer les positions des* ***k*** *médoids.*

*5. Répétez les étapes 3 et 4 jusqu'é ce que l'ensemble des clusters se stabilise.*

***Fin***

### 6.3. Le clustering par densité :

Ces méthodes considèrent les classes comme étant des régions denses dans l'espace d'objets. Un ensemble ouvert dans l'espace Euclidien peut être divisé en un ensemble de ses composantes

connectées. L'implémentation de cette idée pour le partitionnement d'un ensemble fini de points a besoin de tels concepts comme la densité, la connectivité et les limites (extrémité, bordure, frontière) plutôt que la distance. Ils sont fortement reliés aux plus proches voisins d'un point. Un cluster, défini comme une composante densément connectée, évolue dans n'importe quelle direction où la densité le mène. Par conséquent, les algorithmes à base de densité sont capables de découvrir des clusters de formes arbitraires. Ainsi, ce principe fournit une protection naturelle contre les données aberrantes. Les clusters ne sont pas nécessairement convexes (entièrement contenus/remplis). Le bruit et les données isolées sont identifiés. Le nombre de classes n'a pas besoin d'être prédéfini préalablement. Un seul parcours de la base de données est effectué pour la réalisation de ces clusters. La méthode par densité voit les clusters comme des régions denses séparées par des régions qui le sont moins. Le principe :

- un point est voisin d'un autre point s'il est à une distance inférieure à une valeur fixée, cette valeur est à choisir selon l'échelle de l'étude et l'exploitation de la solution recherchée.
- un point est dense si le nombre de ses voisins dépasse un certain seuil. Ce critère est aussi défini par l'utilisateur selon les objectifs de son étude.

Il existe deux types d'approches à base de densité. Le premier type d'approche attache la densité a un point qui appartient aux données d'apprentissage. Il compte des algorithmes spécifiques comme DBSCAN, GDBSCAN, OPTICS et DBCLASD. Le deuxième type d'approche attache la densité à un point dans l'espace d'attributs. Il inclut l'algorithme DENCLUE [89].

- **L'algorithme DBSCAN**

L'algorithme **DBSCAN** (Density-Based Spatial Clustering of Applications with Noise) a été introduit en 1996 dans ce but. Cet algorithme est très utilisé, raison pour laquelle il a obtenu en 2014 une distinction de contribution scientifique ayant résisté à l'épreuve du temps [90].

DBSCAN itère sur les points du jeu de données. Pour chacun des points qu'il analyse, il construit l'ensemble des points atteignables par densité depuis ce point : il calcule l'epsilon-voisinage de ce point, puis, si ce voisinage continent plus de *N-min* points, les epsilon-voisinages de chacun d'entre eux, et ainsi de suite, jusqu'à ne plus pouvoir agrandir le cluster. Si le point considéré n'est pas un point intérieur, c'est à dire qu'il n'a pas suffisamment de voisins, il sera alors étiqueté comme du bruit. Cela permet à DBSCAN d'être robuste aux données aberrantes puisque ce mécanisme les isole [91].

La démarche générale de DBSCAN se résume comme suit [91] :

1. Sélectionner aléatoirement un objet $p$ de l'ensemble d'objets $X$.
2. Déterminer l'ensemble $E$ des objets accessible depuis $p$ dans $X$.
3. Si $p$ est un noyau, alors $E$ est une classe.
4. Sinon sélectionner un autre objet et aller en (2).
5. S'arrêter lorsque tous les objets ont été sélectionnés.

***L'algorithme : DBSCAN***

*1. Prendre un point x qui n'a pas été visité*
*2. (construire N = voisinage (eps, x)*
*3.* ***If*** *l N l < n_min :*
*Marquer x comme bruit*
***Else :***
*Initialiser C {x}*
*Agrandir cluster(c, N, cps, n_min)*
*Ajouter C a la liste des clusters*
*Marquer tous les points de C come visités*
*4.* ***Répéter*** *1-3* ***until*** *tous les points ont été visités*

### 6.4. Autres méthodes :

#### 6.4.1. Grille :

L'idée de ces méthodes est qu'on divise l'espace de données en un nombre fini de cellules formant une grille. Ce type d'algorithme est conçu pour des données spatiales. En fait, avec une telle représentation des données, au lieu de faire la classification dans l'espace de données, on le fait dans l'espace spatial en utilisant des informations statistiques des points dans la cellule. Les méthodes de ce type sont hiérarchiques ou de partitionnement. Les algorithmes les plus connus sont STING [92], CLIQUE [93] [94]. Le partitionnement des données est induit par l'appartenance des points dans des segments qui résultent de l'espace de partitionnement, alors que l'espace de partitionnement est basé sur les caractéristiques de la grille extraites des données d'entrée. Un avantage de ce traitement indirect est que l'accumulation des données sous forme de grille rend les techniques à base de grille indépendante par rapport à l'ordre (rangement) des données. En revanche, les méthodes par réaffectation et les méthodes incrémentales sont très sensibles à l'ordre

des données. Tandis que les méthodes à base de densité fonctionnent mieux sur des variables numériques, les méthodes à base de grille fonctionnent avec des variables de différents types. Dans une certaine mesure, les méthodes à base de grille reflètent un point de vue technique. Cette famille d'approches est éclectique : elle contient les deux types d'algorithmes : hiérarchiques et de partitionnement.

Les différentes étapes de l'algorithme WaveCluster (Wavelet-Based Clustering) sont présentées dans l'algorithme suivant :

***Algorithme: Wavelet-Based Clustering***
***Entrée:***
*X* ***:*** *Un ensemble de données multidimensionnelles.*
***Sortie :***
*Un ensemble de clusters*
***Début :***
*1. Découper l'espace des attributs en cellules.*
*2. Affecter chaque objet a une cellule.*
*3. Appliquer 1a transformation par ondelettes.*
*4. Rechercher les composants connectés (clusters)*
*5. Affecter les objets aux clusters.*
***Fin***

### 6.4.2. Probabiliste :

Ces méthodes supposent que les données suivent une certaine loi de probabilité. L'objectif est d'estimer les paramètres de cette loi et de définir un modèle de mélange de lois pour représenter les différents clusters. Ces méthodes font l'hypothèse qu'à chaque cluster $C_i$ est associée une loi de probabilité $P(x, \theta i)$ de paramètres $\theta_i$ qui permet de déterminer la probabilité d'appartenance de $x$ à $C_i$. Si on note $\pi_i$ la proportion de la I$^{ème}$ loi dans le mélange, les paramètres du modèle sont : $\Phi = (\pi_1,..., \pi_k, \theta_1, ..., \theta_k)$ et la fonction de densité est :

$$p(x, \Phi) = \sum_{i=1}^{k} \pi_i P(x, \theta_i) \quad [2.2]$$

Les méthodes de clustering probabilistes cherchent à approximer les paramètres Φ du modèle. La probabilité d'appartenance aux clusters peut être interprétée comme un degré d'appartenance à ce cluster. On pourra notamment citer l'algorithme EM [95]. A noter qu'en général, les lois considérées sont supposées gaussiennes.

### 6.4.3. Le clustering basé sur les graphes :

Dans ce type de méthodes, un cluster est représenté sous forme d'un graphe dont les sommets correspondent aux objets et dont les arcs correspondent aux distances entre ces objets. La méthode consiste à générer un graphe connexe dont la somme des valeurs des arcs est minimale. La principale limite de ce type d'approches réside dans la complexité de construction du graphe minimal [96].

### 6.4.4. Le clustering stochastique :

Etant donnée une mesure de qualité d'un clustering, ce type de méthodes consiste à parcourir l'espace de solutions à la recherche d'une solution maximisant la mesure de qualité définie en entrée. La recherche Tabou [97], le recuit simulé [98] et les algorithmes génétiques [99] constituent des exemples de méthodes utilisées dans l'approche stochastique

## 7. Comparaison entre les méthodes de clustering :

| | **Hiérarchique** | **Partitionnement** | **Densité** | **Grilles** | **Graphe** | **Stochastique** |
|---|---|---|---|---|---|---|
| **Connaissance à priori** | Nombre de clusters ou seuils | Nombre de clusters | Critère de densité de voisinage | Taille de grille et critère de densité | Distance minimale entre clusters | Nombre de clusters |
| **Présentation des résultats** | Hiérarchie | K centroides | Partition | Ensemble de Cellules connectées | Partition sous forme de graphe | Partition |
| **Complexité** | $O(m \times n^2)$ | $O(m \times n \times k)$ | $O(m \times n^2)$ | $O(m \times$ taille de la grille$)$ | $O(m \times n^2 \log n)$ | $O(m \times n \times k)$ |
| **Tolérance au Bruit** | Non | Non | Oui | Oui | Non | Oui |

*Tableau 2.1: Etude comparative entre les méthodes de clustering*

## 8. Mesure de similarité :

La similarité est une partie importante de la définition d'une méthode de clustering, selon [Bisson, 2000], «tout système ayant pour but d'analyser ou d'organiser automatiquement un ensemble de données ou de connaissances doit utiliser, sous une forme ou une autre, un opérateur de similarité dont le but est d'établir les ressemblances ou les relations qui existent entre les informations manipulées».

Donc, en effet elle consiste à définir et formaliser une mesure de similarité entre deux éléments $i$ et $j$ est noté $s(i, j)$. La similarité entre les données à étudier est l'information principale (et souvent unique) permettant à l'algorithme de classification de partitionner le jeu de donnée. Le plus souvent, cette similarité est calculée selon une mesure de "distance" adaptée au type de données et au problème à résoudre. Une distance, noté de manière générale $d(i, j)$, est une mesure de similarité qui vérifie certaines propriétés mathématiques (symétrie, séparation et inégalité triangulaire). Lorsque les données sont décrites dans un espace vectoriel, la distance la plus utilisée est la distance Euclidienne, alors notée $\|i - j\|$.

### 8.1. Calcul de la similarité :

Le calcul de la similarité a pour objectif de déterminer dans quelle mesure deux utilisateurs ou deux items sont similaires. Il existe plusieurs façons de calculer cette similarité, cependant les méthodes les plus utilisées et qui présentent les meilleurs résultats sont :

- Cosinus :
- Coefficient de corrélation de Pearson
- Le cosinus ajusté (adjusted cosine) :
- Index de Jaccard

### 8.2. Fonctions de distance :

la distance $d(i, j)$ entre les points $x_i$ et $x_j$ ,est ensuite mesurée dans cet espace grâce à l'utilisation d'une fonction de distance dans R+, respectant les propriétés suivantes [100] :

- $d(i, j) \geq 0$ (contrainte de positivité),

- $d(i, j) = 0$ si et seulement si $x_i = x_i$

- $\forall x_i, x_j$, $d(i, j) = d(j, i)$ (contrainte de symétrie),

- $\forall x_i, x_j$, $xl$, $d(i, j) \leq d(i, l) + d(l, j)$ (contrainte d'inégalité triangulaire).

Dans la littérature, il est aisé de trouver plusieurs définitions de métrique de distance. Cependant, toutes ces mesures ont une interprétation identique : il est très probable que des points proches dans l'espace de projection représentent des données d'un même groupe (la distance est alors proche de $0$), alors que des points éloignés représentent, quant à eux, des données appartenant à des groupes différents (la distance tend vers l'infini). La liste suivante non exhaustive, reprend les plus classiques :

- ✓ **La distance de Minkowski** : avec $p$ un entier positif non nul.

$$d(i,j) = \sqrt[p]{\sum_{r=1}^{M} | x_{ir} - x_{jr}|^p} \qquad [2.3]$$

- ✓ **La distance euclidienne** est la plus connue et la plus utilisée. Elle peut être vue comme un cas particulier de la distance de Minkowski pour $p = 2$ :

$$d(i,j) = \sqrt[p]{\sum_{r=1}^{M} | x_{ir} - x_{jr}|^2} \qquad [2.4]$$

- ✓ **La distance de Manhattan** est également un cas particulier de la distance de Minkowski pour $p = 1$ (également appelée "métrique absolue") :

$$d(i,j) = \sum_{r=1}^{M} | x_{ir} - x_{jr}| \qquad [2.5]$$

## 9. Les limites et les points forts :

### Les points forts :

- L'opportunité de transformer les heures ou jours d'indisponibilité en minutes, ou moins, dans le rare cas d'une panne matérielle ou logicielle catastrophique
- sa complexité linaire par rapport au nombre d'objets à traiter
- Applicabilité a tout type d'attribut.

### Les limites :

- Le résultat final est fortement dépendant du choix des clusters initiaux
- La génération des clusters vides.
- traitement d'un grand nombre de dimensions et de grand nombre de données, question peut Être problématique en raison de la complexité du temps de calcule
- l'efficacité de la méthode dépend de la définition de «distance» utilisée
- Beaucoup d'algorithmes de clustering exigent la spécification du nombre de clusters à Produire en entrée de l'ensemble de données, avant l'exécution de l'algorithme

## 10. Conclusion :

Dans ce chapitre, nous avons fait un tour d'horizon des principaux concepts et définitions du clustering. Nous constatons aisément au vu Les exigences et les étapes principales du clustering, est nous avons étudié les grands concepts, les principales méthodes existantes avec qu'un nombre important, qu'aucune méthode ne peut être qualifiée de meilleure par rapport à une autre sur l'ensemble des applications envisageables, Ainsi que leur comparaison et les nombreux limites et avantage.

# Chapitre III

# Proposition

# Et évaluation

## 1. Introduction :

Les systèmes de recommandation personnalisés (SR) aident les utilisateurs à réduire le temps nécessaire à la recherche des items qui les intéressent. Le filtrage collaboratif (FC) est l'une des techniques les plus performantes en SR. La fragmentation des données rend le résultat de la recommandation inexact. Dans ce travail, une nouvelle méthode de classification évolutive est utilisée. Notre but est d'exploiter un algorithme dont l'objectif sera le regroupement d'utilisateurs qui partagent les mêmes intérêts dans le même cluster et de leur recommander des items qui pourraient les intéresser. La performance de notre proposition a été validée par des expériences sur un dataset réel, comparée aux algorithmes de recommandation existants. Dans ce chapitre nous essayons de détailler notre proposition, en présentant le dataset utilisé dans l'expérimentation ainsi que nous discutons les résultats obtenus.

## 2. Présentation des systèmes de recommandation à base de clustering :

Les algorithmes de clustering sont souvent utilisés dans les systèmes de recommandation pour regrouper les utilisateurs (items) en différents clusters [101]. Une matrice bidimensionnelle est souvent utilisée comme matrice de connexion pour les utilisateurs de cluster. Zhang et coll. regroupement des utilisateurs par un algorithme de k-means basé sur les statistiques des comportements des utilisateurs sur différents genres d'items, ce qui a amélioré la qualité de la recommandation. Les statistiques des comportements des utilisateurs sur différents genres d'items ont bien fonctionné pour mesurer les intérêts des utilisateurs et les regrouper [102]. La matrice d'évaluation d'utilisateur-item a été utilisée pour diviser les utilisateurs en plusieurs groupes (cluster), ce qui a amélioré les performances de prédiction [103, 104]. La matrice de notation utilisateur-item a été utilisée pour regrouper les utilisateurs en utilisant différentes méthodes de regroupement. L'approche floue nous offre une solution peu coûteuse pour produire des recommandations de haute qualité et une meilleure robustesse des systèmes de filtrage collaboratifs préservant la confidentialité grâce à son modèle basé sur l'approximation [105]. Certaines méthodes de clustering ont été largement utilisées pour le regroupement d'utilisateurs dans la recherche sur l'apprentissage automatique. Ce sont : Autoclass [106] et l'algorithme d'extraction de cluster à base des graphes [107].

## 3. Notre système de recommandation basé sur clustering :

### 3.1. Le modèle dynamique :

À travers de nombreuses littératures, nous constatons que le modèle dynamique peut produire de meilleurs résultats, du fait de la rapidité de sa vitesse de détection ainsi que les bonnes performances de clustering. Nous utilisons donc cet algorithme pour regrouper les utilisateurs qui possèdent un intérêt similaire. Le principal mécanisme de la dynamique est que les nœuds à l'intérieur des communautés atteignent des valeurs d'état égales par contre on note l'échappement des nœuds entre les communautés [108].

- **kuramoto** a conçu un modèle généralisé pour les réseaux positifs. Il a proposé ce qui suit :

$$\frac{d\theta_i}{dt} = \omega_i + \frac{K_p}{N}\sum_{j=1}^{N} a_{ij}\ \sin(\theta_j - \theta_i) + \frac{K_n}{N}\sum_{j=1}^{N}(1 - a_{ij})\ \sin(\theta_j - \theta_i) \qquad [3.1]$$

Où $\omega_i$ est la fréquence intrinsèque de l'i$^{ème}$ oscillateur (nœud). *N* est le nombre de nœuds du réseau. $\theta_i$ est la variable de phase. Si le i$^{ème}$ nœud est connecté au j$^{ème}$ nœud, $a_{ij} = 1$, sinon $a_{ij} = 0$. $K_p > 0$ signifie que les phases de deux nœuds connectés dans le réseau évoluent ensemble, et $K_n < 0$ signifie que les phases de deux nœuds non connectés dans le réseau évoluent loin.

- **Chen et all [109].** a conçu un modèle de réseau de clustering dynamique pour imiter l'état d'évolution dynamique de chaque nœud dans [110]. Le modèle est donné comme suit :

$$\dot{x}_i(t) = \sigma x_i(t)\ (\overline{x_i}(t) - x_i(t)) \qquad [3.2]$$

Où $x_i$(t) est l'état du i$^{ème}$ nœud au temps t, $\sigma > 0$ est le paramètre évolutif, $\overline{x_i}(t) - x_i(t)$ signifie l'état de différence entre son propre et ses voisins positifs.

- **Liji U. et al [111]** ont inspiré par les deux méthodes de regroupement évolutif dynamique ci-dessus, où ils ont laissé les utilisateurs possédant une distance plus petite, évolués ensemble, si un certain principe est respecté, sinon, les utilisateurs évoluent loin. Ainsi, les utilisateurs d'une même communauté obtiennent des états cohérents lorsqu'ils

sont stables. Différentes valeurs d'état révéleront la structure de la communauté en fonction de leurs différentes valeurs stables.

### 3.2. Mesures de similarité :

Afin de prédire les notes inconnues, la similarité entre les utilisateurs doit être calculée au préalable en fonction de leurs notes d'historique. Il existe plusieurs mesures pour calculer la similarité : cosinus (COS), cosinus ajustée (ACOS), le coefficient de corrélation de Pearson (PCC) [112,113] et le coefficient de Jaccard…etc.

- **La similarité cosinus :**

  D'une manière générale, Cosinus est une mesure de similarité entre deux objets a et b. Très utilisée en recherche d'informations [Salton, 1989], Dans le cas du filtrage collaboratif, chaque utilisateur u est représenté par un vecteur $x_u$, où $x_{ui}$=$r_{ui}$. Pour pouvoir calculer la similarité entre deux utilisateurs $u$ et $v$, le cosinus est calculé sur l'ensemble des items notés par les deux utilisateurs comme l'illustre la formule suivante :

$$\text{sim}(u,v)^{COS} = \frac{\sum_{i=1}^{M} r_{u,i}\ r_{v,i}}{\sqrt{\sum_{i=1}^{M} r^2{}_{u,i}}\ \sqrt{\sum_{i=1}^{M} r^2{}_{v,i}}} \qquad [3.3]$$

  Ici, M est le nombre de tous les items, $r_{u,i}$ est le score de l'item i donné par l'utilisateur $u$.

  Le cosinus peut aussi s'appliquer pour calculer la similarité entre deux items. En effet, il suffit de remplacer dans l'équation précédente les utilisateurs par leurs équivalents en items.

  Le cosinus varie entre 0 et 1. Une valeur égale à 1 indique que les deux utilisateurs ont des préférences identiques, une valeur égale à 0 indique qu'ils n'ont rien en commun. Un inconvénient majeur de l'utilisation du cosinus dans le filtrage collaboratif est qu'il ne tient pas compte de la variation dans le jugement des utilisateurs.

- **Le paramètre Ajuster cosinus :**

Lorsque la similarité de Pearson est utilisée pour calculer la similarité entre deux items, les notes d'un même utilisateur sont centrées par rapport à la moyenne de ses notes. Or, la variation de notation pour un même utilisateur n'est pas aussi importante que la variation entre les différents utilisateurs. C'est pour cette raison qu'il est plus intéressant, lors du calcul de la similarité entre deux items, d'ajuster les notes par rapport à la moyenne des notes des utilisateurs plutôt que par rapport à la moyenne des notes des items. C'est le rôle du Cosinus ajusté, qui a été introduit par [Sarwar et al. 2001]. Selon [Schafer et al. 2007], le cosinus ajusté est considéré comme l'un des moyens les plus efficaces et populaires pour calculer la similarité entre deux items pour les algorithmes de filtrage collaboratif.

$\text{sim}(u, v)^{ACOS}$ est définie comme suit :

$$\text{sim}(u, v)^{ACOS} = \frac{\sum i \epsilon I_{uv} \ (r_{u,i} - \overline{r_u})(r_{v,i-} - \overline{r_v})}{\sqrt{\sum i \epsilon I_u (r_{u,i} - \overline{r_u})^2} \ \sqrt{\sum i \epsilon I_v (r_{v,i-} - \overline{r_v})^2}} \quad [3.4]$$

Ici, $\overline{r_u}$ est le score moyen des items sur lesquels l'utilisateur $u$ a déjà noté. $\overline{r_v}$ Est le score moyen des items sur lesquels l'utilisateur $v$ a déjà noté. $I_{uv}$ Est l'ensemble de tous les items co-évalués par l'utilisateur $u$ et l'utilisateur $v$. $I_u$ est l'ensemble de tous les items notés par l'utilisateur $u$. $I_v$ est l'ensemble de tous les items notés par l'utilisateur $v$.

- **<u>Coefficient de corrélation de Pearson</u> :**

Ce coefficient a été utilisé notamment par les auteurs du système GroupLens [Resnick et al. 1994] pour calculer la similarité entre deux utilisateurs $u$ et $v$. Le coefficient de corrélation de Pearson mesure le rapport entre la covariance et le produit de l'écart-type des notes données par les deux utilisateurs. Il permet ainsi de mesurer la similarité en utilisant les items notés à la fois par $u$ et $v$. Plus les deux utilisateurs auront tendance à noter les mêmes items de façon équivalente, plus ils seront similaires comme l'illustre la formule suivante :

$$\text{sim}(u,v)^{PCC} = \frac{\sum i \epsilon I_{uv} \ (r_{u,i} - \overline{r_u})(r_{v,i-} - \overline{r_v})}{\sqrt{\sum i \epsilon I_{uv} (r_{u,i} - \overline{r_u})^2} \ \sqrt{\sum i \epsilon I_{uv} (r_{v,i-} - \overline{r_v})^2}} \quad [3.5]$$

- **Coefficient de Jaccard :**

  L'indice de Jaccard, connu également sous le nom similarité coefficient de Jaccard (À l'origine du nom Coefficient de Communauté par Paul Jaccard), est un indice statistique utilisé pour comparer la similarité et la diversité de l'ensemble des échantillons.
  Le coefficient de Jaccard est le rapport entre la cardinalité (la taille) de l'intersection des ensembles reconnus et la cardinalité de l'union des ensembles. Il permet d'évaluer la similarité entre les ensembles $\text{sim}(u,v)^j$ est donné :

$$\text{sim}(u,v)^j = \frac{|u \cap v|}{|u \cup v|} = \frac{|u \cap v|}{|u|+|v|-|u \cap v|} \quad [3.6]$$

## 4. Solution proposée :

Dans ce travail, les environnements suivants sont considérés : le nombre d'utilisateurs est N, le nombre d'items est *M*. $R = (r_{ui})$ $N \times M$ est la matrice de note, index utilisateur $u$ = 1, …, $N$, index d'item = 1, …, $M$. $r_{ui}$ désigne le score de l'évaluation de l'utilisateur u sur l'item i. $t_{ui}^{r}$ désigne le temps généré par la note $r_{ui}$.

### 4.1. Générer des clusters utilisateurs basé sur la distance d'attribut utilisateur :

Habituellement, chaque utilisateur a ses propres attributs, tels que l'âge, le sexe, le lieu d'origine, le numéro d'identification, la profession, le salaire, etc. Selon les données connexes de MovieLens, nous constatons au sein de ce travail, que les enfants préfèrent généralement l'animation et le théâtre ; Les jeunes préfèrent les drames romantiques par contre les personnes âgées préfèrent généralement la comédie. Sur l'ensemble de données MovieLens 100k, la situation d'âge de l'utilisateur est indiquée sur la figure 3.1. La situation de l'éducation des utilisateurs d'achats en ligne est également donnée [114]. Les utilisateurs d'âge, de sexe et de profession différents ont des intérêts différents. Selon les statistiques ci-dessus, nous divisons l'âge en trois catégories. Lorsque l'âge est vectorisé à trois valeurs, des utilisateurs similaires peuvent être mieux regroupés et les résultats de la prédiction sont meilleurs. Dans l'ensemble de

données de MovieLens utilisé dans ce travail, il comprend 21 types de professions. Nous séparons les professions en trois (3) grandes catégories selon le professionnalisme. Ce sont les catégories de loisirs, de culture et de gestion. Ainsi, nous analysons les caractéristiques d'attribut des utilisateurs pour calculer la distance de l'utilisateur [114].

$u_1$ Signifie la valeur caractéristique de l'âge de l'utilisateur $u$,

$u_1 =$ { 0, si l'Age de < 15,
1, if 15 ≤ l'Age de l'utilisateur ≤ 55,
2, l'Age de l'$u$utilisateur > 55.

**Attributs des utilisateurs**

| | *Age* | **Genre** | *Occupation* |
|---|---|---|---|
| $User_1$ | 57 | *Male* | **Administrateur** |
| | 2 | 0 | 2 |
| $User_2$ | 30 | ***Femelle*** | *Artist* |
| | 1 | 1 | 1 |

Tableau 3.1 : Attributs des utilisateurs

$u_2$ Signifie la valeur caractéristique du sexe de l'utilisateur $u$,

$u_2 =$ { 0, si l'utilisateur $u$ est male,
1, si l'utilisateur $u$ est femelle.

$u_3$ Signifie la valeur caractéristique de l'occupation de l'utilisateur $u$,

$u_3 =$ { 0, si l'utilisateur $u$ est en classe loisir,
1, si l'utilisateur $u$ est en classe de culture,
2, si l'utilisateur $u$ est en classe de management

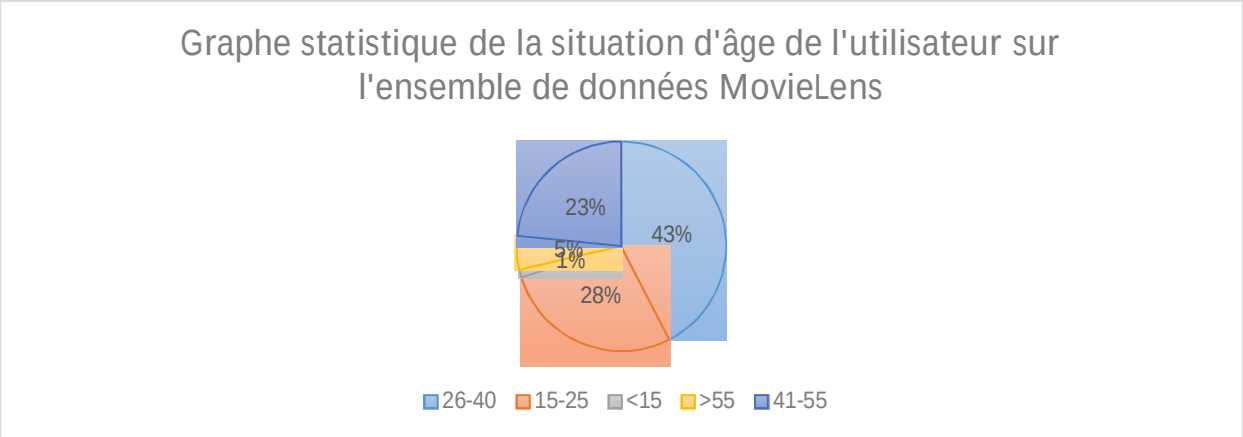


*Figure 3.1 : Graphe statistique de la situation d'âge de l'utilisateur sur l'ensemble de données MovieLens*

Par la suite, la définition de la distance utilisateur $d_{uv}$ est donnée comme suit :

$$d_{uv} = \sqrt{(u_1 - v_1)^2 + (u_2 - v_2)^2 + (u_3 - v_3)^2} \quad [3.7]$$

Ici, $\{u_1, u_2, u_3\}$ sont trois attributs de l'utilisateur u, $\{v_1, v_2, v_3\}$ sont trois attributs de l'utilisateur *v*. Un exemple est donné dans le tableau 3.1 : Ensuite, la distance entre l'utilisateur 1 et l'utilisateur 2 est $d_{12}$ ,

$$\boldsymbol{d_{12} = \sqrt{(2-1)^2 + (0-1)^2 + (2-1)^2} = \sqrt{3}} \quad [3.8]$$

La règle de clustering est générée en fonction de la distance des utilisateurs. Plus la distance utilisateur $d_{uv}$ , est petite, plus les intérêts de l'utilisateur sont proches. Pour regrouper les utilisateurs similaires dans le même cluster, nous utilisons un modèle de clustering évolutif dynamique basé sur la distance des attributs de l'utilisateur tel que :

$$\theta_u(k) = \frac{K_p}{N} \sum_{v \epsilon j_u} \sin\left(\theta_v(k) - \theta_u(k)\right) + \frac{K_n}{N} \sum_{j \notin j_u} \sin\left(\theta_v(k) - \theta_u(k)\right). \quad [3.9]$$

$$Avec\ u = 1.2 \ldots . n$$

Où :

- $\theta_u$ (k) : désigne l'état u$^{ème}$ utilisateur à l'étape d'itération k.

- $k_p$: c'est un coefficient de couplage positif, son rôle est de faire évoluer l'ensemble des utilisateurs de plus petite distance.
- $k_n$: c'est le coefficient de couplage négatif, son rôle est de faire évoluer au loin les utilisateurs ayant une plus grande distance.

L'état initial $\theta_u(0)$ est choisi au hasard et uniformément et distribué dans [0, 2 $\pi$]. $j_u$: est l'ensemble d'index $j_u = \{v \mid d_{uv} < \delta, v = 1, 2, ..., N\}$, il contient des voisins avec une distance plus petite avec l'utilisateur $u$. $\delta > 0$ indique le seuil, déterminé par la distance d'attribut de l'utilisateur. Avec cet algorithme, les états des utilisateurs changent avec le temps. Avec l'évolution du temps, les états des utilisateurs seraient stables à certaines valeurs. La structure du réseau est simulée selon l'équation différentielle [3.9]. Selon les états stables on regroupe les utilisateurs avec des états similaires dans le même cluster, sinon, ils sont rencontrés dans des clusters différents.

### 4.2. Technique de remplissage basée sur le temps de notation et le genre de film :

Les algorithmes de recommandation collaborative traditionnels n'utilisent généralement que les valeurs de score (notes) tout en ignorant l'effet du temps généré par les notes.

La caractéristique de temps de notation est importante pour prédire les notes. Le temps de notation et les genres d'items sont appliqués à la matrice de remplissage (appelée STIG), qui est décrite plus en détail. Le flux de l'algorithme de STIG est donné comme suit :

**Etape 1 :**

Construire la matrice de note utilisateur-item R. Selon l'ensemble d'apprentissage $T_r$, nous construisons une matrice bidimensionnelle. Sur le jeu de données de MovieLens-100k, une matrice de notation R 2 × 10 est présentée. Nous remplirons la note de l'utilisateur 1 sur l'item 10

$$R = \begin{array}{c} \\ User_1 \\ User_2 \end{array} \begin{array}{c} \begin{array}{cccccccccc} I_1 & I_2 & I_3 & I_4 & I_5 & I_6 & I_7 & I_8 & I_9 & I_{10} \end{array} \\ \begin{pmatrix} 5 & 3 & 4 & 3 & 3 & 0 & 4 & 1 & 5 & ? \\ 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{pmatrix} \end{array}.$$

**Etape 2 :**

Sur l'ensemble de test $T_e$, trouvez le temps généré par la note $t_{ui}^{r}$ de l'item cible. Le temps de score de l'item cible 10 évalué par l'utilisateur 1 est 875, 693, 118.

**Etape 3 :**

Construire une nouvelle matrice de notation $R_1$. Tout d'abord, dans $T_r$, trouver le temps de score avant le temps généré par le score de l'item cible. En fonction de ces temps les utilisateurs correspondants, les items, les notations, construisent une nouvelle matrice de score $R_1$.

Dans $R_1$, une note non nulle, signifie que le temps de note est avant les $t_{ui}^{r}$de l'item cible. À propos de l'exemple, dans $T_r$, recherchez le temps généré par le score avant 875 693 118, et les utilisateurs, items et notes correspondants sont utilisés pour trouver la matrice de notation $R_1$ à partir de ces données de temps trouvées.

**Etape 4 :**

Trouvez les attributs caractéristiques de l'item cible. Selon la feuille de film existante, la matrice d'attributs est entrée. 1 indique que le film appartient à ce genre et 0 indique que ce n'est pas le cas. Les films peuvent être de plusieurs genres à la fois. Par exemple, une matrice d'attributs caractéristiques d'items comprenant 6 genres est donnée comme suit. L'item 1 appartient au genre des enfants, et ce n'est pas du genre aventure.

**Etape 5 :**

Dans $R_1$, trouvez différents genres de matrice de score. L'item 10 est dans 2 genres à la fois (Drama et War). Recherchez respectivement les items de genre drama et les items de genre de war dans la matrice d'attributs caractéristique des items. Ensuite, dans $R_1$, sélectionnez la matrice de notation de genre drama (notée $R_2$) et Matrice de notation de genre War (notée $R_3$).

**Etape 6 :**

Remplissez la matrice de score originale de l'utilisateur-item $R \xrightarrow{STIG} Q$. Dans différents genres de matrices, le score moyen d'un utilisateur cible est différent, et sa valeur moyenne est utilisée comme score de prédiction pour remplir la matrice de notation utilisateur-item. Ainsi, la rareté de la matrice est réduite. La matrice de score remplie est notée Q. Dans $R_2$, le score moyen de l'utilisateur 1 est calculé comme 4,2258.

Dans $R_3$, le score moyen de l'utilisateur 1 est calculé comme 4,1667.

$$\text{Score-Moyen} = \frac{le\ score\ total}{nombre\ d'item\ notés}$$

Ainsi, le score prédictif de l'utilisateur 1 pour l'item 10 est calculé comme suit :

$$(4{,}2258 + 4{,}1667) \div 2 = 4{,}19625.$$

Afin de réduire l'erreur, le résultat n'est pas entier. Q est donné comme suit. Le nombre en gras est rempli par STIG :

$$\begin{array}{c} \\ User_1 \\ User_2 \end{array} \begin{array}{c} \begin{array}{cccccccccc} I_1 & I_2 & I_3 & I_4 & I_5 & I_6 & I_7 & I_8 & I_9 & I_{10} \end{array} \\ \begin{pmatrix} 5 & 3 & 4 & 3 & 3 & 0 & 4 & 1 & 5 & \mathbf{4.19625} \\ 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{pmatrix} \end{array}$$

### 4.3. Recommandation :

Liji et al [111], utilisent une similarité de cosinus améliorée (notée ICOS) basée sur une matrice de score remplie $Q = (q_{u,i})\ N \times M$. La formule de cette mesure est définie comme suit :

$$sim_u(a,u) = \frac{\sum_{i=1}^{M} q_{u,i}\, q_{v,i}}{\sqrt{\sum_{i=1}^{M} q_{u,i}^2}\ \sqrt{\sum_{i=1}^{M} q_{v,i}^2}} \cdot \sqrt{\frac{\sum_{i \in I_{uv}} q_{u,i}}{\sum_{i \in I_u} q_{u,i}}} \cdot \sqrt{\frac{\sum_{i \in I_{uv}} q_{v,i}}{\sum_{i \in I_u} q_{v,i}}} \quad [3.10]$$

Dans ce travail, nous proposons une mesure de similarité qui combine le coefficient de corrélation de Pearson avec le coefficient de Jaccard.

$$sim_u(a,u) = \frac{\sum_{i \in I}(\mathrm{r}_{a,i} - \overline{\mathrm{R}_a})(\mathrm{r}_{u,i} - \overline{\mathrm{R}_u})}{\sqrt{\sum_{i \in I}(\mathrm{r}_{a,i} - \overline{\mathrm{R}_a})^2}\ \sqrt{\sum_{i \in I}(\mathrm{r}_{u,i} - \overline{\mathrm{R}_u})^2}} \quad [3.11]$$

Où $sim_u(a,u)$est la similarité entre les utilisateurs $a$ et u, I est l'ensemble des items notés par les deux utilisateurs $a$ et $u$, $\mathrm{r}_{a,i}$ est la note de l'utilisateur $a$ sur item i,$\overline{\mathrm{R}_a}$ Est la note moyenne de l'utilisateur $a$ pour les I items, la similarité dans [3 ,11] ne prend en compte que la coévaluation utilisateurs. Ce problème peut avoir une influence lorsque les items reçoivent un petit nombre de notes. Par exemple, une mesure de similarité de 0,93 calculée entre deux films qui n'ont été évalués que par 7 utilisateurs communs n'est pas aussi fiable qu'une mesure de similarité de 0,83 obtenu auprès de 110 utilisateurs communs. Comme solution à cette question, la similarité de Jaccard est utilisée pour calculer le nombre de films qui sont évalués ainsi que

pour calculer le rapport entre le nombre de films évalués par les utilisateurs a et u et le nombre de films évalués individuellement par a et u.

$$sim_{ujaccer}(a,u) = \frac{U_a \cap U_u}{U_a \cup U_u} \quad [3.12]$$

Par conséquent, les valeurs de similarité finales prendront la similarité de Jaccard comme un schéma de pondération pour avoir la valeur de similarité finale améliorée.

$$sim_{user}(a,u) = sim_u(a,u) * sim_{ujaccer}(a,u) \quad [3.13]$$

Après avoir calculé les valeurs de similarité, les systèmes de recommandation trouvent des valeurs d'estimation pour tous les items non vus par l'utilisateur actif. Pour les recommandations, les items les plus similaires au N-top sont affichés à l'utilisateur. La formule de prédiction la plus populaire pour un système de recommandation basé sur l'utilisateur est :

$$p_{rat_user_{u,i}} = \overline{R_u} + \frac{\sum_{k=1}^{k} sim_{user}(u,k) \times (R_{k,I} - \overline{R_K})}{\sum_{k=1}^{k}(|sim_{user}(u,k)|)} \quad [3.14]$$

Où $p_{rat_user_{u,i}}$ Est la note prédite de l'utilisateur u sur l'item i, K est l'ensemble des voisins de l'utilisateur actif qui sont les plus similaires à lui.

### 4.4. L'algorithme :

La recommandation personnelle améliorée basée sur le regroupement des attributs utilisateur et le remplissage de matrice utilisateur-item est décrite plus en détail dans l'algorithme1. Premièrement, regrouper les utilisateurs qui possède des intérêts similaires dans le même cluster, puis calculer la matrice de similarité et trouver les voisins les plus proches dans le même cluster pour réduire la portée de la recherche de voisins, ce qui peut améliorer la précision de la recommandation. Enfin, des scores prédictifs sont générés et font des recommandations. Le flux de l'algorithme est donné comme suit.

**Étape 1 :**

Clustering d'utilisateurs. Le principe de clustering est déterminé en fonction de la distance d'attribut utilisateur. Et les utilisateurs sont divisés en différents clusters en mettant à jour les états des utilisateurs en utilisant [3.9]

**Étape 2 :**

Une matrice de score remplie est générée. En raison de sa rareté, la matrice de partition originale est d'abord remplie en fonction du temps généré par la partition et des genres d'items.

**Étape 3 :**

Calculez la matrice de similarité entre les utilisateurs d'un même cluster. La matrice de similarité des différents clusters est de dimension différente. Nous calculons la similarité entre les utilisateurs en fonction de la matrice remplie et de [3.13]

**Étape 4 :**

Prédisez les évaluations en fonction du filtrage collaboratif des utilisateurs. Les scores des plus proches voisins sont importants pour prédire les notations. Les utilisateurs ayant une grande similarité sont choisis comme voisins en fonction du nombre de voisins. Ensuite, le score de prédiction est calculé par [3.14]

**Étape 5 :**

Recommandez les N premiers items comme liste de recommandations à l'utilisateur cible. En comparant les scores prédits, les N scores les plus élevés des items sont sélectionnés comme liste de recommandations pour l'utilisateur cible.

***Algorithme 1***

***Exiger :*** *matrice d'adjacence de distance d'attribut utilisateur,* $k_p > 0$,

$k_n < 0, \tau = l$

***Assurer :*** *résultats recommandés*

*// Regroupement d'utilisateurs*

***Pour*** *chaque utilisateur u* ***Faire***

*Mettre à jour les états des utilisateurs selon EQ* [3.9]

***Fin Pour***

***Si*** $|\theta_u(r) - \theta_v(t)| < \tau$ ***Alors***

*Les utilisateurs* ***u*** *et v ; sont divisés dans le même cluster*

***Fin Si***

*// Remplissez la matrice de score utilisateur item basée sur* ***STIG***

*// Calculer la matrice de* similarité *dans le même cluster*

***Pour chaque*** *utilisateur dans son cluster* ***Faire***

*Calculer la* similarité *en fonction de la matrice remplie et d'EQ* [3.13]

***Fin Pour***
*// Prédire les évaluations basées sur le FC des utilisateurs*
***Pour*** *chaque utilisateur cible* $u \in t_e$ ***Faire***
*Trouver les k voisins les plus proches dans le même cluster*
*Prédire les scores en utilisant la formule* [3.14].
***Fin Pour***
*// Recommander les N premiers items en tant que liste de Recommandations à l'utilisateur cible*

## 5. Dataset :

Nous utilisons pour l'expérimentation, le jeu de données réelles du système de recommandation de films MovieLens[1]. Ce jeu de données est très populaire et utilisé dans beaucoup d'études du domaine de filtrage collaboratif. Le site Web MovieLens est développé par le groupe de recherche GroupLens à l'université de Minnesota, États-Unis. On dispose sur ce site de deux jeux de données d'évaluations de films de tailles différentes.

Nous utilisons dans cette expérimentation le jeu contenant 100 000 évaluations de 1 à 5 étoiles, fournies par 943 utilisateurs sur 1682 films pendant une période allant de septembre 1997 à avril 1998. Ainsi, il existe 93,7% de données manquantes (matrice creuse). La distribution de ces évaluations est présentée dans la figure ci-dessous :

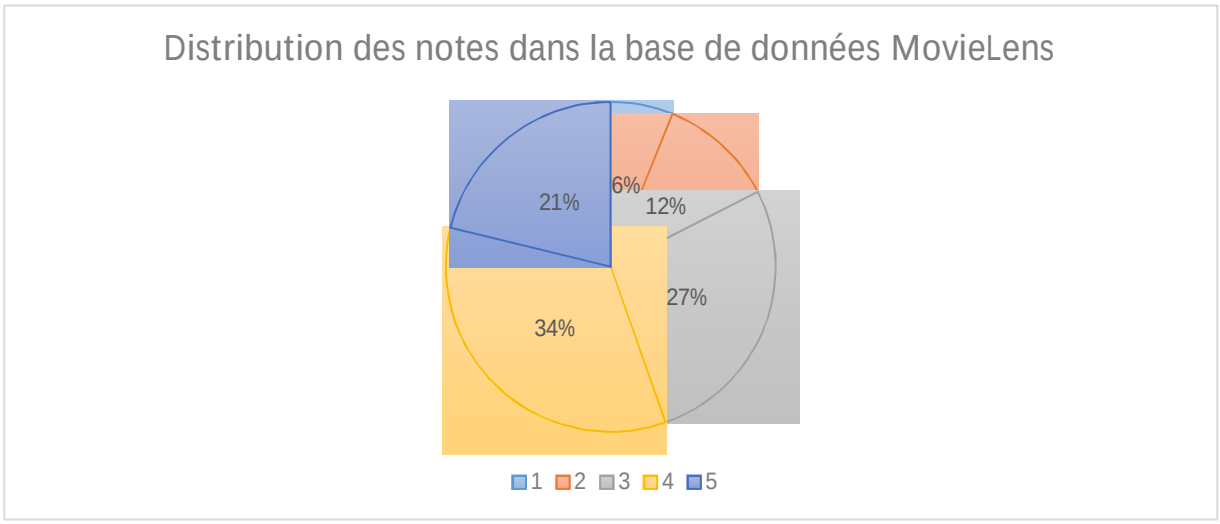


*Figure 3.2 Distribution des notes dans la base de données MovieLens*

À partir de l'ensemble de données initial, cinq divisions distinctes pour l'apprentissage ainsi des données de test ont été générées ($u_1$ .base, $u_2$.base, $u_3$ .base, $u_4$ .base, $u_5$ .base et $u_1$ .test, $u_2$.test, $u_3$.test, $u_4$.test, $u_5$.test). Pour chaque division de données, 80% de l'ensemble original a été inclus dans l'apprentissage et 20% d'entre eux ont été inclus dans les données de test. Les ensembles de tests dans tous les cas étaient disjoints.

## 6. Métrique d'évaluation :

Parmi les mesures qui évaluent la qualité et l'efficacité des systèmes de recherche d'information et de recommandation, il existe :

### 6.1. Précision :

La précision est la proportion des items pertinents dans la liste des articles retournés par le système, ce qui correspond au pourcentage ou au nombre d'items suggérés et s'avérant véritablement pertinent pour l'utilisateur actif. Si l'on considère par exemple, une liste des N. meilleures recommandations, la précision correspondra à la proportion d'items véritablement consultés et appréciés par l'utilisateur courant.

$$précision = \frac{\left|\{\text{Recommandation pertinentes}\} \cap \{\text{Recommandation émises}\}\right|}{\left|\{\text{Recommandation émises}\}\right|} \quad [3.15]$$

### 6.2. Rappel :

Le rappel est la proportion des articles pertinents par rapport à ceux qui sont pertinents au sein du corpus. Il compte le nombre d'articles dans la liste de recommandations n-top ayant déjà été notés par l'utilisateur courant.

$$Rappel = \frac{\left|\{\text{Recommandation pertinentes}\} \cap \{\text{Recommandation émises}\}\right|}{\left|\{\text{Recommandation pertinentes}\}\right|} \quad [3.16]$$

### 6.3. F-Mesure :

C'est la mesure qui combine les deux précédentes :

$$F = 2.\frac{précision\ .rappel}{(\ précision\ + rappel\ )} \quad [3.17]$$

**6.4. <u>Erreur Absolue Moyenne (Mean Absolute Error MAE)</u> :** mesure la moyenne des écarts entre une note prédit et son correspondant dans Rtest. L'erreur absolue moyenne (formule [3.18]) a été utilisée pour évaluer plusieurs systèmes de recommandation [Sarwar et al., 2001; Mobasher et al., 2004; Resnick et al., 1994] et est encore populaire de nos jours.

$$MAE = \frac{\sum_{(u,i)\in Testset} |r_{u,i} - pr_{u,i}|}{|Testset|} \quad [3.18]$$

**6.5. <u>Racine de l'Erreur Quadratique Moyenne : Root Mean Squared Error (RMSE)</u> :** (formule[3.19]) est devenue la mesure de référence surtout après son utilisation pour mesurer la précision dans le challenge Netflix [Bennett and Lanning, 2007] pour la recommandation de films. Comparée à la MAE, la RMSE pénalise les écarts importants, ainsi un écart de 2 points aura pour effet d'augmenter la somme des erreurs par 2 au niveau de la MAE, alors qu'il augmentera la somme par 4 au niveau de la RMSE.

$$RMSE = \sqrt{\frac{\sum_{(u,i)\in TeseSet}(r_{u,i} - pr_{u,i})^2}{|TestSet|}} \quad [3.19]$$

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

## 7. <u>Outils utilisés</u> :

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

Les outils que nous avons utilisés pour le développement sont :

- Python version 3.7 comme des environnements de développement.
- Jeu de données : ml-100k

### 7.1. <u>Python</u> :

Python est un langage informatique de haut niveau, portable, dynamique, extensible, gratuit, structuré et open source conçu pour être orienté objet. Il est multi-paradigme et multi-usage. Développé à l'origine par Guido Van Rossum en 1993, Il est actuellement le langage le plus utilisé au monde. Comme il s'agit d'un langage de haut niveau, il est donc plus facile. Mais reste un langage très puissant.

### 7.2. Caractéristiques du langage :

- Python convient aussi bien à des scripts d'une dizaine de lignes qu'à des projets complexes de plusieurs dizaines de milliers de lignes.
- Python est (optionnellement) multi-threadé.
- Python est portable, non seulement sur les différentes variantes d'UNIX, mais aussi sur les OS propriétaires : MacOS, BEOS, NeXTStep, MS-DOS et les différentes variantes de Window.
- Python est gratuit, mais on peut l'utiliser sans restriction dans des projets commerciaux
- Python gère ses ressources (mémoire, descripteurs de fichiers...) sans intervention du programmeur par un mécanisme de comptage de références (proche, mais différent d'un garbage collector).
- Python est un langage orienté-objet qui supporte l'héritage multiple et la surcharge des opérateurs.
- Python est dynamique (l'interpréteur peut évaluer des chaines de caractères représentant des expressions ou des instructions Python) orthogonal (un petit nombre de concepts suffit à engendrer des constructions très riches), réflectif (il supporte la méta programmation par exemple la capacité pour un objet de se rajouter ou de s'enlever des attributs ou des méthodes ou même de changer de classe en cours d'exécution) et introspectif (un grand nombre d'outils de développement comme le débugger ou le profiler sont implantés en Python lui-même)

## 8. Présentation de l'application :

Après avoir justifié nos choix d'outils de développement, nous allons présenter quelques prises d'écrans de notre application.

- **Lire Data :**

Dans un premier temps, on va charger l'ensemble de données de MovieLens (info users)

***l'ajout du data***

```
In [29]: data = pd.read_csv('user.csv' , delimiter=';')
         data
```

Out[29]:

| | id_user | age | sex | ocu | tim |
|---|---|---|---|---|---|
| **0** | 1 | 24 | M | technician | 85711 |
| **1** | 2 | 53 | F | other | 94043 |
| **2** | 3 | 23 | M | writer | 32067 |
| **3** | 4 | 24 | M | technician | 43537 |
| **4** | 5 | 33 | F | other | 15213 |
| **...** | ... | ... | ... | ... | ... |
| **938** | 939 | 26 | F | student | 33319 |

*Figure 3.3 : l'ajout du data*

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

**2-organisation de la colonne age**

```
In [34]: #0 if the age of  < 15
         #1 if 15 ≤ the age of u ≤ 55
         #2 if the age of u > 55.
         bins = [0,15,55,100]
         age =[0,1,2]
         data_user['age']=pd.cut(data_user["age"],bins,labels=age)
         data_user.head()
```

*Figure 3.4 : code de préparation de la colonne âge*

- **Code de préparation de la colonne sex**

```
In [12]: data_user['sex'].replace('M', 0 ,inplace=True)
         data_user['sex'].replace('F', 1 ,inplace=True)
         data_user.head()
```

*Figure 3.5 : Code de préparation de la colonne sex*

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

```
#occupation in 3 class: 2(management) 1(culture) 0(loisir)
ocu = {'technician': 2,'other': 1 , 'writer':1
       , 'executive': 2,'administrator': 2 ,
       'student':1 , 'lawyer': 1,'educator': 1 ,
       'scientist':1 , 'entertainment': 0,'programmer': 1 ,
       'librarian':2 , 'homemaker': 2,'artist': 1 ,
       'engineer':2 , 'marketing': 2,'none': 0 , 'healthcare':2 ,
       'retired': 0,'salesman': 2 , 'doctor':2
      }
data_user.ocu = [ocu[item] for item in data_user.ocu]
data_user
```

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

- **Résultat final de la préparation du data**

Out[13]:

| | id_user | age | sex | ocu |
|---|---|---|---|---|
| 0 | 1 | 1 | 0 | 2 |
| 1 | 2 | 1 | 1 | 1 |
| 2 | 3 | 1 | 0 | 1 |
| 3 | 4 | 1 | 0 | 2 |
| 4 | 5 | 1 | 1 | 1 |
| ... | ... | ... | ... | ... |

*Figure 3.7 : Résultat final de la préparation du data*

## 9. Résultats et analyses :

### 9.1. Comparaison des résultats de la matrice remplie :

La matrice Q remplie est utilisée pour calculer la similarité entre les utilisateurs. La performance de la matrice remplie est comparée à celle de la matrice d'évaluation d'origine. Les résultats de comparaison sont présentés au sein des figures 3.8, 3.9.

À partir des graphes, on constate que la matrice remplie est meilleure que la matrice d'évaluation d'origine. La matrice de pointage de remplissage est significative pour obtenir de meilleurs résultats de prédiction. Parce que la matrice remplie résout le problème de la rareté des données, et elle peut améliorer la similarité entre les utilisateurs.

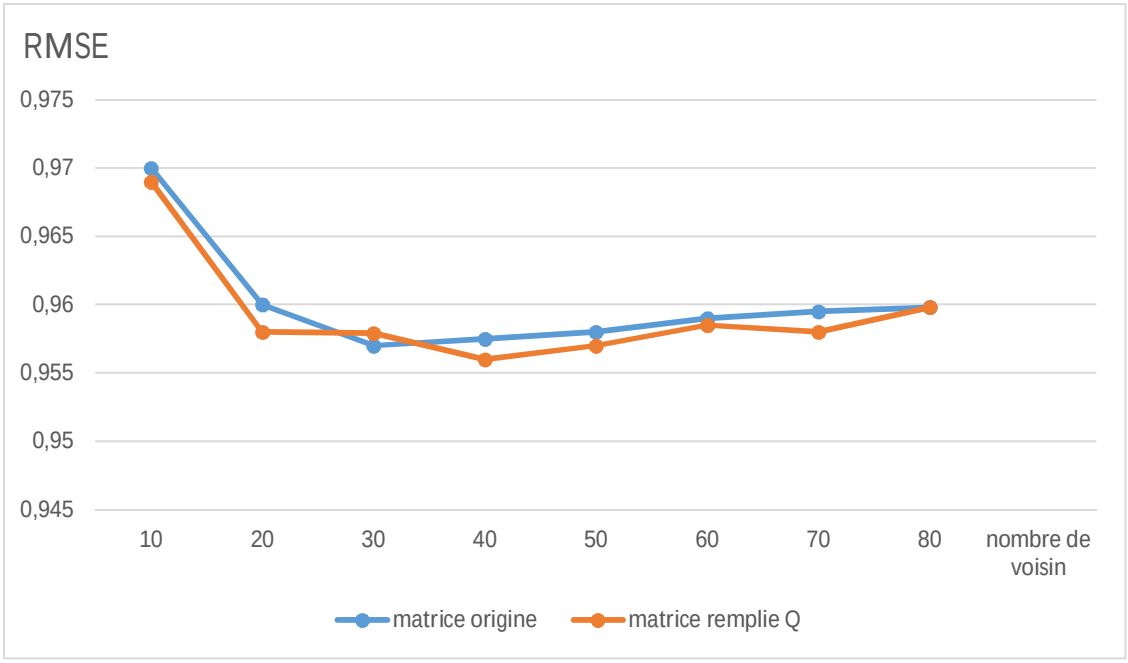


*Figure 3.8 : Comparaison des performances entre la matrice remplie et la matrice de notation d'origine par la mesure RMSE*

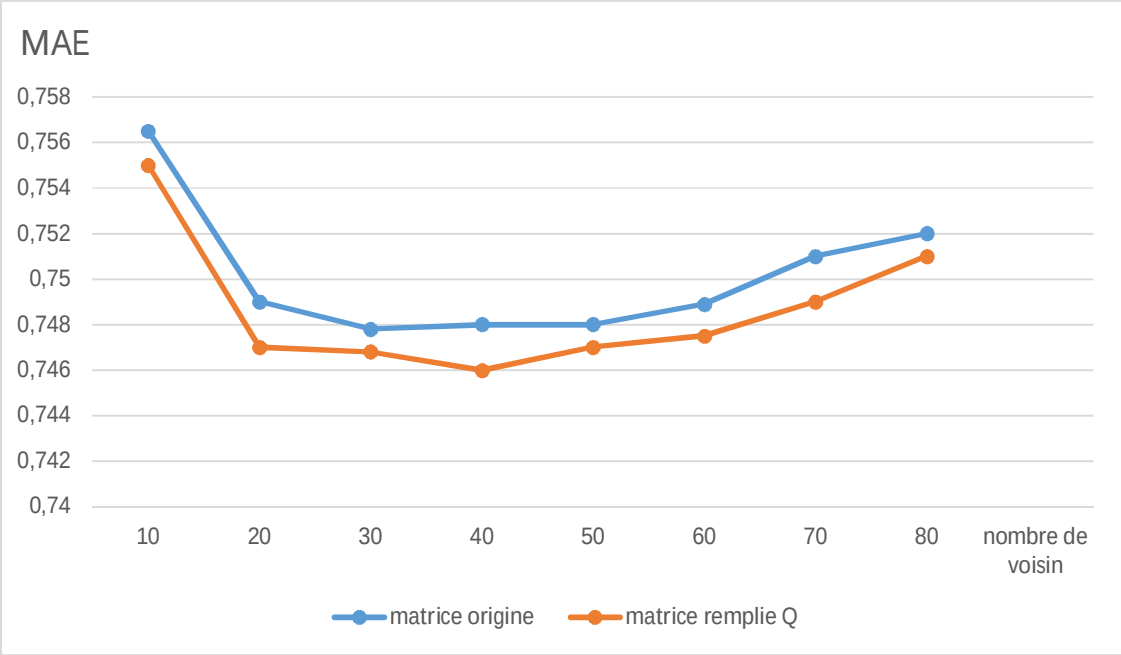


*Figure 3.9 : Comparaison des performances entre la matrice remplie et la matrice de notation d'origine par la mesure MAE*

### 9.2. <u>Comparaison des résultats d'un indice de similarité amélioré</u> :

Nous comparons notre similarité ajustée avec d'autres mesures de similarités sur l'ensemble de données ML-100k au sein des figures 3.10, 3.11. D'après les résultats obtenus, on s'aperçoit que les valeurs RMSE et MAE de la similarité ajusté Sim ($sim_{user}$) sont plus petites que celles des similarités COS, ICOS PCC et ACOS (utilisé dans l'algorithme 1). Quel que soit le nombre de voisins.

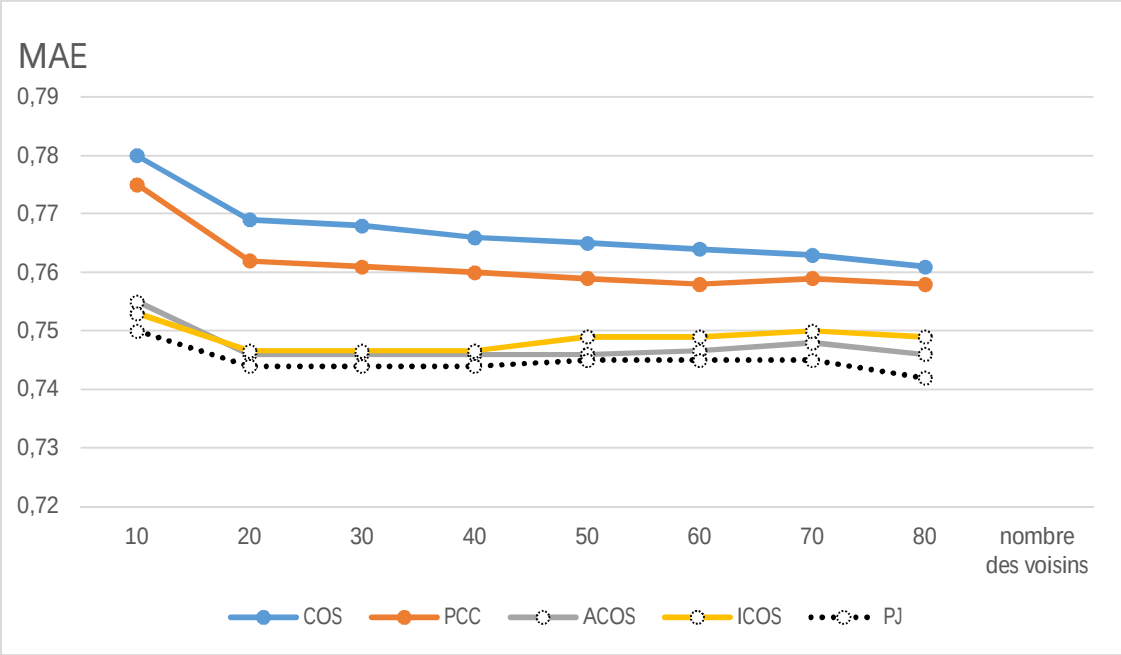


*Figure 3.10 : Comparaison entre les performances de différentes similitudes par la mesure MAE*

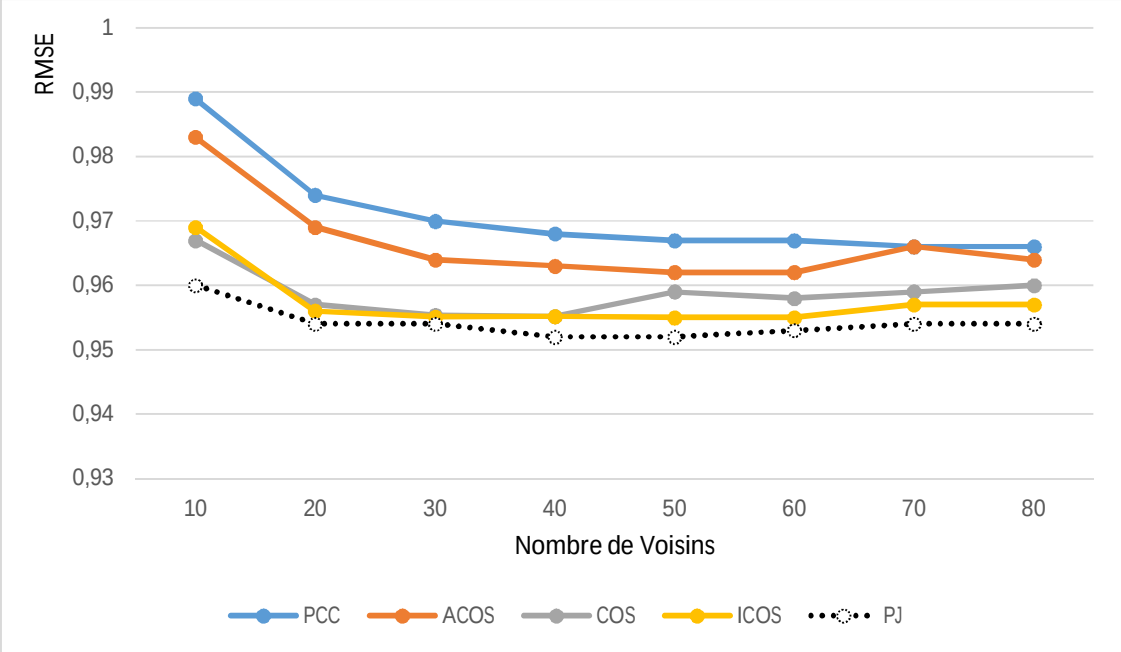


*Figure 3.11 : Comparaison entre les performances de différentes similitudes par la mesure RMSE*

### 9.3. Comparaison des résultats du regroupement dynamique des utilisateurs évolutifs :

Dans ce travail, nous comparons l'algorithme utilisé avec trois algorithmes existants (PR-ACMF), (DTD) [115], ($EHC$-$CF$.) [116]. Ces algorithmes sont récemment proposés et ont obtenu de bonnes performances. Les résultats de comparaison sont présentés au sein des figures 3.12, 3.13.

On remarque que MAE et RMSE de l'algorithme utilisé sont inférieures à (PR-ACMF), (DTD), ($EHC$-$CF$). En outre, l'algorithme (PR-ACMF) est déjà comparé à l'algorithme de clustering K-means classique. Le numéro de cluster K-means est attribué à trois cas et les utilisateurs du réseau sont divisés en 3, 8 et 15 clusters. Les résultats de comparaison sont présentés sur les figures 3.14, 3.15.

On constate que notre méthode proposée est plus efficace que l'algorithme K-means.

En effet, notre méthode ne nécessite pas de définir le nombre de clusters au préalable. Notre modèle de clustering évolutif dynamique peut mieux regrouper les utilisateurs.

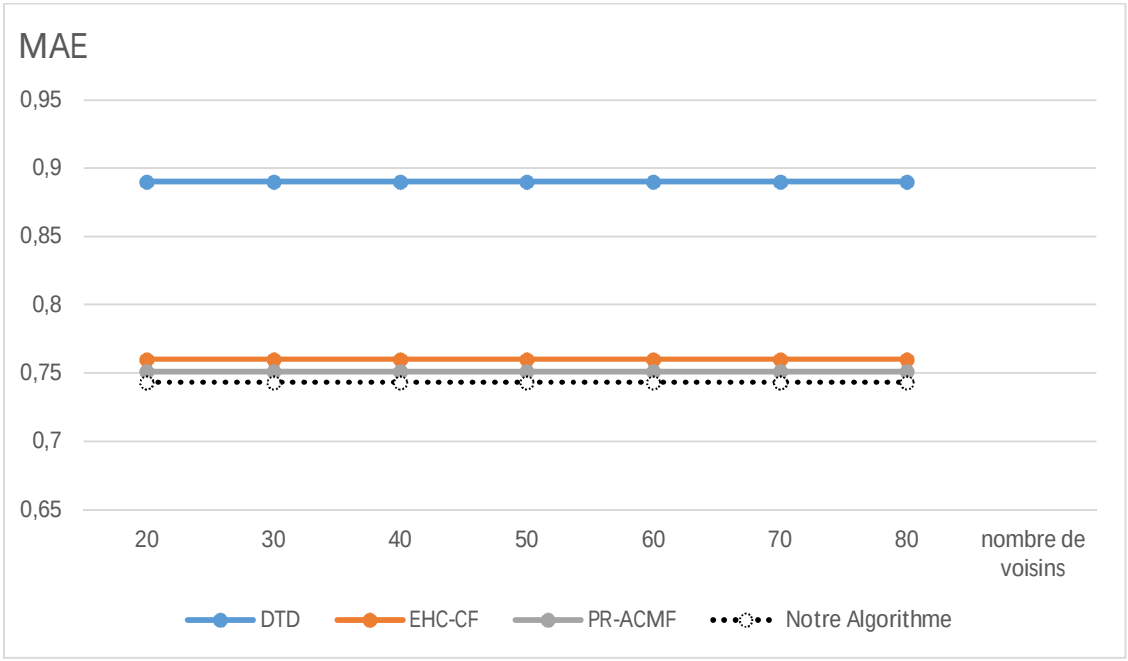


*Figure 3.12 : Comparaison entre l'algorithme réalisé et des algorithmes existants par la mesure MAE*

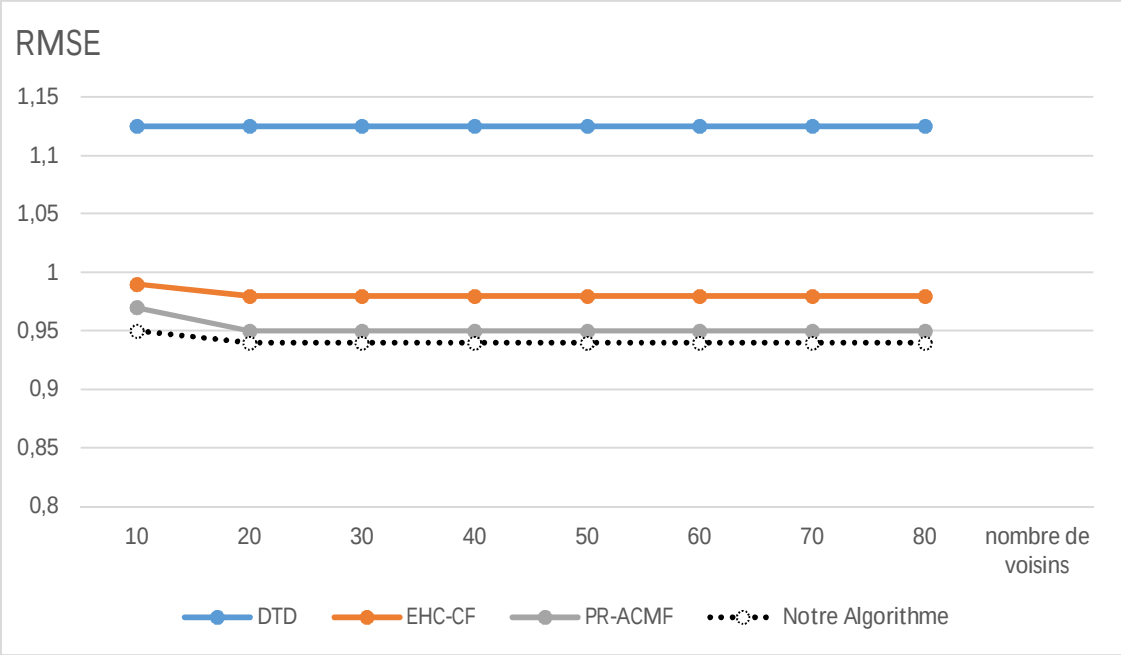


*Figure 3.13 : Comparaison entre l'algorithme réalisé et des algorithmes existants par la mesure RMSE*

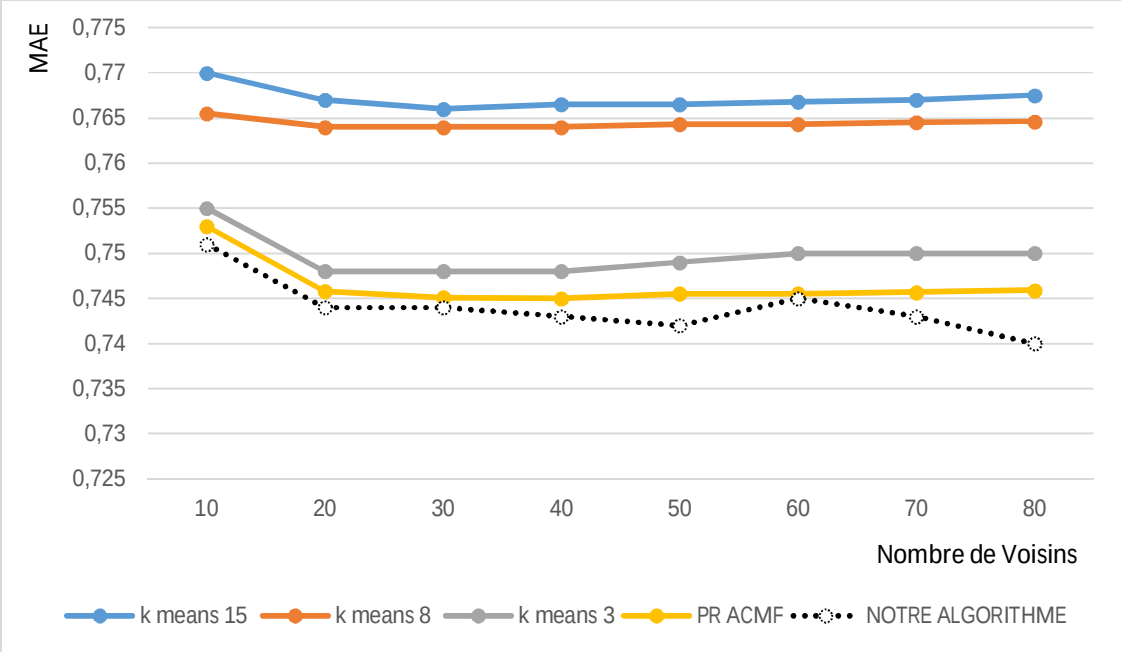


*Figure 3.14: Comparaison entre l'algorithme réalisé et l'algorithme k-means et PR-ACMF par la mesure MAE*

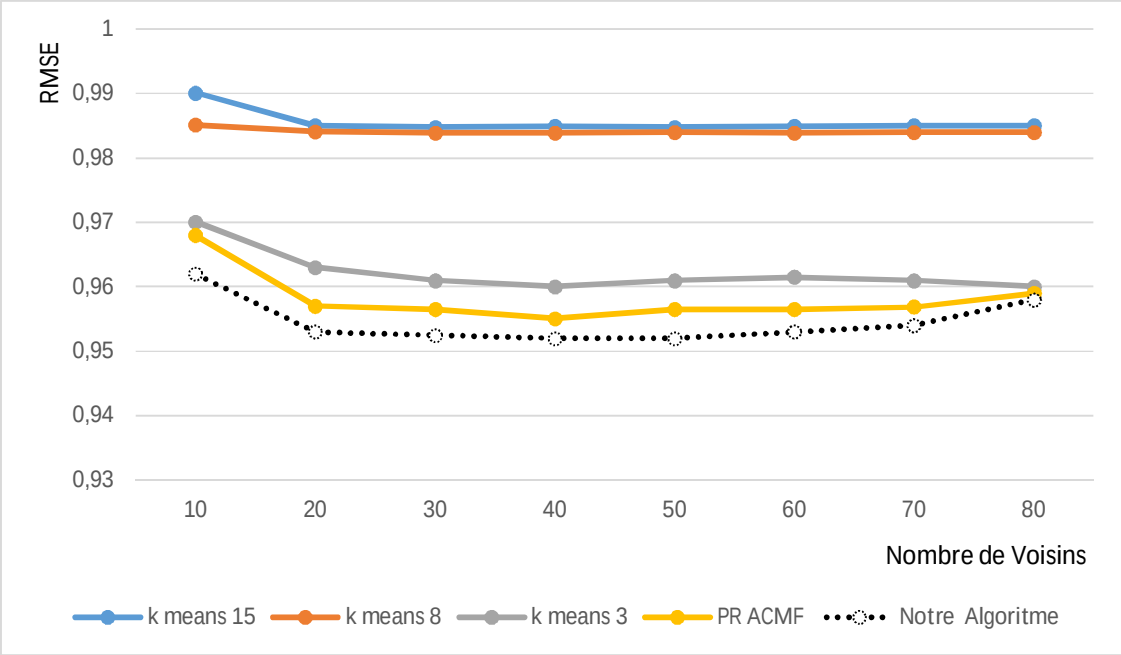


*Figure 3.15 : Comparaison entre l'algorithme réalisé et l'algorithme k-means et PR-ACMF par la mesure RMSE*

## 10. Conclusion :

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

# *Conclusion Générale*

# *Conclusion générale*

Les systèmes de recommandation (SR) sont nés de la volonté de pallier au problème de surcharge d'information sur le Web, combinant ainsi des techniques de filtrage d'information, personnalisation, intelligence artificielle, réseaux sociaux et interaction personne-machine, les SR fournissent à des utilisateurs des suggestions qui répondent à leurs besoins et préférences informationnelles. En effet, les systèmes de recommandation sont particulièrement sollicités dans les applications de commerce électronique. Le filtrage collaboratif qui constitue une des solutions techniques implémentant l'intelligence collective est déjà utilisé avec succès dans plusieurs SR.

La technologie de clustering peut améliorer l'évolutivité du filtrage collaboratif en réduisant l'espace de recherche du voisin le plus proche. Différent de la plupart des méthodes existantes, ce travail utilise un nouveau modèle de clustering évolutif dynamique basé sur la distance des attributs de l'utilisateur. Les valeurs d'état initiales des utilisateurs sont générées de manière aléatoire. Selon le modèle de réseau utilisé, les états des utilisateurs évoluent avec le temps et sont regroupés en différents clusters. La mesure de similarité est importante pour trouver les utilisateurs voisins. La proportion des notes des utilisateurs pour les items de notation courants est prise en compte. Des données manquantes peuvent avoir une influence sur la recommandation. Nous remplissons la matrice de notation basée sur le temps de notation et les genres d'items pour améliorer la précision du calcul de similarité. Nous avons proposé d'utiliser une mesure de similarité qui combine le coefficient de jaccard avec le coefficient de corrélation de Pearson afin de calculer la similarité entre les utilisateurs dans le cluster (pas pour tous les utilisateurs du système). En faisant une comparaison avec les trois algorithmes existants, notre méthode est plus efficace avec des performances de prédiction et de recommandation élevées. Aussi, il existe certaines limitations potentielles de la méthode proposée par Liji [111]. Lorsque le nombre de voisins est inférieur à 40, les valeurs RMSE et MAE de la similarité ICOS ne sont pas meilleures que la similarité de COS et notre mesure de similarité proposée. Nous avons utilisé la méthode de clustering d'utilisateurs sur dataset MoviLens-100k. Nous devrions envisager une méthode de clustering qui s'applique à des ensembles de données plus volumineux.

# *Bibliographie*

# *et*

# *Référence*

# *Bibliographie et Référence*


1- J. Bobadilla , F. Ortega , A. Hernando , Guti , A. Rrez , Recommender systems survey, Knowl. Based Syst. 46 (1) (2013) 109–132 .

2- D. Jannach , M. Zanker , A. Felfernig , G. Friedrich , Recommender Systems : An Intro- duction, Cambridge University Press, New York, NY, USA, 2010 .

3- G. Guo , J. Zhang , N. Yorke-Smith , Leveraging multiviews of trust and similarity to enhance clustering-based recommender systems, Knowl. Based Syst. 74 (1) (2015)

4- L.H. Ungar , D.P. Foster , Clustering methods for collaborative filtering, Aaai Work- shop Recommendation Syst. (1998) 114–129 .

5- M. Ramezani , P. Moradi , F. Akhlaghian , A pattern mining approach to enhance the accuracy of collaborative filtering in sparse data domains, Phys. A 408 (32) (2014) 72–84 .

6- - T. Liang , J. Fan , J. Zhao , Y. Liang , Y. Li , Improved slope one collaborative filtering predictor using fuzzy clustering, Adv. Data Mining Appl. 8346 (2013) 181–192 .

7- 50+ Netflix statistics and facts stats that define the company's dominance [2020 version]

8- [Burke,2002]Burke,R.(2002).Hybridrecommendersystems:Surveyandexperiments.*Usermodelingandu ser-adaptedinteraction*,12(4):331–370.

9- Idir Benouaret. Un système de recommandation contextuel et composite pour la visite personnalisée de sites culturels.

10- Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible ex-tensions. IEEE transactions on knowledge and data engineering, 17(6), (pp. 734-749).

11- Yang, X., Guo, Y., Liu, Y., & Steck, H. (2014). A survey of collaborative fil-tering based social recommender systems. Computer Communications, 41, (pp. 1-10).

12- D. Goldberg, D. Nichols, B. M. Oki, & D. Terry. (1992). "Using Collabora-tive Filtering to Weave an Information Tapestry," Communications of the ACM, vol. 35, no 12, (pp. 61-70).

13- Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, J. (1994, Oc-tober). GroupLens: an open architecture for collaborative filtering of netnews. In Proceedings of the 1994 ACM conference on Computer sup-ported cooperative work (pp. 175-186). ACM.

14- Shardanand, U., & Maes, P. (1995, May). Social information filtering: al-gorithms for automating "word of mouth". In Proceedings of the SIGCHI conference on Human factors in computing systems (pp. 210-217). ACM Press/Addison-Wesley Publishing Co.

15- Ricci, F., Rokach, L. & Shapira, B. (2011). Introduction to Recommender Systems Handbook. In F. Ricci, L. Rokach, B. Shapira, P. B. Kantor, F. Ricci, L. Rokach, B. Shapira & P. B. Kantor (ed.),Recommender Systems Handbook (pp. 1--35). Springer. (ISBN: 978-0-387-85819-7.)

16- Brahim, H. F. F. B. O. (2012). A comparison study of some algorithms in Recommender Systems. , 130-135.

17- Sheth, B. & Maes, P. (1993). Evolving agents for personalized information filtering. In (pp. 345--352).

18- Montaner, M., López, B. & De La Rosa, J. L. (2003). A Taxonomy of Recommender Agents on theInternet. Artif. Intell. Rev. 19 (4), 285--330. (Doi: 10.1023/A:1022850703159.)

19- Park, D. H., Kim, H. K., Choi, I. Y. & Kim, J. K. (2012). A literature review and classification of recommender systems research. Expert Systems with Applications 39 (11), 10059 - 10072. (Doi: 10.1016/j.eswa.2012.02.038.)

20- [Shahabi et al., 2001] Shahabi, C., Banaei-Kashani, F., Chen, Y.-S., and McLeod, D. (2001). Yoda: An accurate and scalable web-based recommendation system. In Cooperative Information Systems, pages 418–432. Springer.

21- [Salton and Buckley, 1988] Salton, G. and Buckley, C. (1988). Term weighting approaches in automatic text retrieval. In INFORMATION PROCESSING AND MANAGEMENT pp. 513–523,

22- Jöran Beel, Bela Gipp, Stefan Langer et Corinna Breitinger: Research-paper recommender systems: a literature survey. Int. J. on Digital Libraries, 17(4):305–338, 2016. URL https: //doi.org/10.1007/s00799-015-0156-0.

23- Ricardo A. Baeza-Yates et Berthier A. Ribeiro-Neto: Modern Information Retrieval. ACM Press / Addison-Wesley, 1999.

24- Gerard Salton: Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer. Addison-Wesley, 1989.

25- Henry Lieberman: Letizia: An agent that assists web browsing. In Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, IJCAI 95, Montréal Québec, Canada, August 20-25 1995, 2 Volumes, pages 924–929. Morgan Kaufmann, 1995.

26- Michael J. Pazzani, Jack Muramatsu et Daniel Billsus: Syskill & webert : Identifying interesting web sites. In William J. Clancey et Daniel S. Weld, éditeurs: Proceedings of the Thirteenth National Conference on Artificial Intelligence and Eighth Innovative Applications of Artificial Intelligence Conference, AAAI 96, IAAI 96, Portland, Oregon, August 4-8, 1996, Volume 1., pages 54–61. AAAI Press / The MIT Press, 1996. URL http://www.aaai.org/Library/ AAAI/1996/aaai96-008.php.

27- Alexandros Moukas: Amalthaea information discovery and filtering using a multiagent evolving ecosystem. Applied Artificial Intelligence, 11(5) :437–457, 1997. URL https://doi.org/ 10.1080/088395197118127.

28- Liren Chen et Katia P. Sycara: Webmate: A personal agent for browsing and searching. In Katia P. Sycara et Michael Wooldridge, éditeurs: Proceedings of the Second International Conference on

Autonomous Agents, AGENTS 1998, St. Paul, Minnepolis, USA, May 9-13, 1998, pages 132–139. ACM, 1998.

29- Daniel Billsus et Michael J. Pazzani : A personal news agent that talks, learns and explains. In Oren Etzioni, Jörg P. Muller ¨ et Jeffrey M. Bradshaw, éditeurs : Proceedings of the Third Annual Conference on Autonomous Agents, AGENTS 1999, Seattle, WA, USA, May 1-5, 1999, pages 268–275. ACM, 1999.

30- Iván Cantador, Alejandro Bellog´in et David Vallet : Content-based recommendation in social tagging systems. In Xavier Amatriain, Marc Torrens, Paul Resnick et Markus Zanker, éditeurs : Proceedings of the 2010 ACM Conference on Recommender Systems, RecSys 2010, Barcelona, Spain, September 26-30, 2010, pages 237–240. ACM, 2010.

31- Stephen E. Robertson et Karen Spärck Jones: Relevance weighting of search terms. JASIS, 27 (3):129–146, 1976.

32- Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado et Jeffrey Dean: Distributed representations of words and phrases and their compositionality. In Christopher J. C. Burges, Léon Bottou, Zoubin Ghahramani et Kilian Q. Weinberger, éditeurs: Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States., pages 3111–3119, 2013.

33- Cataldo Musto, Giovanni Semeraro, Marco De Gemmis et Pasquale Lops: Word embedding techniques for content-based recommender systems: An empirical evaluation. In Pablo Castells, éditeur : Poster Proceedings of the 9th ACM Conference on Recommender Systems, RecSys

34- Yann LeCun, Yoshua Bengio et Geoffrey E. Hinton: Deep learning. Nature, 521(7553) :436–444, 2015.

35- Julian J. McAuley, Christopher Targett, Qinfeng Shi et Anton van den Hengel : Image-based recommendations on styles and substitutes. In Ricardo A. Baeza-Yates, Mounia Lalmas, Alistair Moffat et Berthier A. Ribeiro-Neto, éditeurs: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, August 9-13, 2015, pages 43–52. ACM, 2015. URL http://doi.acm.org/10.1145/ 2766462.2767755.

36- Saul Vargas et Pablo Castells: Rank and relevance in novelty and diversity metrics for recommender systems. In Bamshad Mobasher, Robin D. Burke, Dietmar Jannach et Gediminas Adomavicius, éditeurs : Proceedings of the 2011 ACM Conference on Recommender Systems, RecSys 2011, Chicago, IL, USA, October 23-27, 2011, pages 109–116. ACM, 2011.

37- Sergio Oramas, Oriol Nieto, Mohamed Sordo et Xavier Serra: A deep multimodal approach for cold-start music recommendation. In Balázs Hidasi, Alexandros Karatzoglou, Oren Sar Shalom, Sander Dieleman, Bracha Shapira et Domonkos Tikk, éditeurs : Proceedings of the 2nd Workshop on Deep Learning for Recommender Systems, DLRS@RecSys 2017, Como, Italy, August 27, 2017, pages 32–37. ACM, 2017. URL http://doi.acm.org/10.1145/ 3125486.3125492.

38- Trapit Bansal, David Belanger et Andrew McCallum : Ask the GRU : Multi-task learning for deep text recommendations. In Shilad Sen, Werner Geyer, Jill Freyne et Pablo Castells, éditeurs :

Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, September 15-19, 2016, pages 107–114. ACM, 2016. URL http://doi.acm.org/10. 1145/2959100.2959180.

39- Cataldo Musto, Claudio Greco, Alessandro Suglia et Giovanni Semeraro : Ask me any rating : A content-based recommender system based on recurrent neural networks. In Giorgio Maria Di Nunzio, Franco Maria Nardini et Salvatore Orlando, éditeurs : Proceedings of the 7th Italian Information Retrieval Workshop, Venezia, Italy, May 30-31, 2016.

40- Alessandro Suglia, Claudio Greco, Cataldo Musto, Marco De Gemmis, Pasquale Lops et Giovanni Semeraro : A deep architecture for content-based recommendations exploiting recurrent neural networks.

41- Pasquale Lops, Marco De Gemmis ET Giovanni Semeraro: Content-based recommender systems: State of the art and trends. In Francesco Ricci, Lior Rokach, Bracha Shapira ET Paul B. Kantor, éditeurs: Recommender Systems Handbook, pages 73–105. Springer, 2011.

42- [Michalski et al. 2013] Fondation Jan Michalski – pour l'écriture et la littérature

43- [Billsus et Pazzani, 2000] Billsus, D. et Pazzani, M. (2000). User modeling for adaptive News access. User Modeling and User-Adapted

44- Ricci, F., Rokach, L. & Shapira, B. (2011). Introduction to Recommender Systems Handbook. In F. Ricci, L. Rokach, B. Shapira, P. B. Kantor, F. Ricci, L. Rokach, B. Shapira & P. B. Kantor (ed.),Recommender Systems Handbook (pp. 1--35). Springer. (ISBN: 978-0-387-85819-7.)

45- Gulzar, Z. et al., 2018. PCRS: Personalized Course Recommender System Based onHybrid Approach. Procedia Comput. Sci. 125, 518–524.https://doi.org/10.1016/j.procs.2017.12.067.

46- Tsymbal A., 2004. The problem of concept drift: definitions and related work.Technical Report TCD-CS- 2004-15, Comput. Sci. Dep. Trinity College Dublin.S

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

48- [DES et KAR, 11] Desrosiers, C., & Karypis, G. (2011). A comprehensive survey of neighborhood-based recommendation methods. In Recommender systems handbook, pp. 107-144. Springer US.

49- **[BRE et al., 98]** Breese J. S., Heckerman D., and Kadie C., (1998). Empirical analysis of predictive algorithms for collaborative filtering, In *Proc. Of the 14th Conf. on UAI*, pp. 43–52.

50- [RES et al., 94] Resnick P., Iacovou N., Suchak M., Bergstrom P., and Riedl J., (1994). GroupLens: An open architecture for collaborative filtering of netnews. In Proc. Of the ACM Conf. on Computer Supported Cooperative Work, USA, pp. 175–186.

51- **[SHA et MAE, 95]** Shardanand, U., & Maes, P. (1995). Social information filtering: algorithms for automating “word of mouth”. *In Proc. of the SIGCHI Conf. on HFCS*, pp. 210-217.

52- **[BRE et al., 98]** Breese J. S., Heckerman D., and Kadie C., (1998). Empirical analysis of predictive algorithms for collaborative filtering, In *Proc. Of the 14th Conf. on UAI*, pp. 43–52.

53- [SAR et al., 01] Sarwar B. M., Karypis G., Konstan A., and Riedl J., (2001). Item-based collaborative filtering recommendation algorithms.

54- [Amatriain et al.2011a] Xavier Amatriain, Neal Lathia, Josep M. Pujol, Haewoon Kwak, and Nuria Oliver. The wisdom of the few: a collaborative filtering approach based on expert
.

55- [Breese et al.1998] John S. Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithm for collaborative filtering. In Proceedings of the 14 th Conference on Uncertainty in Artificial Intelligence, page 43–52, 1998.

56- [Sarwar et al.2001] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In Proceedings of the 10th international conference on World Wide Web, page 285–295, 2001.

57- [Sarwar et al.2000] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Analysis of recommendation algorithms for e-commerce. In Proceedings of the 2nd ACM conferenc on Electronic commerce, EC '00, page 158–167, New York, NY, USA, 2000. ACM.

58- [Su and Khoshgoftaar2009] Xiaoyuan Su and Taghi M. Khoshgoftaar. A survey of collaborative filtering techniques. Advances in Artificial Intelligence, 2009:1–19, 2009.

59- [Papagelis et al. 2005] Papagelis, M. ; Plexousakis, D. et Kutsuras, T. (2005). Alleviating the sparsity problem of collaborative filtering using trust inferences. In iTrust. Springer-Verlag Berlin Heidelberg.

60- [Balabanović et Shoham, 1997] Balabanović, M. et Shoham, Y. (1997). Fab : contentbased, collaborative recommendation. Commun. ACM, 40(3) :66–72.

61- [Burke2002] Robin Burke. Hybrid recommender systems: Survey and experiments. User Modeling and User-Adapted Interaction, 12(4) :331–370, November 2002.

62- [Pazzani1999] Michael J. Pazzani. A framework for collaborative, content-based and demographic filtering. Artif. Intell. Rev., 13(5-6) :393–408, 1999.

63- [POR et al, 09] Porcel, C., López-Herrera, A. G., & Herrera-Viedma, E. (2012). A recommender system for research resources based on fuzzy linguistic modeling. Expert Systems with Applications, 36(3), pp. 5173-5183.

64 -[Saporta 1900] [Lebart et al. 200],

65- [kohonene 1982]

66- J. HAN & M. KAMBER, Data mining: Concepts and techniques, Management Systems (The Morgan Kaufmann Series in Data Management Systems),

67- J. Hartigans. Clustering algorithms. John Wiley and Sons, Inc., 1975.

68-Guillaume Cleuziou. Une méthode de classification non-supervisée pour l'apprentissage de règles et la recherche d'information. Autre [cs.OH]. Université d'Orléans, 2004.

69 -El Khadir Lamrani, El Habib Ben Lahmar, Abdelaziz Marzak, Kamal El Guemmat, Hammad Ballaoui. MÉTHODES DE CLUSTERING DES DOCUMENTS TEXTES ARABES : ÉTUDE COMPARATIVE. 5th International Conference on Arabic Language Processing (CITALA 2014), Nov 2014, Oujda, Maroc

70-A. K. Jain, M. N. Murty, and P. J. Flynn. "Data clustering: a review", ACM Computer Survey, 31(3):264– 323. 1999

71- Basavanneppa TALLUR I.R.I. S.A., Université de Rennes 1, Campus de Beaulieu, Avenue du Général Leclerc, 35042 Rennes Cédex

72- [Germain Forestier] Connaissances et clustering collaboratif d'objets complexes multisources 29 Septembre 2010

73- Samuel AMBAPOUR Introduction à l'analyse des données.

74- Jean-Pierre NAKACHE approche pragmatique de la classification.

75- G. Cleuziou, Une méthode de classification non-supervisée pour l'apprentissage de règles et la recherche d'information, Phd Thesis, Université d'Orleans, France, 2006.

76- L. Gaji´, On ultrametric space, Novi Sad Journal of Mathematics (NSJOM), Vol. 31, No. 2, 69-71, 2001.

77- H. Elghazel. Classification et Prèvision des données Hétérogénes : Application aux Trajectoires et Séjours Hospitaliers. Thèse de doctorat, Université Claude Bernard Lyon 1, 2006.

78- O. Briant M. Chavent and Y. Lechevallier. Comparaison d'une méthode de classification descendante hiérarchique monothétique avec ward et les centres mobiles.

79- M. Chavent. Contributions á la classification automatique et á la réduction de dimension. Thèse de doctorat, Université Bordeaux 1, 2008.

80- O. Briant M. Chavent and Y. Lechevallier. Divclus-t: a monothetic divisive hierarchical clustering method. Computational Statistics and Data Analysis, 2(52) :687– 701, 2007.

81- G. Bisson. Une nouvelle approche pour la classification conceptuelle ascendante. CNRS, Projet SHERPA, Unité de recherche INRIA Rhône-Alpes.

82- R.S. Michalski and R.E. Stepp. Learning from observation: Conceptual clustering. In R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, editors, Machine Learning:

83- [Vincent COURJAULT-RADE] Ballstering: un algorithme de clustering dédié à de grands échantillons l'Université Toulouse 3 Paul Sabatier 2018

84- D. Tounissoux. Classification automatique : Recherche de partitions. Master's thesis, Université de Lyon, Lyon, 2001.

85- L. Gaji´, On ultrametric space, Novi Sad Journal of Mathematics (NSJOM), Vol. 31, No. 2, 69-71, 2001.
86- Z.Guellil et L.Zaoui Proposition d'une solution au problème d'initialisation cas du K-mean Université Mohamed Boudiaf USTO -BP 1505 El Mnaouer -ORAN – Algérie

87- Celeux G,.Diday E., Govaert G., Lechevallier Y., Ralam-Bondrainy H. Classification Automatique des Données. Bordas, Paris, 1989.

88-Likas A., Vlassis M. & Verbeek J., The global k-means clustering algorithm, Pattern Recognition, 36, pp. 451-461.,2003

89- une méthode de classification non-suppervisé pour l'apprentissage de règle et la recherche d'information, Guillaume cleuzieu , université d'orléans , 2004 .

90- M.Ester, H.P, J.Sander, Xu X. A density-based algorithm for discovering clusters in large data bases with noises.

91- [yanis chaouache, chloé agathe] Explorez vos données avec des algorithmes non supervisés, Partitionnez vos données avec DBSCAN

92- STING (a STatistical INformation Grid approach) by Wang, Yang, and Muntz (1997)

93- WaveCluster by Sheikholeslami, Chatterjee, and Zhang (VLDB'98) - A multi-resolution clustering approach using wavelet method

94- CLIQUE - Agrawal, et al. (SIGMOD'98)

95- A. P. Dempster, N. M. Laird, et D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm", Journal of the Royal Statistical Society, 39(1): 1–38. 1977.

96- Pettie & Ramachandran, 2002

97- Nemery, 2006

98- Filkov & Skeina, 2004

99- Mohammadi & al. 2008

100- Chen LJ, Zhang ZK, Liu JH, Gao J, Zhou T (2017) A vertex similarity index for better personalized recommendation. Phys. A 466:607–615

101- .C. Aggarwal , C.K. Reddy , Data Clustering: Algorithms and Applications, 1st, Chap- man and Hall/CRC, 2013

102- C.X. Zhang , Z.K. Zhang , L. Yu , C. Liu , H. Liu , X.Y. Yan , Information filtering via collaborative user clustering modeling, Phys. A 396 (2) (2013) 195–203 .

103- H. Koohi , K. Kiani , User based collaborative filtering using fuzzy c-means, Measure- ment 91 (2016) 134–139 .

104- J. Gao , Y.W. Dong , M. Shang , S.M. Cai , T. Zhou , Group-based ranking method for online rating systems with spamming attacks, Europhys. Lett. 110 (2) (2015) 28003 .

105- R. Hanson , J. Stutz , P. Cheeseman , Bayesian Classification Theory, NASA Ames Re- search Center, Artificial Intelligence Research Branch, 1991, pp. 0–8 .

106- G. Paliouras , C. Papatheodorou , V. Karkaletsis , C.D. Spyropoulos , Clustering the users of large web sites into communities, in: Seventeenth International Conference on Machine Learning, in: ICML '00, Morgan Kaufmann Publishers Inc., San Fran- cisco, CA, USA, 2000, pp. 719–726

107- Q. Ba , X. Li , Z. Bai , Clustering collaborative filtering recommendation system based on svd algorithm, in: IEEE International Conference on Software Engineering and Service Science, 2013, pp. 963–967 .

108- J. Wu , L. Jiao , C. Jin , F. Liu , M. Gong , R. Shang , W. Chen , Overlapping community detection via network dynamics., Phys. Rev. E Stat. Nonlinear Soft Matter Phys. 85 (2) (2012) 016115

109- J. Chen , H. Wang , L. Wang , W. Liu , A dynamic evolutionary clustering perspective: community detection in signed networks by reconstructing neighbor sets, Phys. A 447 (2016) 482–492.

110- J. Wu , L. Jiao , C. Jin , F. Liu , M. Gong , R. Shang , W. Chen , Overlapping community detection via network dynamics., Phys. Rev. E Stat. Nonlinear Soft Matter Phys. 85 (2) (2012) 016115 .

111- Chen, J., Zhao, C., Uliji et al, *Collaborative filtering recommendation algorithm based on user correlation and evolutionary clustering*. Complex Intell. Syst. (2019).

112- J. Wu , L. Jiao , C. Jin , F. Liu , M. Gong , R. Shang , W. Chen , Overlapping community detection via network dynamics., Phys. Rev. E Stat. Nonlinear Soft Matter Phys. 85 (2) (2012) 016115 .

113- J. Chen, H. Wang, L. Wang, W. Liu, A dynamic evolutionary clustering perspective: community detection in signed networks by reconstructing neighbor sets, Phys. A 447 (2016) 482–492.

115-114- [10] B.M. Sarwar , Sparsity, Scalability, and Distribution in Recommender Systems, Uni- versity of Minnesota, 2001 Ph.D. thesis

116- T. Srikanth, M. Shashi, A new similarity measure for user-based collaborative filter- ing in recommender systems, Comput. Technol. 14 (9) (2015) 6118–6128

117- S. Ding, C. Xia, C. Wang, D. Wu, Y. Zhang, Multi-objective optimization based rank- ing prediction for cloud service recommendation, Decis. Support Syst. 101 (2017) 106–114.

118- J. Wu, Y. Jiao, Clustering dynamics of complex discrete-time networks and its application in community detection, Chaos 24 (3) (2014) 033104.

119- J. Chen, Uliji, H. Wang, Z. Yan, Evolutionary heterogeneous clustering for rating prediction based on user collaborative filtering, Swarm Evolut. Comput. (2017)