

# Table des matières

---

Introduction générale .....	3
Chapitre 1 : .....	6
1.1. Introduction.....	7
1.2. Historique.....	7
1.3. Définition des systèmes de recommandation .....	8
1.4. Objectif des systèmes de recommandation .....	9
1.5. Approches de recommandation classiques .....	9
1.5.1. Recommandation basée sur le contenu .....	9
1.5.2. Recommandation basée sur le filtrage collaboratif .....	10
1.5.3. Recommandation basé sur le filtrage hybride .....	12
1.5.4. Recommandation communautaire .....	13
1.5.5. Recommandation basé sur le filtrage démographique.....	14
1.5.6. Recommandations à base de connaissances .....	15
1.6. Forces et faiblesses des systèmes de recommandations classiques.....	15
1.7. Conclusion .....	18
Chapitre 2.....	19
Systèmes de Recommandations Sensibles au Contexte .....	19
2.1. Introduction.....	20
2.2. Définition du contexte .....	20
2.3. Catégorisation du contexte .....	22
2.5. Paradigmes pour l'intégration du contexte dans les systèmes de recommandation .....	23
2.5.1. Pré-filtrage Contextuel (contextual pre-filtering) .....	23
2.5.2. Post-filtrage Contextuel.....	24
2.5.3. La modélisation contextuelle.....	24
2.6. Qu'est-ce qu'un système sensible au contexte (Context-awareness) .....	25
2.7.1. Micro-profiling.....	26
2.7.2. Sourcetone Interactive Radio - Musiques selon l'humeur .....	26
2.8. Conclusion .....	27
Chapitre 3.....	28
Utilisation de l'approche pré-filtrage contextuel des CARS.....	28
3.1. Introduction.....	30
3.2. Architecture de l'approche proposée.....	30
3.3. Conception du système .....	34
3.3.1. Diagramme de cas d'utilisation .....	34

## Table des matières

---

3.3.2.	Diagramme de séquence .....	34
3.3.3.	Diagramme de classe.....	36
3.4.	Collection de données .....	36
3.5.	Environnement de développement.....	37
3.6.	Description de l'outil développé.....	38
3.7.	Conclusion .....	41
	Conclusion générale et perspectives .....	42
	Références bibliographiques.....	44
	Liste des figures.....	47
	Liste des tableaux.....	48
	Liste des abréviations.....	49
	Résumé .....	50

# **INTRODUCTION**

## **GENERALE**

# Introduction générale

---

## Contexte

Avec l'avènement d'internet, nous assistons aujourd'hui à une surcharge d'informations énorme. Cette croissance exponentielle des données se traduit par une difficulté à organiser et à analyser ces informations brutes mais ouvrant de nouvelles voies sur les chemins de la connaissance.

La question n'est donc plus de disposer de l'information, mais de trouver l'information pertinente au bon moment. Delà, les systèmes de recommandation sont nés.

Depuis les années 90, les systèmes de recommandation (SRs) émergent comme un domaine de recherche indépendant de la recherche documentaire et de l'intelligence artificielle pour résoudre le problème de surcharge d'information en recommandant des items (qui peuvent être un film, un livre ou un endroit à visiter ...) appropriés à un utilisateur .

Les SRs utilisent des profils représentant des préférences des utilisateurs pour calculer des recommandations. Ce calcul se fait par la prédiction des notes (ou votes) qu'un utilisateur est susceptible d'attribuer aux items. Les approches de recommandation se basent, généralement, sur le filtrage basé sur le contenu (FBC), le filtrage collaboratif (FC) ou sur une combinaison de ces deux approches.

L'approche FBC consiste à recommander à l'utilisateur des items qui correspondent à son profil ; ils utilisent l'appariement entre le profil d'un utilisateur et les descripteurs des contenus.

L'approche FC consiste à recommander à l'utilisateur des items choisis (ou bien notés) par les utilisateurs similaires. Cela suppose que les utilisateurs ayant des intérêts communs, dans le passé, continueront, probablement, à partager les mêmes goûts dans le futur.

## Problématique et objectif

Les SRs dites SRs traditionnels essaient de modéliser le degré d'utilité d'un item à un utilisateur qu'il ne l'a pas déjà évalué par rapport aux items notés dans le passé.

Cependant, ces systèmes ignorent que ces notes attribués peuvent être différentes en allant d'un contexte à un autre. En effet, l'endroit où se trouve l'utilisateur (à la maison ou au travail), le temps (le matin ou le soir), la saison, l'état d'esprit peuvent influencer sur les votes des utilisateurs, et ainsi améliorer l'exactitude des recommandations.

Les systèmes de recommandation sensibles au contexte (les CARS : ContextAware Recommender Systems) apportent une solution à ce problème. Sachant que récemment, les

## **Introduction générale**

---

entreprises ont commencé à intégrer certaines informations contextuelles dans leurs moteurs de recommandation.

Trois approches principales dans lequel les informations contextuelles peuvent être incorporées dans les systèmes de recommandation à savoir : Pré-filtrage Contextuel (contextual pre-filtering), Post-filtrage Contextuel (contextual post-filtering), Modélisation Contextuelle (contextual modeling).

L'objectif de notre travail est d'incorporer l'information contextuelle en appliquant l'approche de Pré-filtrage Contextuel pour la recommandation des films. La particularité de cette approche est qu'elle se fait avant que la méthode principale de recommandation soit lancée sur le reste des données sélectionnées.

L'information contextuelle sera utilisée comme une étiquette permettant de filtrer les évaluations qui ne correspondent pas à notre information contextuelle. Ensuite nous allons lancée sur la base de données réduite la méthode de recommandation filtrage collaboratif afin d'obtenir les recommandations liées au contexte.

Nous utilisons le processus UP du langage de modélisation UML pour la modélisation de notre système et la base de données FILM de MovieLens<sup>1</sup> qui fournit les notes données par des utilisateurs à certains films dont nous allons intégrer la notion du contexte.

## **Organisation**

Ce mémoire comporte trois chapitres qui se répartissent comme suit :

Le premier chapitre, offre un survol des systèmes de recommandation. Ce survol présente l'histoire des systèmes de recommandation, suivie de leur définition, puis les différentes approches de recommandations qui existent ainsi que leurs avantages et inconvénients.

Dans le deuxième chapitre, nous commençons par définir la notion de contexte. Nous enchaînons en montrant les différentes méthodes de son introduction dans le processus de recommandation. Nous terminons en présentant quelques SR sensibles au contexte.

Et enfin dans le troisième chapitre nous allons présenter notre conception ainsi que les principaux outils utilisés pour la mise en place de notre application ainsi que quelques captures d'écran de notre interface graphiques.

---

<sup>1</sup>URL: <http://www.movielens.umn.edu>

# Chapitre 1

## Les systèmes de recommandation classiques

### 1.1. Introduction

Aujourd'hui, la quantité d'informations stockées sur les ordinateurs et échangées sur les réseaux de télécommunication informatiques double tous les jours.

Cette croissance exponentielle se traduit par une difficulté à organiser et à analyser ces informations mais elle ouvre de nouvelles voies sur les chemins de la connaissance.

La question n'est donc plus de disposer de l'information, mais de trouver l'information pertinente au bon moment. Une des approches adoptées pour aider l'utilisateur à avoir l'information qui l'intéresse sans perte de temps consiste à fournir des systèmes de recommandation qui le guide lors de son exploration, en cherchant pour lui les informations qui paraissent pertinentes. D'après Chris Anderson dans « The Long Tail », les bouleversements qu'a subi le Web et la masse de données qui constituent Internet font que « nous quittons progressivement l'âge de l'information pour rentrer dans l'âge de la recommandation » [1].

Dans ce chapitre, nous commençons par donner un historique des systèmes de recommandation puis nous allons définir ce qu'est un système de recommandation, ainsi les différentes approches de recommandations et on finira par exploiter leurs avantages et inconvénients.

### 1.2. Historique

Dans le but de personnaliser la recherche d'information dans un domaine d'application particulier, il fallait collecter, sélectionner, classifier puis suggérer à l'utilisateur les informations qui répondent vraisemblablement à ses intérêts à long terme. D'où l'apparition des systèmes de filtrage d'information ou systèmes de recommandation (SR). Ces derniers sont considérés comme un filtre de flux entrant d'informations de façon personnalisée pour chaque utilisateur.

« Information Lens System » [2] peut être considérée comme le premier système de recommandation. À l'époque, l'approche la plus commune pour le problème de partage d'informations dans l'environnement de messagerie électronique était la liste de distributions basée sur les groupes d'intérêt.

Par la suite, avec l'essor de l'Internet et des applications Web, il y a eu un engouement pour les systèmes de recommandation qui se sont développés dans différents domaines d'applications. Nous pouvons en citer :

- Les systèmes de recommandation de films, tels que : Moviens [3] et Eachmovie [4].
- Les systèmes de recommandation de livres (Bookcrossing).

- Les systèmes de recommandation de musique (LastFM).
- Les systèmes de recommandation d'articles d'actualités.
- Les systèmes de recommandations introduits sur des sites e-commerce (Amazon).
- Les systèmes de recommandation de restaurants [5].
- Les systèmes de recommandations intégrés aux moteurs de recherche (le moteur de recherche d'AOL).

Pour tous les systèmes de recommandation développés jusqu'à nos jours, la collecte de données relatives aux utilisateurs et/ou aux items, représente une phase clé dans le processus de personnalisation.

### 1.3. Définition des systèmes de recommandation

Les systèmes de recommandation peuvent être définis de plusieurs façons, qui peuvent se rapporter à différents types de données ou approches spécifiques. La définition que nous utiliserons dans ce mémoire est une définition générale de Robin Burke [5], que nous avons traduite :

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

Les deux entités de base qui apparaissent dans tous les systèmes de recommandations sont l'utilisateur et l'item. L'«**usager**» est la personne qui utilise un système de recommandation, donne son opinion sur diverses items et reçoit les nouvelles recommandations du système. L'«**Item**» est le terme général utilisé pour désigner ce que le système recommande aux usagers.

Les données d'entrée pour un système de recommandation dépendent du type de l'algorithme de filtrage employé. Généralement, elles appartiennent à l'une des catégories suivantes :

- **Les estimations** :(également appelées les votes), expriment l'opinion des utilisateurs sur les articles (exemple : 1 mauvais à 5 excellent).
- **Les données démographiques** : se réfèrent à des informations telles que l'âge, le sexe, le pays et l'éducation des utilisateurs. Ce type de données est généralement difficile à obtenir et est normalement collecté explicitement.
- **Les données de contenu** : qui sont fondées sur une analyse textuelle des documents liés aux éléments évalués par l'utilisateur. Les caractéristiques extraites de cette analyse sont utilisées comme entrées dans l'algorithme de filtrage afin d'en déduire un profil d'utilisateur. [6]



### 1.4. Objectif des systèmes de recommandation

L'objectif est à la fois de minimiser le temps des utilisateurs passé à la recherche, mais aussi de leur suggérer des ressources pertinentes qu'ils n'auraient pas spontanément consultées et ainsi accroître leurs satisfaction globale.

### 1.5. Approches de recommandation classiques

#### 1.5.1. Recommandation basée sur le contenu

Le filtrage basé sur le contenu (*Content-based Filtering*), s'appuie sur le contenu des documents (thèmes abordés) pour les comparer à un profil lui-même constitué de thèmes. Chaque utilisateur du système possède alors un profil qui décrit ses propres centres d'intérêt. Par exemple, le profil peut contenir une liste des thèmes ou préférences que l'utilisateur aime bien ou qu'il n'aime pas. Lors de l'arrivée d'un nouveau document, le système compare le descriptif du document avec le profil de l'utilisateur pour prédire l'utilité de ce document pour cet utilisateur.

##### 1.5.1.1. Avantages

L'avantage est qu'ils permettent d'associer des documents à un profil utilisateur. Notamment, en utilisant des techniques d'indexation et d'intelligence artificielle. L'utilisateur est indépendant des autres ce qui lui permet d'avoir des recommandations même s'il est le seul utilisateur du système.

Afin de recommander par exemple des films à un utilisateur, le système analyse les corrélations entre ces films et les films consultés antérieurement par cet utilisateur. Ces corrélations sont évaluées en considérant des attributs comme le titre et le genre. De ce fait, parmi ces films, ceux qui seront recommandés à l'utilisateur, sont les plus similaires (En termes d'attribut) aux films consultés par cet utilisateur.

##### 1.5.1.2. Limites

- L'effet "entonnoir" : les besoins de l'utilisateur sont de plus en plus spécifiques, ce qui l'empêche d'avoir une diversité de sujets. Même pire, un nouvel axe de recherche dans un domaine bien précis peut ne pas être pris en compte car il ne fait pas parti du profil explicite de l'utilisateur.
- Filtrage basé sur le critère thématique uniquement, absence d'autres facteurs comme la qualité scientifique, le public visé, l'intérêt porté par l'utilisateur, etc.

- Les difficultés à recommander des documents multimédia (images, vidéos, etc.) et ceci à cause de la difficulté à indexer ce type de documents, c'est en fait la même problématique dont souffrent les systèmes de recherche.
- Problème de démarrage à froid : Un nouvel utilisateur du système éprouve des difficultés à exprimer son profil en spécifiant des thèmes qu'ils intéressent. Ceci malgré les techniques d'apprentissage ou l'utilisateur fournit des textes exemples. [7]

### 1.5.2. Recommandation basée sur le filtrage collaboratif


































Contrairement à la recommandation basée sur le contenu, le filtrage collaboratif se base sur la construction profil d'utilisateur qui sera créé au fur et à mesure de l'évaluation de l'utilisateur, il permet de filtrer n'importe quel type de données (image, texte, vidéo). Le but du filtrage collaboratif est de suggérer de nouveaux items ou de prédire l'utilité d'items inconnus pour un utilisateur donné, en se fondant sur les évaluations déjà exprimées. Il utilise des méthodes statistiques pour faire des prévisions basant sur la corrélation entre son profil personnel et les profils des autres utilisateurs qui présentent des intérêts semblables.

Un système de filtrage collaboratif est organisé comme suit :

- Collecter les appréciations et le comportement des utilisateurs, en général l'utilisateur fournit des évaluations sous forme de notes, sur un ou plusieurs axes : qualité, correspondance au besoin, etc.
- Intégrer ces informations au profil de l'utilisateur.
- Le système utilise ces informations pour faire des recommandations.

La figure I.1 représente un tableau de films avec sur un axe les utilisateurs d'un même Système (ex: un groupe d'amis sur MovieLens) et sur un autre les films. Chaque cellule de la matrice contient l'avis donné par un utilisateur pour un film, la cellule vide signifie qu'il n'a pas d'avis particulier sur ce film.

Afin de prédire si Illyes apprécierait le film "Harry Potter" et probablement lui recommander ce film, on compare les Votes d'Illyes à ceux des autres utilisateurs choisis. On peut alors voir que Illyes et Amel ont des Votes identiques, et que Amel n'a pas aimé le film «Harry Potter», on pourrait alors prédire que Illyes n'aimera pas aussi ce film et de ne lui pas faire cette suggestion.

	 Mohamed	 Haecne	 Anel	 Mourad	 Ilbes
 The Piano					
 MI3					
 Rocky5					
 CliffHanger					
 Harry Potter					

**Figure I.1 :** Exemple de recommandation basé sur le filtrage collaboratif.

Dans un système du filtrage collaboratif, il faut que les utilisateurs fournissent des évaluations des items qu'ils ont déjà utilisés, sous forme des notes, pour constituer leurs profils.

### 1.5.2.1. Avantages

- Ce type de système est très efficace dans le cas où le contenu des objets sont complexes, il est compliqué ou impossible de l'analyser car Il n'y a aucune analyse du sujet ou du contenu des objets à recommander
- l'utilisateur est capable de découvrir divers domaines intéressants, car le principe du filtrage collaboratif ne se fonde absolument pas sur la dimension thématique des profils, et n'est pas soumis à l'effet « entonnoir ».

Les jugements de valeur des utilisateurs intègrent non seulement la dimension thématique mais aussi d'autres facteurs relatifs à la qualité des items tels que la diversité, la nouveauté, l'adéquation du public visé, etc.

### 1.5.2.2. Limites

Sa performance dépend beaucoup de la distribution des évaluations (notes) données par utilisateurs.

Si dans le système il y a plusieurs items qui ont été utilisés et évalués par très peu d'utilisateurs, ces items seraient recommandés très rarement, même si ces utilisateurs ont

donné des notes très hautes pour ces items. Ce problème est connu comme le problème de Parcimonie (sparsity problem).

- Si dans le système il existe des utilisateurs qui ont des goûts très différents en comparaison avec les autres, le système ne peut pas trouver des similarités entre utilisateurs et donc ne peut pas donner des bonnes recommandations.

L'exploitation des données disponibles dans un système de filtrage peut se faire de plusieurs manières. Ces méthodes sont classées en deux familles principales : les algorithmes bases mémoire est les algorithmes bases modèle. [7]

### 1. Filtrage collaboratif basé sur la mémoire

On distingue deux type d'algorithmes à base de mémoire : à base d'utilisateur et à base d'item.

Les algorithmes à base d'utilisateur, utilise la totalité d'informations (évaluations des utilisateurs) disponible pour faire des prédictions. Ces prédictions sont faites par rapport aux usagers les plus proches voisins(PPV), pour choisir ces PPV on doit calculer la similarité entre les usagers en utilisant généralement deux mesures qui sont : la similarité vectorielle et la corrélation de Pearson.

Les algorithmes basés sur item calculent la similarité entre les items, l'item quel usager a évalué et l'item cible. Bien que ce type d'algorithmes est très simple à mettre en œuvre et donne des prédictions de bonnes qualités, la forte complexité combinatoire les empêche d'être utilisés dans les environnements de production. [7]

### 2. Filtrage collaboratif basé sur un modèle

Afin de palier au problème de la complexité des algorithmes à base de mémoire [4], Breese et al ont proposés des approches à base de modèles où l'idée générale est d'obtenir un modèle de données hors ligne pour prédire des estimations en ligne aussi vite que possible. Pour la construction de ce modèle plusieurs méthodes ont été proposées en se basant sur les techniques d'apprentissage automatique et les techniques statistiques tels que le clustering, réseaux des bayes, etc. [7]

#### 1.5.3. Recommandation basé sur le filtrage hybride

Constatant les avantages et inconvénients (voir tableau I.1) des différents type de recommandation, cités ci-dessus, Plusieurs approches ont été proposés, ce sont généralement une des combinaisons des deux classifications : **recommandation à base de contenu, recommandation collaborative.**

En général, l'hybridation s'effectue en deux phases :

- Appliquer séparément le filtrage collaboratif et autres techniques de filtrage pour générer des recommandations candidates.
- Combiner ces ensembles de recommandations préliminaires selon certaines méthodes telles que la pondération, la mixtion, la cascade, la commutation, etc., afin de produire les recommandations finales pour les utilisateurs.

Plus généralement, les systèmes hybrides gèrent des profils d'utilisateurs orientés contenu, et la comparaison entre ces profils donne lieu à la formation de communautés d'utilisateurs permettant le filtrage collaboratif. [7]

### 1.5.4. Recommandation communautaire

La recommandation communautaire ou comme on l'appelle souvent recommandation sociale, vu que la plupart des réseaux sociaux (facebook, twitter, etc) s'appuient sur cette classification dans leurs recommandations (figure I.2).

L'idée de base est donc de dire que si des utilisateurs ont partagés des mêmes intérêts dans le passé, il y a de fortes chances qu'ils partagent aussi les mêmes goûts dans le futur.

Le système propose des recommandations à partir des relations de l'utilisateur avec ces amis dans le réseau social, et parfois cette recommandation dépend aussi de la valeur de confiance d'utilisateur dans chacun de ses amis, l'exemple le plus connu de cette recommandation est la section des pages et des groupes qui apparaît dans la partie droite d'une page facebook. L'importance décisionnelle du bouton « I Like » de Facebook a donné un succès croissant dont 55% des utilisateurs sont influencés par leurs amis. [7]



Figure I.2 : Recommandation communautaire.

### 1.5.5. Recommandation basé sur le filtrage démographique

C'est une recommandation simple qui propose des items par rapport au profil démographique d'utilisateur (figure I.3). Elle comporte à partager les usagers en plusieurs classes ou groupes par rapport aux informations démographiques telles que le sexe, l'âge, la profession, la localisation, la langue, le pays, etc.

Le principe de cette approche est que deux utilisateurs ayant évolué dans un environnement similaire partagent des goûts communs que deux utilisateurs ayant évolué dans des environnements différents et ne partageant donc pas les mêmes codes.

De nombreux sites utilisent cette solution simple à proposer une offre de contenu "personnalisé". Par exemple, les utilisateurs sont redirigés vers un site Web particulier en fonction de leur langue ou de pays. Ces approches ont été très populaires dans la littérature de marketing, mais ont reçu peu d'attention dans le domaine des algorithmes de recommandation. [7]

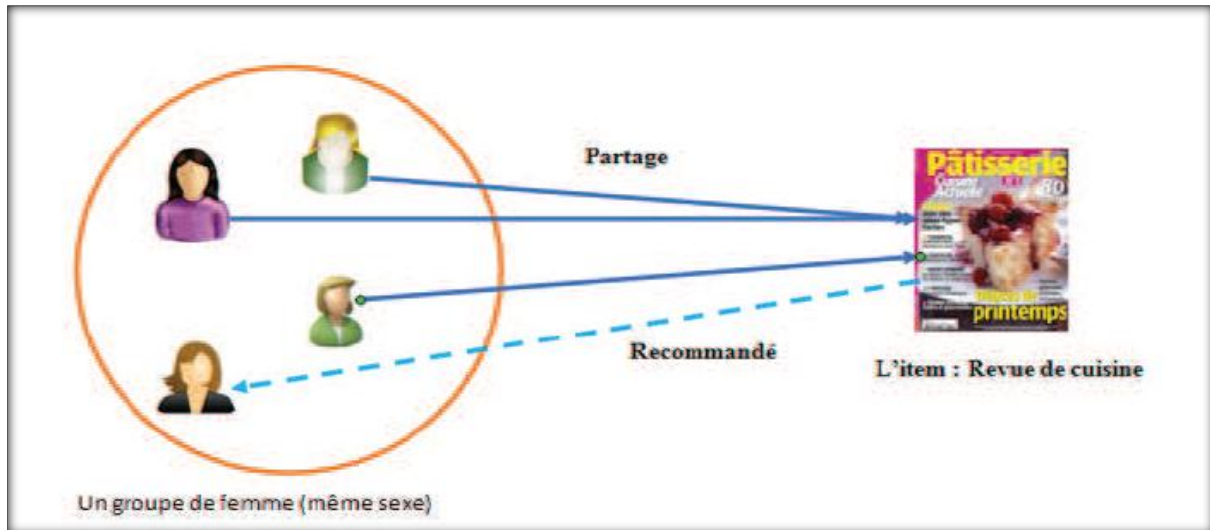


Figure I.3 : Recommandation basé sur le filtrage démographique.

### 1.5.6. Recommandations à base de connaissances

Les recommandations sont générées en utilisant des connaissances spécifiques dont certaines caractéristiques d'items répondent aux préférences de l'utilisateur (figure I.4).

Généralement les Systèmes à base de connaissances sont plus fiables par rapport à d'autres types de recommandation si les données limitées sont disponibles, à savoir, si le système ne peut pas compter sur l'existence d'un historique de l'utilisateur.

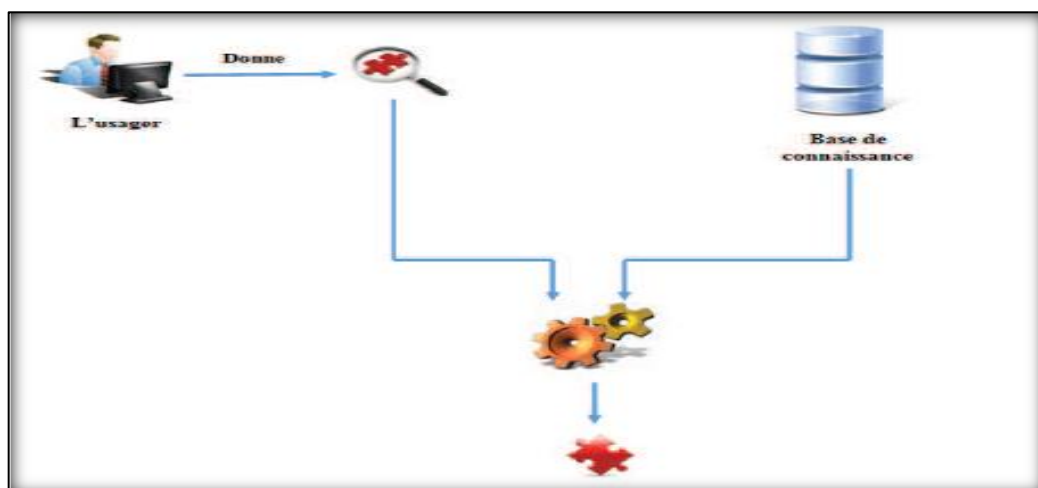


Figure I.4: Recommandation à base de connaissance.

## 1.6. Forces et faiblesses des systèmes de recommandations classiques

Les problèmes les plus rencontrés dans les systèmes de recommandation sont :

- **Adaptabilité:** Au fur et à mesure que la base de données des évaluations augmente, la recommandation devient plus précise.
- **Nouvel utilisateur:** Un nouvel utilisateur qui n'a pas encore accumulé suffisamment d'évaluations ne peut pas avoir de recommandations pertinentes.

- **Nouvel item:** Un item doit avoir suffisamment d'évaluations pour qu'il soit pris en considération dans le processus de recommandation.
- **Démarrage à froid:** Le démarrage à froid est un problème pour les nouveaux utilisateurs qui commencent à jouer avec le système, parce que le système ne dispose pas d'assez d'informations à leur sujet. Si le profil d'utilisateur est vide, il doit consacrer une somme d'efforts à l'aide du système avant d'obtenir une récompense (les recommandations utiles).

D'autre part, quand un nouvel item est ajouté à la collection, le système doit avoir suffisamment d'informations pour être en mesure de recommander cet item aux utilisateurs.

Le tableau I.1 résume les forces et faiblesses des méthodes traditionnelles utilisées par les systèmes de recommandation. [8]



Techniques	Avantages	Inconvénients
Filtrage démographique	<ul style="list-style-type: none"> <li>• N'exige aucun historique d'estimations d'utilisateur.</li> </ul>	<ul style="list-style-type: none"> <li>• Problème de confidentialité. Recommandation très générale.</li> <li>• Utilisateur avec un goût unique.</li> <li>• Nouvel item.</li> <li>• <b>Problème d'induction</b></li> </ul>
Filtrage communautaire	<ul style="list-style-type: none"> <li>• Adaptabilité : la qualité croît avec le nombre d'amis.</li> </ul>	<ul style="list-style-type: none"> <li>• Nouvel utilisateur.</li> <li>• Nouvel item.</li> <li>• <b>Problème d'induction</b></li> </ul>
Filtrage à base du contenu	<ul style="list-style-type: none"> <li>• Pas besoin d'une large communauté d'utilisateurs pour pouvoir effectuer des recommandations.</li> <li>• Une liste de recommandations peut être générée même s'il n'y a qu'un seul utilisateur.</li> <li>• La qualité croît avec le temps.</li> <li>• Pas besoin d'information sur les autres utilisateurs.</li> <li>• Possibilité de recommander de nouveaux items ou même des items qui ne sont pas populaires.</li> </ul>	<ul style="list-style-type: none"> <li>• L'analyse du contenu est nécessaire pour faire une recommandation.</li> <li>• Problème de recommandation des images et des vidéos en absence de Métadonnées et de contexte</li> <li>• Nécessité du profil d'utilisateur.</li> <li>• <b>Problème d'induction</b></li> </ul>
Filtrage collaboratif	<ul style="list-style-type: none"> <li>• Ne demande aucune connaissance sur le contenu de l'item ni sa sémantique.</li> <li>• La qualité de la recommandation peut être évaluée.</li> <li>• Plus le nombre d'utilisateurs est grand plus la recommandation est meilleure.</li> </ul>	<ul style="list-style-type: none"> <li>• Démarrage à froid.</li> <li>• Nouvel Item.</li> <li>• Nouvel utilisateur.</li> <li>• Problème de confidentialité.</li> <li>• La complexité : dans les systèmes avec un grand nombre d'items et d'utilisateurs, le calcul croît linéairement.</li> <li>• <b>Problème d'induction: le contexte change</b></li> </ul>

**Tableau I.1** : Les avantages et les inconvénients des techniques de recommandations.

### 1.7. Conclusion

Dans ce chapitre, nous avons d'abord, présenté l'historique et la notion des systèmes de recommandation, en détaillant les approches des systèmes de recommandation. À savoir : Filtrage basé sur le contenu, filtrage collaboratif et la recommandation hybride.

Ainsi les systèmes basés sur le filtrage démographique, les systèmes basés sur le filtrage à base de données communautaire et les systèmes à base de connaissances.

Finalement nous avons présenté les avantages et les inconvénients de chaque type de système de recommandation.

Parmi les inconvénients évoqués est le fait que ces systèmes ne tiennent pas compte du contexte dans lequel se trouve l'utilisateur lorsqu'il désire une recommandation. Les systèmes de recommandation sensibles au contexte (les CARS : Context Aware Recommender Systems) apportent une solution à ce problème. Nous les détaillons dans le chapitre suivant.

# Chapitre 2

## Systemes de Recommandations Sensibles au Contexte

### 2.1. Introduction

Les systèmes de recommandation jouent un rôle important dans la manipulation de grandes quantités d'informations. Souvent, le contenu et les objets qui pourraient intéresser une personne, dépendent de sa situation spécifique: l'emplacement actuel, la saison, le rôle de l'utilisateur, la température, appelés: *le contexte*. Les systèmes de recommandation sensibles aux contextes (les CARS : Context Aware Recommender Systems) « CARS »tentent d'exploiter l'utilisation du contexte afin d'améliorer le processus de génération des recommandations.

Dans la première partie de ce chapitre, nous commençons d'abord par donner une définition de la notion du contexte, sa catégorisation, son acquisition, sa modélisation, ainsi les différents paradigmes pour son intégration dans les systèmes de recommandation.

Ensuite en présente la définition des systèmes de recommandation sensibles aux contextes. Puis, nous terminons par quelques exemples sur des SRs sensibles au contexte.

### 2.2. Définition du contexte

Avant de présenter les définitions du contexte proposées dans la littérature scientifique, nous commençons par introduire et analyser des définitions proposées dans différents dictionnaires :

- *Techno-Science*<sup>2</sup>, encyclopédie scientifique en ligne : « *Le contexte d'un évènement inclut les circonstances et conditions qui l'entourent* ». En informatique, « *le contexte est l'ensemble des conditions sous lesquelles un dispositif est en train d'être utilisé, par exemple l'occupation actuelle de l'utilisateur* ».
- *Grand Dictionnaire Technologique*<sup>3</sup> (GDT) : « *Texte entourant un terme, ou situation dans laquelle ce terme est employé* ». En informatique, « *ensemble d'informations concernant l'action du stylet, en rapport principalement avec sa localisation à l'écran, qui permet au système d'exploitation de l'ordinateur à stylet de différencier les commandes et l'entrée des données, et de fonctionner en conséquence* ».
- *Centre National de Ressources Textuelles et Lexicales*<sup>4</sup> : « *Ensemble de circonstances liées, situation ou un phénomène apparaît, un évènement se produit* ». En informatique, « *le contexte est*

---

<sup>2</sup><http://www.techno-science.net/>

<sup>3</sup><http://gdt.oqlf.gouv.qc.ca/>

<sup>4</sup><http://www.cnrtl.fr/>

## Chapitre 2 : Systèmes de recommandation sensibles au contexte

---

*l'ensemble d'informations concernant un évènement en rapport principalement avec sa localisation, qui permet au système informatique de fonctionner et de s'adapter en conséquence ».*

Un point commun émerge de ces définitions. Elles décrivent toutes le contexte comme un ensemble d'informations associées à quelque chose « ensemble de ... qui entoure/constitue/concernant/liés/etc. quelque chose ». La nature de ce quelque chose (un évènement, un texte, une action, un phénomène, une situation, etc.) dépend de l'usage dans un domaine d'application (linguistique, informatique, etc.). Cet usage permet de spécifier comment le contexte est utilisé (modifier la valeur des éléments, différencier les commandes, ou fonctionner en conséquence) [9].

En effet, en informatique il est difficile de trouver une définition claire et unique, valable dans tous les travaux impliquant cette notion.

Par exemple, pour Schilit et Theimer ; [10], le contexte est *«la localisation et l'identité des personnes et objets à proximité ainsi que les modifications pouvant intervenir sur ces objets».*

Pour Ryan et al [11], il s'agit plutôt de *«la localisation, l'identité et l'heure».*

Dans un article plus poussé sur le sujet [12], Dey donne la définition suivante : *«toute information pouvant être utilisée pour qualifier la situation des entités [...], typiquement la localisation, l'identité et l'état des personnes, groupes et objets informatiques et physiques».*

La définition la plus vaste a été donnée par Schilit et al. Dans l'un des premiers articles à aborder le sujet [13]:

*« Contexte englobe plus que l'emplacement de l'utilisateur, parce que d'autres choses d'intérêt sont également mobile et changeant. Contexte comprend l'éclairage, le niveau de bruit, la connectivité réseau, les coûts de communication, la bande passante de communication, et même la situation sociale; par exemple, si vous êtes avec votre gestionnaire ou avec un collègue».*

Pour résumer, une information fait partie du contexte si elle influe sur une interaction entre deux entités : *« context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and application themselves»* (Dey, 2000) [14].

Cette dernière définition concerne particulièrement la conception des systèmes sensibles au contexte.

### 2.3. Catégorisation du contexte

Pour bien comprendre et appliquer cette définition du contexte, il est plus simple de passer par une catégorisation des variables du contexte. Selon Shilit et al. [15], le contexte se décompose en trois sous classes où chacune des variables répond à l'une des questions "Où suis-je ?", "Avec qui suis-je ?", "Quelles sont les ressources de mon environnement proche?". Ryan et al. [11] catégorisent le contexte en identité de l'utilisateur, ressources de l'environnement proche, localisation de l'utilisateur et période temporelle d'exécution de l'interaction.

On voit apparaître en filigranes de toutes ces catégories, comme le résumant Dey et al. [16], les notions de temps, d'identité, d'activité et de localisation. En fait toute variable contextuelle est utile pour éclaircir une ou plusieurs des questions "Quand ?, Où ?, Quoi ?, Qui ?". Le système ou les concepteurs du système vont utiliser ces informations catégorisées pour répondre au "Pourquoi ?" de l'occurrence d'une situation.

Un utilisateur équipé d'un système contextuel pour, par exemple une visite de ville, et qui va s'approcher d'un monument, va voir apparaître sur ce guide des informations sur le monument. Le concepteur du système répond à la question "Pourquoi l'utilisateur s'approche-t-il ?" en lui envoyant des informations sur le monument. La variable contextuelle utilisée est la localisation de l'utilisateur mais aussi son environnement, c'est à dire que le monument fait aussi partie du contexte puisqu'il influe sur la réponse du système. [17]

*Temps, identité, activité et localisation* sont des catégorisations primaires du contexte, ce sont les seules nécessaires pour expliciter une situation. Il est possible de les décomposer en sous-variables. De l'identité, on va pouvoir trouver l'adresse, le courriel et d'autres informations personnelles. Ces sous-catégories sont en fait considérées comme des attributs des catégories primaires : le numéro de téléphone d'une personne peut être connu en passant par son identité dans un carnet de numéros. L'utilisation des catégories et des attributs donne une première idée de l'organisation que peuvent avoir des variables du contexte.

### 2.4. Obtention du contexte

Une question cruciale est de savoir comment obtenir les informations contextuelles riches. Dans certaines circonstances, le contexte est explicite, comme une personne qui communique au système de recommandation les films qu'elle désire regarder.

D'autre part, l'information contextuelle peut également être déduite par le système. Par exemple, le temps qui peut être déduit à partir de l'horloge du système ou la localité de l'utilisateur qui peut être déduite à partir de son GPS. Il peut être déduit aussi en observant le comportement d'une personne qui a acheté un objet à partir d'un site Web e-commerce, pendant un jour de semaine ou un week-end. On peut également utiliser des informations spécifiques au domaine, qui sont généralement collectées à partir du CMS (Système de Gestion du Contenu).

Par exemple, si un élément représente un accès à une musique, le genre de la musique peut être utilisé comme une dimension de l'information contextuelle. [18]

### 2.5. Paradigmes pour l'intégration du contexte dans les systèmes de recommandation

La contextualisation peut intervenir à différents endroits dans un système de recommandation. Il existe trois différents paradigmes algorithmiques pour l'intégration des informations contextuelles dans le processus de recommandation sont présentés: [18] [19] [20].

1. L'approche de Pré-filtrage Contextuel (*contextual pre-filtering*),
2. L'approche de Post-filtrage Contextuel (*contextual post-filtering*),
3. L'approche de la Modélisation Contextuelle (*contextual modeling*).

#### 2.5.1. Pré-filtrage Contextuel (contextual pre-filtering)

Dans l'approche de pré-filtrage, l'information contextuelle est utilisée comme une étiquette permettant de filtrer les évaluations qui ne correspondent pas à l'information contextuelle spécifiée. Cela se fait avant que la méthode principale de recommandation soit lancée sur le reste des données sélectionnées.

Autrement dit, si un contexte d'intérêt particulier est  $k$ , alors cette méthode sélectionne à partir de la série initiale toutes les évaluations relatives au contexte spécifié  $k$ , et elle génère la matrice «articles x utilisateur» ne contenant que les données relatives au contexte  $k$ .

Après cela, la méthode des systèmes de recommandation (SR), comme le filtrage collaboratif, est lancée sur la base de données réduite afin d'obtenir les recommandations liées au contexte  $k$ . La figure II.1 montre l'intégration du contexte dans un système de recommandation à deux dimensions (utilisateur x produit). [18]

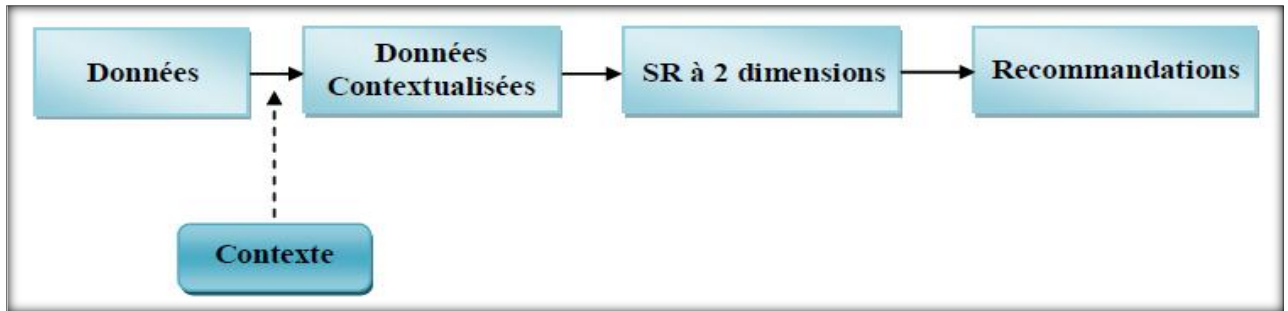


Figure II.1 : Pré-filtrage Contextuel.

### 2.5.2. Post-filtrage Contextuel

Selon l'approche post-filtrage, l'information contextuelle est utilisée après le lancement de la méthode principale de recommandation 2D (à deux dimensions). Une fois les évaluations inconnues sont estimées et les recommandations sont produites, le système analyse les données pour un utilisateur donné dans un contexte précis pour trouver les modèles d'utilisation des articles spécifiques, et utilise ces modèles pour «contextualiser» les recommandations obtenues à partir de la méthode classique de recommandation 2D, comme le filtrage collaboratif.

La figure II.2 montre comment utiliser le contexte après la recommandation (2D). [18]

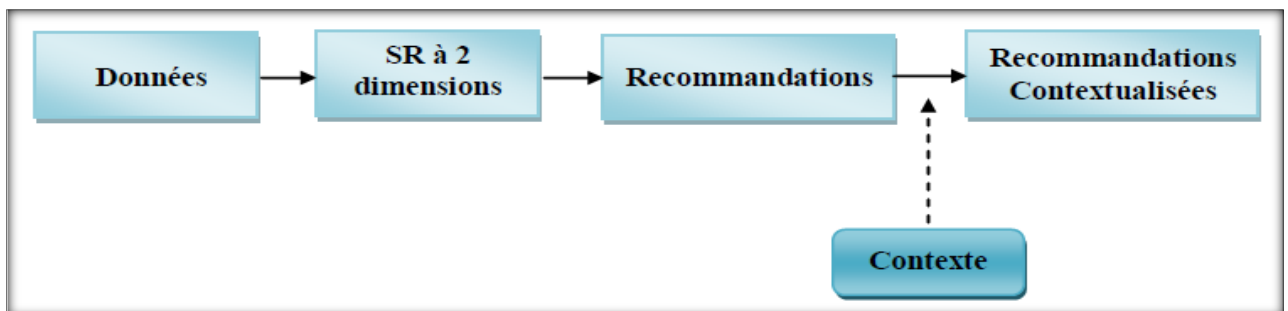


Figure II.2 : Post-filtrage Contextuel.

### 2.5.3. La modélisation contextuelle

En plus des deux méthodes citées auparavant, il existe une troisième méthode à savoir la modélisation contextuelle dans laquelle l'information contextuelle est utilisée directement à l'intérieur des algorithmes de génération des recommandations. La figure II.3 montre l'intégration du contexte dans le processus de recommandation. [18]



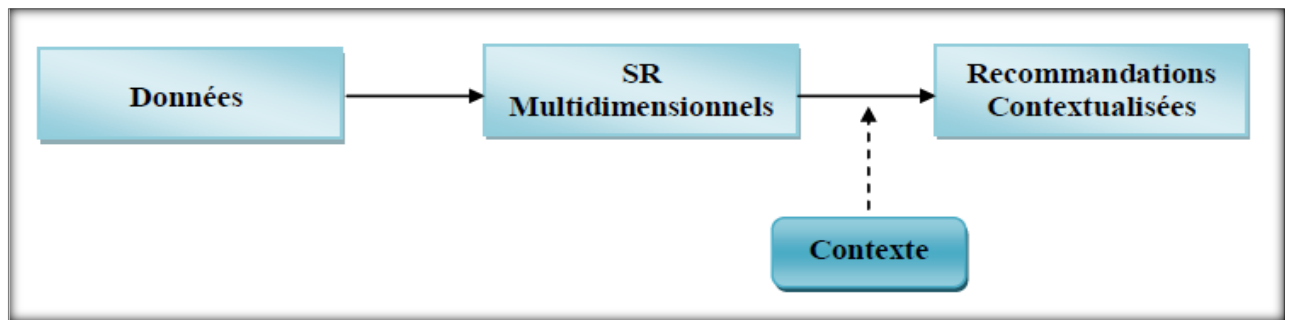


Figure II.3 : La modélisation contextuelle

### 2.6. Qu'est-ce qu'un système sensible au contexte (Context-awareness)

On dit d'un système qu'il est sensible au contexte s'il peut tirer, interpréter et utiliser des informations issues du contexte et adapter sa réponse en fonction du contexte d'utilisation.

On fait maintenant le distinguo entre les applications qui utilisent le contexte, comme par exemple un service de météo qui aura besoin d'informations de localisation et de temps pour produire un bulletin et d'autres applications qui adaptent leur comportement en fonction du contexte.

L'utilisation simple du contexte n'implique pas une modification du comportement du système. Salbert et al. [21] définissent la sensibilité au contexte comme étant la meilleure capacité d'un système à agir en temps réel avec des données provenant du contexte. Dey limite sa définition à l'interface homme machine sans prendre en compte l'application en elle-même.

D'autres définitions sont plus orientées vers l'adaptation au contexte : Brown [22] dit d'une application sensible au contexte qu'elle doit automatiquement extraire de l'information ou effectuer des actions en fonctions du contexte utilisateur détecté par les capteurs.

Enfin Dey et Abowd [16] proposent qu'un système soit sensible au contexte s'il utilise des informations du contexte pour mettre à disposition des informations ou des services utiles à l'utilisateur, l'utilité dépendant de la tâche de l'utilisateur. Avec cette définition, une application qui se contenterait d'afficher à l'écran le contenu des variables de contexte est une application sensible au contexte légitime, même si elle ne modifie pas son comportement.

Ces définitions s'articulent toutes autour de l'aptitude d'un dispositif à adapter son comportement en se basant sur le contexte de l'utilisateur. Un système sensible au contexte est un ensemble de mécanismes destinés pour la collection et la gestion des informations de contexte, et le contrôle du comportement du système en fonction de ces informations.

### 2.7. Exemples de Système de Recommandation sensibles au contexte

Nous présentons, dans cette section, quelques systèmes de recommandation sensibles au contexte.

#### 2.7.1. Micro-profiling

L'approche micro-profiling a pour but de recommander des chansons/artistes inconnus à un utilisateur. Le type du système de recommandation est le **pre-filtering** car les données sont filtrées selon le contexte (le temps : moment de la journée, jour, mois ou année).

Dans cette approche, supposant que les goûts des utilisateurs changent selon une période et peuvent être similaires dans une même période.

Par exemple, un utilisateur préfère écouter un certain genre de chanson quand il travaille, et un autre genre de chanson avant de dormir.

Ce profil sera décomposé de façon à ce qu'il s'adapte à chaque contexte : dans cette approche, le temps représente le moment de la journée, il sera divisé en plusieurs segments en fonction des chansons que l'utilisateur écoute le plus durant chaque moment (segment de temps) de la journée. [18]

#### 2.7.2. Sourcetone Interactive Radio - Musiques selon l'humeur

En 2004, Malcolm Goodman et le Dr Jeff Berger ont fondé un système de recommandation qu'ils ont nommé **Sourcetone** radio interactive ([www.sourcetone.com](http://www.sourcetone.com)), ce système demande à l'auditeur de préciser son état d'esprit ou émotionnel (le contexte) avant de lui recommander une chanson.

La figure II.4 montre une recommandation de la chanteuse « LadyGaga » pour un utilisateur qui a un état d'esprit « agité ». [18]



Figure II.4 : Imprime écran du site [www.sourcetone.com](http://www.sourcetone.com)

### 2.7.3. Amazon

Un autre système de recommandation sensible au contexte est celui d'**Amazon** qui a été créé par Jeff Bezos en juillet 1995 et dont la filiale française a ouvert en 2000.

**Amazon.com** est une entreprise de commerce électronique américaine basée à Seattle. Sa spécialité la plus connue est la vente de livres, mais elle est diversifiée dans d'autres produits, notamment dans la vente de tout type de produits culturels : disques CD, musique en téléchargement, DVD, appareils photos numériques, matériels informatique et électroménager, etc.

Ce système demande aux utilisateurs de se connecter via leur nom (voir figure II.5.A) afin de créer leur profile et fournit un bouton « trouver un cadeau » (voir figure II.5.B) pour chaque utilisateur afin de faire la distinction entre les préférences propres à l'utilisateur et les préférences de la personne à laquelle il va offrir ce cadeau. [18]



Figure II.5 : Barre d'outils du site www.amazon.com

## 2.8. Conclusion

Ce chapitre présente le concept de la sensibilité au contexte, un aspect fondamental dans l'informatique ubiquitaire. Nous avons présenté un point de vue sur la définition de la sensibilité au contexte. Nous avons étudié aussi un large éventail d'exemples sur des SRs sensibles au contexte.

La sensibilité au contexte est un élément central de plusieurs systèmes. Bien que le domaine de l'informatique contextualisé soit déjà en pleine expansion, de plus en plus de capteurs prolifèrent partout dans le monde, sa croissance va s'accélérer et son importance ne fera que croître.

Le prochain chapitre sera consacré à la partie conception et réalisation de notre système de recommandation sensible au contexte.

# Chapitre 3

## Utilisation de l'approche pré- filtrage contextuel des CARS

### 3.1. Introduction

La prise en compte du contexte par les systèmes de recommandation permet une meilleure modélisation du comportement des utilisateurs, en fournissant des informations supplémentaires qui peuvent être exploitées afin d'effectuer de meilleures recommandations.

Ce chapitre est consacré à la partie conception et réalisation de notre application, qui consiste à recommander des films à un utilisateur donné. Nous commençons par proposer l'architecture et la conception de notre système, ensuite nous présentons l'environnement de développement ainsi les outils utilisés. Nous incluons des captures d'écran pour illustrer certaines fonctionnalités de notre système.

### 3.2. Architecture de l'approche proposée

Nous avons opté pour l'utilisation de la méthode de pré-filtrage (définie au chapitre 2) pour l'introduction du contexte dans notre système de recommandation. Dans cette approche, l'information contextuelle est utilisée afin de filtrer les évaluations qui ne correspondent pas au contexte courant de l'utilisateur. Le pré-filtrage est exécuté avant que le module principal de la recommandation ne soit lancé.

Dans notre étude on s'est basé sur l'approche pré-filtrage contextuel qui consiste à exécuter le module principale de la recommandation après avoir récupéré le contexte (jour, nuit) de l'utilisateur, et sur le filtrage collaboratif basé mémoire (corrélation de Pearson) pour mesurer le taux de similarité entre les utilisateurs puis générer la liste des films à recommander.

Comme le montre la figure III.1, l'architecture globale de notre système de recommandation comporte trois modules principaux.

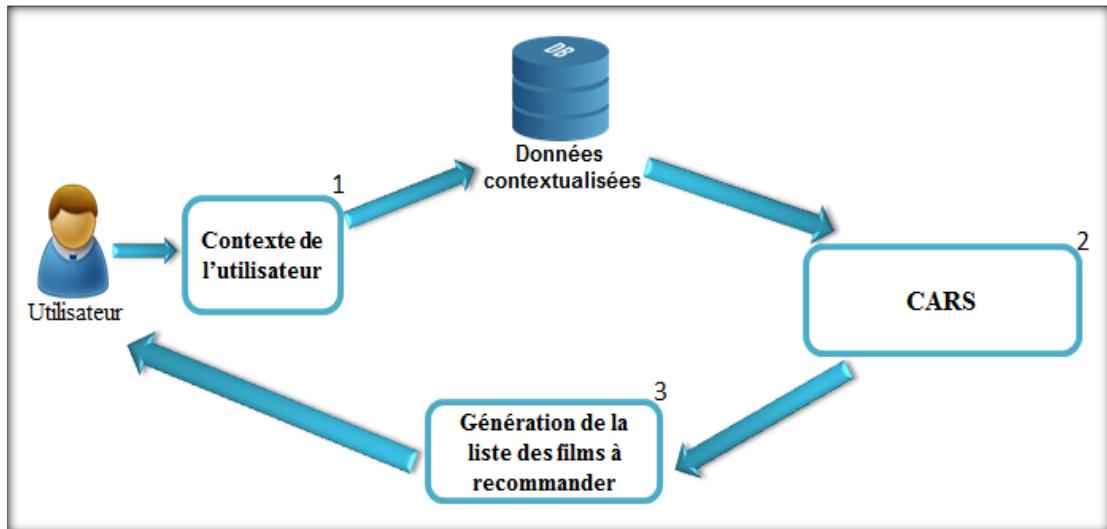


Figure III.1 : Architecture générale de l'application.

- 1. Requête utilisateur :** consiste à saisir la requête de l'utilisateur, à savoir le genre du film demandé par l'utilisateur, et le contexte (Film regardé le JOUR, Film regardé la NUIT).
- 2. Etablissement de taux de similarité :** ce module sert à trouver les recommandations les plus pertinentes. Nous avons pu déterminer la similarité entre les utilisateurs  $i$  et  $j$  qui ont les mêmes données géographique (âge et sexe), en utilisant la corrélation de Pearson (PCC).

$$\omega(a, i) = \frac{\sum_j (v_{a,j} - \bar{v}_a)(v_{i,j} - \bar{v}_i)}{\sum_j (v_{a,j} - \bar{v}_a)^2 \sum_j (v_{i,j} - \bar{v}_i)^2}$$

$v_{a,j}$ : Vote de  $a$  pour film  $j$ .

$v_{i,j}$ : Vote de  $i$  pour le film  $j$ .

$\bar{v}_i$ : La moyenne de l'ensemble des évaluations fournies par l'utilisateur  $i$ .

$\omega(a, i)$  : Le coefficient de pondération liant l'utilisateur actif et  $i$ .

- 3. Génération de la liste des films recommandés :** pour cette étape notre système de recommandation retourne une liste de films évalués par les usagers selon un taux de similarité qui existe entre les utilisateurs, déterminé dans l'étape précédente.

### 3.2.1. Exemple pour illustrer l'algorithme de Pearson

Nous proposons un exemple très simple de la matrice de notes du tableau qui représente trois utilisateurs pouvant voter quatre films.

A partir des notes qu'ils ont attribuées aux films, nous évaluons les similarités de notes entre ces utilisateurs.

	Film1	Film2	Film3	Film4
Utilisateur1	7	-	8	7
Utilisateur2	6	5	-	2
Utilisateur3	-	2	2	5

**Tableau III.1** : Matrice Films-Utilisateurs

Chaque utilisateur peut être vu comme étant un « vecteur » incomplet, on peut limiter notre analyse aux seules composantes qu'ils ont en commun.

Par exemple, pour comparer utilisateur 2 et utilisateur 3, on omet les valeurs manquantes où les items que l'utilisateur 2 et l'utilisateur 3 n'ont pas en commun. Nous avons alors les deux vecteurs suivants :  $v=(5,2)$  et  $w=(2,5)$ .

La corrélation de Pearson est une mesure de la similarité entre deux vecteurs :

$$Pearson = \frac{v - \bar{v} \cdot w - \bar{w}}{v - \bar{v} \cdot v - \bar{v} * w - \bar{w} \cdot w - \bar{w}}$$

Où le produit scalaire (.) est calculé avec les films que les deux utilisateurs sont en commun.

La valeur de Pearson est entre -1 et 1. Si cette valeur est égale à -1, nous disons que les deux utilisateurs sont fortement corrélés opposés. Si la valeur est égale à 1, les deux utilisateurs sont fortement corrélés semblables. Les deux utilisateurs sont considérés être indépendants si la valeur est égale à 0.

Dans le cas d'utilisateur (2) :

$$\bar{v} = \frac{(6 + 5 + 2)}{3} = 4.33$$

Et d'utilisateur (3) :

$$\bar{w} = \frac{(5 + 2 + 2)}{3} = 3$$

Si on ne retient que les composantes correspondantes aux items 2 et 4, les vecteurs sont donc :

$$(5 - 4.33, 2 - 4.33) = (0.67, 2.33) \text{ et } (2 - 3, 5 - 3) = (-1, 2)$$

On a donc :

$$Pearson = \frac{(0.67, -2.33) \cdot (-1, 2)}{\sqrt{(0.67, -2.33) \cdot (0.67, -2.33)} \cdot \sqrt{(-1, 2) \cdot (-1, 2)}} = \frac{-5.33}{5.43} = -1$$

En d'autres termes, l'utilisateur (2) et l'utilisateur (3) sont des « contraires » : l'un préfère l'item 4 et l'autre préfère l'item 2.

Pour calculer la corrélation entre l'utilisateur 1 et l'utilisateur 2, on observe d'abord que la moyenne d'utilisateur 1 est 7.33. On a donc :

$$Pearson = \frac{(1.67, -2.33) \cdot (-0.33, -0.33)}{\sqrt{(1.67, -2.33) \cdot (1.67, -2.33)} \cdot \sqrt{(-0.33, -0.33) \cdot (-0.33, -0.33)}} = \frac{-0.22}{1.33} = -0.16$$

Donc l'utilisateur (2) et l'utilisateur (1) sont peu corrélés.

A travers cet exemple nous avons pu montrer comment peut-on calculer la similarité entre deux utilisateurs en utilisant la corrélation de Pearson.

Dans ce qui suit, nous allons présenter les outils utilisés ainsi que toutes les étapes de conception de notre application allant de la modélisation UML jusqu'à l'implémentation du système.



### 3.3. Conception du système

Nous avons choisi le processus UP pour mettre en évidence notre démarche.

UML « Unified Modeling Language » qui signifie « Langage de modélisation unifié ». C'est un langage visuel constitué d'un ensemble de schémas, appelés des diagrammes, qui donnent chacun une vision différente du projet à traiter. UML nous fournit donc des diagrammes pour représenter le logiciel à développer : son fonctionnement, sa mise en route, les actions susceptibles d'être effectuées par le logiciel, etc. [23]

Nous commencerons dans un premier temps par tracer le diagramme de cas d'utilisations qui nous aidera à définir les diagrammes de séquence pour générer le diagramme des classes représentés.

#### 3.3.1. Diagramme de cas d'utilisation

Les diagrammes de cas d'utilisation représentent toutes les interactions des utilisateurs avec le système.

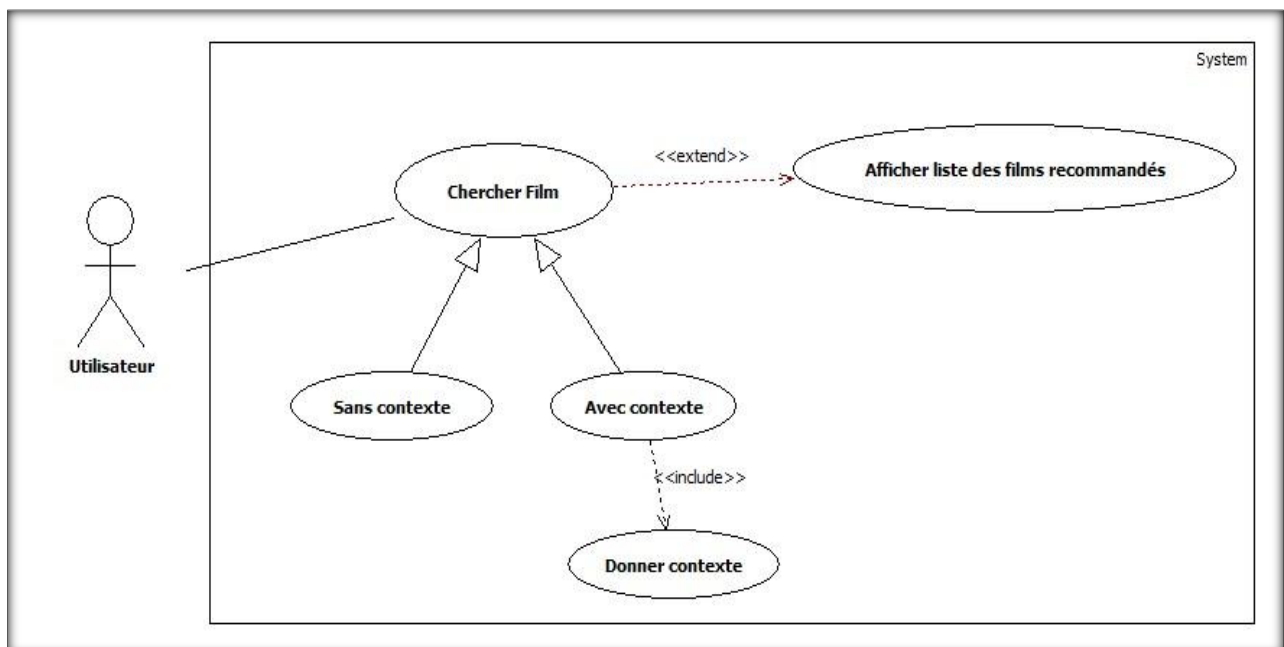


Figure III.2 : Diagramme de cas d'utilisation.

#### 3.3.2. Diagramme de séquence

Montre la séquence verticale des messages passés entre objets au sein d'une interaction.

- **Recommandation avec contexte**

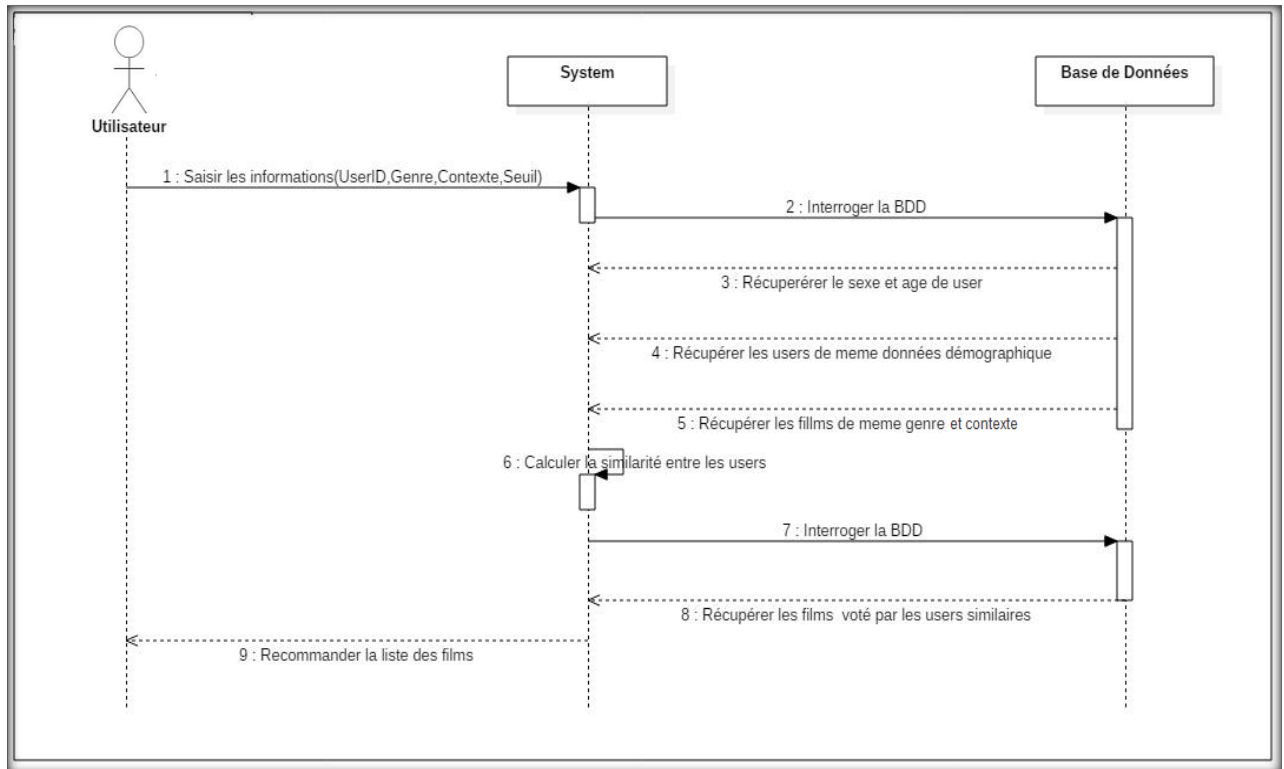


Figure III.3 : Diagramme de séquence du cas : Recommandations avec contexte.

- **Recommandation sans contexte**

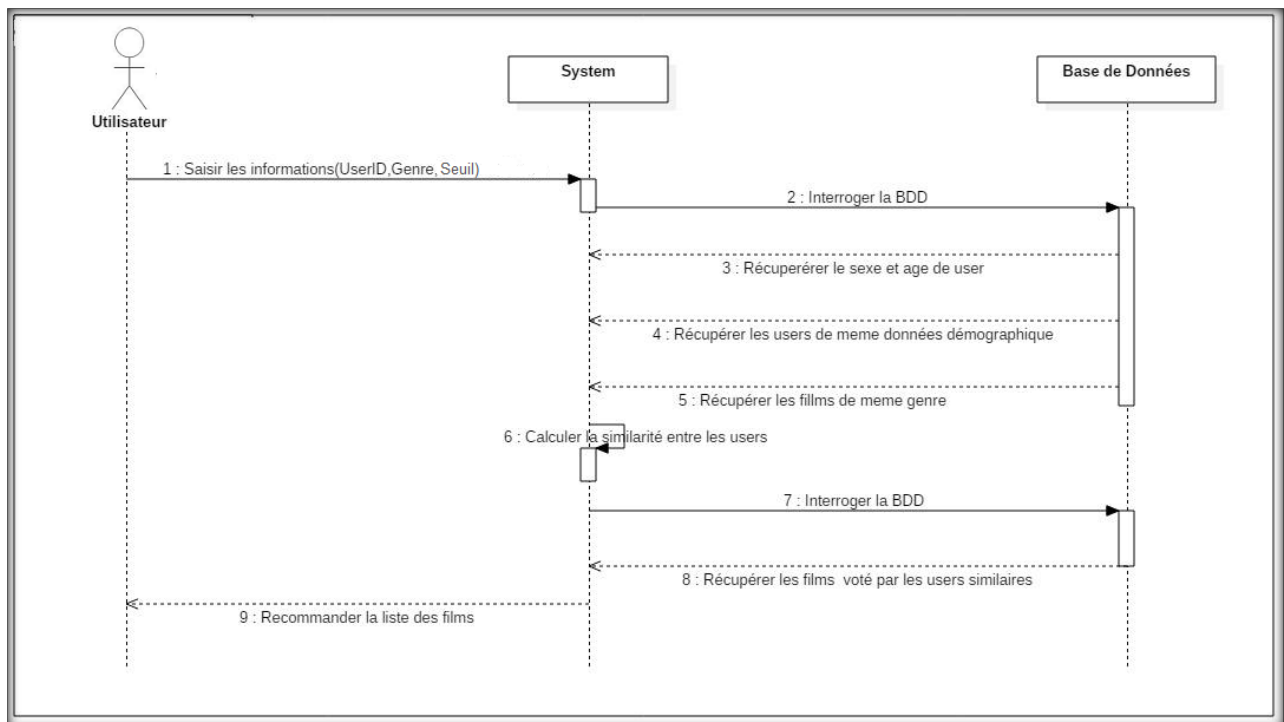


Figure III.4 : Diagramme de séquence du cas : Recommandations sans contexte.

### 3.3.3. Diagramme de classe

Le diagramme de classes représente les classes constituant le système et les associations entre Elles.

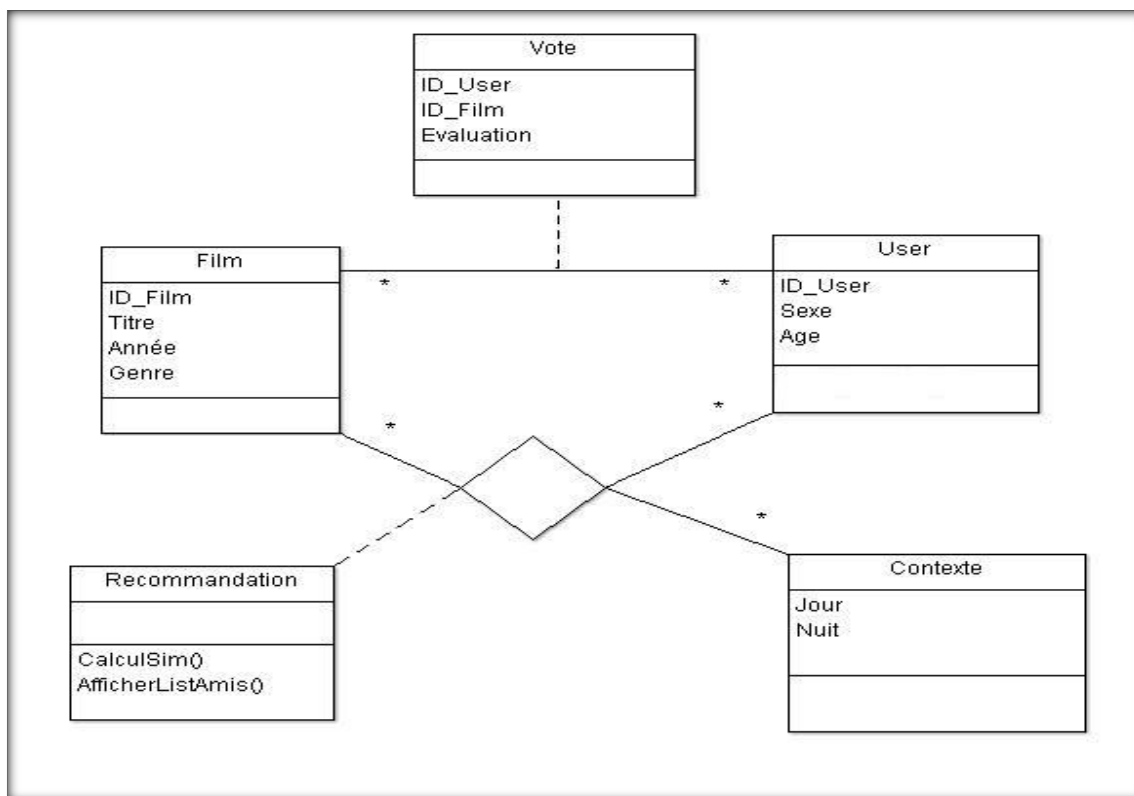


Figure III.5 : Diagramme de classe.

Après avoir détaillé la partie conception de notre système, nous passons à la partie implémentation.

Nous commençons d'abord par décrire la base de données utilisée pour l'évaluation, ainsi que la méthodologie suivie pour la construction des tables. Nous enchaînons en présentant l'environnement de développement de notre système suivi par des captures d'écran de notre application.

### 3.4. Collection de données

Afin d'appliquer des tests sur notre système de recommandation, on a utilisé la base MovieLens<sup>5</sup> qui a été collectée par le Projet de Recherche GroupLens à l'Université de Minnesota. Cette base contient les informations de 3,952 films, plusieurs utilisateurs 6,040 et 100,000,029 votes. Chaque utilisateur a voté au moins 20 films. La valeur d'un vote est entre 1 et 5. Les fichiers principaux de la base MovieLens sont les suivants : users.dat, movies.dat, rating.dat.

<sup>5</sup> <http://www.movielens.org/>

Les informations des films (items) sont contenues dans le fichier movies.dat comme suit :  
MovieID::Title::Genres.

Le fichier users.dat contient des informations démographiques des utilisateurs. Les champs sont séparés par des :: , UserID::Gender::Age::Occupation::Zip-code.

Le fichier rating.dat contient des votes créés par des utilisateurs sur des films (items). C'est une liste dont les champs sont séparés aussi par des :: comme suit :  
UserID::MovieID::Rating::Timestamp.

Puisque les informations originales de la base étaient dans des fichiers de type .dat on les a convertis en fichiers .csv pour pouvoir les importés dans notre système de gestion de base de données. La constitution de ces tables est donnée dans les trois figures suivantes.

### 3.5. Environnement de développement

Nous avons développé notre application logicielle sur une machine Acer 64 bits ayant un processeur Core i3 avec 02 GO de RAM et sous le système d'exploitation Windows 7.

En ce qui concerne l'outil de la gestion de notre base de données on a utilisé :

- **phpMyAdmin** : Il s'agit de l'une des plus célèbres interfaces pour gérer une base de données MySQL sur un serveur PHP. Ce système nous a permis de sauvegarder facilement la base de données sous forme de fichier .csv.

Comme langage de programmation on a utilisé :

**Java** : Le langage Java est un langage de programmation informatique orienté objet créé par James Gosling et Patrick Naughton, employés de Sun Microsystems, avec le soutien de Bill Joy.

- Il existe plusieurs IDE (Integrated Development Environment) pour le langage JAVA par exemple Eclipse, JBuilder et NetBeans.

Le fonctionnement de Java est assuré par JVM (la machine virtuelle Java) et JDK (le Java Développement Kit) qui peuvent être installés dans les différents systèmes d'exploitation.

JDK regroupe l'ensemble des éléments permettant le développement, la mise au point et l'exécution des programmes Java. Il inclut de nombreux outils de développement, un jeu de classes, de services et un ensemble de spécification. [7]

Comme le système de filtrage est développé en Java, on a choisi Netbeans comme outil de développement.

- **Netbeans** : C'est un environnement de développement intégré (EDI), placé en open source par Sun en juin 2000 sous licence CDDL (Common Development and Distribution License).

En plus de Java, NetBeans permet également de supporter différents autres langages, comme Python, C, C++, JavaScript, XML, Ruby, PHP et HTML. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages Web).

Conçu en Java, NetBeans est disponible sous Windows, Linux, Solaris (sur x86 et SPARC), Mac OS X ou sous une version indépendante des systèmes d'exploitation (requérant une machine virtuelle Java). Un environnement Java Development Kit JDK est requis pour les développements en Java. [18]

### 3.6. Description de l'outil développé

Dans cette partie, nous allons présenter notre outil que nous avons développé. L'outil a été implémenté afin d'expérimenter et de vérifier la faisabilité de notre approche. Il est doté d'une interface graphique facile à utiliser qui permet la mise en œuvre des principaux composants de notre architecture.

Dans cette section, nous présentons l'interface graphique (Figure III.6) qui présente la fenêtre principale de notre application elle contient toute les fonctionnalités.

Le bouton Submit permet de retourner une liste de films à recommander pour l'utilisateur après avoir calculé la similarité entre ce dernier et les utilisateurs similaires (même âge et même sexe et qui ont voté sur un film de même genre).

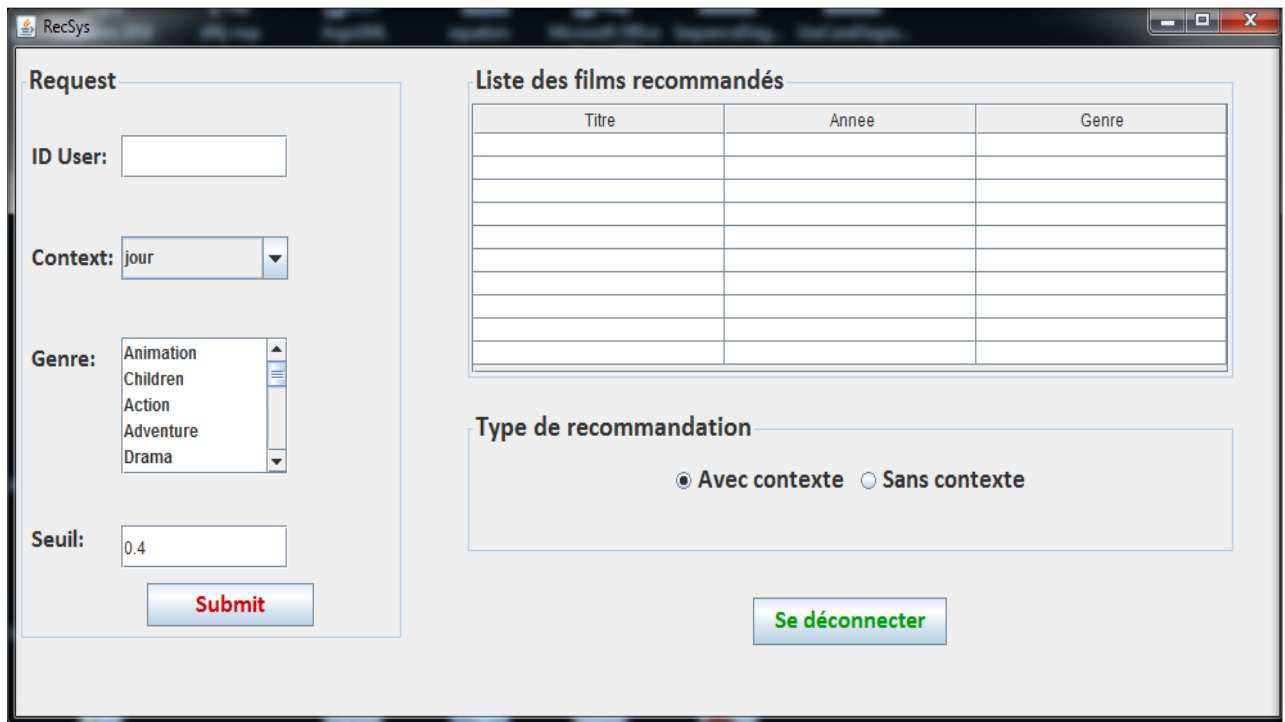


Figure III.6 : Interface principale.

- **Recommandation sans contexte**

Le système retourne à l'utilisateur une liste de films dont le genre est le même qu'il a saisi dans sa requête sans prendre en considération du contexte.



Figure III.7: Recommandation sans contexte.

- Recommandation avec contexte

## 1. Contexte « jour »



Figure III.8 : Recommandation Contexte « jour ».

## 2. Contexte « nuit » avec seuil =0.4

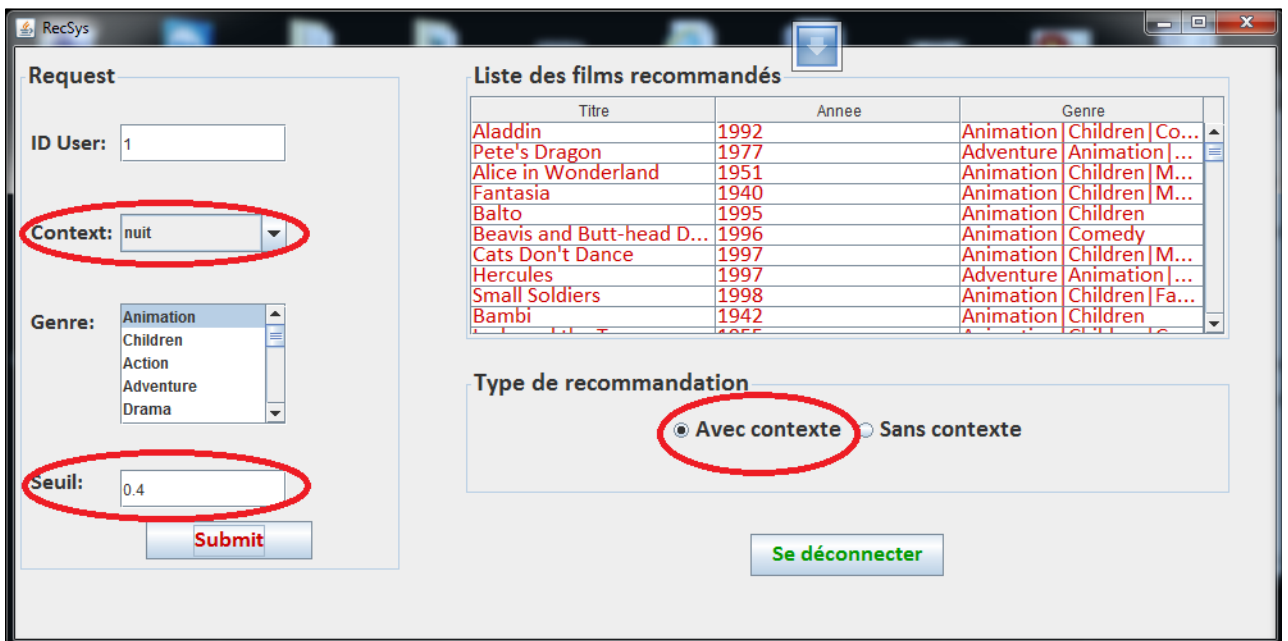


Figure III. 9: Recommandation Contexte « nuit », seuil =0.4.

### 3. Contexte « nuit » avec seuil=0.9



Figure III.10: Recommandation contexte « nuit », seuil=0.9.

Nous avons fait plusieurs expérimentations en effectuant des changements sur le seuil de similarité, on remarque que plus le seuil augmente plus la recommandation est meilleure, nous remarquons également que la méthode contextuelle fournit de meilleurs résultats par rapport à la méthode non contextuelle.

Finalement nous pouvons conclure que l'intégration du contexte dans les systèmes de recommandation a un impact positif sur le processus de recommandation car elle permet d'améliorer les performances de ces derniers.

### 3.7. Conclusion

Au terme de ce chapitre, nous venons de conclure la dernière phase du projet, à savoir l'implémentation de notre système de recommandation hybride. Cette implémentation offre, via un ensemble de programme et de bases de données la possibilité de mesurer la similarité en utilisant la corrélation de Pearson entre un utilisateur donné et les utilisateurs qui sont similaires à fin de lui recommander une liste des films les plus pertinents.



**CONCLUSION**

**GENERALE**

## Conclusion générale et perspectives

Dans ce mémoire, nous avons utilisé l'approche de pré-filtrage contextuel des systèmes de recommandations sensible au contexte (Context Aware ..... ) afin de générer des recommandations plus pertinentes et adapté à la situation contextuelle de l'utilisateur.

Puisque nous avons utilisé l'approche de pré-filtrage contextuel, l'incorporation du contexte dans notre système de recommandation est exécutée avant que le module principal de la recommandation ne soit lancé.

Nous avons utilisé le langage de modélisation UML pour la conception de notre système et après avoir défini l'architecture de notre système, nous avons détaillé la réalisation de notre application.

Notre application retourne une liste de films les plus pertinents à un utilisateur cible, une fois, en prenant en compte de l'information contextuelle et l'autre fois en la négligeant, et ceci en se basant sur la corrélation de Pearson pour établir le taux de similarité entre l'utilisateur ciblé et ceux qui sont similaires de la collection.

Nous pouvons conclure que l'intégration du contexte dans les systèmes de recommandation, permet d'améliorer les performances de ces derniers, en fournissant des informations supplémentaires qui vont être exploitées afin d'effectuer de meilleures recommandations.

Finalement, nous pouvons dire que la contextualisation a un impact positif sur le processus de recommandation.

Ce travail est loin d'être terminé. Des perspectives sont à envisager, comme :

- Expérimenter de nouvelles façons d'intégrer la sensibilité au contexte
- Inclure le contexte social de l'utilisateur (ami, famille, groupe ... ) pour donner des recommandations ; la raison est que généralement les utilisateurs aiment voir les films où les gens qu'ils connaissent ont apprécié
- Intégrer le domaine sémantique par exemple prendre en considération les similarités sémantiques entre les situations contextuelles.

# **REFERENCES**

# **BIBLIOGRAPHIQUES**

## Références bibliographiques

---

- [1] Anderson, The Long Tail. Why the Future of Business Is Selling Less of More. New York, NY: Hyperion, 2006.
- [2] Malone, T., Brobst, S., Cohen, S., Grant, K., and Turbak, Intelligent information des systèmes de partage. In Communications of the ACM, volume 30, pages 390–402, F. (1987).
- [3] Conner, M. and Herlocker. Clustering items for collaborative filtering. In the ACM SIGIR Workshop on Recommender Systems, J. (1999).
- [4] Breese, J., Heckerman, D., and Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In 14<sup>th</sup> Annual Conference on Uncertainty in Artificial Intelligence, pages 43–52, C. (1998).
- [5] Burke. Hybrid recommender systems : Survey and experiments. User Modeling and User-Adapted Interaction, 12(4) :331–370, R. (2002).
- [6] Geoffroy Bonnin, Vers des systèmes de recommandation robustes pour la navigation Web : inspiration de la modélisation statistique du langage, 1 Apr 2011.
- [7] M. KARAOUZENE, Système de recommandation des Services web sémantiques, mémoire de master, univ tlemcen, 14 Juin 2015.
- [8] N. ABDERRAHIM, Contribution des réseaux sociaux dans l'ingénierie des services Web, thèse de doctorat, Université Djillali Liabes de Sidi Bel Abbas, juin 2014.
- [9] Pham, Conception d'un système d'apprentissage et de travail pervasif et adaptatif fondé sur un modèle de scénario. PhDthesis, INFO - Dépt. Informatique (Institut Mines-Télécom-Télécom Bretagne-UEB), UBS - Université de Bretagne Sud, UEB - Université Européenne de Bretagne, C. (2010).
- [10] Bill N. Schilit and Marvin M. Theimer. Disseminating active map information to mobile hosts. Network, IEEE, 8(5):22-32, Sep/Oct 1994.
- [11] N. S. Ryan and J. Pascoe and D. R. Morse. Enhanced Reality Fieldwork: the Context-aware Archaeological Assistant. In V. Gaffney and M. van Leusen and S. Exxon, editor, Computer Applications in Archaeology 1997, British Archaeological Reports, Oxford, October 1998. Tempus Reparatum.

## Références bibliographiques

---

[12] Anind K. Dey and Gregory D. Abowd and Daniel Salber. A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. *Human-Computer Interaction*, 16(2):97-166, 2001.

[13] Bill Schilit and Norman Adams and Roy Want. Context-aware computing applications. In *Proceedings of the Workshop on Mobile Computing Systems and Applications*, pages 85-90. IEEE Computer Society, 1994.

[14] Dey, Providing Architectural Support for Building Context-aware Applications. PhD thesis, Atlanta, GA, USA, A. (2000).

[15] Bill Schilit and Norman Adams and Roy Want. Context-aware computing applications. In *Proceedings of the Workshop on Mobile Computing Systems and Applications*, pages 85-90. IEEE Computer Society, 1994.

[16] Dey, K. A., Abowd, Towards a better understanding of context and context awareness. In *Computer Human Interactions (CHI2000) Workshop on the What, Who, Where and How of Context-Awareness*, D. G. (2000).

[17] Idir Benouaret, Un système de recommandation sensible au contexte pour la visite de musée, Sorbonne universités, Université de technologie de Compiègne.

[18] SOLTANI Réda, Impact de la prise en compte d'information contextuelle sur la pertinence des systèmes de recommandations, mémoire de master, Université d'Oran, 2011.

[19] Gediminas Adomavicius, Bamshad Mobasher, Francesco Ricci, and Alex Tuzhilin, Context-Aware Recommender Systems, article, 2011.

[20] Gediminas Adomavicius, Alexander Tuzhilin, Context-Aware Recommender Systems, article, 2011.

[21] Salber, D., Dey, K. A., Abowd, D. G., (1998). Ubiquitous Computing : Defining an HCI research agenda for an emerging interaction paradigm. In Georgia Tech Gvu technical report. Janvier 1998.

[22] Brown, P. J. Triggering information by context. In *Personal Technologies*. v. 2, p. 1-9, (1998).

[23] <https://openclassrooms.com/courses/debutez-l-analyse-logicielle-avec-uml/uml-c-est-quoi>, consulté le 07/06/2016.

## Liste des figures

---

Figure I.1: Exemple de recommandation basé sur le filtrage collaboratif.....	11
Figure I.2: Recommandation communautaire.....	14
Figure I.3: Recommandation basé sur le filtrage démographique.....	15
Figure I.4: Recommandation à base de connaissance.....	15
Figure II.1: Pré-filtrage Contextuel.....	24
Figure II.2: Post-filtrage Contextuel.....	24
Figure II.3: La modélisation contextuelle.....	25
Figure II.4: Imprime écran du site <a href="http://www.sourcetone.com">www.sourcetone.com</a> .....	26
Figure II.5: Barre d'outils du site <a href="http://www.amazon.com">www.amazon.com</a> .....	27
Figure III.1: Architecture générale de l'application.....	31
Figure III.2: Diagramme de cas d'utilisation.....	34
Figure III.3: Diagramme de séquence du cas : Recommandations avec contexte.....	35
Figure III.4: Diagramme de séquence du cas : Recommandations sans contexte.....	35
Figure III.5: Diagramme de classe.....	36
Figure III.6: Interface principale.....	39
Figure III.7: Recommandation sans contexte.....	39
Figure III.8: Recommandation Contexte « jour ».....	40
Figure III.9: Recommandation Contexte « nuit », seuil =0.4.....	40
Figure III.10: Recommandation Contexte « nuit », seuil =0.9.....	41

## Liste des tableaux

---

Tableau I.1 : Les avantages et les inconvénients des techniques de recommandations.....	17
Tableau III.1 : Matrice Films-Utilisateur.....	32

Rapport-Gratuit.com

## Liste des abréviations

---

SR : Système de recommandation

FBC : Filtrage basé sur le contenu

FC : Filtrage collaboratif

PPV : Plus proches voisins

CMS : Système de Gestion du Contenu

CARS : Context Aware Recommender Systems

UP : Processus unifié

UML : Langage de modélisation unifié



## Résumé

Les systèmes de recommandation traditionnels apportent une solution au problème de surcharge d'informations. Ils sont capables d'estimer l'intérêt d'un item qui peut être livre, film ou hôtel ... pour un utilisateur à partir de certaines informations relatives à d'autres utilisateurs similaires et/ou aux propriétés des ressources. Ces systèmes ignorent que la pertinence d'un item à un utilisateur est faite selon un contexte particulier d'où le rôle des systèmes de recommandation sensible au contexte (Context-aware recommender systems (CARS)). Les CARS permettent d'incorporer l'information contextuelle lors de la recommandation. Dans le cadre de notre projet de fin d'étude, nous allons appliquer l'approche pré-filtrage contextuel des CARS pour la recommandation des films afin de générer des recommandations plus pertinentes et adapté à la situation contextuelle de l'utilisateur.

## Mots clés

Système de recommandation, recommandation sensible au contexte, filtrage collaboratif, pré-filtrage contextuel.

## Absract

Classical recommender systems provide a solution to the information overload problem. They are able to evaluate items (books, movies, hotels ...) for a user based on some information provided by other similar users and/or resource properties. Nonetheless, these systems ignore the relevance of an item to a particular context; it's why recommender systems sensitive to context were developed (Context-aware recommender systems (CARS)).

As part of our final project study, we will apply the contextual pre-filtering approach to CARS for recommending movies to generate more relevant recommendations and adapted to the contextual situation of the user.

## Keywords

Recommender system, Context-aware recommender systems, collaborative filtering, contextual pre-filtering

## ملخص

توفر أنظمة التوصية التقليدية حل لمشكلة المعلومات الزائدة و هي قادرة على تقييم قيمة العنصر الذي يمكن أن يكون كتاب، فيلم فندق مستخدم من بعض المعلومات المتعلقة بمستخدمين آخرين وأو خصائص الموارد. هذه الأنظمة تتجاهل أهمية العنصر للمستخدم على أنها تمت وفق سياق معين حيث دور أنظمة التوصية حساسة للسياق.

في إطار الدراسة النهائية للمشروع، سوف نطبق منهج ما قبل تصفية السياقية لتوصية الأفلام لتوليد توصيات أكثر صلة وتكييفها مع وضعية السياق المحدد من طرف المستخدم.

## الكلمات الرئيسية

نظام توصية، توصية حساسة للسياق، الترشيح ما قبل السياقية، تصفية التعاونية.