Université
Dr Tahar Moulay
Saïda - ALGERIE

Faculté de Technologie **Département d'Informatique**

Système de recommandation a base de logique floue

MÈMOIRE

présentée et soutenue le : **5 Juin 2016**

pour l'obtention du

Master en Informatique

(Spécialité : Modélisation Informatique des Connaissances et du Raisonnement)

par

Idrici Mohamed

Encadré par : Fahsi Mahmoud , Maître Assistant à l'Université de Saida

Résumé

Les systèmes de recommandation sont des outils et des techniques utilisé pour trouver des items convenables et aider les utilisateurs a prendre des décisions on a présenté la c-moyenne floues pour faire des recommandations , dans le premier chapitre on a définie les systèmes de recommandations et ces algorithmes et les problèmes rencontré , dans le deuxième chapitre on a vu la classification non-supervisé (clustering) et ces techniques , le troisième chapitre on a présenté la logique floue qui est l'idée de base du c-moyenne floue enfin on a expliquer l'algorithme Système de recommandation basé sur la c-moyenne floue dans le dernier chapitre.

Mots-clés: Systèmes de recommandation ,C-moyenne floue , Clustering

Abstract

Recommender Systems are software tools and techniques providing suggestions for items to be of use to a user. The suggestions provided are aimed at supporting their users in various decision-making processes we have presented Fuzzy c-means in order to make recommendation, in chapter one we have defined the recommender systems and its various algorithms and its problems ,in the second chapter we have seen clustering and its deferent techniques, the third chapter we have seen fuzzy logic that is the basic idea of the fuzzy c-means, finally we have explained implemented recommender system based on Fuzzy c-means in chapter four.

Keywords: Recommender systems,Fuzzy c-means ,Clustering

Remerciements

En premier lieu, je remercie DIEU de m'avoir aidé et donner la force et la volonté pour achever ce modeste travail. Par la suite ce travail a été réalisé sous la direction du monsieur **Mahmoud Fahsi** qu'il trouve ici ma profonde reconnaissance et mes sincères remerciements, pour ses encouragements, son aide ses conseils précieux et aussi ses idées pour la réalisation de ce mémoire. J'adresse mes vifs remerciements à Mon frère **Nourddine**. Je tiens aussi à remercier, les membres du jury qui ont accepté de juger ce travail. Je remercie aussi l'ensemble des enseignants ayant intervenu aux cours de notre parcours en master MICR, trouvent ici l'expression de ma gratitude. Également, je dois à mes camarades et à l'ensemble de mes collègues . Et enfin, je tien à remercier mes chères parents, mes frères et mes oncle pour leurs encouragements, leurs aides et leurs grande patience avec moi.

Idrici Mohamed

Je dédie ce travail à :

Ma famille

Mes proffesseurs

Mon encadreur Mr Fahsi Mahmoud

A tous mes camarades de promotion MICR2

A mes amis

Idrici Mohamed

Table des matières

Table des figures

Glossaire

SR : Système de Recommandation
RI : La recherche d' Information
TF-IDF : Terme Frequency- Inverse Document Frequency
FC : Filtrage Collaboratif
MAE : Erreur Moyenne Absolue
NMAE : Erreur Moyenne Absolue Normalisée
HMAE : Erreur Moyenne Absolue pour les valeurs Hautes
SVD : Division en Valeur Singulière
RMSE : Root Mean Squared Error
ACP : Analyse en Composantes Principales
GMM : Gaussian Mixture Models
FCM : Fuzzy C-Means (C-Moyenne floue)
CHA : Classification Ascendante Hiérarchique
CDH : Classification Descendante Hiérarchique

Introduction Générale

Les systèmes de recommandation automatique sont devenus, à l'instar des moteurs de recherche, un outil incontournable pour tout site Web focalisé sur un certain type d'articles disponibles dans un catalogue riche, que ces articles soient des objets, des produits culturels, des éléments d'information ou encore simplement des pages. Le but des systèmes de recommandation est de prédire l'affinité entre un utilisateur et un article, en se fondant sur un ensemble d'informations déjà acquises sur cet utilisateur et sur d'autres, ainsi que sur cet article et sur d'autres. Dans ce travaille nous allons présenter premièrement les systèmes de recommandation et ces problèmes majeurs ensuite on va voir le clustering et ces déférents techniques et ces limitations, ensuite un chapitre a propos la logique floue qui est la base du C-Means Floue qui nous allons travailler pour faire une amélioration des Systèmes de recommandation basée sur la logique floue ou recommandation floue pour donnée des meilleurs prédictions et satisfaire l'utilisateur même s'il existe toujours des problèmes tell-que le démarrage a froid et la fluctuation et le manque des données et enfin on va terminer avec une comparaison de notre travaille par rapport les autres algorithmes de recommandation .

Chapitre 1

Les systèmes de recommandation

Sommaire

1.1 Introduction

Ce chapitre rassemble l'état de l'art de différentes méthodes et techniques utilisées dans les système de recommandation. Ces systèmes traitent le problème de la surcharge cognitive (ou surcharge d'informations). Trois types d'approches sont principalement utilisés : le filtrage basé sur le contenu, le filtrage collaboratif, et le filtrage hybride. Nous nous intéressons plus formellement aux deux approches les plus communément utilisées : le filtrage basé sur le contenu et le filtrage collaboratif. La première approche basée sur le contenu recherche des items en se basant sur ses caractéristiques et le profil de l'utilisateur. La deuxième approche par filtrage collaboratif recherche des items en se basant sur les choix d'autres utilisateurs dans le système. Les deux classes d'algorithmes de filtrage collaboratif, sont les algorithmes basés sur la mémoire et les algorithmes basés sur le modèle. Ce chapitre permet, en outre, d'identifier certaines limitations du filtrage basé sur le contenu et du filtrage collaboratif. L'émergence et le développement du commerce électronique a conduit au progrès des systèmes de recommandation, un domaine de recherche en plein essor. Ces derniers permettent aux entreprises de filtrer l'information, puis de recommander de manière proactive des produits à leurs clients en fonction de leurs préférences. Recommander des produits et des services peut renforcer la relation entre l'acheteur et le vendeur, et donc augmenter les bénéfices [14]. Les systèmes de recommandation doivent veiller à accroître la satisfaction des utilisateurs. Ces dernières années sont révélatrices de l'utilisation des systèmes de recommandation sur le Web à travers l'intelligence collective, la sensibilisation au contexte et le social computing [15, 16, 17] .Les articles [18, 19] présentent une classification détaillée des systèmes de recommandation pour le commerce électronique, et élucident la façon dont ils peuvent être utilisés pour fournir un service personnalisé fidélisant le client. Actuellement, les moteurs de recommandation [10] reposent sur ces deux paradigmes.

1.2 Les systèmes de recommandation basés sur le contenu

L'objectif des SR à base du contenu est de cibler des objets pertinents issus d'un large espace de sources possibles d'une façon personnalisée pour les utilisateurs. Son principe consiste à recommander les items similaires à ceux préférés par l'utilisa-

teur dans le passé. Ces systèmes à base du contenu considèrent les caractéristiques des items afin de les corréler au profil des utilisateurs. En effet, chaque utilisateur possède un profil le décrivant à travers ses centres d'intérêts. Dans le but de recommander de nouveaux items intéressants, les SR à base de contenu essayent de faire correspondre les attributs des items avec les préférences et les intérêts de l'utilisateur. Pour un nouvel item, le système compare l'item avec le profil de l'utilisateur afin de prédire le score que pourrait porter l'utilisateur sur l'item. Les items sont alors recommandés en fonction de leur proximité aux utilisateurs.
La recherche dédiée aux SR inclut différents domaines, notamment, la recherche d'information (RI), et l'intelligence artificielle [20]. En recherche d'information, les utilisateurs expriment leurs besoins en donnant une requête, alors que dans le système de filtrage d'information, le besoin est exprimé par le profil de l'utilisateur.
Tandis qu'en intelligence artificielle, la recommandation est fondée sur un modèle appris à l'aide des techniques d'apprentissage en exploitant les préférences passées des utilisateurs qui constituent leur profils. Tout simplement, les profils reflètent les intérêts à long terme de l'utilisateur et ils sont représentés par des vecteurs de mots clefs. [21]propose un SR à base de modèle qui permet de faire des prédictions par construction d'un modèle. Ce système a pour objectif de prévoir l'information qui répond à la satisfaction et les besoins réels sans déranger l'utilisateur. Cela implique l'application des systèmes d'apprentissage qui vont apprendre le profil d'utilisateur sans exiger à le fournir et à catégoriser les nouvelles informations en se basant sur celles déjà stockées. Etant donné un nouveau item, le modèle prédictif fourni par les méthodes des systèmes d'apprentissage sera capable de prédire le degré d'intérêt que peut porter l'utilisateur pour l'item.
Pour cette classe de SR à base de contenu, nous ne pouvons pas oublier les techniques de représentation des items et les algorithmes de recommandation utilisés. Les items sont représentés par un ensemble de caractéristiques, par exemple, les descriptions des items dans la plupart des systèmes de filtrage à base de contenu sont des caractéristiques textuelles contrairement aux données structurées, il n'y a pas d'attribut avec les données bien définies.
A cause de l'ambiguïté du langage, la construction d'un profil utilisateur par analyse de caractéristiques textuelles engendre de nombreuses complications.
Les profils basés sur des mots-clefs traditionnels ne sont pas capables de capturer la sémantique des intérêts des utilisateurs, car ils sont essentiellement générés par une opération de correspondance de chaînes. Alors, si une correspondance est trouvée à la fois dans le profil et dans le document, le document est considéré comme

approprié. Cette correspondance de chaine souffre des problèmes de polysémie et de synonymie. La gestion de ces deux problèmes nécessite le développement de techniques d'analyse sémantique. La polysémie rend pertinents de mauvais documents et la synonymie ne permet pas au système d'identifier toutes les informations pertinentes.

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

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

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

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

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

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

1.3 Les Systèmes de recommandation basés sur l'approche collaborative

Le terme Collaborative Filtering est défini comme une technique utilisant les comportements connus d'une population pour prévoir les agissements futurs d'un individu à partir de l'observation de son attitude dans un contexte donné. Un premier exemple personnalisé, "Tapestry "a été mis en place chez Xerox en 1992 [24]. Deux ans plus tard, Paul Resnick du MIT (Massachsetts Institue of Technology) et ses collaborateurs de l'université de Minnesota ont proposé l'architecture Grouplens pour recommander des articles dans les newsgroups [25]. La librairie Amazon a popularisé le filtrage collaboratif avec sa fonction «les utilisateurs qui ont aimé ce livre ont aussi aimé tel autre livre». En 1998, Brin et Page ont publié leur algorithme PageRank et lance Google. A la même année chez Microsoft, John S. Bresse et ses collaborateurs présentent une comparaison détaillée des divers algorithmes de filtrage collaboratif [21].
Durant les années 2000, les algorithmes de FC étaient basés sur les réseaux bayésiens ou les réseaux de neurones avec une approche basée sur l'utilisateur. En 2003, Amazon dépose un brevet introduisant le filtrage collaboratif basé sur l'item [27]. Ce type d'algorithme a été également publié la même année et de façon indépendante par la communauté Grouplens. En 2006, la compagnie Netflix annonce son challenge avec une récompense très attrayante, rendent ainsi disponible un ensemble de données réelles et volumineuses pour évaluer les SR [26]. En 2009, Netflix a décerné un prix d'un million de dollars à l'équipe qui a réussie à améliorer les performances de son SR [26]. Une classe récente de FC est développée, basée sur la factorisation matricielle et sur le contexte [30, 31, 32, 33].
Sans avoir besoin d'information exogène sur les items et les utilisateurs comme dans le filtrage à base de contenu, le FC se base sur des schémas de notation pour produire des recommandations d'items à des utilisateurs donnés. La plupart des méthodes traditionnelles de FC utilisent l'évaluation d'un item par l'utilisateur sur un item lors du calcul de la similarité. D'autres travaux [34, 35, 36, 37, 38, 125] montrent l'importance de l'information démographique et calculent la similarité des utilisateurs à partir de leurs évaluations et leurs informations démographiques. Aussi, des articles ont mis l'accent sur l'utilisation du temps pour les SR : dans [39, 40] les auteurs ont montré que l'année de production d'un film affecte significativement les préférences des utilisateurs cibles. Loren Terveen et al [41] définissent les préférences des utilisa-

teurs en utilisant leurs histoires personnelles. Kazunari Sugiyama a exploré un type de FC en fonction du temps avec une analyse détaillée de l'historique de navigation de l'utilisateur en un jour [42]. Yanchang Zhao et al ont proposé une fonction de énalisation des scores en les considérants comme une série chronologique [43]. Le caractère récent des évaluations a été étudié aussi dans [44, 45]. De nombreuses applications de FC ont été mises en service pendant une longue période, accumulant plusieurs évaluations d'utilisateurs, dont certains sont très anciennes. Cependant, ce caractère récent des notes n'a pas été utilisé jusqu'à présent dans un modèle convexe pour ajuster la prédiction qui utilise la similarité basée sur les notes des items, et celle basée sur les attributs d'items à prévoir automatiquement la préférence d'un utilisateur.

Actuellement, Web-catch-up TV a révolutionné les habitudes car il offre aux utilisateurs la possibilité de regarder des programmes en temps et lieu préféré, en utilisant une variété de dispositifs. Avec l'offre croissante de contenu de télévision, il y a un besoin émergeant des solutions de recommandation personnalisée, qui aident les utilisateurs à choisir des programmes d'intérêt. [46]a développé une série d'approches de recommandation à partir des modèles de l'observation des utilisateurs d'un fournisseur de service de rattrapage de télévision à l'échelle nationale Australienne. L'approche de FC s'appuie sur l'hypothèse que les gens, lors de la recherche d'information, devraient se servir des notes d'autres utilisateurs, à la différence des approches du filtrage à base de contenu, qui utilisent juste les items précédemment notés par un seul utilisateur. Cette approche vient résoudre certains problèmes de l'approche à base de contenu. Ainsi, il devient possible de traiter n'importe quelle forme du contenu grâce aux retours des autres utilisateurs, et de diffuser des items avec des contenus différents non nécessairement similaires à ceux déjà reçus, tant que les autres utilisateurs manifestent leurs intérêts pour ces différents items. Pour ce faire, pour chaque utilisateur, un ensemble de plus proches voisins doit être identifié, et la décision de proposer ou non un item à un utilisateur, dépendra des appréciations de son voisinage. De plus, les recommandations collaboratives sont basées sur la qualité des items évalués par les utilisateurs, au lieu de s'appuyer sur le contenu qui peut être un mauvais indicateur de qualité.

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

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

Soit C l'ensemble de tous les utilisateurs et P l'ensemble de tous les items qui peuvent

être recommandés.Soit U un ensemble ordonné et $U : C \times P \longrightarrow U$ une fonction mesurant le score de l'utilisateur $c \in C$ sur l'item $p \in P$. Pour chaque utilisateur c, le système de recommandation sélectionne l'item $p' \in P$ qui maximise le score ou l'utilité de c .

$$\boxed{\forall c \in C = argmax_{p \in P} u(p, c)}$$

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

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

TABLE 1.1 – Matrice d'usage de trois utilisateurs et six items

La construction de la matrice d'usage se fait soit à partir du filtrage collaboratif passif, qui repose sur l'analyse des comportements des utilisateurs, par exemple les pages web visitées sur une période de temps prédéfinie [48], soit à partir du filtrage collaboratif actif, qui repose sur les données déclarées par les utilisateurs telles que les notes [49].
La recommandation peut être faite par l'exploitation de la matrice d'usage de deux façons différentes. La première approche est connue sous le terme de recommandation basée sur les utilisateurs (user-based) [51], calcule les similarités entre les utilisateurs et à l'aide de leurs profils. La deuxième approche, une recommandation basée sur les items (item-based) [49, 51], qui calcule les similarités entre items et selon les mesures attribuées par les utilisateurs. Une autre méthode combinant les deux approches est proposée dans [50].
Les méthodes collaboratives peuvent être groupées en deux classes générales [21] : La première se base sur le voisinage (basées sur la mémoire, memory based) [21, 25, 100], il s'agit de comparer chaque recommandation pour l'utilisateur courant à l'ensemble de la base de données. La deuxième classe utilise les méthodes à base de modèle,

qui construisent un modèle de prédiction, souvent probabiliste, sur une partie de la base de données.

1.3.1 Recommandation basée sur le voisinage

Les SR basés sur le voisinage(ou à base de mémoire) se fondent sur l'avis de personnes partageant les mêmes idées pour donner une évaluation sur un item p. Ainsi, dans le FC basé sur le voisinage, les notes sont directement utilisées pour prédire les scores des nouveaux items. Dans ce qui suit, Nous introduisons les deux approches basées sur le voisinage entre item et celui entre utilisateur du FC, ainsi que leurs étapes, y compris le calcul de la similarité, et la phase de prédiction .

Voisinage entre utilisateur

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

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

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

Avec a, b sont deux utilisateurs ou deux items, E_a est l'ensemble des items mesurés

par l'utilisateur a et E_b l'ensemble des items mesurés par l'utilisateur b.

$$S_{pearson}(a,b) = \frac{\sum_{x \in E_a \cap E_b}(u(a,x) - \overline{u}_a) \times (u(b,x) - \overline{u}_b)}{\sqrt{\sum_{x \in E_a \cap E_b}(u(a,x) - \overline{u}_a)^2 \sum_{x \in E_a \cap E_b}(u(b,x) - \overline{u}_b)^2}}$$

$O_a(respectivement_a)$représente la moyenne des valeurs contenues dans le vecteur a (respectivement b).
En revanche, si les éléments contiennent uniquement des données binaires, la distance de Jaccard peut être utilisée [10] :

$$S_{jaccard}(a,b) = \frac{|E_a \cap E_b|}{|E_a \cup E_b|}$$

b) Prédiction : La prédiction consiste à calculer l'intérêt qu'un utilisateur pourrait porter à un item ou plusieurs items encore non mesurés. Le principe consiste d'abord à rechercher les utilisateurs possédant les mêmes comportements que l'utilisateur courant. Dès lors, les recommandations sont prédites en fonction des mesures de ces utilisateurs les plus proches.
On peut prédire la note de l'utilisateur pour l'item par la moyenne des notes u(p,c) de ces voisins, mais le problème avec cette méthode est qu'elle ne prend pas en compte le fait que les voisins peuvent avoir des niveaux différents de similarité S. En effet, une solution pour prédire la note de l'utilisateur, est de pondérer la contribution de chaque voisin par sa similarité à c.
De telle sorte que la note prédite devient :

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

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

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

$Ou_c(respectivement_{c_i})$ représente la moyenne des notes de l'utilisateur c (respectivement c_i).

Voisinage entre Items

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

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

AvecE_c de P l'ensemble de tous les items mesurés par l'utilisateur c .
Pour pallier les différences d'utilisations des mesures, la moyenne des notes de chaque utilisateur est introduite dans la formule suivante :

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

$Ou_p(respectivement_{p_j})$ représente la moyenne des notes p(respectivement p_j)

Recommandation basée sur un modèle

Les approches basées sur un modèle mettent en oeuvre des méthodes issues de l'apprentissage automatique comme les modèles bayésiens ou les méthodes de clustering. Ces méthodes sont généralement performantes mais ont un coût de conception et de fonctionnement plus important que les méthodes basées sur la mémoire [55, 56].

Néanmoins, dans le cas de données dispersées, ces méthodes semblent plus efficaces. Pour le lecteur intéressé, une description précise des approches basées sur les modèles est proposée par Su et Khoshgoftaar [56].

A la différence des systèmes basés sur le voisinage qui utilisent les notes stockées pour le calcul de la prédiction, les approches basées sur un modèle utilisent ces notes pour construire un modèle prédictif par apprentissage. L'idée générale est de modéliser les interactions utilisateur-item avec des facteurs représentant des caractéristiques latentes des utilisateurs et items dans le système, comme des classes d'utilisateurs et d'items. Ce modèle est ensuite conçu à partir des données disponibles, et utilisé plus tard pour prédire les notes des utilisateurs pour de nouveaux items. Les approches basées sur un modèle sont nombreuses, elles incluent les méthodes de clustering [21], l'analyse de la sémantique latente [57], les machines de Boltzmann [85], les machines à support vectoriel [85], et la décomposition en Valeur singulière [60, 61, 62]. Les approches basées sur les modèles comme les méthodes de clustering ont été étudiées pour remédier aux insuffisances des algorithmes de FC à base de mémoire [21, 60].Les méthodes existantes de clustering les plus classiques pour le FC peuvent être classées en trois catégories : Les méthodes de partitionnement, les méthodes basées sur la densité et les méthodes hiérarchiques [61, 62].

Une méthode de partitionnement couramment utilisée est l'algorithme k-means proposée par [63] qui a deux avantages : l'efficacité relative et la mise en oeuvre facile. Les méthodes de clustering basées sur la densité recherche nt généralement des classes denses d'objets séparés par des zones creuzes et elles sont bien connues comme méthodes de classification fondées sur la densité [64, 65].

Les méthodes hiérarchiques décrites dans [66], créent une décomposition hiérarchique de l'ensemble des objets de données en utilisant quelques critères. Dans la plupart des situations, le clustering est une étape intermédiaire et son résultat est utilisé pour le calcul de l'évaluation. Les méthodes de clustering pour le FC peuvent être appliquées de différentes façons à savoir le mono-clustering, le bi-clustering, le co-clustering. Dans le cas de mono-clustering [67, 68, 69] les données sont partitionnées en classes utilisant l'algorithme de FC à base de mémoire et la corrélation de pearson comme mesure de similarité.

Dans le cas de l'approche bi-clustering, le filtrage implique à la fois le clustering des utilisateurs et celui des items simultanément.[70] propose un partitionnement simultané pour le filtrage collaboratif en temps réel. Les auteurs du [71, 72] ont utilisé aussi une méthode de co-clustering, mais en introduisant une analyse de la dualité entre les utilisateurs et les items, avec une proposition d'une nouvelle mesure de

similarité.[56] classe les utilisateurs et les items séparément, utilisant les variations des moyens et l'échantillonnage de Gibs. [74]applique le clustering des utilisateurs en se basant sur les items qu'ils évaluent et le clustering des items en se basant sur les utilisateurs qui les ont notés. Les utilisateurs peuvent être réorganisés en fonction du nombre d'items qu'ils évaluent et les items peuvent être regroupés de la même façon. Chaque utilisateur est affecté à une classe avec un degré d'appartenance proportionnelle à la similarité entre l'utilisateur et la moyenne de la classe. Un modèle de mélange flexible (MMF) regroupe les utilisateurs et les items en même temps, permettant à chaque utilisateur et item d'être dans plusieurs classes et modélisant séparément les classes des utilisateurs et des items [75]. Les résultats expérimentaux montrent que l'algorithme MMF a une meilleure précision que l'algorithme de FC basé sur la corrélation de Pearson [76]. En dépit de la rareté, le défi le plus important de FC est l'évolutivité. De nombreux chercheurs ont trouvé que l'utilisation de la technique de co-clustering est plus robuste pour résoudre ce problème, et elle est un moyen viable pour augmenter l'évolutivité tout en conservant une bonne qualité de recommandation [68, 69, 77]. Ainsi, lorsque la base de données est grande,[78, 79] compressent les données d'abord en construisant un modèle de clustering, les recommandations sont ensuite générées en utilisant une approche efficace basée sur les plus proches voisins. Un résumé des travaux sur le FC basés sur le clustering peuvent être consultés dans [80]. Une classe récente de modèles réussis de filtrage collaboratif est basée sur la factorisation matricielle. De nombreuses méthodes ont étudié l'usage des méthodes de factorisation pour le co-clustering, comme est le cas des méthodes SVD, NMF,Tri-NMF, PMF, Non linear PMF, Bayesian PMF, et NPCA [81, 82, 83, 84, 85, 86, 87, 28, 29, 30, 88].

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

Les SR basés sur des modèles tentent à fournir des résultats plus précis que les systèmes basés sur le voisinage. Cependant, la grande partie des travaux de recherche et des systèmes commerciaux (par exemple, Amazon [27], TiVo [93] et Netflix [94]

sont basés sur le voisinage. Actuellement, il existe beaucoup plus de systèmes de recommandation basés sur le voisinage, car ils sont considérés comme plus faciles et intuitifs à manipuler. Tout d'abord, ils fournissent naturellement des explications plus intuitives du raisonnement derrière les recommandations, ce qui améliore l'expérience utilisateur. Enfin, ils peuvent immédiatement délivrer des recommandations à l'utilisateur en se basant sur le retour qu'il vient de fournir.

1.4 Les limitations des types du système de recommandation

Parmi les méthodes présentées, nous remarquons que les SR souffrent de certaines limitations. Ainsi, les systèmes à base de contenu présentent certains problèmes, entre autre, la difficulté d'indexation des documents multimédia. Le filtrage à base de contenu s'appuie sur un profil qui décrit le besoin de l'utilisateur du point de vue thématique.
Ce profil peut prendre divers formats et il repose toujours sur des termes qui seront comparés aux termes qui indexent le document. De ce fait, la difficulté d'indexer des documents multimédia ou non est un goulet d'étranglement pour cette approche. L'incapacité à traiter d'autres critères de pertinence que les critères strictement thématiques, pose également un problème. Il existe plusieurs facteurs de pertinence comme la qualité scientifique des faits présentés, la fiabilité de sources d'informations, le degré de précision des faits présentés, public visé, etc... C'est-à- dire qu'il y a une analyse limitée du contenu. Les techniques à base de contenu ont une limite sur le nombre et le type de caractéristiques qui sont associées aux objets à recommander. La connaissance du domaine est souvent nécessaire, aucune recommandation basée sur le contenu ne peut fournir de suggestions convenables, si l'analyse de contenu ne contient pas d'information pour discriminer les items que l'utilisateur refuse.
L'effet dit « entonnoir » restreint le champ de vision des utilisateurs. En effet, le profil évolue toujours dans le sens d'une expression de besoins de plus en plus spécifique lors de la mise en place d'un filtrage thématique. Ainsi, l'utilisateur ne reçoit que les recommandations relatives à ses préférences, une fois que son profil devient stable, parce que ce dernier évolue naturellement par restriction progressive sur les thèmes recherchés. Par conséquent, il ne peut pas découvrir de nouveaux domaines pouvant potentiellement l'intéresser. Par exemple, lorsqu'un nouvel axe de recherche surgit

dans un domaine, avec de nouveaux termes pour décrire les nouveaux concepts, ces termes n'apparaissent pas dans le profil, ce qui élimine automatiquement les documents par filtrage, l'utilisateur n'aura jamais l'occasion d'exprimer un retour de pertinence positif en vers ce nouvel axe de recherche, à moins d'en avoir connaissance et de modifier son profil manuellement en ajoutant les termes pertinents. Cet inconvénient est appelé problème de « sur-spécialisation » ou « heureux hasard » ou « entonnoir ».

Le paradigme du filtrage collaboratif apporte précisément une réponse à ces problèmes, en s'appuyant sur l'avis d'une communauté d'utilisateurs. Les trois limitations du système à base de contenu (difficulté d'indexation, l'incapacité à traiter d'autres critères, effet entonnoir) n'apparaissent pas dans le filtrage collaboratif. En réponse au problème d'indexation, la sélection ne s'appuie plus sur le contenu des documents, mais sur les opinions que les utilisateurs émettent sur les documents. Un autre avantage de la recommandation basée sur les opinions, c'est qu'elle reflète les autres facteurs de pertinence utiles aux utilisateurs. En effet, lorsqu'un utilisateur émet une opinion positive sur un document, il affirme non seulement que le document traite bien un sujet qui l'intéresse, mais aussi que ce document est de bonne qualité et qu'il lui convient à lui personnellement (public visé). Ainsi, le problème de l'incapacité à traiter d'autres facteurs est également résolu. La qualité de l'information est connue via des évaluations d'utilisateurs. Enfin, l'effet entonnoir est lui aussi éliminé, du fait que les documents entrants ne sont pas filtrés en fonction du contenu. A l'inverse, le FC n'est pas soumis à l'effet entonnoir, car les utilisateurs peuvent tirer profit des mesures d'intérêt des autres utilisateurs et recevoir les recommandations pour lesquelles les utilisateurs le plus proches ont émis un intérêt. Alors, le système peut suggérer des documents sans rapport explicite avec les thèmes déjà évoqués. Cependant, les systèmes collaboratifs ont leurs propres limites. Un problème du FC est celui du démarrage à froid, Actuellement, il existe trois types de démarrage à froid : le système débutant, le nouvel utilisateur, et le nouvel item. Le problème du « système débutant » survient lorsque la matrice d'usage est vide. Les méthodes de filtrage collaboratif ne peuvent fonctionner que s'il existe des informations dans cette matrice d'usage. La solution consiste soit à trouver des variables descriptives des items afin d'organiser ces derniers entre eux et inciter les utilisateurs à les parcourir, remplissant ainsi la matrice d'usage, soit à collecter des données externes en fonction du domaine applicatif.

Afin de formuler des recommandations précises pour un utilisateur, le système de FC doit d'abord apprendre les préférences de l'utilisateur à partir de ces scores.

Le deuxième type de démarrage à froid concerne le nouvel utilisateur. Plusieurs solutions existent : lui soumettre un questionnaire sur les items, ou faire de la recommandation éditoriale afin de l'inciter à parcourir les items et ainsi enrichir le système. Pour éviter cette tâche fastidieuse pour l'utilisateur, certains auteurs proposent d'associer le nouvel utilisateur à un « stéréotype » en exploitant par exemple une source d'informations démographiques externe comme les pages web personnelles des internautes. Le troisième type est celui du nouveau item : dans le cas du filtrage collaboratif, un item n'ayant reçu aucune note, ou n'ayant jamais été acheté ne peut être recommandé.
De nouveaux items sont ajoutés régulièrement à des systèmes de recommandation. Les systèmes collaboratifs reposent uniquement sur les préférences des utilisateurs pour faire des recommandations. Par conséquent, jusqu'à ce que le nouvel élément soit évalué par un nombre important d'utilisateurs, le système de recommandation ne serait pas en mesure de recommander. Il s'agit alors de le rendre visible aux utilisateurs afin d'obtenir un certain nombre de mesures d'intérêt (typiquement le cas des « fausses » recommandations).
Le système de FC exige une base de données substantielle et plusieurs évaluations de l'utilisateur avant d'être utilisable. Dans tout système de recommandation, le nombre de notes déjà obtenu est généralement très faible par rapport au nombre de notes qui doivent être prédites, ce problème est connu sous l'appellation de la rareté « sparcity ». De plus, le succès du système de recommandation collaboratif dépend de la disponibilité d'une masse critique d'utilisateurs et d'items.

1.4.1 Synthèse de la classification des approches de filtrage collaboratif

Nous présentons dans la table 1.2, une synthèse des approches de FC [95, 96, 97, 10] :

- Les algorithmes basés sur la mémoire offrent l'avantage d'être réactifs, en intégrant dynamiquement des nouveaux utilisateurs ou items. Si ces méthodes fonctionnent bien sur des exemples de tailles réduites, il est souvent difficile de passer à des situations proposant un grand nombre d'items ou d'utilisateurs, à cause de la complexité combinatoire des algorithmes utilisés.

	les techniques de FC	
	Approche a base de mémoire	Approche a base de modéle
Avantages	- Simpliste - Performance - Réactif	- Raisonnement prédictif - Moindre complexité
Inconvénients	- Complexité compbinatoire	- Non dynamique

TABLE 1.2 – Synthèse des techniques de FC

- Les algorithmes basés sur un modèle offrent une valeur ajoutée au-delà de la seule fonction de prédiction. En effet, ils mettent en lumière certaines corrélations dans les données, proposant ainsi un raisonnement intuitif pour les recommandations, une autre manière d'aborder le problème du filtrage collaboratif consiste à classifier les utilisateurs et les items en groupes. Pour chaque groupe d'utilisateurs, il s'agit d'estimer la probabilité qu'un item soit choisi. Ces approches souffrent bien souvent du problème de convergence lié à l'initialisation des clusters et fournissent dans certains cas des recommandations de mauvaise qualité.
- Les algorithmes basés sur un modèle minimisent le problème de la complexité combinatoire. Cependant, ces méthodes ne sont pas dynamiques et elles réagissent mal à l'insertion de nouveaux contenus dans la base de données. - Les algorithmes basés aussi bien sur la mémoire et sur le modèle offrent une alternative combinant les avantages des deux approches.

1.5 Domaines d'applications

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

- Content-based filtering systems : utilisant les données sur les items et le profil

de l'utilisateur courant.
- Collaborative filtering systems : utilisant des données sur un ensemble de comportements D'utilisateurs interagissant avec un item.
- Demographic filtering systems : utilisant des données démographiques telles que l'âge, le sexe, le niveau social, etc. permettant de segmenter des populations les rapprochant de certains items.
- Knowledge-based recommender systems : utilisant de la connaissance fonctionnelle pour générer des recommandations
- Utility-based recommender systems : utilisant une fonction d'utilité sur les items pour aider à la recommandation.
- Hybrid recommender systems : utilisant plusieurs approches pour minimiser les inconvénients de certaines méthodes.
Montaner & all.[101] produisent une taxonomie et classifient les systèmes de recommandation existants en plusieurs catégories :

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

Ricci et al.[10] présentent une classification des domaines de recommandations existants en fonction de plusieurs critères d'évaluation subjectifs dont le risque d'impact sur le client suite à une mauvaise recommandation. Il constate par exemple, que les sites d'assurance vie, de tourisme et de recherche d'emplois ont davantage de risque que les sites de commerce électronique, d'actualités, de films ou de musiques. Les systèmes de recommandation sont vitaux pour les sites de commerce en ligne, dont les exemples les plus frappants sont Amazon, NetFlix, Pandora et Strands. Les systèmes de recommandation touchent principalement aujourd'hui quatre domaines commerciaux en ligne : les films , la musique , les livres et la publicité . La recherche dans le domaine des systèmes de recommandation en m-commerce s'est d'ailleurs accélérée ces dernières années. Dans ce cadre, les applications sont nombreuses et variées, nous pouvons mentionner par exemple le tourisme [102] ou la recommandation dans le domaine de la restauration [103]. Le m-commerce ouvre des perspectives de recherche autour de la mobilité, de la capacité de calcul limitée, des capacités de transmission, de la taille de l'écran, etc. La table 3 présente une liste non exhaustive d'exemples de systèmes de recommandation commerciaux et académiques, leur

domaine d'application et la technique de filtrage utilisée.

		Systèmes collaboratifs		
systeme	Domaine	Thématique	Collaboratif	Hybride
Adaptive Place [104]	Restaurants		X	
Amazon [27]	Livres, films, etc.			X
Eigenstate [105]	Académique.		X	
Fab [106]	Livres			X
Fab [106]	Actualités	X		
Last.fm [108]	Musique			X
LIBRA [109]	Livres			X
Google News [110]	Actualités		X	
GroupLens [47]	Actualités	X		
MovieLens [111]	Films		X	
MYCIN [112]	Prescriptions		X	
Netflix [26]	Films			X
Org. Structure [114]	Appareils photos		X	
Pandora [115]	musique		X	
RecTree [69]	Images			X
Ringo [116]	Musique		X	
Tapestry [24]	Images		X	
SASY [117]	Vacances		X	
Top Case [118]	Vacances		X	
TrustWalker [119]	Académique	X		
IMDb [120]	Films			X
Ebay [121]	Tout article confondu			X
Alibaba [122]	Tout article confondu			X
Google Play [123]	musique, films, etc			X
iTunes [124]	musique, films, etc			X

TABLE 1.3 – Classification des systèmes collaboratifs commerciaux et académiques

1.6 Évaluation des systèmes de recommandation

Nous allons examiner comment on peut évaluer la performance du SR pour s'assurer de sa capacité à satisfaire les besoins qui ont conduit à sa création. Le choix d'une mesure, doit être dépendant du type de données à traiter, et des intérêts des utilisateurs [1].

Comme le domaine de la recommandation dérive du domaine de la recherche d'infor-

mation, il est donc souvent normal d'utiliser des mesures d'évaluation de la recherche d'information [2]. Certaines de ces mesures ont été ajustées aux besoins du domaine de la recommandation.
L'évaluation de performance en SR dans la littérature est souvent limitée au calcul de la précision de prédiction. La précision mesure, en général, la différence entre les valeurs des notes prédites par le système de recommandation et les valeurs réellement fournies par les utilisateurs.
L'évaluation de systèmes de recommandation suit, en général, une des trois méthodes :
hors ligne, études sur un échantillon d'utilisateurs ou évaluation en ligne.
L'évaluation hors ligne est la plus simple à réaliser et la moins risquée. Il s'agit globalement de diviser les données disponibles en deux parties, la partie d'apprentissage et la partie test, avant d'utiliser la partie d'apprentissage pour prédire la partie test. Ce type d'évaluation ne pose pas de problème de fuite d'utilisateurs, alors on peut prendre le risque de tester même des approches très fluctuantes. Elle permet d'intégrer facilement une grande masse d'utilisateurs. Elle n'est pas très sensible aux potentiels changements dans le comportement de l'utilisateur. Elle essaie de recopier un comportement que l'utilisateur a eu alors que le système de recommandation n'intervenait pas pour conseiller l'utilisateur, mais elle n'est pas capable de mesurer l'impact d'une telle intervention .

La deuxième méthode (études sur un échantillon) consiste à recruter un groupe de volontaires, auxquels on demande d'exécuter des tâches bien précises en utilisant le SR, de surveiller et d'enregistrer leurs comportements durant l'expérimentation. Ensuite, on peut aussi poser des questions aux participants concernant leurs impressions sur l'expérimentation et le SR.
Ce type de test peut répondre à un large éventail de questions. Il permet de suivre le comportement d'un utilisateur au cours de son interaction avec le SR, et d'observer si ce dernier a influencé le comportement de l'utilisateur. Le questionnaire permet aussi de collecter des données qualitatives pour expliquer les résultats quantitatifs.
Ce type d'évaluation est coûteux. Il n'est pas toujours facile de recruter un nombre suffisant d'utilisateurs, parfois il faut les motiver par des récompenses ou des dédommagements. Le nombre de participants est souvent limité, et on ne peut pas tirer de conclusions concernant une masse d'utilisateurs. De plus, à cause des contraintes de temps des participants, on ne peut pas leur demander de faire des tests excessivement longs. Néanmoins, chaque scénario doit être répété plusieurs fois afin d'assurer

la fiabilité du résultat [3].
Le dernier type est l'évaluation en ligne. Elle est appliquée sur les vrais utilisateurs du système en temps réel [140]. Ce test peut être une simple comparaison des chiffres d'affaires avant et après l'application du SR. On l'applique sur un échantillon d'utilisateurs (tirés au hasard), on observe leurs réactions, et on les compare avec ceux du reste de la population. Ce type d'évaluation comporte des risques. En effet, on peut perdre un utilisateur si le système recommande des items non pertinents. Pour cette raison, on recommande de procéder préalablement à une évaluation hors risque afin de garantir un minimum de qualité de recommandation [3].
Les caractéristiques des tests font que le premier type d'évaluation (hors ligne) est le plus approprié à notre travail, car ce travail se focalise sur des phénomènes comportementaux chez un grand nombre d'utilisateurs. Afin que les résultats émergent, il faut intégrer le comportement d'un nombre considérable d'utilisateurs. Un test à base d'échantillonnage ne sera pas apte à conclure par ce genre de résultats. Dans la section suivante, nous détaillerons les tests hors ligne en montrant les mesures de qualité utilisées pour valider nos résultats.

1.6.1 Évaluation par tests hors ligne

Il est nécessaire de commencer par distinguer deux stratégies selon lesquelles les SR communiquent leurs résultats aux utilisateurs.
Dans la première, le système répond à la question : est-ce que l'utilisateur va apprécier cet item ?
Ces systèmes cherchent à prédire toutes les valeurs manquantes de la matrice de notes, et affichent leur valeur prédite à côté de l'item lors de sa consultation par l'utilisateur, Movielens est un exemple de cette stratégie.
Dans la deuxième, le SR répond à la question suivante : Quels sont les items que l'utilisateur va apprécier ?
Les systèmes qui suivent cette stratégie donnent en sortie une liste ordonnée des meilleurs items que l'utilisateur va apprécier, la valeur numérique de la note prédite n'est pas une priorité, l'essentiel est que la liste contienne des items pertinents.
La mesure d'évaluation doit s'accorder avec la stratégie suivie par le système. Deux classes de mesures d'évaluation sont utilisées : la classe des mesures de précision prédictive, et la classe des mesures de précision du classement [1].

1.6.2 Mesures de précision prédictive

Le but de ces mesures est d'évaluer la précision de la prédiction. La mesure la plus connue dans cette catégorie est l'erreur moyenne absolue (MAE) [141], c'est une mesure statistique qui s'appuie sur la moyenne des différences entre chaque valeur prédite et sa valeur réelle :

$$MAE = \frac{\sum |p_i - a_i|}{N}$$

Cette mesure est bonne pour évaluer globalement la capacité du SR à prédire les notes des utilisateurs. Un grand nombre de variantes de cette mesure ont été proposées pour évaluer de manière plus pointue la précision [142] :

	L'item est sélectionné	L'item n'est pas sélectionné	Somme
Pertinent	NB_{ps}	NB_{pr}	NB_p
Impertinent	NB_{ms}	NB_{mr}	NB_m
	NB_s	NB_r	NB

TABLE 1.4 – F-Mesure

Erreur moyenne absolue normalisée (NMAE) : c'est la version normalisée de MAE. Elle sert principalement à comparer deux modèles où les échelles de notation ne sont pas égales.

$$NMAE = \frac{MAE}{v_{max} - v_{min}}$$

où v_{max} et v_{min} sont respectivement la valeur maximale et minimale de notation.
Erreur moyenne absolue par utilisateur (UMAE) : cette mesure est très importante pour estimer la satisfaction par utilisateur. On calcule une moyenne locale des valeurs prédites par utilisateur, après on calcule la moyenne générale. Supposons qu'un SR prédit 10 notes dont 8 de bonne qualité. Il est possible que les 8 valeurs ne concernent qu'un seul utilisateur (qui note beaucoup), et les deux autres valeurs sont pour deux autres utilisateurs. Dans ce cas, ce système aura un bon score MAE alors qu'il ne satisfait qu'un utilisateur sur trois. UMAE sert à détecter la fluctuation du SR, et

mesure sa capacité à satisfaire le maximum de ses utilisateurs.

$$UMAE = \frac{\sum_{j=1}^{N} \frac{\sum_{i=1}^{N} j|p_{ij} - a_{ij}|}{N_j}}{N}$$

où N est le nombre total d'utilisateurs, N_j est le nombre des notes prédites pour l'utilisateur j.
- Erreur moyenne absolue pour les valeurs hautes (HMAE) : les versions examinées pour l'instant mesurent la capacité du système à prédire les valeurs. HMAE considère les erreurs de prédiction en fonction de leurs impacts sur la recommandation. Elle tolère les erreurs dans les notes basses, mais pas dans les notes élevées. L'équation de HMAE est identique à celle de MAE, sauf qu'elle ne s'applique que quand les notes réelles sont élevées (pour nous une note élevée est 4 ou 5 dans une échelle de notation entre 1 et 5).

1.6.3 Mesures de précision de classement

Ces mesures ne prennent pas en compte la valeur que le SR prédit, mais elles considèrent sa décision de recommander ou non un item. Le SR est récompensé pour les bonnes décisions (l'intégration des items pertinents dans sa liste de recommandations), et pénalisé pour les mauvaises (l'intégration des items non pertinents dans la liste, ou l'absence d'un item pertinent de la liste).
L'objectif est de mesurer la fréquence des bons et mauvais jugements portés par le système de recommandation à l'égard des items.
La F-mesure est la mesure la plus utilisée de cette catégorie. Elle a été proposée la première fois par Cleverdon en 1968 pour les systèmes de recherche d'information. Elle était appliquée la première fois pour les SR par [4]. Elle se compose de deux valeurs : le rappel et la précision. Nous allons expliquer cette mesure avec l'aide du tableau 1.4.
Le rappel est le rapport entre le nombre d'items pertinents sélectionnés et le nombre total d'items pertinents. Formellement :

$$\boxed{R = \frac{Nb_{ps}}{Nb_p}}$$

La précision est le rapport entre le nombre d'items pertinents sélectionnés et le nombre total d'items sélectionnés. Formellement :

$$\boxed{P = \frac{Nb_{ps}}{Nb_s}}$$

La F-mesure est un compromis entre le rappel et la précision :

$$\boxed{F = \frac{2 \times R \times P}{R + P}}$$

Plusieurs autres approximations du rappel et de la précision existent [1, 5]. Dans notre travail, nous présentons une approximation du rappel en considérant qu'il est le rapport entre le nombre de valeurs que le SR a prédites et le nombre de valeurs qu'on lui a demandé de prédire. Nous présentons aussi une approximation de la précision en considé-rant qu'elle est le complément à 1 de l'erreur moyenne absolue normalisée. Notre objectif est de profiter de la puissance de MAE pour mesurer la précision, et d'insérer la notion de couverture (représentée par le rappel) dans cette mesure.

Ces mesures sont utiles dans le cas où le SR doit prédire l'ordre de la liste d'items recommandés. La pertinence d'un item n'est pas une valeur binaire mais condition-née par sa position dans la liste de recommandations. Le système peut être pénalisé par ces mesures même si la liste ne contient que des items pertinents, si ces items sont mal ordonnés.

Dans cette catégorie, nous nous sommes intéressés à la Moyenne du rang réciproque (MRR). Elle est une mesure de qualité utilisée pour évaluer les SR qui doivent donner comme sortie une liste ordonnée avec un seul élément pertinent. Le rang réciproque (RR) d'une liste est égale à 1/r, où r est le rang donné par l'approche évaluée à l'élément pertinent. La moyenne du rang réciproque est la valeur moyenne de RR de toutes les listes. La valeur de cette mesure varie entre 0 et 1, où 1 est le meilleur score de précision. Cette mesure est appropriée à certaines applications comme les systèmes de question-réponse, où le SR doit rendre la réponse la plus exacte à une question donnée.

1.7 Conclusion

Dans ce chapitre, nous avons présenté les systèmes de recommandation. Nous avons examiné plusieurs approches de recommandation. Nous avons ensuite analysé la dominance du FC dans le domaine, nous avons présenté plusieurs facteurs importants pour le fonctionnement du SR, et comparé les approches disponibles en fonction de ces facteurs. en rappelant que les approches de recommandation souffrent en général d'un problème de manque de données qui baisse la performance pour un sous-ensemble d'utilisateurs.

Chapitre 2

La classification automatique « Clustering »

Sommaire

2.1 Introduction

Comme nous avons pu le voir dans le premier chapitre qu'il ya deux grandes approches en classification : la discrimination (classement) et la classification automatique (clustering), dans ce chapitre nous détaillerons les méthodes du deuxième type «clustering» qui est une des techniques statistiques largement utilisées dans la Fouille de Données. 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 savoir d'imiter le mécanisme humaine d'apprentissage sans aucune information disponible auparavant, en établant des méthodes qui permettent d'apprendre à partir d'un certain nombre de données et de règles (d'exemples), selon certains caractéristiques sans aucune expertise ou intervention requise. En effet, ce processus requit certains traitements ou combinaison avec d'autres méthodes, en pre- ou en post-processing,surtout pour une grande masses de données, pour bien réaliser entièrement sa tâche de classification, L'ensemble des techniques de traitement est souvent regroupé sous le terme de «fouille de données».
Dans ce chapitre, nous nous intéressons qu'aux techniques de classification automatique (clustering) et nous montrons, quels sont leurs avantages et difficultés (voir section : Discussion). En tentant décrire quelques remèdes et de présenter l'avantage de Clustering dans le domaine de traitement de données non-étiquetées (sans connaissance préalable).

2.2 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.
C'est un traitement sur un ensemble d'objets qui n'ont pas été étiquetés par un superviseur. Généralement lié au domaine de l'analyse des données comme ACP (analyse linéaire en composantes principales) [7, 11], 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) non définies à l'avance. Parmi les méthodes qu'on peut trouver dans ce type de classification : les cartes

auto-organisatrices de kohonen [9], GMM . . .etc
Un « 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

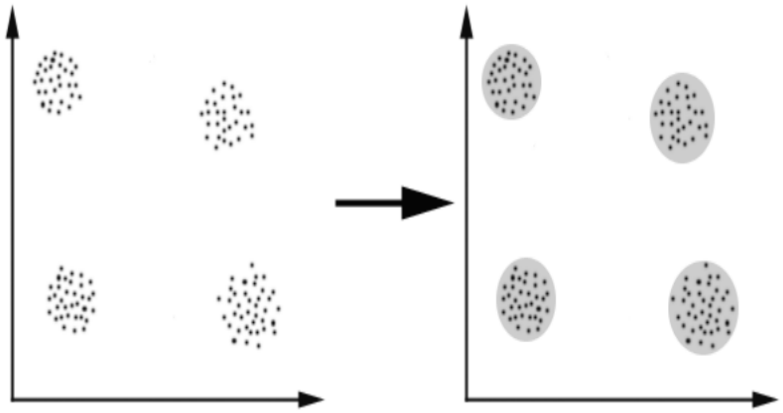

FIGURE 2.1 – Illustration de regroupement en clusters

Dans ce cas, il est très facile pour une personne d'identifier 4 Clusters dans lesquels les données (nuage des points) peuvent êtres divisées, le critère de similarité est la distance : deux ou plusieurs objets appartiennent au même cluster s'ils sont « proches », bien sur cela dépend d'une distance donnée (dans ce cas la distance géométrique).
Un autre type de regroupement est le clustering conceptuel : deux ou plusieurs objets appartiennent au même cluster si celui-ci définit un concept commun à tous les objets. En d'autres termes, les objets sont regroupés en fonction de leur adéquation aux concepts descriptifs, et non pas en fonction de mesures de similarité simple.

2.3 Principe général

Contrairement à la classification (méthodes supervisées), on ne possède pas des connaissances a priori sur les classes prédéfinies des éléments. Donc La division des objets dans les différents groupes (clusters) se procède en se basant sur le calcul de similarité entre les éléments.
Alors que l'objectif des méthodes du Clustering est de grouper des éléments proches

dans un même groupe de manière à ce que deux données d'un même groupe soient le plus similaires possible et que deux éléments de deux groupes différents soient le plus dissemblables possible [8].
Mathématiquement, on a un ensemble X de N données décrites chacune par leurs P attributs.
Donc Le Clustering consiste à créer une partition ou une décomposition de cet ensemble en sous parties (clusters) telle que :

- Les données appartenant au même groupe se ressemblent.
- Les données appartenant à deux groupes différents soient peu ressemblantes.

Exemple

On utilise souvent ce type de classification en traitement d'images pour fixer les divers objets qu'elles contiennent (segmentation) : routes, villes, rues , des organes humaines (pour les images médicales) . .

2.4 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 traiter les dimensionnalités élevées. l'intelligibilité et la convivialité.

2.5 Les types de Clustering

Il existe deux grands types du clustering :

A le clustering hiérarchique : d'agglomération («bottom-up»)

B le clustering non-hiérarchique : de division («top-down»)

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 algorithmes de ce type peuvent aussi être utilisés comme algorithmes de division dans le clustering hiérarchique.

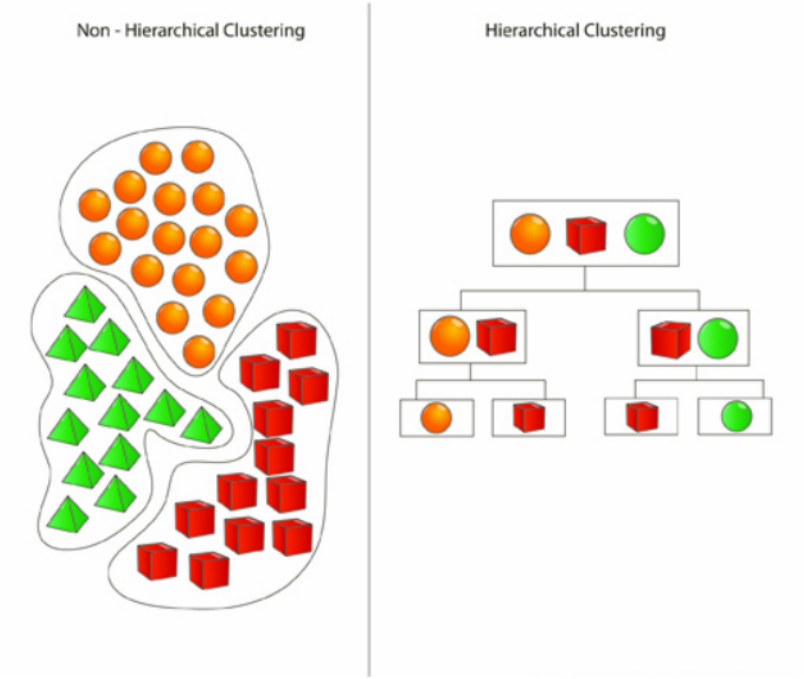

FIGURE 2.2 – les deux types de clustering non-hiérarchique/hiérachique

Cependant dans certains ouvrages on classifie les types des Algorithmes de clustering en 4 groupes à cause des méthodes qui ne respectent plus les normes du premier classement comme le cas de la règle «Chaque objet doit appartenir à un seul groupe.» alors que les versions floues la tempèrent et permettent à un objet d'appartenir à plusieurs classes selon un certain degré.
Les 4 types sont :

1. Clustering exclusif
2. Overlapping Clustering (fuzzy clustering)
3. Clustering Hiérarchique
4. Clustering probabiliste

Dans le premier cas, les données sont regroupées d'une manière exclusive, de sorte que si une donnée certaine appartient à un amas définie alors il ne pourrait pas être inclus dans un autre cluster. Un simple exemple de cela est montré dans la figure ci-dessous, où la séparation des points est définie par une ligne droite sur un plan bidimensionnel.

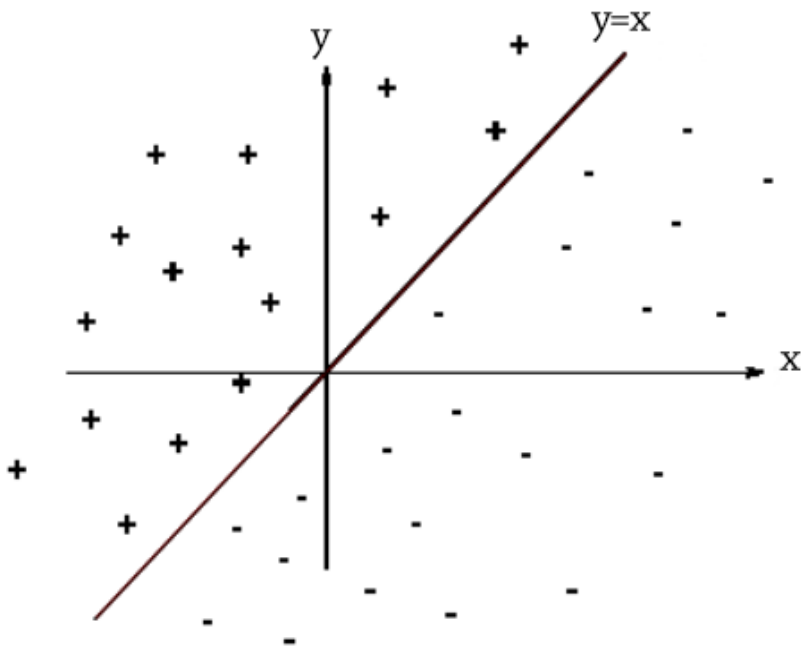

FIGURE 2.3 – Exemple d'un problème de discrimination à deux classes

Avec une séparatrice linéaire : la droite d'équation $y = x$. Le problème est linéairement séparable.
Au contraire le second type, le regroupement overlapping, utilise des ensembles flous aux données de cluster, de sorte que chaque point peut appartenir à deux ou plusieurs groupes avec différents degrés d'appartenance. Dans ce cas, les données seront associées à une valeur d'une composition appropriée.
Comme nous l'avons dit, un algorithme de clustering hiérarchique est fondé sur l'union entre les deux plus proches clusters cad : consiste à trouver des clusters successifs utilisant des clusters précédemment établis. La première condition est de mettre, au début, chaque objet dans un cluster distinct et les fusionner en clusters successivement plus grand. Après quelques itérations on atteint le final Cluster voulu qui regroupe tous les sous-clusters (sous-partitions).
Enfin, le dernier type de regroupement utilise une approche complètement probabi-

liste basant sur la probabilité d'appartenance aux clusters.

2.6 Les algorithmes de Clustering

Dans ce qu'il suit nous présentons quelques algorithmes de Clustering, voilà quelques exemples :

1. K-means
2. Fuzzy C-means
3. Hierarchical clustering
4. Mixture of Gaussians (Expectation maximisation)

Chacun de ces algorithmes appartient à l'un des types de clustering énumérés ci-dessus. Par exemple , K-means est un algorithme de clustering exclusif ,pendant que Fuzzy C-means est un algorithme de Overlapping Clustering, alors que clustering hiérarchique il est claire qu'il s'agit de troisième type de clustering, et enfin Mélange de Gaussien est un algorithme de clustering probabiliste. Nous allons discuter et définir les principes de ces méthode de clustering dans quelques lignes.

2.6.1 K-means

L'algorithme k-means mis au point par McQueen en 1967 [12], un des plus simples algorithmes d'apprentissage non supervisé , appelée algorithme des centres mobiles [13, 29] , il attribue chaque point dans un cluster dont le centre (centroïde) est le plus proche. Le centre est la moyenne de tous les points dans le cluster ,ses coordonnées sont la moyenne arithmétique pour chaque dimension séparément de tous les points dans le cluster cad chaque cluster est représentée par son centre de gravité.

Principe

L'idée principale est de définir les k centroïdes arbitraires c_1,c_2,...,c_k (k le nombre de clusters fixé a priori, chaque ci représente le centre d'une classe), Ces centroïdes doivent être placés dans des emplacements différents. Donc, le meilleur choix est de les placer le plus possible éloignés les uns des autres. La prochaine étape est

de prendre chaque point appartenant à l'ensemble de données et l'associer au plus proche centroïde. Càd Chaque classe Si sera représentée par un ensemble d'individus les plus proches de son ci , Les nuées dynamiques sont une généralisation de ce principe, où chaque cluster est représenté par un noyau mais plus complexe qu'une moyenne.
Lorsqu'aucun point n'est en attente, la première étape est terminée et un groupage précoce est fait. À ce point nous avons besoin de recalculer les k nouveaux centroïdes mi des groupes issus de l'étape précédente qui vont remplacer les ci (m j est le centre de gravité de la classe Sj, calculé en utilisant les nouvelles classes obtenues). Après, on réitère Le processus jusqu'à atteindre un état de stabilité où aucune amélioration n'est possible, nous pouvons constater que les k centroïdes changent leur localisation par étape jusqu'à plus de changements sont effectués.
En d'autres termes les centroïdes ne bougent plus.

Algorithme

Choisir k moyennes c_1, c_2,..., c_k initiales (par exp au hasard)

1. Répéter :
   affectation de chaque point à son cluster le plus proche :

$$\boxed{S_i^{(t)} = \{x_i : \|x_j - m_i^{(t)}\| \leq \|x_j - m_{i*}^{(t)}\| \ for all \ i^* = 1, ..., K\}}$$

   mettre à jour la moyenne de chaque cluster

$$\boxed{m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} X_j}$$

2. Jusqu'à : atteindre la convergence quand il n'y a plus de changement.
   Fin.

Discussion

Cette méthode est la plus populaire des méthodes de clustering , malgré ça , un de ses problèmes majors est qu'il tend à trouver des classes sphériques de même taille. en plus K-means est connu par sa complexité de «NP-difficile». Il est donc

fréquemment faire appeler une heuristique en pratique, ce qui explique qu'elle est convergente et surtout avantageuse du point de vue calcul mais elle dépend essentiellement de la partition initiale (Des initialisations différentes peuvent mener à des clusters différents «problèmes de minima locaux») cela risque d'obtenir une partition qui ne soit pas optimale pourtant qu'elle donne surement une partition meilleure que la partition initiale. De plus, la définition de la classe se fait à partir de son centre, qui pourrait ne pas être un individu de l'ensemble à classer, d'où le risque d'obtenir des classesvides.

2.6.2 méthode Fuzzy C-means

Principe

Fuzzy C-means (FCM) est une méthode de clustering qui permet à un objet de données d'appartenir à deux ou plusieurs clusters. Cette méthode dérivée de l'algorithme c-means [54], identique à l'algorithme k-means décrit précédemment , elle a été développée par Dunn [59]en 1973 et améliorée par Bezdek [107] en 1981, est fréquemment utilisée dans la reconnaissance des formes. Il est basé sur la minimisation de la fonction objective suivante :

$$J_m = \sum_{i=1}^{N}\sum_{j=1}^{C} u_{ij}^{m}\|x_i - c_j\|^2 \, avec \, 1 \leq m \leq \infty$$

où m est un nombre réel (1), U_{ij} est le degré d'appartenance de x_i dans le j ème Cluster, x_i est le ième élément des données mesurées, c_j est le centre d'un cluster et $K * K$ est toute norme exprimant la similarité entre les données mesurées et le centre. Ce Partitionnement logique floue (fuzzy) est réalisé grâce à une optimisation itérative de la fonction objectif indiqué ci-dessus, avec la mise à jour de l'appartenance ui j et les centres des clusters c_j.
On peut résumer la différence entre fuzzy C-means et k-means dans la fonction d'appartenance d'un nuage de points dans deux clusters dans l'exemple suivant :

Dans le cas de k-means un objet ne peut pas appartenir dans deux clusters Simultanément, ce qui explique la Discrimination binaire entre les clusters mais en FCM il est possible qu'un objet appartient à deux ou plusieurs clusters selon différents pourcentages cad que les données sont liés à chaque groupe par le biais d'une

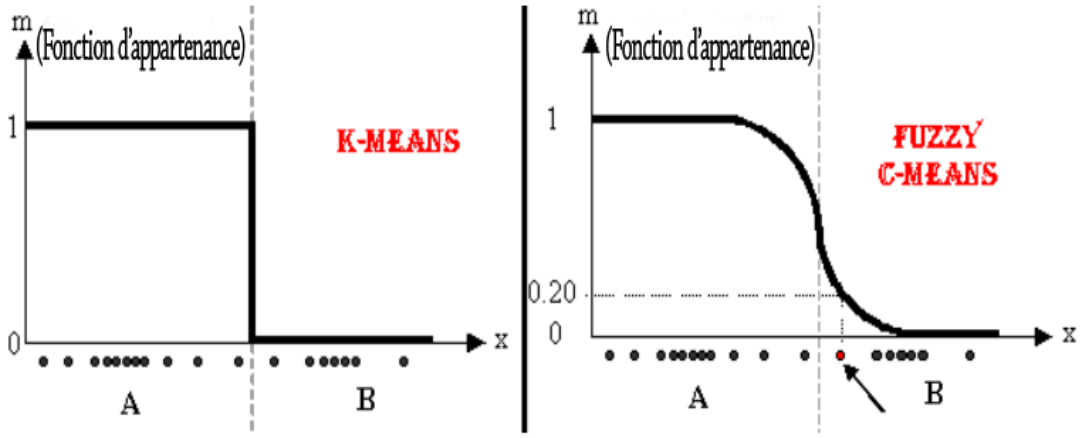

FIGURE 2.4 – Exemple d'un problème de discrimination à deux classes

fonction d'appartenance, ce qui représente le comportement floue de cet algorithme. Pour le faire, nous devons simplement construire une matrice appropriée nommée U dont les facteurs sont des nombres entre 0 et 1, et représentent le degré d'appartenance entre les centres de données et des clusters..

$$U_{N\times C} = \begin{bmatrix} 0.8 & 0.2 \\ 0.3 & 1.7 \\ 0.6 & 0.4 \\ .. & .. \\ 0.9 & 0.1 \end{bmatrix}$$

Il est également important de noter que les initialisations différentes causent différentes évolutions de l'algorithme. En fait, il pourrait converger vers le même résultat, mais probablement avec un nombre différent d'itérations.

Algorithme

Il ya des parties des equations cathé il faut les récriture

1. Initialiser $U = [u_{ij}]$ matrice $U_{(o)}$.
2. A la k-étape : calculer les centres $c_{(k)} = [c_j]$ avec $U_{(k)}$

$$\boxed{c_j = \sum_{i=1}^{N} u_{ij}^m . x_i \sum_{i=1}^{N} u_{ij}^m}$$

3. Mise à jour de $U_{(k)}, U_{(k+1)}$

$$\boxed{u_i^{j=1} C_{k=1} \|x_i - c_j\| \|x_i - c_k\| 2m - 1}$$

4. Si $\|U_{(k+1)} - U_{(k)}\| < \varepsilon, (0 < \varepsilon < 1)$, alors STOP, sinon le retour à l'étape 2.

Discussion

Une méthode que son caractère hybride (la notion de centre de gravité et la notion Floue) le rend simple, rapide . La FCM exige des paramètres d'entrées, et que la matrice de partition floue, doit être initialisée d'une manière appropriée . Ces paramètres sont choisis d'une façon arbitraire, ces paramètres ont une grande influence sur le résultat attendu. Ce qu'il nous oblige de faire une étude approprié sur les données en entrée et le regroupement que l'on souhaite obtenir.
Ce type d'algorithme est fort utilisé en traitement d'images [126, 127, 128] afin d'identifier des zones similaires (contours, coins, région homogènes. ..).

2.6.3 Méthodes hiérarchiques

Le processus basique des méthodes hiérarchiques a été donné par [129, 130], Ce type de clustering consiste à effectuer une suite de regroupements en Clusters de moins en moins fines en agrégeant à chaque étape les objets (simple élément) ou les groupes d'objets (un Cluster-partition) les plus proches. Ce qui nous donne une arborescence de clusters [29]. Cette approche utilise la mesure de similarité pour refléter l'homogénéité ou l'hétérogénéité des classes.

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.
Parmi les algorithmes plus connus de ce type : La classification ascendante hiérarchique (CHA) où le mot ascendante est utilisé pour désigner qu'elle part d'une situation dont tous les individus représentent des clusters à part entière, puis on cherche les rassembler en classes de plus en plus grandes. Ainsi Le qualificatif "hiérarchique" désigne le fait qu'elle produit une hiérarchie, (une amélioration a été proposée en 2002 par P. Bertrand , appelée Classification Ascendante 2-3 Hiérarchique).

Algorithme de CHA

1. Initialisation :
   Chaque individu est placé dans son propre cluster, Calcul de la matrice de ressemblance
   M entre chaque couple de clusters (ici les points)

2. Répéter :
   - Sélection dans M des deux clusters les plus proches C_i et C_j
   - Fusion de C_i et C_j par un cluster C_G plus général
   - Mise à jour de M en calculant la ressemblance entre C_G et les clusters existants

     Jusqu'à fusionner les 2 derniers clusters.
     Dans la figure suivante, on représente une illustration du principe de CHA et la hiérarchie finale obtenue où Les liens hiérarchiques apparaissent clairement.

Note : Un dendrogramme = la représentation graphique d'une classification ascendante hiérarchique sous forme d'un arbre binaire

Discussion

la CAH ne nécessite pas de connaître le nombre de clusters a priori. De plus, il n'y a pas de fonction d'initialisation, ainsi une seule construction d'un cluster (équi-

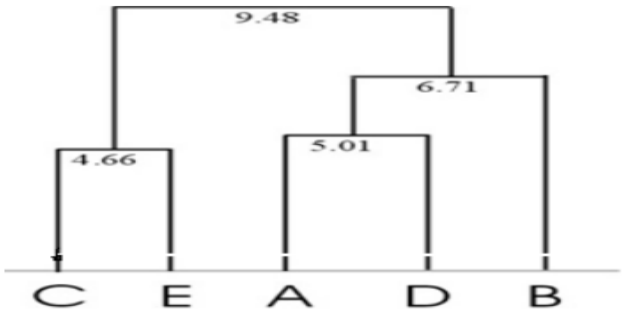

FIGURE 2.5 – le dendrogramme de la hiérarchie H de la suite de partitions d'un ensemble {a,b,c,d,e}

valent à une itération pour les méthodes de partitionnement).
En ce qui concerne généralement les méthodes hiérarchiques le problème qu'on peut rencontrer réside dans la sélection d'une ultra-métrique (distance pour calculer la similarité entre clusters) soit la plus proche de la métrique utilisée pour les individus, car ces méthodes sont heuristiques, pour cela ya plusieurs techniques permet de le faire : Saut minimal (single linkage) ; Saut maximal (complete linkage) ; Saut moyen ; Barycentre. . .
Une autre faiblesse est : la complexité de temps d'au moins O(n2), où n est le nombre d'objets au total, ainsi qu'on pourrait jamais défaire ce qui a été fait précédemment. Il est difficile parfois d'apporter une justification aux méthodes hiérarchique (CAH, CDH..), Cependant, dans [131] , une interprétation probabiliste de la CAH, basée sur une estimation par maximum de vraisemblance des modèles de mélange, est proposée comme solution pour mieux interpréter les résultats.
Un autre inconvénient de ce type de méthodes est qu'une action effectuée (fusion ou décomposition), elle ne peut être annulée. Celà permet de réduire le champ d'exploration, mais une telle astuce ne peut corriger une décision erronée. Afin améliorer la qualité d'une classification hiérarchique, on peut profiter de deux techniques :

- analyser attentivement les liens entre objets à chaque étape [132, 133].
- améliorer la partition obtenue avec une méthode de deuxième type de clustering (partitionnement) [134]

2.7 Mesure de similarité

Pour comparer homogénéité ou le ressemblance, la similarité entre deux objets (points,images, classes , phonème ..), il faut pouvoir mesurer la similarité (ou la

dissimilarité) entre eux.
Nous allons décrire maintenant des mesures de similarité pour prouver la similarité entre les objets, selon [135], « tout systeme 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 la similarité est une partie importante de la définition d'une méthode de clustering, elle consiste en effet à définir et formaliser une mesure de similarité adaptée aux caractéristiques des données. Si les composantes des vecteurs de données d'instance sont toutes dans les mêmes unités physiques alors il est possible que la distance euclidienne est suffisante pour réussir à grouper les données similaires. Cependant, même dans ce cas, la distance euclidienne peut parfois être trompeuse. La Figure ci-dessous illustre ceci avec un exemple vu selon la largeur et la hauteur d'un objet. Malgré que les deux mesures aient été prises dans les mêmes unités physiques.

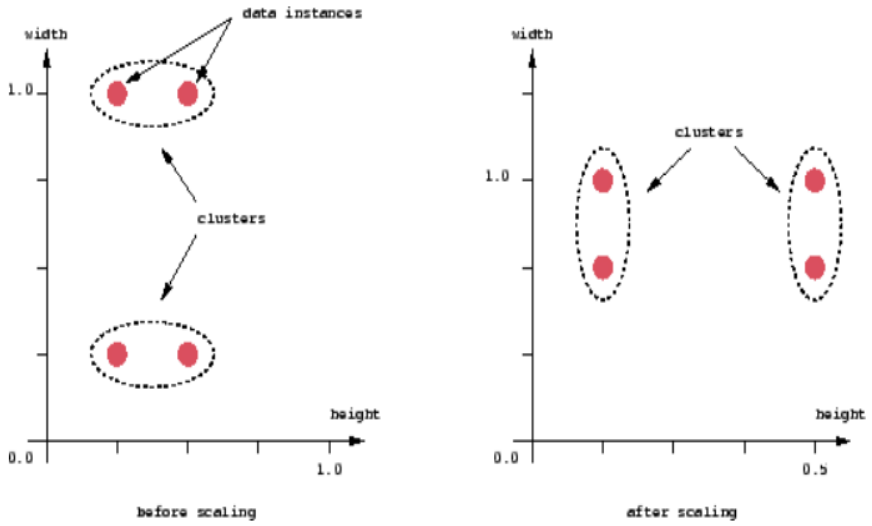

FIGURE 2.6 – différents écaillages peuvent conduire à différents clustering

Donc une décision éclairée doit être faite quant à la mise à l'échelle relative. Comme le montre la figure, différents écaillages peuvent conduire à différents clustering.

2.7.1 Vocabulaire

Il est à noter qu'il ya deux concepts pour exprimer la notion de proximité entre les objets à classifier :

1. Mesure de dissimilarité : plus la mesure est faible plus les points sont similaires (distance).
2. Mesure de similarité : plus la mesure est grande, plus les points sont similaires.
3. On parle souvent de « distances » en désignant une mesure de similarité, lorsque ces mesures ont les propriétés de non-négativité, réflexivité, symétrie (la distance entre l'objet A à B est la même que la distance de B à A) et qui respectent l'inégalité triangulaire.

Il existe un grand nombre de mesures de similarité, dans ce qui suit, nous présentons quelques'unes des fonctions entre deux objets d(x1 ; x2).

2.7.2 Fonctions de similarité

a) La distance euclidienne :

(aussi appelée la distance à vol d'oiseau) Un rapport de clusters analysis en psychologie de la santé a conclu que la mesure de la distance la plus courante dans les études publiées dans ce domaine de recherche est la distance euclidienne ou la distance au carré euclidienne

$$\boxed{d^2(x_{1i}, x_{2i}) = \sum_i (x_{1i} - x_{2i})^2 = (x_1 - x_2)(x_1 - x_2)`}$$

b)La distance de Manhattan : (appelée aussi taxi-distance)

$$\boxed{d^2(x_1 - x_2) = \sum_i |x_{1i} - x_{2i}|}$$

c) La distance de Mahalanobis

corrige les données pour les différentes échelles et des corrélations dans les variables, L'angle entre deux vecteurs peuvent être utilisés comme mesure de distance quand le regroupement des données de haute dimension. Voir l'espace produit scalaire.

$$\boxed{d^2(x_1, x_2) = (x_1 - x_2)C^{-1}(x_1 - x_2)\text{`}}$$

$$\boxed{C = convariance}$$

d) La distance de Sebestyen

$$\boxed{d^2(x_1, x_2) = (x_1 - x_2)W(x_1 \vdash x_2)\text{`}}$$

$$\boxed{W = Matrice\ Diagonal\ De\ Pondration}$$

e)La distance de Hamming

mesure le nombre minimum de substitutions nécessaires pour changer un membre dans un autre. Elle permet ainsi , de quantifier la différence entre deux séquences de symboles, généralement utilisée dans le cas des valeurs discrètes (vecteurs)

$$\boxed{d(a, b) = \sum_{i=0}^{n-1} (a_i \oplus b_i)}$$

Exemple : Considérons les suites binaires suivantes :
$a = (0001111)$ et $b = 1101011$ alors $d = 1 + 1 + 0 + 0 + 1 + 0 + 0$ La distance entre a et b est égale à 3 car 3 bits diffèrent.

f) Distances entre distributions

La similarité entre distributions consiste à déterminer si deux distributions peuvent être issues de la même distribution de probabilités.
Le test statistique du X_2 chi-square) permet de décider si deux vecteurs x et y sont engendrés par la même distribution. La version symétrique du test est :

$$\boxed{x^2(\vec{x},\vec{y}) = \sum_{i=1}^{n} \frac{(x_i - y_i)^2}{x_i + y_i}}$$

Cependant que pour les données de grandes dimensions, il ya une distance spécifique très utilisée :

g) La métrique Minkowski

Pour les données dimensionnelles, c'est la mesure populaire

$$\boxed{d_p(x_i, x_j) = (\sum_{k=1}^{d} |x_{ik} - x_{jk}|^P) \frac{1}{P}}$$

où d est la dimensionnalité des données. La distance euclidienne est un cas particulier où $p = 2$, alors que Manhattan $p = 1$. Néanmoins, il n'existe pas de directives générales théoriques pour la sélection d'une mesure à une application donnée. Une autre question, est de savoir comment mesurer la distance entre 2 classes $D(C_1; C_2)$?
Pour cela il ya certaines fonctions permettent de mesurer cette distance comme :
plus proche voisin :

$$\boxed{min(d(i,j), i \in C_1, j \in C_2)}$$

diamètre maximum :

$$\boxed{max(d(i,j), i \in C_1, j \in C_2)}$$

distance moyenne :

$$\boxed{d(\mu_1, \mu_2)}$$

distance de Ward :

$$\boxed{\sqrt{\frac{n_1 n_2}{n_1 + n_2}} d(\mu_1, \mu_2)}$$

2.7.3 Discussion

Une note importante est de savoir si le clustering utilise une distance symétrique ou asymétrique. Bon nombre des fonctions énumérées ci-dessus ont la propriété que les distances sont symétriques. Dans d'autres applications (par exemple, la séquence-alignement des méthodes, ce n'est pas le cas. Certaines mesures sont spécifiques aux domaines particuliers comme histogrammes ou aux distributions.
En plus, ces mesures rencontrent certaines difficultés lorsque'on change le jeu de données comme le fait de travailler sur des espaces de couleurs où quelque distance ne sont pas recommandées. L'inconvénient major de la plupart de ces fonctions, c'est qu'elles sont coûteuses en temps de calcul et sont de plus sensibles à la dimension des données. Pour remédier le problème de dimensions, il ya des techniques ont été proposées pour la réductions de dimensions, qui permettent d'appréhender cette difficulté [136].

2.8 Les limites de Clustering

Il ya un certain nombre de problèmes avec le clustering. Parmi eux :

- les techniques de clustering actuelles ne traitent pas tous les besoins de façon adéquate (et simultanément), comme le fait que si nous n'avons pas des variables continuées (la longueur), mais les catégories nominales, comme les jours de la semaine. Dans ces cas encore, la connaissance du domaine doit être faite pour formuler le clustering appropriée.
- 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.
- si la mesure de la distance n'existe pas, nous devons la « définir », ce qui n'est pas toujours facile, surtout dans des espaces multidimensionnels.
- le résultat de l'algorithme de clustering peut être interprété de différentes manières.
- 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. ie : connaissance de la valeur correcte à l'avance, la valeur appropriée doit être déterminée.

2.9 Les caractéristiques des différentes méthodes

Quel que soit le type de la classification il ya Trois éléments permettent de caractériser les différentes méthodes :

1. La classification se déroule séquentiellement en regroupant les observations les plus 'semblables' (méthodes hiérarchiques) ou elle regroupe en k groupes toutes les observations simultanément (méthodes non-hiérarchiques).
2. Le critère de 'ressemblance' entre deux observations.
3. Le critère de 'ressemblance' entre deux groupes ou entre une observation et un groupe.

Ces trois éléments permettent de définir le déroulement ainsi que le type de la méthode, le deuxième et le troisième caractère ont un point primordial dans la performance et la qualité du résultat attendu d'une méthode, car il y aura certainement une différence de calcule (précision) entre le fait d'utiliser la distance euclidien au lieu de la distance de Hamming (cad que la distance utilisée est prise en considération afin d'amélioré les résultats) [137, 138]

2.10 Conclusion

Les méthodes de clustering comme toutes les autres méthodes de classification , ont leurs avantages , faiblesses (voir section : discussion) , cependant , il n'y a pas que le type statistique , il y'en a d'autre type qui s'appuie sur la théorie de probabilité . Dans le chapitre suivant nous nous intéresserons à une nouvelle méthode

de conception totalement différente de ce que nous l'avons vu jusqu'à maintenant, basée sur la conception du modèle de mélange, une méthode qui a été classé la 5ème parmi les méthodes de classification les plus utilisées/populaires de DATA MINING ces dernières années [139], un classement prouve le succès qu'il a commencé à rencontrer très rapidement ce type de méthodes.

Chapitre 3

La logique floue

Sommaire

3.1 Introduction

La plupart des problèmes rencontrés sont modélisables mathématiquement Mais ces modèles nécessitent des hypothèses parfois trop restrictives, rendant délicate l'application au monde réel. Les problèmes du monde réel doivent tenir compte d'informations imprécises et incertaines.
La logique floue généralise la logique classique avec des variables logiques et des formules logiques prenant des degrés de valeur de vérité quelconques entre 0 (faux) et 1 (vrai) inclusivement. La logique classique avec ses valeurs de vérité booléennes de 0 et 1 est considérée comme un cas particulier de la logique floue [143].

3.2 Définition

Le terme d'ensemble flou apparaît pour la première fois en 1965 lorsque le professeur Lotfi A. Zadeh, de l'université de Berkeley aux USA, publie un article intitulé «Ensembles flous» (Fuzzy sets). Il a réalisé depuis de nombreuses avancées théoriques majeures dans le domaine et a été rapidement accompagné par de nombreux chercheurs développant des travaux théoriques. La logique floue s'appuie sur la théorie mathématique des ensembles flous. C'est une théorie formelle et mathématique dans le sens où Zadeh, en partant du concept de fonction d'appartenance pour modéliser la définition d'un sousensemble d'un univers donné, a élaboré un modèle complet de propriétés et de définitions formelles. Il a aussi montré que cette théorie des sous-ensembles flous se réduit effectivement à la théorie des sous-ensembles classiques dans le cas où les fonctions d'appartenance considérées prennent des valeurs binaires (0,1).

3.3 La théorie des sous ensembles flous

La théorie des sous-ensembles flous et les outils de raisonnement qui en découlent, proposent un cadre formel qui permet de modéliser le langage naturel et de gérer l'imprécis et l'incertain.
Les sous-ensembles flous (ou parties floues) ont été introduits afin de modéliser la représentation humaine des connaissances, et ainsi améliorer les performances des systèmes de décision qui utilisent cette modélisation. Les sous-ensembles flous sont

utilisés soit pour modéliser l'incertitude et l'imprécision, soit pour représenter des informations précises sous forme lexicale assimilable par un système expert [144].

3.4 Les fonctions d'appartenance

On peut donner un coefficient de confiance à l'affirmation « X appartient à un ensemble A », par exemple : le coefficient d'appréciation de « l'air à la température égale à 30°C est chaud » vaut 0.6, ce qui signifie que cette température correspond à « plutôchaude ».
On peut pour toute température, donc pour tout X, définir ce coefficient directement à X.
Cette propriété se présente facilement par une fonction dite d'appartenance $\mu_A(X)$ à valeurs dans [0,1], la notion signifie « coefficient d'appartenance de X à l'ensemble caractérisé par A », l'argument X se rattache à la variable linguistique et l'indice A désigne l'ensemble concerné De la même manière, une variable Y appartiendra à un ensemble B avec une fonction d'appartenance notée $\mu_B(Y)$ par exemple « le vent est fort ». On peut associer X et Y dans une même phase, par exemple l'ensemble C : « l'air est chaud et le vent est fort ». La variable Z définie par : « air chaud et vent fort » correspond à l'intersection de « air est chaud » et de « vent est fort ». L'ensemble C correspond à l'intersection des ensembles A et B. la valeur de $\mu_C(Z)$ se déduit des valeurs de $\mu_A(X)$ $\mu_B(Y)$ . . Il existe diverses solutions pour traduire mathématiquement le problème.
Un fait certain aura une fonction d'appartenance égale à 1 pour le point de fonctionnement considéré. Un fait incertain aura une fonction d'appartenance inférieure ou égale à 1.
Lorsque le fait certain correspond à l'énoncée de la valeur d'une variable $X = X_0$on aura $\mu_{X_0}(X) = 1$pour $X = X_0$et $\mu_{X_0}(X) = 0$pour X différent de X_0 : on a un singleton. Un fait incertain tel que x à peu prés égale à X_0 : aura une fonction d'appartenance en forme de tringle. L'affirmation de X à peu prés compris entre X_0 et X_1 correspond à une fonction trapézoïdale. Les fonctions d'appartenance peuvent avoir diverses formes selon leur définition : triangulaire, trapézoïdale, Gaussienne, Sigmoïdes...

3.5 Les caractéristiques d'un sous ensemble flou

Un ensemble flou F de l'univers X est caractérisé par :

Le noyau

Noté noy(F), qui représente l'ensemble des éléments de X pour lesquels la fonction d'appartenance vaut 1 :

$$\boxed{noy(F) = \{x \in X / \mu_F(x) = 1\}}$$

Le support

Noté supp(F), qui représente l'ensemble des éléments de X appartenant, même très peu, à F, c.-à-d., ayant $\mu_F(X)$ qui n'est pas nulle :

$$\boxed{supp(F) = \{x \in X / \mu_F(x) \neq 0\}}$$

La hauteur

Notée h(F), qui représente la plus grande valeur prise par sa fonction d'appartenance :

$$\boxed{h(F) = sup\mu_F(x), x \in X}$$

α -Coupe

Qui représente l'ensemble contenant les éléments ayant un degré d'appartenance Supérieur ou égal à a :

$$\boxed{\alpha - Coupe_F = \{x \in X / \mu_F(x) \geqslant a\}}$$

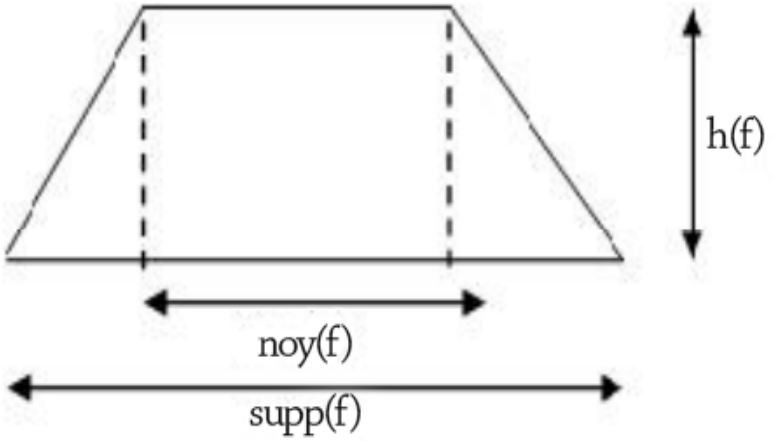

FIGURE 3.1 – le noyau , le support ,la hauteur d'un sous ensemble flou

3.6 Les opérations ensemblistes

Il existe de nombreuses variantes dans ces opérateurs. Cependant, les plus répandus sont ceux dits « de Zadeh » décrits ci-dessous. Leur utilisation sera reprise dans l'exemple didactique d'utilisation d'une base de règles floues.
Dans ce qui suit, le degré de vérité d'une proposition A sera noté μ_A

3.6.1 La réunion

L'opérateur logique correspondant à l'union d'ensembles est le OU. Le degré de vérité de la proposition « A OU B » est le maximum des degrés de vérité de A et de B.

Exemple 1

A est l'ensemble flou des personnes petites.

B est l'ensemble flou des personnes moyennes.

L'ensemble des personnes petites OU moyennes est un ensemble flou de fonction d'appartenance :

$$\boxed{\mu_{A\cup B}(x) = max(\mu_A(x)(x), \mu_B(x))\ \forall x \in U}$$

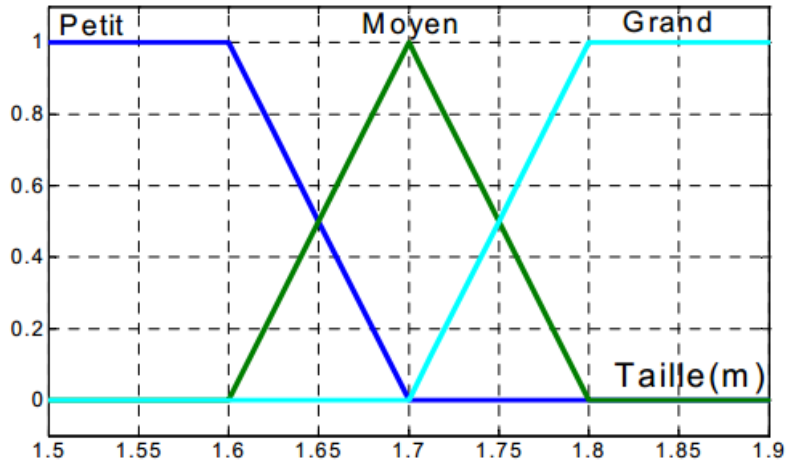

FIGURE 3.2 – Partition floue de l'univers du discours

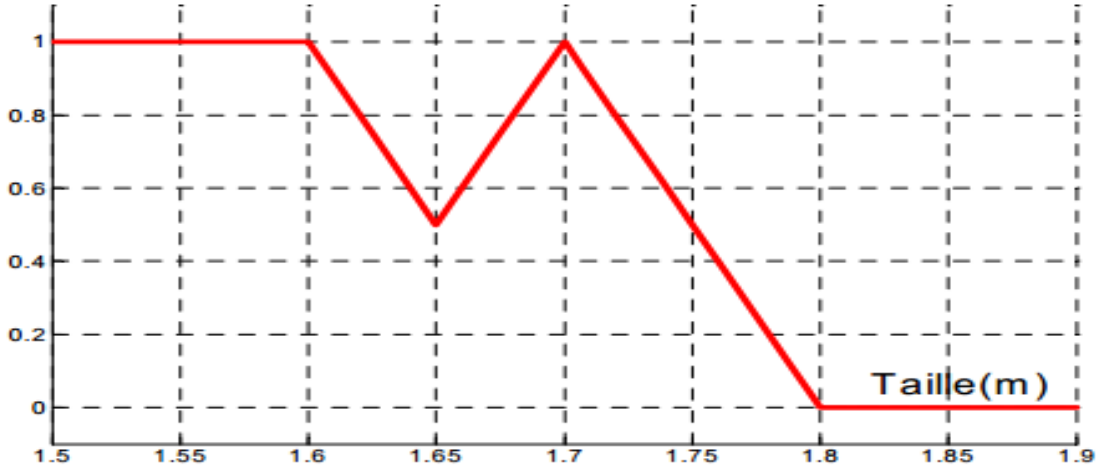

FIGURE 3.3 – Ensemble flou « la réunion »

3.6.2 L'intersection

L'opérateur logique correspondant à l'intersection d'ensembles est le ET. Le degré de vérité de la proposition « A ET B » est le minimum des degrés de vérité de A

et de B.

Exemple 2

A est l'ensemble flou des personnes petites. B est l'ensemble flou des personnes moyennes. L'ensemble des personnes petites ET moyennes est un ensemble flou de fonction d'appartenance :

$$\boxed{\mu_{A\cup B}(x) = min(\mu_A(x)(x), \mu_B(x))\ \forall x \in U}$$

3.6.3 Le complément

L'opérateur logique correspondant au complément d'un ensemble est la négation.

Exemple 3

A A est l'ensemble flou des personnes petites. L'ensemble des personnes NON petites est un ensemble flou de fonction d'appartenance : [145].

$$\boxed{\mu_A(x) = 1 - \mu_A(x)\ \forall x \in U}$$

3.6.4 Différentes représentations de sous-sembles flous

Appellation	ET	OU	NON
Zedah	$min(x, y)$	$Max(x, y)$	$1 - x$
Pro balistique	xy	$x \div y - xy$	$1 - x$
Lukasiewiez	$min(x + y - 1, 0)$	$min(x + y, 1)$	$1 - x$
Ha ...($\beta > 0$)	$xy/(\beta + (1 - \beta)(x + y.xy))$	$\frac{(x \div y + xy - (1 - \beta)xy)}{(1 - (1 - \beta)xy)}$	$1 - x$
Weber	x si y=1,y si x=1	x si y=0,y si x=0 , 1 sinon	$1 - x$

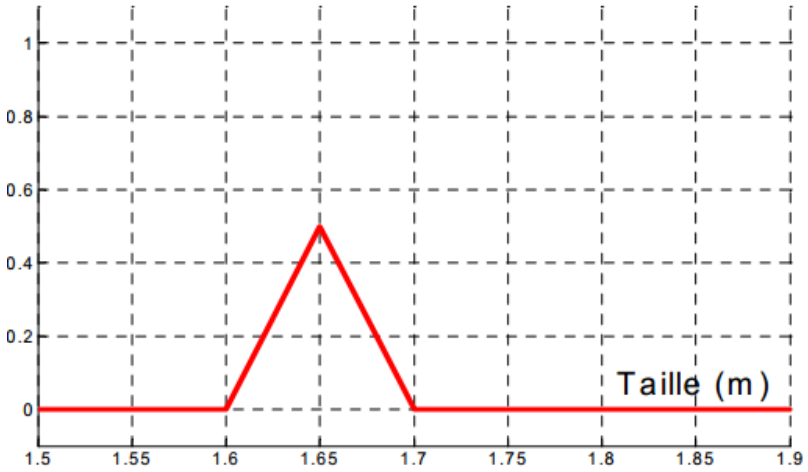

FIGURE 3.4 – Ensemble flou «l'intersection» «complément»

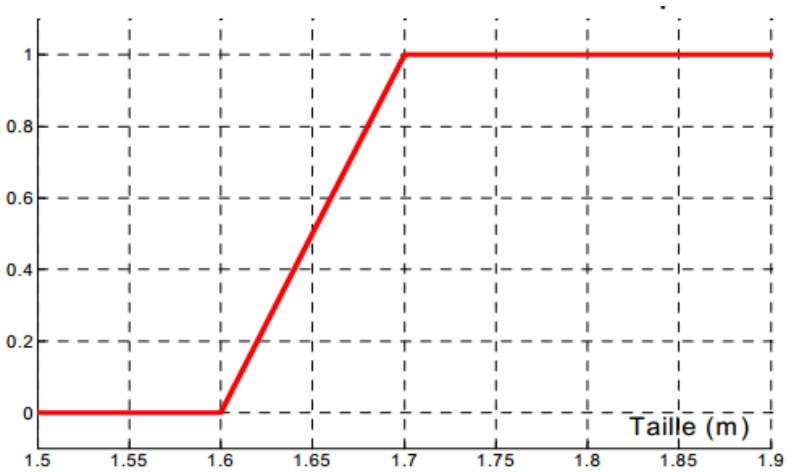

FIGURE 3.5 – Ensemble floue

3.6.5 Les relations floues

Le concept de relation floue généralise celui de relation classique. Il met en évidence des liaisons imprécises ou graduelles entre les éléments d'un ou plusieurs ensembles [146].

Une relation floue R sur les univers de références X_1, X_2,..., X_n est définie comme un ensemble flou du produit cartésien $X_1 \times X_2 \times .. \times X_n$ ayant la fonction d'appartenance μ_R.

Exemple :

$$\boxed{\mu_R = \frac{1}{1+(x-y)^2}}$$

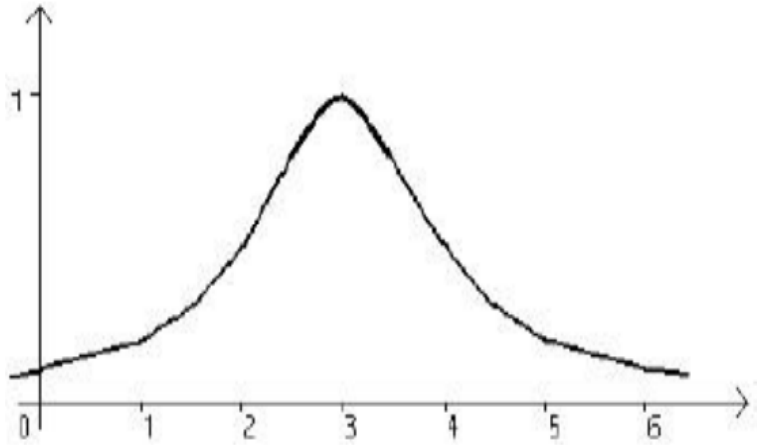

FIGURE 3.6 – x approximativement égal à 3

La différence entre une relation floue et une relation classique (exacte) est que pour la première, toute valeur d'appartenance dans l'intervalle [0,1] est permise alors que pour la seconde, seules les valeurs 0 et 1 sont permises [147].
Une relation floue R sur X est dite :

- symétrique
- antisymétrique
- réflexive
- transitive

Remarque :

- Une relation de similarité est une relation réflexive, transitive et symétrique. Une relation de similarité permet de modéliser la ressemblance et la proximité.
- Une relation d'ordre floue est une relation réflexive, transitive et antisymétrique. Une relation d'ordre exprime la notion de préférence et d'antériorité.

3.6.6 Règles floues

Les règles floues sont beaucoup plus faciles à exprimer car elles sont très proches du langage naturel.

Définition

Une règle floue est une proposition floue de la forme : « Si P alors Q » utilisant une implication entre deux propositions floues quelconques P et Q.

Exemple

Règle : « Si la surface est grande » « Alors prix est élevé »

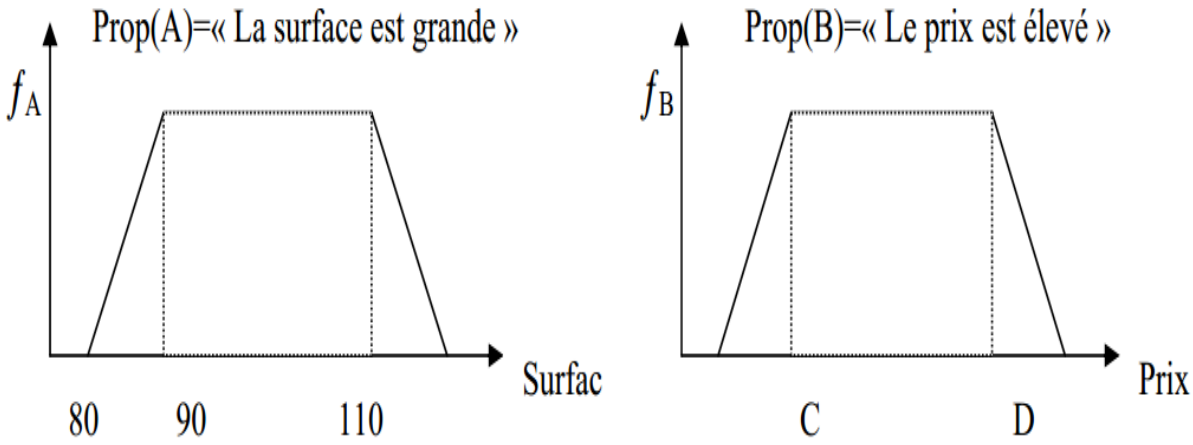

FIGURE 3.7 – Représentation d'une règle floue

3.6.7 Variables linguistiques

Une variable linguistique est un triplet (V, X, T) dans lequel V est une variable définie sur un ensemble de références X, l'ensemble $Tv = \{A_1, A_2, .., A_n\}$, fini ou infini contient des sous-ensembles flous normalisés de X, utilisables pour caractériser V.

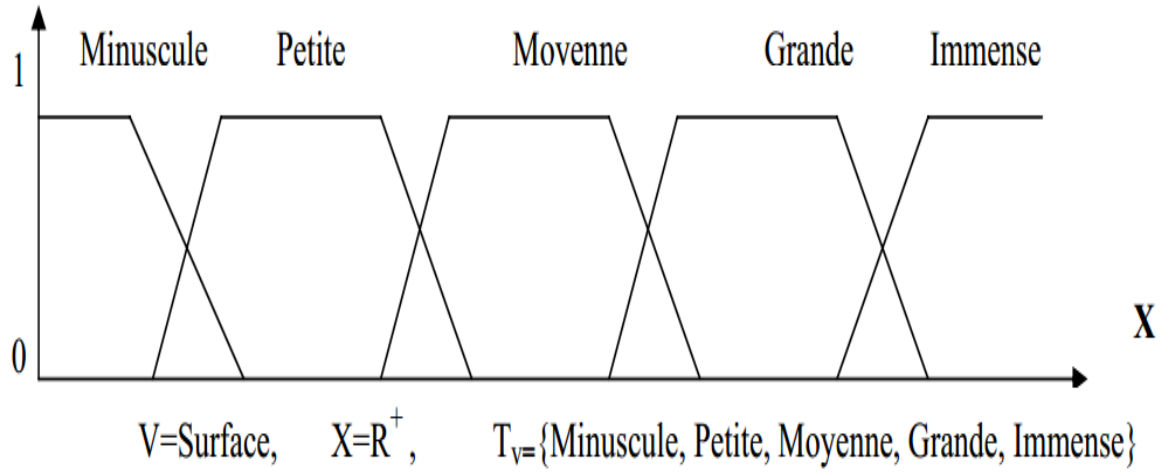

FIGURE 3.8 – Variable linguistique (V,X,TV) décrivant la surface d'un appartement

Fonction d'appartenance

Pour utiliser dans le même cadre, connaissance numérique et connaissance symbolique et si l'attribut A peut subir des variations graduelles, liées à un environnement imprécis, on utilise des expressions qui sont toujours de la forme « V est A » mais pour laquelle la variable est associée à une variable linguistique. On se limite alors à des descriptions de la forme ««Le temps est beau», «La robe est chère».
Etant donnée un ensemble flou L de variables linguistiques (V, X, T) de L pour la qualification « V est A » où A est une caractérisation floue de T_V.

Proposition générale floue

Une proposition générale floue est obtenue par l'utilisation conjointe de propositions floues élémentaires « V est A », « W est B » pour des variables V,W, ...supposée non interactives. La plus simples s'exprime comme la conjointe de propositions floues élémentaires « V est (A et W est B) où V et W sont définies sur des ensembles de références X et Y (exemple : le nombre de page est petit et le prix est élevé) et elle est associée au produit cartésien $A \times B$ caractérisant (V, W) sur l'ensemble $X \times Y$.
Sa valeur de vérité est définie par .Généralement, on peut construire des propositions floues par $min(f_A(X), f_B(Y))$. conjonction, disjonction ou implication portant sur des propositions floues quelconques.

3.7 Système flou

Généralement, un système à base de règles floues se compose de trois modules successifs [148] :

- Un module de fuzzification et calcul des propositions élémentaires.
- Un module d'inférence floue (base de règles et moteur d'inférence).

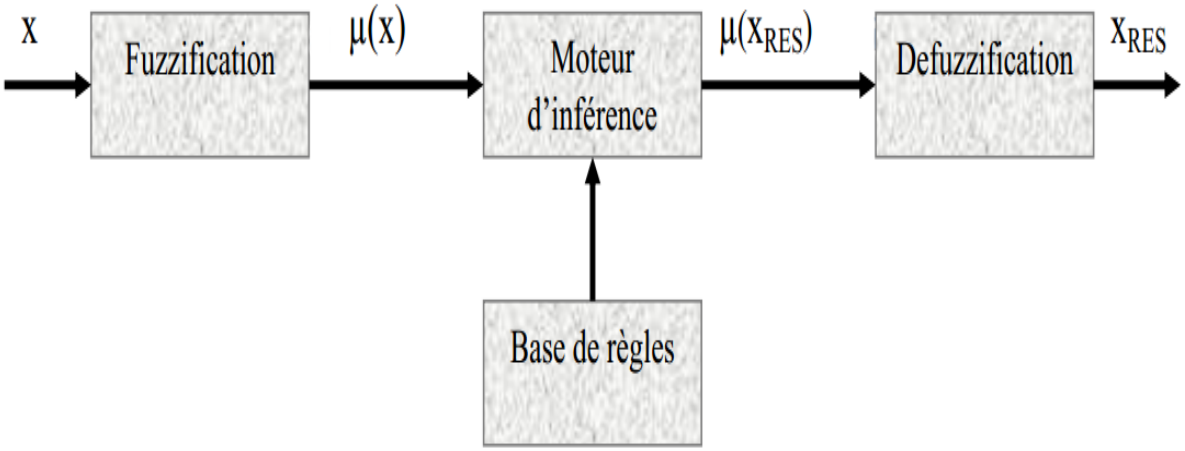

FIGURE 3.9 – Schéma d'un système floue

Où x représente le vecteur des entrées, X_{Res}celui des commandes, $\mu(X)$ et $\mu(X_{Res})$les fonctions d'appartenance correspondantes.

- Le premier module traite les entrées du système, on définit tout d'abord un univers de discours, un partitionnement de cet univers en classes pour chaque entrée, et des fonctions d'appartenance pour chacune de ces entrées (exemple : pression grande, petite, faible). La première étape, est appelée fuzzification, consiste à attribuer à la valeur réelle de chaque entrée, au temps t, sa fonction d'appartenance à chacune des classes préalablement définies, donc à transformer l'entrée réelle en un sous-ensemble flou.
- Le deuxième module consiste en l'application de règles de type « si l'écart de température est grand, diminué le débit du fuel ». Ces règles permettent de passer d'un degré d'appartenance d'une grandeur réglant aux degrés d'appartenance d'une commande. Ce module est constitué d'une base de règles et d'un moteur d'inférence qui permet le calcule.
- Le troisième et le dernier module décrit l'étape de deffuzzification qui est la

transformation inverse de la première. Il permet de passer d'un degré d'appartenance d'une commande à la détermination de la valeur à donner à cette commande.

3.7.1 Fuzzification ou quantification floue

Cette première étape consiste à déterminer le degré d'appartenance de chaque variable d'entre a chaque état. Celui-ci est déterminé à l'aide des fonctions d'appartenance définies dans le système.
Pour fuzzifier, il faut donner :

- L'univers du discours i.e. : Plage de variations possibles de l'entrée considérée.
- Une partition en classe floue de cet univers.
- Les fonctions d'appartenances de chacune de ces classes.

Remarque :

- Il faut fuzzifier les entrées et les sorties du processus flou.
- La fuzzification des variables est une phase délicate du processus mis en oeuvre par la logique floue. Elle est souvent réalisée de manière itérative et requiert de l'expérience.

3.7.2 Inférence

Opération logique par laquelle on admet une proposition en vertu de sa liaison avec d'autres propositions tenues pour vraies, par exemple :

$$\boxed{Si\,(X est A)\,Alor\,(Y est B)}$$

∗ La variable floue X appartient à la classe floue A avec un degré de validité μ_{X_0}.

∗ La variable floue Y appartient à la classe floue B à un degré qui dépend du degré de validité μ_{X_0} de la prémisse.
Pour l'inférence :
- Détermination du degré d'appartenance de chacune des conditions des règles.
- Activation de la règle, détermination de la conséquence (min).
- Agrégation des règles (max).

– La méthode choisie à peu d'influence sur le résultat (trois méthodes d'inférence : méthode de Mamdani, méthode de Larsen et méthode de Takagi-Segeno)

3.7.3 La défuzzification

L'objectif de la défuzzification est de transformer un ensemble flou en une valeur exacte. Soit C un ensemble flou, et defuzz l'opérateur de défuzzification :
$Z_0 = defuzz(C)$, est une valeur précise.
Les opérateurs de défuzzification sont nombreux, citons par exemple :

La méthode du maximum

Elle consiste à choisir comme solution défuzziée l'abscisse du maximum de la Fonction d'appartenance des solutions.
Si plusieurs points conviennent, on peut par exemple utiliser une variante, la méthode de la moyenne des maxima, qui consiste à prendre comme solution la moyenne des abscisses des maxima.

$$\boxed{Z_U = \frac{\int_G w.dw}{\int_G dw} \text{ , } G = \{g \in Z / \mu_c(g) = max_{z \in Z}(\mu_c(Z))\}}$$

Ou

$$\boxed{Z_U = \frac{1}{N}\sum_{i=1}^{n} Z_i \text{ , ou } z_i \in Z \text{ et } max_{z_j \in Z}(\mu_c(Z_j))}$$

L'avantage de cette méthode est qu'elle ne nécessite pas une grande puissance de calcul.

Le centre de gravité

Cette méthode est la plus souvent utilisée et donne généralement les meilleurs résultats. Elle consiste à prendre comme solution l'abscisse du centre de gravité des

solutions.

$$\boxed{Z_U = \frac{\int_Z w.\mu_c(w).dw}{\int_Z \mu_c(w).dw} \text{ Ou } Z_U = \frac{\int_Z w_i \mu_c(w_i)}{\int_Z \mu_c(w_i)}}$$

Par comparaison avec la méthode du maximum, elle exige une plus grande puissance de calcul [149].

3.8 Modèles flous

La description des systèmes suffisamment complexes est faite au moyen d'un traitement approprié de l'information, basé sur des règles «If-Then», de certaine admission d'incertitude et/ou d'imperfection et d'imitation des mécanismes d'apprentissage (appliquer avec succès par l'être humain). [150].

3.8.1 Définition d'un modèle flou

Un modèle flou est un ensemble $\Re$, de règles floues reproduisant approximativement la relation existant entre les données en entrée et les données en sortie d'un système.
Un modèle flou $\Re$ est dit utile, si les données en entrée et les données en sortie sont représentatives et si $\Re(xK) \approx yk$ tel que K=1. N.. [151].
La modélisation floue de systèmes se divise typiquement en deux catégories qui différent dans leurs capacités de représenter différents types d'informations : le modèle de MamdaniAsilian(MA) ou le modèle linguistique, basé sur l'expérience et le modèle de TakagiSugeno(TS), plus approprié pour une approche basée sur les données. Pour chaque modèle on présente la structure des règles, la méthode d'inférence et de deffuzzification.

3.8.2 Modèle de mamdani-assilian (MA)

Le modèle MA a été introduit comme un moyen pour capturer les connaissances qualitatives (ou semi-qualitatives) disponibles, en forme de règles «If-Then» [150].
Etant données :

L'ensembles en entrée $X = \{x_1, x_2, .., x_n\} \in RP$, l'ensemble en sortie $Y = \{y_1, y_2, .., y_k\} \in Rq$, l'ensemble $Z = \{z_1, z_2, .., z_n\} \subset Rp + q$ tel que z_i est une concaténation entre un vecteur d'entrée et un vecteur de sortie, et le nombre de règles c. Le modèle MA peut est défini comme suit :
Chaque règle R_i peut être vue comme une relation floue, elle est de la forme :

$$\boxed{R_i : If \bigwedge_{\ell=1..P} \mu_{i\ell}(x^{(\ell)}) Then\, v_{im}(y^{(m)}) = \mu_{(p+m)}(y^{(m)}); \forall m = 1, .., q.}$$

- $X = (X_{(1)}, X_{(2)}, .., X_{(p)})$: est le vecteur en entrée du système.
- $Y = (Y_{(1)}, Y_{(2)}, .., Y_{(q)})$:est le vecteur en sortie du système.
- $\mu_{i\ell}, v_{im} : R \longrightarrow [0, 1]; telque : i = 1, .., c; \ell = 1, .., p + q; m = 1, .., q.$
- $\mu_{i\ell}(X(\ell))$: est appelée la fonction d'appartenance précédente ; $i = 1, .., c; \ell = 1, p.$
- $\mu_i(p + m)(Y(m))$: est appelée la fonction d'appartenance conséquente ; $i = 1, .., c; m = 1, q.$
- $\bigwedge : [0, 1]_2 \longrightarrow [0, 1]$ est une T-norme (Conjonction floue).

Chaque $v_{im}(Y(m)), i = 1, .., c; m = 1, ..., q;$ représente la sortie floue de R_i pour la composante de sortie $y\ell$, elle est calculée comme :
$v_{im}(y^{(m)}) = \mu_{i(p+m)}(y^{(m)}) \bigwedge \bigwedge_{\ell=1...P} \mu_{i\ell}(x^{(\ell)}), m = 1, .., q.$
Les composantes en sortie de tout le système sont calculées comme :

$$\boxed{y^{(m)} = d(\bigvee_{i=1,...,c} v_{im}(y^{(m)}), m) = 1, q.}$$

- $d : (\{\mu : R \longrightarrow [1, 0]\}) \longrightarrow R$ est une deffuzzification.
- $\bigvee : [1, 0]_2 \longrightarrow [1, 0]$ est une T-conorme (Disjonction floue). L e modèle flou MA est un modèle qui a une large acceptation, intuitif et exprime bien les données en entrée fournies par l'être humain (généralement l'expert). [152].

3.8.3 Le modèle de Takagi-sugeno

Le modèle TS est basé sur la méthode de raisonnement TS qui a été proposée par Takagi et sugeno en 1985 [153]. Ce type de modèle est formé par des règles logiques qui ont un antécédent flou et un conséquent qui est une fonction concrète des variables qui interviennent dans l'antécédent. Il est défini comme suit [151] :

Chaque règle est de la forme :

$$R_i : If \bigwedge_{\ell.....p} \mu_{i\ell}(X^{(\ell)}) Then Y = f_i(X), tel que , f_i : R \longrightarrow R, i = 1, .., c.$$

Les composantes en sortie de tout le système sont calculées comme :

$$y^{(m)} = \frac{\sum_{i=1}^{c} \bigwedge_{l=1}^{p} (\mu_{il}(x^{(l)})) \int_i *(x)}{\sum_{i=1}^{c} \bigwedge_{l=1}^{p} \mu_{il}(x^{(l)})}$$

Modèle peut être appliqué pour les systèmes statiques et dynamiques multi entrée et multi sortie, permet une application relativement facile des techniques d'identification à partir des données, convenables pour les analyses mathématiques et fonctionne mieux avec les techniques adaptatives et celle d'optimisation. [152]

3.9 Caractéristiques, avantages et limitations de la logique floue

3.9.1 Caractéristiques

- Les calculs sont numériques.

3.9.2 Avantages

- Facilité de construction et d'interprétation des règles : les règles sont formulées de manière naturelle par les experts, comme en symbolique.
- Interprétation numérique entre les règles : lorsque deux règles sont activées en même temps car leurs prémisses sont en partie vérifiées, alors la conclusion proposée du système peut prendre une valeur intermédiaire entre les conclusions proposées par celle-ci.
- Intégration de la connaissance à priori.
- Robustesse vis à vis des incertitudes.

3.9.3 Limitations

- Construction manuelle de règle suivant l'intuition de l'opérateur ou les connaissancesde l'expert.
- Optimisation manuelle des fonctions d'appartenance : nombre de termes linguistiques, choix de la forme des fonctions d'appartenance (triangle, trapèze,..), position du centre, largeur.

3.10 Conclusion

La théorie des sous-ensembles fous a été largement utilisée dans déffèrent domaine de raisonnement de sa capacité à représenter des connaissances imprécises. En revanche, La logique floue ouvre des possibilités remarquables de codification des connaissances des experts. Cependant, les applications utilisant la logique floue ne sont pas fondamentalement plus performantes. Elles sont tout simplement plus faciles à réaliser et à utiliser : l'utilisation faite par la logique floue d'expressions du langage courant permet au système flou de rester compréhensible pour les personnes non expertes. C'est ainsi que des machines complexe peuvent devenir plus convivial grâce à l'utilisation de la logique floue. Malheureusement la manipulation de règles non précises peut générer un nombre d'erreurs non négligeable. La mise en place d'un système floue nécessite donc une attention particulière lors de la phase de test de manière à détecter les éventuelles aberrations du système.

Chapitre 4

Implémentation

Sommaire

4.1 Introduction :

Les travaux que nous avons réalisés se basent en particulier sur les systèmes de recommandation, mais certains présentant des méthodes de clustering peuvent être utilisés dans plusieurs domaines. Pour valider nos résultats, nous présentons la base et la structure des données utilisées, les métriques d'évaluation méthodes de recommandation qui ont été implémentés avec les algorithmes cités auparavant ; à savoir l'algorithme de filtrage collaboratif centré utilisateur et centré item qui ont été implémenter aves l'Api Java de Apache Mahout et la méthode de Recommandation floue qui a été implémenter avec la fameuse méthode de clustring C-Means Floue .

4.2 Le Langage de programmation :

4.2.1 Java :

Le langage Java est un langage de programmation informatique orienté objet créé par James Gosling et Patrick Naughton, employés de Sun Microsystems, avec le soutien de Bill Joy (cofondateur de Sun Microsystems en 1982), présenté officiellement le 23 mai 1995 au SunWorld.
La société Sun a été ensuite rachetée en 2009 par la société Oracle qui détient et maintient désormais Java. La particularité et l'objectif central de Java est que les logiciels écrits dans ce langage doivent être très facilement portables sur plusieurs systèmes tels que UNIX, Windows, Mac OS ou GNU/Linux, avec peu ou pas de modifications. Pour cela, divers plate- forme set Framework associés visent à guider, sinon garantir, cette portabilité des applications développées en Java.

4.2.2 Aperçu :

Le langage Java reprend en grande partie la syntaxe du langage C++, très utilisée par les informaticiens. Néanmoins, Java a été épuré des concepts les plus subtils du C++ et à la fois les plus déroutants, tels que les pointeurs et références, et l'héritage multiple remplacé par l'implémentation des interfaces. Les concepteurs ont privilégié l'approche orientée objet de sorte qu'en Java, tout est objet à l'exception des types primitifs (nombres entiers, nombres à virgule flottante, etc.) Java permet

de développer des applications client-serveur. Côté client, les applets sont à l'origine de la notoriété du langage. C'est surtout côté serveur que Java s'est imposé dans le milieu de l'entreprise grâce aux servlets, le pendant serveur des applets, et plus récemment les JSP (JavaServer Pages) qui peuvent se substituer à PHP, ASP et ASP.NET.
Java a donné naissance à un système d'exploitation (JavaOS),à des environnements de développement (eclipse/JDK), des machines virtuelles (MSJVM, JRE) applicatives multiplateformes (JVM), une déclinaison pour les périphériques mobiles/embarqués (J2ME), une bibliothèque de conception d'interface graphique (AWT/Swing), des applications lourdes (Jude, Oracle SQL Worksheet, etc.), des technologies web (servlets, applets) et une déclinaison pour l'entreprise (J2EE). La portabilité du bytecode Java est assurée par la machine virtuelle Java, et éventuellement par des bibliothèques standard incluses dans un JRE. Cette machine virtuelle peut interpréter le bytecode ou le compiler à la volée en langage machine. La portabilité est dépendante de la qualité de portage des JVM sur chaque OS.

4.2.3 Lancement :

En octobre 1994, HotJava et la plate-forme Java furent présentés pour Sun Exécutives. Java 1.0a fut disponible en téléchargement en 1994 mais la première version publique du navigateur HotJava arriva le 23 mai 1995 à la conférence SunWorld.
L'annonce fut effectuée par John Gage, le directeur scientifique de Sun Microsystems. Son annonce fut accompagnée de l'annonce surprise de Marc Andressen, vice-président de l'exécutif de Netscape que Netscape allait inclure le support de Java dans ses navigateurs. Le 9 janvier 1996, le groupe Javasoft fut constitué par Sun Microsystems pour développer cette technologie3. Deux semaines plus tard la première version de Java était disponible.
La société Oracle a acquis en 2009 l'entreprise Sun Microsystems. On peut désormais voir apparaître le logo Oracle dans les documentations de l'api Java. Le 12 avril 2010, James Gosling, le créateur du langage de programmation Java démissionne d'Oracle pour des motifs qu'il ne souhaite pas divulguer. Il était devenu le directeur technologique de la division logicielle client pour Oracle.

4.2.4 Indépendance vis-à-vis de la plate-forme :

L'indépendance vis-à-vis de la plate-forme signifie que les programmes écrits en Java fonctionnent de manière parfaitement similaire sur différentes architectures matérielles. La licence de Sun pour Java insiste ainsi sur le fait que toutes les implémentations doivent être compatibles. On peut ainsi théoriquement effectuer le développement sur une architecture donnée et faire tourner l'application finale sur toutes les autres.
Ce résultat est obtenu par :

- des bibliothèques standard fournies pour pouvoir accéder à certains éléments de la machine hôte (le graphisme, le multithreading, la programmation réseau ..) exactement de la même manière sur toutes les architectures.

- des compilateurs Java qui compilent le code source «à moitié» afin d'obtenir un bytecode (plus précisément le bytecode Java, un langage de type assembleur, proche de la machine virtuelle et spécifique à la plate-forme Java).

Ce bytecode a ensuite vocation à être interprété sur une machine virtuelle Java (JVM en anglais), un programme écrit spécifiquement pour la machine cible qui interprète le bytecode Java et fait exécuter par la machine les instructions traduites en code natif.
Noter que même s'il y a explicitement une première phase de compilation, le bytecode Java est soit interprété, soit converti à la volée en code natif par un compilateur à la volée (just in time, JIT).

4.2.5 Types de compilations :

Les premières implémentations du langage utilisaient une machine virtuelle interprétée pour obtenir la portabilité. Ces implémentations produisaient des programmes qui s'exécutaient plus lentement que ceux écrits en langage compilé (C, C++, etc.) si bien que le langage souffrit d'une réputation de faibles performances. Des implémentations plus récentes de la machine virtuelle Java (JVM) produisent des programmes beaucoup plus rapides qu'auparavant, en utilisant différentes techniques :

- La première technique est de compiler directement en code natif comme un compilateur traditionnel, supprimant complètement la phase de bytecode. Des compilateurs Java tels que GNU Compiler for Java (GCJ) compilent ainsi directement le Java en code objet natif pour la machine cible. On obtient ainsi de bonnes performances, mais aux dépens de la portabilité : le code final produit par ces compilateurs ne peut de ce fait être exécuté que sur une seule architecture.
- Une autre technique appelée compilation 'juste-à-temps', ou 'à la volée' (just in time, JIT) traduit le byte code en code natif durant la phase de lancement du programme.
- Certaines machines virtuelles plus sophistiquées utilisent une recompilation dynamique durant laquelle la machine virtuelle analyse le comportement du programme et en recompile sélectivement certaines parties. La recompilation dynamique permet d'obtenir de meilleurs résultats que la compilation statique car les compilateurs dynamiques peuvent optimiser en fonction de leur connaissance de l'environnement cible et des classes qui sont utilisées. La compilation JIT et la recompilation dynamique permettent à Java de tirer profit de la rapidité du code natif sans perdre la portabilité.

4.2.6 Portabilité de java :

Après que Sun a constaté que l'implémentation de Microsoft ne supportait pas les interfaces RMI et JNI, et comportait des éléments spécifiques à certaines plates-formes par rapport à sa plate-forme initiale, Sun déposa plainte en justice contre Microsoft en 32, et obtint des dommages et intérêt (20 millions de dollars). Cet acte de justice renforça encore les termes de la licence de Sun. En réponse, Microsoft arrêta le support de Java sur ses plates-formes et, sur les versions récentes de Windows, Internet Explorer ne supporte pas les applets Java sans ajouter de plug-in. Cependant, Sun met à disposition gratuitement des environnements d'exécution de Java pour les différentes plates-formes Microsoft.
La portabilité est techniquement un objectif difficile à atteindre et le succès de Java en ce domaine est mitigé. Quoiqu'il soit effectivement possible d'écrire des programmes pour la plate-forme Java qui fonctionnent correctement sur beaucoup de machines cibles, le nombre important de plates-formes avec de petites erreurs et des incohérences a abouti à un détournement du slogan de Sun « Write once, run anywhere » (« Écrire une fois, exécuter partout ») en « Writ0e once, debug everywhere »

(« Écrire une fois, déboguer partout ») !
L'indépendance de Java vis-à-vis de la plate-forme est cependant un succès avec les applications côté serveur comme les services web, les servlets et le Java Beans aussi bien que les systèmes embarqués sur OSGi, utilisant l'environnement Embedded Java.

4.2.7 Exécution sécurisée de code distant :

La plate-forme Java fut l'un des premiers systèmes à offrir le support de l'exécution du code à partir de sources distantes. Une applet peut fonctionner dans le navigateur web d'un utilisateur, exécutant du code téléchargé d'un serveur HTTP. Le code d'une applet fonctionne dans un espace très restrictif, ce qui protège l'utilisateur des codes erronés ou mal intentionnés. Cet espace est délimité par un objet appelé gestionnaire de sécurité. Un tel objet existe aussi pour du code local, mais il est alors par défaut inactif.
Le gestionnaire de sécurité (la classe SecurityManager) permet de définir un certain nombre d'autorisations d'utilisation des ressources du système local (système de fichiers, réseau, propriétés système, ..). Une autorisation définit :

- un code accesseur (typiquement, une applet - éventuellement signée - envoyée depuis un serveur web)
- une ressource locale concernée (par exemple un répertoire)
- un ensemble de droits (par exemple lire/écrire)

Les éditeurs d'applet peuvent demander un certificat pour leur permettre de signer numériquement une applet comme sûre, leur donnant ainsi potentiellement (moyennant l'autorisation adéquate) la permission de sortir de l'espace restrictif et d'accéder aux ressources du système local.
La programmation peut se faire depuis une invite de commandes en lançant un compilateur Java (souvent nommé javac), mais pour avoir plus de confort, il est préférable d'utiliser un environnement ou IDE, certains sont gratuits. On peut citer :

- Eclipse
- Idea
- JBuilder

- JCreator
- jDeveloper
- NetBeans
- Xcode

4.3 Les outils de développement

4.3.1 NetBeans

NetBeans est un environnement de développement intégré IDE pour Java, placé en open source par Sun. En plus de Java, NetBeans permet également de supporter diérents autres langages, comme C, C++, XML et HTML. Il comprend toutes les caractéristiques d'un IDE moderne (éditer en couleur, projets multi-langage, éditeur graphiques d'interfaces et de pages web).
NetBeans est disponible sous Windows, Linux et d'autres systèmes d'exploitation. Il est lui-même développé en Java, ce qui peut le rendre assez lent et gourmand en ressources mémoires voire la figure 4.1

FIGURE 4.1 – NetBeans 8.1

4.3.2 Apache mahout

Nous avant utilisé apache mahout pour construire les 2 premier systèmes de recommandation de Filtrage collaboratif centré utilisateur et le deuxième centré

item , L'idée est de mettre en place des algorithmes de ML et plus particulièrement de filtrage collaboratif (Collaborative Filtering (CF)) afin de pouvoir utiliser les données des clients pour recommander à un client spécifique des films en adéquation avec ce qui est susceptible de l'intéresser.

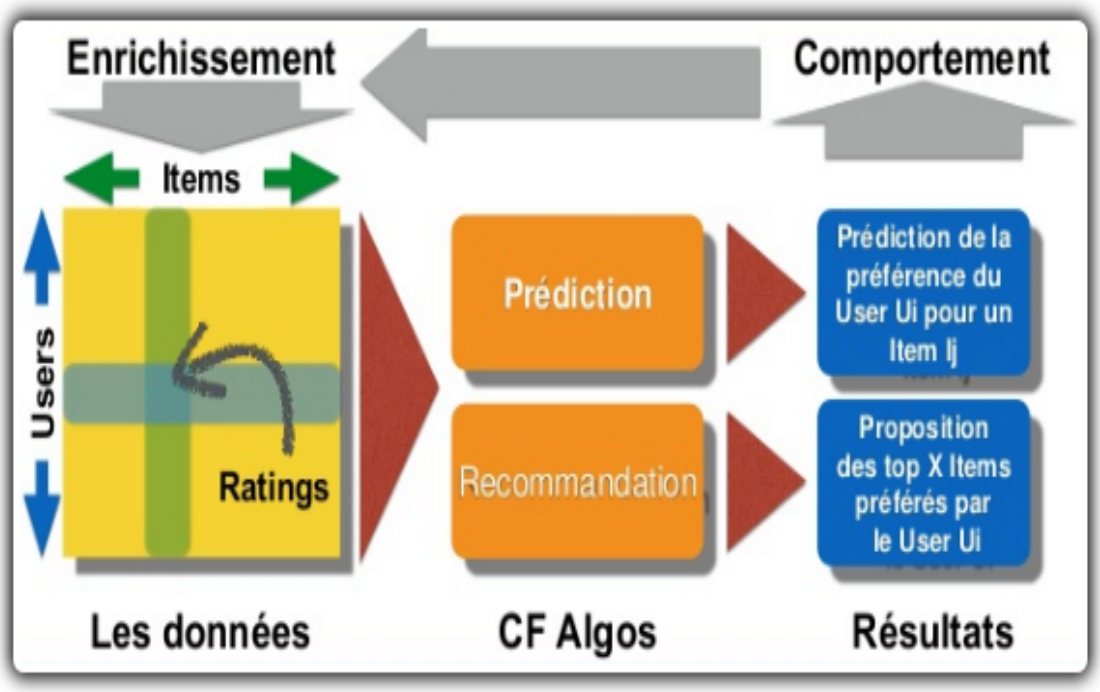

FIGURE 4.2 – Processus de recommandation présenté par Apache Mahout

Idéalement, ces recommandations personnalisées peuvent être générées en temps réel.
Note : une fois recommandé, le client va lui-même participer au processus d'enrichissement des données en fournissant ses propres évaluations.
Aujourd'hui sur le web, de nombreux sites utilisent des algorithmes de recommandation pour enrichir l'expérience utilisateur avec des fonctionnalités « intelligentes » :

- **sites e-commerce :** Amazon, eBay, Fnac
- **réseaux sociaux :** Facebook, Twitter, Viadeo, LinkedIn, Foursquare
- **plateformes multimédia :** Deezer, IMDb, Netflix, Spotify, Flickr.

Ainsi une étude a révélé que près de 30 % du CA total d'Amazon était généré grâce à son système de recommandation, les utilisateurs naviguant sur le site au gré des recommandations basées sur les achats réalisés précédemment par des acheteurs

intéressés par des produits similaires.

Les moteurs de recherche s'enrichissent eux aussi de fonctionnalités de Machine Learning pour vous proposer des résultats de recherche de plus en plus pertinents et personnalisés.
Toutes ces sociétés contribuent souvent activement aux développements des outils de Machine Learning en reversant leurs propres algorithmes à la communauté open source. Différents types d'outils de Machine Learning peuvent être utilisés :

- outils mathématiques : Matlab, Mathematica, Maple ;
- logiciels d'analyses statistiques : Scilab, SPSS, Stata ;
- bibliothèques open source :
  - Python : Pandas, StatsModels, scikit-learn
  - Java : Weka, Apache Mahout

Si pour quelques Ko ou Mo de données, de simples outils mathématiques permettent aisément de réaliser du prototypage, il en va tout autrement avec des Go ou des To de données.
les systèmes de recommandation doivent s'appuyer sur des systèmes distribués, tout en utilisant des algorithmes de Machine Learning eux-mêmes distribués.
Pour moi, divers arguments plaident en faveur de l'utilisation d'Apache Mahout pour répondre à notre besoin de mise en place d'un moteur de recommandation :

- bibliothèque complète de Machine Learning qui couvre les principaux algorithmes, notamment les 3C
  - Collaborative Filtering : algorithmes permettant de recommander des items à un utilisateur en identifiant des similarités

  - Clustering : algorithmes permettant de découvrir des groupements parmi un ensemble de données (ex : profilage d'utilisateurs, analyse de tendances)
  - Classification : algorithmes permettant de classifier automatiquement des documents à partir de documents déjà classifiés (ex : association automatique de tags à des documents, filtrage des mails de type spams)
- algorithmes bien testés et supportés
- open source (licence Apache)
- s'appuie sur Hadoop MapReduce, ce qui répond à des problématiques de scalabilité (stockage réparti redondant) et de **parallélisme** (calculs distribués de type MapReduce) pour être utilisé avec de gros volumes de données
- communauté active et dynamique qui assure une croissance
- Java
- bibliothèque extensible avec la possibilité d'ajout de nouvelles collections d'algorithmes

Aujourd'hui, Mahout est en version 0.11 (stable) et le projet s'oriente vers la mise en place d'un DSL Scala pour l'écriture des algorithmes ainsi que vers l'utilisation d'Apache Spark offrant des performances de calcul bien plus importantes que celles d'Hadoop MapReduce, ce qui devrait être très prometteur pour l'avenir de Mahout.

Mise en oeuvre du filtrage collaboratif (CF) avec Mahout

Rentrons maintenant dans le vif du sujet en utilisant Apache Mahout pour mettre en place notre Système de recommandation (Pour le filtrage collaboratif seulment). Le **DataModel** de Mahout s'appuie sur différentes données :

- les Users modélisant les utilisateurs ayant noté les films
- les Items qui correspondent aux différents films
- les Ratings qui peuvent être des notations, des visualisations des films , des achats concrets

Ces données peuvent être stockées dans des bases de type relationnel ou NoSQL et de volumétrie variable. Dans notre cas on utilisons qu'un fichier texte qui a 100 mille votes.

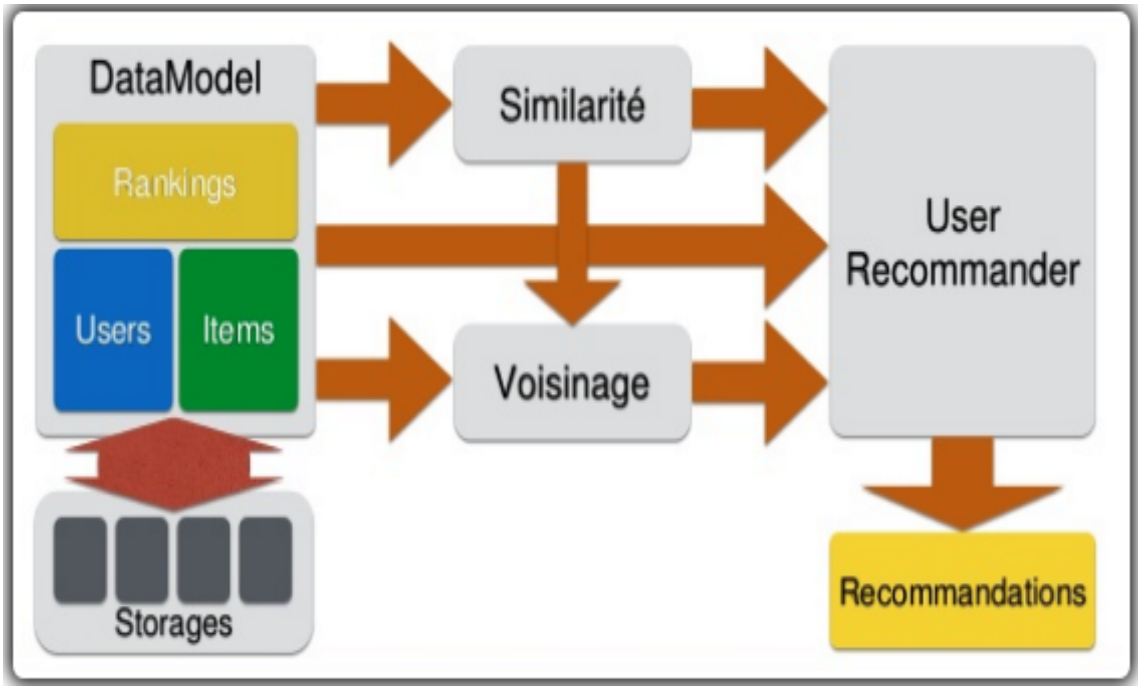

FIGURE 4.3 – FC Centré utilisateur

Pour réaliser ses recommandations, Apache Mahout s'appuie ensuite sur :

- des algorithmes de **similarité** pour déterminer les utilisateurs les plus proches en utilisant par exemple la , **la corrélation de Pearson**, **la similarité cosinus**
- des algorithmes de voisinage pour déterminer un ensemble d'utilisateurs proches selon la règle de similarité choisie. On distingue des algorithmes de type Nearest (les X utilisateurs
- les plus similaires) ou Threshold (tous les utilisateurs dépassant un certain seuil de similarité)

Outre le filtrage collaboratif basé sur les utilisateurs (User Recommender) vu ci-dessus, il est également possible de mettre en place des moteurs de recommandation se basant uniquement sur les items (Item Recommender) et les similarités entre ceux-ci.

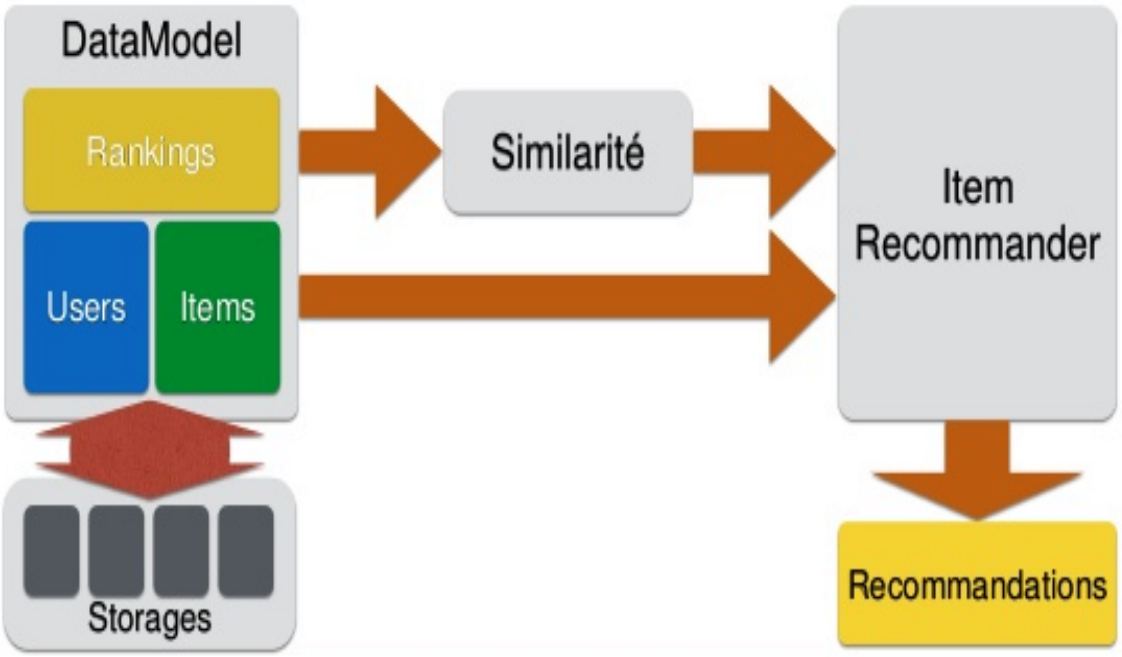

FIGURE 4.4 – FC Centré item

4.4 Corpus utilisée :

Afin d'évaluer la performance d'un algorithme de filtrage collaboratif, nous utilisons le jeu de données MovieLens9 fournie par l'équipe américaine de recherche GroupLens10 de l'Université du Minnesota. MovieLens11 est un site web de recommandation de films (https ://movielens.org/) à travers lequel les utilisateurs évaluent d'abord un sousensemble de films qu'ils ont déjà vu. L'application capture les notes de l'utilisateur pour les films et fournit une recommandation formée d'une liste de films. Le jeu de données MovieLens a été largement utilisé par la communauté scientifique pour évaluer et comparer les algorithmes de filtrage collaboratif . Il présente en effet l'avantage de reposer sur des votes réels et fournit de ce fait un bon support de validation. La base de données historique se compose de 100 000 votes (U) de 943 utilisateurs (C), et 1682 films (P), chaque utilisateur a au moins 20 évaluations, ainsi que ses caractéristiques. La matrice de votes présente une dispersion de 6,30 %, c'est-à-dire qu'il y a 93,70 % de données manquantes, considérées comme des non-votes. Aujourd'hui, le site a plus de 45 000 utilisateurs qui ont exprimé des opinions sur les 6600 films différents. Les notes sont sur une échelle Likert avec des valeurs entières comprises entre 1 et 5, avec 1 et 2 représentant les notes négatives,

3, 4 et 5 représentant les avis positifs.

4.5 Mesure de similarité utilisée :

4.5.1 La distance euclidienne :

$$d^2(x_{1i}, x_{2i}) = \sum_i (x_{1i} - x_{2i})^2 = (x_1 - x_2)(x_1 - x_2)`$$

L'algorithme recommandation a base du C-moyenne floue (FCM) :

1. Appliques l'algorithme de C-moyenne floue (FCM) classique cité précédemment dans le chapitre 2 sur l'ensemble des utilisateurs
   - Pour tout utilisateur U faire :
   -
   $$D - A(U_i, C_j) = \frac{d^2(U_i, C_j)}{\sum_{j=1}^{n} d^2(U_i, C_j)}$$

2. Pour l'ensemble des utilisateurs qu'on veux prédire ces notes faire
   -
   $$prd(U_j, item_k) = D - A(U_i, C_j) \times \frac{1}{4} \sum_{i=1}^{m} pp \frac{Note(U_i, item_k)}{N}$$

   d^2 : Distance euclidienne
   $D - A(U_i, C_j)$: Degré d'appartenance de l'utilisateur U_i au cluster C_j
   U_i : Utilisateur i
   C_j : Cluster j
   m : le m utilisateurs les plus proche a l'utilisateur U_i qui ont notés $item_k$
   pp : plus proches

4.6 Utilisation de l'application :

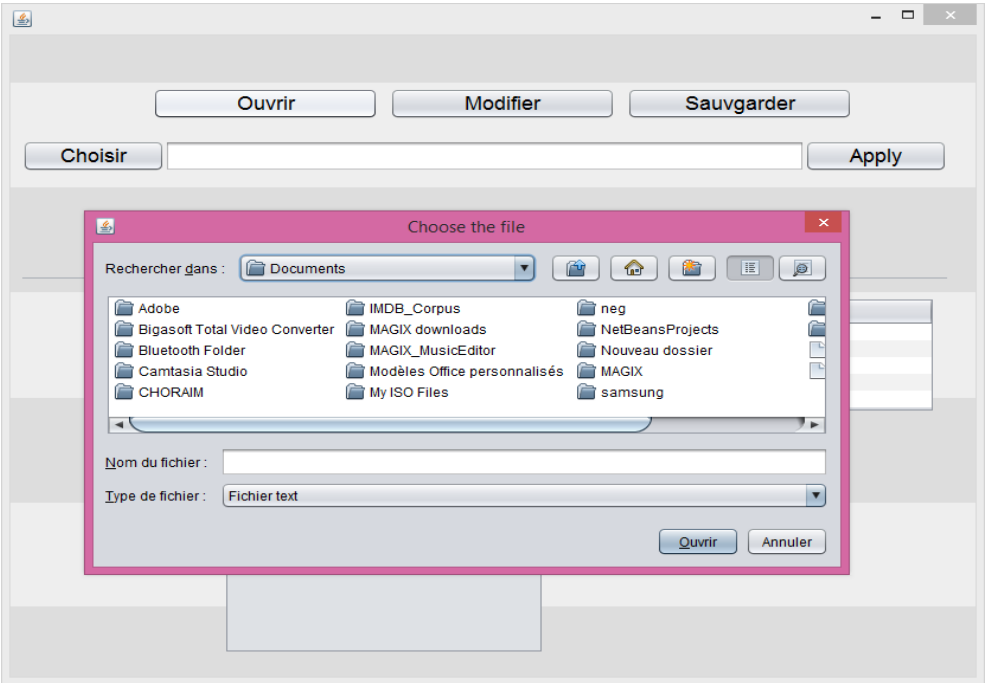

FIGURE 4.5 – Charger le corpus utilisée (Ouvrir)

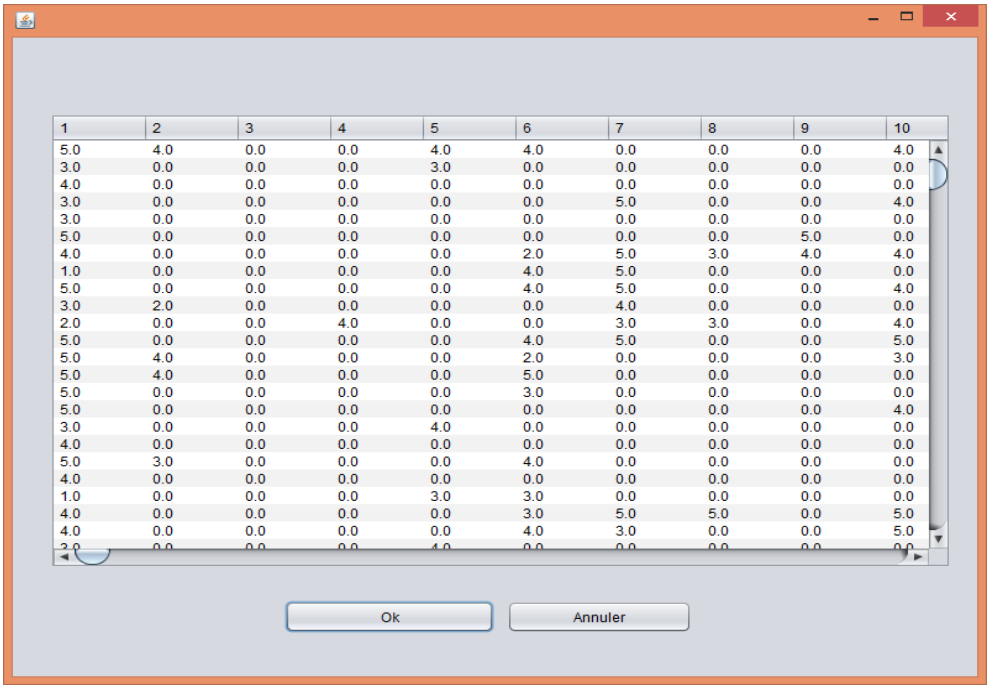

FIGURE 4.6 – Modification et visualisation du corpus

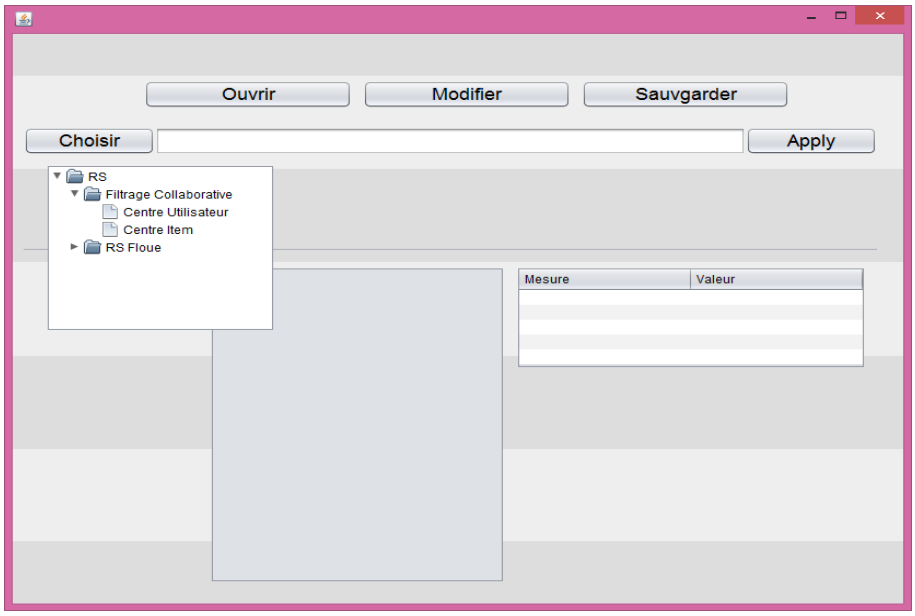

FIGURE 4.7 – Choix de l'algorithme qu'on veut exécuter

Charger les paramètre de l'exécution : Cette option est disponible pour chaque 3 algorithmes implémenté

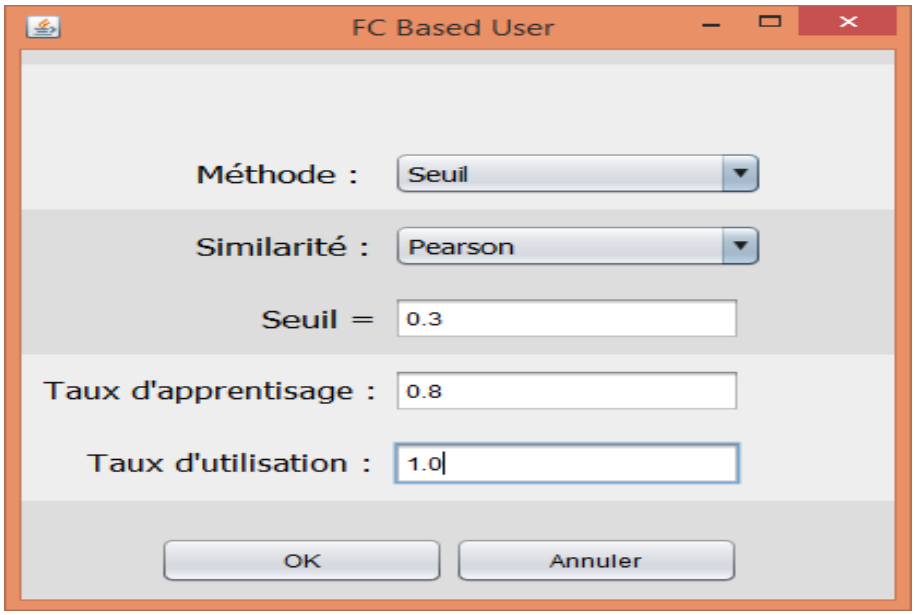

FIGURE 4.8 – les options du FC Centré utilisateur

Méthodes : seuil ou kpp-v

Taux d'apprentissage par exemple dans ce cas est 80 %

Taux d'utilisation c'est le % utilisée du corpus par exemple dans ce cas est 100%

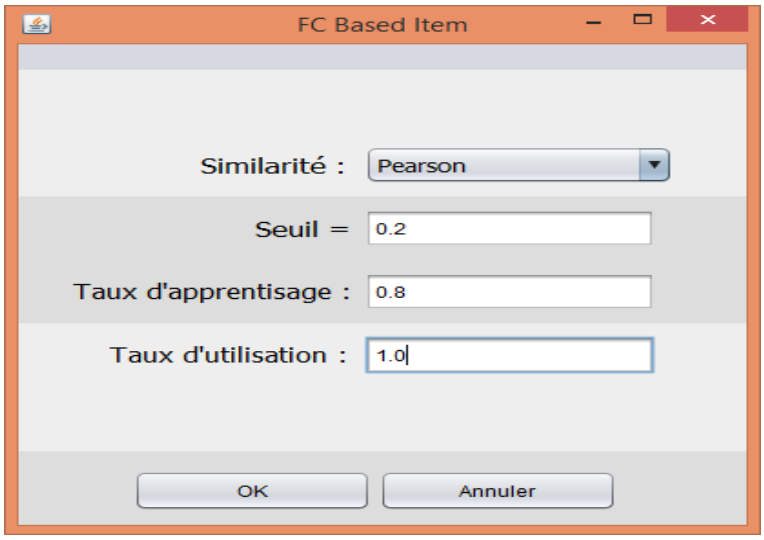

FIGURE 4.9 – les options du FC Centré item

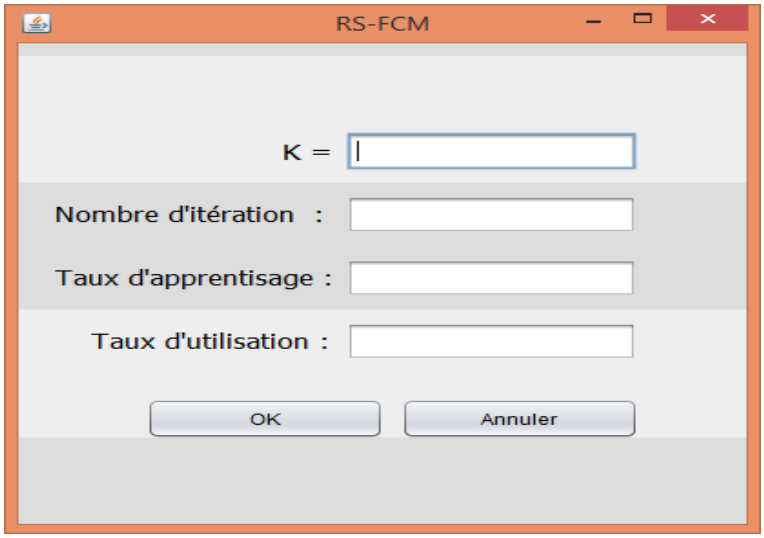

FIGURE 4.10 – pour le 3-éme algorithme RS-FCM

K : est le nombre des cluster

Nombre d'itération : nb itération pour que l'exécution s'arrête

4.7 Les mesure d'évaluation utilisée

MAE : est l'erreur moyenne absolue (MAE) ,c'est une mesure statistique qui s'appuie sur la moyenne des différences entre chaque valeur prédite et sa valeur réelle :

$$MAE = \frac{\sum_{i=1}^{N} |p_i - a_i|}{N}$$

RMSE : est une mesure fréquemment utilisée de la différence entre les valeurs prédites

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (p_i - a_i)^2}{n}}$$

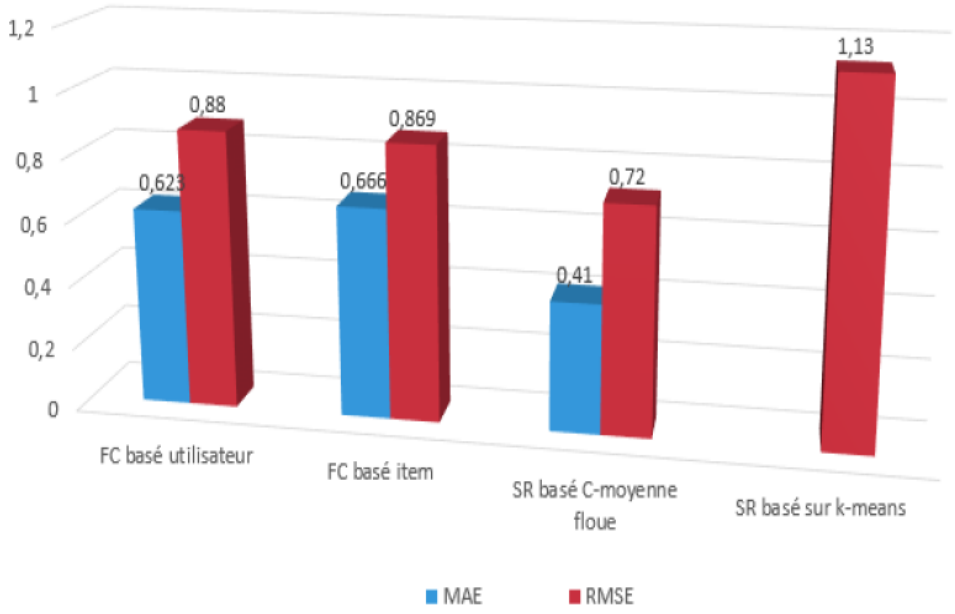

FIGURE 4.11 – Résultat d'évaluation

4.8 Discussion

les résultats précédentes ont montré les taux d'erreurs MAE et RMSE de notre algorithme implémenté sont minimales par rapport aux autres algorithmes telle que les deux algorithmes de filtrage collaboratif basé item et le deuxième basé utilisateur et l'autre algorithme basé sur l'algorithme k-moyenne construit par Urszula Kuzelewska en 2014 (Clustering Algorithms in Hybrid Recommender System on MovieLens Data) ,on remarque très bien l'apport de l'approche floue dans les méthodes de recommandation .

4.9 Conclusion :

Nous avons présenté dans ce chapitre l'approche de Recommandation a base de C-moyenne floue .Cette approche consiste a regrouper les utilisateurs en clusters et faire une prédictions des notes a base des degrés d'appartenances (floue),il existe beaucoup des données manquantes. Notre but est d'améliorer Les systèmes de recommandation par l'un des méthode floue . Avec les résultats expérimentaux, nous avons pu remarquer que l'approche proposée est compétitive comparant à d'autres algorithmes de recommandation telle que le filtrage collaboratif .
Les résultats montrent de cette méthode est satisfaisant . Cette méthode peut être utilisé avec n'importe de données qui sont basée sur les notations ou les votes telle que les pages de Facebook ou les produits du Amazon.com ou eBay pour faire une recommandation satisfaisante .

Conclusion Générale

Dans ce mémoire nous avons propose une méthode de Système de recommendation floues.
Notre travail consistait à améliorer les resultats de recommendation en utilisant la mèthode du clustering C-moyenne floue et quelques adaptation pour choisir les meilleurs utilisateurs qui représentent chaqu'un des clusters .
Nous avons essayé de prouver l'efficacité de cette methode en l'appliquant sur un corpus de Movielens 100 mille votes crée par GroupeLens.
Pour argumenter notre jugement, nous avons utilisé deux mesures d'evaluations RMSE et MAE le résultat obtenue des valeurs de ces deux mesures ont montré l'efficacite de notre méthode par rapport a le filtrage collaborative et la prediction des notes classique. En fait, chaque fois on changeons les parametres on obtient toujours des meilleurs résultats.
Sur le plan personnel, ce travail était une occasion pour apprendre les dif-férents systèmes de recommandation et les différents problèmes rencontré telle que le démarrage à froid et le manque données.

Bibliographie

[1] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, John, andT. Riedl. Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems, 22 :5-53, 2004.

[2] EllenM. Voorhees. The philosophy of information retrieval evaluation. In Carol Peters, Martin Braschler, Julio Gonzalo, and Michael Kluck, editors, Evaluation of CrossLanguage Information Retrieval Systems, volume 2406 of Lecture Notes in Computer Science,pages 355-370. Springer Berlin Heidelberg, 2002.

[3] Guy Shani and Asela Gunawardana. Evaluating recommendation systems. In Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors, Recommender Systems Handbook, pages 257-297. Springer US, January 2011.

[4] Chumki Basu, Haym Hirsh, and William Cohen. Recommendation as classification : Using social and content-based information in recommendation. In Proceedings of the Fifteenth National Conference on Artificial Intelligence, page 714-720. AAAI Press, 1998.

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

[6] Linyuan Lu, Matus Medo, Chi Ho Yeung, Yi-Cheng Zhang, Zi-Ke Zhang, and Tao Zhou. Recommender systems. Physics Reports, 519(1) :1-49, October 2012.

[7] Saporta G, probabilities, analyse des données et statistiques. Technip ,paris,

1990.

[8] J. Hartigans. clustering algorithms. John Wiley and Sons, Inc., 1975.

[9] Kohonen T. self-organized formation of topologically correct feature maps. Biological cybernetics no 43, pp59-69, reprinted in Anderson & Rosenfeld , Eds, Neurocomuting : foundations of research, MIT press, Cambridge Ma, 1988.

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

[11] Lebart L., Morineau A. & piron M.statistique exploratoire multidimensionnelle. Dunod, 3ème édition, paris, 2000.

[12] J. B. MacQueen (1967) : "Some Methods for classification and Analysis of Multivariate Observations, Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability", Berkeley, University of California Press,no 1,pp281-297.1967.

[13] Benzécri J.P. L'analyse des données. Dunod, Paris, 1973.

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

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

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

[17] Negar, Hariri DePaul - Query-Driven Context Aware Recommendation,

Proceedings of the 7th ACM conference on Recommender systems , 2013

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

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

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

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

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

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

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

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

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

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

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

[29] Celeux, G., Diday, E., Govaert, G., Lechevallier, Y., and Ralambondrainy, H. Classification automatique des données, environnement statistique et informatique. DUNOD informatique. 1989.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

[50] www.last.fm,2016

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

[52] S. Castagnos. Modélisation de Comportements et Apprentissage Stochastique non Supervisé de Stratégies d'Interactions Sociales au Sein de Systèmes Temps Réels de Recherche et d'Accès à l'Information. Thèse, IAEM-Lorraine, 2008

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

[54] Ball, G. H. et Hall, D. J. ISODATA, an Iterative Method of Multivariate Analysis and Pattern Recognition. Behavior Science, 153, 1967.

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

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

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

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

[59] J. C. Dunn (1973) : "A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters", Journal of Cybernetics ,no 3, pp 32-57. 1973.

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

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

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

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

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

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

Conference, 1999

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

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

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

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

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

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

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

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

[74] S. Geman and D. Geman, Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images, Transactions on Pattern Analysis and Machine

Intelligence, Vol. 6, N6,p.721-741, 1984

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

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

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

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

[79] AM. Rashid,, S. K. Lam, A. LaPitz, G. Karypis and J. Riedl. Towards a Scalable kNN CF Algorithm :Exploring Effective Applications of Clustering. LNCS Vol 4811, p.147-166,Advances in Proceedings of the Web Mining and Web Usage Analysis ,2007

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Conference on Knowledge Discovery and Data Mining, 2004

[94] www.netflix.com,consulté le 20 Janvier 2016

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

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

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

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

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

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

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

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

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

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

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

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

[107] J. C. Bezdek (1981) : "Pattern Recognition with Fuzzy Objective Function Algoritms", Plenum Press, New York. 1981

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

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

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

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

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

[113] Stéphanie Calpié et Antoine Renard. (2003) Web Services communication inter langage ,Université Claude Bernard Lyon 1, UFR Informatique.6-7,2003

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

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

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

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

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

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

[120] www.imdb.com ,consulté le 23 Janvier 2016

[121] www.ebay.com,consulté le 21 Janvier 2016

[122] www.alibaba.com,consulté le 20 Janvier 2016

[123] www.play.google.com,consulté le 25 Janvier 2016

[124] www.apple.com,consulté le 20 Janvier 2016

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

[126] V. di Gesu. «Mathematical Morphology and Image Analysis : A Fuzzy Approach». Workshop on Knowledge-Based Systems and Models of Logical Reasoning,Reasoning, 1988.

[127] Fairouz Hadi , Khier Benmahammed , Etude comparative entre la morphologie mathématique floue et le regroupement flou , Faculté des Sciences de l'Ingénieur, Université Ferhat Abbas-Sétif, Algérie. 3rd International Conference : SETIT 2005

[128] F. Hoppner, F. Klawonn, R. Kruse, T. Runkler. Fuzzy Cluster Analysis, Methods for classification, data analysis and image recognition. Wiley, 2000.

[129] S. C. Johnson : Hierarchical Clustering Schemes Psychometrika, no 2, pp 241-254, 1967.

[130] Lance, G.N., & Williams, W.T. : A general theory of classificatory sorting strategies : I. Hierarchical systems. Computer Journal, no 9,pp 373-380, 1967.

[131] Kamvar, S. D., Klein, D., & Manning, C. D., Interpreting and Extending Classical Agglomerative Clustering Algorithms using a Model-Based approach. Pp 283-290 of : International Conference on Machine Learning (ICML).2002

[132] Guha, S., Rastogi, R., et Shim, K. CURE : an e-cient clustering algorithm for large databases. Dans Proceedings of ACM SIGMOD International Conference on Management of Data, pp 73-84, 1998.

[133] Karypis, G., Eui-Hong, H., et Kumar, V. Chameleon : Hierarchical Clustering Using Dynamic Modeling. Computer, no 32(8) :68-75, 1999.

[134] Zhang, T., Ramakrishnan, R., et Livny, M. BIRCH : an efficient data clustering method for very large databases. Dans Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data, pp 103-114, 1996.

[135] Bisson, G. , La similarité : une notion symbolique/numérique. Chap. XX of :

Apprentissage symbolique-numérique (tome 2). Editions CEPADUES.2002.

[136] Berrani, S.-A., Amsaleg, L., & Gros, P. Recherche par similarités dans les bases de données multidimensionnelles : panorama des techniques d'indexation. Ingénierie des systèmes d'information (RSTI série ISI-NIS), 7(5-6), pp 65-90.2002.

[137] Gower, J. C. & P. Legendre : Metric and Euclidean properties of dissimilarity coefficients, Journal of Classification, vol 3, pp. 5-48 .1986.

[138] Dalirsefat S, Meyer A, Mirhoseini S. Comparison of similarity coefficients used for cluster analysis with amplified fragment length polymorphism markers in the silkworm, Bombyx mori. Journal of Insect Science 9 :71, available online : insectscience.org/9.71 .2009.

[139] Xindong Wu , vipin Kumar , the top ten Algorithms in Data mining ,chapman & hall/CRC, pp :93-116, 2009.

[140] Ron Kohavi, Roger Longbotham, Dan Sommerfield, and Randal M. Henne. Controlled experiments on the web : survey and practical guide. Data Mining and Knowledge Discovery, 18(1) :140-181, February 2009.

[141] Upendra Shardanand and Pattie Maes. Social information filtering : algorithms for automating & ldquo ;word of mouth & rdquo ;. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 95, page 210-217, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co

[142] S. Andersen and Fenandez-Luna. On the use of weighted mean absolute error in recommender systems.pdf. In RUE 2012 - Workshop on Recommendation Utility Evaluation : Beyond RMSE, Dublin, Ireland, 2012.

[143] Benoit Lavoie , Logique classique et logique floue, 2007.

[144] Olivier Strauss, Serge Guillaume, Sous-ensembles flous, logique floue et systèmes d'inférence floue, 2005

[145] F. Chevrie F. Guély ; la logique floue ; 2006.

[146] Lambalgen : Fuzzy logic, ILLC University of Amesterdam, 2000

[147] J.C. Bezdek : Fuzzy Models What Are They, and Why ? IEEE Transactions on Fuzzy Systems, Vol. 1, No. 1, February 1993

[148] Bernadette, Bouchon Meunier, La logique floue et ses applications, Addison Wesley, 1995.

[149] Sabour Elkosantini, Introduction à la logique floue, les concepts fondamentaux et application, 2011.

[150] V.H. Grisales, A. Gauthier, C.V. Isaza et G.A. Villamarín, Identificación y Modelado Difuso, TS de un Bioreactor Anaerobio, X Congrès Latino américain d'Automatique, Mexique, 2002 (version Française).

[151] Thomas A.Runkler,James C.Bezdek, Alternating Cluster Estimation : A new Tool for Clustering and Function Approximation, IEEE Transaction on Fuzzy Systems, vol 7, $N°$ 4, August 1999.

[152] Matlab 7.1, Fuzzy Logic Toolbox, 2011 Mausumi Acharyya and Malay K. Kundu, Image Segmentation Using Wavelet

[153] T. Takagi, M. Sugeno,Fuzzy Identification of Systems and its Application to Modeling and Control, IEEE Transactions on Systems, Man and Cybernetics, 15, 1985

Résumé

Les systèmes de recommandation sont des outils et des techniques utilisé pour trouver des items convenables et aider les utilisateurs a prendre des décisions on a présenté la c-moyenne floues pour faire des recommandations , dans le premier chapitre on a définie les systèmes de recommandations et ces algorithmes et les problèmes rencontré , dans le deuxième chapitre on a vu la classification non-supervisé (clustering) et ces techniques , le troisième chapitre on a présenté la logique floue qui est l'idée de base du c-moyenne floue enfin on a expliquer l'algorithme Système de recommandation basé sur la c-moyenne floue dans le dernier chapitre.

Mots-clés: Systèmes de recommandation ,C-moyenne floue , Clustering

Abstract

Recommender Systems are software tools and techniques providing suggestions for items to be of use to a user. The suggestions provided are aimed at supporting their users in various decision-making processes we have presented Fuzzy c-means in order to make recommendation, in chapter one we have defined the recommender systems and its various algorithms and its problems ,in the second chapter we have seen clustering and its deferent techniques, the third chapter we have seen fuzzy logic that is the basic idea of the fuzzy c-means, finally we have explained implemented recommender system based on Fuzzy c-means in chapter four.

Keywords: Recommender systems,Fuzzy c-means ,Clustering