

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Mouloud MAMMARI de TIZI-OUZOU
Faculté de Génie Electrique et Informatique
Département d'Informatique



Domaine : Mathématiques et Informatique

Filière : Informatique

Spécialité : Ingénierie des Systèmes d'Information

THEME

Problème du Démarrage à Froid dans les Systèmes de Recommandation

Présenté par :

M^{elle} BERRICHI Thanina

&

M^{elle} DJOUAHER Lila

Soutenu publiquement le : 11/11/2020

Devant le jury composé de :

Mme FELLAG Samia
Mme ACHEMOUKH Farida
Mme BENTAYEB Mouna

U.M.M.T.O. Présidente
U.M.M.T.O. Promotrice
U.M.M.T.O. Examinatrice

Année universitaire : 2019/2020

Remerciements

On adresse nos premiers remerciements à dieu qui nous a donné la volonté, le courage ainsi que la patience afin de réaliser ce travail.

On exprime toute notre gratitude à notre promotrice madame Achemoukh Farida.

Merci pour sa disponibilité indéfectible, son aide ainsi que ses remarques judicieuses qui nous ont permis de faire progresser ainsi que d'enrichir notre travail.

On exprime notre profonde reconnaissance et nos chaleureux remerciements aux membres de jury qui ont fait l'honneur de bien vouloir évaluer et juger notre travail.

C'est avec un plaisir non dissimulé qu'on exprime notre reconnaissance à nos chers parents. Merci d'être toujours là pour nous, pour nous encourager durant toute la période de réalisation de ce travail. Sans oublier nos chers frères et sœurs et ainsi nos amis.

Finalement nous remercions tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail.

Merci

Dédicace

*Je dédie ce travail
à
La mémoire de mon père
La mémoire de mon frère Nabil
La mémoire de ma grand-mère
Ma chère maman que dieu lui accorde Santé et longue vie.
Mon frère Jugurtha
Mes sœurs : Rabéa et Radia et Thiziri
Mes neveux : Meziane et Said
Mes nièces : Imane et Yasmine et Lydia
Mon cher binôme : Lila
Mes amies : Lydia et Aicha et Fatiha et Zineb ainsi qu'à
toutes les personnes qui m'y sont chère.*

BERRICHI THANINA

Je dédie ce mémoire à ...

Allah

Qu'il nous couvre de sa bénédiction.

AMEN

A mes chers parents

C'est une évidence de dire que sans vous rien de tout cela n'aurait été possible, mais c'est tellement vrai. Vous m'avez toujours soutenue dans les bons et les mauvais moments. Tous les mots ne sauraient exprimer l'immense amour que je vous porte, ni la profonde gratitude que je vous témoigne pour tous les efforts et les sacrifices que vous avez consentis pour mon instruction et mon bien-être. Après tout, c'est grâce à vous que je suis qui je suis avec mes qualités et mes défauts.

Merci de m'avoir tant donné sans attendre à recevoir.

Je vous dédie ce modeste travail en guise de ma reconnaissance éternelle pour toute l'affection que vous n'avez jamais cessé de me prodiguer. Que Dieu tout puissant vous garde et vous procure santé et bonheur pour que vous demeuriez le flambeau illuminant le chemin de ma vie.

A mes sœurs

Que ce travail soit pour vous la preuve de mon attachement, Vous m'avez soutenu et comblé tout au long de mon parcours Puisse dieu vous procurer bonheur, prospérité et la réussite dans votre vie.

A toute Ma famille

ℒ

Mes très chers amis

Je vous dédie ce travail avec mes sincères remerciements

Que dieux vous procure joie, bonheur et réussite et que notre amitié reste à jamais

A

Ma chère binôme Thanina et sa famille

A

Tous ceux qui me sont chers et que j'ai involontairement omis de citer

DJOUAHER LILA.

Sommaire

Table des matières

Liste des tableaux

Liste des figures

Introduction générale..... 1

Chapitre 1 : Les systèmes de recommandation

1.1.Introduction 4

1.2.Définition..... 4

1.3.Concepts de base, notations et notions liées 5

1.3.1. Les entités utilisateur et items 5

1.3.2. Evaluation 5

1.3.3. Matrice d'évaluation 6

1.3.4. Notion de communauté 7

1.3.5. Notion de profil 7

1.3.5.1. Profil utilisateur 7

1.3.5.2. Profil item 8

1.4.Classification des systèmes de recommandation 8

1.4.1. La recommandation basée sur le contenu 9

1.4.1.1. Représentation d'un item..... 10

1.4.1.2. Représentation des utilisateurs..... 10

1.4.1.3. Les types de recommandation basée sur le contenu 11

1.4.1.4. Les avantages et les inconvénients de la recommandation basée sur le contenu..... 14

1.4.2. Le Filtrage Collaboratif..... 15

1.4.2.1. Les techniques de filtrage..... 15

1.4.2.2. Mesure de similarité 19

1.4.2.3. Recommandation multicritères 20

1.4.2.4. Avantages et inconvénients de la recommandation collaborative : 20

1.4.3. La recommandation hybride 21

1.4.3.1. Les méthodes d'hybridation 21

1.4.3.2. Exemple de système de recommandation hybride..... 23

1.5.Conclusion..... 26

Chapitre 2: La modélisation du profil utilisateur dans les systèmes de recommandation

2.1. Introduction	27
2.2. La modélisation du profil utilisateurs	27
2.2.1. Représentation du profil utilisateur	27
2.2.1.1. Représentation ensembliste	27
2.2.1.2. Représentation sémantiques	29
2.2.1.3. Représentation multidimensionnelle	30
2.2.2. Construction du profil utilisateur	32
2.2.2.1. Acquisition des données	33
2.2.2.1.1. La méthode d'acquisition implicite	33
2.2.2.1.2. La méthode d'acquisition explicite	35
2.2.2.1.3. La méthode d'acquisition hybride	37
2.2.2.2. Prétraitement des données	37
2.2.3. Adaptation et mise à jour du profil utilisateur	38
2.3. Conclusion	39

Chapitre 3: Le démarrage à froid dans les systèmes de recommandation

3.1. Introduction	40
3.2. Problématique du Démarrage à froid	40
3.2.1. Le démarrage à froid pour un nouveau système	41
3.2.2. Le démarrage à froid pour une nouvelle ressource	42
3.2.3. Le démarrage à froid pour un nouvel utilisateur	43
3.3. Les solutions proposées	44
3.3.1. Filtrage collaboratif	44
3.3.2. Filtrage basé sur le contenu	45
3.3.3. Filtrage hybride	47
3.3.3.1. Combinaison avec le filtrage basé sur le contenu	47
3.3.3.2. Combinaison avec d'autres techniques	48
3.4. Conclusion	49

Chapitre 4: Solution proposée pour remédier au problème du démarrage à froid

4.1. Introduction	50
--------------------------------	-----------

4.2. Problématique et motivation	50
4.3. Description de l'approche.....	51
4.3.1. Architecture de l'approche explicite	51
4.3.1.1. Modélisation du profil utilisateur	52
4.3.1.2. Exploitation du profil utilisateur dans le processus de recommandation	53
4.3.1.3. Profil item	53
4.3.1.4. Illustration de l'approche explicite proposée	54
4.3.2. Description de l'approche implicite	58
4.3.2.1. Architecture de l'approche	58
4.3.2.2. Exploitation du profil utilisateur dans le processus de recommandation	61
4.3.2.3. Inférence du profil utilisateur et du profil item	61
4.3.2.4. Illustration de l'approche implicite proposée.....	62
4.4. Conclusion.....	68

Chapitre 5: Implémentation de la solution proposée

5.1. Introduction	69
5.2. Environnement de développement	69
5.3. Implémentation de notre solution	72
5.3.1. Jeu de données	72
5.3.1.1. Table Links	72
5.3.1.2. Table Movie	72
5.3.1.3. Table Tags.....	72
5.3.1.4. Table Rating	72
5.3.2. Présentation de l'application	73
5.3.2.1. Interface d'accueil.....	73
5.3.2.2. Interface de connexion.....	73
5.3.2.3. Interface d'inscription	74
5.3.2.4. Interface de la fenêtre principale.....	75
5.4. Conclusion.....	76

Conclusion générale

Conclusion générale.....	77
---------------------------------	-----------

Bibliographie

Tableau 1.1 : exemple de matrice d'évaluation.

Tableau 2.1 : exemple montrant les problèmes rencontrés et les solutions proposées.

Tableau 4.1. Représentation binaire des films.

Tableau 4.2. Les notes des utilisateurs attribués pour les films

Tableau 4.3. Représentation des totales des genres

Tableau 4.4. Représentation des vecteurs des films par rapport à l'utilisateur après normalisation.

Tableau 4.5. Vecteurs profil utilisateur

Tableau 4.6. Vecteurs des films

Tableau 4.7. Degré de pertinence des films par rapport à l'utilisateur.

Tableau 4.8. Représentation binaire des films de la collection MovieLens.

Tableau 4.9. Représentation binaire des films de l'utilisateur.

Tableau 4.10. Nombre de films dans chaque genre.

Tableau 4.11. Vecteur de films par rapport à l'utilisateur.

Tableau 4.12. Vecteur des films du profil item après normalisation.

Tableau 4.13. Degré de similarité entre les vecteurs pondérés des films et les vecteurs pondérés des utilisateurs.

Figure 1.1. : Schéma expliquant le fonctionnement d'un système de recommandation

Figure 1.2 : le processus de recommandation hybride

Figure 1.3 : Conception d'hybridation monolithique.

Figure 1.4 : Conception d'hybridation parallèle.

Figure 1.5 : Conception d'hybridation tubulaire.

Figure 2.1 : Exemple de profil représenté par des mots clés.

Figure 2.2 : un extrait d'un profil utilisateur sémantique.

Figure 2.3 : Méta-modèle de profil utilisateur.

Figure 2.4. Construction du profil utilisateur.

Figure 4.1. Architecture de l'approche explicite.

Figure 4.2. Image de MovieLens.

Figure 4.3. Architecture de l'approche implicite.

Figure 5.1. Interface d'accueil

Figure 5.2. Interface de connexion

Figure 5.3. Interface de connexion avec Facebook

Figure 5.4. Interface d'inscription

Figure 5.5. Interface de la fenêtre principale

Figure 5.6. Interface d'un exemple de résultat affiché à l'utilisateur

Introduction générale

Le monde de la science sait pertinemment le rôle que s'octroie un article de recherche dans la conquête de l'univers du savoir. Un univers en permanente activité, qui ne cesse de pousser ses frontières de jour en jour avec la publication de travaux dans les conférences et les journaux scientifique.

La recherche d'information est née assez tôt par rapport aux autres domaines de l'Informatique.

En 1945, Vannevar Bush a écrit son article classique *Tel que nous pourrions penser* et publié dans *Atlantic Monthly*. Cet article décrit le système <<Memex >> qui contient plusieurs idées révolutionnaires comme l'accès aux grandes quantités de données en utilisant des nouveaux matériels pour stocker les documents, créer des liens entre les documents etc.

Ce système est souvent considéré comme le premier modèle d'une bibliothèque numérique et du Web. Cependant, dans ce système Bush n'a pas prévu de mécanisme pour indexer et rechercher automatiquement des documents.

Dans les années 1950 et 1960, avec le développement de la technologie de l'ordinateur, les premiers systèmes de recherche d'information qui utilisent des ordinateurs ont été construits. Parmi eux, le système le plus notable est le système SMART développé par Gerard Salton et son équipe. Ce système a introduit plusieurs concepts importants comme le modèle vectoriel, la méthode de retour de pertinence, le regroupement de documents (clustering) etc.

Les systèmes d'informations actuels sont caractérisés par leur volume croissant, leur hétérogénéité, et par le fait qu'ils ne sont pas suffisamment adaptés aux besoins des utilisateurs. Au vu de l'état actuel de ces systèmes en termes d'hétérogénéité de domaines, de sources, de représentation et de structuration des informations, l'accès à une information pertinente et adaptée aux utilisateurs est un vrai challenge. Les besoins de l'utilisateur sont difficiles à traiter, d'une part, parce qu'ils ne sont pas formulés explicitement et, d'autre part, parce qu'ils sont évolutifs.

L'utilisation des systèmes de recommandation est devenue une nécessité vu qu'ils permettent d'identifier automatiquement les préférences des utilisateurs à travers leurs

interactions avec le système en se basant sur le feedback implicite [HU *et al.*, 2008] [SALTON *et al.*, 2013] ou feedback explicite [BOB *et al.*, 2010] ou les deux ensembles [SHA et Sum, 2013] [SAL et KAM, 2013], pour leur suggérer des recommandations en utilisant le filtrage d'information.

Pour générer des recommandations, un certain nombre d'approches ont été identifiées [BURKE, 2002], où les plus utilisées sont le Filtrage Collaboratif [BAL et SHO, 97] et le Filtrage basé Contenu [PAZZANI et BILLSSUS, 2007]. Les deux techniques ont leurs forces et faiblesses [BURKE, 2002], où l'hybridation entre eux a été rapidement adoptée pour profiter de leurs avantages.

La majorité des systèmes de recommandation souffre du problème de démarrage à froid et plusieurs ressources d'informations sont exigées. Notre travail se situe dans ce contexte, notamment dans le cadre des systèmes de recommandation des films. Nous adoptons une approche basée sur le contenu et une approche hybride qui évite le problème du démarrage à froid.

Notre travail est réparti en cinq chapitres :

Chapitre 1 : Le premier chapitre présente une vue générale sur les systèmes de recommandation, nous définissons d'abord ce qu'est un système de recommandation. Ensuite nous détaillons ses différents types, ainsi les avantages et inconvénients de chaque approche.

Chapitre 2 : Dans le deuxième chapitre nous verrons les modèles de représentation de profils utilisateurs, les méthodes d'acquisition des informations des utilisateurs, les techniques de constructions et de mise à jour des profils.

Chapitre 3 : Dans ce chapitre, nous expliquons le problème du démarrage à froid.

Chapitre 4 : Dans ce chapitre, nous expliquons les deux approches que nous avons proposées.

Chapitre 5 : Implémentation des deux approches proposées, ce dernier chapitre se base sur les détails d'implémentation et de mise en œuvre de nos approches, ainsi qu'à la présentation d'un exemple des résultats obtenus.

Nous terminons notre mémoire par une conclusion générale.

CHAPITRE 1 :
Les systèmes de recommandation

1.1. Introduction

Fréquemment, lors de la recherche d'une information dans différents domaines tels que la recherche documentaire, les sciences de gestion et marketing, les sciences cognitives et les théories d'approximation ...etc. On se retrouve souvent devant une très grande liste des choix ; Évaluer ces choix et trouver ce qui nous convient est une tâche difficile et prend beaucoup de temps.

Pour faire face à cette surcharge d'information, de nouvelles techniques ont été développées ; à savoir les systèmes de recommandation.

L'objectif des systèmes de recommandation est de présenter les informations (films, musique, livres, images, pages Web, etc.) qui sont susceptibles d'intéresser l'utilisateur et de l'informer de l'existence d'une information à laquelle il n'a pas forcément fait attention afin de l'aider à trouver ce qu'il cherche dans une petite période du temps.

Dans ce chapitre nous expliquons qu'est-ce qu'un système de recommandation et puis définir les concepts de base et les notations et les notions liées. Ensuite nous représentons les techniques de recommandation avec leurs avantages et inconvénients.

1.2. Définition

La recommandation sert à guider l'utilisateur dans son exploration des données afin qu'il trouve des informations pertinentes.

Les systèmes de recommandation sont des systèmes capables de fournir des recommandations personnalisées, permettant de guider l'utilisateur vers des ressources intéressantes et utiles au sein d'un espace de données important. [BURKE, 2002]

La Figure 1.1 illustre le processus d'un système de recommandation qui consiste à filtrer et extraire l'information pertinente pour un utilisateur particulier.

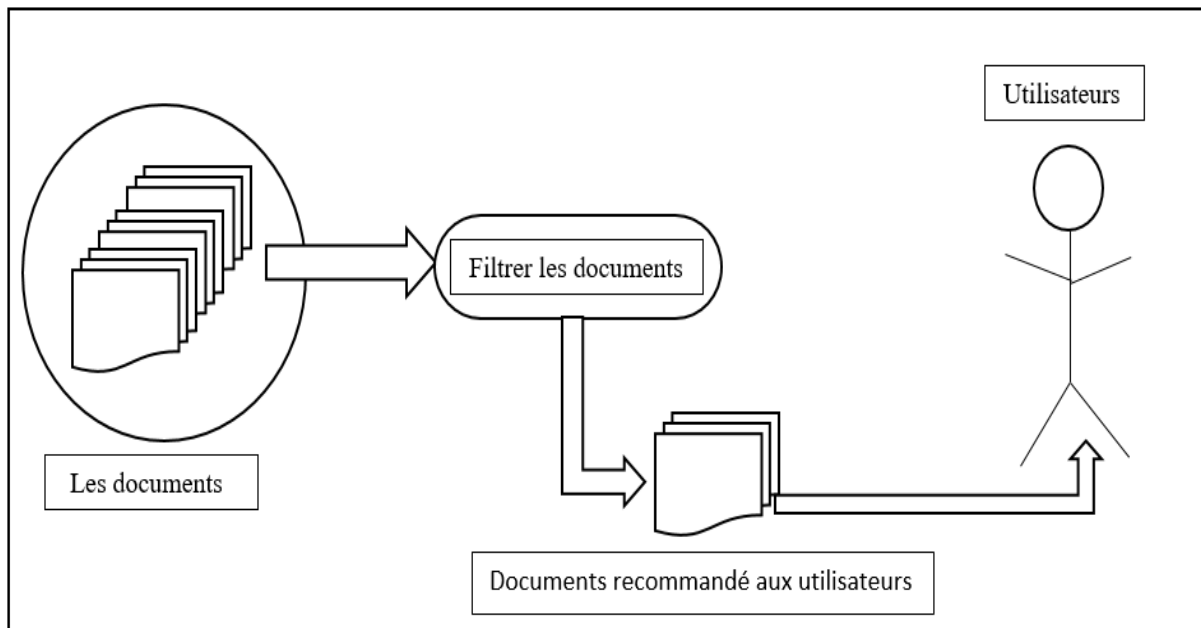


Figure 1.1. : Schéma expliquant le fonctionnement d'un système de recommandation

1.3. Concepts de base, notations et notions liées

Dans cette section, nous définissons quelques concepts relatifs au système de recommandation.

1.3.1. Les entités utilisateur et items

Les deux entités de base qui apparaissent dans tous les systèmes de recommandation sont : les utilisateurs et les items.

Utilisateur : est la personne qui utilise un système de recommandation, donne son opinion sur diverses items et reçoit les nouvelles recommandations du système.

Item : est l'entité qui désigne ce que le système recommande aux utilisateurs. Notons qu'un item peut être un individu ou un ensemble d'individus suggérés à l'utilisateur dans les réseaux sociaux (suggestion d'amis sur Facebook), incluant tout produit susceptible d'être vendu (livres, produit...etc. dans les sites e-commerce tel que Amazon.com), vu (films dans les sites de TV en lignes tel que Netflix), écouté (la musique) ou lu (tel que les informations dans les journaux en ligne, les revues dans les bibliothèques numériques)...etc.

1.3.2. Evaluation

L'évaluation représente la préférence ou non de l'utilisateur pour l'item. Elle est appelée note. C'est une valeur **numérique** ou **binaire**.

Valeur numérique : on vote dans une échelle quelconque. La plus utilisée est [1-5]. [BURKE , 2002]

Valeur binaire : aimer / ne pas aimer, bon / mauvais ... etc.

L'évaluation donné par un utilisateur **U** à un item **I** est représenté par $R_{U,I}$ ou par un triplet $\langle U,I,R \rangle$.

Exemple :

- Une note de 5 désigne une grande préférence (utilisateur a aimé).
- Une note de 1 désigne une faible préférence (utilisateur n'a pas aimé).

Un utilisateur peut attribuer directement une note à un item en donnant une valeur numérique ou binaire via un système appelée « évaluation explicite ». [BURKE, 2002].

1.3.3. Matrice d'évaluation

L'ensemble de tous les triplets du système $\langle U,I,R \rangle$ sont enregistrés dans une base de données creuse¹ appelée « MATRICE D'EVALUTION » (*rating matrix*) ou encore « MATRICE UTILISATEUR-ITEM » (*user-item matrix*) et elle est noté par **R** .

- Chaque ligne correspond aux évaluations fournies par un seul utilisateur.
- Chaque colonne correspond aux évaluations d'un seul item par l'ensemble des utilisateurs.

Nous illustrons dans le tableau (**tableau 1.1**) un exemple d'une matrice d'évaluation pour trois utilisateurs et quatre films. Les valeurs marquées par « ? » indiquent que l'utilisateur n'a pas donnée d'avis.

	Harry Potter	Blade	Titanic	Casa de Papel
U1	4	?	5	3
U2	4	5	3	?
U3	2	1	?	4

Tableau 1.1 : exemple de matrice d'évaluation.

¹ Creuse : un système peu couplé, la majorité de ces entrées sont des zéro.

La matrice d'évaluation utilisateur-item est l'entrée des systèmes de recommandation et la base technique du filtrage collaboratif, qui utilisent des préférences (notes) pour la génération des recommandations.

Les tâches du système de recommandation : le système de recommandation se focalise sur deux tâches :

- **Prédiction** : C'est le calcul de la note probable que l'utilisateur va attribuer à un item qu'il n'a pas encore vu ou évalué. Le calcul de la prédiction se base sur l'utilisation des notes données par **les voisins de l'utilisateur** (*prédiction basée utilisateur*) ou attribuées aux items voisins de l'item test évalué par **l'utilisateur actif** (*prédiction basée item*) ou bien donnée par **un modèle** (*prédiction basée modèle*). Ensuite, les items ayant les plus grandes valeurs de prédiction seront recommandés à l'utilisateur. [MAATALLAH, 2016]
- **Recommandation** : C'est une préparation d'une liste d'items (les N-Items) qui plaira à l'utilisateur. Le calcul des listes de recommandation se fait en attribuant des scores pour les items selon leurs popularités ou leurs préférences.

La liste des tops-n recommandations n'est pas forcément la liste des N-Items avec les plus hautes valeurs de prédiction. [BENOUARET, 2018]

1.3.4. Notion de communauté

Une communauté est un ensemble d'utilisateurs similaires qui partagent les mêmes préférences et qui sont regroupés relativement à un critère donné. Selon chacun de ces critères, les communautés créées par le système varient et les positions des utilisateurs dans ces communautés changent. Par conséquent, chaque utilisateur peut appartenir à autant de communautés qu'il y a de critères pour leur formation. [NGUYEN et *al.*, 2006].

1.3.5. Notion de profil

Le profil d'un objet est un ensemble de qualités et de caractéristiques permettant de l'identifier ou de le représenter. Il y'a deux type : Profil utilisateur et Profil item.

1.3.5.1. Profil utilisateur

La description des caractéristiques de l'utilisateur, qui peuvent être ses centres d'intérêts ou ses préférences exprimées sous forme d'évaluation,... etc.

Afin de construire le profil utilisateur, il existe plusieurs approches d'acquisition de ces informations :

- **Approches manuelles** : se fondent sur l'intervention de l'utilisateur.
- **Approches automatiques ou semi-automatiques** : déduisent automatiquement le profil de l'utilisateur, parmi ces approches, nous pouvons distinguer le *profiling* [CHO et al., 2002] et les *approches par stéréotype*. [MAATALLAH, 2016].

Le *profiling* consiste à examiner, analyser et enregistrer les actions d'un utilisateur lors de sa recherche et interactions avec le système pour déterminer son profil.

Approche par stéréotypes est fondée sur l'identification des groupes d'utilisateurs et la détermination des critères de chaque groupes.

1.3.5.2. Profil item

C'est la description des caractéristiques de l'item. Par exemple, dans un système de recommandation de films, les items (films) sont représentés par leurs Id, titre, genre,...etc. tandis que dans un système de recommandation de documents, les caractéristiques sont des mots clés qui décrivent le contenu sémantique du document. Ces mots clés et leurs poids sont obtenus en général par une opération d'indexation. [MAATALLAH, 2016].

1.4. Classification des systèmes de recommandation

Les techniques de recommandation peuvent être classées de différentes manières. Parfois plusieurs termes sont utilisés pour distinguer une même méthode ou approche.

Nous basons notre étude sur une classification selon deux approches : le filtrage basée sur le contenu et le filtrage collaboratif [SHAHABI et al., 2001];

[ADOMAVICIUS et TUZHILIN, 2005].

En plus de ces deux approches, Robin Burke [BURKE, 2007] propose de considérer trois autres approches :

- La recommandation basée sur les données démographiques.
- La recommandation basée sur la connaissance (Knowledge –based)
- La recommandation basée sur l'utilité (utility-based).

Mais il note que ces trois approches sont des cas particuliers des approches classiques.

Le filtrage peut utiliser un calcul de similarité entre les documents ou entre les profils des utilisateurs.

Cette distinction est à la base des trois principales approches de filtrage utilisées :

- Le filtrage basé sur le contenu compare les nouveaux documents au profil de l'utilisateur et recommande ceux qui sont les plus proches.
- 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.
- Le filtrage hybride combine le filtrage basé sur le contenu et le filtrage collaboratif pour exploiter au mieux les avantages de chacun.

Nous présentons dans ce qui suit les différentes approches :

1.4.1. La recommandation basée sur le contenu

Les approches utilisées dans les systèmes de recommandations basées sur le contenu sont généralement inspirées des domaines de la recherche documentaire et du filtrage d'information.

Un système de recommandation basé sur le contenu [PAZZANI et BILLSUS, 2000] recommande à l'utilisateur courant des items ayant une description similaire aux items qu'il a appréciés par le passé. Ce qui revient à identifier les caractéristiques communes aux items ayant reçus une évaluation favorable de la part de l'utilisateur courant. Ceci suppose qu'une description riche et variée est disponible sur les items à recommander. Dans la plupart des systèmes, la description est donnée sous forme textuelle [PAZZANI et BILLSUS, 2007].

Un système de filtrage basé sur le contenu suit deux fonctionnalités principales :

- La sélection des items et\ou documents pertinents vis-à-vis du profil de l'utilisateur.
- La mise à jour du profil de l'utilisateur en fonction du retour de pertinence fourni par l'utilisateur sur les items et\ou documents qu'il a reçus. La mise à jour se fait par l'intégration des caractéristiques des items et\ou des thèmes abordés dans les documents jugés pertinents.

L'utilisateur ne saisit aucune requête et son profil est construit à partir de l'historique de ses évaluations, plus généralement à partir de l'analyse des usages.

Selon LOPS et *al.*, (2011) l'architecture d'un système de recommandation basé sur le contenu est découpée en trois composants :

- **Analyseur du contenu** : Réalise le prétraitement pour extraire les informations pertinentes, les structurer et les rendre sous forme exploitable par le composant.

- **L'apprentissage du profil utilisateur :** Ce composant collecte les données issues de l'analyse des usages pour construire le profil utilisateur, le profil utilisateur définit le modèle de préférence des utilisateurs pour le contenu des items à partir de leurs appréciations pour les items.
- **Le filtrage des items :** Ce module se charge de filtrer les items à recommander à l'utilisateur courant, le profil utilisateur est comparé à celui des items à recommander en utilisant des mesures de similarités, le résultat de la comparaison peut être binaire (item pertinent, item non pertinent) ou continu auquel cas une liste triée selon l'ordre de pertinence des items est établie. Une top-N liste composée de n items susceptibles d'être les plus pertinents sera recommandée à l'utilisateur courant.

1.4.1.1. Représentation d'un item

Dans la plupart des systèmes basées sur le contenu, la description de l'item est sous forme de texte, extrait de page web, email ou fiche produit dans un site de e-commerce.

Contrairement aux données structurées, il n'y a pas d'attribut avec des valeurs bien définies. Cela rend plus difficile l'apprentissage du profil item, en raison de l'ambiguïté du langage naturel et notamment de la polysémie (multiples significations pour un mot) et de la synonymie (plusieurs mots qui ont le même sens).

Un prétraitement peut alors être nécessaire afin d'extraire l'information pertinente et de la structurer sous forme d'un ensemble d'attributs.

1.4.1.2. Représentation des utilisateurs

Le profil de l'utilisateur peut être constitué de différents types d'informations qui représentent ses intérêts, dont deux types sont les plus considérés par les systèmes de recommandation [PAZZANI et BILLSUS, 2007] afin de le créer : Les préférences de l'utilisateur et l'historique des interactions de l'utilisateur avec le système [PAZZANI et BILLSUS, 2007].

L'historique des interactions est le type d'information le plus important pour les systèmes de recommandation basés sur le contenu, vu qu'il peut être utilisé pour servir des données d'apprentissage pour les algorithmes d'apprentissage automatique, ayant comme but d'apprendre une fonction pour modéliser les intérêts de chaque utilisateur.

La fonction utilisée par le modèle élaboré prédit si l'utilisateur sera intéressé par un item donné ou non, sous forme d'une estimation de la probabilité que l'item est intéressant ou non pour lui,

ou en prédisant directement une valeur numérique telle que le degré d'intérêt de l'item pour l'utilisateur.

La création d'un modèle de préférences de l'utilisateur à partir de son historique peut être vue comme une forme d'apprentissage d'une classification [PAZZANI et BILLSUS, 2007], où l'ensemble de données d'apprentissage du modèle est divisé en catégories, par exemple en catégories binaires « items que l'utilisateur a aimés » et « items que l'utilisateur n'a pas aimés », en se basant soit sur les évaluations explicites ou implicites.

Nous distinguons différents types de méthodes d'apprentissage du profil de l'utilisateur utilisées par les systèmes de recommandation basés sur le contenu :

- Les méthodes classiques inspirées des méthodes utilisées dans le domaine de la recherche d'information.
- Les méthodes adaptées pour apprendre un modèle des préférences de l'utilisateur utilisant les techniques de l'apprentissage automatique ou les modèles probabilistes.

1.4.1.3. Les types de recommandation basée sur le contenu

On distingue deux types de recommandation basé sur le contenu : recommandation basée sur « les mots clés » et recommandation basé sur « la sémantique ».

1.4.1.3.1. Recommandation basé sur les mots clés

Le Modèle d'Espace Vectoriel (ou Modèle Vectoriel MV) est une représentation algébrique des documents textuels, dont chaque document est représenté par un vecteur dans un espace à n dimensions. Chaque dimension correspond à un terme du vocabulaire d'une collection de documents donnée, et elle contient le poids associé au terme, représenté par cette dimension, dans le document donné.

Formellement, chaque document d_j est représenté comme un vecteur pondéré à long terme, où chaque poids indique le degré d'association entre le document d_j et le terme t_k [LOPS et al., 2011].

Soient $\mathbf{D} = \{d_1, d_2, \dots, d_n\}$ l'ensemble de documents de la collection et $\mathbf{T} = \{t_1, t_2, \dots, t_n\}$ l'ensemble des mots dans les documents ou le corpus. Chaque document d_j est représenté comme $d_j = \{w_{1j}, w_{2j}, \dots, w_{nj}\}$, où w_{kj} est le poids associé au terme t_k dans le document d_j , qui exprime l'importance de ce terme dans le document. Généralement, l'ensemble des termes \mathbf{T}

est obtenu en appliquant certaines techniques de traitement du langage naturel standard, Ensuite, les vecteurs de poids des termes sont calculés.

La représentation de documents en utilisant le modèle d'espace vectoriel fait apparaître deux difficultés : la **pondération** des termes et la mesure de **similarité** des vecteurs représentant les documents.

La méthode de pondération de termes la plus couramment utilisée est la pondération TF.IDF, qui est basée sur des observations empiriques sur le texte :

- Des occurrences multiples d'un terme dans un document sont souvent plus pertinentes que de simples occurrences (TF).
- Les termes rares ne sont pas forcément moins discriminants par rapport aux termes fréquents (IDF).
- Des documents longs ne sont pas préférables à des documents plus courts.

Plus explicitement, les termes qui apparaissent fréquemment dans un document, mais rarement dans le reste du corpus ont plus de chances de représenter le sujet du document [LOPS et al., 2011]. De plus, la normalisation des vecteurs résultats empêche les documents longs d'avoir plus de chances d'être retrouvés que les documents courts.

Cela est bien pris en compte par la fonction TF-IDF:

$$\mathbf{TF.IDF}(\mathbf{t}_k, \mathbf{d}_j) = \mathbf{TF}(\mathbf{t}_k, \mathbf{d}_j) * \log\left(\frac{N}{n_k}\right) \quad (1.1)$$

Où N dénote le nombre de documents dans le corpus, et n_k représente le nombre de documents de la collection dans lesquels le terme \mathbf{t}_k apparaît au moins une fois, avec :

$$\mathbf{TF}(\mathbf{t}_k, \mathbf{d}_j) = \frac{f_{k,j}}{\max_z f_{z,j}} \quad (1.2)$$

Où : $f_{k,j}$ représente le nombre d'occurrences du terme \mathbf{t}_k dans le document \mathbf{d}_j ,

$\max_z f_{z,j}$ est le maximum des fréquences $f_{z,j}$ des termes \mathbf{t}_z apparaissant dans le document \mathbf{d}_j .

Afin que tous les poids appartiennent à l'intervalle $[0,1]$, et que tous les documents soient représentés par des vecteurs de même longueur, les poids obtenus par la fonction TF.IDF sont généralement normalisés en utilisant **la normalisation cosinus** :

$$\mathbf{W}_{k,j} = \frac{\mathbf{TF.IDF}(\mathbf{t}_k, \mathbf{d}_j)}{\sqrt{\sum_{s=1}^{|\mathbf{T}|} \mathbf{TF.IDF}(\mathbf{d}_s, \mathbf{d}_j)^2}} \quad (1.3)$$

Une fois que les poids sont calculés et normalisés. Le contenu d'un item d_j , est défini par :

$$\mathbf{Content}(d_j) = (w_{1j}, w_{2j}, \dots, w_{kj}) \quad (1.4)$$

Après cette étape de pondération des termes et de normalisation, il faut définir une mesure de similarité des vecteurs caractéristiques. Cette mesure de similarité est requise pour déterminer la proximité entre deux documents. Il existe de nombreuses mesures de similarité, mais la mesure la plus largement utilisée dans la littérature est **la similarité cosinus**(définie en plus bas dans la section 1.4.2.2) :

$$\mathbf{Sim}(d_i, d_j) = \frac{\sum_k w_{ki} * w_{kj}}{\sqrt{\sum_k w_{ki}^2} * \sqrt{\sum_k w_{kj}^2}} \quad (1.5)$$

A la fin de cette étape, le système obtient une liste des termes avec leurs poids indiquant leur importance pour chaque document. Afin de calculer la similarité entre les documents (ou vecteur de documents).

L'inconvénient de cette technique est qu'elle ne considère pas la sémantique des termes. Cependant, de nouvelles techniques pour retirer les connaissances dans la phase d'indexation, en employant *la sémantique par les ontologies*² ou à partir *des sources de connaissances encyclopédiques* sont présentées dans LOPS et al., (2011).

1.4.1.3.2. Recommandation basée sur la sémantique

Un algorithme de filtrage basé sur le contenu peut réaliser une correspondance entre un descripteur de contenu et un profil utilisateur et déterminer le degré de pertinence de chaque contenu pour l'utilisateur courant. Si dans une période de temps de nombreux documents s'accumulent, cet algorithme de filtrage de contenu peut ordonner ces documents en fonction de leur pertinence pour chacun des utilisateurs potentiels.

Descripteur de contenu : Un descripteur de document se compose d'un ensemble de concept qui peut être représentés par une ontologie de domaine.

Un document peut être représenté avec de nombreux concepts de l'ontologie, chaque concept peut apparaître dans n'importe quelle branche de la hiérarchie de l'ontologie et à tout niveau cela dépend du contenu réel de ce document.

² Ontologie : est un ensemble structuré des termes et concepts représentant le sens d'un champ d'informations, que ce soit par les métadonnées d'un espace de noms, ou les éléments d'un domaine de connaissances.

Le profil peut inclure des concepts frères, c'est-à-dire les fils d'un même concept.

Le profil d'utilisateur : Un profil utilisateur basé sur le contenu se compose d'une liste pondérée de concepts de l'ontologie, représentant ses intérêts.

Le profil d'utilisateur peut comporter de nombreux concepts de l'ontologie, chacun figurant dans les différentes branches et différents niveaux de la hiérarchie.

Similarité entre un descripteur de contenu d'un document et un profil utilisateur : Un descripteur de document et un profil utilisateur peuvent avoir de nombreux concepts communs ou relatifs ; de toute évidence, plus les concepts sont communs ou relatifs, plus forte est leur similitude.

1.4.1.4. Les avantages et les inconvénients de la recommandation basée sur le contenu

Les avantages :

- Ils recommandent des éléments similaires à ceux que les utilisateurs ont aimés dans le passé.
- Ils prennent en compte le profil des utilisateurs qui est la clé pour avoir les recommandations les plus pertinentes pour chacun.
- Faire coïncider les préférences de l'utilisateur et les caractéristiques des éléments fonctionne pour de nombreux types de données (textuelles, numériques, etc.) puisqu'on utilise généralement des listes de mots-clés.
- Les données relatives aux autres utilisateurs sont inutiles.
- Il n'y a pas de problème de démarrage à froid lorsqu'un nouvel élément est ajouté au catalogue puisqu'il s'agit de faire coïncider les préférences de l'utilisateur et les caractéristiques des éléments.
- Il est possible de faire des recommandations à des utilisateurs avec des goûts « uniques ».
- Il est possible de recommander de nouveaux documents ou même des documents qui ne sont pas populaires.

Les inconvénients :

- Tous les contenus ne peuvent pas être représentés avec des mots-clés (par exemple, les images).
- Des documents représentés par le même ensemble de mots-clés ne peuvent pas être distingués.

- Les utilisateurs ayant visualisé un très grand nombre de documents posent un problème (trop d'informations dans le profil de l'utilisateur à faire coïncider avec les caractéristiques des documents).
- Lorsqu'un nouvel utilisateur commence à utiliser le système, il n'existe pas d'historique.
- Un risque de « surspécialisation » apparaît, c'est-à-dire que l'on se limite aux documents similaires et que les réponses sont trop homogènes.
- Les profils des utilisateurs restent difficiles à élaborer et, qui plus est, il faut prendre en compte l'évolution des intérêts de l'utilisateur.
- Pour que le système produise des recommandations précises, l'utilisateur doit fournir un *feedback*³ sur les suggestions retournées mais cela est chronophage pour lui.

1.4.2. Le Filtrage Collaboratif

Les systèmes de recommandations basés sur le filtrage collaboratif recommandent à l'utilisateur actif, les items que d'autres utilisateurs ayant des préférences similaires. Cette méthode de recommandation repose sur l'idée que, si les utilisateurs sont d'accord sur la qualité de certains items, ils seront probablement d'accord sur d'autres items.

1.4.2.1. Les techniques de filtrage

On distingue dans ce qui suit ces cinq techniques du filtrage collaboratif :

- **Filtrage collaboratif basé sur les utilisateurs :**

EKSTRAND et *al.*, (2011) expliquent que cette technique de recommandation se base sur le principe de trouver des utilisateurs similaires à l'utilisateur courant puis d'utiliser leurs évaluations pour prédire ce que l'utilisateur courant peut aimer. Les utilisateurs similaires à l'utilisateur courant, appelés voisins de cet utilisateur, sont ceux qui ont un comportement d'évaluation similaire à celui de l'utilisateur courant.

Les 3 étapes de cette technique de recommandation sont présentées comme suit :

1. Calculer la similarité entre l'utilisateur courant et tous les utilisateurs du système.
2. Sélectionner un sous ensemble d'utilisateurs à utiliser comme recommandateurs. Il s'agit des utilisateurs voisins les plus proches.
3. Calculer les prédictions en utilisant une combinaison pondérée des évaluations appartenant aux voisins sélectionnés.

³ Feedback : les informations de retour fournis par l'utilisateur.

L'évaluation prédite $\mathbf{R}(\mathbf{u}, \mathbf{i})$ de l'item \mathbf{i} par l'utilisateur \mathbf{u} dépend de :

- la similarité entre cet utilisateur et ses voisins les plus proches notée par $\mathbf{Sim}(\mathbf{u}, \mathbf{v})$ se calcule de deux méthodes (définie en plus bas dans la section 1.4.2.2) :

1. Coefficient de corrélation de Pearson :

$$\mathbf{Sim}(\mathbf{u}, \mathbf{v}) = \frac{\sum_{\mathbf{i} \in \mathbf{I}} (\mathbf{R}(\mathbf{u}, \mathbf{i}) - \bar{\mathbf{R}}(\mathbf{u}, \cdot)) * (\mathbf{R}(\mathbf{v}, \mathbf{i}) - \bar{\mathbf{R}}(\mathbf{v}, \cdot))}{\sqrt{\sum_{\mathbf{i} \in \mathbf{I}} (\mathbf{R}(\mathbf{u}, \mathbf{i}) - \bar{\mathbf{R}}(\mathbf{u}, \cdot))^2} * \sqrt{\sum_{\mathbf{i} \in \mathbf{I}} (\mathbf{R}(\mathbf{v}, \mathbf{i}) - \bar{\mathbf{R}}(\mathbf{v}, \cdot))^2}} \quad (1.6)$$

$\bar{\mathbf{R}}(\mathbf{u}, \cdot)$ est la moyenne des évaluations de l'utilisateur \mathbf{u} .

\mathbf{I} est l'ensemble des items Co-évalués par \mathbf{u} et \mathbf{v} .

2. Similarité basée sur le cosinus :

$$\mathbf{Sim}(\mathbf{u}, \mathbf{v}) = \frac{\sum_{\mathbf{i} \in \mathbf{I}} (\mathbf{R}(\mathbf{u}, \mathbf{i})) * (\mathbf{R}(\mathbf{v}, \mathbf{i}))}{\sqrt{\sum_{\mathbf{i} \in \mathbf{I}} (\mathbf{R}(\mathbf{u}, \mathbf{i}))^2} * \sqrt{\sum_{\mathbf{i} \in \mathbf{I}} (\mathbf{R}(\mathbf{v}, \mathbf{i}))^2}} \quad (1.7)$$

- $\mathbf{R}(\mathbf{v}, \mathbf{i})$ est l'évaluation de l'utilisateur \mathbf{v} sur l'item \mathbf{i} .
- \mathbf{k} est le facteur de normalisation donné par l'équation suivante :

$$\mathbf{k} = \frac{1}{\sum_{\mathbf{v} \in \mathbf{N}} |\mathbf{Sim}(\mathbf{u}, \mathbf{v})|} \quad (1.8)$$

\mathbf{v} appartient à l'ensemble \mathbf{N} qui représente les voisins les plus proches de l'utilisateur \mathbf{u} .

$\mathbf{R}(\mathbf{u}, \mathbf{i})$ est décrit par la formule suivante :

$$\mathbf{R}(\mathbf{u}, \mathbf{i}) = \mathbf{k} \sum \mathbf{R}(\mathbf{v}, \mathbf{i}) * \mathbf{Sim}(\mathbf{u}, \mathbf{v}) \quad (1.9)$$

- **Filtrage collaboratif basé sur les items**

Le filtrage collaboratif basé sur les utilisateurs souffre de problèmes de montée en charge si la base d'utilisateurs est importante. La technique du filtrage collaboratif basée sur les items [SARWAR et *al.*, 2001] a été développée pour répondre à cette problématique. Cette technique est utilisée lorsqu'il s'agit de trouver des items similaires à l'item courant.

Selon les modèles des évaluations des items. Si deux items ont tendance à avoir les mêmes utilisateurs qui les aiment et les mêmes utilisateurs qui ne les aiment pas, alors ces items sont similaires. Les utilisateurs ont des préférences similaires pour les items similaires. Comme défini par [GABRIELSSON et GABRIELSSON, 2006], cette technique se compose de 3 étapes :

1. Calculer la similarité entre l'item courant et tous les items du système.
2. Sélectionner les voisins les plus proches de l'item courant. Il s'agit des items les plus proches.
3. Calculer les prédictions en utilisant un algorithme basé sur l'évaluation par l'utilisateur courant des items appartenant au voisinage de l'item courant.

L'évaluation prédite $\mathbf{R}(\mathbf{u}, \mathbf{i})$ de l'item \mathbf{i} par l'utilisateur \mathbf{u} dépend de :

- La similarité entre cet item et les items évalués par l'utilisateur \mathbf{u} notée par $\mathbf{Sim}(\mathbf{i}, \mathbf{j})$:

Se calcule de deux méthodes (définie en plus bas dans la section 1.4.2.2) :

1. coefficient de corrélation de Pearson :

$$\mathbf{Sim}(\mathbf{i}, \mathbf{j}) = \frac{\sum_{\mathbf{u} \in \mathbf{U}} (\mathbf{R}(\mathbf{u}, \mathbf{i}) - \bar{\mathbf{R}}(\cdot, \mathbf{i})) * (\mathbf{R}(\mathbf{u}, \mathbf{j}) - \bar{\mathbf{R}}(\cdot, \mathbf{j}))}{\sqrt{\sum_{\mathbf{u} \in \mathbf{U}} (\mathbf{R}(\mathbf{u}, \mathbf{i}) - \bar{\mathbf{R}}(\cdot, \mathbf{i}))^2} * \sqrt{\sum_{\mathbf{u} \in \mathbf{U}} (\mathbf{R}(\mathbf{u}, \mathbf{j}) - \bar{\mathbf{R}}(\cdot, \mathbf{j}))^2}} \quad (1.10)$$

$\bar{\mathbf{R}}(\cdot, \mathbf{i})$ est la moyenne des évaluations de l'item \mathbf{i} .

\mathbf{U} est l'ensemble des utilisateurs qui ont Co-évalués les items \mathbf{i} et \mathbf{j} .

2. Similarité basée sur le cosinus :

$$\mathbf{Sim}(\mathbf{i}, \mathbf{j}) = \frac{\sum_{\mathbf{u} \in \mathbf{U}} (\mathbf{R}(\mathbf{u}, \mathbf{i})) * (\mathbf{R}(\mathbf{u}, \mathbf{j}))}{\sqrt{\sum_{\mathbf{u} \in \mathbf{U}} (\mathbf{R}(\mathbf{u}, \mathbf{i}))^2} * \sqrt{\sum_{\mathbf{u} \in \mathbf{U}} (\mathbf{R}(\mathbf{u}, \mathbf{j}))^2}} \quad (1.11)$$

- $\mathbf{R}(\mathbf{u}, \mathbf{j})$ est l'évaluation de l'utilisateur \mathbf{u} sur l'item \mathbf{j} .
- \mathbf{k} est le facteur de normalisation donné par l'équation suivante :

$$\mathbf{K} = \frac{1}{\sum_{\mathbf{j} \in \mathbf{I}} |\mathbf{Sim}(\mathbf{i}, \mathbf{j})|} \quad (1.12)$$

\mathbf{j} appartient à l'ensemble \mathbf{I} qui contient les items, voisins les plus proche de l'item \mathbf{i} , évalués par l'utilisateur \mathbf{u} .

$\mathbf{R}(\mathbf{u}, \mathbf{i})$ est décrit par la formule suivante :

$$\mathbf{R}(\mathbf{u}, \mathbf{i}) = \mathbf{K} \sum_{\mathbf{j} \in \mathbf{I}} \mathbf{R}(\mathbf{u}, \mathbf{j}) * \mathbf{Sim}(\mathbf{i}, \mathbf{j}) \quad (1.13)$$

- **Filtrage collaboratif basée sur la réduction de dimension**

Comme expliqué précédemment, les évaluations des utilisateurs peuvent être considérées comme des vecteurs. Ces derniers peuvent avoir une grande dimension. Un item

est représenté par un vecteur à U dimensions (U étant le nombre d'utilisateurs du système) et un utilisateur est représenté par un vecteur à R dimensions (R étant le nombre des items du système).

Vu que ces dimensions contiennent des redondances, cette technique de recommandation propose de réduire ces dimensions. *Singular Value Decomposition*⁴ (SVD) est l'une des techniques de réduction de dimensions.

- **Filtrage collaboratif basée sur les méthodes probabilistes**

Il existe plusieurs méthodes de recommandation probabiliste. Ces méthodes construisent des modèles probabilistes du comportement des utilisateurs et utilise ces modèles pour prédire les futurs comportements. *Cross-sell*⁵[KITTS et al., 2000] est l'un des systèmes qui utilise les méthodes probabilistes, basées sur la classification bayésienne naïve.

- **Filtrage collaboratif basée sur les méthodes basées sur les graphes**

DALIA, (2014) explique que les méthodes de la théorie des graphes sont souvent utilisées pour calculer les voisins les plus proches d'un utilisateur donné dans un système de filtrage collaboratif. Le graphe **utilisateur/item** est un **graphe biparti** dans lequel il existe deux types de sommets : les items et les utilisateurs. Si un utilisateur évalue, visite ou achète un item alors une arête est créée dans le graphe entre cet utilisateur et cet item.

Dans ce graphe, une séquence du chemin **utilisateur/item** peut être par exemple : $U1 \rightarrow I7 \rightarrow U5 \rightarrow I4 \rightarrow U2$. Dans cet exemple les utilisateurs $U1$ et $U5$ ont visité l'item $I7$ et $U5$ et $U2$ ont visité l'item $I4$.

La représentation en **graphe biparti** permet d'exploiter les relations transitives entre les utilisateurs pour effectuer la recommandation.

Les systèmes de recommandation basés sur des graphes utilisent plusieurs méthodes comme le *Plus court chemin* [HUANG et al., 2004] et *la marche aléatoire* [JAMALI et ESTER, 2009].

⁴ SVD : En mathématiques est un outil important de factorisation des matrices rectangulaires réelles ou complexes.

⁵ Cross-sell : une méthode de recommandation client élément promotion accordable rapide basée sur des probabilités conditionnellement indépendantes

1.4.2.2. Mesure de similarité

Plusieurs mesures de similarité entre utilisateurs et entre items ont été proposées dans la littérature. Selon [BELIAKOV et al., 2011] les deux types de mesures de similarité les plus populaires sont *le coefficient de corrélation de Pearson* et *la similarité basée sur le cosinus*.

- **Coefficient de corrélation de Pearson**

Ce coefficient calcule la corrélation statistique de Pearson entre deux vecteurs d'évaluation pour déterminer la similarité. S'il s'agit de calculer la similarité entre deux utilisateurs, la corrélation entre eux est mesurée à l'aide des deux lignes, appartenant aux deux utilisateurs, de la matrice d'évaluation (matrice utilisateur-item). Les colonnes des items non évaluées par les deux utilisateurs sont ignorées. Seuls les items Co-évalués sont utilisés dans ce calcul.

Ce coefficient se situe entre -1 et 1. Une similarité proche de -1 signifie une corrélation négative et inversement, une similarité proche de +1 signifie une corrélation positive. Il n'existe pas de corrélation entre les deux utilisateurs si la similarité est autour de 0.

La similarité **Sim (u,v)** entre les utilisateurs **u** et **v** est donnée par l'équation suivante :

$$\mathbf{Sim (u,v)} = \frac{\sum_{i \in I} (R(u,i) - \bar{R}(u,.)) * (R(v,i) - \bar{R}(v,.))}{\sqrt{\sum_{i \in I} (R(u,i) - \bar{R}(u,.))^2} * \sqrt{\sum_{i \in I} (R(v,i) - \bar{R}(v,.))^2}} \dots\dots 1.14$$

$\bar{R}(u, .)$ est la moyenne des évaluations de l'utilisateur **u**.

I est l'ensemble des items Co-évalués par **u** et **v**.

La similarité **Sim (i,j)** entre les items **i** et **j** est donnée par l'équation suivante :

$$\mathbf{Sim (i,j)} = \frac{\sum_{u \in U} (R(u,i) - \bar{R}(.,i)) * (R(u,j) - \bar{R}(.,j))}{\sqrt{\sum_{u \in U} (R(u,i) - \bar{R}(.,i))^2} * \sqrt{\sum_{u \in U} (R(u,j) - \bar{R}(.,j))^2}} \dots\dots\dots 1.15$$

$\bar{R}(., i)$ est la moyenne des évaluations de l'item **i**.

U est l'ensemble des utilisateurs qui ont Co-évalués les items **i** et **j**.

- **Similarité basée sur le cosinus**

Dans la matrice d'évaluation (matrice utilisateur-item), les lignes associées aux utilisateurs sont considérées comme des vecteurs d'évaluation. Ce type de mesure de similarité est calculé en utilisant l'angle cosinus entre deux vecteurs d'évaluation. Cet angle est mesuré

dans un espace à N dimensions où N est le nombre d'items Co-évalués entre les deux utilisateurs. Cette similarité se situe entre 0 et 1 où le 0 signifie aucune similarité et 1 une forte similarité. Cette similarité entre les utilisateurs est décrite par la formule :

$$\text{Sim}(\mathbf{u}, \mathbf{v}) = \frac{\sum_{i \in I} (R(\mathbf{u}, i)) * (R(\mathbf{v}, i))}{\sqrt{\sum_{i \in I} (R(\mathbf{u}, i))^2} * \sqrt{\sum_{i \in I} (R(\mathbf{v}, i))^2}} \dots \dots \dots 1.16$$

et entre les items par la formule suivante :

$$\text{Sim}(\mathbf{i}, \mathbf{j}) = \frac{\sum_{u \in U} (R(\mathbf{u}, i)) * (R(\mathbf{u}, j))}{\sqrt{\sum_{u \in U} (R(\mathbf{u}, i))^2} * \sqrt{\sum_{u \in U} (R(\mathbf{u}, j))^2}} \dots \dots \dots 1.17$$

1.4.2.3. Recommandation multicritères

La majorité des systèmes de recommandation basés sur le filtrage collaboratif prennent en compte les évaluations des items par des utilisateurs suivant un seul critère pour recommander une liste d'items à l'utilisateur. Cependant, d'autres systèmes intègrent plusieurs critères qui peuvent augmenter la pertinence des recommandations.

Par exemple, dans un système de recommandation de films prenant en compte qu'un seul critère d'évaluation, un utilisateur peut fournir une seule évaluation pour un film donné. Cependant, dans un système de recommandation de film prenant en compte deux critères d'évaluation, les utilisateurs peuvent spécifier leurs préférences suivant deux attributs (par exemple, l'histoire et les effets spéciaux). [ADOMAVICIUS et *al.*, 2011] classent les techniques de recommandation utilisées dans les systèmes de recommandation multicritères en deux catégories :

- **Les techniques basées sur les heuristiques** : les similitudes entre les utilisateurs sont calculées en agrégeant les similarités de chaque critère ou en utilisant des métriques de distance multidimensionnelles
- **Les techniques basées sur les modèles** : est basée sur un modèle prédictif, utilisant des méthodes statistiques ou d'apprentissage automatique, pour prédire l'évaluation d'un item par un utilisateur.

1.4.2.4. Avantages et inconvénients de la recommandation collaborative :

Avantage :

- Utiliser les scores d'autres utilisateurs pour évaluer l'utilité des documents ;
- Trouver des utilisateurs ou groupes d'utilisateurs dont les intérêts correspondent à l'utilisateur courant ;

- Plus il y a d'utilisateurs plus il y a de scores : meilleurs sont alors les résultats.

Inconvénients :

- Trouver des utilisateurs ou groupes d'utilisateurs similaires est difficile.
- Le système de recommandation se heurte à la faible densité⁶ de la matrice *Utilisateurs X Documents*.
- Il existe aussi le problème du démarrage à froid (*cold-start problem*) : lorsqu'un nouvel utilisateur utilise le système, ses préférences ne sont pas connues et lorsqu'un nouvel élément est ajouté au catalogue, personne ne lui a attribué de score.
- Dans les systèmes avec un grand nombre de documents et d'utilisateurs, le calcul croît linéairement ; des algorithmes appropriés sont donc nécessaires.

1.4.3. La recommandation hybride

Les systèmes de recommandation hybrides combinent les techniques de recommandation dites pures pour atteindre une meilleure performance [BURKE, 2002], en tirant profit des avantages des approches basées sur contenu et des approches basées sur le filtrage collaboratif, et limitant les problèmes qui leurs sont liés.

L'approche hybride a été conçue pour faire face aux problèmes des systèmes de recommandation classiques [BURKE, 2002], l'hybridation entre le filtrage collaboratif et le filtrage basé sur le contenu est généralement adoptée en utilisant le contenu des items, qui représente le profil des utilisateurs, pour détecter les similarités entre eux et construire les différentes communautés utilisées dans le filtrage collaboratif [NGUYEN et al., 2006]. Ou bien, pour détecter la similarité entre les items les plus préférés par les voisins de l'utilisateur actif, générés par un processus du filtrage collaboratif, et les items évalués par l'utilisateur afin de lui générer ceux qui sont intéressants pour lui [WENG et al., 2008].

1.4.3.1. Les méthodes d'hybridation

Différentes possibilités de combinaison entre les méthodes de recommandation à donner naissance à sept différentes méthodes d'hybridation proposées par Burke [BURKE, 2002] : la pondération, le mixage, la distribution, la combinaison de caractéristiques, cascade,

⁶ Densité : le rapport de la masse volumique d'un objet à la masse volumique d'un objet pris comme référence.

l'augmentation de caractéristique, et le méta-niveau. Nous définissons dans les sous sections suivantes ces différentes possibilités d'hybridation.

1.4.3.1.1. L'hybride pondéré

L'hybridation pondérée [MOBASHER et *al.*, 2003] consiste à calculer le score d'un item candidat à la recommandation (un item potentiellement pertinent pour l'utilisateur et donc recommandable) grâce à une fonction définie effectuant la somme pondérée du score de l'item dans chaque approche de recommandation présente dans le système. Une fois les scores de tous les items candidats à la recommandation calculés, ceux-ci sont classés par ordre décroissant avant d'être présentés à l'utilisateur. Ce type d'hybridation nécessite que les systèmes de recommandations utilisés effectuent une tâche de prédiction de note [ROZA LEMDANI, 2016]

1.4.3.1.2. L'hybride mixé

L'hybridation mixée [SMYTH et COTTER, 2000] consiste à présenter à l'utilisateur des recommandations issues de plusieurs approches de recommandation.

1.4.3.1.3. Hybridation par combinaison de caractéristiques

Dans cette hybridation, des caractéristiques d'une approche de recommandation sont injectées dans une autre approche de recommandation. Par exemple, lors de l'utilisation d'une approche collaborative, les items peuvent être remplacés par une caractéristique des approches basées sur le contenu, qui est la description des attributs des items. De ce fait, au lieu de traiter l'information « l'utilisateur apprécie l'item i », on traite l'information « l'utilisateur a apprécié l'attribut a ». Le calcul des similarités entre les utilisateurs est ensuite effectué en se basant sur les attributs au lieu de se baser sur les items.

1.4.3.1.4. Hybridation en cascade

L'hybridation en cascade implique l'application successive des approches de recommandation du système [BURKE, 2002]. La première approche a pour but de générer un ensemble d'items candidats à la recommandation. Chaque approche de recommandation du système est ensuite appliquée à l'ensemble d'items candidats à la recommandation. Les items de cet ensemble sont donc petit à petit filtrés et l'ensemble final est recommandé à l'utilisateur.

1.4.3.1.5. Hybridation méta-niveau

Dans ce type d'hybridation, la première approche de recommandation génère un modèle qui est ensuite utilisé par la deuxième approche de recommandation. La deuxième approche de recommandation remplace complètement son entrée (la source des données) par le modèle généré par la première approche. Cette hybridation n'est pas applicable à toutes les approches de recommandation car elle nécessite l'utilisation d'approches basées sur des modèles.

1.4.3.1.6. Hybridation par ajout de caractéristiques

Cette hybridation nécessite également une application successive des approches de recommandation disponibles. Cependant, chaque approche prend en paramètre les résultats de l'approche précédente et l'utilise comme information supplémentaire durant son exécution [MELVILLE et *al.*, 2002]. Par exemple, une première approche de recommandation est utilisée afin de générer un ensemble d'items. Ces items sont intégrés au profil de l'utilisateur afin de l'enrichir. La deuxième approche de recommandation est ensuite appliquée sur ce profil enrichi afin de générer des items à proposer à l'utilisateur.

1.4.3.1.7. Hybridation à bascule

Dans ce type d'hybridation, le système sélectionne une approche de recommandation plutôt qu'une autre en fonction d'un certain critère. Par exemple, si la confiance du système pour les résultats obtenus est insuffisante, une autre approche sera sélectionnée [PAZZANI et BILLSUS, 2000].

1.4.3.2. Exemple de système de recommandation hybride

Cet exemple illustre la combinaison des deux systèmes déjà cités, prend comme entrée :

- Le profil utilisateur et modèle de connaissance (système de recommandation basé sur le contenu).
- Les données communautaires et caractéristiques des éléments (filtrage collaboratif).

Et à partir de ces données il présente à l'utilisateur une liste de recommandation.

Le processus de recommandation hybride est illustré dans la **Figure 1.9**

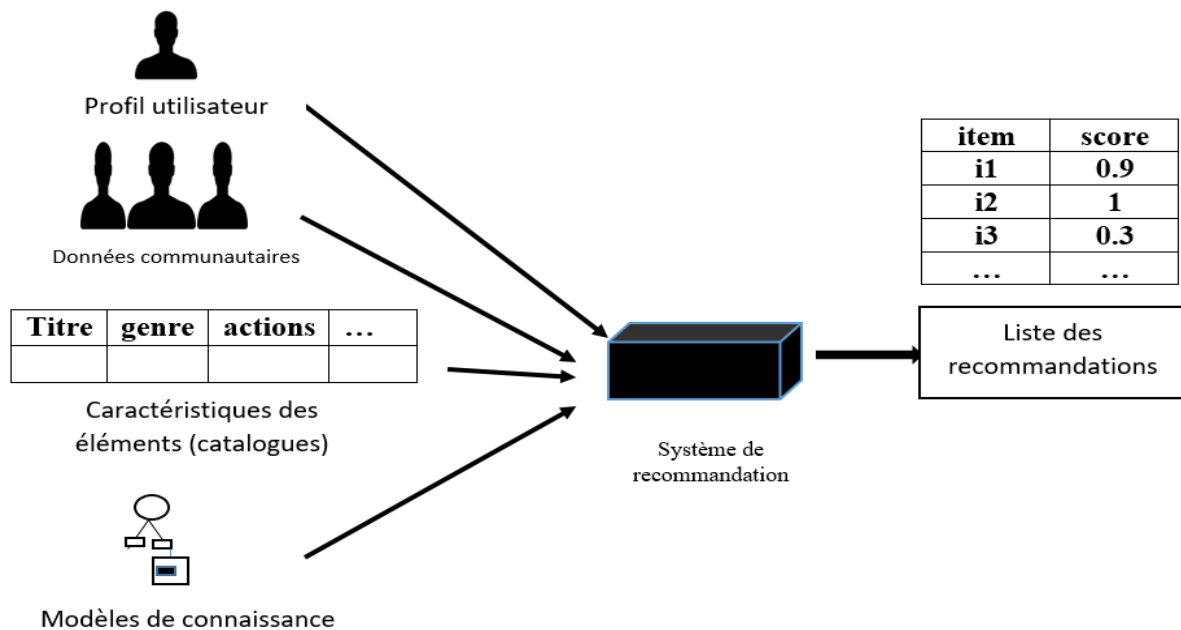


Figure 1.2 : le processus de recommandation hybride

Notons qu'une approche hybride étant une combinaison d'approches, elle présente les avantages des approches qui la composent tout en limitant leurs inconvénients [NEGRE, 2015]. Il existe trois grandes catégories de combinaisons de systèmes de recommandation pour concevoir un système de recommandation hybride [BURKE, 2002, JANNACH et *al.*, 2010] :

➤ **la combinaison monolithique** (*monolithic hybridization design*) :

« **Monolithique** » décrit une conception d'hybridation qui intègre les aspects de différentes stratégies de recommandation en un seul algorithme. Comme illustré sur la **figure 1.10**, différents systèmes de recommandation y contribuent puisque l'approche hybride utilise des données d'entrée additionnelles qui sont spécifiques à un autre algorithme de recommandation, ou bien les données d'entrée sont complétées par une technique et exploitées par une autre. Par exemple, un système de recommandation basé sur le contenu qui exploite également des données communautaires pour déterminer des similarités entre éléments relève de cette catégorie.

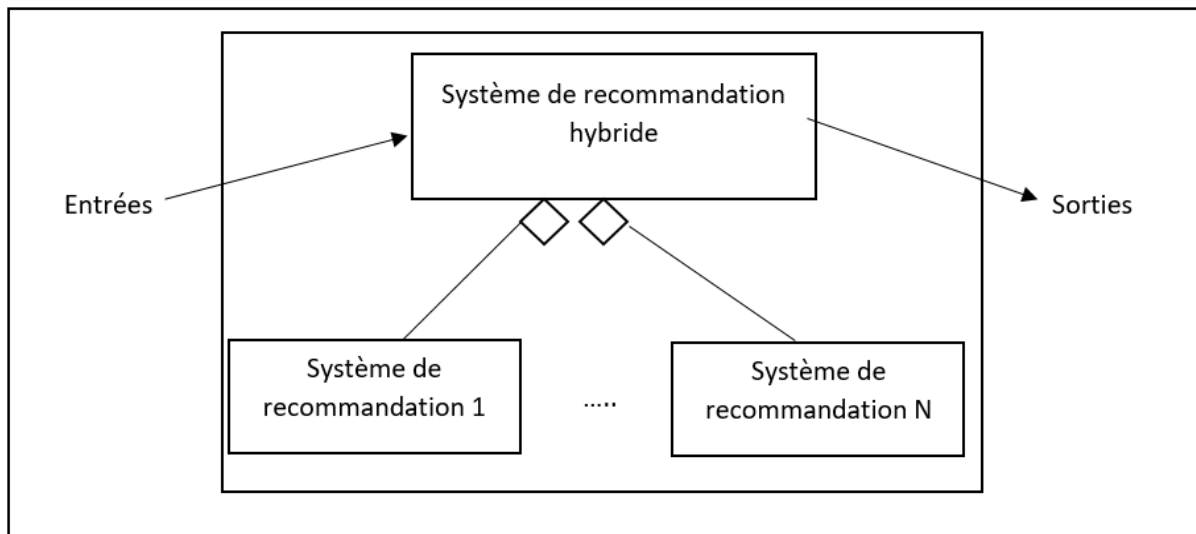


Figure 1.3 : Conception d'hybridation monolithique.

➤ **la combinaison parallèle** (*parallelized hybridization design*) :

Les deux autres approches hybrides nécessitent au moins deux mises en œuvre de recommandations séparées qui sont combinées en conséquence. Sur la base de leurs données d'entrée, les systèmes hybrides de recommandation **parallèles** fonctionnent indépendamment l'un de l'autre et produisent des listes de recommandations distinctes, comme illustré sur la **figure 1.11**.

Dans une étape ultérieure d'hybridation, leurs sorties sont combinées en un ensemble final de recommandations.

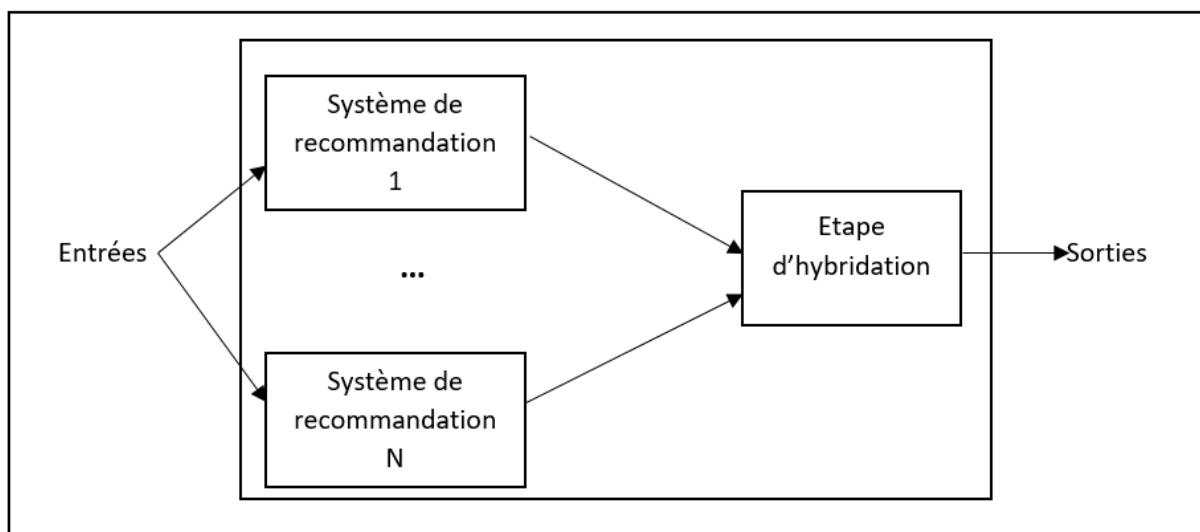


Figure 1.4 : Conception d'hybridation parallèle.

➤ **la combinaison tubulaire** (*pipelined hybridization design*) :

Lorsque plusieurs systèmes de recommandation sont joints dans une architecture **tubulaire**, comme illustré par la **figure 1.12**, la sortie de l'un des systèmes de recommandation devient une partie des données d'entrée du système suivant.

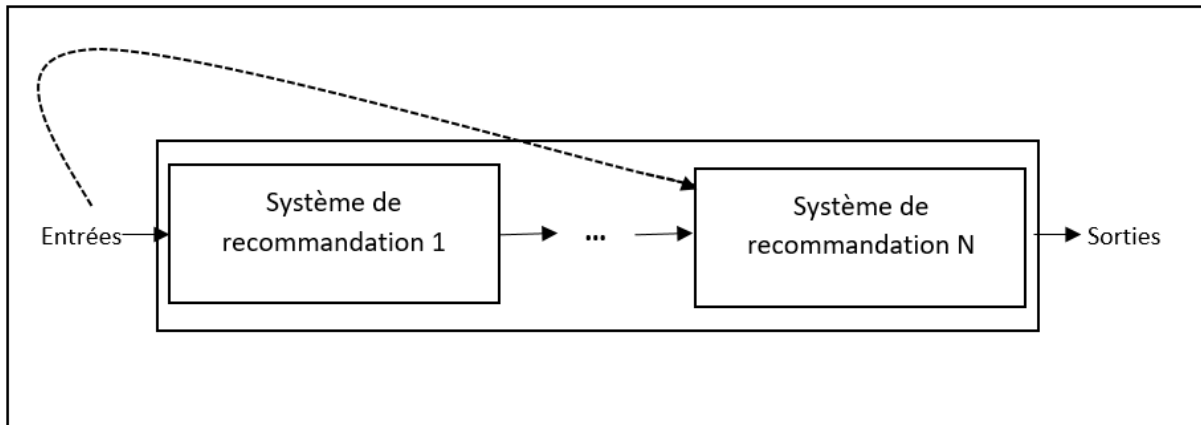


Figure 1.5 : Conception d'hybridation tubulaire.

1.5. Conclusion

Dans ce chapitre, nous avons présenté les systèmes de recommandation qui sont devenus omniprésents ces dernières années dans de nombreux domaines et détaillé les trois approches les plus utilisées : filtrage basé contenu, filtrage collaboratif ainsi que leur hybridation.

La mise en œuvre de ces systèmes nécessite l'intégration d'un profil utilisateur, ce qu'on va aborder dans le chapitre suivant.

Chapitre 2 :

La modélisation du profil utilisateur dans les systèmes de recommandation

2.1. Introduction

La modélisation du profil utilisateur dans les systèmes de recommandation consiste à la représentation, la construction ainsi de l'évolution de ce profil au cours du temps .La représentation du profil utilisateur consiste à représenter de manière interne ses centres d'intérêts. Indépendamment du modèle de représentation du profil, la construction du profil utilisateur repose sur des outils et des algorithmes permettant de collecter les sources d'information pertinentes servant à la dérivation du profil utilisateur en première phase, et des techniques de déploiement de ces sources d'information dans le but de le représenter en deuxième phase.

Nous présentons dans ce chapitre les techniques d'acquisition des sources d'information et les techniques de construction du profil selon plusieurs modèles de représentation possibles et aussi l'évolution du profil utilisateur.

2.2. La modélisation du profil utilisateurs

La modélisation du profil repose sur des techniques et des outils permettant non seulement de représenter et construire le profil de l'utilisateur mais aussi de gérer son évolution de manière dynamique au cours du temps. Les défis majeurs présents dans cette phase concernent l'identification implicite des sources d'évidence utiles à une tâche de recherche courante et l'exploitation de ces informations dans la dérivation et l'évolution du profil.

2.2.1. Représentation du profil utilisateur

Le modèle du profil utilisateur consiste à spécifier sous quelle forme les données du profil doivent être représentées.

Le modèle de base le plus communément utilisé pour la représentation des centres d'intérêt de l'utilisateur est le modèle vectoriel où chaque centre est représenté par une liste de termes représentatifs. Cependant, on distingue trois principales approches de représentation : ensembliste, sémantique et multidimensionnelle.

2.2.1.1. Représentation ensembliste

Cette représentation consiste à représenter le profil de l'utilisateur par des paquets de termes pondérés qui traduisent les centres d'intérêts de l'utilisateur. Ce type de représentation est le premier conçu pour modéliser le profil utilisateur.

Le poids associé à chaque terme permet de représenter son degré d'importance dans le profil utilisateur. Il est souvent calculé selon le schéma TF.IDF qui est utilisé dans la recherche d'information.

Nous citons trois sous-modèles de représentation ensemblistes du profil utilisateur :

1. Un ensemble de termes pondérés où chaque terme représente un centre d'intérêt possible de l'utilisateur [BUDZIK et al., 2000], [DUMAIS et al., 2003].
2. Un vecteur de termes pondérés représentant un centre d'intérêt [TEBRI et al., 2005].
3. Un ensemble de vecteurs de termes pondérés dont chacun représente un centre d'intérêt [GOWAN, 2003], [SIEG et al., 2004].

Exemple de profil utilisateur représenté par des mots clés :

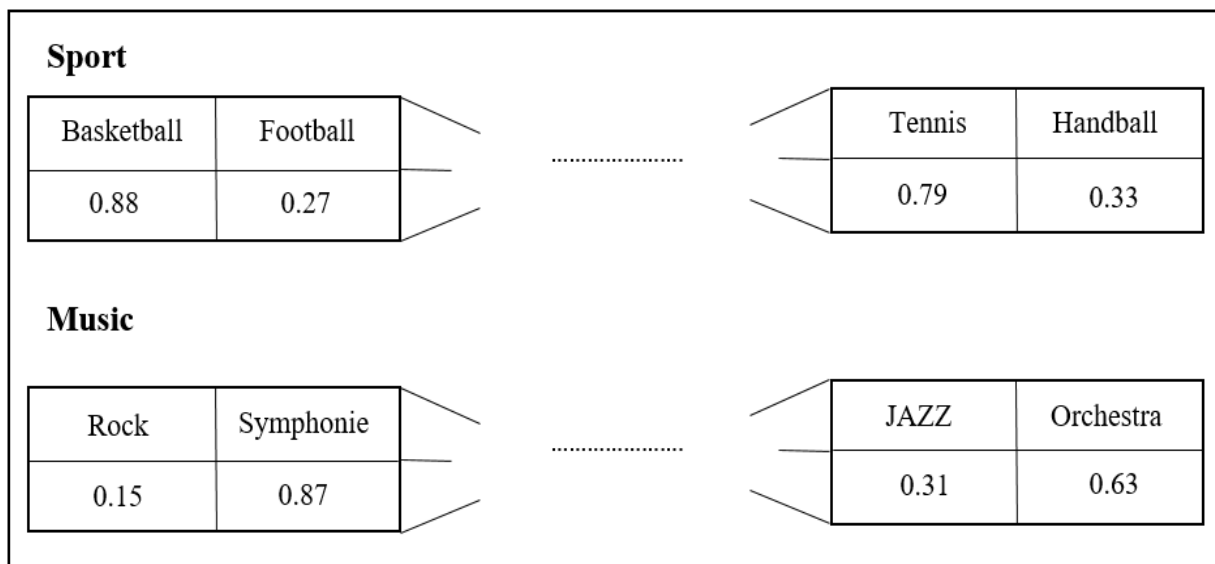


Figure 2.1 : Exemple de profil représenté par des mots clés

La Figure 2.1 donne un exemple de profil utilisateur représenté par des mots clés pondérés. Ce profil contient deux centres d'intérêts : Sport et Music.

Chaque centre d'intérêt est représenté par un ensemble de termes pondérés.

- **Music** = < (Rock, 0.15), (Symphonie, 0.87),..., (JAZZ, 0.31), (Orchestra, 0.63)> → c'est un extrait de l'ensemble de termes pondérés représentant le centre d'intérêt Music.
- **Sport** = < (Basketball, 0.88), (Football, 0.27),..., (Tennis, 0.79), (Handball, 0.33)> → c'est un extrait de l'ensemble de termes pondérés représentant le centre d'intérêt Sport.

La représentation ensembliste du profil utilisateur apporte l'avantage de la simplicité de mise en œuvre. Cependant, même si les modèles de représentation ensembliste permettent de traduire une multiplicité des centres d'intérêts de l'utilisateur, cette représentation manque de structuration, de cohérence, des niveaux de généralité et des relations de corrélation entre les divers centres d'intérêts de l'utilisateur.

2.2.1.2. Représentation sémantiques

Afin de résoudre le problème de polysémie des termes inhérents à la représentation ensembliste, une première solution consiste à représenter les centres d'intérêts de l'utilisateur par un réseau de nœuds pondérés dans lequel chaque nœud représente un concept qui traduit un centre d'intérêt utilisateur.

Ce type de représentation offre le double avantage de la structuration et de la représentation associative (relation entre les termes) permettant de considérer l'ensemble des aspects représentatifs du profil. [GENTILI et *al.*, 2003].

Les centres d'intérêts sont souvent représentés par des relations de paires de nœuds dans lesquelles chaque nœud contient un terme issu de données implicites utilisées pour construire le profil.

Les arcs reliant les nœuds sont créés sur la base de cooccurrences entre ces termes [ZEMIRLI, 2008].

La **figure 2.2** montre un exemple extrait d'un modèle d'utilisateur basé sur la représentation sémantique. [ZEMIRLI, 2008]

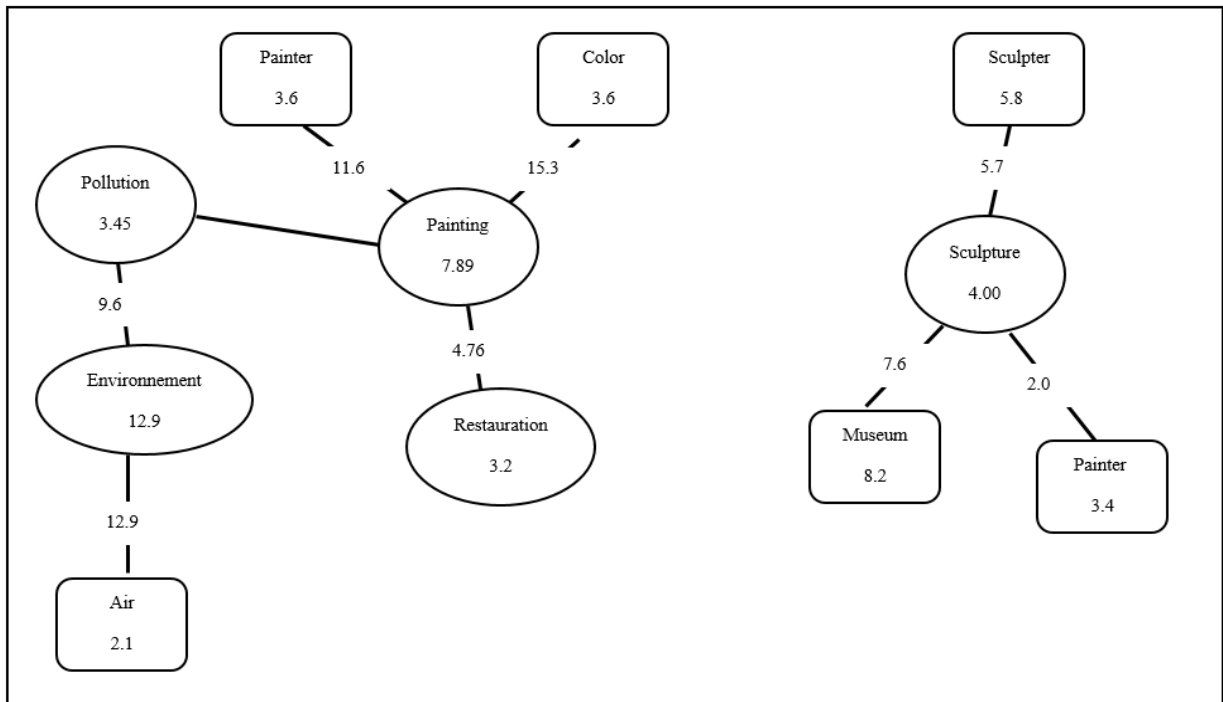


Figure 2.2 : un extrait d'un profil utilisateur sémantique

Sur cette figure (**Figure 2.2**), les cercles représentent les *planètes* (*Painting*, *Sculpture*, *Restauration*, *Environnement*, *Pollution*), et les carrés aux bords arrondis représentent les nœuds auxiliaires *satellites* liés aux planètes (pour la *Sculpture* par exemple, les *satellites* associés dans ce profil sont *Museum*, *Sculpter*, *Painter*). Les *planètes* peuvent également être reliées entre elles (cas de la relation entre *Pollution* et *Environnement* par exemple). Ces deux types de relations (entre *planètes* et entre *planètes* et *satellites*) permettent d'obtenir des profils sémantiques beaucoup plus structurés que de simples vecteurs de termes.

2.2.1.3. Représentation multidimensionnelle

Le profil est représenté par un modèle structuré de dimensions (ou catégories) prédéfinies :

- catégorie de données personnelles contient des données d'identification personnelles de l'utilisateur (nom, date de naissance, contact...)
- catégorie de données de la source contient des préférences et des restrictions sur les documents que l'utilisateur est en train de rechercher.

- catégorie de données de livraison c'est des spécifications concernant le mode de livraison des informations trouvées (courriel, fax, Web, temps de livraison etc.)
- catégorie de données de comportement contient des enregistrements sur l'interaction de l'utilisateur avec le système de recherche et les données de navigation (pages Web visités, documents lus, jugements de pertinence etc.)
- catégorie de données de sécurité est une collection des préférences de l'utilisateur concernant des conditions d'accès aux informations du profil utilisateur.

Ou bien par une structure générique [KOSTADINOV, 2008] (**Figure 2.3**) où chaque dimension est constituée d'un ensemble d'attributs éventuellement organisés en sous dimensions offrant ainsi, la possibilité d'élargir l'ensemble des données représentées en fonction de domaine d'application et des besoins utilisateur. Les attributs peuvent être simples ou composés.

Une sous dimension regroupe un ensemble d'attributs simples qui sont liés sémantiquement (par exemple l'adresse est composée de la commune, la daïra, la wilaya).

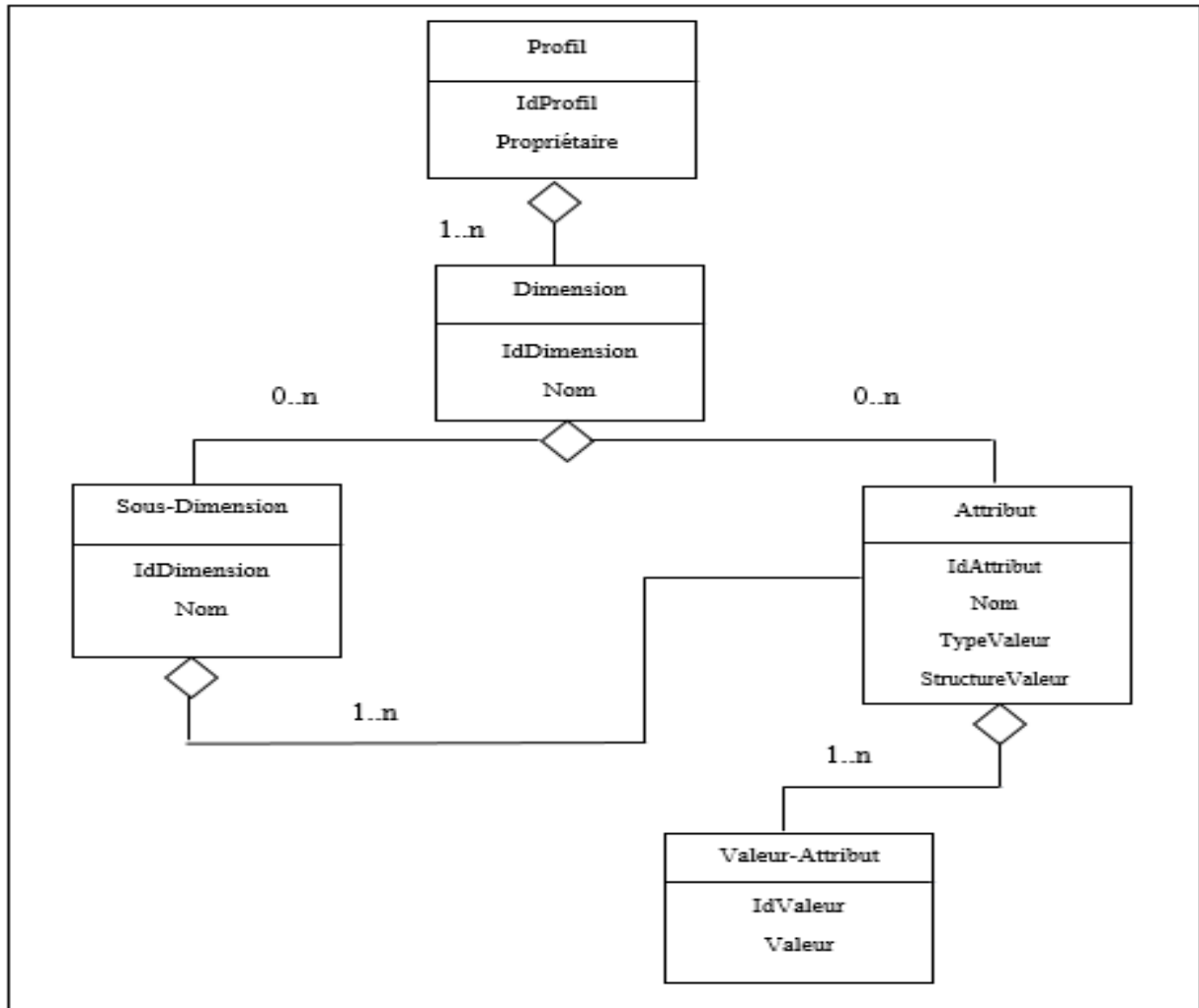


Figure 2.3 : Méta-modèle de profil utilisateur

2.2.2. Construction du profil utilisateur

Dans cette partie nous allons aborder les techniques de construction du profil utilisateur en se basant sur deux phases principales :

- La phase de collecte des données.
- La phase d'exploitation de ces données.

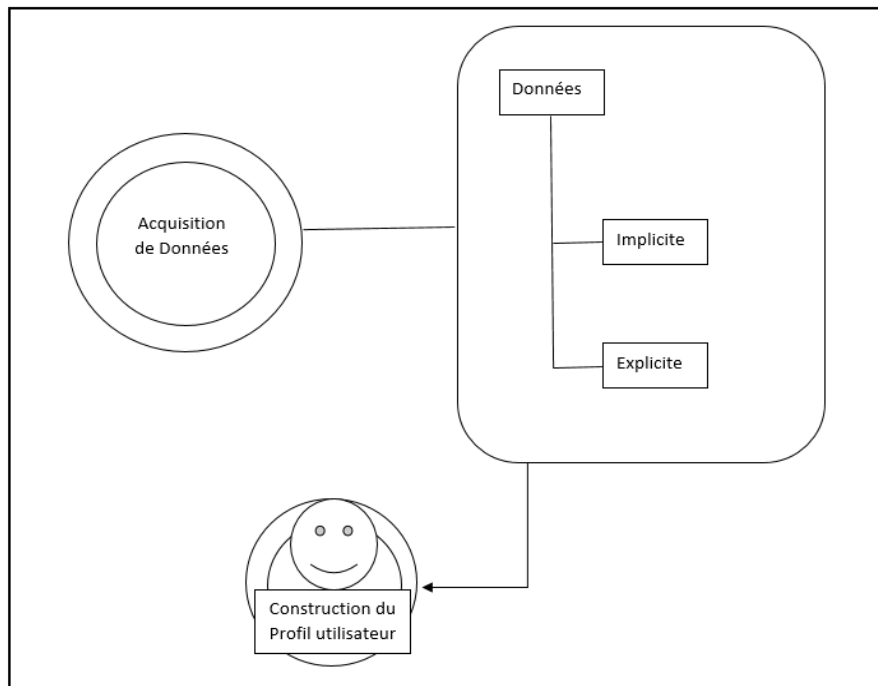


Figure 2.4. Construction du profil utilisateur

- Données : représentent la base de tout le système de recommandation, on retrouve deux type de données (implicite, explicite).
- Implicite : c'est les données collectées durant l'expérience utilisateur (navigation, clicks,...etc.).
- Explicite : c'est les données qui servent à recueillir le feedback du l'utilisateur (préférences, notes,...etc.).

2.2.2.1. Acquisition des données

L'acquisition des données se fait en suivant trois principales méthodes :

2.2.2.1.1. La méthode d'acquisition implicite

Les données implicites sont les données collectées en observant le comportement des utilisateurs ou en scrutant leurs activités (et qui sont généralement utilisées pour déterminer leurs centres d'intérêts). L'activité peut correspondre à :

- L'utilisation d'un moteur de recherche : requêtes et documents sélectionnés,

- La navigation sur le Web : pages Web consultées, liens consultés, les publications sur un réseau social numérique, statuts, commentaires, opinions,
- Les diverses applications : applications de bureau (exemple de MS Office), outils de messagerie électronique, éditeurs de textes, fichiers de logs,
- La consultation de bases de données ou de bases documentaires,...etc.

Le principal avantage de la collecte de données implicites est qu'elle ne nécessite aucune implication directe de l'utilisateur, ni de temps passé à émettre des jugements, ni un effort d'attention particulier lors de ses interactions.

- Données disponibles : sont collectées en observant le comportement des utilisateurs, ces données peuvent être des documents propres à l'utilisateur ou bien sélectionnées lors de l'utilisation d'un moteur de recherche...etc [KELLY et *al.*, 2011]
- Producteurs des données : sont produites soit par l'utilisateur lui-même ou bien par d'autres utilisateurs qui sont proches de l'utilisateur :

Données produites par l'utilisateur : L'acquisition des données à partir des données produites par l'utilisateur est le cas le plus courant dans la littérature. Toutefois, ce type d'acquisition peut poser des problèmes lorsque le système dispose de très peu de données générées par l'utilisateur. C'est souvent le cas des nouveaux utilisateurs ou des utilisateurs peu actifs comme évoqué dans [ZHANG et *al.*, 2010].

Données produites par les individus similaires à l'utilisateur : Les informations à acquérir sont extraites des informations produites par les utilisateurs similaires à L'utilisateur courant. On calcule généralement La similarité entre le profil de l'utilisateur avec ceux de tous les autres utilisateurs du système par la similarité du cosinus [GAO et *al.*, 2010]. Ensuite les informations des individus similaires sont utilisées pour calculer les informations ou les intérêts de l'utilisateur. Ce principe est à la base des différentes techniques de filtrage collaboratif.

Inconvénients des données produites par les individus similaires à l'utilisateur :

- Cette technique nécessite beaucoup de temps de calcul de similarités entre les utilisateurs pour des systèmes qui possèdent un nombre très élevé d'utilisateurs.

- Elle ne peut pas être exploitée efficacement pour les utilisateurs ayant un profil vide ou presque vide (par exemple : nouvel utilisateur dans le système), car il devient impossible de retrouver des utilisateurs similaires à ce dernier.

Données produites par les individus dans le réseau social de l'utilisateur : Un réseau social est un graphe de relations entre individus. Les liens entre les individus dans ce réseau représentent les relations entre eux et donc une certaine similarité. Cette approche se base sur les théories qui montrent qu'un utilisateur crée des relations avec ceux qui lui sont similaires car il partage avec eux des intérêts communs [ARAL et *al.*, 2013], [CARMEL et *al.*, 2009]. Cette approche consiste à construire le nombre d'individus similaires à l'utilisateur en évitant d'explorer tous les utilisateurs du système. Comme elle permet de palier au cas de manque d'informations sur l'utilisateur (profil vide ou presque vide).

Avantages de l'acquisition implicite des données :

- Ne nécessite aucune action explicite de la part de l'utilisateur.
- On peut faire face aux problèmes d'informations légèrement faussé ou de manque d'informations.

Inconvénients de l'acquisition implicite des données :

- Les données acquises sans vérification de la part de l'utilisateur ne peut pas être pertinentes pour lui (exemple : les données que produit par erreur ou qui contiennent des informations obsolètes).
- On peut manquer des informations importantes (exemple : extraire des connaissances sur l'utilisateur).
- Les données ne seront pas suffisantes pour extraire les préférences ou les intérêts de cet utilisateur, lorsque l'utilisateur n'interagit pas souvent avec le système.

2.2.2.1.2. La méthode d'acquisition explicite

Les données explicites sont les données que l'utilisateur doit fournir au système. Donc cette méthode se base sur les informations fournit par l'utilisateur. Différentes techniques

sont utilisées pour représenter les utilisateurs et structurer la connaissance à leur propos. Nous décrivons brièvement les approches les plus utilisées [VILLANOVA, 2002].

➤ **Le modèle individuel :**

Ce modèle regroupe les caractéristiques propres à un individu. Il s'agit d'informations pouvant être soit renseignées par l'utilisateur, soit déduites par le système, soit acquises lors de l'utilisation de l'application.

➤ **Le modèle de recouvrement (overlay model) :**

Il ne peut exister que si le système possède un modèle de domaine, car le modèle de recouvrement associe pour un utilisateur particulier, une valeur à chaque concept du modèle de domaine.

Chaque valeur correspond à une estimation du niveau de connaissance de l'utilisateur pour le concept. On peut donner des précisions sur l'état des connaissances : binaires (**su, pas su**), qualitatives (**bon, moyen, pauvre**) ou quantitatives.

Le modèle de recouvrement est facile à mettre à jour mais souvent difficile à initialiser surtout au niveau de la mesure du niveau de Connaissance .Les actions de l'utilisateur sont analysées pour augmenter ou réduire le niveau de connaissance des concepts du domaine [BRUSILOVSKY et EKLUND, 1998].

➤ **Le stéréotype :**

Le modèle de stéréotype correspond à un condensé des caractéristiques les plus représentatives d'un groupe ou classe d'individus, que nous pouvons qualifier de valeurs par défaut [RICH, 1989]. Des combinaisons de couples (items, valeurs) définissent différents stéréotypes tels que novice, avancé, expert. L'utilisateur est donc associé à une des catégories élaborées et hérite de ses propriétés.

➤ **Le modèle partagé :**

Il s'agit d'un modèle réutilisable dans plusieurs applications. En effet, de nombreuses caractéristiques d'un utilisateur sont utilisées systématiquement par toutes les applications adaptatives. L'idée du modèle partagé est de disposer d'une partie commune et de parties spécifiques à chacune des applications ou tâches à réaliser.

Cette approche permet d'une part de partager des informations entre applications et d'autre part d'obtenir plus facilement une participation des utilisateurs qui n'ont pas à redéfinir un modèle pour chacune des applications.

Avantages de l'acquisition des données explicite :

- Les données acquises sont pertinente pour l'utilisateur.
- Les informations sont complètes et correctes.

Inconvénients de l'acquisition des données explicite :

- Nécessite des actions explicites de la part de l'utilisateur.

2.2.2.1.3. La méthode d'acquisition hybride

Cette méthode consiste à combiner les deux systèmes précédents (acquisition implicite et acquisition explicite), elle est développée pour donner une meilleure performance. Un profil utilisateur contient des termes pondérés, chaque fois qu'un document est jugé pertinent, le poids d'un terme dans son profil est mis à jour en utilisant les paramètres suivants : le vote explicite, le temps utilisé pour lire ce document, le nombre de liens suivis et l'action sauvegarder dans les signets de ce document. [YOUNG WOO SEO et *al.*, 2000]

2.2.2.2. Prétraitement des données

Les méthodes d'acquisition des données expliquées précédemment peuvent retourner des données incomplètes, des incohérences...etc. comme ces données peuvent ne pas être conformes au format d'entrée de l'algorithme de construction de profil utilisateur. C'est pour remédier à ces problèmes que l'étape de prétraitement des données à était rajouter avant la phase finale de construction du profil utilisateur.

On utilise plusieurs types de prétraitement selon les données [GRACIA et *al.*, 2015] et [LIU, 2007].

Exemple :

Problème	Solution
les données incomplètes, biaisées ou incohérentes.	appliquer des techniques de nettoyage de données, qui consistent à ignorer les données manquantes ou à utiliser la valeur moyenne d'un attribut en remplacement ou encore à utiliser la valeur la plus probable (formule bayésienne ou arbre de décision) en remplacement, etc.
Les attributs continus.	La discrétisation des données peut être appliquée pour convertir des attributs continus vers des attributs ordinaux.
Grand volume de jeu de données.	La réduction des données peut être appliquée pour obtenir une représentation réduite du jeu de données, plus petite en volume, mais qui produit (ou presque) les mêmes résultats analytiques.
Données non conforme au format d'entrée de l'algorithme.	appliquer des techniques de transformation de données qui permettent par exemple, de ne conserver qu'un résumé d'un texte à partir d'un texte entier et de traduire un texte dans une autre langue ...

Tableau 2.1. exemple montrant les problèmes rencontrés et les solutions proposées.

2.2.3. Adaptation et mise à jour du profil utilisateur

Avec évolution de la quantité des données web, le processus d'adaptation devient de plus en plus compliqué ce qui nécessite une représentation fidèle du profil reflétant ses besoins (intérêts, préférences, etc.). L'utilisateur s'intéresse à des sujets différents dans une

courte période de temps. La mise à jour de son profil est donc une étape très importante. Pour cela plusieurs approches on était définies :

- **Adaptation manuelle** : C'est l'utilisateur qui doit mettre à jour son propre profil, mais cette méthode n'est pas adaptable si l'utilisateur change de gout souvent.
- **Adaptation par l'ajout d'information** : La mise à jour du profil utilisateur est assurée par les retours d'expérience explicites, ou implicites recueillis par le système par analyse des usages qui consiste à analyser les données disponibles dans les fichiers log afin de modéliser les préférences de l'utilisateur. L'analyse des usages doit, par conséquent, aboutir à inférer les évaluations apportées par l'utilisateur sur les items qu'il a consultés.

L'évaluation que porte l'utilisateur pour l'item peut être une note qu'il attribut implicitement ou explicitement comme elle peut être considéré comme étant un retour d'expérience sur les recommandations effectuées par le système en mesurant le taux de recommandations pertinentes.

Le retour d'expérience se présente soit sous forme de vote explicite ou implicite ou sous forme d'annotations.

Les systèmes tels que Amazon [LINDEN et *al.*, 2003], MovieLens [MILLER et *al.*, 2003] utilisent les retours d'expérience comme moyen d'adaptation de leurs systèmes de recommandation.

Le problème posé par cette méthode est qu'elle ne permet pas d'éliminer les anciennes préférences.

2.3. Conclusion

La modélisation de l'utilisateur dans un système de recommandation a pour but la construction de son profil qui contient des acquisitions pouvant être utile pour le comportement du système. Ce profil peut être relié à une seule ou groupe de personnes ayant des points communs .Il peut englober toutes les variations qui caractérisent un groupe d'utilisateur et peut être représenté de différentes manières : ensembliste, réseaux sémantique, multidimensionnelle.

Un problème de démarrage à froid se produit lors de la création du profil d'un nouvel utilisateur. Ce que nous allons expliquer dans le chapitre suivant.

Chapitre 3 :
Le démarrage à froid dans les systèmes de
recommandation

3.1. Introduction

Le profil utilisateur est un élément très important dans un système de recommandation. Il permet de recommander aux utilisateurs les informations pertinentes selon leurs préférences. Mais lorsqu'il y'a un manque d'information, le système ne pourra pas attribuer une grande pertinence aux résultats lors de la recommandation des informations, c'est ce qu'on appelle << **le démarrage à froid** >>.

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

Ce chapitre approfondit la problématique du démarrage à froid. Il est organisé de la manière suivante ; nous commençons par la définition de ce problème. Ensuite nous abordons les différentes approches développées pour atténuer les effets du démarrage à froid.

3.2. Problématique du Démarrage à froid

Le problème du démarrage à froid est très fréquent dans les systèmes de recommandation. Ce problème est double, il affecte à la fois les utilisateurs mais aussi les items. Il désigne le manque d'information sur un utilisateur ou sur un item qui vient d'être ajouté [BENOUARET, 2015]. Lorsqu'un nouvel utilisateur entre dans le système, on ne sait pas forcément quelles sont ses préférences et cela rend difficile la tâche de recommandation.

Le démarrage à froid affecte beaucoup plus les utilisateurs que les items.[BENOUARET, 2015].

Lorsque le système ne peut recommander à un utilisateur que des items qui sont en relation avec son profil, l'utilisateur est limité aux recommandations de ressources qui sont similaires à celles qu'il a déjà aimées. En revanche, la diversité des recommandations est souvent une caractéristique souhaitable pour les systèmes de recommandation [BENOUARET, 2015].

Les systèmes de recommandation de type filtrage basé contenu, prennent en compte les items préalablement choisis ou notés par l'utilisateur ainsi que le profil utilisateur (« User Profile ») [BOBADILLA et al.,2013]. Chaque item est décrit conformément à une liste d'attributs (taille, couleur, etc.) pondérés. L'utilisateur est également décrit par cette liste d'attributs. Cette description de l'utilisateur est appelée profil utilisateur. En plaçant en

confrontation dans l'algorithme de filtrage le profil utilisateur et les attributs de la liste des items, une recommandation est proposée à l'utilisateur [LOPS et al., 2011].

Les approches basées sur le contenu ont pour particularité d'utiliser un analyseur de contenu (« Content Analyzer ») sur l'information utilisée avant de la stocker en base de données. L'analyseur de contenu peut implémenter un modèle d'espace vectoriel basé sur les mots-clefs (« Keyword-based Vector Space Model » VSM) et le TF-IDF (« Term Frequency-Inverse Document Frequency ») ou bien différents algorithmes d'analyses sémantiques.

Une autre particularité des approches basées sur le contenu est l'utilisation d'un module permettant l'apprentissage du profil utilisateur (« *Profile Learner* »). Ce module peut implémenter des méthodes probabilistes comme les méthodes naïves Bayésienne ou analyser la pertinence des retours utilisateurs (« *Feedback relevance* ») et utiliser l'algorithme de Rocchio, ou bien implémenter des arbres de décision, ou encore des classificateurs de règles de décision et l'un des algorithmes du plus proche voisin [LOPS et al., 2011]. Certaines de ces méthodes font partie des techniques d'intelligence informatique [BOUZEKRI, 2017].

Pour les systèmes de recommandation de type filtrage collaboratif, le nombre de notes déjà obtenues est généralement très faible par rapport au nombre de notes qui doivent être prédites. Un item qui a alors reçu peu d'avis de la part des utilisateurs a moins de chances d'être recommandé par rapport aux autres. Aussi, pour un utilisateur qui a noté des items qui n'ont pas reçu beaucoup d'avis, il est difficile de lui trouver des utilisateurs similaires, du coup il sera difficile de faire des recommandations pertinentes pour cet utilisateur.

On distingue trois types de problèmes de démarrage à froid : [BURKE, 2002]

- Démarrage à froid pour un nouveau système.
- Démarrage à froid pour une nouvelle ressource (item).
- Démarrage à froid pour un nouvel utilisateur.

3.2.1. Le démarrage à froid pour un nouveau système

Comme son nom l'indique ce type concerne les nouveaux systèmes qui souffrent d'une absence en informations (utilisateurs, items ainsi que les relations entre eux), ce qui conduit certainement à de très mauvaises performances.

Ce type de problème a été nommé dans [BURKE, 2002] par le problème du démarrage et qui a été défini par la nécessité d'accumuler suffisamment de données pour être utilisées comme stéréotypes au démarrage.

La solution traditionnelle de ce problème est de spécifier une phase, après la phase du développement du système, à la collecte de données qui seront utilisées comme données d'apprentissage, et d'utiliser les outils de classification automatique afin de trouver les corrélations entre les entités du système.

3.2.2. Le démarrage à froid pour une nouvelle ressource

Recommander de nouvelles ressources constitue également un enjeu de taille pour les systèmes de recommandation. Ce problème (connu sous le nom de « latence ») peut être défini par l'insuffisance de données sur la nouvelle ressource, c'est à dire cette ressource n'a pas eu suffisamment d'évaluations. Donc, elle ne pourra pas être facilement recommandée.

Les systèmes les plus affecté par ce problème sont les systèmes de recommandation qui recommandent des items (articles) d'actualité [SOLLENBORN et FUNK, 2002].

Le problème de latence a été pallié par la proposition des nouvelles ressources à l'utilisateur actif pour y attribuer des appréciations. Cependant, cette stratégie pourrait occasionner une insatisfaction chez l'utilisateur qui risque d'abandonner le système.

L'approche de filtrage basée sur le contenu peut se présenter comme une solution alternative [KRULWICH et BURKEY, 1996].

La technique basée sur le contenu compare le contenu d'une nouvelle ressource avec les ressources qui ont été évaluées par les utilisateurs afin de l'impliquer dans la recommandation. Néanmoins, les recommandations générées par cette approche sont de faible qualité à cause du manque de diversité (le problème de sur spécialisation).

Un nouveau type de filtrage basé sur le contenu exploitant les ontologies a été proposé également comme solution au problème de latence. Ce type de filtrage a été notamment adopté par le système *Entree* (qui recommande des restaurants) [BURKE, 2002] et le système *Quickstep-Foxtrot* [MIDDLETON et al., 2004].

Ces deux systèmes sont spécialisés dans la recommandation des articles scientifiques. Dans ces systèmes, les ontologies sont utilisées pour classifier et de catégoriser les ressources et générer les modèles utilisateurs. La nécessité d'une construction préalable d'une ontologie relative au domaine de connaissance est l'inconvénient majeur de cette technique [ESSLIMANI, 2010].

Dans les systèmes de filtrage collaboratif, L'approche KNN est utilisée pour retrouver les k plus proche voisins d'un utilisateur actif dans le but de lui recommander les ressources préférées par ces voisins. La détermination d'un seuil de similarité ou la détermination d'un seuil d'items

co-notés est nécessaire pour approche KNN. Or, la détermination de ce type de seuil reste problématique. En effet, l'ensemble des k voisins les plus proches varie avec les changements de l'état du système (intégration de nouveaux utilisateurs et de ressources).

De ce fait, ces seuils doivent être adaptés au fur et à mesure de la réinitialisation du système, tout en évitant de fixer des valeurs extrêmes pour que le pouvoir prédictif du système ne soit pas faible et pour que le bruit ne soit pas engendré à cause de voisins peu pertinents.

En l'absence de ces ressources co-notées, le système ne peut pas sélectionner un voisinage fiable parce que cette approche repose sur les ressources notées en commun.

Dans cette optique, les associations transitives ont été utilisées par plusieurs travaux pour relier les utilisateurs aux ressources ou entre eux. Dans le but d'améliorer la qualité des recommandations [PAPAGELIS et al., 2005] exploitent le principe d'inférence pour définir des voisins potentiellement fiables. Néanmoins, considérant que les systèmes de recommandation sont dynamiques et que la phase de calcul du voisinage requiert un temps de calcul important, l'application de ce type d'association devrait se baser sur des stratégies permettant de limiter par exemple le nombre d'utilisateurs concernés, afin de permettre le passage à l'échelle [ESSLIMANI, 2010].

3.2.3. Le démarrage à froid pour un nouvel utilisateur

Un nouvel utilisateur qui arrive dans le système éprouve des difficultés à exprimer son profil en spécifiant des thèmes qui l'intéressent.

Lorsque les utilisateurs s'inscrivent, ils n'ont pas encore émis de vote, par conséquent, ils ne peuvent recevoir aucune recommandation personnalisée (qui correspond aux préférences de l'utilisateur) [RASHID, KARYPIS et RIEDL, 2008].

Le problème du nouvel utilisateur, fait partie des grandes difficultés rencontrées par les systèmes de recommandation en action.

Afin de surmonter ce problème, le système peut décider de ne pas émettre de recommandations à un utilisateur dont le système n'a pas atteint un certain nombre d'informations requises, car il n'est pas capable de cibler ses préférences avec le peu d'information disponible.

Dans le cas d'un nouveau utilisateur, le système ne pourra pas recommander efficacement pour lui parce qu'il connaît peu sur ses préférences parce qu'il n'a pas fait suffisamment d'évaluations.

Dans le cadre du filtrage collaboratif, le système ne peut pas définir ses voisins, comme le système ne peut pas extraire les attributs des items qui lui intéressent dans le cas du filtrage basé sur le contenu. Pour remédier à ce problème, la sollicitation est l'une des solutions qui ont été proposées (à travers la sollicitation de notes explicites, de critiques ou d'informations démographiques). Or, cette sollicitation directe peut entraîner l'abandon de l'utilisateur.

La proposition des recommandations arbitraires à l'utilisateur au démarrage, est un autre moyen pour pallier à ce problème. Cependant, cette stratégie risque d'occasionner une insatisfaction chez l'utilisateur au vu de la faible qualité des recommandations.

AL MAMUNNUR RASHID suggèrent d'autres techniques pour faire face au problème de nouveauté de l'utilisateur. Ces techniques utilisent la popularité des items et l'entropie consistant à évaluer la dispersion des avis des utilisateurs sur un item. L'utilisation des informations démographiques a été proposée comme une solution. Ainsi, deux utilisateurs sont considérés comme similaires, s'ils appartiennent à un même segment démographique, [PAZZANI, 1999]. Or, même si des utilisateurs appartiennent à un même segment démographique, ils ne partagent pas nécessairement les mêmes goûts [ESSLIMANI, 2010].

3.3. Les solutions proposées

Les solutions proposées pour le problème de démarrage à froid varient selon le type de filtrage.

3.3.1. Filtrage collaboratif

Les systèmes de filtrage collaboratif ont pour principe d'exploiter les « évaluations » que des utilisateurs ont faites de certains documents (items), afin de recommander ces mêmes documents (items) à d'autres utilisateurs.

Il se base sur les communautés formées à partir des évaluations d'utilisateurs [BRESESE *et al.* 1998].

Deux fonctionnalités centrales ressortent de ce type de systèmes :

- calculer la similarité entre utilisateurs et inférer les communautés.
- prédire des notes pour quelques documents et ne sélectionner que les documents avec un score élevé.

La technique classique consiste à demander à l'utilisateur d'évaluer un ensemble de documents que le système lui présente. Ces documents n'ayant a priori aucun rapport privilégié avec les centres d'intérêt potentiels de l'utilisateur. Cette tâche est simple mais fastidieuse.

Exemple :

Le système de recommandation de films MovieLens exige de l'utilisateur au moins 15 évaluations avant de fournir des recommandations, mais les films proposés n'étant pas ciblés pour cet utilisateur. Ce dernier peut ne rencontrer que des films qu'il n'aime pas, ou des films qu'il ne sait pas évaluer, comme par exemple des films dont il n'a jamais entendu parler. Ainsi l'utilisateur devra parcourir une liste parfois très longue avant d'atteindre ce nombre de 15 évaluations.

3.3.2. Filtrage basé sur le contenu

Les systèmes de filtrage à base de contenu recommandent des documents (items) similaires à ceux que l'utilisateur a déjà appréciés.

Ceci est calculé en rapprochant les centres d'intérêt des utilisateurs (introduits de manière explicite à travers un questionnaire par exemple ou de manière implicite, exemple à travers la surveillance de son comportement) avec la métadonnée ou les caractéristiques des documents, sans prendre en compte les avis des autres utilisateurs.

Deux fonctionnalités centrales ressortent de ce type de systèmes :

- La sélection des documents pertinents vis-à-vis du profil de l'utilisateur
- La mise à jour du profil en fonction du retour de pertinence fourni par l'utilisateur sur les documents (items) qu'il a reçus.

Plusieurs solutions ont été proposées basées sur de nouvelles approches pour la représentation du contenu.

On trouve des solutions comme celle de demander à l'utilisateur d'identifier ses centres d'intérêts à partir d'une liste de termes ou d'items qui correspondent le mieux à ses intérêts et préférences [MELVILLE et al., 2002], [CLAYPOOL, 99]. Cependant, cette étape est fatigante et ennuyante pour l'utilisateur du fait qu'il doit choisir à partir d'une grande liste de termes, ceux qui synthétisent leurs centres d'intérêts, ou bien il doit rechercher les items

pertinents pour lui [NGUYEN et al., 2006a]. Afin de réduire la tâche des utilisateurs, les données externes sur les utilisateurs peuvent être utilisées pour aider à trouver les items pertinents, automatiquement par le système [NGUYEN et al., 2006a], par exemple si le système dispose d'une ontologie du domaine, le profil de l'utilisateur peut être inféré automatiquement pour identifier ses centres d'intérêts [MIDDLETON *et al.*, 2002], [MIDDLETON *et al.*, 2004]. En général, le profil résultant avec ces méthodes est incomplet et bruité [NGUYEN *et al.*, 2006].

On trouve aussi : les ontologies [Middleton, 2002], les folksonomies [DURAO, 2010], et les taxonomies [WENG, 2008].

Les ontologies sont difficiles à construire et exploiter car elles sont liées à d'autres technologies.

Les folksonomies sont un ensemble de mots clés (tags) proposés arbitrairement par des utilisateurs pour décrire le contenu d'un certain nombre de documents. En plus, les folksonomies ont une structure non hiérarchique ce qui implique que ce type d'information n'est pas appropriées pour représenter le contenu pédagogique.

Les taxonomies offrent une représentation hiérarchique et simple de n'importe quel type de contenu, ceci rend ce type d'information plus populaire.

Weng [WENG, 2008] a proposé une approche puissante pour résoudre le problème de démarrage à froid relatif à un nouvel utilisateur et/ou un nouvel item en se basant sur une approche hybride qui utilise le concept des taxonomies (CSHTR : Cold Start Hybrid Taxonomic Recommender). Ils ont parti de ses principes de fonctionnement globaux pour proposer leurs approche : NPR_eL. CSHTR propose aux utilisateurs d'évaluer un ensemble de documents (principe adopté par le filtrage collaboratif et basé sur le contenu pour construire les profils des utilisateurs) et transformer ces évaluations en préférences taxonomiques (profil). Dans leurs travail, ils ont proposé aux apprenants de choisir directement leurs domaines d'intérêt en répondant à un questionnaire (une taxonomie représentant un ensemble de domaines d'apprentissage) pour éviter la perte du temps dans la lecture des documents qui peuvent être loin de leurs préférences.

3.3.3. Filtrage hybride

Les méthodes hybrides cherchent à atténuer les insuffisances de chacune des deux précédentes approches en les combinant de différentes manières.

L'idée principale de l'approche par hybridation est d'associer les nouveaux utilisateurs aux profils types «stéréotypes» prédéfinis correspondants [NGUYEN *et al.*, 2006], en utilisant leurs informations démographiques ou une autre source d'informations externes telles que les pages Web personnelles [PAZZANI, 1999], ou bien en exploitant les informations recueillies à partir des réponses fournies par l'utilisateur à une série de questions [KRULWICH, 1997].

Ces informations seront par la suite comparées par le processus du démarrage à froid avec les stéréotypes existants afin de trouver le plus approprié.

Exemple :

- [CLAYPOOL *et al.*, 1999] ont combiné les recommandations produites par les deux méthodes à base de contenu et collaboratives appliquées séparément.
- [PAZZANI, 1999] a appliqué les algorithmes de filtrage collaboratifs sur une matrice décrivant les préférences des utilisateurs sous forme de mots-clés pondérés au lieu de la traditionnelle matrice des votes.
- [MELLVILLE *et al.*, 2002] ont utilisé les recommandations à base de contenu pour compléter la matrice des votes et ensuite appliquer l'algorithme de recommandation collaboratif sur cette matrice.

Exemples de quelques combinaisons :

3.3.3.1. Combinaison avec le filtrage basé sur le contenu

Pour la méthode de combinaison du filtrage collaboratif avec le filtrage basé sur le contenu dans un système hybride [BURKE, 2002], la réponse la plus courante consiste à demander à l'utilisateur de définir ses centres d'intérêt, en termes de contenu, à partir d'une liste de termes et/ou d'exemples décrivant au mieux ses centres d'intérêt [CLAYPOOL *et al.*, 1999], [MELVILLE, 2002], [MOONEY *et al.*, 2000].

Melville et ses collègues à l'université de Texas ont proposé une méthode hybride où les valeurs manquantes dans la matrice des évaluations sont d'abord remplacées par les prédictions

par le filtrage basé sur le contenu [MELVILLE, 2002], [MOONEY et *al.*,2000] .Enfin, les recommandations collaboratives sont générées à partir de ces « pseudo évaluations » des utilisateurs.

Dans leur approche, Claypool et al ont proposé la méthode de pondération des prédictions des filtrages qui se réalisent séparément [CLAYPOOL et *al.*, 1999]. Ainsi, afin de générer des recommandations pour un nouvel utilisateur, on élimine simplement la prédiction collaborative dans le calcul de la formule de prédiction.

Dans les méthodes précitées, l'effort demandé au nouvel utilisateur est important : il doit procéder à une réflexion pour synthétiser ses centres d'intérêt sous la forme de termes, ou bien rechercher des exemples pertinents de documents. Cette dernière tâche peut être automatisée lorsque le système dispose de données externes sur les utilisateurs, par exemple pour des chercheurs académiques qui sont aussi auteurs de publications [MIDDLETON et *al.*, 2002], [MIDDLETON et *al.*, 2004], mais en règle générale, on ne dispose pas de telles données. Dans la plupart des cas, le profil résultant est incomplet et bruité.

3.3.3.2. Combinaison avec d'autres techniques

Une approche alternative pour le démarrage à froid consiste à associer le nouvel utilisateur à un profil-type (« stéréotype ») parmi ceux prédéfinis. Le processus de construction de ces profils-type nécessite un ensemble de données d'apprentissage, puis le processus de démarrage à froid doit confronter à ces profils-type certaines informations relatives au nouvel utilisateur.

Le système obtient généralement les informations nécessaires en interagissant avec l'utilisateur, par exemple en lui posant une série de questions [KRULWICH, 1997]. Cette méthode demande souvent des experts dans le domaine applicatif.

Par contre, on peut exploiter une source d'informations « démographiques » externe comme les pages Web personnelles des utilisateurs [PAZZANI, 1999]. PAZZANI a étudié entre autres les performances de cette approche en la prenant pour unique base d'un système de recommandation « démographique », et nous retenons de son travail que les performances de recommandation via les données démographiques sont certes moins bonnes que celles de systèmes basés sur le contenu ou collaboratifs, mais toutefois acceptables.

De la même façon, dans son système ProfBuilder, Wasfi a proposé d'explorer implicitement l'historique des pages Web visitées pour construire les profils [WASFI, 1999]. Mais, cette méthode pose le problème de la confidentialité.

3.4. Conclusion

Au long de ce chapitre, nous avons expliqué qu'est-ce que c'est le problème du démarrage à froid. Et nous avons décrit quelques solutions proposées.

Nous allons présenter dans le chapitre qui suit l'approche que nous allons utiliser pour faire face à ce problème.

Chapitre 4 :

Solution proposée pour remédier au problème
du démarrage à froid

4.1. Introduction

La recommandation est un domaine scientifique qui cherche à personnaliser l'accès à l'information pour un utilisateur donnée et ainsi lui faciliter le choix de contenu dans un catalogue trop vaste pour qu'il puisse s'en faire une idée d'ensemble. En pratique, les systèmes de recommandation, à partir de connaissance sur un utilisateur filtrent un ensemble de contenus et produisent une liste souvent ordonnée de ces quelques contenus sélectionnés et jugés pertinents pour lui.

Dans ce qui suit on va représenter deux approches qui vont nous permettre de définir le profil d'un utilisateur d'une manière explicite et implicite. Pour cela nous avons exploité le jeu de données MovieLens qui a été collecté par le groupe de recherche GroupLens de l'université du Minnesota portant sur les évaluations faites par des utilisateurs sur un ensemble de films pour avoir les centres d'intérêts permettant de construire les profils utilisateurs.

4.2. Problématique et motivation

Le démarrage à froid est l'une des problématiques les plus importantes dans les systèmes de recommandation lors de la première interaction de l'utilisateur avec le système, car son profil est inexistant. Par conséquent le système ne peut pas lui fournir des recommandations pertinentes. La réponse à ce problème est que l'utilisateur est mis à contribution,

-Pour remédier à ce problème, nous proposons deux approches :

1-Approche explicite :

Lorsqu'un premier utilisateur arrive dans le système, un formulaire lui sera proposé afin de décrire ses intérêts. Les intérêts décrits correspondent aux items les plus populaires pour le prochain utilisateur afin de les évaluer pour avoir son profil, en se basant sur la représentation vectorielle du profil.

2-Approche implicite :

Lorsqu'un nouvel utilisateur arrive dans le système, ce dernier lui propose de se connecter via un réseau social. Cela lui permet de récupérer les intérêts de cet utilisateur afin d'avoir son profil.

En se basant sur la détection des intérêts d'un utilisateur par l'exploitation des données de son réseau social.

4.3. Description de l'approche

4.3.1. Architecture de l'approche explicite

Les principaux composants dans notre approche de recommandation sont illustrés dans la figure 4.1. Ils intègrent l'indexation des items ainsi que la modélisation du profil utilisateur et leurs exploitations dans le processus de recommandation.

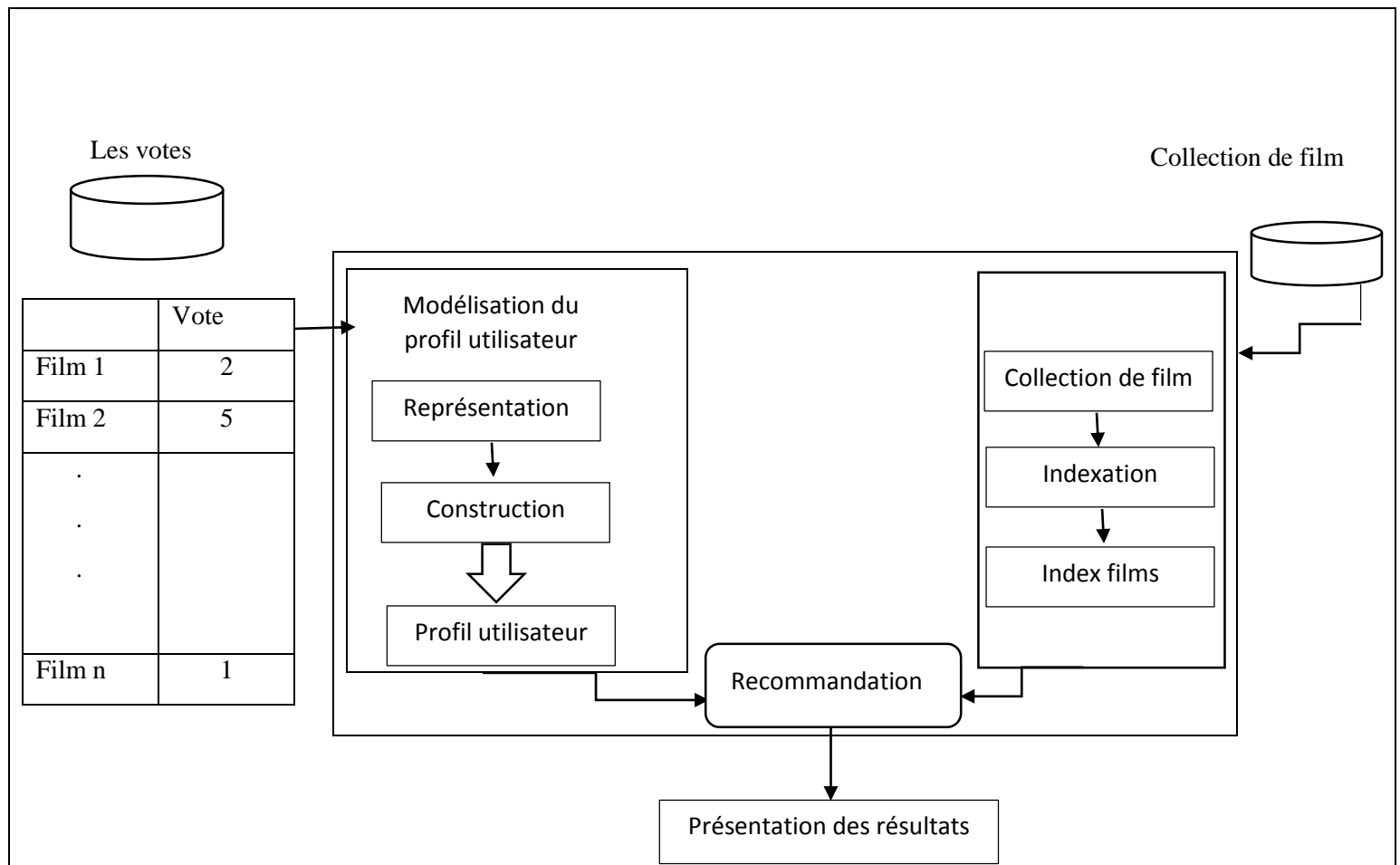


Figure 4.1. Architecture de l'approche explicite.

4.3.1.1. Modélisation du profil utilisateur

Pour réaliser la modélisation du profil utilisateur dans le système de recommandation, il faut passer par deux étapes :

- Représentation du profil utilisateur.
- Construction du profil utilisateur.

1. Représentation du profil utilisateur :

La représentation du profil utilisateur peut être réalisée selon différentes approches. Nous allons utiliser une représentation basée sur le modèle vectoriel du filtrage basé contenu pour la simplicité de sa mise en œuvre.

2. Construction du profil utilisateur :

La construction du profil utilisateur se fait à partir des informations collectées. Cependant, dans la plupart des cas, une technique de construction du profil utilisateur est nécessaire. Nous appliquerons la technique *tf-idf* qui est la plus utilisée dans le modèle basé contenu.

En se basant sur l'approche vectorielle, on va trouver le degré de pertinence des films pour chaque utilisateur.

Un profil utilisateur \mathbf{U} est alors représenté par un vecteur de termes pondérés,

Où chaque terme représente le genre d'un film et le poids w_{cik} dénote l'importance du genre \mathbf{c}_i par rapport au profil de l'utilisateur \mathbf{U} :

$$\mathbf{U} = (w_{c1,k}, w_{c2,k}, \dots, w_{ci,k})$$

Il est calculé comme suit :

$$w_{ci,k} = \sum_{F_j \in A} w_{ij} * n_j \dots \dots \dots 4.1$$

Où :

n_j : représente la note attribuée par l'utilisateur pour le film \mathbf{F}_j

w_{ij} : représente le poids du genre \mathbf{c}_i dans le film \mathbf{F}_j . Il est calculé comme suit :

$$W_{i,j} = \frac{Occ_{i,j}}{N_{c_j}^2} \dots\dots\dots 4.2$$

Occ_{i,j} : Occurrence de la genre **C_{fi}** dans le film **F_j**

N_{c_j} : Le nombre de genre dans le film **F_j**

4.3.1.2. Exploitation du profil utilisateur dans le processus de recommandation

Le système de recommandation permet de recommandé au mieux des éléments qui correspondent au gout des utilisateurs.

Dans le cas d’un nouveau utilisateur, le système lui suggère une liste de 15 films à noter ((j’aime =1) ou (j’aime pas=-1)). Puis, il va lui recommander des films selon ses préférences.

Après avoir définit le profil utilisateur par son centre d’intérêt, il sera inclut dans le module de recommandation.

Pour la prédiction des scores de pertinence, on effectue le produit scalaire entre le vecteur profil utilisateur et les vecteurs profils items (vecteurs pondérés des films). Ceci est dénoté par la formule suivante :

$$\text{Produit scalaire} = \sum_{F_j \in D} W_{c_i,k} * W_{s_i,k} \dots\dots\dots 4.3$$

Le résultat obtenu, correspond au degré de pertinence d’un film pour l’utilisateur U.

4.3.1.3. Profil item

Dans notre approche un item correspond à un film. Chaque film **F_j** est représenté sous forme d’un vecteur de catégories **c_{f_j}**

$$F_j = (w_{s1 j}, w_{s2 j}, \dots w_{s_i k})$$

Le poids d’une catégorie dans le film **F_j** est dénoté par la formule suivante :

$$w_{i, k} = tf_{i,k} * \log \frac{N}{n_i} \dots\dots\dots 4.4$$

Avec **N** est le nombre total des films et **n_i** le nombre des films qui contiennent la catégorie **c_{fi}**.

4.3.1.4. Illustration de l'approche explicite proposée

Pour l'illustration de notre approche, nous avons une collection de films issue du système de recommandation MovieLens.

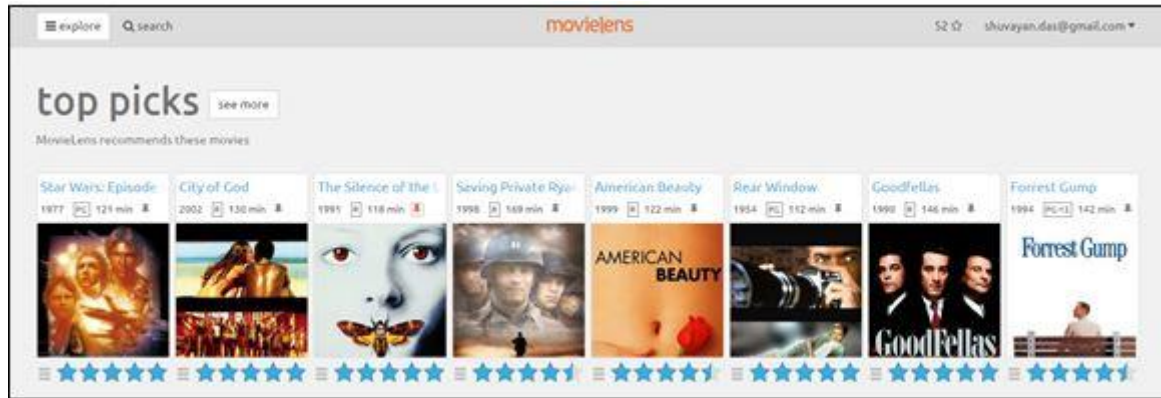


Figure 4.2. Image de MovieLens

La Figure 4.2 illustre ce système de recommandation des films, qui recommande les films à regarder en fonction des films déjà notés et de leur note.

À un niveau rudimentaire, les entrées sont prises en compte par le système et un profil est créé par rapport aux attributs qui contiennent des goûts et des dégoûts (peuvent être basés sur des mots-clés / balises de films déjà aimés ou non noté).

L'utilisateur a aimé 52 films dans différents genres. Ces éléments (films) sont des attributs et peuvent nous aider à déterminer le goût d'un utilisateur.

On prend la liste des 6 meilleurs films recommandés par MovieLens.

$$F = \{F1, F2, F3, F4, F5, F6\}$$

L'index de la collection inclut les différents genres de films.

Les films sont représentés sous forme binaire. Le tableau 4.1 représente les valeurs associées à ces films.

Les films qui sont du genre i sont notés par 1 sinon ils sont notés par 0.

Films	Aventure	Action	Science-fiction	Drama	Crime	Suspense
Film 1	1	1	1	0	0	0
Film 2	0	0	0	1	0	0
Film 3	0	0	0	1	0	0
Film 4	0	0	0	1	1	0
Film 5	0	0	1	1	0	0
Film 6	1	1	1	0	0	1

Tableau 4.1. Représentation binaire des films

Le tableau 4.2 suivant, spécifie si un utilisateur a aimé ou non un film recommandé, L'utilisateur 1 a aimé Film 1 pour cela la valeur 1 est attribuée, n'a pas aimé le Film 2 pour cela la valeur -1 est attribuée et si l'utilisateur n'a pas consulté le Film on met rien.

	Film 1	Film 2	Film 3	Film 4	Film 5	Film 6
Utilisateur 1	1			-1	1	
Utilisateur 2	-1			1		1

Tableau 4.2. Les notes des utilisateurs attribués pour les films

1-Représentation du profil des utilisateurs :

Le tableau 4.3 représente le nombre de genres qui apparaissent dans chaque film

	F1	F2	F3	F4	F5	F6
Totale	3	1	1	2	2	4

Tableau 4.3. Représentation des totale des genres

Le tableau 4.4 représente le poids des genres des films proposé.

On prend exemple du genre aventure dans le film 1

$$W_{1,1} = \frac{1}{9} \text{ et on normalise comme suit : } \frac{w_{1,1}}{\sqrt{w_{1,1}^2 + w_{1,2}^2 + w_{1,3}^2}} = 0.577350269$$

	Aventure	Action	Science-fiction	Drama	Crime	Suspense
Film 1	0.577350269	0.577350269	0.577350269	0	0	0
Film 2	0	0	0	1	0	0
Film 3	0	0	0	1	0	0
Film 4	0	0	0	0.707106781	0.707106781	0
Film 5	0	0	0.707106781	0.707106781	0	0
Film 6	0.5	0.5	0.5	0	0	0.5

Tableau 4.4. Représentation des vecteurs des films par rapport à l'utilisateur après la normalisation

2- Modélisation du profil utilisateur :

Une fois on a calculé les poids des genres des films on applique la formule 4.1 pour avoir les poids des genres des films par rapport à l'utilisateur :

utilisateur 1 dans le genre aventure : $WC_{1,1} = 0.577350269$ et la note de l'utilisateur est 1 pour le film 1.

Donc le poids du genre adventre sera comme suit : $WC_{1,1} = 0.577350269 \times 1 = 0.577350269$.

Les résultats sont représentés dans le tableau 4.5 ci-dessus :

	Aventure	Action	Science-fiction	Drama	Crime	Suspense
Utilisateur 1	0.577350269	0.577350269	1.28445705	- 1.414213562	- 0.707106781	0
Utilisateur 2	- 0.077350269	- 0.077350269	- 1.78445705	0.707106781	0	0.5

Tableau 4.5. Vecteurs profil utilisateur

3- vecteurs des films :

Le tableau 4.6 représente les poids des genres des films proposés.

Prenant exemple du genre aventure dans le film 1 son poids est calculé comme suit :

Dans cet exemple on prend le nombre total des films, égale à 100. (N=100)

$w_{i,k} = tf \times idf = 1 \times \log\left(\frac{100}{3}\right)$. (n=3 est le nombre de film dont apparait le genre « science-fiction »)

	Adventure	Action	Science-fiction	Drama	Crime	Suspense
Film 1	0.980900788	0.980900788	0.879234453	0	0	0
Film 2	0	0	0	1.397940009	0	0
Film 3	0	0	0	1.397940009	0	0
Film 4	0	0	0	1.201353211	1.414213562	0
Film 5	0	0	1.076837887	1.201353211	0	0
Film 6	0.849485002	0.849485002	0.761439372	0	0	1

Tableau 4.6. Vecteurs des films

3- La prédiction des scores de pertinence :

Pour la prédiction des films, le système utilise le produit scalaire des vecteurs pondérés des films et vecteur pondérés des utilisateurs (profil). Le résultat obtenu représente le degré de pertinence d'un film pour un utilisateur.

Les résultats obtenus sont représentés dans le tableau (4.7) suivant :

	Film 1	Film 2	Film 3	Film 4	Film 5	Film 6
Utilisateur 1	2.26198556	-1.414213562	-1.414213562	2.698970003	-0.315817987	1.958936958
Utilisateur 2	-1.720701997	0.707106781	0.707106781	0.849485001	-1.072085957	-1.990171642

Tableau 4.7. Degré de pertinence des films par rapport à l'utilisateur

D'après ces résultats le Film 4, Film 1, Film 6 sont le mieux apprécié par l'utilisateur 1.

Et pour l'utilisateur 2 aussi c'est le Film 4, Film 3, Film 2 qu'il a apprécié le plus.

4.3.2. Description de l'approche implicite

4.3.2.1. Architecture de l'approche

Les principaux composants de notre approche implicite de recommandation sont illustrés par la figure (4.3). Ils intègrent une base de données qui représente les films collectés du réseau social et la collection de film issue de MovieLens.

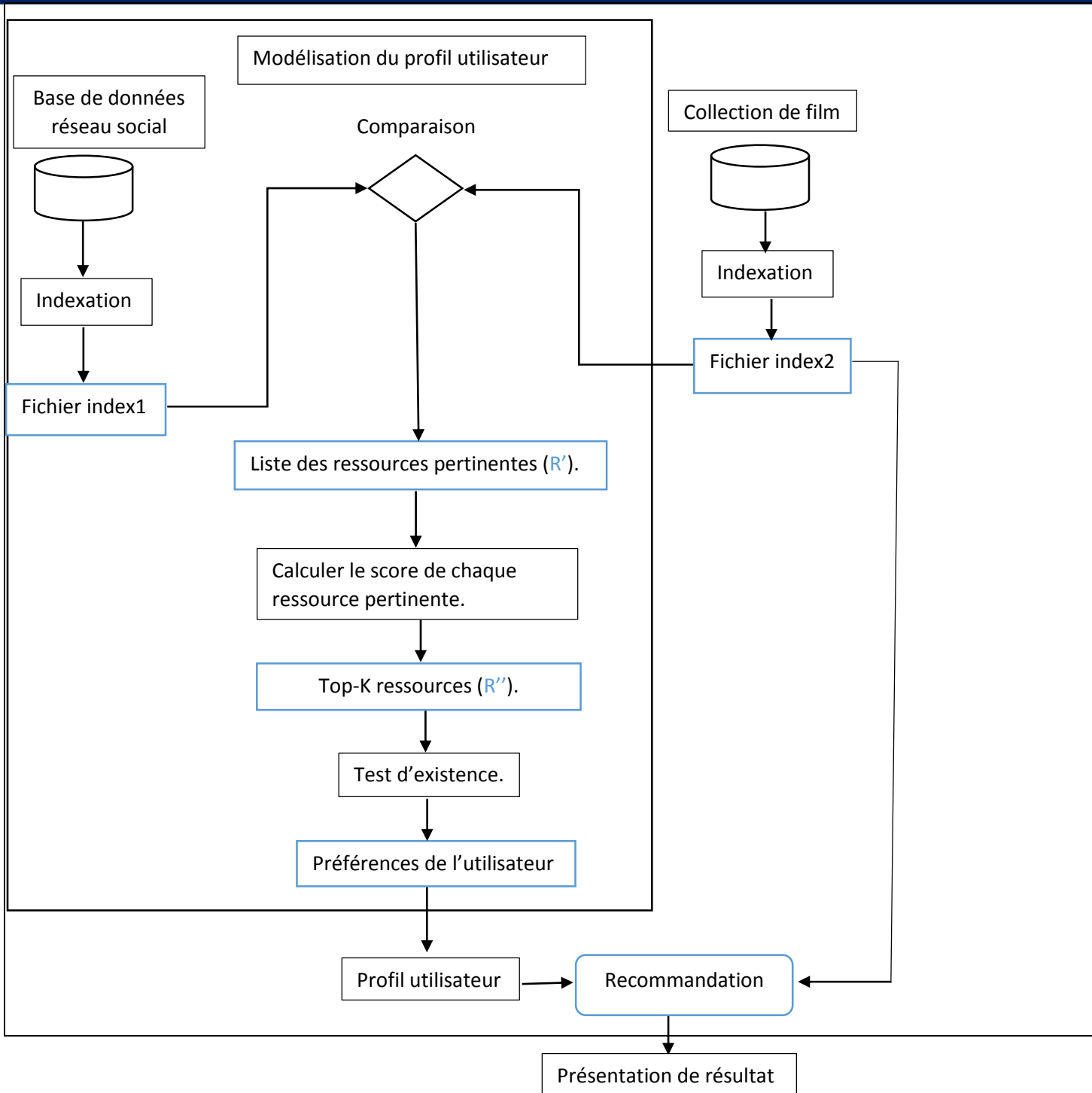


Figure 4.3. Architecture de l'approche implicite.

Notre approche est constituée à partir de l'hypothèse que l'environnement social et en particulier les amis proches d'un individu, fournissent une information à partir de laquelle les intérêts de cet individu peuvent être extraits.

Les personnes proches peuvent être celles partageant certains comportements communs.

-Nous nous concentrons sur les informations fournies par des personnes dans leurs réseaux sociaux. Elle se base sur l'ensemble des films ajoutés par une personne sur son profil facebook.

Représentation et construction du profil utilisateur :

- A. Extraction des préférences de chaque ami de l'utilisateur à partir de son réseau social.
- B. Extraction des préférences de l'utilisateur à partir de son réseau social.
- C. Vérifier l'existence des items collectés à partir du réseau social (facebook) de chaque ami de l'utilisateur dans la collection de l'item choisi, cela nous fournit une liste de ressources pertinentes (R').
- D. Construction d'une top-K ressources (R'') en attribuant des scores pour chaque ressources déjà extraites des profils des amis proches de l'utilisateur.
Ce score est le résultat d'une fonction de similarité qui prend en considération la ressource fournie par l'ami proche et celle de la collection.
- E. Vérifie l'existence des ressources fournies par l'ami proche dans les ressources fournies par l'utilisateur lui-même
- F. A la fin, on aura le profil de l'utilisateur.

Une fois le profil représenté, sa construction consiste à l'acquisition et la sélection des données nécessaires à sa représentation. Nous appliquons la technique `tf_idf` qui est la plus utilisée dans le modèle basé contenu.

Cette approche tire parti des films que l'utilisateur à ajouter dans son profil pour lui recommander des films similaires.

4.3.2.2. Exploitation du profil utilisateur dans le processus de recommandation

L'objectif de notre système de recommandation est de générer des films qui seront appropriés aux goûts des utilisateurs.

Dès lors, notre système consiste à inclure le profil d'un utilisateur défini par son centre d'intérêt, comme composante dans le module de prédiction du processus de recommandation.

4.3.2.3. Inférence du profil utilisateur et du profil item

Le profil utilisateur et le profil item sont représentés sous forme de deux vecteurs de termes pondérés chaque terme correspond à une catégorie de film.

Le profil utilisateur est inféré en se basant sur les intérêts récupérés du profil social de l'utilisateur. Il est construit à partir de l'ensemble des films. Et le profil item représente l'ensemble des films de la collection.

Un profil utilisateur U est alors représenté par un vecteur des termes pondérés. Tel qu'un terme correspond au genre d'un film et son poids dénote l'importance du genre cf_i par rapport au profil de l'utilisateur U :

$$U = (w_{c1,k}, w_{c2,k}, \dots, w_{ci,k})$$

Et un profil item F est représenté par un vecteur des films pondérés :

$$F = (w_{s1,k}, w_{s2,k}, \dots, w_{si,k})$$

$w_{ci,k}$: Le poids du genre de film cf_i dans le vecteur profil utilisateur U .

$w_{si,k}$: Le poids du genre de film cf_i dans le vecteur profil item F .

Les deux poids précédents se calculent de la même façon comme suit :

$$\mathbf{TF.IDF}(cf_i, fi) = \mathbf{TF}(cf_i, fi) * \log\left(\frac{N}{n_k}\right) \dots \dots \dots 4.5$$

Où : N : dénote le nombre total des films f_i .

n_k : représente le nombre de films contenant le genre cf_i apparaît au moins une fois,
Avec :

$$\mathbf{TF}(c_{fi}, f_i) = \frac{f_{k,i}}{\max_z f_{z,i}} \dots\dots\dots 4.6$$

Où : $f_{k,i}$ représente le nombre d'occurrences du genre c_{fi} dans le film f_i ,

$\max_z f_{z,i}$ est le maximum des fréquences $f_{z,i}$ des genres c_{fz} apparaissant dans le films f_i .

Afin que tous les poids appartiennent à l'intervalle [0,1], et que tous les documents soient représentés par des vecteurs de même longueur ; les poids obtenus par la fonction TF.IDF sont généralement normalisés en utilisant **la normalisation cosinus** :

$$W_{k,i} = \frac{\mathbf{TF.IDF}(c_{fk}, f_i)}{\sqrt{\sum_{s=1}^{|T|} \mathbf{TF.IDF}(f_s, f_i)^2}} \dots\dots\dots 4.6$$

Une fois que les poids sont calculés et normalisés. Le contenu d'un film f_i , est défini par :

$$\mathbf{Content}(f_i) = (w_{1i}, w_{2i}, \dots, w_{ki})$$

A la fin de cette étape, le système obtient une liste des genres avec leurs poids indiquant leur importance pour chaque film. Afin de calculer la similarité entre les films (ou vecteur de films), la mesure de similarité la plus utilisée est **la Similarité Cosinus**

$$\mathbf{Sim}(f_i, f_j) = \frac{\sum_k w_{cik} * w_{sjk}}{\sqrt{\sum_k w_{cik}^2} * \sqrt{\sum_k w_{sjk}^2}} \dots\dots\dots 4.7$$

4.3.2.4. Illustration de l'approche implicite proposée

Pour illustrer notre approche, nous avons pris un exemple de recommandation de film.

1- Représentation binaire des films de la collection :

	Adventure	Comedy	Fantasy	Drama	Action
Toy Story	1	1	1	0	0
Jumanji	1	0	1	0	0
Grumpier Old Men	0	1	0	0	0
Waiting to Exhale	0	1	0	1	0
Father of the Bride Part II	0	1	0	0	0
Heat	0	0	0	0	1
Sabrina	0	1	0	0	0
Tom and Huck	1	0	0	0	0
Sudden death	0	0	0	0	1
Golden eye	1	0	0	0	1
American president	0	1	0	1	0
Dracula	0	1	0	0	0
Balto	1	0	0	0	0
Nixon	0	0	0	1	0
Cutthroat Island	1	0	0	0	1
Casino	0	0	0	1	0
Sense and Sensibility	0	0	0	1	0
Four Rooms	0	1	0	0	0
Ace Ventura: When Nature Calls	0	1	0	0	0
Money Train	0	1	0	1	1

Tableau 4.8. Représentation binaire des films de la collection MovieLens

- Profil récupéré du réseau social Facebook :

Prenant un exemple d'un profil utilisateur construit à partir de son profil social.

Nom utilisateur : Mathieu.

Données collectées :

Films: Toy Story, Jumanji, City of lost children, Mortal kombat, Georgia, Friday, Fair game.

2-Vecteur du profil utilisateur :

Le tableau 4.9 montre la représentation binaire des films récupérés du profil Facebook de l'utilisateur.

	Adventure	Comedy	Fantasy	Drama	Action
Toy Story	1	1	1	0	0
Jumanji	1	0	1	0	0
City of lost children	1	0	1	1	0
Mortal kombat	1	0	1	0	1
Georgia	0	0	0	1	0
Friday	0	1	0	0	0
Fair game	0	0	0	0	1

Tableau 4.9. Représentation binaire des films de l'utilisateur

$N = 7$ représente le nombre total des films de l'utilisateur.

Le tableau 4.10 montre le nombre des films de chaque genre cfi :

n_k : est le nombre de film dans chaque genre.

	Adventure	Comedy	Fantasy	Drama	Action
n_k	4	2	4	2	2

Tableau 4.10. Nombre de films dans chaque genre

Le tableau 4.11 suivant représente le score de chaque ressource pertinente.

	Adventure	Comedy	Fantasy	Drama	Action
Toy Story	0.298661118	0.668586552	0.298661118	0	0
Jumanji	0.707106786	0	0.707106786	0	0
City of lost children	0.298661118	0	0.298661118	0.668586552	0
Mortal kombat	0.298661118	0	0.298661118	0	0.668586552
Georgia	0	0	0	1	0
Friday	0	1	0	0	0
Fair game	0	0	0	0	1

Tableau 4.11. Vecteur de films par rapport à l'utilisateur

3-profil item :

Le tableau 4.12 suivant représente les vecteurs des films de la collection :

	Adventure	Comedy	Fantasy	Drama	Action
Toy Story	0.44770343	0.257750315	0.856228015	0	0
Jumanji	0.463359629	0	0.886170329	0	0
Grumpier Old Men	0	1	0	0	0

Waiting to Exhale	0	0.498937802	0	0.866637798	0
Father of the Bride Part II	0	1	0	0	0
Heat	0	0	0	0	1
Sabrina	0	1	0	0	0
Tom and Huck	1	0	0	0	0
Sudden death	0	0	0	0	1
Golden eye	0.655713231	0	0	0	0.755010039
American president	0	0.498937802	0	0.866637798	0
Dracula	0	1	0	0	0
Balto	1	0	0	0	0
Nixon	0	0	0	1	0
Cutthroat Island	0.655713231	0	0	0	0.755010039
Casino	0	0	0	1	0
Sense and Sensibility	0	0	0	1	0
Four Rooms	0	1	0	0	0
Ace Ventura: When Nature Calls	0	1	0	0	0
Money Train	0	0.353177248	0	0.613456729	0.706354497

Tableau 4.12. Vecteurs des films du profil item après la normalisation

4-Degré de similarité entre les vecteurs des films et les vecteurs des utilisateurs.

Les résultats de calcul de similarité entre le vecteur U et le vecteur F sont dans le tableau ci-dessous :

	Toy story	Jumanji	City of lost children	Mortal kombat	Georgia	Friday	Fair game
Toy Story	0.710365293	0.909785695	0.577894029	0.577894029	0	0.257750315	0
Jumanji	0.509660034	1	0.50966003	0.50966003	0	0	0
Grumpier Old Men	0.37765744	0	0	0	0	1	0
Waiting to Exhale	0.421816345	0	0.732680482	0	0.866637797	0.2494689	0
Father of the Bride Part II	0.845402122	0	0	0	0	1	0
Heat	0	0	0	0.845428717	0	0	1
Sabrina	0.845428724	0	0	0	0	1	0
Tom and Huck	0.37765744	0.707106781	0.37765744	0.37765744	0	0	0
Sudden death	0	0	0	0.845428717	0	0	1
Golden eye	0.24763498	0.463659271	0.24763498	0.885942148	0	0	0.755010039
American president	0.421816345	0	0.732680482	0	0.866637797	0.2494689	0
Dracula	0.845428717	0	0	0	0	1	0
Balto	0.37765744	0.707106781	0.37765744	0.37765744	0	0	0
Nixon	0	0	0.845428717	0	1	0	0
Cutthroat Island	0.24763498	0.463659271	0.24763498	0.885942148	0	0	0.755010039
Casino	0	0	0.845428717	0	1	0	0
Sense and Sensibility	0	0	0.845428717	0	1	0	0
Four Rooms	0.845428717	0	0	0	0	1	0
Ace Ventura: When Nature Calls	0.845428717	0	0	0	0	1	0
Money Train	0.298586187	0	0.51861762	0.472259117	0.613456728	0.176588623	0.706354496

Tableau 4.13. Degré de similarité entre les vecteurs pondérés des films et les vecteurs pondérés des utilisateurs.

4.4. Conclusion

Dans ce chapitre, nous avons décrit la solution que nous avons proposée pour remédier au problème du démarrage à froid dans un système de recommandation basé contenu pour un nouvel utilisateur.

Nous avons illustré l'architecture de nos approches de recommandation, et nous avons défini la modélisation du profil utilisateur à savoir sa représentation et sa construction ainsi que son exploitation dans le processus du système de recommandation dans chacune des approches.

Nous avons illustré toutes les étapes des deux approches avec des exemples.

Nous présentons dans le chapitre qui suit l'implémentation de la solution proposée.

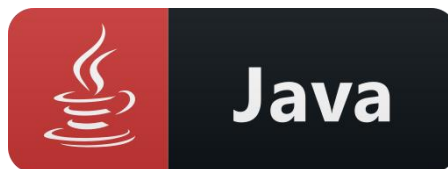
Chapitre 5 :
Implémentation de la solution proposée

5.1. Introduction

Dans ce chapitre, nous présentons l'implémentation des approches de notre solution ainsi que les différents outils que nous avons exploités et enfin nous présentons un exemple de résultat de recommandation obtenu.

5.2. Environnement de développement

- **Le langage Java :** est un langage de programmation orienté objet, basé sur des classes, conçu pour avoir le moins de dépendances d'implémentation possible. Il est un usage général langage de programmation destiné à permettre aux développeurs d'applications écrire une fois, exécuter partout (WORA), ce qui signifie que compilé le code Java peut fonctionner sur toutes les plateformes supportant Java sans avoir besoin de recompilation. Les applications Java sont généralement compilées en bytecode qui peut s'exécuter sur n'importe quelle machine virtuelle Java (JVM) quelle que soit l'architecture informatique sous-jacente. La syntaxe de Java est similaire à C et C++, mais a moins de fonctionnalités de bas niveau que l'un ou l'autre. L'environnement d'exécution Java fournit des fonctionnalités dynamiques (telles que la réflexion et la modification du code d'exécution) qui ne sont généralement pas disponibles dans les langages compilés traditionnels.

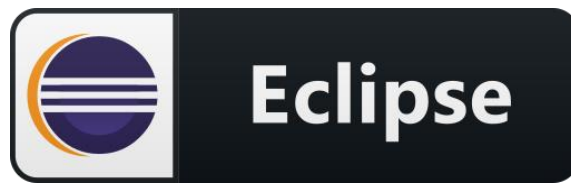


- **JAVA EE :** signifie Java Enterprise Edition, qui était auparavant connu comme J2EE et est actuellement connu sous le nom de Jakarta EE. Il s'agit d'un ensemble de spécifications entourant Java SE (Standard Edition). Java EE fournit une plate-forme aux développeurs avec des fonctionnalités d'entreprise telles que l'informatique distribuée et les services Web. Les applications Java EE sont généralement exécutées sur des temps d'exécution de référence tels que des microserveurs ou des serveurs d'applications. Des exemples de certains contextes où Java EE est utilisé sont le commerce électronique, la comptabilité et les systèmes d'information bancaire.

- **L'IDE Eclipse :**

C'est un outil puissant, gratuit, libre et multiplateforme. Les avantages d'un IDE dans le développement d'application web java EE sont multiples, et sans toutes fois être exhaustif, voici une liste :

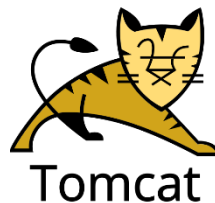
- ✓ Intégration des outils nécessaire au développement et au déploiement d'une application.
- ✓ Paramétrage aisé et centralisé des composants d'une application.
- ✓ Multiple moyen de visualisation de l'architecture d'une application.
- ✓ Génération automatique de portions de code.
- ✓ Assistance à la volée lors de l'écriture du code.
- ✓ Outil de débogage ...etc.



- **JAVASCRIPT :** est un langage de programmation de scripts principalement employé dans les pages web interactives et à ce titre est une partie essentielle des applications web. Avec les technologies HTML et CSS, JavaScript est parfois considéré comme l'une des technologies cœur du World Wide Web. Une grande majorité des sites web l'utilisent, et la majorité des navigateurs web disposent d'un moteur JavaScript dédié pour l'interpréter, indépendamment des considérations de sécurité qui peuvent se poser le cas échéant. C'est un langage orienté objet à prototype, c'est-à-dire que les bases du langage et ses principales interfaces sont fournies par des objets qui ne sont pas des instances de classes, mais qui sont chacun équipés de constructeurs permettant de créer leurs propriétés, et notamment une propriété de prototypage qui permet de créer des objets héritiers personnalisés. En outre, les fonctions sont des objets de première classe.



- **SERVEUR APACHE TOMCAT :** est une implémentation open source des technologies java Servlet, java Server Pages, Java Expression Language et WebSocket. TOMCAT fournit un environnement de serveur Web http "pure java" dans lequel le code java peut s'exécuter.



- **Java Database Connectivity (JDBC) :** est une interface de programmation d'application (API) pour le langage de programmation Java , qui définit comment un client peut accéder à une base de données . Il s'agit d'une technologie d'accès aux données Java utilisée pour la connectivité aux bases de données Java. Il fait partie de la plate- forme Java Standard Edition d' Oracle Corporation . Il fournit des méthodes pour interroger et mettre à jour les données dans une base de données et est orienté vers les bases de données relationnelles . Un pont JDBC vers ODBC permet les connexions à n'importe quelle source de données accessible ODBC dans l'environnement hôte de la machine virtuelle Java (JVM). Les classes JDBC sont contenues dans le package Java (java.sql).
- **PHPMYADMIN :** est un outil logiciel libre écrit en PHP , destiné à gérer l'administration de MySQL sur le Web. phpMyAdmin prend en charge un large éventail d'opérations sur MySQL et MariaDB. Les opérations fréquemment utilisées (gestion des bases de données, des tables, des colonnes, des relations, des index, des utilisateurs, des autorisations, etc.) peuvent être effectuées via l'interface utilisateur, tandis que vous avez toujours la possibilité d'exécuter directement toute instruction SQL.
- **La bibliothèque de balise standard de Java Server Pages (JSTL) :** est une collection de quatre bibliothèques de balises personnalisées qui étendent la spécification JSP. En tant que composant, il est alloué dans la plate-forme de développement d'applications Web Java EE.

5.3. Implémentation de notre solution

5.3.1. Jeu de données

Pour tester notre solution, nous avons utilisé le jeu de données MovieLens portant sur les évaluations faites par des utilisateurs sur un ensemble de films. Ce jeu de données contient 610 fournisseurs, 100836 évaluations collectées explicitement et implicitement d'environ 9742 films. Les données de ce jeu de données sont regroupées dans quatre différentes tables.

5.3.1.1. Table Links

Les identificateurs qui peuvent être utilisés pour établir des liens vers d'autres sources de données de films sont contenus dans le fichier `links.csv`.

Chaque ligne de ce fichier après la ligne d'en-tête représente un film selon le format suivant :

movieId, imdbId, tmdbId

5.3.1.2. Table Movie

Les informations sur le film sont contenues dans le fichier `movies.csv`. Chaque ligne de ce fichier après la ligne d'en-tête représente un film, selon le format suivant :

movieId, titre, genres

5.3.1.3. Table Tags

Toutes les balises sont contenues dans le fichier «tags.csv». Chaque ligne de ce fichier après la ligne d'en-tête représente une balise appliquée à un film par un utilisateur selon le format suivant :

userId,movieId,tag,timestamp

5.3.1.4. Table Rating

Toutes les évaluations sont contenues dans le fichier «ratings.csv». Chaque ligne de ce fichier après la ligne d'en-tête représente une évaluation d'un film par un utilisateur, selon le format suivant :

userId,movieId,rating,timestamp

5.3.2. Présentation de l'application

Maintenant qu'on a présenté notre environnement de travail, cette partie est consacrée à la présentation des interfaces de notre application.

Elle prend en charge plusieurs interfaces que nous allons présenter en illustrant son fonctionnement.

5.3.2.1. Interface d'accueil

Cette interface permet à l'utilisateur de choisir soit de se connecter et il sera rediriger à une interface de connexion, ou bien de s'inscrire s'il n'a pas de compte et il sera rediriger à l'interface d'inscription.

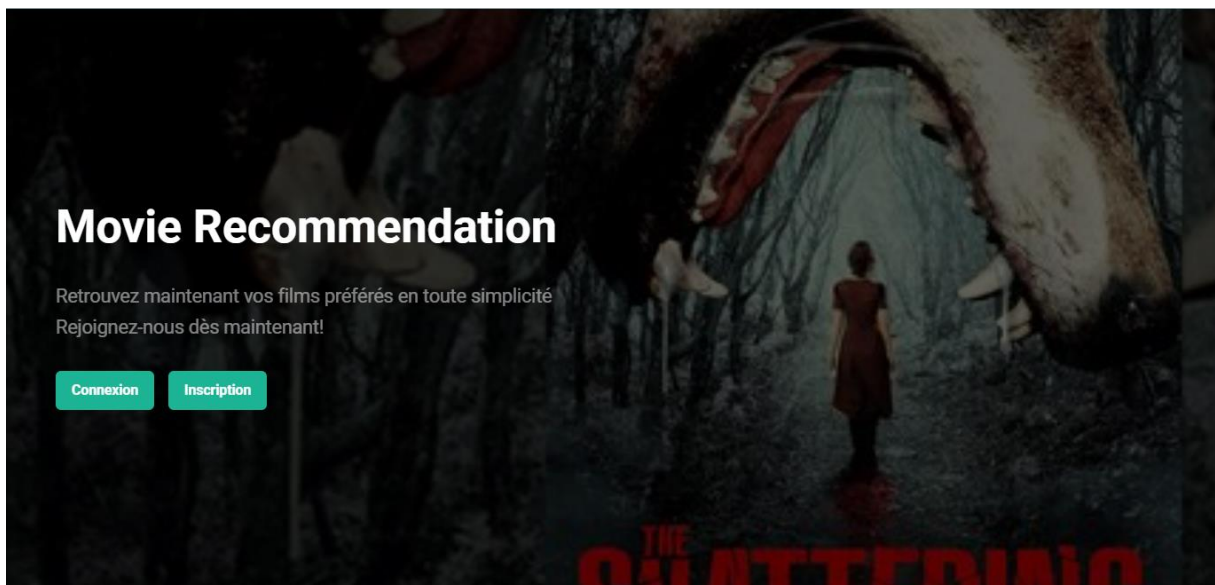


Figure 5.1. Interface d'accueil

5.3.2.2. Interface de connexion

Une fois l'utilisateur a cliqué sur le bouton « connexion » de l'interface accueil. Il sera redirigé dans l'interface de connexion, et un choix lui sera proposé soit de se connecter avec le réseau social Facebook, ou bien une connexion simple.

La connexion simple, permet à l'utilisateur de se connecter s'il possède déjà un compte. Notons qu'il est nécessaire de se connecter via « email » et « mot de passe ».

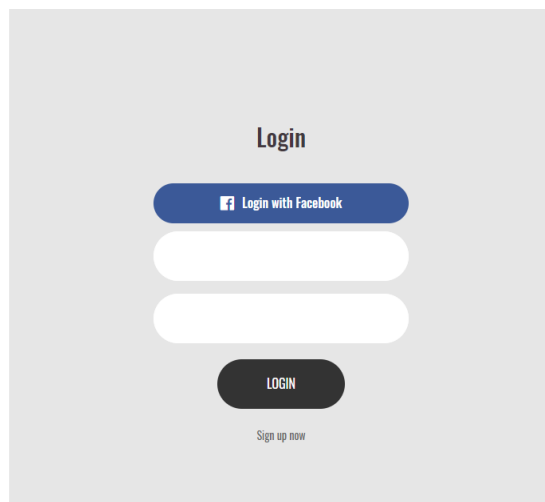


Figure 5.2. Interface de connexion

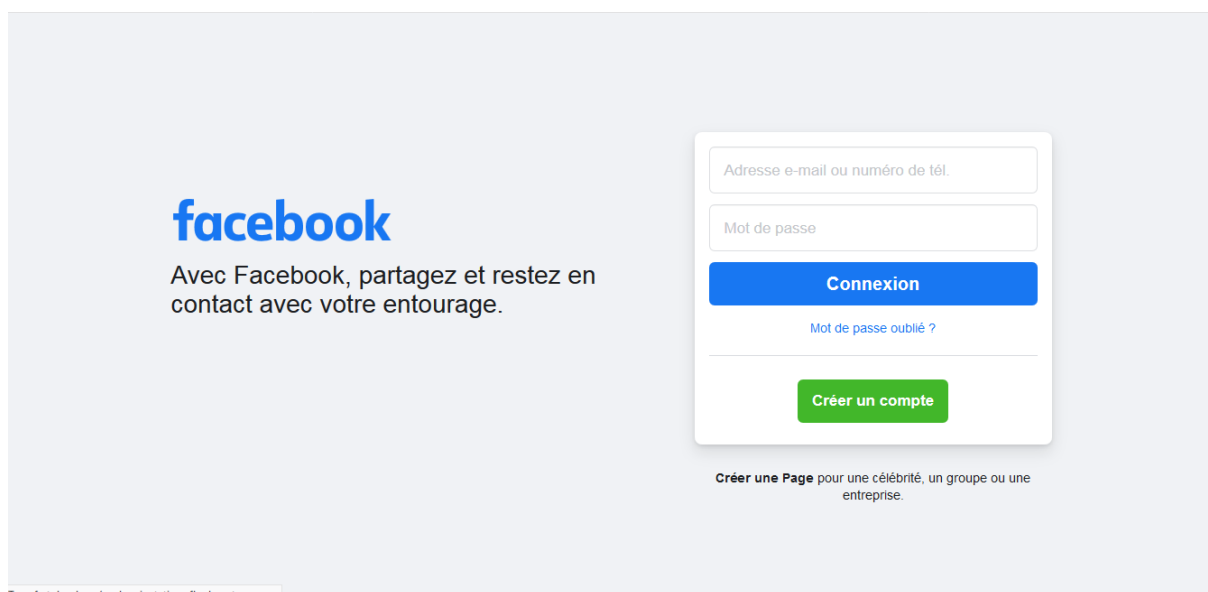


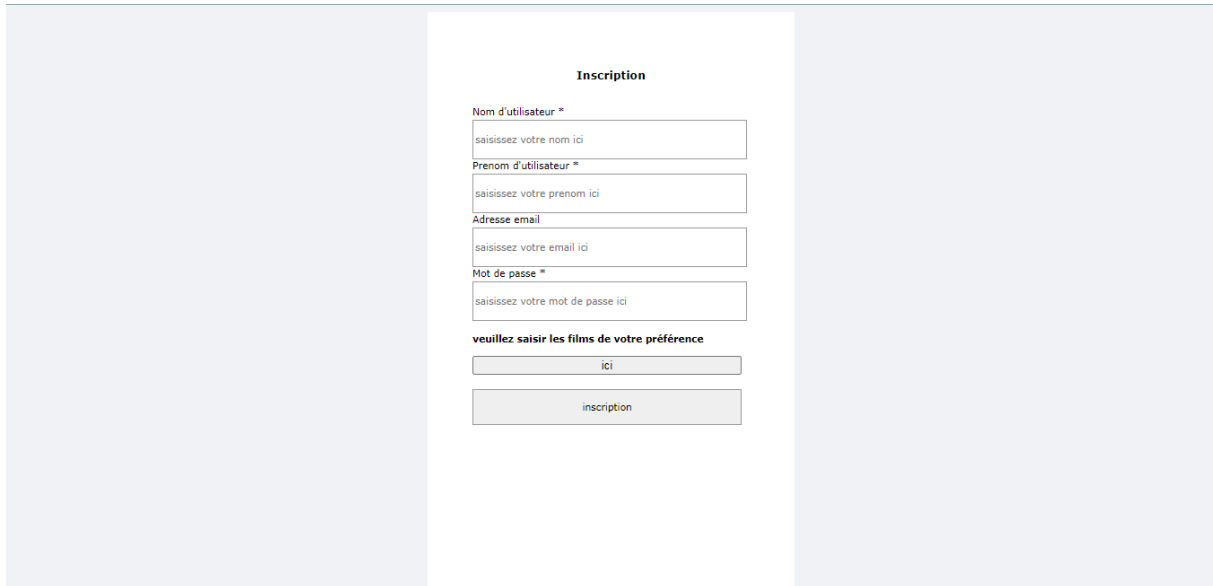
Figure 5.3. Interface de connexion avec Facebook

5.3.2.3. Interface d'inscription

Une fois l'utilisateur a cliqué sur le bouton « inscription » de l'interface accueil, il sera redirigé dans l'interface d'inscription.

Ici, l'utilisateur peut s'inscrire en saisissant son « nom » et « prénom », « adresse mail », « mot de passe », et il doit saisir au moins un film de sa préférence.

Toutes ces informations seront enregistrées dans la base de données.



The screenshot shows a registration form titled "Inscription". It contains the following fields and elements:

- Nom d'utilisateur ***: Input field with placeholder "saisissez votre nom ici".
- Prénom d'utilisateur ***: Input field with placeholder "saisissez votre prenom ici".
- Adresse email**: Input field with placeholder "saisissez votre email ici".
- Mot de passe ***: Input field with placeholder "saisissez votre mot de passe ici".
- veuillez saisir les films de votre préférence**: A section with a list box containing "ici" and a button labeled "inscription".

Figure 5.4. Interface d'inscription

5.3.2.4. Interface de la fenêtre principale

Après inscription et vérification d'email, l'utilisateur peut à présent se connecter.

En se connectant, l'utilisateur va être redirigé à la fenêtre principale. Dans cette partie, on affiche le nom et le prénom de l'utilisateur et un bouton qui va afficher les films suggérés à cet utilisateur.

Bienvenue

nina nina

afficher les suggestions

[Logout](#)

Figure 5.5. Interface de la fenêtre principale

Une fois l'utilisateur accède à la fenêtre principale, s'il clique sur le bouton afficher les suggestions, un ensemble de films lui sera afficher comme dans la figure 5.6.

Liste des films

Toy Story
Jumanji
Tom and Huck
GoldenEye
Balto
Cutthroat Island
City of Lost Children
Mortal Kombat
Lamerica
White Squall

Figure 5.6. Interface d'un exemple de résultat affiché à l'utilisateur

5.4. Conclusion

Au long de ce dernier chapitre, nous avons présenté l'étape réalisation de notre application ainsi, nous avons présenté les outils logiciels qui nous ont permis la réalisation de notre travail à savoir l'environnement de développement et les langages de programmation.

Ensuite, nous avons présenté notre application en décrivant ses fonctionnalités et tu présenté plusieurs interfaces.

Conclusion générale

Les systèmes de recommandation automatique sont devenus, à l'instar des moteurs de recherche, un outil incontournable pour tout site Web focalisé sur un certain type d'articles disponibles dans un catalogue riche, que ces articles soient des objets, des produits culturels (livres, films, morceaux de musique, etc.), des éléments d'information (news) ou encore simplement des pages (liens hypertextes).

L'objectif de ces systèmes est de sélectionner, dans leur catalogue, les items les plus susceptibles d'intéresser un utilisateur particulier. [Rao and Talwar, 2008] ont répertorié un vaste ensemble de systèmes de recommandation pour différents domaines applicatifs, dans des contextes académiques et industriels.

Dans notre travail nous nous sommes intéressés à l'étape de construction du profil utilisateur pour remédier au problème du démarrage à froid dans un système de recommandation. L'objectif principal de ce système est de fournir le nombre maximal des recommandations pertinentes.

Nous avons présenté les différentes étapes de construction de notre profil ainsi que son implémentation pour le système de recommandation et on a vu l'intérêt que l'on peut avoir en mettant en place un filtrage basé sur le contenu et le filtrage hybride pour la recommandation des films.

Comme perspective à ce travail, Nous envisageons de :

- récupérer les préférences de l'utilisateur de différents réseaux sociaux, tels que Twitter...etc.
- appliquer cette solution sur différents types d'items tels que musique, livre...etc.

Bibliographie

A

- [ADOMAVICIUS et TUZHILIN, 2005] ADOMAVICIUS, G. and TUZHILIN, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6) :734–749.
- [ADOMAVICIUS et al., 2011] Adomavicius, G. and Tuzhilin, A. (2011). Context-aware recommender systems. In *Recommender systems hand book*, pages 217–253. Springer.
- [ARALE et al, 2013] Aral et al. (2013). tie strength embeddedness and social influence.

B

- [BELIAKOV et al., 2011]. Beliakov, G., Calvo, T., & James, S. (2011). Aggregation of preferences in recommender systems. In *Recommender systems handbook* (pp. 705-734). Springer, Boston, MA.
- [BENOURET, 2015] Idir Benouaret, “ Un système de recommandation sensible au contexte pour la visite de musée”, in proceedings of the CORIA ; Conférence en Recherche d’Information et Applications – 12th French Information Retrieval Conference, Paris, 2015.
- [BENOURET, 2018]. Idir Benouaret. Un système de recommandation contextuel et composite pour la visite personnalisée de sites culturels. Autre [cs.OH]. Université de Technologie de Compiègne, 2017. Français. NNT : 2017COMP2332. tel-01767997
- [BILLSUS et PAZZANI, 1999] Daniel Billsus et al. (1999). Daniel Billsus , Michael J. Pazzani. An hybrid user model for news story classification. daniel billsus et al. (1999). Daniel Billsus and Michael J. Pazzani , An hybrid user model for news story classification.
- [BOBADILLA et al., 2013] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez. 2013. Recommender systems survey. *Knowledge-Based Systems* 46: 109–132. <https://doi.org/10.1016/j.knosys.2013.03.012>

[BOUZEGHOUB et KOSTADINOV, 2005] BOUZEGHOUB M., KOSTADINOV D., « Personnalisation de l'information : Aperçu de l'état de l'art et définition d'un modèle flexible de définition de profils », *Actes de la 2nde Conférence en Recherche d'Information et Applications CORIA*, 2005, p. 201-218.

[BOUZEKRI, 2017] Rapport de stage de Elodie Bouzekri « Conception, spécification et validation d'un système de recommandation pour la gestion des alarmes dans les cockpits d'avion civils »

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

[BRUSILOVSKY et EKLUND, 1998] BRUSILOVSKY P., EKLUND L « Web-based Education for all: A tool for development adaptive courseware. Seventh International World Wide Web Conference. Computer Networks and ISDN Systems» (30): 291-300

[BUDZIK *et al.*, 2000] J. Budzik et K.J Hammond. User interactions with every applications as context for just-in-time information access. In *Proceedings of the 5th international conference on intelligent user interfaces*, pages 44–51, Mars 2000.

C

[CARMEL *et al.*,2009] CARMEL, D., N. ZWERDLING, I. Guy, S. OFEK-KOIFMAN, N. HAR'EL, I. RONEN, *et al.* (2009). Personalized Social Search Based on the User's Social Network. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, p. 1227–1236. New York, NY, USA: ACM.

[CHALLAM, 2004] CHALLAM V., « Contextual information retrieval using ontology based user profiles », *Master of science in computer science*, Jawaharlal Nehru Technological University, 2004.

[CHEN, CHEN, 2015] Chen G., Chen L. (2015, août). Augmenting service recommender systems by incorporating contextual opinions from user reviews. *User Modeling and User-*

Adapted Interaction, vol. 25, no 3, p. 295–329. Consulté sur <https://doi.org/10.1007/s11257-015-9157-3>

[CHO *et al.*, 2002]. Cho Y. H., Kim S.H., Kim J.K. (2002). A personalized recommender system based on web usage mining and decision tree induction, *ESA*, 23(3), pp. 329-342.

[CLAYPOOL *et al.*, 1999] Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D. and Sartin, M., (1999). Combining Content-Based and Collaborative Filters in an Online Newspaper. SIGIR '99 Workshop on Recommender Systems: Algorithms and Evaluation. Berkeley, CA.

D

[DALIA, 2014]. Dalia Sulieman. (2014). Systèmes de recommandation sociaux et sémantiques. Thèse. Université de Cergy Pontoise.

[DUMAIS *et al.*, 2003] S. Dumais, E. Cuttrel, J.J. Cadiz, G. Jancke, R. Sarin et D.C Robbins. Stuff i've seen: A system for a personal information retrieval and reuse. In *Proceedings of the 26th ACM SIGIR International Conference on Research and Development*, pages 72–79, 2003.

[DURAO, 2010], Durao, F., Dolog, et P. 2010. Extending a Hybrid Tag-Based Recommender System with Personalization. In: Proc. 2010 ACM Symposium on Applied Computing, SAC 2010, Sierre, Switzerland, p. 1723–1727.

E

[EKSTRAND *et al.*, 2011]. Ekstrand, M. D., Riedl, J. T., and Konstan, J. A. (2011). Collaborative filtering recommender systems. *Foundations and trends in Human-Computer Interaction*.

[ESSLIMANI, 2010]. Esslimani, I. 2010. Vers une approche comportementale de recommandation : apport de l'analyse des usages dans le processus de personnalisation. Thèse de doctorat en informatique. Université de Nancy 2, France.

G

[GABRIELSSON et GABRIELSSON, 2006]. Gabrielsson, S. & Gabrielsson, S. 2006. The use of Self-Organizing Maps in Recommender Systems, A survey of the Recommender Systems field and a presentation of a State of the Art Highly Interactive Visual Movie Recommender System. Mémoire de master, Uppsala University.

[GAO et al, 2010] Gao et al. (2010). personnalisation in web computing and informatics theories techniques applications and future research.

[GAUCH et al, 2003] Gauch et al. (2003). Jason Chaffee, and Alexander Pretschner. Ontology-based personalized search and browsing.

[GENTILI et *al.*, 2003] G.Gentili, A.Micarelli, and F.Sciarrone. Infoweb: An adaptive information filtering system for the cultural heritage domain. Applied Artificial Intelligence (AAI), 17(8-9) :715–744, 2003.

[GOOD *et al.*, 1999] Good N., Schafer J. B., Konstan J. A., Borchers A., Sarwar B., Herlocker J., Riedl J., (1999). Combining collaborative filtering with personal agents for better recommendations, In Proc. of the 6th Nat. Conf. on AI, Orlando, USA, pp. 439- 446.

[GOWAN, 2003] Gowan J., «A multiple model approach to personalised information access», Master thesis in computer science, Faculty of science, Université de College Dublin, February 2003.

[GRACIA et al, 2015] Gracia et al. (2015). Data processing in data mining.

H

[HERNANDEZ-RUBIO et al., 2018] Hernández-Rubio M., Cantador I., Bellogín A. (2018, novembre). A comparative analysis of recommender systems based on item aspect opinions extracted from user reviews. *User Modeling and User-Adapted Interaction*. Consulté sur <https://doi.org/10.1007/s11257-018-9214-9>

[HUANG et al., 2004]. Huang, Z., Chen, H., & Zeng, D. (2004). Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering. *ACM Transactions on Information Systems (TOIS)*, 22(1), (pp. 116-142).

[HYOUNG et al., 2003] Hyoung R et al. (2003). Hyoung R ,Kim and Philip K. Chan. Learning implicit user interest hierarchy for context in personalization.

J

[JAMALI et ESTER, 2009]. Mohsen Jamali and Martin Ester. Using a trust network to improve top-n recommendation. In *Proceedings of the third ACM conference on Recommender systems, RecSys '09*, page 181–188, New York, NY, USA, 2009. ACM.

[JANNACH et al., 2010] Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. *Recommender Systems: An Introduction*. Cambridge University Press, New York, NY, USA, 1st edition, 2010.

K

[KELLY et al., 2011] Kelly et al. (2011). KELLY D., TEEVAN J , Implicit feed back for inferring user preference.

[KITTS et al., 2000]. Kitts, B., Freed, D. and Vrieze, M. (2000). Cross-sell: a fast promotion-tunable customer-item recommendation method based on conditionally independent probabilities. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 437–446. ACM.

[KOSTADINOV, 2008] D.Kostadinov. Personnalisation de l'information : une approche de gestion de profils et de reformulation de requêtes. Thèse de Doctorat en informatique présentée à l'Université de Versailles Saint-Quentin- Yvelines, France, Septembre 2008.

[KRULWICH, 1997] Krulwich B., (1997). Lifestyle Finder: Intelligent user profiling using large-scale demographic data, *AI Magazine*, 18(2), p. 37-45.

[KRULWICH et BURKEY, 1996]. Krulwich, B. et Burkey, C. 1996. Learning user information interests through extraction of semantically significant phrases. In Proceedings of the AAAI Spring Symposium on Machine Learning in Information Access. Stanford, CA.

L

[LIN, 2005] LIN C., XUE G., ZENG H., YU Y., « Using probabilistic latent semantic analysis for personalised Web search », *Proceedings of the APWeb Conference*, 2005, p. 707-711.

[LINDEN et al., 2003] Linden et al. (2003). Linden, G., Smith, B., and York, J. (2003). Amazon.com Recommendations.

[LIU et YU, 2004] LIU F., YU C., « Personalized Web search for improving retrieval effectiveness », *IEEE Transactions on knowledge Data Engineering*, vol. 16, 2004, p. 28-40.

[LIU, 2007] Liu. (2007). exploring hyperlinks content and usage data.

[LIREN CHEN et al, 1998] .pondération des termes.

[LOPS et al., 2011]. Lops, P., De Gemmis, M., and Semeraro,G.(2011).Content-based recommender systems: State of the art and trends. In *Recommender systems handbook* , pages 73–105.Springer.

M

[MAATALLAH, 2016]. Majda MAATALLAH : Une Technique Hybride pour les Systèmes de Recommandation. Université Badji Mokhtar –Annaba-2016.

[MARKO BALABANOVIC, 1997] marko balabanovic. (1997). An adaptative web page recommendation service. content-based, collaborative recommendation.

[MC GOWAN, 2003] MC GOWAN J., « A multiple model approach to personalised information access », *Master thesis in computer science*, Faculty of science, University college Dublin, February 2003.

[MELVILLE *et al.*, 2002] P. Melville, R. J. Mooney, and R. Nagarajan. Content-boosted collaborative filtering for improved recommendations. In *Proceedings of the National Conference on Artificial Intelligence*, page 187–192, 2002.

[MICRO SPERETTA *et al.*, 2004] Micro speretta *et al.* (2004). Mirco Speretta and Susan Gauch. personalizing search based on user search histories.

[MIDDLETON *et al.* 2002] Middleton S. E., Alani H., Shadbolt N. R., De Roure C., (2002). Exploiting Synergy Between Ontologies and Recommender Systems. *Proc. of the 11th Inter. WWW Conf., Workshop on the Semantic Web.*

[MIDDLETON *et al.*, 2004] Middleton S. E., Shadbolt N. R., De Roure D. C., Ontological user profiling in Recommender systems, *ACM Transactions on Information Systems*, 22(1), pp.54-88.

[MIQUEL MONTANER *et al.*, 2003] Miquel montaner *et al.* (2003). Miquel Montaner, Beatriz López, and Josep Lluís De La Rosa, A taxinomy of recommender agents of the internet.

[MOONEY *et al.*, 2000] Mooney R.-J., Roy L., Content-based book recommending using learning for text categorization, *Proceedings of the 5th ACM Conference on Digital Libraries (DL '00)*, USA, 2000, p. 195-204.

[MOBASHER *et al.*, 2003].Bamshad Mobasher, Xin Jin, and Yanzan Zhou. Semantically enhanced collaborative filtering on the web. In *Proceedings of the First European Web Mining Forum -EWMF 2003*, pages 57-76. Springer, 2003.

N

[NANAS et al., 2003] NANAS N., UREN U., DEROECK A., « Building and applying a concept hierarchy representation of a user profile », *Proceedings of the 26th Annual ACM Conference on Research and Development in Information Retrieval SIGIR*, 2003, p. 154-204.

[NEGRE, 2015]. Elsa Negre. Systèmes de recommandation – Introduction. ISTE, 2015.K

[NGUYEN et al., 2006a] Nguyen A., Denos N., Berrut C., (2006). Exploitation des données "disponibles à froid" pour améliorer le démarrage à froid dans les systèmes de filtrage d'information, in *INFORSID '06*, pp. 81-95.

[NGUYEN et al., 2006b] Nguyen A., Denos N., and Berrut C., (2006). Modèle des espaces de communautés orienté vers la diversité des recommandations pour les systèmes de filtrage, *Conférence en recherche information et applications No3*, Lyon, France, 6(2), pp.125-150.

[NGUYEN et al., 2006]. Nguyen A. and Denos N. and Berrut C., (2006). Modèle d'espaces de communautés basé sur la théorie des ensembles d'approximation dans un système de filtrage hybride, Conf. en Recherche Information et Applications (CORIA), Lyon, France, pp. 303-314.

O

[OARD et KIM ,2001] Feedback implicite pour les systèmes de recommandation. Dans l'atelier AAAI sur les systèmes de recommandation, Madison, WI: 81-83.

P

[PAZZANI, 1997] Pazzani M., (1999). A framework for collaborative, content-based and demographic filtering, *Artificial Intelligence Review*, 13(5), pp. 393-408.

[PAZZANI et BILLSUS, 2007] Pazzani M., Billsus D., (2007). Content-based Recommendation Systems. In *The Adaptive Web: Methods and Strategies of Web Personalization*, LNCS, Vol. 4321. Springer.

[PAZZANI et BILLSUS, 2000]. Billsus, D., Pazzani, M. (2000): User modeling for adaptive news access *user-modeling and user-adapted interaction*, 10(2-3), 147-180.

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

R

[RASHID, KARYPIS et RIEDL, 2008] *Al Mamunnur Rashid, George Karypis, and John Riedl. SIGKDD Workshop on Web Mining and Web Usage Analysis (WEBKDD), 2008*

[RAZMERITA, 2003] RAZMERITA . classification du procédé d'acquisition de données implicites.

[RICH, 1989] RICH E. (1989). Stereotypes and user modeling. user models in dialog systems. A. Kobsa & W. Wahlster. Berlin, Springer verlag: 35-51.

[ROBERTSON et SPARCK JONES, 1998] document retrieval systems.

[ROZA LEMDANI, 2016]. Roza Lemdani. Système hybride d'adaptation dans les systèmes de recommandation. Autre. Université Paris-Saclay, 2016. Français. ffNNT : 2016SACL050ff. fftel-01371650

S

[SAADIA MALIK, 2005] SAADIA MALIK. weighting of terms

[SARWAR *et al.*, 2001]. Sarwar B. M., Karypis G., Konstan A., and Riedl J., (2001). Item-based collaborative filtering recommendation algorithms. *In Proc. of the 10th Inter. Conf. on WWW*, pp. 285–295.

[SCHEIN *et al.*, 2001] Andrew I Schein, Alexandrin Popescul, Lyle H Ungar, and David M Pennock. Generative models for cold-start recommendations. In Proceedings of the 2001 SIGIR Workshop on Recommender Systems, volume 6. Citeseer, 2001.

[SIEG *et al.*, 2004] Sieg A., Mobasher B., Burke R., Prabu G., and Lytinen S., «Using concept hierarchies to enhance user queries in web-based information retrieval, », *In The IASTED International conference on artificial intelligence and applications*, 2004.

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

[SOLLENBORN *et* FUNK, 2002]. Sollenborn, M. *et* Funk, P. 2002. Category-based filtering and user stereotype cases to reduce the latency problem in recommender systems. In Proceedings of the 6th European Conference on Advances in Case-Based Reasoning (ECCBR'02), p. 395–420, London, UK. Springer-Verlag.

[SMYTH *et* COTTER, 2000]. Barry Smyth and Paul Cotter. A personalised tv listings service for the digital tv age. *Knowl.-Based Syst.*, 13(2-3):53-59, 2000.

T

[TEBRI *et al.*, 2005] Tebri *et al.* (2005). H. Tebri, M. Boughanem, and C. Chrisment. Incremental profile learning based on a reinforcement method.

V

[VILLANOVA, 2002] VILLANOVA, la méthode d'acquisition explicite.

W

[WASFI, 1999] Wasfi A.-M., Collecting User Access Patterns for Building User Profiles and Collaborative Filtering, *Proceedings of the 4th International Conference on Intelligent User Interfaces (IUI'99)*, California, USA, 1999, p. 57-64.

[WEN et al., 2004] WEN J., LAO N., MA W. Y., « Probabilistic model for contextual retrieval », *Proceedings of the 27th annual international ACM SIGIR Conference on Research and development in Information retrieval*, August 2004, p. 57-63.

[WEN et al., 2008]. Weng L. T., Xu Y., Li Y., Nayak R., (2008). Exploiting Item Taxonomy for Solving Cold-Start Problem in Recommendation Making. In *the 20th IEEE Inter. Conf.*, Vol. 2, pp. 113-120.

[WENG, 2008]. Weng, L. T., Xu, Y., Li, Y., et Nayak, R. 2008. Exploiting Item Taxonomy for Solving Cold-Start Problem in Recommendation Making. *ICTAI* (2), p. 113-120.

Y

[YOUNG WOO et al, 2000] young woo seo et al. (2000). Young-Woo Seo and Byoung-Tak Zhang, A reinforcement-learning agent for personalized information filtering.

Z

[ZEMIRLI, 2008] N.Zemirli. Modèle d'accès personnalisé à l'information basé sur les diagrammes d'influence intégrant un profil utilisateur évolutif. Thèse de Doctorat en informatique présentée à l'Université Paul Sabatier de Toulouse III, Juin 2008.

[ZHANG et al, 2010] al, Z. e. (2010). ZHANG Z.-K., LIU C., ZHANG Y.-C., ZHOU T, solving the cold start problem in recommender system with social tags.