

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



Université Ibn Khaldoun

Faculté des Sciences de l'Ingénieur

Département d'Informatique

Tiaret



Mémoire

Présenté pour l'obtention du diplôme de

Magistère - Ecole Doctorale STIC -

Sciences et Technologies de l'Information et de la Communication

Option : Systèmes d'Information et de Connaissance
- SIC -

Par :

M. SAFA Benaïssa

Thème

Formation multicritère d'espaces de communautés
d'utilisateurs pour un système de filtrage
d'information collaboratif

Soutenu devant le jury :

Dr. Y. DAHMANI	Université de Tiaret	Président
Dr. M. BELARBI	Université de Tiaret	Examineur
Dr. M. A. CHIKH	Université de Tlemcen	Examineur
Dr. O. NOUALI	Maître de recherche CERIST	Directeur de mémoire

Remerciements

Cette partie est probablement la plus facile à écrire. Elle commence le mémoire et pourtant elle est écrite en dernier, lorsque celui-ci est terminé.

Je remercie tout d'abord mon directeur de mémoire, le docteur Omar Nouali maître de recherche au C.E.R.I.S.T, qui a su m'accompagner, me diriger, m'orienter et me soutenir durant la préparation de ce mémoire.

Remerciements spéciaux pour les membres du jury, pour l'intérêt qu'ils ont porté au thème du mémoire.

Pour mes études de post graduation, j'ai été accueilli à l'université de Tlemcen, où j'ai trouvé d'excellentes conditions pour suivre mes études. L'ambiance scientifique et humaine, au sein de cette université m'a permis de suivre sereinement mes études. En conséquence, je remercie tous les enseignants de l'université, surtout le docteur Azeddine Chikhi et son frère le docteur Mohamed Amine.

Mes remerciements s'adressent également à mes camarades de mes études de post graduation, surtout le groupe de Tiaret, comme je n'oublie pas celui de Tlemcen. A mes frères et mes amis pour leur encouragement et leur soutien moral

Résumé

Résumé

Dans le contexte des systèmes de filtrage collaboratif, les utilisateurs reçoivent de l'information que leur recommande le système sur la base de leurs profils et/ou de leurs communautés. Donc, l'un des facteurs clés dans cette catégorie de systèmes est le regroupement des utilisateurs en communautés.

La formation en communautés d'utilisateurs dans la plupart de ces systèmes est basée sur un seul critère : la proximité des évaluations des utilisateurs. Est-il raisonnable de limiter cette classification sur un seul critère alors qu'il existe une diversité d'informations qui caractérisent les utilisateurs toutes aussi importantes à prendre en charge dans ce processus ?

La présente recherche a pour objectif d'étudier la formation des espaces de communautés s'appuyant sur une multitude de critères, dans la perspective d'améliorer les systèmes de filtrage. Nous proposons une approche de formation multicritères d'espaces de communautés d'utilisateurs basée sur une méthode de classification hybride. Nous introduisons aussi la notion de priorité entre ces divers critères dans le processus de génération de recommandations.

Mots clés

Système de filtrage d'information, Filtrage collaboratif, Profil utilisateur, Communauté d'utilisateurs, Apprentissage automatique, Classification supervisée et Classification non supervisée.

Abstract

In the context of filtering collaborative, users receives documents recommended on the basis of their profiles and/or communities. Most of them rely on collaborative filtering that compares users on the basis of their ratings in order to group them into communities and produce recommendations with respect to these communities. So, the regrouping of users in communities is one of key factors in this system category.

The objective of this research is the study of community's space creation on the basis on the multi-criteria notion in order to ameliorate the systems filtering.

Keywords

Information filtering systems, Collaborative filtering, User Profile, Users community, Automatic learning, Supervisee and not supervisee classification.

Table des matières

<i>Remerciements</i>	I
<i>Résumé</i>	II
<i>Table des matières</i>	III
<i>Liste des figures</i>	VI
<i>Liste des tableaux</i>	VIII
<i>Liste des algorithmes</i>	VIII
<i>Introduction générale</i>	1
<i>Chapitre I : Contexte de recherche et objectifs</i>	3
1 Présentation du contexte	4
2 Problématique	6
3 Objectifs	7

Partie I : Etat de l'art

<i>Chapitre II : Systèmes de filtrage d'information</i>	9
1 Introduction	10
2 Description générale	10
3 Fonctionnement d'un système de filtrage	11
4 Rôle d'un système de filtrage	12
5 Composantes d'un système de filtrage	13
6 Profil utilisateur	13
6.1 Définition	13
6.2 Modélisation d'un profil utilisateur	14
6.3 Représentation d'un profil utilisateur	15
6.4 Construction et acquisition de profil	15
7 Différentes formes de filtrage d'information	16
7.1 Filtrage basé sur le contenu « cognitif »	16
7.2 Filtrage collaboratif « social »	18
7.3 Filtrage hybride	18
8 Recherche d'information versus Filtrage d'information	18
8.1 Recherche d'information (RI)	18
8.2 Filtrage d'information (FI)	19
9 Conclusion	19

<i>Chapitre III : Systèmes de filtrage d'information collaboratif</i> -----	20
1 Introduction-----	21
2 Principe général-----	21
3 Architecture générale-----	22
4 Calcul de la prédiction-----	22
4.1 Algorithmes basés « mémoire »-----	23
4.2 Algorithmes basés « modèle »-----	24
4.3 Algorithmes d'apprentissage en ligne-----	25
5 Avantages et inconvénients-----	25
5.1 Avantages-----	25
5.2 Inconvénients-----	25
6 Complémentarités entre approches : collaborative et basée contenu-----	26
7 Conclusion-----	26
<i>Chapitre IV : Méthodes de formation des communautés</i> -----	28
1 Introduction-----	29
2 Quelques notions de base-----	30
3 Méthodes de classification supervisées-----	32
3.1 Approche des k-voisins les plus proches « <i>K-NN</i> »-----	32
3.2 Arbres de Décision « <i>Decision Trees</i> »-----	34
3.3 Approche probabiliste-----	36
3.4 Réseaux de neurones-----	37
3.5 Machines à vecteurs supports-----	42
4 Méthodes de classification non supervisées-----	45
4.1 Réseaux sociaux-----	45
4.2 Classification Ascendante Hiérarchique « <i>CAH</i> »-----	46
4.3 Algorithme des k-moyennes « <i>k-means</i> »-----	46
4.4 Algorithme des C-moyennes floues « <i>Fuzzy C-means</i> »-----	47
4.5 Mémoires hétéro-associatives : (<i>cartes auto-organisatrices de Kohonen</i>)-----	49
5 Conclusion-----	50

Partie II : Proposition et évaluation

<i>Chapitre V : Solution proposée</i> -----	51
1 Introduction-----	52
2 Limites des systèmes de recommandation actuels-----	52
3 Méthode proposée-----	53
3.1 Algorithme des k-moyennes (<i>k-means</i>)-----	54
3.2 Méthode de <i>classification ascendante hiérarchique</i> -----	55
3.3 Distance de corrélation-----	57
3.4 Architecture globale du système-----	58

3.5	Modélisation de données	65
4	Conclusion	67
<i>Chapitre VI : Evaluation</i>		68
1	Introduction	69
2	Jeu de données	69
2.1	Analyse du jeu de données	69
2.2	Critères de formation des communautés	70
2.3	Construction de la table de communautés	72
3	Métriques d'évaluation	72
4	Résultats	74
4.1	Expérience 1 – Formation des communautés	74
4.2	Expérience 2 – Recommandation par priorité	76
4	Conclusion	80
<i>Conclusion et Perspectives</i>		81
<i>Bibliographie</i>		82
<i>Annexe</i>		87
1	Outil d'évaluation	87
2	Captures d'écran	87
<i>Glossaire</i>		89

Liste des figures

Figure I.1 – Espaces de communautés pour chaque critère.	5
Figure II.1 – Filtrage d'information.	11
Figure II.2 – Modèle général pour le filtrage d'information, adapté de Belkin et Croft.	12
Figure II.3 – Éléments de base d'un système de filtrage.	13
Figure II.4 – Filtrage basé sur le contenu.	17
Figure III.1 – Filtrage collaboratif.	21
Figure III.2 – Architecture générale d'un système de filtrage collaboratif.	22
Figure IV.1 – Exemple de communautés pour le critère « <i>Evaluation</i> ».	30
Figure IV.2 – Principes du <i>Clustering</i>	32
Figure IV.3 – Matrice des évaluations $V_{m \times n}$	32
Figure IV.4 – Illustration de sélection des <i>voisins les plus proches</i> par le seuil δ (en 2D).	33
Figure IV.5 – Exemple d'un <i>arbre de décision</i> ($D = \{Evaluation\}$).	34
Figure IV.6 – Matrice des évaluations binaires $V_{m \times n}$	36
Figure IV.7 – Matrice de transformation V'	36
Figure IV.8 – Mise en correspondance neurone biologique / neurone artificiel.	38
Figure IV.9 – Modèle d'un neurone formel.	38
Figure IV.10 – Structure d'un Perceptron monocouche.	40
Figure IV.11 – Exemple d'un réseau multicouche.	41
Figure IV.12 – Schéma du modèle de la rétro-propagation de l'erreur.	42
Figure IV.13 – Séparation linéaire dans un espace à deux dimensions.	44
Figure IV.14 – Exemple d'un réseau social de 5 personnes, par la relation « <i>être ami</i> ».	45
Figure IV.15 – Construction des réseaux G_s et G_f pour la production de recommandations.	46
Figure IV.16 – Architecture d'un modèle de <i>Kohonen</i>	49
Figure IV.17 – Topologie de voisinage pour une carte à 2 dimensions.	49
Figure V.1 – Première utilisation de MovieLens par l'utilisateur safab_2002@yahoo.fr.	53
Figure V.2 – Exemple d'application de l'algorithme des <i>k-means</i>	55
Figure V.3 – Illustration de la méthode <i>CAH</i>	56
Figure V.4 – Architecture globale du système.	58
Figure V.5 – Fonctionnement d'un système de classification des utilisateurs.	59
Figure V.6 – Diagramme de classes d'un modèle de données d'un <i>SFC</i>	65
Figure V.7 – Modèle statique d'enchaînement des processus d'un <i>SFC</i>	65
Figure V.8 – Composants du processus de recommandation.	66
Figure V.9 – Diagramme d'états des processus d'un <i>SFC</i>	66
Figure VI.1 – Répartition des données d'expérimentation.	70
Figure VI.2 – Répartition des évaluations du jeu MovieLens.	70
Figure VI.3 – Pourcentage d'utilisateurs par tranche d'âge.	71
Figure VI.4 – Pourcentage d'utilisateurs par profession.	71
Figure VI.5 – Précision et rappel en Filtrage d'information.	73

Figure VI.6 – MAE obtenu pour le critère « <i>Evaluation</i> » - Approche classique -	74
Figure VI.7 – MAE obtenu pour le critère « <i>Evaluation</i> » - Approche proposée -	75
Figure VI.8 – Graphe des MAE comparatif entre les deux approches.	75
Figure VI.9 – MAE obtenu pour le critère « <i>Âge</i> ».	76
Figure VI.10 – Précision & Rappel obtenus pour le critère « <i>Âge</i> ».	77
Figure VI.11 – MAE obtenu pour le critère « <i>Profession</i> ».....	77
Figure VI.12 – Précision & Rappel obtenus pour le critère « <i>Profession</i> ».....	78
Figure VI.13 – MAE obtenu pour le critère le plus prioritaire.	78
Figure VI.14 – Précision & Rappel obtenus pour le critère le plus prioritaire.	79
Figure VI.15 – Graphe des MAE comparatif entre critères.....	79

Liste des tableaux

Tableau I.1 – Table de communautés d’un système de recommandation de films.	5
Tableau II.1 – Tableau comparatif des principes de R.I et de F.I fondé sur le contenu.	19
Tableau III.1 – Comparaison entre l’approche collaborative et l’approche basée contenu.	26
Tableau IV.1 – Table de communautés.	31
Tableau IV.2 – Table ordonnée des dissimilarités entre l’utilisateur u et tous les autres.	33
Tableau IV.3 – Fonctions de transfert : $a = f(n)$	39
Tableau V.1 – T_{mxm} : Tableau initial de valeurs d’approximation entre les utilisateurs.	56
Tableau V.2 – Exemple d’inscription des nouveaux utilisateurs.	60
Tableau V.3 – Nombre des évaluations données par les membres des communautés.	61
Tableau V.4 – Exemple de priorités entre critères (espaces).	62
Tableau V.5 – Matrice des scores de la communauté <i>Commerçant</i>	64
Tableau VI.1 – Comparaison entre approches par le taux d’erreur (MAE %).	75
Tableau VI.2 – Comparaison entre critères par le taux d’erreur (MAE %).	79

Liste des algorithmes

Algorithme IV.1 – Algorithme d’apprentissage d’un Perceptron.	41
Algorithme IV.2 – Algorithme d’apprentissage d’un <i>réseau de Kohonen</i>	50
Algorithme V.1 – Algorithme des <i>k-moyennes (k-means)</i>	54
Algorithme V.2 – Algorithme de la classification <i>ascendante hiérarchique</i>	56

Introduction générale

Le volume d'informations disponibles sur Internet ne cessant d'augmenter chaque jour, il survient aujourd'hui un problème de plus en plus important de *surcharge de données*, lors d'une recherche d'un utilisateur sur le réseau mondial. Il devient donc, de plus en plus nécessaire de développer des outils permettant de filtrer cet ensemble important de données, pour cibler au mieux les réponses fournies aux utilisateurs, afin qu'elles soient plus proches de leurs attentes personnelles.

Le filtrage de l'information est un nom donné à une variété de processus dont le but est de faire parvenir, à partir de larges volumes d'informations générées dynamiquement, les informations aux personnes qui en ont besoin, des personnes décrites par des profils pour mieux filtrer les données. Donc, nécessité de filtrer l'information par rapport à un profil personnalisé. Un système de filtrage d'information pourrait être défini comme étant la technique qui vise à filtrer les informations pertinentes d'un flux pour faire l'acheminement vers des groupes de personnes, en se basant sur leurs profils (intérêts) à long terme, contrairement à la recherche d'information où l'utilisateur à l'aide d'une requête sélectionne l'information pertinente à partir du flux.

Le filtrage collaboratif est la technique qui vise à prédire l'intérêt d'un utilisateur pour une ressource donnée. Les systèmes de filtrage présentent un accès passif à l'information, à la différence des systèmes de recherche où l'accès est actif. Un autre avantage spécifique pour l'approche collaborative, se présente dans la possibilité de recommander tout type de ressources (images, vidéos etc.). Cependant, ce type de systèmes de filtrage soulève des problématiques, comme, le cas d'un nouvel utilisateur ou d'une nouvelle ressource où nous n'avons aucune évaluation, et encore pire lors du démarrage à froid du système pour lequel aucune information n'est disponible.

Un système de filtrage collaboratif, comme son intitulé indique, la notion de *collaboration* est traduite dans le fonctionnement de ce type de système, qui se base principalement sur l'échange d'information entre les utilisateurs réunis en communautés suivant un critère donné. Un tel critère est une représentation réduite d'un profil utilisateur. Le regroupement des utilisateurs en communautés selon divers critères représente l'axe primordial de cette recherche, afin de garantir une certaine performance pour ce type de systèmes.

Afin que le filtrage soit efficace, la modélisation des intérêts de l'utilisateur est nécessaire, toutefois cette modélisation reste difficile, de ce fait, une automatisation d'un système de création de profils est indispensable.

Le présent mémoire est organisé en six chapitres :

- ✍ **Chapitre I** : introduit le contexte et la problématique de notre recherche, ainsi que notre finalité et objectifs sont mentionnés afin de contourner la problématique.

- ✍ **Chapitre II :** dans ce chapitre, nous présentons un état de l'art des systèmes de filtrage d'information : nous donnons l'architecture globale, le principe, le rôle, et le fonctionnement général d'un système de filtrage. Nous présentons la modélisation du profil utilisateur ainsi que les différentes formes de représentation et d'acquisition. Nous énonçons les différentes formes de filtrage ainsi que les principales caractéristiques par rapport au domaine de recherche d'information.

- ✍ **Chapitre III :** le troisième chapitre, présente un état de l'art d'une des catégories des systèmes de filtrage d'information dite « collaborative » : il présente l'architecture générale, le principe de base ainsi que le fonctionnement global d'un système de filtrage collaboratif. Il présente les différents algorithmes de prédiction utilisés et décrit les avantages et inconvénients de ce type de filtrage. Il termine par une étude comparative avec le filtrage basé contenu.

- ✍ **Chapitre IV :** traite les méthodes et les techniques de classification et de regroupement, il en existe deux grandes catégories : la classification supervisée et la classification non supervisée.
Dans ce chapitre, nous décrivons pour chaque catégorie les différentes méthodes utilisées telles que : les arbres de décision, les réseaux de neurones, naïve bayes, les k-plus proches voisins, la classification ascendante hiérarchique « dendrogramme », la méthode des k-moyennes, carte auto organisatrice de Kohonen.

- ✍ **Chapitre V :** dans ce chapitre, nous présentons notre proposition pour répondre à la problématique. Nous décrivons la conception détaillée de la solution en présentant l'architecture globale de l'approche préconisée ainsi que la description des différents modules.

- ✍ **Chapitre VI :** dans ce dernier chapitre, nous montrons par deux expérimentations différentes sur un jeu de données réel, comment notre approche apporte des améliorations à la qualité du processus de filtrage.

Enfin, nous terminons ce présent document par une conclusion et une description des perspectives.

Chapitre I

Contexte de recherche et objectifs

Nous commençons par présenter dans ce présent chapitre, le contexte général de notre recherche enveloppée par la construction des communautés d'utilisateurs des systèmes de filtrage d'information collaboratif selon une diversité de critères de formation de communautés. Nous définissons ensuite notre problématique et enfin, nous décrivons la finalité et les objectifs de cette recherche.

1 Présentation du contexte

Aujourd'hui, l'exploitation d'une grande masse d'information, celle de bases de données volumineuses, d'Internet (du web en particulier),... engendre un vrai problème de notre vie quotidienne, dans cette situation, les outils de recherche d'information répondent aux requêtes des utilisateurs par des résultats massifs dans un temps considérable, cette surcharge informationnelle génère chez l'utilisateur des difficultés pour distinguer l'information pertinente d'une autre secondaire, ou même du bruit. Donc, il est indispensable de trouver une solution à ce vrai problème, cette solution permettant de retourner à l'utilisateur que les ressources (ou d'url pour le web) jugées pertinentes (intéressantes) comme des réponses à sa requête formulée, ou au moins de lui interdire l'accès à des ressources contenant des informations estimées non souhaitables. Des travaux déjà réalisés et des autres sont en cours de développement pour construire des outils (systèmes de filtrage, ou systèmes de recommandation) permettant à l'utilisateur de filtrer, adapter et personnaliser sa recherche d'information.

Les systèmes de filtrage adaptatif, ou systèmes de recommandation ont pour objectif principal est de filtrer un flux entrant d'informations (documents) de façon personnalisée pour chaque utilisateur, tout en s'adaptant en permanence au besoin d'information de chacun [MLD03]. Pour cela, les moteurs de ces systèmes exploitent les profils d'utilisateurs pour produire une meilleure sélection des documents (les recommandations). Les systèmes sont conçus pour adapter ces profils en exploitant au mieux le retour de pertinence fourni par les utilisateurs.

Les systèmes de recommandation se basent sur l'une des trois grandes familles de filtrage, à savoir : cognitive, collaborative ou hybride qui combine les deux familles.

Dans la plupart des systèmes de filtrage collaboratif existants, les communautés sont généralement formées selon un seul critère [PGF03], par exemple la proximité des évaluations des utilisateurs. Pourtant il existe de multiples critères sur lesquels la formation des communautés peut s'appuyer.

Prenons un exemple d'un système de recommandation de films, qui sera le pivot de notre expérimentation par la suite, et dans lequel une multiplicité de critères constitue le profil d'un utilisateur [AGS97] tels que sa profession, sa ville où il habite, son genre de film préféré et ses évaluations sur certains films. Pour chacun de ces critères, on peut former un espace de communautés d'utilisateurs selon leur proximité relativement à ce critère. D'une façon plus précise, le système peut créer des groupes ou des partitions embrassant les utilisateurs les plus proches entre eux, pour chacun des critères *Profession*, *Ville* et *Genre préféré*, en plus de l'espace de communautés créé par le critère traditionnel *Evaluation* basé sur les évaluations d'utilisateurs. Comme indiqué dans le « Tableau I.1 » par exemple, l'utilisateur u_9 a comme valeur « Commerçant » dans l'espace *Profession*, « New York » dans l'espace *Ville* et « Documentaire » dans l'espace *Genre préféré* et une liste de ses évaluations passées, appartient simultanément à quatre communautés différentes selon ces quatre critères.

Le tableau I.1 présente une table de communautés d'un échantillon de 12 utilisateurs, construite selon ces quatre critères cités précédemment, où chaque valeur est une étiquette d'une communauté. Chaque colonne de la table correspond à un espace de communautés relatif à un critère spécifique, et chaque ligne contient les communautés auxquelles appartient un utilisateur particulier. La figure I.1 nous montre que les utilisateurs sont associés très différemment les uns aux autres selon le critère

choisi. Par exemple, on trouve les utilisateurs u_2 , u_5 et u_7 dans une même communauté dans l'espace *Ville*, mais dans des communautés entièrement différentes dans l'espace *Genre préféré*.

Utilisateur	Espace			
	Profession	Ville	Genre préféré	Evaluation
u_1	Commerçant	Paris	Aventure	Groupe 1
u_2	Chercheur	Paris	Aventure	Groupe 4
u_3	Chercheur	Paris	Documentaire	Groupe 2
u_4	Chercheur	Paris	Policier	Groupe 1
u_5	Chercheur	Paris	Policier	Groupe 4
u_6	Chercheur	Paris	Policier	Groupe 3
u_7	Chercheur	Paris	Fiction	Groupe 5
u_8	Chercheur	New York	Documentaire	Groupe 5
u_9	Commerçant	New York	Documentaire	Groupe 5
u_{10}	Commerçant	Londres	Documentaire	Groupe 3
u_{11}	Commerçant	Londres	Documentaire	Groupe 2
u_{12}	Commerçant	Londres	Documentaire	Groupe 3

Tableau I.1 – Table de communautés d'un système de recommandation de films.

Afin d'interpréter le tableau précédent d'une façon plus claire, nous procédons de le représenter schématiquement par la figure suivante (Figure I.1).

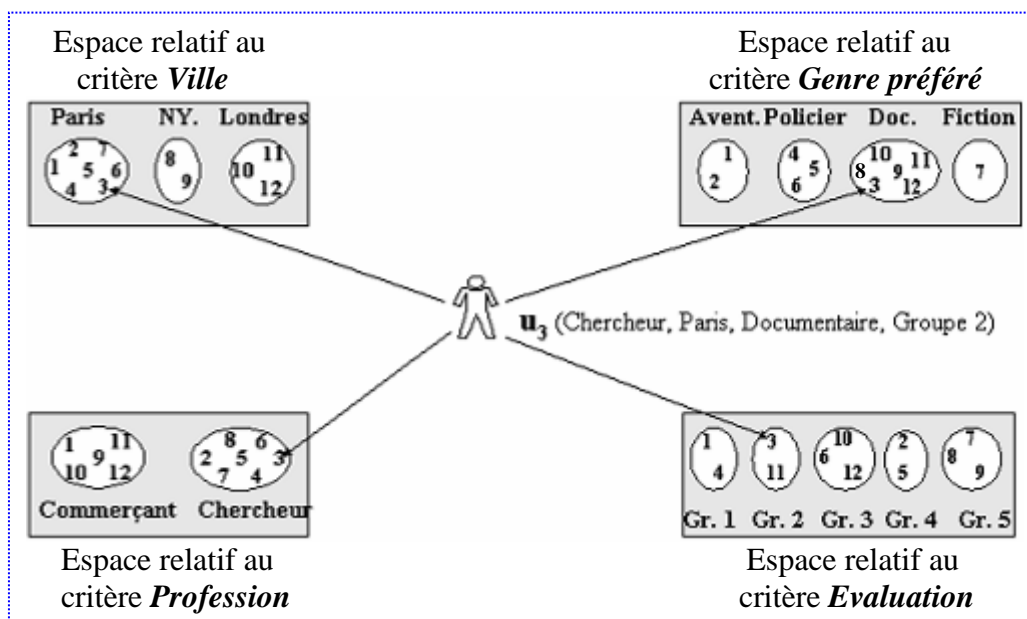


Figure I.1 – Espaces de communautés pour chaque critère.

Dans un espace donné relatif à un critère particulier de formation des communautés, il est difficile et sensible de positionner un utilisateur dans sa communauté appropriée. Donc, une mauvaise affectation des utilisateurs, conduira à une mauvaise production de recommandations.

Comme la figure I.1 montre, dans le contexte de multiplicité de critères, chaque utilisateur peut se situer dans plusieurs communautés différentes, chacune relative à un critère particulier, d'où cette diversité est exploitée pour enrichir la production de

recommandations. Ainsi, un utilisateur peut recevoir les recommandations de chacune des communautés auxquelles il appartient. Dans l'exemple ci-dessus, l'utilisateur u_3 peut recevoir les films envoyés via les communautés *Chercheur*, *Paris* et *Documentaire*, de plus les films émanant de la communauté *Groupe2* formée selon le critère *Evaluation*.

2 Problématique

Afin d'atteindre la finalité de cette recherche qui enveloppée dans l'amélioration de la performance globale des systèmes de filtrage d'information collaboratif, il est nécessaire de trouver des solutions aux difficultés qui traquent ce type de systèmes, ou réduire au moins l'effet de ses difficultés. Ces dernières sont résumées dans les trois grands inconvénients tels que le *démarrage à froid*, la *masse critique* et le *rapport coût-bénéfice*.

Burke distingue trois types concernant la première difficulté [Bur02]. Le *démarrage à froid* d'un nouveau système dans lequel, aucune information disponible sur laquelle baser le processus de filtrage personnalisé. Le deuxième type représente le parvenu d'un nouveau document qui ne peut être diffusé aux utilisateurs intéressés parce qu'il n'a pas encore été évalué. L'inscription d'un nouvel utilisateur crée la troisième variété du problème de démarrage à froid, où le participant commence par un profil vide ou inexistant et ses communautés sont encore inconnues, ce qui conduit à des mauvaises recommandations.

Le principe d'automatisation des recommandations s'appuie généralement sur l'hypothèse que le volume des données disponible (nombre d'utilisateurs, nombre des documents et nombre d'évaluations) a atteint un seuil critique, chose qui dépend du comportement de l'utilisateur face au système. En effet, le système exige une *masse critique* du nombre d'appréciations en commun pour comparer les utilisateurs entre eux et former de bonnes communautés, afin de produire de recommandations. Par exemple, si le système n'a pas d'évaluations, alors il ne produit pas de recommandations. De même, s'il y a peu de recouvrements entre les profils des utilisateurs, alors il y a peu de recommandations.

Au cours de son exploitation d'un système de filtrage collaboratif, l'utilisateur ne retient que deux pensées : « je suis invité à évaluer des documents », pour lui cela signifie un coût (ou un effort) ; et « le système doit me fournir des recommandations », cela représente pour lui un bénéfice [Gal05]. Par contre, il oublie qu'il se trouve dans un milieu collaboratif. Le fait qu'un utilisateur soit conscient du milieu collaboratif pourrait influencer positivement sur son comportement face au système. D'un autre côté, son unique critère pour juger l'efficacité du système est basé sur le rapport entre ses attentes de retour de recommandations pertinentes et l'effort investi dans la tâche d'évaluation des documents. Généralement l'utilisateur se décourage lorsqu'il perçoit une constante baisse dans la pertinence des documents ou dans le nombre de recommandations, l'évaluation devient une tâche fastidieuse et cela peut le conduire à l'abandon du système.

Le principe des systèmes de filtrage collaboratif compare les utilisateurs entre eux sur la base de leurs jugements passés pour les regrouper en communautés, et chaque utilisateur reçoit les documents jugés pertinents par sa communauté [BHK98]. Ces communautés sont généralement formées selon un seul critère : la proximité des évaluations explicites ou implicites des utilisateurs sur les recommandations déjà

reçues. Pourtant, il existe une multiplicité d'informations importantes qui modélisent les utilisateurs et sur lesquelles appuyer la formation des communautés, à savoir : informations personnelles, domaines d'intérêt, historique des évaluations, préférences de sécurité, etc. [AS99, BK05]. Alors, la question qui se pose : est-il possible de négliger toutes ces informations et limiter le regroupement des utilisateurs selon un seul critère ? Néanmoins, dans le cadre de la diversité de critères, chaque utilisateur appartient à plusieurs communautés en même temps. Cette multiplicité d'appartenance permet d'enrichir et de diversifier les recommandations générées pour les utilisateurs et par conséquent, l'utilisateur contribue activement au système. En effet, cette contribution améliore la qualité et la performance du système.

En effet, la plupart des systèmes de filtrage collaboratif existants, sont monocritère, et donc il existe un seul espace de communautés créé généralement sur la base du critère choisi ; néanmoins l'utilisation de la notion de la multiplicité de critères conduit à l'existence de plusieurs espaces de communautés à la fois, et donc par conséquent, les diverses positions d'un utilisateur peut contribuer positivement à l'évolution de son profil.

En général, le positionnement des utilisateurs dans les espaces de communautés est une tâche difficile, en plus, la qualité de ce positionnement est conditionnée principalement par la qualité des valeurs attribuées par le système pour chaque utilisateur à chaque critère. Cependant, certains critères demandent beaucoup d'efforts de la part des utilisateurs, et peuvent être coûteux également pour le système [MLD03]. Par exemple, dans un système de recommandation de films, un nouvel utilisateur peine à définir son genre de films préférés ou évaluer un grand nombre de films est une tâche lourde pour lui [Paw04, MLD03], et pourtant le critère usuel dans la formation des communautés est celui de l'*Evaluation*.

3 Objectifs

En général, dans un système de filtrage collaboratif classique, les communautés d'utilisateurs sont construites selon un critère donné (le critère traditionnel est la proximité des évaluations des utilisateurs) afin de former un seul espace de communautés.

Notre objectif est d'améliorer la qualité globale du fonctionnement des systèmes de filtrage d'information collaboratif. Ceci conduit à améliorer la qualité de la formation des communautés d'utilisateurs qui sont considérées comme le noyau de ces systèmes et d'adopter des solutions pour dépasser certaines difficultés que ce type de systèmes souffre, comme le problème du démarrage à froid. Pour cela, il est nécessaire d'intégrer la notion de multiplicité de critères sur lesquels on peut former des communautés d'utilisateurs. Cette variété de critères conduira à l'existence de multiples espaces de communautés à la fois, et donc un utilisateur peut enrichir son profil par la réception d'une diversité de recommandations de chacune de ses communautés auxquelles il appartient. Cette diversité de recommandations héritée par un utilisateur conduit à évoluer et développer son profil au cours du temps.

Afin d'atteindre notre finalité, et de construire des communautés d'utilisateurs répondant aux besoins de leurs membres, nous nous intéressons dans notre conception, d'utiliser la classification non supervisée du fait que nous n'avons pas de classes prédéfinies au départ pour les exploiter comme modèle ; nous choisissons de combiner deux méthodes de cette famille tels que l'algorithme des *k-moyennes* et la méthode de la

classification ascendante hiérarchique. Notre contribution figure aussi dans le processus de production de recommandations et du filtrage collaboratif, à savoir nous introduisons la notion de priorité entre critères tel que l'espace de communauté relatif au critère le plus prioritaire sera concerné pour la production de recommandations.

Nous rappelons, que notre recherche concerne la formation multicritères d'espaces de communautés d'utilisateurs (classification des utilisateurs) pour un système de filtrage d'information collaboratif. En effet, après une étude détaillée des différentes méthodes de classification, nous proposons une approche pour classifier les utilisateurs dans différents espaces, chacun relatif à un critère bien déterminé, nous passons ensuite à une phase d'apprentissage, en utilisant une base de données réelle et enfin, nous terminons par une phase de tests réalisée sur cette base de données.

Chapitre II

Systèmes de filtrage d'information

Le filtrage de l'information pourrait être défini comme étant le processus qui vise à filtrer les informations d'un flux pour ne faire parvenir que celles qui intéressent l'utilisateur, contrairement à la recherche d'information où l'utilisateur à l'aide d'une requête sélectionne l'information pertinente à partir du flux.

On distingue trois catégories des systèmes de filtrage d'information :

- ✂ Filtrage cognitif : basé sur le contenu des documents.
- ✂ Filtrage collaboratif : basé sur les évaluations des utilisateurs sur les ressources.
- ✂ Filtrage hybride : combinaison des deux approches précédentes.

1 Introduction

Généralement, on considère qu'un système de recherche d'information a pour but principal « d'amener à l'utilisateur les documents qui vont lui permettre de satisfaire son besoin en information » [BC92].

Aujourd'hui, plusieurs façons pour accéder à l'information : la recherche active de documents à travers des systèmes de recherche d'information, la réception de documents par des différentes personnes, la rencontre inattendue d'un document par navigation sur Internet par exemple, etc.

Pour avoir une information, les moteurs de recherche d'information, demandent à l'utilisateur de formuler son besoin systématiquement, ces derniers permettent ainsi la découverte ponctuelle de documents.

Etre informé étant une nécessité professionnelle et citoyenne, recevoir des informations ayant un certain niveau d'intérêt individuel permet à chacun d'apprendre, d'analyser, et de critiquer toute nouvelle source d'information. Ainsi recevant toute nouveauté, l'utilité du filtrage permet donc d'éviter de procéder régulièrement à une recherche d'éventuelles avancées. Cela procure à l'utilisateur bien évidemment une économie d'effort mais également une certaine sérénité.

Il existe aujourd'hui de nombreux systèmes de filtrage appliqués à plusieurs domaines de la vie quotidienne. A l'origine, ces systèmes se sont appliqués aux forums électroniques, ce gisement d'informations où de nouveaux documents naissent chaque jour est un exemple concret. Il existe aussi des systèmes de recommandation du courrier électronique qui permettent de catégoriser les messages reçus automatiquement, des systèmes utilisés pour exploiter les archives électroniques de documents et d'autres utilisés dans le commerce électronique, les loisirs, la recherche documentaire scientifique, la gestion des connaissances, ...

2 Description générale

Dès qu'une recherche d'une information est lancée par un utilisateur en ligne, un volume important d'informations se présente, mais qu'elle est l'importante, la pertinente et la convenable ? La réponse est la suivante : c'est le rôle du système de filtrage, qui a pour objectif est de « *fournir la bonne information à la bonne personne au bon moment* ».

« Filtrage d'information » est l'expression utilisée pour décrire une variété de processus ayant pour but de fournir des informations à des personnes, informations en adéquation avec des centres d'intérêt de ces personnes [BC92]. Le filtrage peut être vu comme la sélection d'informations pertinentes sur un flux entrant.

Comme le présente la figure II.1, pour sélectionner les informations pertinentes à partir d'un volume important de documents (*d*) « Flux entrant », le système fait une « prédiction ». Cette prédiction s'appuie sur le « profil » de l'utilisateur et se termine par une prise de décision : information « recommander » ou « ne pas recommander ».

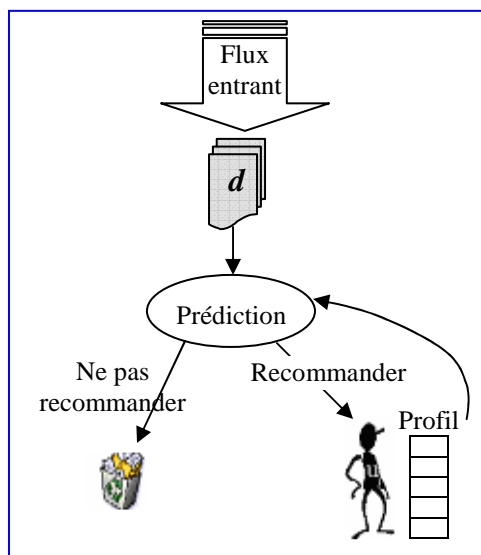


Figure II.1 – Filtrage d'information.

Les utilisateurs décrivent leurs centres d'intérêt eux-mêmes interprétés par un « profil ». Ce dernier explique le besoin d'information en permanence de l'utilisateur.

Le filtrage d'information est réalisé à partir d'un volume important d'informations disponibles dynamiquement [Ter93] via le « flux entrant ». Ces informations proviennent éventuellement de plusieurs sources différentes ; elles peuvent être collectées passivement (news), activement (www) ou les deux à la fois.

« Un système de filtrage d'information est défini par son modèle de représentation des profils utilisateurs, son modèle de représentation des documents et sa fonction de prédiction sur la pertinence des documents reçus » [Tma02].

3 Fonctionnement d'un système de filtrage

Un système de filtrage d'information « achemine des documents qui se présentent vers des groupes de personnes, en se basant sur leurs profils à long terme », et élaborés à partir de données d'apprentissage [Cro93].

Généralement, le filtrage d'information est considéré comme l'élimination d'informations non souhaitables sur un flux entrant, plutôt que la recherche d'informations spécifiques sur ce flux. L'approche la plus répandue est basée sur le contenu sémantique des documents. Elle trouve ses racines dans le monde de la recherche d'information, et utilise plusieurs de ses principes ; les documents textuels sont proposés sur la base d'une comparaison de leur contenu et du profil de l'utilisateur. Ce profil est présenté sous forme d'un ensemble de termes et de pondérations, établis à partir de documents que l'utilisateur a jugés pertinents. Cette approche est simple, rapide et a fait ses preuves en recherche d'information classique [BS97].

Les applications de filtrage impliquent typiquement des flux de données entrantes [BC92], données émises par une source distante ou envoyées directement par d'autres sources. Le filtrage est basé sur des descriptions d'individus et de groupes, généralement appelées profils. Habituellement, un profil représente un thème d'intérêt à long terme.

Comme le présente la figure II.2, le filtrage d'information commence avec des personnes (les utilisateurs du système de filtrage d'information) qui ont des objectifs ou des désirs relativement stables, à long terme ou périodiques. Des groupes, aussi bien que des personnes peuvent être caractérisés par de tels buts. Ceci amène à des besoins réguliers d'information qui peuvent évoluer lentement au cours du temps au fur et à mesure que les conditions, objectifs et connaissances changent. De tels intérêts engagent les utilisateurs dans un processus relativement passif de recherche d'information. Ce processus est réalisé à travers la représentation des besoins en information par des profils ou des requêtes destinés au système de filtrage d'information.

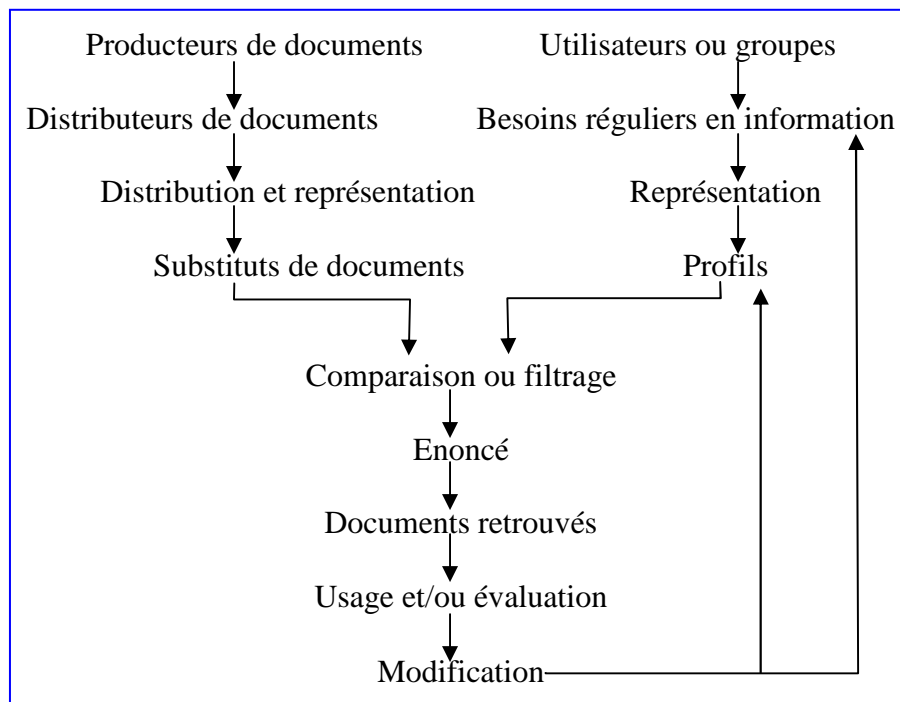


Figure II.2 – Modèle général pour le filtrage d'information, adapté de Belkin et Croft [BC92].

D'un autre côté, les producteurs de documents, qui sont généralement des institutions, entreprennent de distribuer leurs produits dès qu'ils sont générés. Pour accomplir cette tâche, on associe aux documents une représentation de leur contenu, qui est ensuite comparée aux profils. Les documents sont utilisés et évalués en termes de réponse aux besoins exprimés. Cette évaluation peut mener à la modification des profils et des domaines d'intérêt (besoin réguliers en information).

4 Rôle d'un système de filtrage

- Faire parvenir à partir de larges volumes d'informations générées dynamiquement, les informations aux personnes qui en ont besoin ;
- Augmenter la qualité d'informations pertinentes (degré de pertinence) collectées à partir de différentes sources ;
- Cibler l'information vraiment pertinente suivant les besoins de connaissance établis par l'utilisateur.

5 Composantes d'un système de filtrage [Nou04]

Comme l'illustre la figure suivante, un système de filtrage d'information est composé d'un ensemble d'éléments de base, à savoir :

- Analyse et représentation de l'information : cette dernière peut être représentée d'une manière non structurée par des termes sans structure (un sac de mots), ou bien d'une manière structurée par des unités syntaxiques ou sémantiques (des structures syntaxiques des termes ou par les sens des termes respectivement). Il existe plusieurs techniques pour la représentation de l'information parmi lesquelles : MI , χ^2 ... ;
- Modélisation des utilisateurs par la représentation de leurs intérêts (profils) ;
- Utilisation d'une fonction de mesure de similarité pour déterminer le degré de similitude entre l'information et le profil d'utilisateur, par l'usage des méthodes de similarité comme *cosinus*, *S-Matching*, ... ;
- Exploitation des résultats du processus de filtrage par l'application de l'action adéquate tel que : classement, diffusion, ... ;
- Exploitation des résultats de mesure tel que l'apprentissage et l'adaptation.

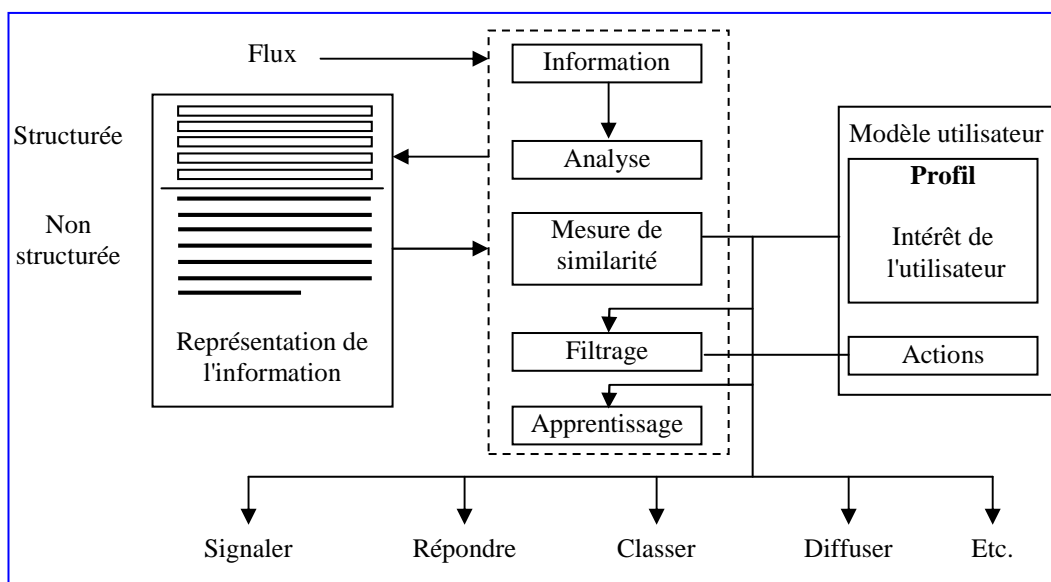


Figure II.3 – Éléments de base d'un système de filtrage.

6 Profil utilisateur

6.1 Définition

Afin de définir le comportement d'un utilisateur, il faut d'abord connaître l'ensemble d'informations décrivant les différents aspects de cet utilisateur. Plus précisément dans le contexte des systèmes de filtrage et de recommandation, l'utilisateur est le membre principal. Ainsi que la qualité de ces systèmes est fortement dépendante de la quantité et de la qualité d'informations disponibles décrivant cet utilisateur. Donc, la description du profil de l'utilisateur est une étape indispensable, afin qu'un système de filtrage génère des recommandations.

La description initiale du profil de l'utilisateur est donc une étape cruciale pour un système de filtrage, sans laquelle on ne pourra faire de recommandation, et l'utilisateur abandonnera tout simplement le système.

Le profil utilisateur peut être décrit par deux types de caractéristiques [Bru01], l'un de ces types est explicite qui est en général objectif, et l'autre type implicite plus subjectif et difficile à traduire mais contenant plus d'information sur les réels besoins de l'utilisateur.

Les caractéristiques explicites représentent les traits de l'utilisateur qui sont ses acquis, ses connaissances, ses objectifs, ses préférences, son centre d'intérêt, etc. Les recommandations produites par le système doivent être correspondre au niveau de connaissances de l'utilisateur, à ses objectifs et finalités, à ses références (profession, domaine de recherche, etc.) et expérience, à ses préférences (langue, type et format de l'information, etc.), sans oublier les critères du centre d'intérêt qui peuvent être décrites par des mots clés, des tags, etc., ainsi ne néglige pas les traits de son caractère. Un utilisateur pouvant être introverti, extraverti, timide, exubérant, etc. Il devrait recevoir plusieurs types de ressources correspondant à son trait de caractère.

Les caractéristiques implicites sont induites à partir de l'activité de navigation, d'évaluation, marquage avec des tags, réponse à des questions, etc. L'avantage d'utiliser des techniques implicites se fait sentir car l'utilisateur est déchargé de certaines actions, telles que la définition de ces préférences, de son caractère, etc.

6.2 Modélisation d'un profil utilisateur

L'objectif principal de la modélisation d'un profil utilisateur est la structuration des informations qui caractérisent l'utilisateur, cette structuration se traduit par l'énumération des informations nécessaires à la description de l'utilisateur.

Différents travaux ont abordé la structuration et la modélisation du profil utilisateur, par exemple, certains chercheurs proposent cinq catégories pour modéliser l'utilisateur d'une bibliothèque digitale : données personnelles, données collectées (contenu, structure des documents, etc.), données de livraison (temps et moyen de livraison), données de comportement (interaction de l'utilisateur avec le système), données de sécurité (conditions d'accès) [AS99]. D'autres recherches distinguent principalement huit dimensions capables d'accueillir la plupart des informations caractérisant un profil [BK04], une huitième dimension contenant toutes les informations inclassables.

- ✍ Données personnelles ;
- ✍ Centres d'intérêt ;
- ✍ Ontologie du domaine ;
- ✍ Qualité attendue des résultats délivrés ;
- ✍ Customisation (personnalisation) ;
- ✍ Sécurité et confidentialité ;
- ✍ Retour de préférences (feedback) ;
- ✍ Informations diverses.

Les données personnelles représentent la partie statique du profil. Elle est caractérisée par : l'identité de la personne, le nom, le numéro de sécurité sociale, l'âge, le sexe, la profession, les langues parlées, ... Ces informations sont généralement stables dans le temps et ne demandent pas de mise à jour automatique. Les centres d'intérêt d'un utilisateur représentent la partie dynamique du profil, ils peuvent être divers et variés, généralement décrits à travers des mots-clés. L'ontologie du domaine complète la définition du centre d'intérêt, elle permet de rendre explicite la sémantique de certains

termes employés dans le profil. La qualité attendue exprime le niveau de qualité, l'origine, la précision, la fraîcheur et la crédibilité de l'information. La customisation concerne l'adaptation et la personnalisation de l'interface selon les préférences et les commodités de l'utilisateur tels que les modalités de présentation des résultats et les choix esthétiques ou visuels de l'utilisateur, la quantité de résultats qu'il souhaite recevoir, etc. La sécurité et confidentialité concernent la définition des droits d'accès au système et la définition du degré de visibilité de certaines opérations faites par l'utilisateur. Le retour de préférences désigne le feedback de l'utilisateur concernant les ressources, qu'il soit explicite ou bien implicite en analysant le comportement de l'utilisateur sur le système. La dernière dimension peut regrouper certaines informations spécifiques selon l'exigence de l'application ou du contexte de travail.

Dans le contexte de confidentialité des profils, et selon une étude effectuée en France, 72% des internautes se méfient d'Internet, parmi eux, 47% s'inquiètent justement de la réutilisation de leurs données personnelles sur le web. Dans ce cadre, le consortium W3C¹ a développé le projet P3P² (Plate-forme pour les Préférences de Confidentialité) pour la sécurisation des profils. Ce projet fournit un moyen standard, simple et automatique, permet aux utilisateurs de prendre davantage le contrôle de l'utilisation de leurs données personnelles lorsqu'ils visitent des sites Web.

6.3 Représentation d'un profil utilisateur

Il existe plusieurs méthodes pour représenter le profil utilisateur, parmi elles, nous mentionnons trois modèles de représentation les plus utilisés :

1. Modèle vectoriel

On peut utiliser un modèle vectoriel où les coordonnées d'un vecteur représentent le poids lié à un terme [ZLB05]. Le profil est alors représenté par un ou plusieurs vecteurs définis dans un espace de termes et dont les coordonnées correspondent à leurs poids respectifs. L'utilisation de plusieurs vecteurs pour représenter le profil permet la prise en compte de la diversité des centres d'intérêts et de leur évolution à travers le temps.

2. Modèle sémantique

On peut utiliser une représentation sémantique du profil, où l'utilisateur est décrit par un ensemble d'attributs dont les valeurs appartiennent à l'ensemble des termes d'une ontologie.

3. Modèle multidimensionnel

Le profil utilisateur peut contenir plusieurs types d'informations telles que les données démographiques, centres d'intérêts, objectif, information historique et autres. Chaque type d'information est une dimension dans le modèle multidimensionnel [Kie06]. Comme on peut aussi utiliser un modèle relationnel simple où l'utilisateur est décrit par un ensemble d'attributs.

6.4 Construction et acquisition de profil

La construction du profil n'est pas une tâche aisée, car l'utilisateur peut ne pas être sûr de ses centres d'intérêts ou ne peut les spécifier d'une manière définitive d'une part et ne souhaite pas fournir trop d'efforts pour sa création d'autre part.

¹ W3C: World Wide Web Consortium

² P3P: Platform for Privacy Preference

Dans le contexte d'acquisition et de construction d'un profil utilisateur, plusieurs approches ont été proposées. Cependant, la qualité de cette acquisition dépend fortement des caractéristiques et des fonctionnalités offertes par les systèmes eux-mêmes et de leurs capacités à extraire des informations qui décrivent précisément les besoins de l'utilisateur.

Le profil utilisateur peut être construit à partir des informations renseignées par l'utilisateur lui-même à travers sa définition explicite d'une liste pondérée de mots-clés [BR99], ou par une série des évaluations sur des documents. Néanmoins le problème majeur réside sans doute, dans le choix des termes de cette liste et leurs pondérations respectives et elle demande beaucoup d'effort pour évaluer les documents proposés. Les données caractérisant le profil peuvent être annoncées par la sélection d'un profil prédéfini créé par un expert du domaine, ce qui nous permet d'obtenir un profil expert, ou annoncées par l'adaptation d'un profil prototype existant, c'est le profil prototype adapté. Comme, ces données peuvent être apprises par le système au cours de son utilisation, ce qui constitue le profil dynamique [Nou04]. Cette définition implicite est réalisée par l'observation du comportement de l'utilisateur pour prédire ses besoins et intentions en utilisant des approches d'acquisition dynamique ou d'apprentissage.

Les profils créés par les experts sont des profils statiques difficiles de mettre à jour dès que le système démarre. Une autre forme de profils réflexifs où l'utilisateur doit remplir des formulaires pour configurer son propre profil, cette forme offre plus de précision et d'adaptation. Une troisième forme de profils, qui sont corrigés dynamiquement dont le système de modélisation observe l'activité de l'utilisateur et apprend le profil de l'utilisateur à partir de ses actions.

Le profil utilisateur peut évoluer, cette évolution montre leur adaptation à la variation des centres d'intérêt des utilisateurs qu'ils décrivent, et par conséquent, de leurs besoins en information au cours du temps.

7 Différentes formes de filtrage d'information

Il existe deux grandes familles de filtrage d'information : basé sur le contenu (cognitif) et collaboratif (social), la troisième est une combinaison de ces deux familles.

7.1 Filtrage basé sur le contenu « cognitif »

L'approche la plus ancienne dans les systèmes de filtrage d'information est basée sur le contenu des documents.

Dès qu'un document arrive, il est indexé par thèmes (généralement sous forme de termes) puis comparé aux profils des utilisateurs et recommande ceux qui sont les plus proches. Un profil exprime le centre d'intérêt d'un utilisateur sous forme d'un ensemble de termes donnés explicitement par ce dernier.

Le profil est généralement modifié au cours du temps à partir des documents que l'utilisateur a jugés pertinents ; il intègre souvent des pondérations associées aux termes.

Les systèmes de filtrage basés sur le contenu permettent donc d'identifier des documents correspondant à un certain sujet ou centre d'intérêt. Ce type de filtrage peut être vu comme un système de recherche d'information dont la fonction de correspondance entre une requête exécutée par un utilisateur pour trouver des documents. Cette fonction joue le rôle d'un filtre permanent entre le profil de l'utilisateur (sorte de

requête à long terme et évolutive) et le flot de documents entrant (sorte de corpus évolutif).

Comme le montre la figure II.4, ce type de système fait une prédiction de l'appréciation qu'un utilisateur aura d'un document donné. Cette prédiction est calculée par le rapprochement des thèmes énoncés par l'utilisateur comme constituant son profil, aux thèmes extraits des documents par un processus d'indexation.

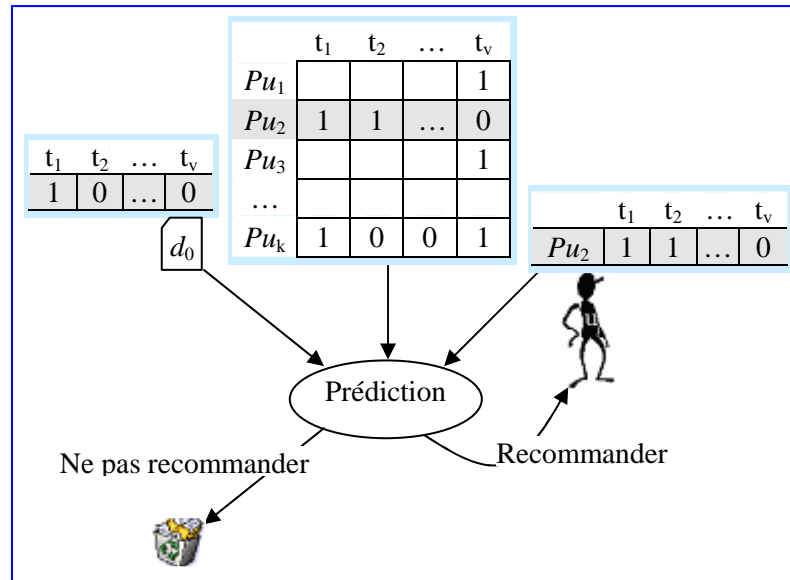


Figure II.4 – Filtrage basé sur le contenu.

Deux fonctionnalités importantes ressortent, pour le système de filtrage cognitif :

- ✂ La sélection des documents pertinents vis-à-vis du profil ;
- ✂ La mise à jour du profil en fonction du retour de pertinence fourni par l'utilisateur sur les documents reçus ; la mise à jour se fait par intégration des thèmes abordés dans les documents jugés pertinents.

Parmi les atouts des systèmes de filtrage basés sur le contenu est que les termes qui indexent le document seront comparés directement avec ceux qui indexent le profil d'utilisateur. Donc l'utilisateur est indépendant des autres ce qui lui permet d'avoir des recommandations même s'il est le seul utilisateur du système.

Néanmoins, cette approche de systèmes présente certaines limitations :

- ✂ Le filtrage basé sur le contenu ne permet pas d'intégrer d'autres facteurs de pertinence que le facteur thématique. Pourtant il existe de nombreux facteurs de pertinence comme la qualité scientifique, le public visé, la fiabilité de la source d'information, le degré de précision, etc. ;
- ✂ Difficulté à indexer les documents multimédia (images, vidéos, etc.) et donc la difficulté à recommander ce genre de documents ;
- ✂ Problème de démarrage à froid : Un nouvel utilisateur du système risque des difficultés à exprimer son profil en spécifiant des thèmes qui l'intéressent ;

- ✎ L'effet dit « entonnoir » : Un nouvel axe de recherche est décrit par des nouveaux concepts dans un domaine bien précis, peut ne pas être pris en compte, car ces nouveaux concepts ne fait pas partis du profil de l'utilisateur.

7.2 Filtrage collaboratif « social »

Le filtrage collaboratif compare les utilisateurs entre eux sur la base de leurs jugements passés pour créer des communautés, et chaque utilisateur reçoit les documents jugés pertinents par sa communauté [Bre98].

Dans ce type de filtrage, le choix des documents proposés est basé sur les opinions d'utilisateurs sur ces documents.

La prédiction de l'opinion qu'un utilisateur u_0 aura d'un document donné est calculée en rapprochant les évaluations passées de l'utilisateur des évaluations que d'autres utilisateurs de la communauté ont données par le passé sur les mêmes documents.

Quand un document arrive, il doit avoir au moins une évaluation pour pouvoir être recommandé par le système de filtrage. Par ailleurs, un profil se présente sous la forme d'un ensemble d'évaluations sur des documents, faites dans le passé par l'utilisateur. Le profil est mis à jour au cours du temps à partir des nouvelles évaluations que l'utilisateur réalise.

7.3 Filtrage hybride

La combinaison des deux familles présentées précédemment conduit à la création d'une nouvelle approche hybride utilisée par un nombre important de systèmes.

En général, dans cette approche, les profils sont orientés contenu, et la comparaison entre ces profils donne lieu à la formation de communautés permettant le filtrage collaboratif. La façon dont ces deux approches s'articulent varie, mais les deux ont des atouts complémentaires. Les documents peuvent alors être acheminés vers d'autres utilisateurs en utilisant les critères de filtrage collaboratif (évaluation) et basé sur le contenu (contenu des documents).

8 Recherche d'information versus Filtrage d'information

Bien que proches dans un certain nombre de fonctionnalités, recherche d'information et filtrage d'information s'opposent en un certain nombre de points :

8.1 Recherche d'information (RI)

- ✎ Spécifiquement concernée par des usages singuliers du système, avec une personne ayant un objectif et une requête à la fois ;
- ✎ Dans la recherche d'information, les requêtes reflètent des intérêts à court terme d'un utilisateur ;
- ✎ La principale fonctionnalité des systèmes de recherche d'information est la collecte et organisation des documents ;
- ✎ La recherche d'information permet la sélection des documents à partir d'une base relativement classique (source d'information stable) ;
- ✎ La recherche d'information implique le processus de collecte (Finding) de l'information dans la base de données.

8.2 Filtrage d'information (FI)

- ✍ Concerné par des usages répétitifs du système, par une personne ou groupe de personnes avec des buts et des intérêts à long terme (besoin en information stable) ;
- ✍ Dans les systèmes de filtrage d'information, la requête est remplacée par le profil de l'utilisateur qui est une sorte de requête à long terme ;
- ✍ La fonctionnalité de base des systèmes de filtrage d'information est la diffusion et distribution des documents à des groupes ou à des individus ;
- ✍ Le filtrage d'information sélectionne et/ou élimine des documents à partir d'un flux dynamique de données ;
- ✍ Le filtrage d'information entraîne le processus de déplacement (Removing) de l'information du flux de données.

Cette comparaison est résumée dans le tableau suivant :

	Recherche d'Information	Filtrage basé sur le contenu
Objectif	Trouver l'information recherchée	Filtrer l'information non désirée
Livraison	Corpus statique ; sur demande (requête temporaire et adverbiale)	Flux dynamique ; livraison si compatibilité entre le besoin et le profil de l'utilisateur
Persistance	Des besoins à court terme	Des intérêts à long terme
Personnalisation	Non personnalisé	Profil d'utilisateur requis
Analyse du contenu	Utilise souvent des mots-clés	Différents et multiples dispositifs utilisés (ex : nombre d'occurrences des mots clés)
Fonctionnalités	Non personnalisé Non adaptatif au changement du profil de l'utilisateur Non dynamique A court terme	Personnalisé S'adapte au changement du profil de l'utilisateur Filtre dynamiquement l'information entrante A long terme

Tableau II.1 – Tableau comparatif des principes de R.I et de F.I fondé sur le contenu.

9 Conclusion

Le filtrage de l'information est un processus de diffusion de documents à l'utilisateur selon une modélisation de ses besoins, cette modélisation est un pas indispensable pour que le système soit filtre efficacement.

Nous avons présenté dans ce chapitre le principe de fonctionnement d'un système de filtrage d'information, son rôle, ses composantes, ..., ainsi nous avons énoncé les différentes formes de filtrage qui sont le filtrage basé sur le contenu, le filtrage hybride et le filtrage collaboratif. Ce dernier a un avantage majeur qui est l'ouverture vers des recommandations inattendues. En effet, pour qu'un utilisateur reçoive un document il suffit qu'un utilisateur proche de lui l'ait jugé intéressant, et cela quelque soit les termes qui indexent le contenu du document. Cette approche de filtrage d'information sera détaillée dans le prochain chapitre.

Chapitre III

Systèmes de filtrage d'information collaboratif

Les systèmes de filtrage ont pour but de distribuer des informations de façon personnalisée aux utilisateurs, tout en s'adaptant en permanence au besoin en information de chacun. Dans un système de filtrage collaboratif, la production de recommandations se base sur des communautés d'utilisateurs qui sont généralement formées conformément au seul critère de proximité des évaluations des utilisateurs sur les recommandations reçues dans le passé. De plus ces communautés restent généralement implicites.

Nous présentons dans ce chapitre un état de l'art des systèmes de filtrage d'informations collaboratif.

1 Introduction

Après avoir en général dans le chapitre précédent, l'objectif et le principe de base d'un système de filtrage d'information, et une visite rapide de l'approche « basé sur le contenu », on détaillant maintenant dans ce présent chapitre, la deuxième approche du filtrage d'information dite « collaborative » et son principe de fonctionnement.

2 Principe général

Le filtrage collaboratif se base sur les opinions d'un groupe de personnes appelé « communauté » pour prévoir qu'un document est qualifié ou non pour un membre du groupe, particulièrement grâce à l'exploitation de la « base de profils » qui contient les « évaluations » (appréciations) fournies explicitement par cet utilisateur sur l'ensemble des documents et les « évaluations » que les autres utilisateurs de la communauté ont données par le passé sur les mêmes documents.

La base de données des « évaluations » des utilisateurs considérée comme une source d'information collaborative qui contient la totalité des appréciations attribuées par les utilisateurs. Ces « appréciations » comprises entre deux valeurs (de 1 à 7 ou de 1 à 10 par exemple) interprétant les divers niveaux d'intérêts des utilisateurs. D'autres systèmes de recommandations mesurent l'intérêt d'une façon implicite comme, par exemple, le temps de consultation d'une page. Ces informations permettent ensuite de rechercher, par comparaison, les utilisateurs ayant des comportements similaires.

Dans la plupart des communautés partageant des centres d'intérêts, les personnes se recommandent ou s'échangent régulièrement des documents parmi leurs amis ou collègues. Ainsi le système doit faire découvrir à des utilisateurs des documents qu'ils ne connaissent pas encore, sans même analyser leur contenu, mais en s'appuyant sur les évaluations faites par d'autres utilisateurs proches. Donc, l'approche collaborative résout les problèmes de l'approche basée sur le contenu sémantique. Pour ce faire, pour chaque utilisateur d'un système de filtrage collaboratif, un ensemble de proches voisins est identifié, et la décision de proposer ou non un document à un utilisateur dépendra des appréciations des membres de son voisinage.

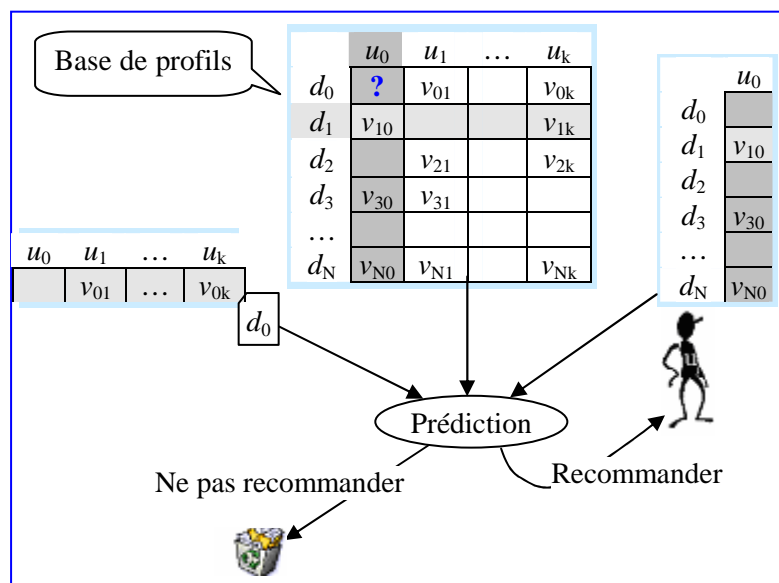


Figure III.1 – Filtrage collaboratif.

Comme le présente la figure III.1, le système de filtrage collaboratif doit calculer la prédiction de l'intérêt pour que le document d_0 être qualifié ou non pour l'utilisateur u_0 , cette opération pourra donc s'effectuer en tenant compte des évaluations existantes relatives au document d_0 (ligne horizontale de la base des profils), ainsi que du profil de l'utilisateur u_0 (colonne verticale de la base des profils). Le document d_0 sera donc recommander pour u_0 , si la valeur prédite dépasse un certain seuil donné fixé auparavant.

3 Architecture générale

Un système de filtrage collaboratif est structuré selon deux principales fonctionnalités : le calcul de la proximité entre les utilisateurs, et le calcul de la prédiction de l'évaluation qu'un document présente pour un utilisateur. La base des profils sera enrichie par la fonctionnalité de mise à jour au fur et à mesure que des nouvelles évaluations arrivent.

Concernant le côté utilisateur, deux interfaces indispensables sont les suivantes :

- Une permettant d'évaluer un document, cette évaluation peut être **explicite** (une note donnée de 1 à 10 par exemple) ou **implicite** (le système interprète les actions et le comportement de l'utilisateur) ;
- Et l'autre permettant de visualiser les documents reçus par le processus du filtrage collaboratif.

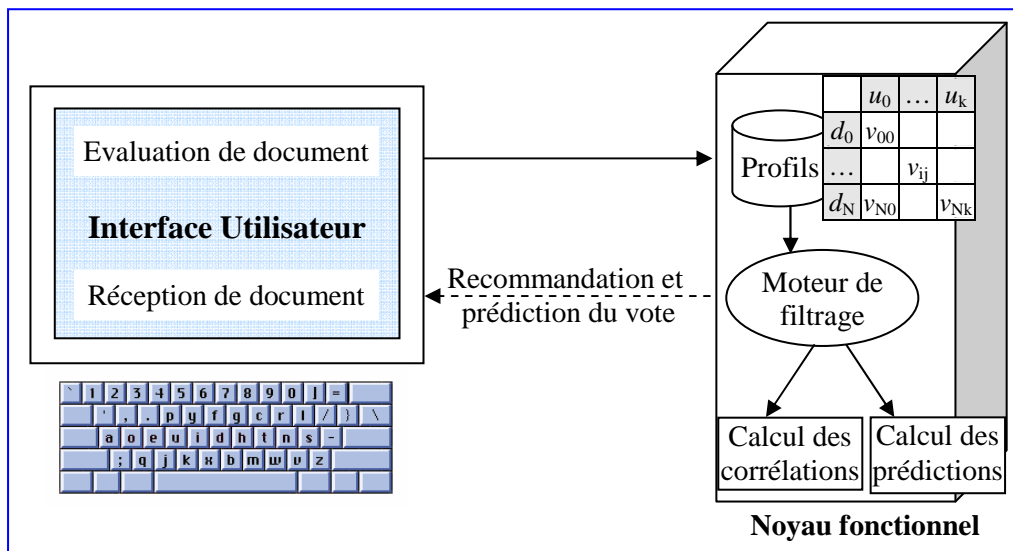


Figure III.2 – Architecture générale d'un système de filtrage collaboratif.

4 Calcul de la prédiction

Pour calculer la prédiction dans un système filtrage collaboratif, deux méthodes sont proposées par Breese et al. [Bre98] : les algorithmes basés « mémoire », et les algorithmes basés « modèle ». Un nouveau procédé est ajouté par Delgado [Del00] : les algorithmes d'apprentissage en ligne.

4.1 Algorithmes basés « mémoire »

Les algorithmes basés « mémoire » utilisent l'ensemble de la base de données des évaluations des utilisateurs pour faire les prédictions : les évaluations d'un utilisateur actif sont prédites à partir d'informations partielles concernant cet utilisateur, et un ensemble de poids calculés à partir de la base de données des évaluations des utilisateurs.

On trouve dans la catégorie basée mémoire, divers algorithmes utilisés pour calculer ce poids, parmi lesquels ceux qui sont basés sur la corrélation (coefficient de corrélation de Pearson) et des autres basés sur la similarité de vecteurs (Cosinus).

L'évaluation prédite sur le document j pour l'utilisateur actif a est une somme pondérée des évaluations des autres utilisateurs, cette évaluation est calculée par la formule suivante :

$$P_{a,j} = \bar{v}_a + k \sum_{i=1}^n w(a,i)(v_{i,j} - \bar{v}_i) \quad (\text{III.1}), \quad \text{avec : } k = \frac{1}{\sum_{i=1}^n |w(a,i)|} \quad (\text{III.2})$$

Avec n est le nombre d'utilisateurs dans la base de données qui ont un poids non nul, et k est un facteur de normalisation. Le poids $w(a,i)$ détermine la distance (la proximité) entre l'utilisateur actif a et les autres utilisateurs $i \in [1..n]$, cette distance est calculée de façon variable, selon l'algorithme.

L'appréciation moyenne pour un utilisateur i peut être définie comme suit :

$$\bar{v}_i = \frac{1}{|I_i|} \sum_{j \in I_i} v_{i,j} \quad (\text{III.3})$$

où I_i est l'ensemble des documents appréciés par l'utilisateur i .

D'une manière textuelle, cette appréciation calculée par la somme de toutes les évaluations données par l'utilisateur i aux différents documents j , sur le nombre total des documents appréciés par cet utilisateur.

Les détails de calcul des poids (proximité entre utilisateurs) donnent lieu à des algorithmes différents. Nous présentons ci-après les formules de deux principaux algorithmes :

1. Algorithme basé sur la corrélation ;
2. Algorithme basé sur la similarité de vecteurs.

Concernant l'algorithme basé sur la corrélation, le poids est calculé comme la corrélation entre les utilisateurs a et i , comme suit :

$$w(a,i) = \frac{\sum_j (v_{a,j} - \bar{v}_a)(v_{i,j} - \bar{v}_i)}{\sqrt{\sum_j (v_{a,j} - \bar{v}_a)^2 \sum_j (v_{i,j} - \bar{v}_i)^2}} \quad (\text{III.4})$$

avec : $w(a,i)$: La distance entre l'utilisateur actif a et l'utilisateur i ;
 $v_{a,j}$: L'évaluation du document j par l'utilisateur actif a ;
 \bar{v}_i : L'évaluation moyenne de l'utilisateur i ;
 $v_{i,j}$: L'évaluation du document j par l'utilisateur i ;
 Les sommes sur les j concernent les documents pour lesquels à la fois a et i ont donné des évaluations ($j \in I_a \cap I_i$).

Concernant l'algorithme basé sur la similarité des vecteurs, le poids est calculé comme un cosinus entre les vecteurs formés par les évaluations des utilisateurs, comme suit :

$$w(a,i) = \sum_j \frac{v_{a,j}}{\sqrt{\sum_{k \in I_a} v_{a,k}^2}} \frac{v_{i,j}}{\sqrt{\sum_{k \in I_i} v_{i,k}^2}} = \cos(\vec{v}_a, \vec{v}_i) = \frac{\vec{v}_a \cdot \vec{v}_i}{\|\vec{v}_a\| \|\vec{v}_i\|} \quad (\text{III.5})$$

avec : \vec{v}_i : Vecteur formé par les appréciations de l'utilisateur i ;
 $v_{i,j}$: L'évaluation du document j par l'utilisateur i .

Parmi les atouts de la méthode basée « mémoire », nous mentionnons la simplicité de la méthode et l'évolutivité dynamique. Une limite de cette méthode figure dans la complexité combinatoire ($O(m^2n)$ avec n : utilisateurs, m : ressources).

4.2 Algorithmes basés « modèle »

Les algorithmes basés « modèle » utilisent la base de données des évaluations des utilisateurs pour créer ou apprendre un modèle de prédiction via un processus d'apprentissage.

Cette méthode basée sur les modèles probabilistes comme : modèles à base de réseaux Bayésiens, modèles hiérarchiques, ...

Dans un contexte probabiliste, la prédiction d'un document j pour un utilisateur actif a , peut être vue comme un calcul de l'espérance de cette évaluation.

Supposons que les évaluations s'effectuent sur une échelle d'entiers de 0 à m . Alors la valeur prédite est calculée par la formule suivante :

$$P_{a,j} = E(v_{a,j}) = \sum_{i=0}^m i \cdot \Pr(v_{a,j} = i | v_{a,k}, k \in I_a) \quad (\text{III.6})$$

avec : I_a : Ensemble des documents appréciés par l'utilisateur actif a ;
 $0, \dots, m$: Ensemble des valeurs possibles des appréciations ;
 La probabilité exprimée dans la formule, s'interprète comme la probabilité que l'utilisateur a attribue l'évaluation i au document j , sachant les évaluations qu'il a déjà attribuées sur les autres documents.

Cette méthode avoir d'un côté comme avantage, la réduction de la complexité combinatoire, par contre avoir d'un autre côté comme inconvénients, une lente phase d'apprentissage, et en plus le problème d'évolutivité.

D'après les experts du domaine, une approche hybride combine les deux méthodes précédentes (mémoire + modèle) est proposée comme une solution.

4.3 Algorithmes d'apprentissage en ligne [Del00]

Après avoir les deux principales méthodes de calcul de la prédiction dans les systèmes de filtrage collaboratif. Une nouvelle méthode ajoutée par Delgado : « les algorithmes d'apprentissage en ligne », l'idée principale de cette approche est d'associer à chaque utilisateur du système un agent autonome pour calculer la prédiction. Chaque agent sera confronté à un ensemble d'essais avec une prédiction à faire à chaque étape en fonction des agents qui l'entourent et qui ont un comportement similaire, neutre ou opposé à la fonction que cet agent cherche à atteindre.

5 Avantages et inconvénients

5.1 Avantages

Parmi les avantages d'un système de filtrage collaboratif nous citons :

- Permet à l'utilisateur d'exprimer ses centres d'intérêt par une variété de critères de pertinence tels que « goût », « qualité »... . En effet, dès qu'un utilisateur apprécie un document positivement, il certifie que le document traite bien d'un sujet qui l'intéresse et de plus ce document est de bonne qualité, les autres utilisateurs peuvent bénéficier de cette appréciation ;
- Offre une ouverture vers des recommandations inattendues, par la réception d'un document jugé intéressant par un collègue de la communauté, donc, l'utilisateur peut enrichir son profil par un nouveau thème en retournant un avis de pertinence positif sur ce document ;
- La sélection de documents s'appuie sur la base de profils qui traduit les opinions que les utilisateurs ont émis sur les documents [BD03]. Donc, il devient possible de traiter toute forme de contenu et d'indexer des documents pas forcément similaires à ceux déjà reçus.

5.2 Inconvénients

Les difficultés majeures de l'approche collaborative sont :

- Une contrainte majeure des systèmes de filtrage collaboratif est que leur efficacité dépend de la participation active des utilisateurs.
- L'arrivée d'un nouveau document, engendre deux problèmes. D'un côté, il ne peut être diffusé que si un nombre minimum d'appréciations est atteint. D'autre côté, les utilisateurs ayant des goûts peu fréquents risquent de ne pas recevoir de propositions. En réalité, ces deux difficultés sont liées au volume et à la constitution du monde d'utilisateurs.

- Cette approche souffre aussi du problème de démarrage à froid. Les nouveaux utilisateurs débutent avec un profil vide et doivent le constituer à partir de zéro. Même avec un profil de démarrage, une période d'apprentissage est toujours nécessaire avant que le profil ne reflète concrètement les préférences de l'utilisateur. Pendant cette période, le système ne peut pas filtrer efficacement pour le compte de l'utilisateur.
- Un nouveau système de filtrage collaboratif ne peut pas démarrer avec une base de profils vide, l'utilisation du filtrage basé sur le contenu peut être une solution de ce problème.

6 Complémentarités entre approches : collaborative et basée contenu

L'approche collaborative apporte des réponses aux problèmes rencontrés dans le filtrage basé sur le contenu. C'est en cela que ces deux approches se complètent avantageusement. Le tableau suivant synthétise les éléments de comparaison de ces deux approches.

	Filtrage basé sur le contenu	Filtrage collaboratif
Amorçage (démarrage de l'exploitation du système)	Le filtrage peut commencer après l'établissement du profil.	Exige une base de données substantielle et plusieurs évaluations de l'utilisateur avant d'être utilisable.
Qualité de l'information (lisibilité, fiabilité, nouveauté, etc.)	La qualité de l'information n'est pas connue.	La qualité de l'information est connue via des évaluations d'utilisateurs.
Contexte de l'information (domaine d'intérêt)	L'identification du domaine se fait généralement par la co-occurrence des termes dans chaque document.	L'identification du domaine se fait par la différence des domaines d'intérêt des utilisateurs.
Effet « entonnoir »	Le système ne suggère que des documents dont le thème a déjà été évoqué explicitement.	Le système peut suggérer des documents sans rapport explicite avec les thèmes déjà évoqués.

Tableau III.1 – Comparaison entre l'approche collaborative et l'approche basée contenu.

7 Conclusion

Nous avons présenté dans ce chapitre un état de l'art sur les systèmes de filtrage collaboratif.

L'approche collaborative compare les utilisateurs entre eux sur la base de leurs jugements passés pour créer des communautés, donc, elle permet de faire bénéficier chacun des opinions des autres et chaque utilisateur reçoit les documents jugés pertinents par sa communauté. Cette dernière reste généralement implicite.

En effet, dans la plupart des systèmes de filtrage collaboratif existants, les communautés sont généralement construites selon un seul critère, et donc il existe un seul espace de communautés créé généralement par la proximité des évaluations

explicites ou implicites des utilisateurs sur les recommandations déjà reçues. Pourtant, on trouve une multiplicité de critères sur lesquels appuyer la formation des communautés, à savoir : informations personnelles, domaines d'intérêt, historique des évaluations, préférences de sécurité, etc. [AS99, BK05]. Il s'agit de l'existence de plusieurs espaces de communautés à la fois, et donc chaque utilisateur appartient à plusieurs communautés en même temps. Cette multiplicité d'appartenance permet d'enrichir et de diversifier les recommandations générées pour les utilisateurs. Un utilisateur peut donc recevoir les recommandations de chacune de ses communautés.

Le problème de la formation d'espaces de communautés selon des critères variés et l'exploitation de ces espaces pour générer les recommandations dans un système de filtrage collaboratif est une vraie problématique. Cette dernière sera traitée dans les chapitres qui suivent.

Chapitre IV

Méthodes de formation de communautés

L'objectif principal de la formation de communautés est de regrouper les utilisateurs en fonction d'un critère bien déterminé afin de générer des recommandations collaboratives. Un utilisateur peut appartenir à diverses communautés selon la multiplicité des critères de formation de communautés. Généralement la qualité d'un système de filtrage collaboratif, est conditionnée par la qualité de la formation des communautés.

La procédure de la formation de ces communautés se base sur plusieurs méthodes. Dans ce chapitre, nous présentons les méthodes les plus répandues.

1 Introduction

La majorité des systèmes de recommandation se basent partiellement ou totalement sur le filtrage collaboratif, en raison de ses importants avantages tels que la diversité de critères de satisfaction qui vont au-delà du centre d'intérêt de l'utilisateur (goûts, qualité,...), la possibilité de traiter n'importe quelle forme de contenu, l'ouverture vers des recommandations inattendues (chaque utilisateur reçoit les documents jugés pertinents par sa communauté), l'indexation des documents non requise, etc. D'un autre côté, ce type de filtrage présente certains inconvénients qui ne sont pas négligeables tels que le *démarrage à froid*, la *masse critique* et le *rapport coût-bénéfice*.

On distingue trois types de problèmes de *démarrage à froid* [Bur02] :

- Un nouveau système « *new system* » [MAS+02], aucune information disponible sur laquelle baser le processus de filtrage personnalisé. Ce problème est généralement traité en exploitant des données externes, données dont on ne dispose pas toujours, selon le cadre applicatif [MSR04].
- L'arrivée d'un nouveau document « *new item* » qui ne peut être diffusé aux utilisateurs concernés parce qu'il n'a pas encore été évalué. Généralement, ce problème est traité par la combinaison du filtrage basé sur le contenu avec le filtrage collaboratif (filtrage hybride), où les profils sont orientés contenu, et la comparaison entre ces profils donne lieu à la formation de communautés permettant le filtrage collaboratif.
- Un nouvel utilisateur « *new user* » commence par un profil vide ou inexistant et ses communautés sont encore inconnues, ce qui conduit à des mauvaises recommandations.

Le filtrage collaboratif souffre aussi du problème de la *masse critique*. En effet, pour comparer les utilisateurs entre eux et former de bonnes communautés, le système exige un nombre suffisant d'appréciations en commun.

Le *rapport coût-bénéfice* est représenté par les efforts fournis par l'utilisateur pour évaluer les documents, et ses attentes de retour de recommandations pertinentes à court ou à moyen terme.

Les systèmes de filtrage collaboratif comparent les utilisateurs entre eux sur la base de leurs jugements passés pour créer des communautés, et chaque utilisateur reçoit les documents jugés pertinents par sa communauté [BHK98].

Le processus de formation de communautés est le noyau d'un système de filtrage collaboratif. Le système doit identifier la communauté (la classe) à laquelle appartient chaque utilisateur à partir d'un critère donné. Dans ce cadre, plusieurs méthodes existent pour ce processus.

Ce chapitre est organisé principalement en deux parties : une première partie introduit les notions de base fréquemment utilisées dans le domaine des systèmes de filtrage d'information collaboratif, la deuxième partie présente les méthodes de formation de communautés d'utilisateurs les plus répandues dans ces systèmes, et nous terminons ce chapitre par une conclusion.

2 Quelques notions de base

- **Communauté :**

Une communauté est composée d'utilisateurs qui sont proches les uns des autres relativement à un critère particulier, par exemple l'historique des évaluations des utilisateurs, les informations personnelles, le critère géographique,... afin que le système calcule des recommandations [PGF03].

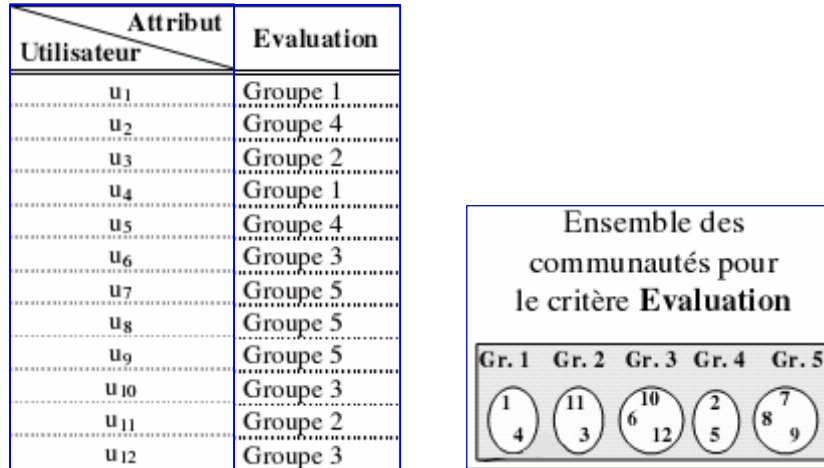


Figure IV.1 – Exemple de communautés pour le critère « *Evaluation* ».

- **Formation de communautés :** résultat de l'estimation de la proximité entre utilisateurs via la comparaison de leurs profils. Les communautés ne sont pas toujours de façon explicite dans les systèmes. Généralement, la formation de communautés reste implicite, et n'intervient qu'au moment de calcul de prédiction. Ces communautés sont formées selon un seul critère [PGF03], comme la proximité des évaluations des utilisateurs.

- **Calcul de proximité entre utilisateurs :**

Pour chaque critère de formation des communautés, il faut définir une mesure de proximité entre les utilisateurs. Pour les critères simples, la formation des communautés est simplement le regroupement des utilisateurs par comparaison des valeurs des données dans les profils. Bien que les données des critères simples puissent être autant quantitatives que qualitatives, la distance entre utilisateurs dans une communauté d'un tel critère n'est guère significative. Par exemple, on peut regrouper deux utilisateurs de 12 et 15 ans dans une communauté des adolescents, mais l'écart d'âge entre eux nous donne peu de signification.

- **Espace de communautés :**

Dans la plupart des systèmes de filtrage collaboratif existants, les communautés des utilisateurs sont généralement créés selon le seul critère de la proximité des évaluations, il existe donc un seul ensemble ou espace de communautés Ω . Pourtant, on trouve une multiplicité de critères sur lesquels appuyer la construction des communautés, tels que : informations personnelles, centres d'intérêt, préférences de livraison et de sécurité, etc. [BK05]. Il s'agit de l'existence de plusieurs espaces de communautés Ω_a , $a \in A$ (A : ensemble de critères) à la fois.

▪ **Vecteur de positionnement :**

Chaque utilisateur d'un système de filtrage collaboratif est rattaché à une communauté dans chacun des espaces Ω_a , $a \in A$. Nous appelons *vecteur de positionnement* de l'utilisateur u , noté P_u , la liste des étiquettes de ses propres communautés selon chaque critère.

▪ **Table de communautés :**

Une table de communautés $T_{m \times n}$ représentant les espaces de communautés est caractérisée par une paire de deux ensembles non vides : $T_{m \times n} = \langle U, A \rangle$

où U est l'ensemble des utilisateurs, et A est l'ensemble des critères de formation de communautés (Tableau IV.1).

		$T a_j$				
		a_1	...	a_j	...	a_n
$T[u_i]$	<i>Critère A</i> <i>Utilisateur</i>					
	u_1					
	...					
	u_i			$T[u_i, a_j]$		
	u_m					

← Vecteur de positionnement P_u

Tableau IV.1 – Table de communautés.

où la valeur $T[u_i, a_j]$ représente l'étiquette de la communauté dans l'espace Ω_j à laquelle l'utilisateur u_i est appartient.

▪ **Classification :** on distingue classiquement deux types de classification : la classification supervisée et la classification non supervisée.

La classification supervisée, consiste à analyser de nouvelles données et à les affecter, en fonction de leurs caractéristiques ou attributs, à telle ou telle classe prédéfinie. Généralement, les méthodes de la classification supervisée sont basées sur un modèle de classes qui doit être connu préalablement, ce modèle est construit par apprentissage sur un ensemble d'échantillons. Parmi les méthodes les plus connues : les k-voisins les plus proches (*K-Nearest Neighbor*), la classification Bayésienne, les arbres de décision, les réseaux de neurones et les machines à vecteur de supports (*Support Vector Machine : SVM*)...

La classification non supervisée (*clustering, segmentation*) ressemble à celle de la classification supervisée, mais diffère dans le sens où il n'existe pas de classes prédéfinies. L'objectif est de grouper des individus qui semblent similaires dans une même classe. La problématique est de trouver une homogénéité dans les groupes (*clusters*) (les individus appartenant à un même cluster doivent être les plus similaires possibles), et hétérogénéité entre groupes (les individus appartenant à différents clusters doivent être les plus dissemblables possibles) dans une population. On peut citer trois algorithmes parmi les plus populaires de cette classification : la classification ascendante hiérarchique (*CAH*), l'algorithme des k-moyennes (*k-means*) et l'algorithme des C-moyennes floues (*fuzzy C-means*), qui sont respectivement les représentants des trois approches de classification hiérarchique, de partitionnement, et de classification floue [JMF99]. Un exemple illustratif d'une classification non supervisée est donné sur la figure IV.2.

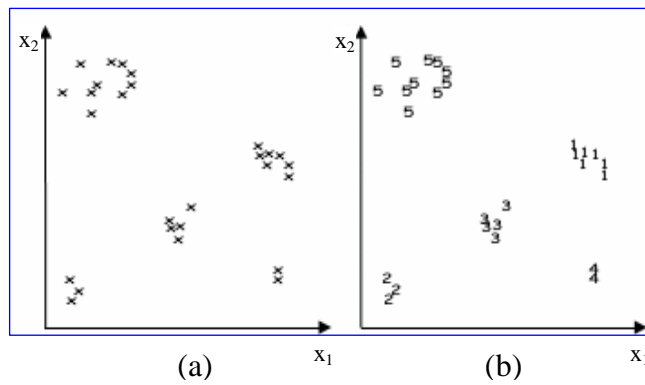


Figure IV.2 – Principes du Clustering.

Les vecteurs d'entrée sont décrits sur la figure (IV.2.a) et les clusters désirés sur la figure (IV.2.b).

3 Méthodes de classification supervisées

Dans cette partie, nous présentons les méthodes de classification supervisée les plus répandues. Ces méthodes sont basées sur un modèle de classes prédéfini, ce modèle est construit par apprentissage dit « supervisé » à partir d'un ensemble d'échantillons.

3.1 Approche des k-voisins les plus proches « K-NN »

L'approche des *k-voisins les plus proches* « *K-Nearest Neighbor* » est une méthode très connue dans le domaine de la classification supervisée, pour la formation de communautés d'utilisateurs [BHK98], [RIS+94]. Afin de sélectionner les utilisateurs les plus proches d'un utilisateur donné, le système doit traiter la matrice des évaluations $V_{m \times n}$ qui contient les évaluations fournies par les utilisateurs (*lignes*) sur les documents (*colonnes*) (Figure IV.3).

	d_1	...	d_j	...	d_n
u_1	$v_{1,1}$		$v_{1,j}$		$v_{1,n}$
...					
u_i	$v_{i,1}$		$v_{i,j}$		$v_{i,n}$
...					
u_m	$v_{m,1}$		$v_{m,j}$		$v_{m,n}$

où $v_{i,j}$: évaluation de u_i sur d_j .

Figure IV.3 – Matrice des évaluations $V_{m \times n}$.

Généralement, cette approche est réalisée en deux étapes :

1. La première étape concernant le calcul de (dis)similarité entre l'utilisateur u et les autres, et pour cela, plusieurs mesures existent telles que la corrélation de Pearson, la corrélation de Spearman, le cosinus, et autres possibilités [BHK98]. En plus, la matrice $V_{m \times n}$ est traitée par paires de lignes (V_u, V_i), par exemple avec la corrélation de Pearson la plus utilisée en raison de sa performance dans le calcul de prédiction [Her00] :

$$\text{corrélation}(V_u, V_i) = \frac{\sum_j (v_{u,j} - \bar{v}_u) \cdot (v_{i,j} - \bar{v}_i)}{\sqrt{\sum_j (v_{u,j} - \bar{v}_u)^2 \cdot \sum_j (v_{i,j} - \bar{v}_i)^2}} \quad (\text{IV.1})$$

où, \bar{v}_u et \bar{v}_i sont les scores moyens des utilisateurs u et i respectifs.

Avec cette mesure, une table ordonnée par (dis)similarité est obtenue pour l'utilisateur u , comme illustrée ci-dessous :

<i>Utilisateur</i>	$s_i = \text{dissimilarité}(V_u, V_i)$
u_1	s_1
...	...
u_d	s_d
($s_d \leq \delta$)	
...	...
u_k	s_k
($k \text{ voisins les plus proches}$)	
...	...
...	...
u_t	s_t

Tableau IV.2 – Table ordonnée des dissimilarités entre l'utilisateur u et tous les autres.
($s_i \leq s_j, i \leq j$)

2. La deuxième étape consiste à sélectionner les utilisateurs les plus proches à l'utilisateur u , en se basant sur la table résultat de l'étape précédente. Pour cela, deux stratégies possibles se présentent :

La première stratégie, consiste à utiliser un seuil δ pour la proximité entre utilisateurs (Tableau IV.2 et Figure IV.4). Cette méthode permet de contrôler la qualité des communautés, mais la taille des communautés risque d'être très faible pour calculer la prédiction, si le seuil de proximité δ est trop fort.

La deuxième stratégie permette de fixer la taille maximale k de l'ensemble des voisins (k -voisins les plus proches) [RIS+94]. Par son expérience, Herlocker propose un seuil k qui varie de 20 à 50 en raison de la précision des prédictions [Her00].

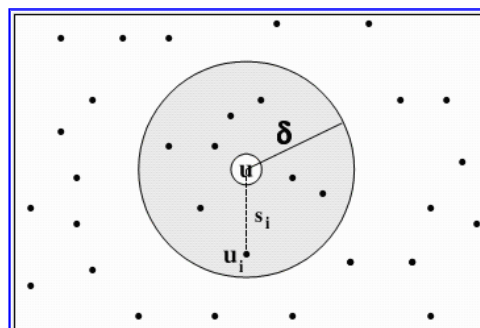


Figure IV.4 – Illustration de sélection des voisins les plus proches par le seuil δ (en 2D).

En résumé, la méthode des voisins les plus proches exploite les relations explicitement disponibles dans les profils des utilisateurs. Ainsi, cette méthode de

classification soit parmi les plus simples, elle fait partie des plus efficaces dans la plupart des cas. Mais, le problème de cette approche est qu'elle est coûteuse en terme de temps de classification, étant donné qu'il faut pour chaque utilisateur calculer sa proximité à tous les autres, et le système doit régénérer la matrice entière des évaluations afin de former des communautés dès qu'un nouvel utilisateur arrive.

3.2 Arbres de Décision « *Decision Trees* »

Parmi les méthodes de classification supervisée utilisées pour former les communautés dans un système de filtrage collaboratif, on trouve également la technique des *arbres de décision*, qui a pour but de classer les individus (les utilisateurs) par une division hiérarchique en classes, selon un ensemble de variables discriminantes (âge, catégorie socio-professionnelle, ...), [Mit97]. Il y a deux types de nœuds dans un tel arbre :

- Les feuilles sont les étiquettes des classes (communautés), et
- Les nœuds non feuilles y compris la racine sont les attributs de test.

Chaque branche sortant d'un nœud correspond à une valeur possible du nœud (Figure IV.5).

Pour construire un arbre de décision par apprentissage à partir d'un ensemble d'exemples, on cherche à chaque pas l'attribut le plus discriminant pour les exemples en utilisant une fonction de qualité. Une fois l'attribut le plus discriminant choisi, si les exemples concernés sont dans une même classe, le nœud contextuel devient une feuille. Sinon, l'algorithme itère jusqu'à ce qu'un des tests d'arrêt suivants soit satisfait :

- Tous les exemples restants sont dans une même classe, ou
- Il n'y a plus d'attribut, ou
- Il n'y a plus d'exemple dans l'ensemble d'apprentissage.

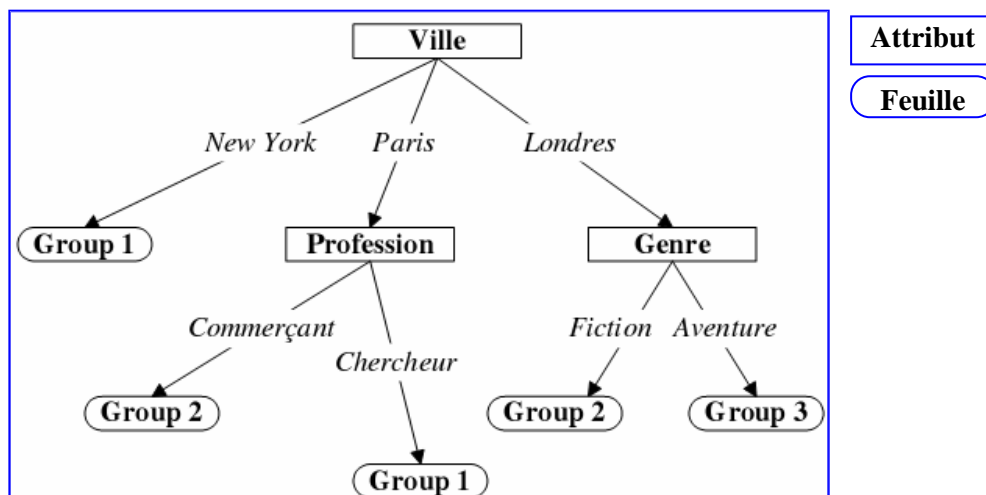


Figure IV.5 – Exemple d'un arbre de décision ($D = \{Evaluation\}$).

ID3 (*Inductive Decision Tree*) est l'algorithme le plus connu pour construire un arbre de décision [Qui86]. Dans cet algorithme, le choix de l'attribut le plus discriminant à chaque pas s'appuie sur l'entropie d'information [Sha48].

Soient l'attribut a avec le domaine $V_a = \{va_i\}$ et l'ensemble des exemples X . Alors, l'entropie de l'ensemble X par rapport à l'attribut a est mesurée par :

$$Entropie(X, a) = -\sum_{i \in I} |X_i| \cdot \log(|X_i|) \quad (IV.2)$$

où $\{X_i, i \in I\}$ est l'ensemble des exemples de X qui prennent va_i pour valeur pour l'attribut a .

Ensuite, on définit le gain d'information de X par rapport à l'attribut a comme suit :

$$Gain(X, a) = Entropie(X, a) - \sum_{i \in I} \frac{|X_i|}{|X|} \cdot Entropie(X_i, a) \quad (IV.3)$$

Plusieurs algorithmes ont été proposés, notamment *CART* (*Classification And Regression Tree*), en 1984 par Breiman et al., l'algorithme *ID3* de R. Quinlan proposé en 1986 qui a été raffiné par la suite (*C4.5* puis *C5*) [Qui93]. Il est constaté expérimentalement que ces algorithmes sont très performants : ils construisent rapidement des arbres de décision qui prédisent avec une assez grande fiabilité la classe de nouvelles données, autres algorithmes tels que : *QUEST* (*Quick, Unbiased, Efficient Statistical Trees*), *CHAID* (*CHi-square Automatic Interaction Detection*).

Le principe de l'algorithme *ID3* pour déterminer l'attribut à placer à la racine de l'arbre de décision peut maintenant être exprimé : rechercher l'attribut qui possède le gain d'information maximum, le placer en racine, et itérer pour chaque fils, c'est-à-dire pour chaque valeur de l'attribut.

Le principe de classification d'une nouvelle observation par les arbres de décision est que l'on commence les tests par la racine, et que l'on suit ensuite le chemin, jusqu'à ce que l'on atteigne une feuille particulière, et cette feuille est désignée comme la classe prédite pour la nouvelle observation.

Il faut noter que l'on peut considérer les arbres de décision comme une méthode de classification par règles, car on peut facilement transformer un arbre de décision en un ensemble de règles. En effet, on prend chaque chemin reliant la racine et une feuille en combinant les tests des attributs dans le chemin pour générer une règle de classification. Par exemple, l'arbre de décision dans la figure IV.5 nous donne les règles suivantes :

- (Ville = « New York ») → (Evaluation = « Groupe 1 ») ;
- (Ville = « Paris », Profession = « Commerçant ») → (Evaluation = « Groupe 2 ») ;
- (Ville = « Paris », Profession = « Chercheur ») → (Evaluation = « Groupe 1 ») ;
- (Ville = « Londres », Genre = « Fiction ») → (Evaluation = « Groupe 2 ») ;
- (Ville = « Londres », Genre = « Aventure ») → (Evaluation = « Groupe 3 »).

Pour conclure, la technique des arbres de décision est autant populaire en statistique qu'en apprentissage automatique. Son succès réside en grande partie à ses caractéristiques :

- **Lisibilité** du modèle de prédiction. Cette caractéristique est très importante, car le travail de l'analyste consiste aussi à faire comprendre ses résultats afin d'emporter l'adhésion des décideurs.

- **Capacité** à sélectionner automatiquement les variables discriminantes dans un fichier de données contenant un très grand nombre de variables potentiellement intéressantes. En ce sens, un arbre de décision constitue une technique exploratoire privilégiée pour appréhender de gros fichiers de données.

3.3 Approche probabiliste

Dans cette approche, et à partir des évaluations fournies par les utilisateurs, le système essaie de construire par apprentissage un modèle probabiliste, afin de l'appliquer et prévoir la satisfaction d'un utilisateur pour un document donné. Par exemple *like* ou *dislike*, de l'utilisateur sur un document qu'il n'a pas encore évalué. L'approche probabiliste est moins utilisée pour la formation de communautés d'utilisateurs dans un système de filtrage collaboratif [BHK98].

Miyahara et Pazzani ont proposé d'utiliser la classification Bayésienne naïve. Dans cette méthode de classification supervisée, les utilisateurs sont considérés comme des attributs ou caractéristiques de données [MP00], et la formation de communautés est basée sur la sélection des caractéristiques les plus discriminantes (*Feature Selection*). Par exemple, Miyahara et Pazzani utilisent une matrice des évaluations binaires $V_{m \times n}$ (Figure IV.6), où chaque ligne V_i de la matrice est divisée en deux lignes $V'_{i,like}$ et $V'_{i,dislike}$ d'une nouvelle matrice V' (Figure IV.7), sauf la ligne V_4 de l'utilisateur dont on veut prédire la satisfaction sur le document d_5 .

	d_1	d_2	d_3	d_4	d_5
u_1	like	dislike	dislike		like
u_2	dislike			dislike	dislike
u_3		like	like		like
u_4	like	dislike	like	like	?

Figure IV.6 – Matrice des évaluations binaires $V_{m \times n}$.

	d_1	d_2	d_3	d_4	d_5
$f_1 = (u_1, \text{like})$	1	0	0	0	1
$f_2 = (u_1, \text{dislike})$	0	1	1	0	0
$f_3 = (u_2, \text{like})$	0	0	0	0	0
$f_4 = (u_2, \text{dislike})$	1	0	0	1	1
$f_5 = (u_3, \text{like})$	0	1	1	0	1
$f_6 = (u_3, \text{dislike})$	0	0	0	0	0
u_4	like	dislike	like	like	?

Figure IV.7 – Matrice de transformation V' .

Pour déduire la valeur manquante dans la matrice V' , et sur la base de l'hypothèse de l'indépendance des attributs, ou caractéristiques dans la classification Bayésienne naïve, on utilise la formule suivante :

$$\Pr(C) \prod_{s=1}^t \Pr(C|f_s) \quad (\text{IV.5})$$

qui est proportionnelle à la formule :

$$\Pr(C|f_1, f_2, \dots, f_t) \quad (\text{IV.6})$$

Donc, dans notre exemple :

$$\Pr(C|f_1, f_2, \dots, f_6) \cong \Pr(C) \prod_{s=1}^6 \Pr(C|f_s) \quad (\text{IV.7})$$

où, C est la classe à prédire (*like*, *dislike*), et f_s est une caractéristique (Figure IV.7).

Rappels de probabilités :

- A et B deux événements indépendants $\Leftrightarrow \Pr(A|B) = \Pr(A)$ (IV.8)

$$\Leftrightarrow \Pr(B|A) = \Pr(B) \quad (\text{IV.9})$$

- Formule de Bayes : $\Pr(A|B) = \frac{\Pr(B|A) \Pr(A)}{\Pr(B)}$ (IV.10)

En effet, le regroupement des utilisateurs s'appuie sur la sélection des caractéristiques f_s les plus discriminantes pour le modèle de prédiction.

Parmi les atouts de cette technique, la rapidité du calcul de prédiction, par contre cette approche est très compliquée et la période d'apprentissage est souvent longue.

3.4 Réseaux de neurones

Les réseaux de neurones sont l'une des techniques de classification. Il existe plusieurs types de ces réseaux tels que le Perceptron, le Perceptron multicouches,... Un autre type de réseau de neurones, le réseau à auto-organisation (*carte de Kohonen*), est présenté dans la deuxième branche dans le cadre de la classification non supervisée.

Les réseaux de neurones artificiels sont des systèmes numériques permettant la modélisation de processus généraux par l'établissement de modèles fonctionnels. Ces réseaux offrent une panoplie de techniques adaptatives pour de nombreux problèmes génériques : la classification, le classement, la modélisation, la prévision. Les applications de ces techniques sont très stratégiques, notamment pour la fouille de données et la reconnaissance des formes.

Les réseaux de neurones artificiels « *RNA* » sont des assemblages fortement connectés d'unités de calcul, les *neurones formels*. Ces derniers ont pour origine un modèle du *neurone biologique*.

Le neurone formel

Un « réseau de neurones » (neural network), est un réseau de neurones artificiels basé sur un modèle simplifié de neurone « neurone formel ».

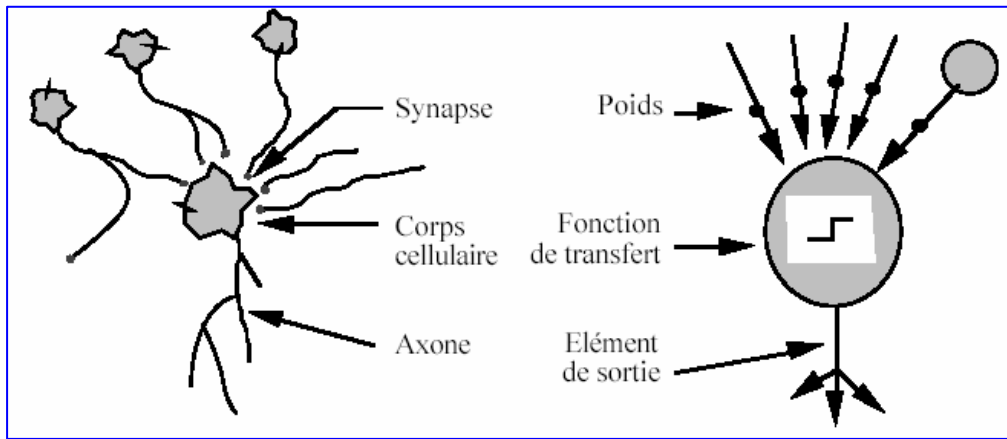


Figure IV.8 – Mise en correspondance neurone biologique / neurone artificiel.

Le neurone formel est un processeur très simple (simulé sur ordinateur ou réalisé sur circuit intégré) imitant grossièrement la structure et le fonctionnement d'un neurone biologique. La première version du neurone formel est celle de W. S. Mc Culloch et W. Pitts, datée de 1943.

C'est donc une modélisation mathématique qui reprend les principes du fonctionnement du neurone biologique, en particulier la sommation des entrées. Sachant qu'au niveau biologique, les synapses n'ont pas toutes la même « valeur », donc les auteurs ont associé une pondération à chaque entrée, cette pondération représente un poids synaptique (coefficient de pondération).

Interprétation mathématique

D'un point de vue mathématique, le neurone formel est une fonction non linéaire, paramétrée, il peut être représenté de la manière suivante :

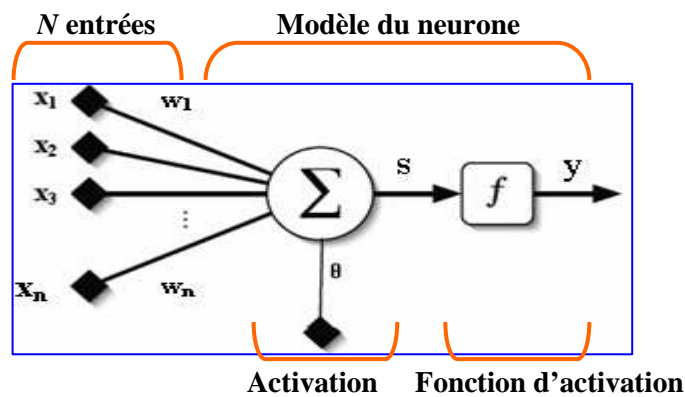


Figure IV.9 – Modèle d'un neurone formel.

Un neurone est considéré comme un élément élémentaire de traitement. Il reçoit les entrées et produit un résultat à la sortie. Il est essentiellement constitué d'un intégrateur qui effectue la somme pour un nombre quelconque N d'entrées (x_1, \dots, x_n), pondérées par les poids synaptiques (w_1, \dots, w_n). Le résultat S de cette somme est ensuite transformé par une fonction de transfert (activation) f qui produit la sortie y du neurone.

La sortie de l'intégrateur est donnée par l'équation suivante :

$$S = \sum_{j=1}^N w_j x_j - \theta \tag{IV.11}$$

$$y = f(S) \tag{IV.12}$$

Notation :

- x_1, \dots, x_n : représente N **entrées** du neurone provenant de l'environnement externe au réseau ou d'autres neurones.
- w_1, \dots, w_n : représente N **poids synaptiques** pour les N entrées du neurone associés à chaque connexion.
- θ : appelé le biais du neurone.
- S : résultat de la somme pondérée, appelé **le niveau d'activation** du neurone.
- f : fonction d'activation ou de transfert.
- y : sortie du neurone.

Un poids d'une entrée représente donc l'efficacité d'une connexion synaptique. Un poids négatif vient inhiber une entrée, alors qu'un poids positif vient l'accentuer.

La fonction de transfert

Plusieurs fonctions de transfert pouvant être utilisées comme fonction d'activation du neurone (Tableau IV.3).










Nom de la fonction	Relation d'entrée/sortie	Icône	Nom Matlab
Seuil	$a = 0 \quad si \quad n < 0$ $a = 1 \quad si \quad n \geq 0$		Hardlim
seuil symétrique	$a = -1 \quad si \quad n < 0$ $a = 1 \quad si \quad n \geq 0$		Hardlims
Linéaire	$a = n$		Purelin
linéaire saturée	$a = 0 \quad si \quad n < 0$ $a = n \quad si \quad 0 \leq n \leq 1$ $a = 1 \quad si \quad n > 1$		Satlin
linéaire saturée symétrique	$a = -1 \quad si \quad n < -1$ $a = n \quad si \quad -1 \leq n \leq 1$ $a = 1 \quad si \quad n > 1$		Satlins
linéaire positive	$a = 0 \quad si \quad n < 0$ $a = n \quad si \quad n \geq 0$		Poslin
Sigmoïde	$a = \frac{1}{1 + \exp^{-n}}$		Logsig
tangente hyperbolique	$a = \frac{e^n - e^{-n}}{e^n + e^{-n}}$		Tansig
Compétitive	$a = 1 \quad si \quad n \text{ maximum}$ $a = 0 \quad autrement$		Compet

Tableau IV.3 – Fonctions de transfert : $a = f(n)$.

Dans sa première version, le neurone formel de Mc Culloch et Pitts était implémenté avec une fonction à seuil, mais de nombreuses versions existent.

Il existe plusieurs modèles de *RNA* tels que le Perceptron, le Perceptron multicouches, mémoires hétéro associatives « *cartes auto-organisatrices de Kohonen* »,...

Le Perceptron

Le Perceptron est le premier *RNA* de Rosenblatt. Il est inspiré du système visuel et de ce fait a été conçu dans un premier but de reconnaissance des formes. Cependant, il peut aussi être utilisé pour faire de la classification et pour résoudre des opérations logiques simples, ... Le Perceptron monocouche est un réseau simple, il possède un neurone qui ne se compose que d'une couche d'entrée « *perceptive* » et d'une couche de sortie « *décisionnelle* ».

Le Perceptron utilise une fonction seuil de Heaviside et les sorties des neurones ne peuvent prendre que deux états (-1 et 1 ou 0 et 1).

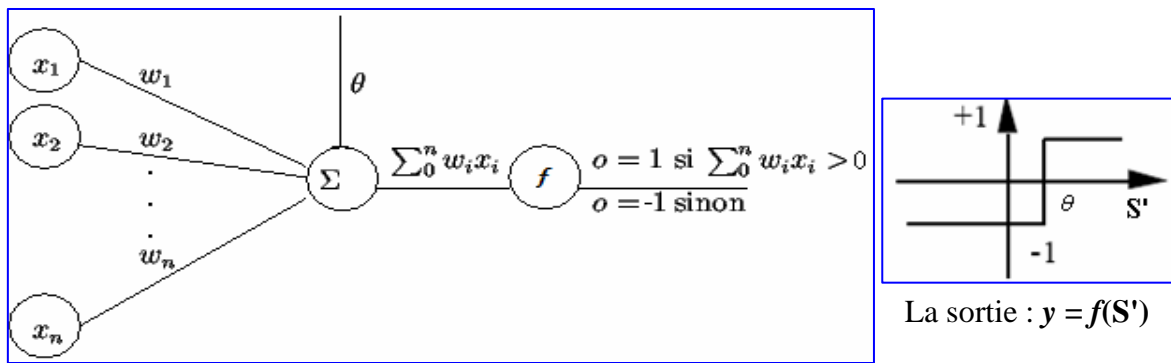


Figure IV.10 – Structure d'un Perceptron monocouche.

avec : $x_0 = -1$ et $w_0 = \theta$

L'objectif est d'entraîner le réseau jusqu'à ce que la valeur obtenue en sortie soit égale à la valeur désirée (Il suit généralement un apprentissage supervisé). Pour cela on utilise la règle de correction de l'erreur puisque les poids ne sont ajustés que si la sortie attendue diffère de la sortie calculée.

Algorithme :

1. Initialisation des poids et du seuil θ à des valeurs (petites) choisies au hasard ;
2. Présentation d'une entrée $X_1 = (x_1, x_2, \dots, x_n)$ de la base d'apprentissage. (1 : est l'indice de l'entrée dans la base d'apprentissage.)
3. Calcul de la sortie y obtenue pour cette entrée :

$$S = \sum_{j=1}^N w_j x_j - \theta$$

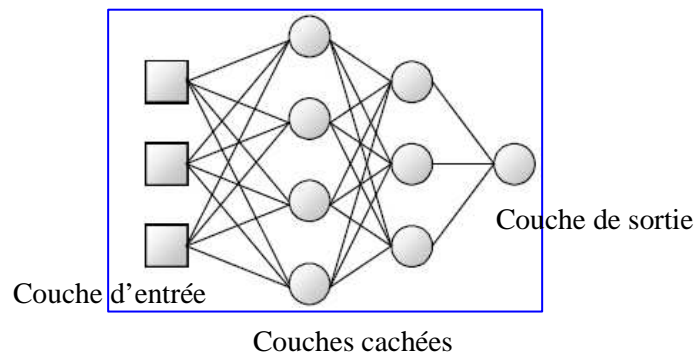
$$y = f(S) = \begin{cases} +1 & S > 0 \\ -1 & S \leq 0 \end{cases}$$
4. Si la sortie y est différente de la sortie désirée d_1 pour l'exemple d'entrée X_1 alors il y a modification des poids :

$$w_j(t+1) = w_j(t) + \Delta w_j(t) = w_j(t) + \eta(d_1 - y) * x_j$$
 avec $(d_1 - y)$: Estimation de l'erreur ; η : Pas d'apprentissage ($0 < \eta < 1$) ;
5. Tant que tous les exemples de la base d'apprentissage ne sont pas traités correctement, on retourne à l'étape 2.

Algorithme IV.1 – Algorithme d'apprentissage d'un Perceptron.

L'apprentissage par Perceptron est rien d'autre qu'une technique de séparation linéaire. Outre, il est bien évident que la plupart des problèmes d'apprentissage qui se posent naturellement ne peuvent pas être résolus par des méthodes aussi simples. Une manière de résoudre cette difficulté serait soit de mettre au point des séparateurs non linéaires. C'est ce que permettent de faire les réseaux multicouches.

Le Perceptron multicouche *PMC* (*Multi-Layer Perceptron MLP*) est un assemblage de couches concaténées les unes aux autres, de la gauche vers la droite, en prenant les sorties d'une couche et en les injectant comme les entrées de la couche suivante. Chaque neurone d'une couche intermédiaire (couche cachée) est connecté à tous les neurones de la couche précédente et ceux de la couche suivante et il n'y a pas de connexions entre les cellules d'une même couche.

**Figure IV.11** – Exemple d'un réseau multicouche.

L'apprentissage dans un Perceptron peut donc se décomposer en quatre phases :

- La propagation du signal de la couche d'entrée jusqu'à la couche de sortie ;

- Le calcul de l'erreur en sortie ;
- La rétro-propagation de l'erreur ;
- Correction des poids.

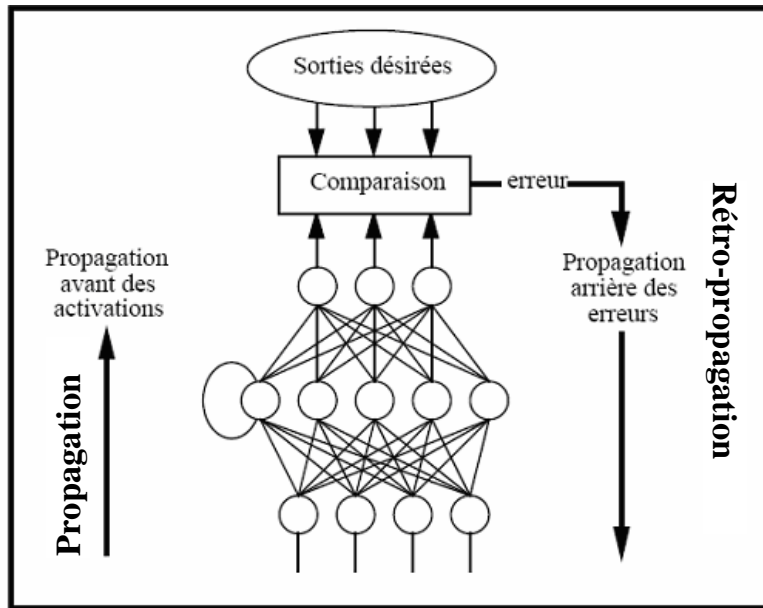


Figure IV.12 – Schéma du modèle de la rétro-propagation de l'erreur.

Atouts et limites

- Bonne capacité de généralisation ;
- Très utilisé pour des tâches difficiles : caractères écrits, analyse signal, ... ;
- Fonctionne bien même en présence de données bruitées ;
- Difficile à utiliser dans la pratique : nécessite beaucoup de savoir-faire et d'expérience ;
- A n'appliquer que sur des problèmes difficiles pour lesquels les autres méthodes ne fonctionnent pas ;
- Calibrage du réseau pas forcément évident (nombre de couches, type de réseau, nombre de neurones par couche, ...) ;
- Très difficile d'extraire un modèle de l'apprentissage effectué par le réseau.

3.5 Machines à vecteurs supports

Les *machines à vecteurs supports* ou *séparateurs à vaste marge* (en anglais *Support Vector Machine, SVM*) sont un ensemble de techniques d'apprentissage supervisé destinées à résoudre des problèmes de discrimination et de régression. Les *SVM* sont une généralisation des classifieurs linéaires.

Les séparateurs à vaste marge reposent sur deux idées clés : la notion de marge maximale et la notion de fonction noyau. En 1992, ces idées seront bien comprises et rassemblées par Boser, Guyon et Vapnik dans un article, qui est l'article fondateur des séparateurs à vaste marge [BIV92].

La première idée clé est la notion de *marge maximale*. La marge est la distance entre la frontière de séparation et les individus les plus proches. Ces derniers sont appelés *vecteurs supports*. Dans les *SVM*, la frontière de séparation est choisie comme

celle qui maximise la marge [Mar98]. Le problème est de trouver la frontière séparatrice optimale, à partir d'un ensemble d'apprentissage.

Afin de pouvoir traiter des cas où les données ne sont pas linéairement séparables, la deuxième idée clé des *SVM* est de transformer l'espace de représentation des données d'entrées en un espace de plus grande dimension, dans lequel il est probable qu'il existe une séparatrice linéaire. Ceci est réalisé grâce à une *fonction noyau*, qui a l'avantage de ne pas nécessiter la connaissance explicite de la transformation à appliquer pour le changement d'espace, mais cette technique de transformation est coûteuse.

Les *SVM* ont été appliqués à de très nombreux domaines (bio-informatique, recherche d'information, vision par ordinateur, finance, ...) [BA02].

Principe général

Les machines à vecteurs supports (*SVM*) constituent l'algorithme de classification discriminatif. Il sépare deux types de données par un hyperplan séparateur de marge maximum dans un espace de dimension supérieure.

Les *SVM* peuvent être utilisés pour résoudre des problèmes de discrimination, c'est-à-dire décider à quelle classe appartient un individu, ou de régression, c'est-à-dire prédire la valeur numérique d'une variable. La résolution de ces deux problèmes passe par la construction d'une fonction f qui à un vecteur d'entrée x fait correspondre une sortie y : $y = f(x)$.

Dans cette partie, nous présentons les notions essentielles des machines à vecteurs supports dans le cas le plus simple, celui où l'on cherche un hyperplan séparateur, c'est-à-dire le cas où les données sont linéairement séparables, donc, ce problème est de discrimination à deux classes (discrimination binaire), c'est-à-dire $y \in \{-1, 1\}$, le vecteur d'entrée x étant dans un espace X muni d'un produit scalaire.

Dans notre cas, le problème est linéairement séparable, donc, comme illustré par la figure IV.13 par exemple, les individus positifs (+) sont séparables des individus négatifs (-) par un hyperplan H . On représente par H_+ l'hyperplan qui contient l'individu positif le plus proche de H , respectivement H_- pour l'individu négatif. H_+ et H_- , tous deux parallèles à H . B est un point de H_+ et A est le point le plus proche de B qui appartient à H_- .

Une *SVM* linéaire recherche l'hyperplan qui sépare les données de manière à ce que la distance entre H_+ et H_- soit la plus grande possible. Cet écart entre les deux hyperplans H_+ et H_- est dénommé la « *marge* ».

Un hyperplan a pour équation : $y = \langle \vec{w}, \vec{x} \rangle + b$, où $\langle \vec{w}, \vec{x} \rangle$ dénote le produit scalaire entre les vecteurs \vec{w} et \vec{x} . Pour une donnée \vec{x} de classe y , on cherche \vec{w} tel que :

$$\begin{cases} \langle \vec{w}, \vec{x} \rangle + b \geq 1 & \text{si } y = +1 \\ \langle \vec{w}, \vec{x} \rangle + b \leq -1 & \text{si } y = -1 \end{cases} \quad (\text{IV.13})$$

Donc, on a : $y(\langle \vec{w}, \vec{x} \rangle + b) - 1 \geq 0$.

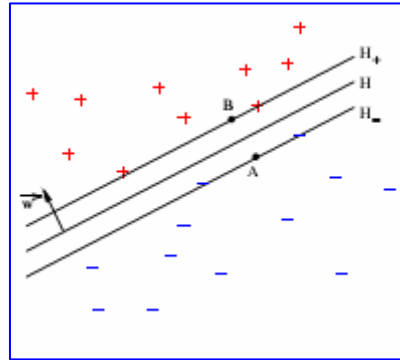


Figure IV.13 – Séparation linéaire dans un espace à deux dimensions.

L'objectif est de maximiser la largeur de la marge. Calculons-la :

1. Le vecteur \vec{w} est perpendiculaire à l'hyperplan H (mathématiques élémentaires) ;
2. Soit B un point de H_+ et A le point le plus proche de B sur H_- ;
3. Pour tout point O , on a :

$$\vec{OB} = \vec{OA} + \vec{AB} ;$$

4. Par définition des points A et B , \vec{AB} est parallèle à \vec{w} . Donc, il existe $\lambda \in \mathfrak{R}$ tel que :

$$\vec{AB} = \lambda \vec{w} ;$$

Soit $\vec{OB} = \vec{OA} + \lambda \vec{w} ;$

5. On veut que A, B, H_- et H_+ soient tels que :

$$\begin{cases} B \in H_+ \Rightarrow \langle \vec{w}, \vec{OB} \rangle + b = 1 \\ A \in H_- \Rightarrow \langle \vec{w}, \vec{OA} \rangle + b = -1 \end{cases}$$

6. Donc : $\langle \vec{w}, (\vec{OA} + \lambda \vec{w}) \rangle + b = 1 ;$

7. Donc : $\underbrace{\langle \vec{w}, \vec{OA} \rangle + b}_{-1} + \langle \vec{w}, \lambda \vec{w} \rangle = 1 ;$

8. Soit : $-1 + \underbrace{\langle \vec{w}, \lambda \vec{w} \rangle}_{\lambda \langle \vec{w}, \vec{w} \rangle} = 1 ;$

9. Donc : $\lambda = \frac{2}{\langle \vec{w}, \vec{w} \rangle} = \frac{2}{\|\vec{w}\|^2} ;$

10. Donc : $|\lambda| = \frac{2}{\|\vec{w}\|^2} .$

La largeur de la marge est $\|\lambda \vec{w}\|$, \vec{w} en donnant la direction et $|\lambda|$ son amplitude.
 Conclusion : on veut maximiser la marge ; donc, on doit minimiser la norme de \vec{w} .
 Donc minimiser $\|\vec{w}\|$, c'est la même chose que minimiser $\|\vec{w}\|^2$.

De plus, on veut vérifier les contraintes :

$$\gamma_i = y_i (\langle \vec{w}, \vec{x}_i \rangle + b) - 1 \geq 0 \quad \forall i \in \{1, \dots, N\}.$$

Cela est donc un problème d'optimisation non linéaire (minimiser $\|\vec{w}\|^2$) avec contraintes (les γ_i) que l'on résolvait habituellement par la méthode de Lagrange [Fou04].

4 Méthodes de classification non supervisées

Dans cette partie, nous présentons les méthodes de classification non supervisée les plus répandues. Dans cette catégorie de classification, il n'existe pas de classes préalablement définies. L'apprentissage est dit « non supervisé ».

4.1 Réseaux sociaux

La construction des communautés d'utilisateurs est basée sur les relations implicites qui existent entre utilisateurs dans un réseau social. Cette approche est généralement composée de trois phases :

- Collecter et fouiller des données transactionnelles, par exemple communication, messages, favoris, évaluations, etc.,
- Reconnaître et modéliser des intérêts souvent implicites, et induire les communautés existantes, et
- Explorer et exploiter les communautés induites.

On peut définir un *réseau social* comme un graphe non orienté, les nœuds sont des éléments de la classe personne ou d'autre classe d'objets (Documents,...), et les relations entre ces nœuds constituent les arcs qui doivent être de même nature (même type). Par exemple la relation implicite « être ami » entre les utilisateurs. Afin de former les communautés, en se basant sur le décèlement des relations sociales existantes dans le graphe.

Généralement, les études des réseaux sociaux visent à détecter les relations sociales existantes dans les données plutôt qu'à modéliser explicitement les intérêts des utilisateurs [PGF03].

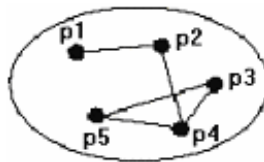


Figure IV.14 – Exemple d'un réseau social de 5 personnes, par la relation « être ami ».

Dans cette approche, la matrice des évaluations $V_{m \times n}$ est représentée par un réseau biparti R (Figure IV.15), comporte deux classes de nœuds {personne p_i } et {document d_j }, en reliant les documents avec les personnes qui les évaluent. Puis, le système transforme ce réseau en un réseau social G_s dont ses nœuds appartiennent à l'unique classe des personnes, et enfin, ces deux réseaux sont rattachés en un troisième réseau G_r pour l'explorer et l'exploiter dans la production de recommandations [MKR03].

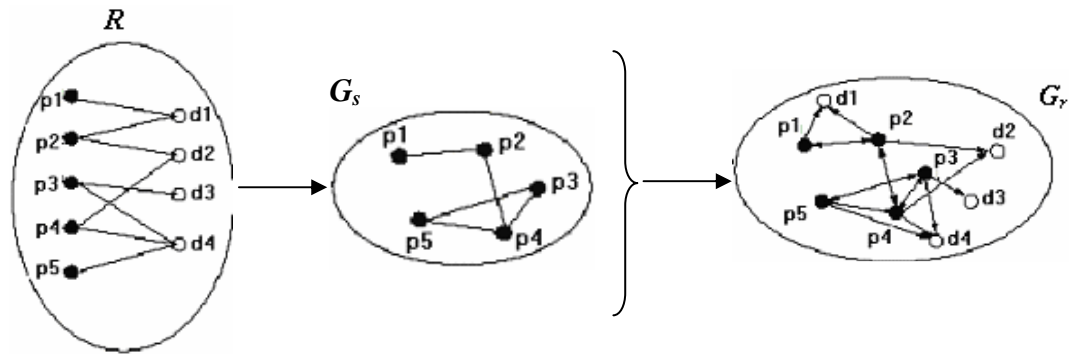


Figure IV.15 – Construction des réseaux G_s et G_r pour la production de recommandations.

Les relations sociales qui existent entre les utilisateurs, nous ont permis de fournir une solution pour dépasser le problème de la masse critique. Cette solution représentée par la formation des communautés par transitivité sans tenir compte des évaluations en commun entre utilisateurs, mais cette notion de communautés est beaucoup moins forte que dans l'approche des voisins les plus proches [Ngu06].

4.2 Classification Ascendante Hiérarchique « CAH »

La *classification ascendante hiérarchique* « CAH » est l'une des méthodes de classification non supervisée qui consiste à regrouper les individus ayant un comportement similaire en classes en fonction de deux critères : les individus d'une même classe sont le plus semblables possibles, et les classes sont les plus disjointes possibles.

Les méthodes ascendantes sont agglomératives ; à chaque étape du processus, on crée une partition en agrégeant deux à deux les individus, ou les groupes d'individus les plus proches. Pour un niveau de précision plus élevé, deux individus peuvent être confondus dans le même groupe.

Soient l'algorithme suivant de l'approche CAH :

- Au départ, on dispose de m éléments (utilisateurs ou groupe d'utilisateurs) à regrouper ;
- On calcule les distances les séparant deux à deux ;
- On agrège les deux éléments les plus proches en un nouvel élément ;
- On se retrouve avec $(m-1)$ éléments à classer ;
- On calcule les distances entre ce nouvel élément et les $(m-2)$ éléments restants ;
- On cherche, de nouveau, les deux éléments les plus proches ;
- On réitère le processus jusqu'à ce que tous les éléments aient été regroupés.

4.3 Algorithme des k -moyennes « k -means »

L'algorithme des « k -moyennes » est l'une des techniques de classification non supervisée « *clustering* » les plus utilisées, et le représentant de la classification par partitionnement.

Le principe de l'algorithme des k -moyennes, est de classifier (partitionner) des objets en k « classes », « groupes » ou « *clusters* » ne chevauchant pas (k : prédéfini).

En minimisant la variance intra-classe et en maximisant l'écartement inter-classes [JMF99]. Cet algorithme commence par positionner au hasard k centres de gravité dans l'espace d'objets à partitionner. Afin de former les classes initiales autour de ces centres, en affectant chaque objet à la classe dont le centre de gravité est le plus proche (règle de la « *Distance Minimale* »). A chaque itération, on recalcule les centres en fonction de la variance intra-classe, et on construit les nouvelles classes jusqu'à ce que l'on n'obtienne plus de changement de partition.

4.4 Algorithme des C-moyennes floues « *Fuzzy C-means* »

L'algorithme des k -moyennes est l'une des méthodes les plus connues parmi les techniques de classification non supervisée. La version *C-moyennes floues* est une extension directe de cet algorithme, où l'on introduit la notion d'ensemble flou dans la définition des classes. Cet algorithme utilise un critère de minimisation des distances intra-classes et de maximisation des distances inter-classes, mais en tenant compte des degrés d'appartenance des individus.

Soit la matrice $W_{n \times c}$ dont tous les éléments u_{ij} appartiennent à l'intervalle $[0,1]$ et dont la somme des éléments d'une ligne quelconque vaut 1.

$$\forall i, \forall j, u_{ij} \in [0,1] \tag{IV.14}$$

$$\forall i, \sum_{j=1}^c u_{ij} = 1 \tag{IV.15}$$

Etant donné n individus, et un entier c tel que $2 \leq c < n$, une partition floue des n individus peut être représentée par une matrice dont les éléments sont définis comme suit :

$$\begin{pmatrix} u_{11} & u_{12} & \dots & u_{1j} & \dots & u_{1c} \\ u_{21} & u_{22} & \dots & u_{2j} & \dots & u_{2c} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ u_{i1} & u_{i2} & \dots & u_{ij} & \dots & u_{ic} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ u_{n1} & u_{n2} & \dots & u_{nj} & \dots & u_{nc} \end{pmatrix}$$

- 1- La $i^{ème}$ ligne de la matrice, $U_i = (u_{i1}, u_{i2}, \dots, u_{ic})$ contient les c degrés d'appartenance du $i^{ème}$ individu aux c sous-ensembles flous ;
- 2- La $j^{ème}$ colonne $U_j = (u_{1j}, u_{2j}, \dots, u_{nj})$ contient les n degrés d'appartenance des individus au $j^{ème}$ sous-ensemble flou ;
- 3- La somme de tous les degrés d'appartenance d'un individu quelconque étant égale à 1, la somme de tous les éléments d'une même ligne vaut par conséquent 1 :

$$\forall i \in \{1, 2, \dots, n\}, \sum_{j=1}^c u_{ij} = 1 ;$$

- 4- Aucun sous-ensemble flou ne doit être vide, par conséquent, la somme de tous les éléments d'une même colonne doit être strictement supérieure à 0 :

$$\forall j \in \{1, 2, \dots, c\}, \sum_{i=1}^n u_{ij} > 0 ;$$

- 5- Aucun sous-ensemble flou ne peut être l'ensemble original lui même, donc le nombre d'individus d'un sous-ensemble flou donné est strictement inférieur à n .

Il existe des sous-ensembles particuliers dont leurs éléments u_{ij} associés ne prennent leurs valeurs que dans la paire $\{0,1\}$. Ces sous-ensembles constituent l'espace solution des méthodes classiques (non floues) de classification.

Les principales étapes de l'algorithme des *C-moyennes floues* sont :

- La fixation arbitraire d'une matrice d'appartenance $[u_{ij}]$;
- Le calcul des centroïdes des classes ;
- Le réajustement de la matrice d'appartenance suivant la position des centroïdes ;
- Le calcul du critère d'évaluation de la qualité de la solution, la non convergence de ce critère impliquant le retour à l'étape 2 (il est prouvé que l'algorithme converge toujours vers un minimum local [Bez81]).

La partition floue obtenue est présentée sous forme d'une matrice $n \times c$, où n est le nombre d'individus, c le nombre de classes obtenues, et l'élément u_{ij} de la matrice le degré d'appartenance de l'individu i à la classe j . Contrairement aux classifications dures, la valeur d'appartenance d'un individu à une classe ne prend pas seulement les valeurs 0 ou 1, mais toutes les valeurs possibles dans l'intervalle $[0,1]$.

La technique de base est de minimiser la variance intra-groupe. Dans l'algorithme des *C-moyennes floues*, Bezdek [Bez81] a proposé la fonction de discrimination suivante :

$$\min \sum_{j=1}^c \sum_{i=1}^n (u_{ij})^\lambda d^2(x_i, \beta_j) \quad (\text{IV.16})$$

avec :

- λ : Coefficient ≥ 1 (fixé généralement à 2 dans les applications) appelé exposant flou ;

Le poids de cet exposant s'interprète comme un paramètre de distorsion. Le « flou » de la partition augmente avec ce coefficient. Un tel coefficient accentue les faibles niveaux d'appartenance et contribue donc à mieux séparer les classes ;

- β_j : Centre de gravité de la classe j ;
- $d(x_i, \beta_j)$: Distance entre l'élément x_i et le centre de gravité β_j de la classe j .

Cette formule exprime donc la variance intra-groupe, où la distance entre chaque élément et le centre de gravité de la classe à laquelle il appartient est pondérée par le degré d'appartenance.

Quant à la validité des groupes obtenus, plusieurs mesures sont proposées. Pal et Bezdek [PB95], après avoir testé de différentes mesures, ont recommandé de prendre l'index de Xie-Beni (*compactness and separation index*) [XB91] comme critère pour choisir le nombre de groupes, l'optimum est obtenu en minimisant cet indice.

$$v_{XB} = \frac{\sum_{j=1}^c \sum_{i=1}^n (u_{ij})^2 d^2(x_i, \beta_j)}{n \left[\min_{k \neq j} (d^2(\beta_k, \beta_j)) \right]} \quad (\text{IV.17})$$

Le résultat direct fourni par l'algorithme est une matrice des degrés d'appartenance de chaque individu à chaque classe. Cette matrice donne déjà une image graduée de l'appartenance des individus aux classes ainsi qu'une image du chevauchement des classes. On peut très bien arrêter ici l'algorithme en affectant

l'individu à la classe la plus plausible, mais le problème de l'opposition entre la taille des groupes et leur degré d'homogénéité reste entier.

Il s'agit de révéler différentes configurations possibles au sein même d'une classe, et ceci grâce à un indice appelé seuil d'affectation (ou α -coupe). Le principe est d'affecter à une classe tous les individus ayant un degré d'appartenance à cette classe supérieur ou égal à un seuil donné, chaque individu ayant désormais la possibilité d'appartenir simultanément à plusieurs classes. En faisant varier ce seuil, on obtiendra plusieurs configurations pour une même classe.

La variation de ce seuil modifiera non seulement la taille des classes, mais aussi leur contenu ou bien encore leur degré d'homogénéité.

4.5 Mémoires hétéro-associatives : (*cartes auto-organisatrices de Kohonen*)

Avec les *mémoires hétéro-associatives*, on fournit une information au réseau et celui-ci rend une information différente. Par exemple, si la clef d'entrée est une image de visage, le système répond par le nom de la personne correspondante.

Un exemple de *mémoire hétéro-associative* est la *carte auto-organisatrice de Kohonen* abrégée en *SOM (Self Organizing Map)*.

Les réseaux de neurones à auto-organisation sont des réseaux non supervisés avec un apprentissage compétitif où l'on apprend non seulement à modéliser l'espace des entrées avec des prototypes, mais également à construire une carte à une ou deux dimensions permettant de structurer cet espace.

La carte de *Kohonen* est en général à deux dimensions (Figure IV.16). Chaque neurone de la couche d'entrées est relié à chaque neurone de la carte de *Kohonen*. Chaque neurone de la carte de *Kohonen* est relié à tous les neurones de la carte. Les neurones de ce réseau sont constitués d'un vecteur de poids dans l'espace des entrées. La carte des neurones définit quant à elle des relations de voisinage entre les neurones. Par exemple, la figure IV.17 montre un neurone gagnant (en bleu) et son voisinage (carrée en vert).

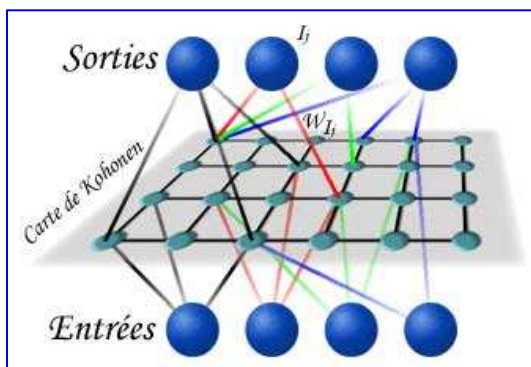


Figure IV.16 – Architecture d'un modèle de Kohonen.

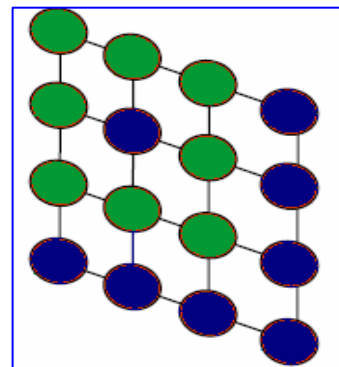


Figure IV.17 – Topologie de voisinage pour une carte à 2 dimensions.

Les réseaux de Kohonen ont des applications dans la classification, le traitement de l'image, l'aide à la décision et l'optimisation.

A la présentation d'une entrée, un neurone sur la carte est sélectionné. Il correspond le plus possible à cette entrée (minimisation d'une distance). Seul ce neurone gagnant, selon une certaine fonction, ainsi que un certain voisinage de ce neurone, verra son poids modifié.

Algorithme

1. Initialisation aléatoire des poids W et donner une valeur aléatoire positive au taux d'apprentissage η
2. A chaque itération t
 - Présentation d'un exemple d'apprentissage $x(t)$, choisi au hasard, à l'entrée de la carte.
 - Comparaison de l'exemple à tous les vecteurs poids, le neurone gagnant $g(x)$ est celui dont le vecteur poids $w_j(t)$ est le plus proche de l'entrée $x(t)$
(Phase de compétition)

$$g(x) = \min \|x(t) - w_j(t)\|, \quad j=1,2,\dots,m$$

où m le nombre de neurones du réseau.

- Mise à jour des poids pour tous les neurones de la carte
(Phase de coopération)

$$\Delta w_j(t) = \begin{cases} \eta(t)[x(t) - w_j(t)] & j \in A_g(x) \\ 0 & \text{sinon} \end{cases}$$

où $\eta(t)$ correspond au taux d'apprentissage et $A_g(t)$ à un voisinage autour du neurone gagnant g ; $\eta(t)$ et $A_g(t)$ sont toutes deux des fonctions décroissantes dans le temps.

Algorithme IV.2 – Algorithme d'apprentissage d'un réseau de Kohonen.

5 Conclusion

Nous avons commencé ce chapitre par l'introduction de quelques notions de base indispensables et fréquemment utilisées dans le domaine des systèmes de filtrage d'information collaboratif. Ensuite, nous avons passé en revue quelques méthodes les plus répandues pour former des communautés d'utilisateurs dans ces systèmes. Ces communautés sont construites selon un critère donné afin de former un espace de communautés.

La multiplicité de critères sur lesquels on peut former des communautés d'utilisateurs, conduit à l'existence de multiples espaces de communautés à la fois, et donc un utilisateur peut recevoir une diversité de recommandations de chacune de ses communautés auxquelles il appartient.

La procédure de formation des communautés nécessite des méthodes pour classier et affecter les utilisateurs aux communautés les plus appropriées, nous avons présenté ci-dessus quelques méthodes parmi les plus répandues, comme l'approche des *k-voisins les plus proches*, la classification par l'algorithme des *k-moyennes*, la classification par *arbres de décision*, ...

Chapitre V

Solution proposée

Afin d'améliorer l'efficacité et la qualité d'un système de filtrage d'information collaboratif, il est nécessaire d'améliorer la procédure de formation de communautés selon une multiplicité de critères disponibles dans la base des profils des utilisateurs. Pour cela, nous procédons de proposer en détail dans ce chapitre, la combinaison de deux méthodes de classification de la famille non supervisée, à savoir l'algorithme des *k-moyennes* et la méthode de la *classification ascendante hiérarchique*.

1 Introduction

La procédure permettant de regrouper et de réunir les utilisateurs en communautés, est une étape indispensable pour un fonctionnement efficace et de qualité d'un système de filtrage d'information collaboratif. Cette procédure, nécessite des méthodes pour classifier et affecter les utilisateurs aux communautés les plus appropriées.

Dans le domaine du filtrage collaboratif, le problème de la classification des utilisateurs, est un problème inévitable, et qu'il doit être pris en considération ; parce que l'efficacité des systèmes de filtrage collaboratif est conditionnée étroitement par la bonne classification des utilisateurs de ces systèmes en communautés. En général, l'opération de la classification est la construction d'une procédure permettant d'associer une communauté à un utilisateur. Classiquement, cette opération se décline en deux variantes : l'approche *supervisée* et l'approche *non supervisée* (voir chapitre précédent).

Nous nous intéressons dans notre proposition à la deuxième technique et nous proposons de combiner deux parmi ses méthodes, pour regrouper les utilisateurs en communautés afin de former autant d'espaces de communautés relatifs aux critères disponibles dans la base des profils des utilisateurs.

Après la classification des utilisateurs en classes (communautés) homogènes, l'intervention d'un processus de filtrage collaboratif est nécessaire afin de produire et faire changer des recommandations entre utilisateurs dans une même communauté : c'est la finalité mentionnée dans ce présent chapitre.

Ce chapitre présente dans un premier temps, notre proposition de solution afin de contourner la problématique, et dans un deuxième temps avant de terminer ce chapitre par une conclusion, nous présentons une expérimentation réalisée sur un jeu de données réel d'un système de recommandation de films « MovieLens ».

2 Limites des systèmes de recommandation actuels

Les systèmes de filtrage collaboratif souffrent du problème de démarrage à froid. Lorsqu'un nouvel utilisateur arrive, le système est incapable de lui fournir des recommandations vu qu'il ne dispose d'aucune information sur les centres d'intérêt de ce dernier. En effet, son profil évaluation est vide. La mono-criticité est donc une autre limite pour ce type de systèmes de filtrage.

Le filtrage actif constitue un moyen de résoudre les problèmes de la situation actuelle ; mais cela ne fonctionne que dans un contexte très étroit ; c'est-à-dire dans les petites communautés formées de collègues ou amis,... donc pour les systèmes où chaque utilisateur connaît parfaitement les centres d'intérêt des autres.

Un autre moyen pour contourner le problème du démarrage à froid consiste à forcer l'utilisateur à donner des évaluations dès les premiers jours d'utilisation du système. Par exemple, le système de recommandations de films « MovieLens » exige d'un nouvel utilisateur qu'il évalue au moins 15 films avant de pouvoir recevoir des recommandations.

Ces deux techniques permettent de mettre en place le profil de l'utilisateur qui s'enrichira au cours du temps grâce aux évaluations produites par l'utilisateur.

So far you have rated 3 movies.
MovieLens needs at least 15 ratings from you to generate predictions for you.
Please rate as many movies as you can from the list below.

Your Rating	Movie Information
??? Not seen	Best in Show (2000) Comedy
??? Not seen	Bullets Over Broadway (1994) Comedy
??? Not seen	Don Juan DeMarco (1995) Comedy, Drama, Romance
??? Not seen	Gandhi (1982) Drama
??? Not seen	In the Name of the Father (1993) Drama
??? Not seen	Junior (1994) Comedy, Sci-Fi
??? Not seen	Long Kiss Goodnight, The (1996) Action, Drama, Thriller
??? Not seen	Miracle on 34th Street (1994) Drama
??? Not seen	People vs. Larry Flynt, The (1996) Comedy, Drama
??? Not seen	Under Siege 2: Dark Territory (1995) Action

L'utilisateur doit évaluer au moins 15 films afin d'obtenir des recommandations.

Figure V.1 – Première utilisation de MovieLens par l'utilisateur safab_2002@yahoo.fr.

3 Méthode proposée

Pour un critère donné, notre proposition compare les utilisateurs entre eux à partir de leurs bases de profil, afin de les regrouper en communautés. Un profil d'un utilisateur est l'ensemble des informations traduisant les intérêts de celui-ci. Ces intérêts doivent prendre en compte non seulement les préférences de l'utilisateur mais aussi et surtout leurs évolutions dans le temps.

Aucune méthode de classification n'est parfaite ou typique, car toute méthode a des avantages et des inconvénients, par exemple, si le temps d'apprentissage est inexistant dans la classification par l'approche des *k-plus proches voisins*, puisque les données sont stockées telles quelles, la classification d'un nouveau cas est par contre coûteuse puisqu'il faut comparer ce cas à tous les exemples déjà classés. Et si en utilisant la classification par les réseaux de neurones, on peut arriver à une bonne capacité de généralisation, mais le calibrage du réseau pas forcément évident (nombre de couches, type de réseau, nombre de neurones par couche,...). Pour cela, nous proposons une approche hybride, combine deux méthodes, afin de compenser les inconvénients de l'une par les avantages de l'autre, donc, nous procédons de combiner deux méthodes appartenant à la famille non supervisée, à savoir l'algorithme des *k-moyennes* et la méthode de la *classification ascendante hiérarchique*.

Nous avons proposé d'utiliser l'algorithme des *k-moyennes*, suivi par la méthode de la *classification ascendante hiérarchique* (dendrogramme) en vue d'obtenir un nombre flexible de communautés [Ngu06].

3.1 Algorithme des *k-moyennes* (*k-means*)

Bien qu'elle ne fasse appel qu'à un formalisme limité et que son efficacité soit dans une large mesure attestée par les seuls résultats expérimentaux, la méthode des *k-means* est probablement la technique de partitionnement la mieux adaptée actuellement aux vastes recueils de données ainsi que la plus utilisée pour ce type d'applications.

Description de la méthode (*k-means*)

- On considère l'espace de n points de dimension p suivant :

$$X : \begin{bmatrix} x_1^1 & \dots & x_1^j & \dots & x_1^p \\ \dots & \dots & \dots & \dots & \dots \\ x_i^1 & \dots & x_i^j & \dots & x_i^p \\ \dots & \dots & \dots & \dots & \dots \\ x_n^1 & \dots & x_n^j & \dots & x_n^p \end{bmatrix} ;$$

- On suppose que les n points peuvent être groupés en c clusters $c < n$;
- Les clusters sont décrits par leurs centres de gravité :

$$V_k = [v_k^1, v_k^2, \dots, v_k^j, \dots, v_k^p], 1 \leq k \leq c ;$$

- On note $d(i,k)$ la distance entre le point x_i et le centre de gravité V_k , la mesure de distance peut être de Pearson, Euclidienne, ... ;
- Le point x_i est affecté au cluster dont le centre est le plus proche (au sens de d) ;
- On note m_k la moyenne des vecteurs dans le cluster k ;

Algorithme :

Initialiser la position des centres :

$$V_k = [v_k^1, v_k^2, \dots, v_k^j, \dots, v_k^p], 1 \leq k \leq c$$

Calculer les m_k ;

Jusqu'à ce qu'il n'y ait plus de changement sur les m_k **Faire :**

Chaque point X_i est affecté au cluster le plus proche ;

Calculer les nouveaux m_k ;

Fin Jusqu'à.

Algorithme V.1 – Algorithme des *k-moyennes* (*k-means*).

La figure suivante illustre le déroulement de l'algorithme pour :

La dimension : $p = 2$;

Le nombre de clusters : $c = 4$;

$V_k = [0,0], 1 \leq k \leq 4$;

Les cercles rouges représentent les positions successives des centres de gravité des 4 clusters.

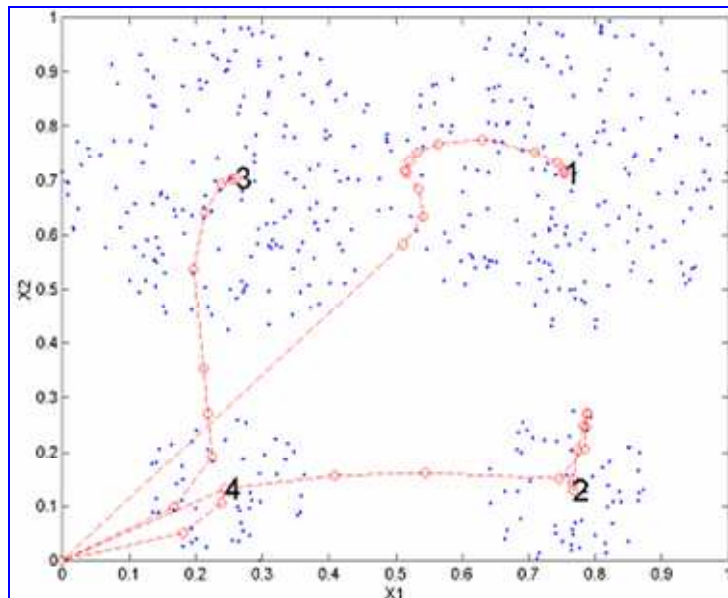


Figure V.2 – Exemple d'application de l'algorithme des *k-means*.

Les centres de gravité sont recalculés après chaque itération. Puisque les représentants (les centres de gravité) des communautés ne sont calculés qu'une fois par itération et que tous les individus sont pris en compte à chaque itération, les partitions (communautés) obtenues sont indépendantes de l'ordre dans lequel les individus sont considérés.

Avantages

L'algorithme des *k-moyennes* est une des méthodes de classification les plus populaires en raison de son efficacité dans la plupart des cas [JMF99]. Cette méthode est efficace même si les données sont volumineuses.

Relativement efficace : $O(tkn)$, où n est le nombre d'objets, k est le nombre de clusters, et t est le nombre d'itérations. Normalement, $k, t \ll n$, c'est-à-dire très rapide.

Inconvénients

Parmi les principaux inconvénients de cette méthode réside dans la criticité du choix des clusters initiaux, pouvant influencer sur la qualité de la classification ; c'est-à-dire on doit spécifier le nombre de clusters k (nombre de communautés) au départ.

3.2 Méthode de classification ascendante hiérarchique

La *classification ascendante hiérarchique* ou « *par agrégation* » procède par fusions successives de groupes « *clusters* » déjà existants. A chaque étape, les deux clusters qui vont fusionner sont ceux dont la « *distance* » est la plus faible (Figure V.3). Donc, il est indispensable de trouver une bonne définition de la « *distance* » entre deux groupes.

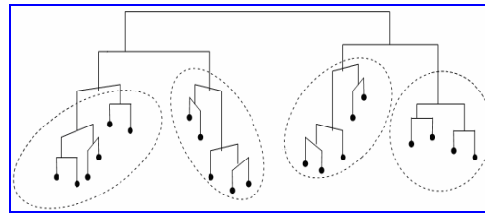


Figure V.3 – Illustration de la méthode CAH.

$T_{m \times m}$	u_1	...	u_i	...	u_m
u_1	0		$va_{1,i}$		$va_{1,m}$
...		0			
u_i			0		$va_{i,m}$
...				0	
u_m					0

où $va_{i,j}$: valeur d'approximation entre les deux utilisateurs u_i et u_j

Tableau V.1 – $T_{m \times m}$: Tableau initial de valeurs d'approximation entre les utilisateurs.

Le tableau V.1 est un tableau symétrique par rapport à sa diagonale, rempli initialement par les valeurs d'approximation entre les m utilisateurs deux à deux. Si l'une des deux parties séparées par la diagonale est complètement remplie, on obtient donc, $m(m-1)$ valeurs d'approximation dans le tableau.

Algorithme :	
$(m-1)$ étapes	Initialement : Calculer les dis(similarités) entre les utilisateurs deux à deux ;
	Entrées : $m(m-1)/2$ valeurs de dis(similarités) ;
	Jusqu'au regroupement de tous les utilisateurs en un seul groupe Faire : Regrouper les deux utilisateurs/groupes les plus proches ; Calculer la nouvelle matrice de dis(similarités) entre les utilisateurs (individuels ou groupes) restants et le nouveau groupe ;
	Fin Jusqu'à.

Algorithme V.2 – Algorithme de la classification ascendante hiérarchique.

Stratégie d'agrégation

1^{ère} étape :

Si $va_{i,j}$ est une dis(similarité) entre les deux utilisateurs u_i et u_j , tel que $va_{i,j}$ min(max)imum, alors, ces deux utilisateurs doivent regrouper dans un nouveau groupe : $G_1 = \{u_i, u_j\}$;

2^{ème} étape :

Un nouveau tableau de dis(similarités) de dimension $(m-1) \times (m-1)$ sera créé, en remplaçant u_i et u_j par son groupe G_1 ; donc, il est nécessaire de définir une **méthode**

d'agrégation entre un utilisateur et un groupe d'utilisateurs ou entre deux groupes d'utilisateurs.

Méthodes d'agrégation

Pour estimer la distance entre deux classes A et B (une classe peut être constituée de singleton), les approches les plus populaires sont : [JMF99].

Distance minimum : $d(A, B) = \min\{d(a, b), a \in A, b \in B\}$

Distance maximum : $d(A, B) = \max\{d(a, b), a \in A, b \in B\}$

Distance moyenne : $d(A, B) = \frac{1}{|A||B|} \sum_{a \in A, b \in B} d(a, b)$

Distance des centres de gravité : $d(A, B) = d(g_a, g_b)$

où g_a et g_b sont respectivement les centres de gravité de classes A et B .

Après avoir testé ces alternatives de distance par A. T. Nguyen, nous choisissons pour notre expérimentation la distance minimale en raison de sa meilleure performance de classification [Ngu06].

La *classification ascendante hiérarchique* est très populaire étant donné que son algorithme est efficace et le résultat est assez satisfaisant. Une fois appliquée à une base d'individus bien représentés, elle produit des partitions homogènes à un certain niveau d'agglomération choisi (nombre de partitions).

3.3 Distance de corrélation

Il existe plusieurs distances dans le domaine de la classification telles que la distance euclidienne, la corrélation de Pearson, la distance du *Chi-2* ou bien la similarité de vecteurs (Cosinus) et bien d'autres créées de jour en jour selon des besoins particuliers [GL86].

Pour notre système et afin de mesurer la distance entre deux utilisateurs, nous avons choisi la distance basée sur le coefficient de corrélation de Pearson, ce coefficient est calculé par la formule suivante : (chapitre III)

$$w(a, i) = \frac{\sum_j (v_{a,j} - \bar{v}_a)(v_{i,j} - \bar{v}_i)}{\sqrt{\sum_j (v_{a,j} - \bar{v}_a)^2 \sum_j (v_{i,j} - \bar{v}_i)^2}} \quad (\text{V.1})$$

avec : $w(a, i)$: La distance entre l'utilisateur actif a et l'utilisateur i ;

$v_{a,j}$: L'évaluation du document j par l'utilisateur actif a ;

$v_{i,j}$: L'évaluation du document j par l'utilisateur i ;

\bar{v}_i : L'évaluation moyenne de l'utilisateur i .

Les sommes sur les j concernent les documents pour lesquels à la fois les utilisateurs a et i ont donné des évaluations ($j \in I_a \cap I_i$; avec I_a, I_i : ensembles des documents évalués par les utilisateurs a et i respectivement).

Normalement, si l'on commence par 943 classes de singleton, la classification hiérarchique est inefficace pour ne pas dire irréaliste [Ngu06]. Ainsi, nous appliquons d'abord l'algorithme des *k-moyennes* avec $k=100$, pour créer 100 communautés initiales, et nous effectuons la *classification ascendante hiérarchique* sur ces classes pour former les communautés finales.

3.4 Architecture globale du système

L'architecture globale de notre système est composée principalement de deux modules suivants (Figure V.4) :

- Un module de formation des espaces de communautés d'utilisateurs : ce module a pour but de construire des espaces de communautés selon les différents critères disponibles dans la base de profils des utilisateurs. Il se charge pour chaque critère, de classer les utilisateurs du système en communautés selon une mesure de similarité définie et d'affecter un nouvel utilisateur (nouvel inscrit) à sa communauté adéquate.
- Un module de filtrage d'information collaboratif : consiste à filtrer les documents pertinents pour un utilisateur en se basant sur la table de communautés remplie à partir des résultats du premier module et sur le corpus de documents.

Après avoir sélectionné les documents considérés pertinents par le système, ce dernier procède de les recommander à l'utilisateur concerné.

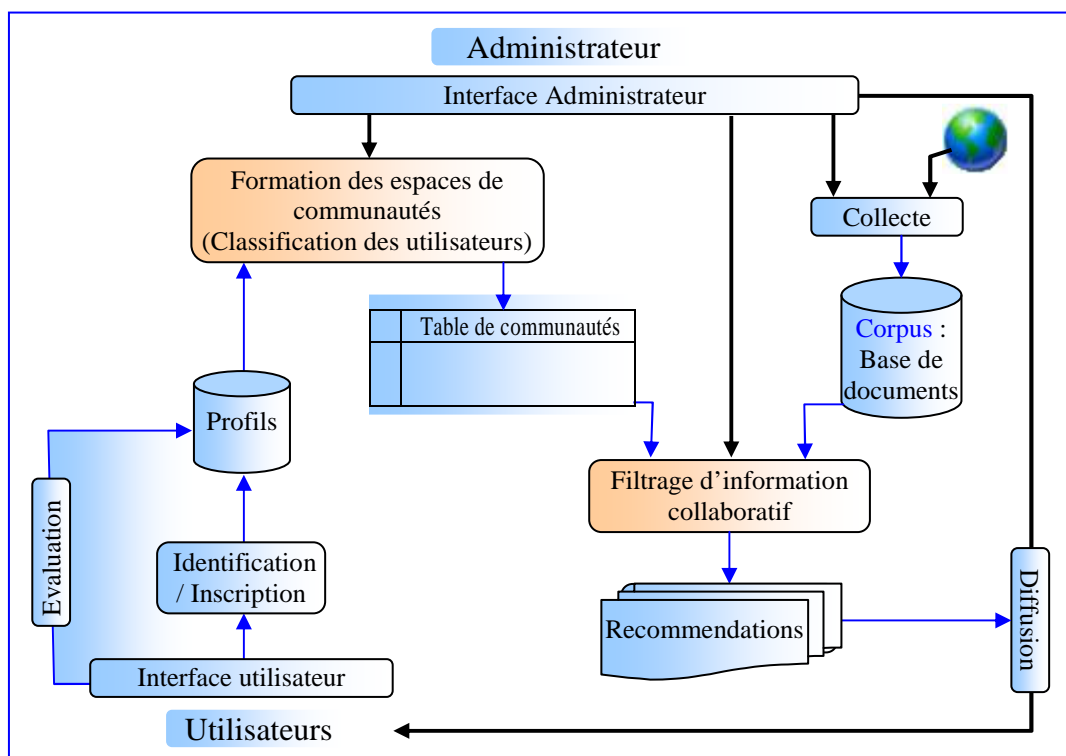


Figure V.4 – Architecture globale du système.

Comme le montre la figure V.4, pour chaque critère disponible dans la base de profils des utilisateurs, le système fait appel au module de formation des espaces de communautés afin de regrouper les utilisateurs en communautés. Les communautés

engendrées selon un critère donné, doivent être réunies dans un espace de communautés particulier relatif à ce critère.

La tâche de formation des communautés d'utilisateurs, et leur regroupement en espaces est très importante, puisque le module de filtrage d'information collaboratif se base principalement sur la table de communautés, remplie par ces espaces formés, afin de produire des recommandations aux utilisateurs.

1. Module de formation des espaces de communautés

Un système de classification est constitué des principales composantes suivantes (Figure V.5) :

1. Définir une **représentation** des utilisateurs (profils des utilisateurs).
2. Définir une **mesure de similarité**, le choix de la mesure de distance entre individus et le traitement approprié selon chaque attribut est très important.
3. Appliquer un **algorithme de classification** reposant sur cette mesure afin de classer les utilisateurs en communautés selon un critère de classification spécifique et disponible dans la base de profils des utilisateurs.

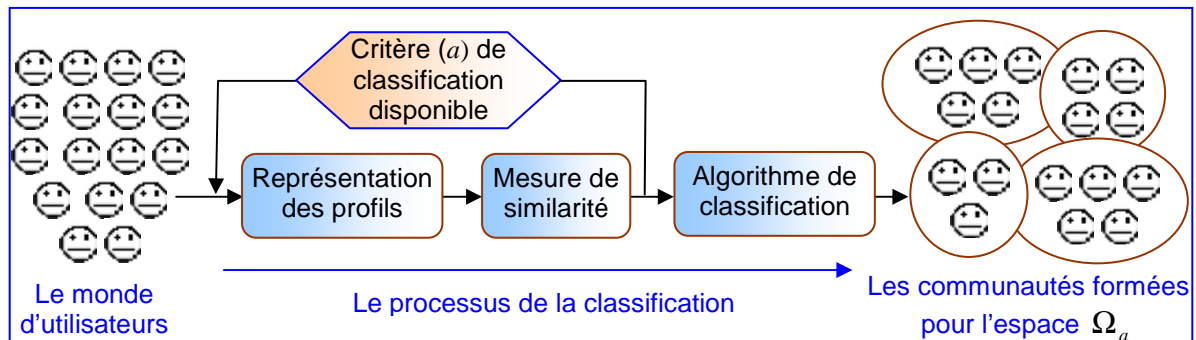


Figure V.5 – Fonctionnement d'un système de classification des utilisateurs.

Afin de représenter les profils des utilisateurs, plusieurs modèles peuvent être utilisés. En effet, on peut utiliser un modèle relationnel simple où l'utilisateur est décrit par un ensemble d'attributs, on peut aussi représenter les profils en utilisant un modèle sémantique, où l'utilisateur est décrit par un ensemble d'attributs dont les valeurs appartiennent à l'ensemble des termes d'une ontologie. Une autre représentation possible par une modélisation vectorielle où les coordonnées d'un vecteur représentent le poids lié à un terme [ZLB05].

Avant de regrouper les utilisateurs en communautés, la définition de la mesure de distance entre individus est nécessaire afin de déterminer la valeur de similitude qui existe entre les utilisateurs selon un critère donné. Ensuite, la procédure de classification est continuée par l'application de l'algorithme de classification défini préalablement afin de former les communautés d'utilisateurs finales.

Les résultats de ce module sont des communautés (groupes d'utilisateurs) virtuellement regroupées en des espaces de communautés d'utilisateurs, chaque espace est relatif à un critère particulier. Nous donnons par exemple, l'espace relatif au critère *Âge*, noté $\Omega_{\text{Âge}}$ rassemble les différentes communautés ; chacune de ces communautés correspondante à l'une des cinq tranches d'âge suivantes : moins de 16 ans, de 16 à 25 ans, de 26 à 45 ans, de 46 à 60 ans et plus de 60 ans.

Etant donné huit nouveaux utilisateurs par exemple, ils déclarent lors de ses inscriptions, leurs informations personnelles : *Âge*, *Profession*, *Ville*, *Genres préférés* et ils possèdent des « *profils Evaluation* » nuls pour ses évaluations passées comme présenté dans le tableau suivant :

Utilisateur	Informations d'inscription				
	Âge	Profession	Ville	Genre préféré	Evaluation
u_{13}	14 ans	Commerçant	Paris	Aventure	-
u_{14}	31 ans	Chercheur	Paris	Policier	-
u_{15}	25 ans	Commerçant	New York	Documentaire	-
u_{16}	28 ans	Chercheur	New York	Aventure	-
u_{17}	46 ans	Chercheur	Paris	Policier	-
u_{18}	15 ans	Commerçant	Londres	Policier	-
u_{19}	46 ans	Chercheur	Londres	Fiction	-
u_{20}	22 ans	Commerçant	Londres	Documentaire	-

Tableau V.2 – Exemple d'inscription des nouveaux utilisateurs.

Le système doit classier chacun de ces nouveaux utilisateurs dans les communautés les plus appropriées dans les différents espaces de communautés.

2. Module de filtrage d'information collaboratif

L'objectif primordial du module de filtrage d'information collaboratif est la production des recommandations pour les utilisateurs en fonction de leurs communautés multicritères figurantes dans la table de communautés. La prédiction de recommandation d'un document donné à un utilisateur particulier dans un système de filtrage collaboratif classique est calculée par la proximité des évaluations en utilisant la corrélation de Pearson. Les difficultés de cette tâche apparaissent dans le démarrage à froid où le nouvel utilisateur n'a pas encore d'évaluations, ou dans le cas d'un faible nombre d'évaluations en commun entre les utilisateurs [Ngu06].

Pour dépasser ces difficultés, nous travaillons sur une plateforme de multiple critère, et nous proposons dans ce cadre, d'introduire dans le module de filtrage collaboratif, la notion de priorité entre les critères. L'espace de communautés relatif au critère le plus prioritaire sera concerné pour la production de recommandations.

Priorité d'un critère

Une autre partie de notre contribution est l'introduction de la notion de priorité entre les divers critères disponibles dans la base de profils des utilisateurs. Ces critères prennent initialement la même priorité (au début, les données personnelles ont la même priorité par exemple). Cette dernière est recalculée pour chaque critère suivant le nombre d'évaluations données par les utilisateurs d'une même communauté. Pour un utilisateur donné, sa communauté la plus prioritaire est celle qui possède le plus grand nombre d'évaluations de ses membres. L'échelle de la priorité est fixée dans l'intervalle $[0,1]$; La valeur « 1 » indique la priorité la plus importante.

Etant donné un utilisateur, si le nombre d'évaluations données par les membres de sa communauté *Âge* est plus grand que celui relatif à sa communauté *Profession*,

donc le critère *Âge* est plus prioritaire que celui de *Profession* et donc, cet utilisateur recevra les recommandations via sa communauté *Âge*.

Notons que, deux utilisateurs appartenant à une même communauté, peuvent avoir des communautés plus prioritaires différentes. Donc, chacun peut recevoir des recommandations auprès de sa propre communauté considérée prioritaire.

Considérons un utilisateur u_0 appartenant à la communauté G dans l'espace de communautés *Evaluation*. Le système pourrait faire des recommandations à cet utilisateur en se basant sur le critère usuel *Evaluation* si l'utilisateur u_0 évalue au moins 20 ressources (documents), et donc la priorité la plus élevée sera attribuée à ce critère. La contrainte minimale de 20 évaluations est une limite suffisante pour que le système puisse faire des recommandations sur la base du critère *Evaluation*. Cette limite est inspirée de l'ensemble de données du système de recommandation de films « MovieLens » sur lequel notre expérimentation sera réalisée, ainsi que la plupart des membres de ce système n'évaluent que les ressources qu'ils aiment et le nombre minimal d'évaluations faites par chaque utilisateur est de 20.

Après le classement des nouveaux utilisateurs dans les communautés les plus appropriées dans les différents espaces de communautés, le système doit calculer la priorité de chacun des critères afin de déterminer les communautés les plus prioritaires qui seront concernées par la production de recommandations pour ces nouveaux utilisateurs.

Avant de calculer les priorités entre les divers critères, le système doit calculer le nombre des évaluations fournies par les membres des communautés dont lesquelles les nouveaux utilisateurs appartiennent (Tableau V.3).

<i>User</i>	Le nombre des évaluations dans la communauté :				
	<i>Âge</i>	<i>Profession</i>	<i>Ville</i>	<i>Genre préféré</i>	<i>Evaluation</i>
u_{13}	1 259	7 554	953	2 321	0
u_{14}	51 948	38 730	953	28 254	0
u_{15}	29 305	7 554	5 256	31 891	0
u_{16}	51 948	38 730	5 256	2 321	0
u_{17}	16 277	38 730	953	28 254	0
u_{18}	1 259	7 554	8 151	28 254	0
u_{19}	16 277	38 730	8 151	787	0
u_{20}	29 305	7 554	8 151	31 891	0

Tableau V.3 – Nombre des évaluations données par les membres des communautés.

Ensuite, les priorités des critères seront calculées sur la base du nombre des évaluations, le plus grand. En divisant les nombres des évaluations obtenus dans la phase précédente par celui le plus grand (la valeur « 1 » indique le critère le plus prioritaire).

User	La priorité des critères :					La communauté la plus prioritaire
	Âge	Profession	Ville	Genre préféré	Evaluation	
u_{13}	0,17	1,00	0,13	0,31	0,00	Profession
u_{14}	1,00	0,75	0,02	0,54	0,00	Âge
u_{15}	0,92	0,24	0,16	1,00	0,00	Genre préféré
u_{16}	1,00	0,75	0,10	0,04	0,00	Âge
u_{17}	0,42	1,00	0,02	0,73	0,00	Profession
u_{18}	0,04	0,27	0,29	1,00	0,00	Genre préféré
u_{19}	0,42	1,00	0,21	0,02	0,00	Profession
u_{20}	0,92	0,24	0,26	1,00	0,00	Genre préféré

Tableau V.4 – Exemple de priorités entre critères (espaces).

Etant donné, deux utilisateurs appartenant à une même communauté, peuvent avoir des communautés prioritaires différentes. Les utilisateurs u_{13} et u_{16} appartiennent à la même communauté *Genre préféré* « Aventure » (Tableau V.2), mais, ils reçoivent leurs recommandations auprès des communautés entièrement différentes situées dans des espaces aussi différents (Tableau V.4).

Le système fait donc des recommandations aux nouveaux inscrits sur la base de la communauté la plus prioritaire pour chacun. Pour recommander un document aux utilisateurs u_{13} , u_{17} et u_{19} , le système doit se baser sur les communautés situées dans l'espace profession ($\Omega_{\text{Profession}}$) de ces trois utilisateurs qui sont « *Commerçant* », « *Chercheur* » et « *Chercheur* » respectivement lors du calcul de la prédiction, par contre, les utilisateurs u_{14} et u_{16} recevront les recommandations auprès de leurs communautés *Âge* (les deux utilisateurs appartiennent à la même communauté noté $\hat{\text{Age}}_{26-45}$), etc.

Après la fourniture des évaluations par les nouveaux utilisateurs sur les recommandations diffusées par le système, ce dernier effectuera l'opération de la mise à jour des priorités entre les critères de formation de communautés afin de faire de nouvelles recommandations.

Si un utilisateur fournit 20 évaluations au minimum (voir §.3.4.2), le système doit trouver la communauté d'évaluation la plus adéquate pour cet utilisateur qui sera la plus prioritaire et par la suite il recevra les recommandations auprès de sa communauté *Evaluation*.

Production de recommandations

Après la détermination le critère le plus important, et afin de produire des recommandations pour un utilisateur donné et dans le contexte de multiplicité de critères, nous utilisons l'approche de « recommandation par niveau d'accord ». Cette approche est une méthode ad hoc pour produire des recommandations en fonction du « niveau d'accord » au sein des communautés concernées. Le principe de cette méthode de production de recommandations repose sur une forme de « quasi-unanimité » de la communauté sur la qualité d'un document. En général, ce vote des utilisateurs dans la communauté peut être symbolisé par deux seuils : S_{accord} pour le nombre d'évaluations et S_{score} pour le score moyen de la communauté sur un document particulier [Ngu06].

Considérons un utilisateur u_0 appartenant à la communauté G dans un espace de communautés spécifique. Pour chaque critère, c'est-à-dire pour chaque espace de communautés, la méthode doit sélectionner les documents à recommander parmi ceux qui ont été évalués par la communauté G à laquelle l'utilisateur u_0 appartient. Alors, le document d est suggéré à l'utilisateur u_0 s'il vérifie les deux conditions suivantes :

$$1. \quad \frac{1}{|\{v_{G,d}\}|} \sum_{u \in G} v_{u,d} \geq S_{score} \quad (V.2)$$

$$2. \quad |\{v_{u,d} / (u \in G) \text{ et } (v_{u,d} \neq null)\}| \geq S_{accord} \quad (V.3)$$

où $v_{u,d}$: Score donné par l'utilisateur u sur le document d ;
 $|\{v_{G,d}\}|$: Le nombre des scores non nuls donnés sur le document d par les membres de la communauté G .

Suite à la première condition, le système procède à un premier filtre en ne considérant que les documents évalués par la communauté G avec un score moyen supérieur ou égal au seuil S_{score} . Enfin, le système réalise un deuxième filtre, selon la deuxième condition, où on ne conserve que les documents qui ont été ainsi évalués par un nombre suffisant des membres de la communauté G , dont la limite est fixée par un autre seuil S_{accord} .

Ce type de filtrage collaboratif par niveau d'accord permet d'une part de dépasser le problème du démarrage à froid pour les nouveaux utilisateurs qui commencent par une base de profil *Evaluation* vide et d'autre part d'améliorer cette base de profil par la diversification de recommandations de la part de multiples communautés.

Après la détermination des communautés intéressées pour la production de recommandations, le système procède au calcul des prédictions afin de diffuser les documents pertinents aux nouveaux utilisateurs en se basant sur le filtrage collaboratif par niveau d'accord.

Nous suivons notre exemple et nous considérons l'utilisateur u_{13} appartenant à la communauté « *Commerçant* » dans l'espace de communautés $\Omega_{\text{Profession}}$ (Tableau V.2). Pour le critère le plus prioritaire *Profession*, le système doit sélectionner les documents à recommander parmi ceux qui ont été évalués par la communauté « *Commerçant* » à laquelle l'utilisateur u_{13} appartient. Alors, le document d_3 (Tableau V.5) est suggéré à l'utilisateur u_{13} si les deux conditions suivantes sont vérifiées :

$$\text{✎} \quad \frac{1}{|\{v_{\text{Commerçant},d_3}\}|} \sum_{u \in \text{Commerçant}} v_{u,d_3} \geq S_{score} \quad (V.4)$$

$$\text{✎} \quad |\{v_{u,d_3} / (u \in \text{Commerçant}) \text{ et } (v_{u,d_3} \neq null)\}| \geq S_{accord} \quad (V.5)$$

où v_{u,d_3} : score donné par l'utilisateur u sur le document d_3 .
 $|\{v_{\text{Commerçant},d_3}\}|$: Le nombre des scores non nuls donnés sur le document d_3 par les membres de la communauté *Commerçant*.

Le tableau suivant illustre un exemple d'une matrice d'évaluations d'un échantillon de la communauté *Commerçant* pour cinq documents.

Utilisateur commerçant	d_1	d_2	d_3	d_4	d_5
u_1	5	2	4		1
u_9	4	3	5	4	
u_{10}	2	4		5	2
u_{11}	5		4	3	
u_{12}	3	4	4	4	3
u_{13}			?		?

Tableau V.5 – Matrice des scores de la communauté *Commerçant*.

La vérification des deux conditions citées ci-dessus pour recommander le document d_3 à l'utilisateur u_{13} est procédée comme suit :

1. Pour la première condition, l'algorithme calcule le score moyen des évaluations données sur le document d_3 par les membres de la communauté des commerçants :

$$ScoreMoyen(d_3) = \frac{1}{4}(4 + 5 + 4 + 4) = \frac{17}{4} = 4,25 > S_{score}; \quad (S_{score} = 3 \text{ par exemple})$$

Donc, la première condition est vérifiée, et l'algorithme passe à la vérification de la deuxième contrainte.

2. Pour la deuxième condition, l'algorithme énumère le nombre d'évaluations non nulles attribuées au document d_3 par les commerçants.

$$NbScores(d_3) = 4.$$

Cette valeur (4 parmi les 5 utilisateurs commerçants, qui évaluent le document d_3) représente une proportion de $80\% = (\frac{4}{5} \times 100) > S_{accord}$; ($S_{accord} = 25\%$ par exemple), ce qui traduit une part importante d'utilisateurs qui évaluent le document d_3 .

Donc, les deux conditions sont vérifiées et le système prend une décision positive pour recommander le document d_3 à l'utilisateur u_{13} .

Par contre, le document d_5 ne peut pas être recommandé à l'utilisateur u_{13} à cause de la non vérification de la première contrainte ($ScoreMoyen(d_5) = 2 < S_{score}$) et donc sans passer à la vérification de la deuxième condition.

Donc, suivant la première condition, le système réalise un premier filtre en ne considérant que les documents évalués avec un score moyen important ($ScoreMoyen \geq 3$ étoiles) par une communauté donnée. Enfin, selon la deuxième contrainte, le système procède à un deuxième filtre par la conservation que les documents qui ont été évalués par un nombre suffisant ($NbScores \geq 25\%$) des membres de la communauté en question. Par conséquent, les documents résultants de deux filtres méritent la recommandation.

3.5 Modélisation de données

Nous présentons maintenant une modélisation de système de filtrage collaboratif composée d'un modèle de données, d'un modèle statique de processus et un modèle dynamique pour ces processus. En utilisant pour cela le langage de modélisation *UML*¹.

1. Le modèle de données

Le diagramme de classes de la figure suivante présente le modèle de données composé de quatre entités manipulées par le système et reliées entre elles, telles que : utilisateur, communauté, profil et document.

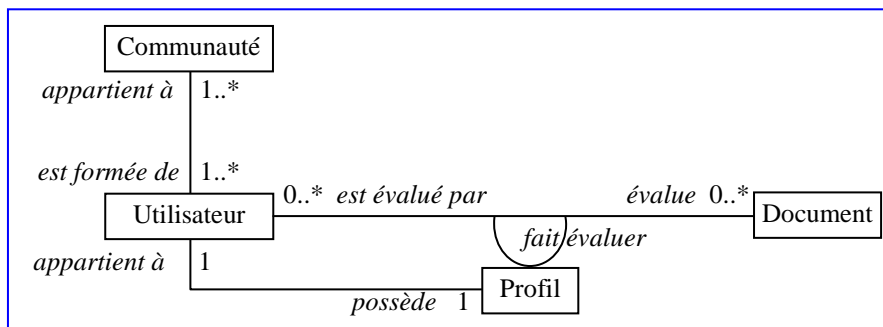


Figure V.6 – Diagramme de classes d'un modèle de données d'un *SFC*².

2. Le modèle statique des processus

Généralement, un système de filtrage collaboratif est composé de trois principaux processus et sont exécutés dans un ordre bien déterminé (Figure V.7).

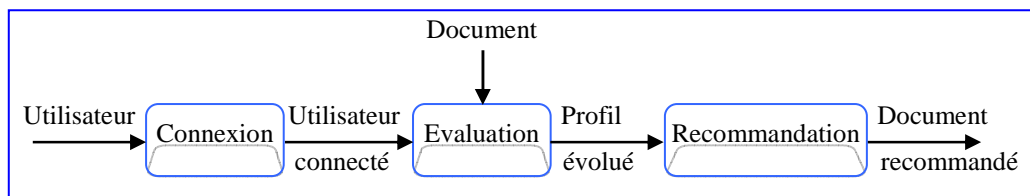


Figure V.7 – Modèle statique d'enchaînement des processus d'un *SFC*.

Le processus de recommandation inclut les activités nécessaires pour faire des propositions de documents censées être intéressantes pour un utilisateur particulier. Pour cela trois sous processus sont nécessaires comme présenté dans la figure suivante, à savoir : *formation de communautés*, *calcul de prédiction* et *sélection des recommandations*.

¹ *UML* : Unified Modeling Language

² *SFC* : Système de Filtrage Collaboratif

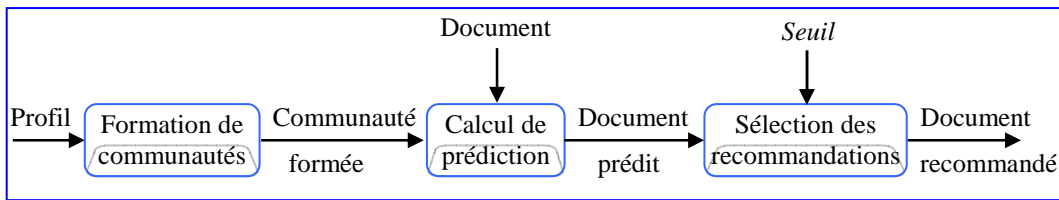


Figure V.8 – Composants du processus de recommandation.

3. Le modèle dynamique des processus

Le comportement dynamique d’un système de filtrage collaboratif est décrit par le schéma suivant :

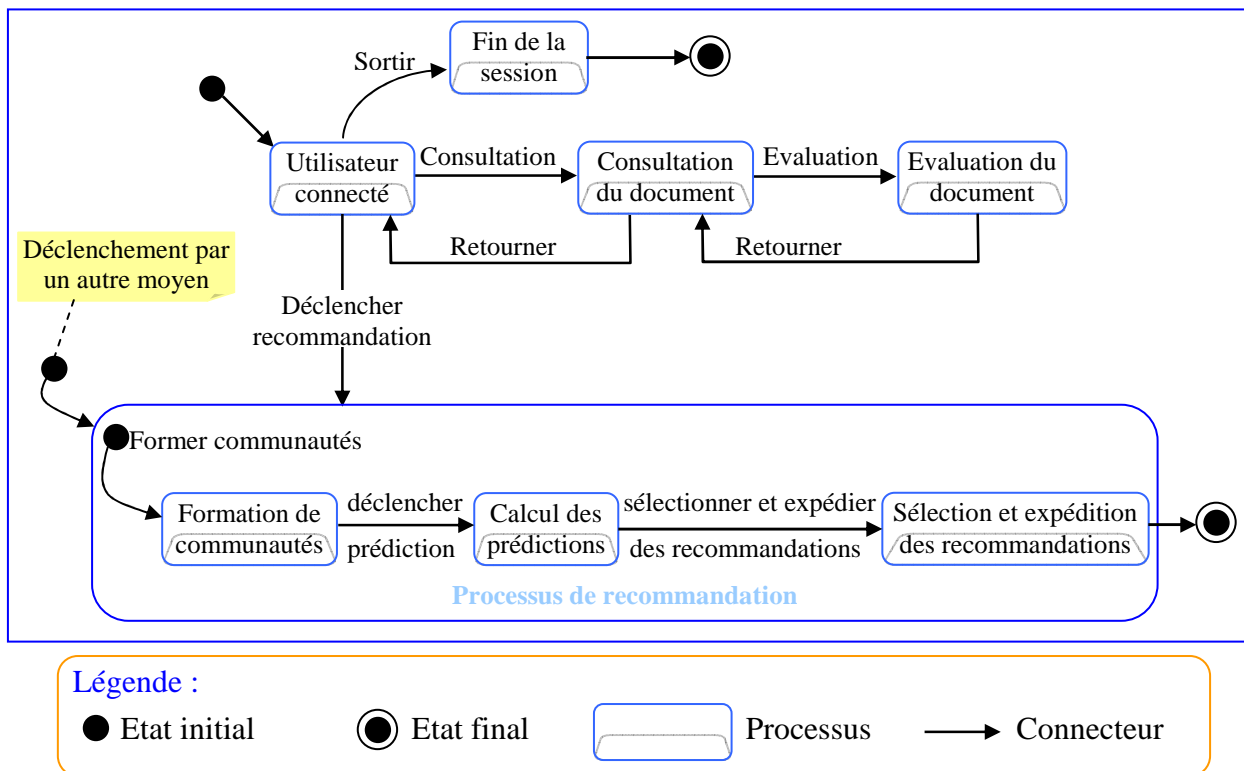


Figure V.9 – Diagramme d’états des processus d’un SFC.

Quand un utilisateur demande une connexion, il doit s’identifier, en fournissant un « pseudo » ou une « adresse e-mail ». L’utilisateur connecté peut procéder à la consultation des documents avec tous les niveaux de lecture possibles, comme il peut exécuter une autre action ou sortir du système (*fin de session*). Quand il évalue un document, il peut retourner vers l’état de consultation et changer son évaluation ou exécuter une autre action.

Le processus de recommandation peut être déclenché par plusieurs évènements : par un mécanisme temporel (à intervalles de temps réguliers, lors de la connexion d’un utilisateur, à la demande de l’utilisateur,...) ; ou bien par l’administrateur du système. Le premier état de ce processus est de réunir les utilisateurs afin de former les communautés selon un critère donné ; ensuite, le système passe à l’état de calcul de prédictions sur un ensemble de documents. Les documents atteignant le seuil de pertinence voulu sont envoyés à l’utilisateur.

4 Conclusion

Pour conclure, nous rappelons que notre objectif est d'améliorer la qualité globale du fonctionnement des systèmes de filtrage d'information collaboratif. Cette qualité est entièrement conditionnée par celle de la formation des communautés d'utilisateurs de ces systèmes. Pour cela, nous choisissons une hybridation de deux méthodes de la famille de la classification non supervisée, à savoir, l'algorithme des *k-moyennes* et la méthode de la *classification ascendante hiérarchique*, afin de regrouper les utilisateurs en communautés homogènes suivant chaque critère disponible dans la base des profils.

Dans le cadre de l'amélioration des performances de ce type de systèmes, nous introduisons toutes les caractéristiques de l'utilisateur dans le processus de formation des communautés afin de générer de bonnes recommandations aux utilisateurs, et nous intégrons la notion de priorité entre les caractéristiques. Nous choisissons aussi pour la production de recommandations, la technique de filtrage collaboratif basée sur les « recommandations par niveau d'accord ».

Chapitre VI

Evaluation

Nous procédons dans ce chapitre à l'évaluation de l'approche proposée. D'abord, nous commençons par présenter le jeu de données choisi ainsi que les métriques d'évaluation utilisées. Ensuite, nous analysons les résultats obtenus pour les différentes expériences réalisées.

1 Introduction

Après avoir présenté dans le chapitre précédent l'approche proposée et son architecture, ce chapitre est consacré notamment à la validation de notre proposition en utilisant un jeu de données réel. En premier, nous présentons ce jeu de données avec une description chiffrée, ensuite nous procédons à la construction des espaces de communautés. Nous distinguons deux types de critères : simples et complexes.

Nous présentons deux expériences : nous évaluons en premier l'efficacité de la méthode proposée pour le processus de formation de communautés appliquée en particulier avec le critère usuel et complexe « *Evaluation* » ; ensuite nous mesurons les améliorations apportées par l'introduction de la notion de priorité entre les critères lors d'exécution du processus de recommandations, nous choisissons cette fois-ci deux critères simples « *Âge* » et « *Profession* ».

2 Jeu de données

Pour l'expérimentation proposée dans ce chapitre, nous utilisons un jeu de données réel du système de recommandation de films MovieLens¹. Ce jeu de données est très populaire et utilisé dans beaucoup d'études du domaine de filtrage collaboratif [Ngu06]. Le site Web MovieLens est développé par le groupe de recherche GroupLens² à l'université de Minnesota, Etats-Unis. On dispose sur ce site deux jeux de données d'évaluations de films de tailles différentes. Nous utilisons dans cette expérimentation le jeu qui contient 100000 évaluations³ de 1 à 5 étoiles, fournies par 943 utilisateurs sur 1682 films pendant la période du Septembre 1997 au Avril 1998.

Un utilisateur est décrit par son identificateur (numéro séquentiel), son âge, son sexe, sa profession (poste occupé) et le zip code (code postal) de sa ville d'habitation. Un film de données MovieLens est caractérisé par son code (numéro séquentiel), son titre, sa date de réalisation, son genre et son url *IMDB*⁴.

2.1 Analyse du jeu de données

Nous divisons l'ensemble des évaluations en deux, une partie « Apprentissage » qui contient les données d'apprentissage utilisées par l'algorithme et une partie « Test » contenant les données qui vont servir à évaluer l'algorithme.

La partie « Apprentissage » contient 70759 évaluations (soit un pourcentage de 70,76% de l'ensemble des évaluations du jeu MovieLens) données par 647 utilisateurs (représentent une proportion de 68,61% de l'ensemble des utilisateurs) inscrits dans les cinq premiers mois. Le nombre d'évaluations faites par chaque utilisateur dans cette partie varie de 20 à 737.

La partie « Test » contient 29241 évaluations (soit un pourcentage de 29,24% de l'ensemble des évaluations du jeu) données par 296 utilisateurs (représentent une proportion de 31,39% de l'ensemble des utilisateurs) inscrits dans les trois derniers mois. Le nombre d'évaluations faites par chaque utilisateur dans cette partie varie de 20 à 685.

¹ <http://movielens.umn.edu/>

² <http://www.grouplens.org/>

³ <http://www.grouplens.org/data/>

⁴ *IMDB* : Internet Movie Data Base

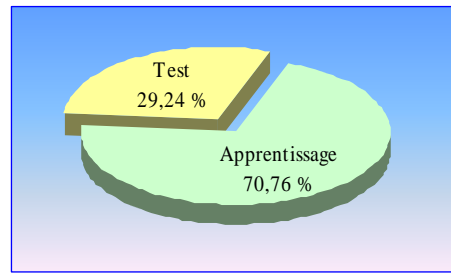


Figure VI.1 – Répartition des données d'expérimentation.

Nous donnons quelques statistiques simples pour un premier éclairage sur le jeu de données MovieLens :

- Le nombre d'évaluations faites par chaque personne varie de 20 à 737 ;
- Le nombre de films jugés en commun entre utilisateurs est généralement faible ;
- Pour la répartition des scores donnés par les utilisateurs, nous constatons une proportion faible pour les scores défavorables (6,11% et 11,37% pour 1 et 2 étoiles respectivement), ce qui montre la tendance des utilisateurs à n'évaluer que les films qu'ils préfèrent (Figure VI.2).

La plupart des utilisateurs évaluent par des scores de 4, 3 et 5, dans l'ordre décroissant.

- ✍ 34,17% des évaluations effectuées par des scores de 4 ;
- ✍ 27,15% par des scores de 3 ;
- ✍ 21,20% par des scores de 5.

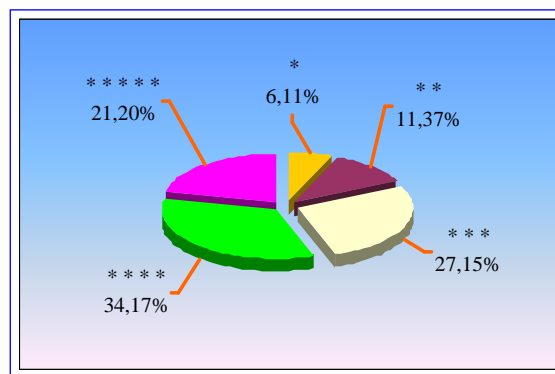


Figure VI.2 – Répartition des évaluations du jeu MovieLens.

2.2 Critères de formation des communautés

Les critères tirés du jeu MovieLens pour former les espaces de communautés sont répartis en trois catégories : critères d'informations personnelles, critère de centres d'intérêts et critères relatifs à l'historique des évaluations.

Concernant les critères d'informations personnelles, chaque utilisateur du système MovieLens doit déclarer à l'inscription ses informations personnelles : âge, profession et ville de résidence (zip code) aux Etats-Unis. Ces données nous permettent de définir respectivement les trois critères démographiques suivants : *Âge*, *Profession* et *Géographie (Ville)*.

Les âges des utilisateurs sont limités par l'intervalle de 7 à 73 ans, et sont répartis sur 5 tranches d'âge : moins de 16 ans, 16-25 ans, 26-45 ans, 46-60 ans et plus de 60 ans. La figure suivante représente le pourcentage des membres du système par tranche d'âge.

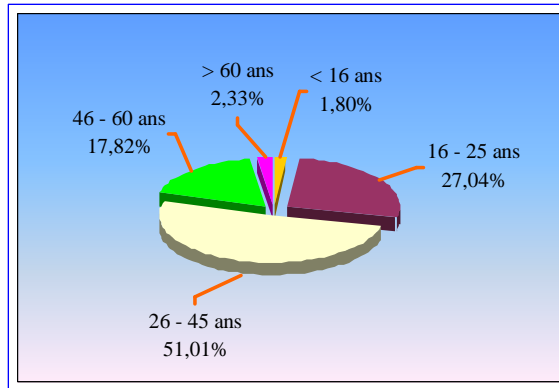


Figure VI.3 – Pourcentage d'utilisateurs par tranche d'âge.

Dans le jeu MovieLens, il y a 21 professions. Ce nombre a été réduit à 7 catégories de profession par A. T. Nguyen par le regroupement des professions assez proches. La nouvelle répartition est la suivante :

- Enseignant – Chercheur (student, educator, librarian, scientist) ;
- Commerçant (lawyer, salesman, marketing, executive) ;
- Ingénieur (engineer, technician, programmer, administrator) ;
- Artiste (artist, entertainment, writer) ;
- Santé publique (doctor, healthcare) ;
- Retraité (retired, homemaker) ;
- Autres professions (other, none).

Ainsi, chaque utilisateur aura une des 7 catégories ci-dessus selon sa profession déclarée à l'inscription. La figure suivante montre le pourcentage d'utilisateurs par profession.

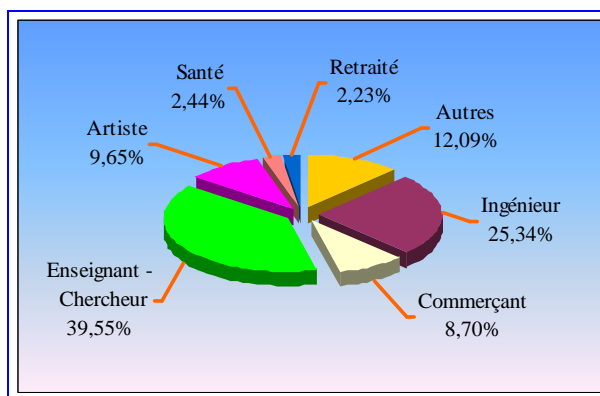


Figure VI.4 – Pourcentage d'utilisateurs par profession.

Pour le critère *Géographie*, les utilisateurs ont aussi donné leurs codes postaux (zip code) aux Etats-Unis. En réalité, 44 états sont présents dans les données.

Pour le critère de centres d'intérêts, l'utilisateur peut constituer son profil thématique traduit par les genres de film qu'il préfère, à partir de la caractéristique « genre » du film, cette dernière considérée comme l'information décrivant l'aspect contenu du film. La base de données du système MovieLens, contient au total 19 genres

de film, et chaque film peut être associé à plusieurs genres à la fois. Par exemple, certains films sont associés à 6 genres. Pour chaque utilisateur, son profil thématique, aussi appelée « *profil Contenu* » est construit comme un vecteur de 19 valeurs.

Enfin, le critère de l'historique de l'interaction entre l'utilisateur et le système, est traduit par l'attribut classique *Evaluation* s'appuyant sur les évaluations fournies par les utilisateurs, nous utilisons également le terme « *profil Evaluation* » pour ce critère.

En résumé, chaque utilisateur du système de recommandation de films MovieLens est caractérisé par les critères suivants : *Âge*, *Profession*, *Ville*, *Contenu* et *Evaluation*, afin de former autant d'espaces de communautés.

2.3 Construction de la table de communautés

Il faut d'abord construire la table des communautés des 943 vecteurs de positionnement avec des colonnes relatives aux critères suivants : *Âge*, *Profession*, *Géographie*, *Contenu* et *Evaluation*. Pour les trois premiers critères, la création des espaces Ω est simple, avec un nombre fixe de communautés dans chaque espace Ω :

- Critère *Âge* : 5 communautés correspondantes aux cinq tranches d'âge (7 à 73 ans) ;
- Critère *Profession* : 7 communautés de catégories de professions ;
- Critère *Géographie* : 44 communautés via les états des Etats-Unis.

Concernant l'attribut *Contenu*, on regroupe les utilisateurs partageant les mêmes centres d'intérêt. Ces intérêts sont représentés par la caractéristique du genre de film, alors que le critère *Evaluation*, on regroupe les utilisateurs selon leur proximité d'évaluations des films.

Afin de réaliser la construction des espaces de communautés, nous appliquons la méthode des *k-moyennes*, puis nous la remplaçons par une *classification ascendante hiérarchique* [JMF99] en vue d'obtenir un nombre de communautés flexible.

3 Métriques d'évaluation

Cette partie est consacrée à l'évaluation des performances de notre approche de filtrage. Nous présentons quelques mesures ou métriques de performance fréquemment utilisées dans le contexte du traitement automatique de l'information. Généralement, la plupart des systèmes se basent sur les mesures de *MAE*, de *rappel* et de *précision*.

Le MAE : « *Mean Absolute Error* » qui correspond à l'erreur absolue moyenne entre l'évaluation réelle et la prédiction. Cette mesure est calculée par la formule suivante [MH04] :

$$E = \frac{\sum_{j=1}^T |V_{u,j} - P_{u,j}|}{T} \quad (\text{VI.1})$$

avec : $V_{u,j}$: La valeur réelle donnée par l'utilisateur u sur le document j ;

$P_{u,j}$: La prédiction calculée du document j pour l'utilisateur u ;

T : L'ensemble des évaluations réelles attribuées par l'utilisateur u .

La figure suivante présente les différents cas qui peuvent se présenter lors du processus du filtrage des documents par exemple.

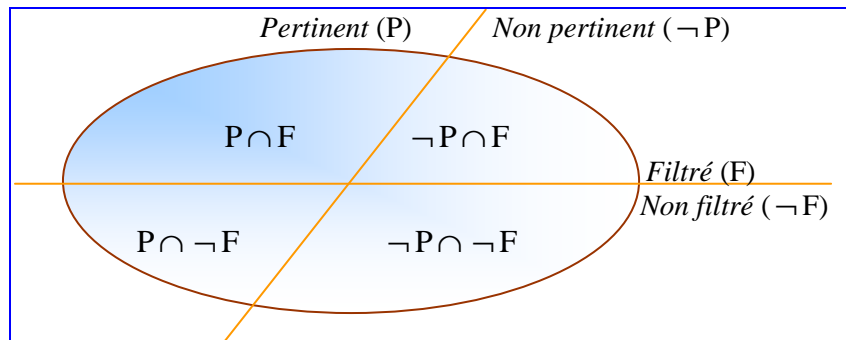


Figure VI.5 – Précision et rappel en Filtrage d'information.

La précision « *Precision* » : La *précision* d'un résultat de filtrage est la proportion des documents pertinents bien filtrés par le système par rapport au nombre total de documents filtrés par ce dernier. La *précision* traduit la qualité du système.

$$\text{Précision} = \frac{|P \cap F|}{|F|} \quad (\text{VI.2})$$

Le rappel « *Recall* » : Le *rappel* d'un résultat de filtrage est la proportion des documents pertinents bien filtrés par le système par rapport à ceux qui sont pertinents dans le corpus. Le *rappel* traduit l'efficacité du système.

Donc, la *précision* et le *rappel* sont des fonctions à maximiser pour un système [Pil00].

$$\text{Rappel} = \frac{|P \cap F|}{|P|} \quad (\text{VI.3})$$

F_Mesure : L'indicateur *F_Mesure* a été proposé par van Rijsbergen pour estimer la qualité globale d'un système [vRi79]. Cette qualité est calculée à partir de la *précision* et du *rappel*. Alors, cet indicateur est calculé par :

$$F_Mesure = 2 \cdot \frac{\text{précision} \cdot \text{rappel}}{\text{précision} + \text{rappel}} \quad (\text{VI.4})$$

Fonction d'efficacité (Utilité) [Nou04] : Cette fonction permet de calculer la qualité du filtrage [Hul98] en tenant compte de la différence entre les documents pertinents filtrés et non pertinents filtrés. Elle est définie comme suit :

$$F(C) = A \cdot R^+ - B \cdot N^+ \quad (\text{VI.5})$$

- où C : Corpus de documents ;
 A : Nombre de documents pertinents ;
 B : Nombre de documents non pertinents ;
 R^+ : Nombre de documents pertinents filtrés ;
 N^+ : Nombre de documents non pertinents filtrés.

4 Résultats

Afin d'évaluer notre approche proposée, nous utilisons la partie « Test » qui contient 296 utilisateurs inscrits dans les trois derniers mois, soit une matrice d'évaluations contenant 254 films jugés par les 296 utilisateurs.

Nous divisons cette évaluation en deux expériences, dans la première, nous évaluons l'efficacité de la méthode proposée pour le processus de la formation des communautés ; la deuxième expérience concerne l'évaluation de la performance de l'approche proposée en utilisant la notion de priorité entre critères pour la production de recommandations.

4.1 Expérience 1 – Formation des communautés

Concernant le critère « *Evaluation* », nous faisons une comparaison du résultat de la formation des communautés dans l'approche du filtrage collaboratif classique avec celle que nous proposons (combinaison de *k-moyenne* et *CAH*). Nous rappelons que, la majorité des systèmes de filtrage collaboratif classiques se basent sur la méthode des *K-voisins les plus proches* dans le processus de construction des communautés [RIS+94]. Par son expérience, Herlocker propose un seuil *K* qui varie de 20 à 50 en raison de la précision des prédictions [Her00].

La figure VI.6 représente l'erreur absolue moyenne « MAE » entre l'évaluation réelle et la prédiction calculée par le système de filtrage collaboratif classique, nous choisissons par exemple 20-voisins les plus proches comme une taille maximale des communautés lors du calcul de la prédiction.

Concernant notre méthode proposée, nous regroupons les utilisateurs en 10 communautés par exemple, c'est-à-dire, les communautés construites regroupent environ 29 utilisateurs (taille moyenne des communautés).

La figure VI.7 représente l'erreur absolue moyenne « MAE » entre l'évaluation réelle et la prédiction calculée par le système, nous prenons en compte la communauté où il appartient l'utilisateur dans l'espace « *Evaluation* » lors du calcul de la prédiction. Cette figure illustre la moyenne du MAE en fonction du nombre des utilisateurs ($k = 10, 20, \dots, 296$).

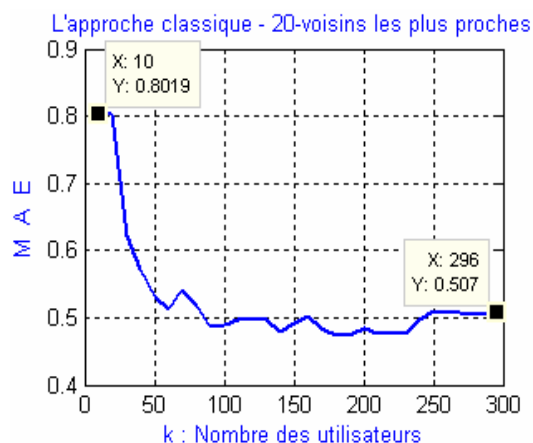


Figure VI.6 – MAE obtenu pour le critère « *Evaluation* » - Approche classique -.

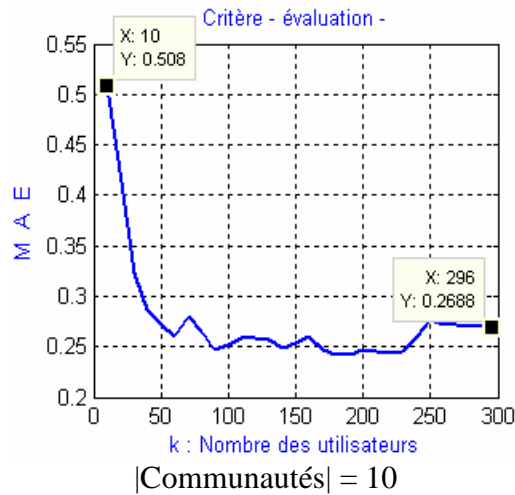


Figure VI.7 – MAE obtenu pour le critère « Evaluation » - Approche proposée -.

Le tableau VI.1 ci-dessous récapitule une comparaison entre l’approche classique et celle que nous proposons pour la formation de communautés en utilisant différentes configurations.

Approche		utilisateurs						Moyenne
		50	100	150	200	250	296	
Classique	20 voisins PP	53,12	48,98	49,18	48,33	51,13	50,70	50,24
	30 voisins PP	52,18	47,84	48,02	47,24	49,90	49,42	49,10
	40 voisins PP	50,99	46,60	46,80	45,99	48,56	48,07	47,83
	50 voisins PP	49,60	45,32	45,61	44,82	47,33	46,83	46,58
Proposée	10 Communautés	27,26	25,16	25,38	24,76	27,60	26,88	26,17
	7 Communautés	26,46	23,03	23,50	22,91	25,33	24,49	24,29
	5 Communautés	23,89	20,92	22,18	21,34	23,55	22,96	22,47
	3 Communautés	21,58	19,96	20,66	19,96	22,04	21,78	21,00

Tableau VI.1 – Comparaison entre approches par le taux d’erreur (MAE %).

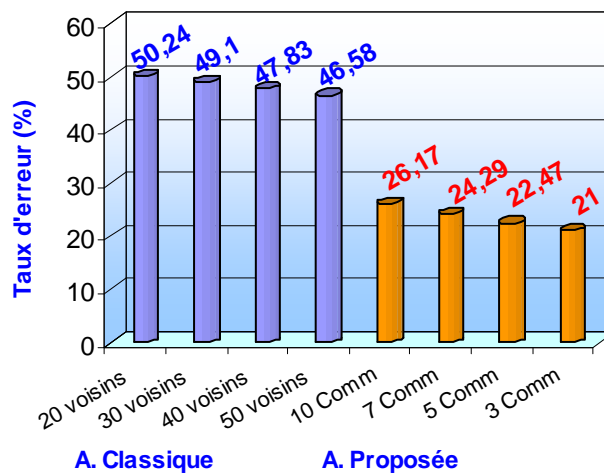


Figure VI.8 – Graphe des MAE comparatif entre les deux approches.

Le tableau VI.1 est schématisé graphiquement par la figure VI.8 ci-dessus. A partir de ce dernier tableau et son graphe associé, nous constatons que l'utilisation de l'approche proposée dans le processus de formation de communautés donne des meilleurs résultats, comparée à l'approche classique.

Nous remarquons aussi que les performances du système se dégradent relativement avec l'augmentation du nombre de communautés (3 5 7 10) dans l'espace « *Evaluation* ». Cette dégradation est justifiée par la diminution des tailles des communautés dans cet espace et le faible nombre d'évaluations en commun des items.

4.2 Expérience 2 – Recommandation par priorité

En premier, nous analysons les résultats de recommandation pour chacun des deux critères simples choisis « *Âge* » et « *Profession* » par exemples. Ensuite, nous introduisons la contrainte de priorité entre ces deux critères et nous analysons les résultats obtenus. Enfin, nous comparons les deux résultats obtenus.

Les résultats de mesure de précision obtenus pour le critère « *Âge* » :

La figure VI.9 représente l'erreur absolue moyenne « MAE » entre l'évaluation réelle et la prédiction calculée par le système, nous prenons en compte la communauté où il appartient l'utilisateur dans l'espace « *Âge* » lors du calcul de la prédiction.

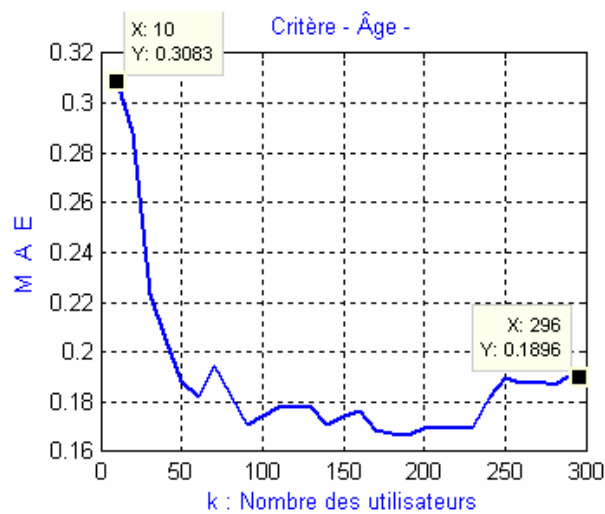


Figure VI.9 – MAE obtenu pour le critère « *Âge* ».

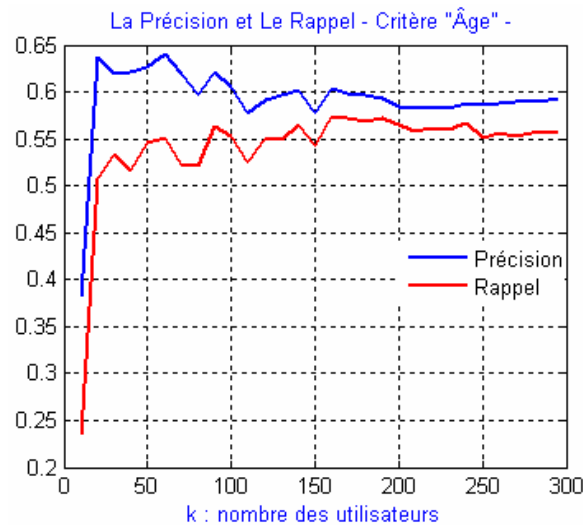


Figure VI.10 – Précision & Rappel obtenus pour le critère « Âge ».

La figure VI.10 représente la précision et le rappel du système, obtenus si nous prenons en compte le critère « Âge » lors du calcul de la prédiction. Nous rappelons que la précision est le rapport entre le nombre des items pertinents bien filtrés (la valeur de la prédiction ≥ 3) par le système et le nombre total des items filtrés par ce dernier. Nous rappelons aussi que le rappel est la proportion du nombre des items pertinents bien filtrés par le système par rapport au nombre total des items considérés pertinents dans le corpus.

Les résultats de mesure de précision obtenus pour le critère « Profession » :

La figure VI.11 représente l’erreur absolue moyenne « MAE » entre l’évaluation réelle et la prédiction calculée par le système, nous prenons en compte dans ce cas, la communauté où il appartient l’utilisateur dans l’espace « Profession » lors du calcul de la prédiction. La figure ci-dessous illustre la moyenne du MAE en fonction du nombre des utilisateurs ($k = 10, 20, \dots, 296$).

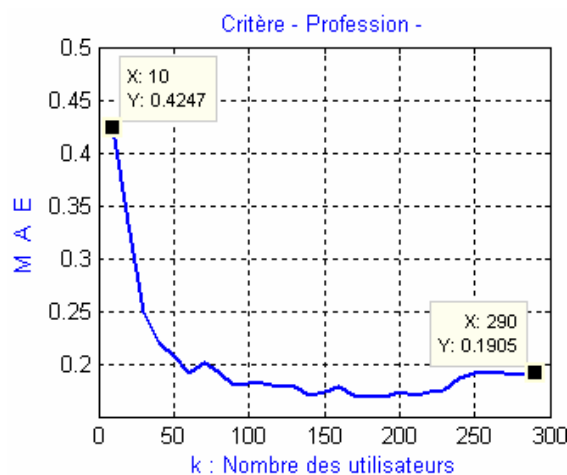


Figure VI.11 – MAE obtenu pour le critère « Profession ».

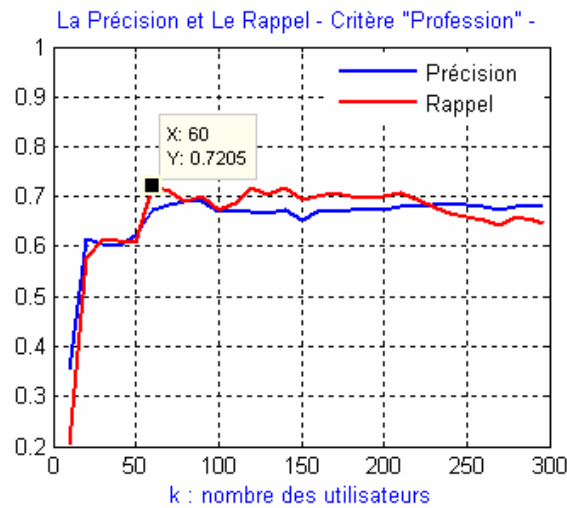


Figure VI.12 – Précision & Rappel obtenus pour le critère « Profession ».

La figure VI.12 représente la précision et le rappel du système obtenus si nous prenons en compte le critère « Profession » lors du calcul de la prédiction. La figure ci-dessus illustre la moyenne de la précision et du rappel pour chaque variant du nombre des utilisateurs ($k = 10, 20, \dots, 296$). Nous remarquons des valeurs optimales : 0,7014 de précision et 0,7205 de rappel correspondent aux nombres des utilisateurs ($k = 90$) et ($k = 60$) respectivement.

Les résultats d’évaluation obtenus pour le critère le plus prioritaire :

Le « MAE » obtenu par l’utilisation de la contrainte de priorité entre les critères lors du calcul de la prédiction, est schématisée par la figure VI.13.

Nous remarquons une diminution de cette mesure par rapport aux critères « Âge » et « Profession » et par conséquent une amélioration de la qualité du système a été retenue. Par exemple pour ($k = 10$), nous obtenons un taux d’erreur de 30,77% si en utilisant la contrainte de priorité entre les critères. Cette valeur est une valeur minimale comparée avec celles de 30,83% et 42,47% correspondent aux critères « Âge » et « Profession » respectivement pour le même nombre d’utilisateurs k .

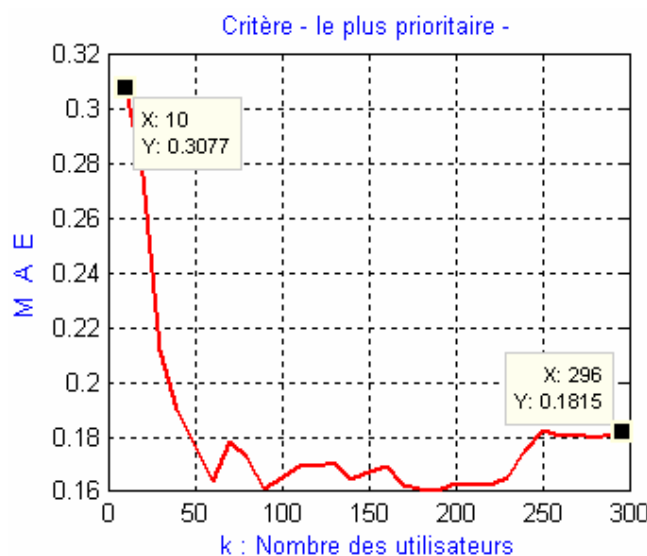


Figure VI.13 – MAE obtenu pour le critère le plus prioritaire.

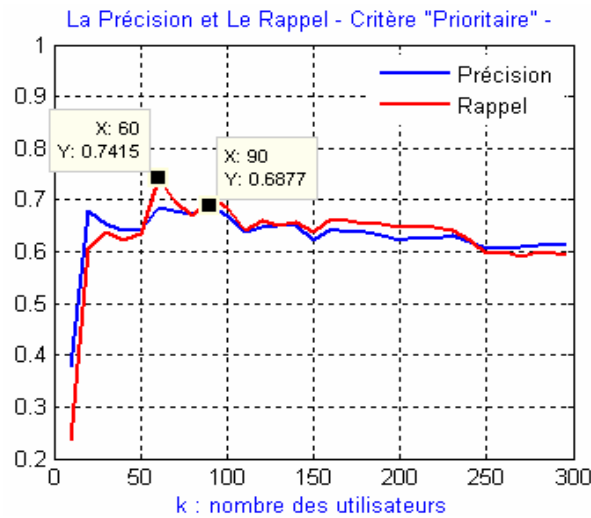


Figure VI.14 – Précision & Rappel obtenus pour le critère le plus prioritaire.

La figure ci-dessus représente la précision et le rappel en utilisant la contrainte de priorité entre les critères lors du calcul de la prédiction en fonction du nombre des utilisateurs (k). Nous remarquons une moyenne optimale de précision (précision = 0,6877) pour une taille de 90 utilisateurs, et une moyenne optimale de rappel (rappel = 0,7415) pour une taille de 60 utilisateurs.

Le tableau VI.2 ci-dessous récapitule une comparaison entre les critères de formation de communautés utilisés dans notre première expérience.

Critère \ utilisateurs	50	100	150	200	250	296	Moyenne
Âge	18,85	17,43	17,44	16,96	18,98	18,96	18,10
Profession	20,75	18,15	17,40	17,29	19,14	19,07	18,63
C. Prioritaire	17,60	16,53	16,69	16,32	18,23	18,15	17,25

Tableau VI.2 – Comparaison entre critères par le taux d’erreur (MAE %).

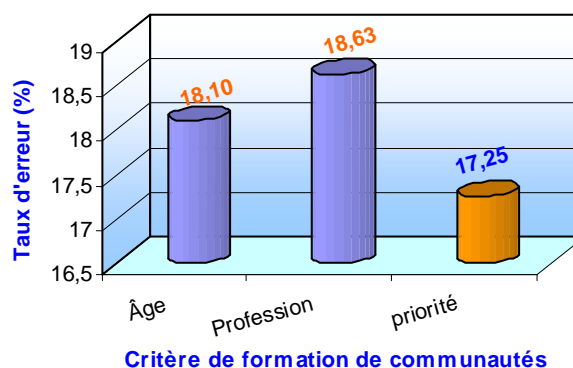


Figure VI.15 – Graphe des MAE comparatif entre critères.

Le tableau VI.2 est schématisé graphiquement par la figure VI.15 ci-dessus. A partir de ce dernier tableau et son graphe associé, il est clair de conclure que l’utilisation de la contrainte de priorité entre les critères de formation de communautés donne des meilleurs résultats, comparée avec ceux de l’approche classique (utilisation d’un seul critère).

4 Conclusion

Nous avons présenté dans ce chapitre les résultats de deux expériences réalisées sur le jeu de données réel « MovieLens ». Une première concerne le choix d'une méthode de formation d'espace de communautés d'utilisateurs suivant l'un des critères complexes, à savoir : « *Evaluation* ». Après comparaison avec l'algorithme de filtrage collaboratif classique, nous constatons que notre proposition permet d'augmenter les performances globales des systèmes de filtrage collaboratif. Une deuxième expérience montrant que l'introduction de la notion de priorité entre les espaces de communautés d'un utilisateur est une alternative pouvant dépasser certaines difficultés des systèmes classiques de filtrage et améliorer le démarrage à froid. Donc, l'utilisation de la multiplicité de critères et l'introduction de la priorité entre ces derniers, sont deux mécanismes qui apportent plus de performance aux systèmes de filtrage collaboratif.

Conclusion et Perspectives

Le domaine du filtrage d'information est un domaine très vaste, compliqué et passionnant à la fois. Il s'agit d'un sujet interdisciplinaire dans la mesure où il touche à plusieurs disciplines à savoir les sciences de l'information et de la communication, l'informatique, la psychologie cognitive, etc. En effet, ce sujet n'apparu qu'avec l'explosion des documents numériques sur le Web. Dans ce contexte, une évolution et un développement très intéressants ont touché les technologies de l'information et de la communication à travers la réalisation des nouveaux outils de communication, le développement des protocoles d'accès à l'information et surtout des normes et des standards de structuration et d'échange de données, afin de satisfaire le besoin en information des utilisateurs.

Dans ce contexte, les systèmes de filtrage d'information collaboratif sont considérés comme des outils particuliers pour filtrer l'information, en tenant en compte le profil de chaque utilisateur du système. La notion de collaboration dans ces systèmes est apparue dans le mécanisme de regroupement des utilisateurs en communautés suivant un critère spécifique. Les outils de filtrage actuels sont limités et incapables de puiser dans le grand gisement d'informations de la toile, d'où la nécessité de se rester à côté de l'utilisateur, pour exploiter toute information caractérisant ce dernier, afin de concevoir des outils de filtrage plus adaptés et plus appropriés répondant le besoin de l'utilisateur avec un minimum d'effort.

Pour cela, nous avons commencé dans ce mémoire par présenter un état de l'art concernant trois grands sujets à savoir les systèmes de filtrage d'information, les systèmes de filtrage collaboratif en particulier et les différentes méthodes de classification et de partitionnement, nous distinguons pour ce dernier point les approches supervisées et celles non supervisées. Ensuite, nous avons proposé de combiner deux méthodes de l'approche non supervisée afin de former autant d'espaces de communautés d'utilisateurs conformément aux divers critères disponibles dans la base des profils des utilisateurs. Nous choisissons la branche non supervisée car ne nécessite pas de classes prédéfinies, la combinaison de l'algorithme des *k-moyennes* avec la classification ascendante hiérarchique pour éviter la limite de la première méthode concernant le choix de *k*, et obtenir un nombre flexible de communautés (par la deuxième méthode). Notre contribution figure aussi dans l'utilisation de la notion de multiplicité de critères et l'intégration de la priorité entre ces critères dans le processus de formation de communautés afin d'améliorer le système de production des recommandations.

Les expériences menées sur une partie restreinte de jeu de données réel MovieLens nous ont permis d'évaluer notre proposition. Dans les travaux futurs, nous projetons d'utiliser des bases de données plus complètes et importantes pour la phase de test de l'approche proposée. Nous envisageons aussi dans cette phase d'exploiter tous les critères disponibles afin d'augmenter en plus le degré de performances des systèmes de filtrage collaboratif.

La perspective de faire rendre visible les communautés aux utilisateurs est un facteur important, afin de motiver l'utilisateur d'exploiter son système, car l'efficacité de ce dernier est étroitement conditionnée par la participation active des utilisateurs.

Bibliographie

- [AS99] Amato G., Straccia U.
User Profile Modeling and Applications to Digital Libraries, Proceedings of the 3rd European Conference on Research and Advanced Technology for Digital Libraries (ECDL99), LNCS, vol. 1696, p. 184-197, France, 1999.
- [AGS97] Agrawal R., Gupta A., Sarawagi S.
Modeling multidimensional databases. Proceedings of the 13th International Conference on Data Engineering (ICDE '97), p. 232-243, UK, 1997.
- [BA02] Bernhard S., Alexander J. S.
Learning With Kernels : Support Vector Machines, Regularization, Optimization and Beyond, MIT Press, 2002.
- [BR99] Baeza-Yates R., Ribeiro-Neto B.
Modern Information Retrieval, ACM press et Addison Wesley, New York, janvier 1999.
- [BS97] Balabanovic M., Shoham Y.
Fab : content-based, collaborative recommendation, Communications of the ACM, vol. 40, n° 3, p. 66-72, Mars 1997.
- [BC92] Belkin N. J., Croft W. B.
Information filtering and information retrieval : two sides of the same coin?, Communication of the ACM, vol. 35, n°12, p. 29-38, ACM, Décembre 1992.
- [BD03] Berrut C., Denos N.
Filtrage collaboratif. In Assistance intelligente à la recherche d'informations, p. 241-269, Hermes - Lavoisier, 2003.
- [Bez81] Bezdek J. C.
Pattern recognition with fuzzy objective function algorithms, New York, Plenum, 1981.
- [BIV92] Bernhard E. B., Isabelle M. G., Vladimir N. V.
A Training Algorithm for Optimal Margin Classifiers In Fifth Annual Workshop on Computational Learning Theory, p. 144-152, Pittsburgh, ACM, 1992.
- [BHK98] Breese J. S., Heckerman D., Kadie C.
Empirical Analysis of Predictive Algorithms for Collaborative Filtering, Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI'98), p. 43-52, Wisconsin, USA, 1998.

- [BK05] Bouzeghoub M., Kostadinov D.
Personnalisation de l'information : Aperçu de l'état de l'art et définition d'un modèle flexible de profils, Actes de la 2^{ème} Conférence en Recherche d'Information et Applications (CORIA'05), p. 201-218, France, 2005.
- [BK04] Bouzeghoub M., Kostadinov D.
Une approche multidimensionnelle pour la personnalisation de l'information, INRIA Rocquencourt et Laboratoire PRiSM, Université de Versailles, France, 2004
- [Bre98] Breese J. S., Heckerman D., Kadie C.
Empirical analysis of predictive algorithms for collaborative filtering, Proceedings of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence, p. 43-52, Juillet 1998.
- [Bru01] Brusilovsky P.
Adaptive Hypermedia, Adaptive Hypertext and user Modelling and user adapted interaction, vol. 11, p. 87-110, 2001.
- [Bur02] Burke R.
Hybrid Recommender Systems : Survey and Experiments, Journal of Personalization Research, User Modeling and User-Adapted Interaction, vol. 12 (4), p. 331-370, Kluwer Academic Publishers, 2002.
- [Cro93] Croft W. B.
Knowledge-based and Statistical approaches to Text Retrieval, IEEE EXPERT, vol. 8, n° 2, p. 8-12, Avril 1993.
- [Del00] Delgado J.
Agent-based Recommender Systems and Information Filtering on the Internet, PhD, Thesis, Nagoya Institute of Technology, Mars 2000.
- [Fou04] Fourer R.
Nonlinear programming frequently asked questions, 2004.
www.unix.mcs.anl.gov/otc/Guide/faq/nonlinear-programming-faq.html
- [GBD03] Gallardo-López M. L., Berrut C., Denos N.
Une approche pour le contrôle de la qualité des Systèmes de Filtrage Collaboratif, Manifestation de Jeunes Chercheurs STIC (MAJESCTIC03), France, 2003.
- [GL86] Gower J. C., Legendre P.
Metric and Euclidean properties of dissimilarity coefficients, Journal of Classification, p. 3, 5-48, 1986.
- [Heb49] Hebb D.
The organization of behavior, New York, Wiley, 1949.
- [Her00] Herlocker J. L.
Understanding and Improving Automated Collaborative Filtering Systems, PhD Dissertation, University of Minnesota, 2000.

- [Hul98] Hull D.
The TREC-7 Filtering Track : Description and analysis, dans les actes de TREC (*Text REtrieval Conference*), University of Maryland, 1998.
- [JMF99] Jain A. K., Murty M. N., Flynn P. J.
Data Clustering : A Review, *ACM Computing Surveys*, vol. 31 (3), p. 264-323, 1999.
- [Kie06] Kien D. N.,
Moteurs de composition pour le système d'information sémantique et adaptatif, mémoire de fin d'études master en informatique, Institut de la francophonie pour l'informatique, 13 septembre 2006.
- [MAS+02] Middleton S. E., Alani H., Shadbolt N. R., De Roure D. C.
Exploiting Synergy Between Ontologies and Recommender Systems, *Proceedings of the 11th International World Wide Web Conference (WWW'02)*, International Workshop on the Semantic Web, Hawaii, USA, 2002.
- [Mar98] Marti A. H.
Support Vector Machines, *IEEE Intelligent Systems*, vol. 13, n° 4, p. 18-28, Jul/Aug, 1998.
- [MH04] McLaughlin R., Herlocher J.
A collaborative filtering algorithm and evaluation metric that accurately model the user experience. In *Proceedings of the ACM SIGIR conference*, p. 329-336, 2004.
- [MLD03] Montaner M., López B., De La Rosa J. L.
A Taxonomy of Recommender Agents on the Internet. *Artificial Intelligence Review*, vol. 19, p. 285-330, Kluwer Publishers, 2003.
- [MSR04] Middleton S. E., Shadbolt N. R., De Roure D. C.
Ontological user profiling in recommender systems, *ACM Transactions on Information Systems (TOIS)*, vol. 22 (1), p. 54-88, ACM Press, 2004.
- [MP00] Miyahara K., Pazzani M. J.
Collaborative Filtering with the Simple Bayesian Classifier, *Proceedings of the 6th Pacific Rim International Conference on Artificial Intelligence (PRICAI'00)*, p. 679-689, Australia, 2000.
- [MKR03] Mirza B. J., Keller B. J., Ramakrishnan N.
Studying Recommendation Algorithms by Graph Analysis, *Journal of Intelligent Information Systems*, vol. 20 (2), p. 131-160, 2003.
- [Mit97] Mitchell T. M.
Machine Learning, MIT Press and McGraw-Hill, 1997.
- [MR88] Mc Clelland J. L., Rumelhart D. E.
Explorations in parallel distributed processing, a handbook of models, programs and examples, MIT Press, Cambridge, 1988.

- [Nou04] Nouali O.
Filtrage d'information textuelle sur les réseaux – une approche hybride.
Thèse doctorat, CNRS/France – USTHB/Alger, 2004.
- [Ngu06] Nguyen A. T.
COCOFil2 : Un nouveau système de filtrage collaboratif basé sur le modèle
des espaces de communautés, Thèse doctorat, université Joseph Fourier -
Grenoble I, 2006.
- [Paw04] Pawlak Z.
Some Issues on Rough Sets. Transaction on Rough Sets I, LNCS 3100,
2004.
- [PB95] Pal N. R., Bezdek J. C.
On cluster validity for the fuzzy c-means model, IEEE Transaction on fuzzy
systems, vol. 3, p. 3, 370-379, 1995.
- [PGF03] Perugini S., Gonçalves M. A., Fox. E. A.
A Connection-Centric Survey of Recommender Systems Research, Journal
of Intelligent IS, vol. 23 (1), 2003.
- [Pil00] Pillet V.
Méthodologie d'extraction automatique d'information à partir de la
littérature scientifique en vue d'alimenter un nouveau système
d'information, Thèse doctorat en sciences, Aix-Marseille III, Janvier 2000.
- [Qui93] Quinlan J. R.
C4.5 : Programs for Machine Learning, Morgan Kaufmann, San Mateo,
USA, 1993.
- [Qui86] Quinlan J. R.
Induction of decision trees, Machine Learning, vol. 1 (1), p. 81-106, 1986.
- [RIS+94] Resnick P., Iacovou N., Suchak M., Bergstrom P., Riedl J.
GroupLens : An Open Architecture for Collaborative Filtering of Netnews,
Proceedings of the Conference on Computer Supported Cooperative Work
(CSCW'94), NC, USA, 1994.
- [RHW86] Rumelhart D., Hinton G., Williams R.
Parallel distributed processing, vol. 1, Cambridge, MIT Press, 1986.
- [Ros58] Rosenblatt F.
The Perceptron : a probabilistic model for information storage and
organization in the brain, Psychological Review 65, p. 386-408, 1958.
- [Sha48] Shannon C.
A mathematical theory of information, The Bell System Technical Journal,
27, 1948.
- [Ter93] Terry. D. B.
A tour through tapestry. In COOCS, p. 21-30, ACM, 1993.

- [Tma02] Tmar Mohamed.
Modèle auto adaptatif de filtrage d'information : apprentissage incrémental du profil et de la fonction, Université Paul Sabatier, Toulouse, 2002.
- [Vap95] Vapnik V. N.
The nature of statistical learning theory, Springer-Verlag, 1995.
- [vRi79] van Rijsbergen C. J.
Information Retrieval, Butterworth Publisher, 1979.
- [XB91] Xie X. L., Beni G. A.
Validity measure for fuzzy clustering, IEEE Transactions on pattern analysis and machine intelligence, vol. 13, p. 8, 841-846, 1991.
- [Zei76] Zeigler B.
Theory of Modeling and Simulation, Malabar, Robert E. Krieger Publishing Company Inc, 1976.
- [ZLB05] Zemirli N., Lechani T. L., Boughanem M.
Accès personnalisé à l'information : proposition d'un profil utilisateur multidimensionnel, 7th ISPS Algérie, Mai 2005.

http://www.cavalex.com/pdf/livre_touzet.pdf

http://fr.wikipedia.org/wiki/Neurone_formel

<http://www.peoi.org/Courses/Coursesfr/neural/>

[http://www.grouplens.org/.](http://www.grouplens.org/)

Annexe

1 Outil d'évaluation

Afin d'évaluer notre approche proposée, nous utilisons le langage MATLAB version 7.1.0. MATLAB pour MATrix LABoratory, est une application qui a été conçue afin de fournir un environnement de calcul matriciel efficace, interactif et portable. Avec ses fonctions spécialisées, MATLAB peut être aussi considéré comme un langage de programmation adapté pour les problèmes scientifiques.

MATLAB est constitué d'un noyau relativement réduit, capable d'interpréter puis d'évaluer les expressions numériques matricielles qui lui sont adressées :

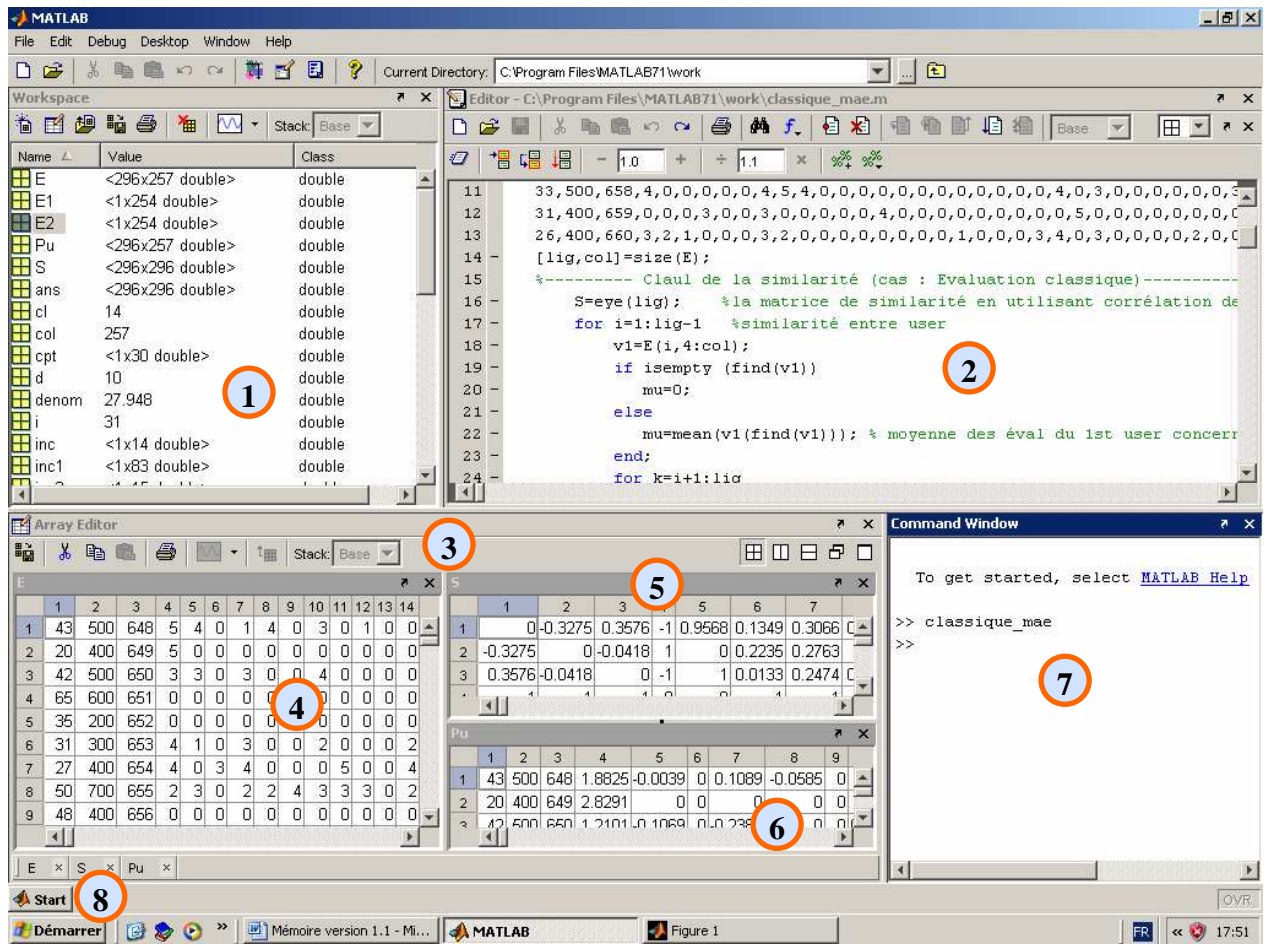
- ✗ Soit directement au clavier depuis une fenêtre de commande ;
- ✗ Soit sous forme de séquences d'expressions ou scripts enregistrées dans des *fichiers-texte* appelés *m-files* et exécutées depuis la fenêtre de commande ;
- ✗ Soit plus rarement sous forme de fichiers binaires appelés *mex.files* générés à partir d'un compilateur *C* ou *Fortran*.

Ce noyau est complété par une bibliothèque de fonctions prédéfinies, très souvent sous forme de fichiers *m-files*, et regroupés en *paquetages* ou *toolboxes*. A côté des *toolboxes* requises *local* et *matlab*, il est possible d'ajouter des *toolboxes* spécifiques à tel ou tel problème mathématique, *Optimization Toolbox*, *Signal Processing Toolbox* par exemple ou encore des *toolboxes* créés par l'utilisateur lui-même. Un système de chemin d'accès ou *path* permet de préciser la liste des répertoires dans lesquels MATLAB trouvera les différents fichiers *m-files*.

2 Captures d'écran

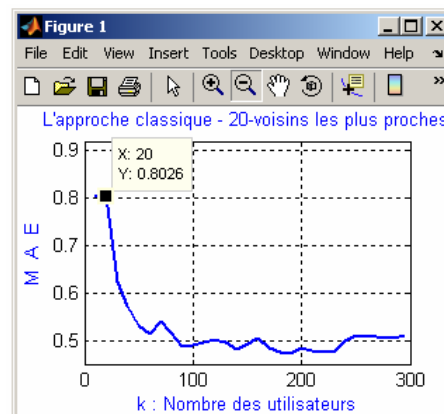
La première capture d'écran (01) représente les principales fenêtres du langage de programmation MATLAB, tel que :

- ① Fenêtre de l'espace de travail « *Workspace* » ;
- ② Fenêtre de l'éditeur « *Editor* » ;
- ③ Fenêtre de l'éditeur de tableau « *Array Editor* » contient les résultats d'exécution :
 - ④ La matrice initiale « *E* » ;
 - ⑤ La matrice de la similarité « *S* » ;
 - ⑥ La matrice de la prédiction « *Pu* » ;
- ⑦ Fenêtre de commande « *Command Window* » ;
- ⑧ Le bouton démarrer « *Start* ».



Capture d'écran – 01

La deuxième capture d'écran (02) illustre le résultat d'exécution du programme que MATLAB affichera dans une fenêtre indépendante.



Capture d'écran – 02

Glossaire

Nous rappelons ci-dessous les concepts fondamentaux fréquemment utilisés dans le domaine des systèmes de filtrage d'information :

A

- **Apprentissage automatique** : paramétrage d'un système automatique par des données à partir desquelles le système induit des règles.

B

- **Base de profil** : ensemble des profils stockés dans la base de données du système de filtrage collaboratif.

C

- **Corpus** : ensemble d'objets soumis à un traitement donné (ex. : corpus des items, corpus des utilisateurs,...).
- **Communauté** : ensemble des utilisateurs qui sont proches les uns des autres selon un critère de comparaison.

E

- **Extraction d'information (*information extraction*)** : activité de recherche d'information visant la mise à jour automatique de bases de données [Nou04].
- **Espace de communautés** : ensemble des communautés formées par un critère donné.
- **Evaluation** : Formalisation du goût d'un utilisateur pour un document. L'évaluation peut être « explicite » : cela signifie qu'elle est donnée par l'utilisateur sous la forme d'une note sur une échelle de [1 à 10] par exemple. Elle peut aussi être « implicite », et dans ce cas c'est le système qui interprète certaines actions ou certains comportements de l'utilisateur comme des évaluations : par exemple le fait d'imprimer un document, ou le télécharger, ou de passer un certain temps à le lire...

F

- **Filtrage d'information** : sélection et acheminement de documents extraits d'un flux d'information vers une personne ou un groupe de personnes, en se basant sur leur(s) profil(s) à long terme.

- **Formation de communautés** : résultat de l'estimation de la proximité entre utilisateurs via la comparaison de leurs profils. Les communautés ne sont pas toujours de façon explicite dans les systèmes. Généralement, la formation de communauté reste implicite, et n'intervient qu'au moment de calcul de prédiction.

M

- **Moteur de recherche** : est un programme informatique qui permet aux utilisateurs de faire des recherches sur les documents disponibles dans des sources de données (bases de données, Internet, Intranets particuliers, etc.) [Nou04].

P

- **Profil** : liste des évaluations passées d'un utilisateur. Etant donné un utilisateur, son profil est vu comme un vecteur formé des couples (document, évaluation). La date de l'évaluations est souvent ajouté, donnant ainsi lieux à des triplets (document, date, évaluation). D'une façon générale, un profil est une description d'un utilisateur selon plusieurs dimensions : informations personnelles, centres d'intérêt, etc.
- **Polymorphisme** : capacité d'un utilisateur appartenant à la fois à plusieurs communautés.
- **Prédiction** : estimation que le système fait de l'intérêt qu'un document présente pour un utilisateur. C'est sur la base de la prédiction que le document est recommandé ou non ; la prédiction tient compte des évaluations des autres sur ce document, et de leur proximité avec l'utilisateur à qui est destiné la recommandation.
- **Précision (*precision*)** : taux de documents pertinents bien filtrés par un système de filtrage d'information, par rapport au nombre total de documents filtrés par ce dernier.

R

- **Rappel (*recall*)** : taux de documents pertinents bien filtrés par le système de filtrage d'information par rapport à ceux qui sont pertinents dans le corpus.
- **Recherche d'information (*information retrieval*)** : activité visant à (re)trouver et présenter l'information pertinente à chaque utilisateur des systèmes de recherche d'information [Nou04].

S

- **Système** : Un système peut être défini comme un ensemble d'éléments en interaction entre eux et avec leur environnement, de sorte que cet ensemble peut être considéré comme un tout.

- **Sélection des recommandations** : choix des documents dont la valeur de prédiction calculée est considérée pertinente pour l'utilisateur en question. Pour réaliser une telle sélection, un seuil de pertinence est défini.

T

- **Table de communautés** : table de décision représentant des espaces de communautés.
- **TREC¹** : conférence internationale d'évaluation de systèmes de fouille de textes. Elle est consacrée à différentes activités de recherche d'information, de l'indexation des documents, etc.

V

- **Vecteur de positionnement** : vecteur des communautés d'un utilisateur, représentant le polymorphisme de son positionnement au sein des communautés.

¹TREC : Text REtrieval Conference. <http://trec.gov/>