

Sommaire

Introduction générale.....	9
1. Les Systèmes de Recommandation Classique	11
1.1 Introduction	12
1.2 Définition des Systèmes de Recommandation	12
1.3 Les Différentes Approches de Recommandation	14
1.3.1 Filtrage Basé sur le Contenu	14
1.3.2 Filtrage Collaboratif	16
1.3.3 Filtrage Hybride	22
1.4 Factorisation matricielle (FM).....	23
1.4.1 Principe de FM	23
1.4.2 Les modèles de la factorisation matricielle	24
1.4.3 Le rôle de la factorisation matricielle dans les SR	27
1.5 Les Inconvénients des Systèmes de Recommandation.....	27
1.6 L'avenir des Systèmes de Recommandation	29
1.7 Conclusion	29
2. Les Systèmes de Recommandation Sensible au Contexte	30
2.1 Introduction	31
2.2 Définition du Contexte	31
2.3 Caractéristiques du Contexte	32
2.4 Dimension du Contexte	33
2.5 Système de Recommandation Sensible au Contexte.....	35
2.5.1 Définition de la sensibilité au contexte	35
2.5.2 Approches d'intégration du contexte dans les systèmes de recommandation ...	36
2.6 Conclusion	38
3. Etat de l'Art des Systèmes de Recommandation Sensible au Contexte.....	39
3.1 Introduction	40
3.2 Les algorithmes utilisés pour chaque paradigme.....	40
3.2.1 Approche Pré-filtrage contextuel	40
3.2.2 Approche Post-filtrage contextuel.....	42

3.2.3	Modélisation contextuelle	43
3.3	Conclusion	44
4	. Conception et Réalisation du Système MovieRec_FM.....	45
4.1	Introduction	46
4.2	Architecture du système MovieRec_FM.....	46
4.3	Conception du Système MovieRec_FM.....	47
4.3.1	Diagramme de cas d'utilisation.....	47
4.3.2	Diagramme de séquence.....	48
4.4	Collection de Données.....	49
4.5	Exemple d'application de FM	51
4.6	Outils d'Implémentation.....	51
4.7	Description de l'Application	52
4.8	Conclusion	57
	Conclusion générale	59
	Liste des figures	65
	Liste des tableaux	66

***INTRODUCTION
GENERALE***

Introduction générale

L'énorme croissance des données a abouti à la création de nouveaux domaines de recherche dans le domaine de l'informatique. Le système de recommandations, un système complètement automatisé qui analysent les préférences des utilisateurs et prédisent le comportement des utilisateurs, en est l'un parmi eux. L'intérêt de la recherche dans ce domaine est encore très élevé principalement en raison de la signification pratique du problème.

Des exemples de ces systèmes incluent la recommandation d'un ensemble d'items par exemple produits d'Amazon, les publicités de produits présentées par Google en fonction de l'historique des recherches, des recommandations de films de Netflix, Yahoo, Movies et MovieLens.

Un système de recommandation se compose de deux tâches : Prédire les notes qu'un utilisateur est susceptible d'attribuer aux items (peut être un livre, un film..etc) puis classer les prédictions.

Parmi les approches des systèmes de recommandation, nous distinguons l'approche de filtrage collaboratif qui ne nécessite pas une description sur l'item à recommander, elle se base sur les notations des utilisateurs pour trouver ceux qui sont les plus similaires. Ces notations se trouvent dans une matrice de $n \times m$, n étant le nombre des utilisateurs et m le nombre d'items.

Plus précisément, le filtrage collaboratif consiste à remplir la matrice de notation de sorte à prédire la note qu'un utilisateur va donner à un item en se basant sur les notations déjà fournies par d'autres utilisateurs.

Toutefois, deux problèmes se présentent, d'un côté, la taille importante de la matrice de notation dont la majorité des données sont inconnues dû au manque d'évaluation par les utilisateurs ; et d'un autre côté, ces systèmes ignorent le fait que des utilisateurs interagissent avec le système selon un contexte particulier et que la note attribuée par un utilisateur à un item peut être différente en changeant le contexte, ce qui influe sur les prédictions et la pertinence des items recommandés.

L'objectif de notre travail est de remédier ces deux limites, pour cela, nous utilisons l'algorithme de factorisation matricielle SVD++ du filtrage collaboratif qui repose sur la

réduction de la dimensionnalité de la matrice de notation ainsi pour la prise en considération du contexte, nous allons exploiter les avantages des systèmes de recommandation sensible au contexte (Context Aware Recommendation system CARS). CARS qui viens pallier les limites des systèmes de recommandations dite traditionnels et fournissent des prédictions plus précises et des recommandations plus pertinentes car seules les notes collectées dans le contexte de l'utilisateur sont utilisées pour générer une prédiction.

Trois approches principales dans lequel les informations contextuelles peuvent être incorporées dans les systèmes de recommandation sensible au contexte : Pré-filtrage Contextuel, et la Modélisation Contextuelle, dans notre travail, nous nous somme basés sur le principe de l'approche Post-filtrage Contextuel pour avoir des recommandations contextuelles.

Pour réaliser notre travail, nous allons travailler sur l'ensemble de données des films du site MovieLens.

Organisation du document

Outre l'introduction et la conclusion, ce document est articulé en 4 chapitres :

Le chapitre 1 définit les systèmes de recommandations classiques Ensuite nous détaillons ses différentes approches et la factorisation matricielle, ses modèles. Enfin nous citons les inconvénients et l'avenir de ces systèmes.

Le chapitre 2 a pour objectif de définir le la notion du contexte donné. En donnant les caractéristiques du contexte, ses différentes dimensions et les différents paradigmes pour intégrer le contexte au CARS.

Le chapitre 3 dédié à l'état d'art des systèmes de recommandation sensible au contexte. Nous présentons quelques algorithmes utilisés pour chaque paradigme d'intégration de contexte.

Et le dernier chapitre sera consacré à notre application de recommandation « MovieRec_FM », nous commençons par, présenter la conception de notre système et nous terminons par une description sur l'application.

1. Les Systèmes de Recommandation Classiques

1.1 Introduction

La masse des données échangées aujourd'hui sur Internet constitue un atout sans précédent pour l'accès à l'information. En parallèle, elle constitue également une difficulté car elle oblige à d'importants traitements de filtrage pour que les données retournées soient effectivement pertinentes pour l'utilisateur le plus rapidement possible, il apparaît que les capacités humaines ne permettent pas l'analyse exhaustive de toutes ces informations.

Pour aider l'esprit humain dans son processus de sélection, des systèmes de recommandation grand public ont vu le jour dans la dernière décennie du vingtième siècle.

Dans ce chapitre, nous commençons d'abord par définir ce qu'est un système de recommandation puis, nous présentons les trois approches principales de filtrage qui permettent la recommandation, nous poursuivons avec la factorisation matricielle, ses modèles et son rôle. Enfin nous citons les inconvénients et l'avenir de ces systèmes.

1.2 Définition des Systèmes de Recommandation

Il existe plusieurs définitions des systèmes de recommandation, nous citons quelques une d'entre eux :

Les systèmes de recommandation (RS) sont des outils logiciels et des techniques fournissant des suggestions pour les articles qui peuvent être utiles à un utilisateur. Les suggestions concernent divers processus de décision, tels que les articles à acheter, la musique à écouter ou les nouvelles en ligne à lire. [1]

Comme l'a déclaré Gediminas et Alexander, « *le système de recommandation aide à faire face à la surcharge d'informations et à fournir des recommandations, du contenu et des services personnalisés* ». [2]

Nous pouvons aussi définir un système de recommandation comme une forme spécifique de filtrage de l'information visant à présenter les éléments d'information (films, musique, livres, news, images, pages Web, etc) qui sont susceptibles d'intéresser l'utilisateur.

Dit autrement, un système de recommandation cherche à prédire la valorisation ou préférence qu'un utilisateur attribuerait à un objet (livre, musique, film...) ou à un élément social (personne, groupe, communauté) qu'il n'avait pas encore considéré.

Les systèmes de recommandation peuvent être vus comme une réponse donnée aux utilisateurs ayant des difficultés à prendre une décision dans le cadre d'utilisation d'un système de recherche d'information. Ils ont pour objectif de fournir à un utilisateur des ressources pertinentes en fonction de ses préférences.

Les systèmes de recommandation jouent un rôle important dans des sites Internet aussi bien classés que Amazon.com, YouTube, Netflix, Yahoo, Tripadvisor, Last.fm et IMDb.

En outre, de nombreuses entreprises de médias développent et déploient actuellement des systèmes de recommandation dans le cadre des services qu'ils fournissent à leurs abonnés. Par exemple Netflix, le service de location de films en ligne, a décerné un prix de 1 million de dollars à l'équipe qui a réussi à améliorer sensiblement les performances de son système de recommandation. L'entreprise a intégré les propositions les plus pertinentes dans sa version du système de recommandation mis en production.

Un système de recommandation requiert généralement 3 étapes [3]:

- *Etape 1* : consiste à recueillir de l'information sur l'utilisateur,
- *Etape 2* : consiste à bâtir une matrice ou un modèle utilisateur contenant l'information recueillie.
- *Etape 3* : consiste à extraire à partir de cette matrice une liste de recommandations.

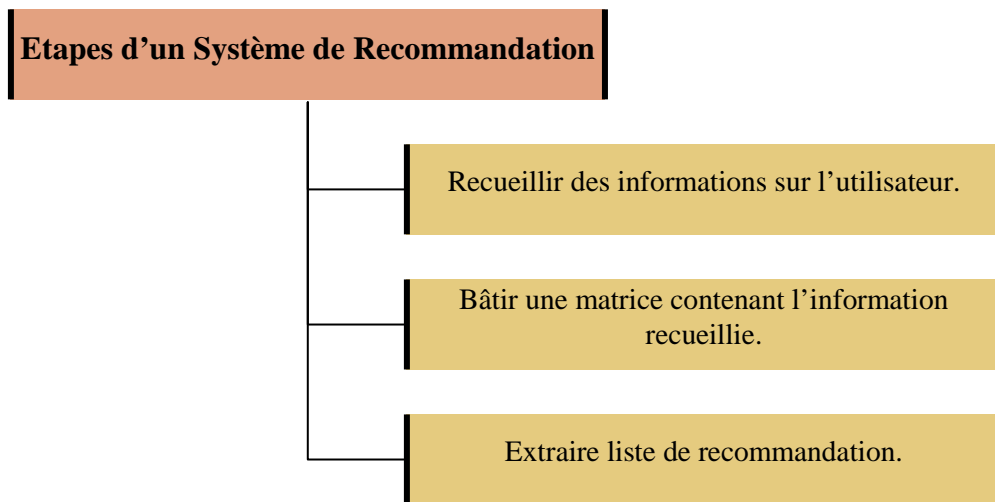


Figure 1-1: Les étapes d'un système de recommandation

1.3 Les Différentes Approches de Recommandation

Un système de recommandation doit pouvoir faire des prédictions sur les intérêts des utilisateurs. Il faut donc pouvoir collecter un certain nombre de données sur ceux-ci afin d'être capable de construire un profil pour chaque utilisateur. Selon [4], nous pouvons distinguer trois grandes classes d'approches selon la nature de ces données.

1.3.1 Filtrage Basé sur le Contenu

Selon Ricci et al. [1], Les systèmes de recommandation basés sur le contenu (*content-based*) fonctionnent en analysant les caractéristiques des objets à recommander (produits, etc.) puis en les regroupant.

Par la suite, le système va suggérer aux utilisateurs ayant acheté/consommé un produit quelconque par le passé, les objets/produits estimés similaires.

Les systèmes de recommandation basés sur le contenu s'appuient sur des évaluations effectuées par un utilisateur sur un ensemble de documents ou items. L'objectif est alors de comprendre les motivations l'ayant conduit à juger comme pertinent ou non un item donné. Le système peut alors proposer à l'utilisateur un choix parmi de nouveaux items jugés proches des items qu'il a précédemment appréciés.

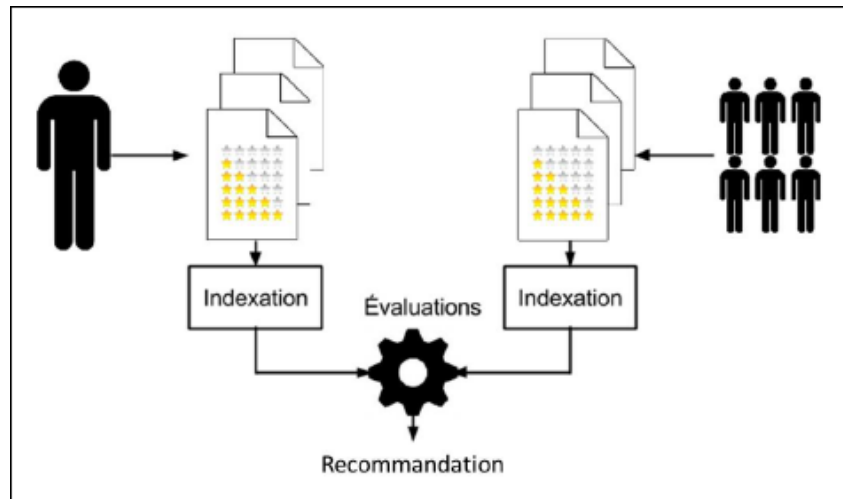


Figure 1-2: Filtrage basé sur le contenu

Les items qui peuvent être recommandés aux utilisateurs sont représentés par un ensemble de caractéristiques, aussi appelées attributs, variables ou propriétés dans la littérature. Un item est représenté dans le système au moyen d'une donnée structurée. Cette donnée structurée est un vecteur

$X = (x_1; x_2; \dots; x_n)$ de n composantes.

Dans la plupart des systèmes de recommandations basés sur le contenu, les descriptions des items sont des caractéristiques textuelles extraites à partir de pages Web, emails, articles de news ou descriptions de produits. Les items correspondent souvent à des mots-clés.

Exemple, lorsqu'un utilisateur a tendance à consulter souvent des articles portant sur le domaine du Web sémantique, le système lui proposera alors des articles liés à ce domaine. [4]

Selon Adomavicius et al. [2] la majorité des recherches sur la recommandation basée sur le contenu portent sur des données textuelles, et la mesure la plus populaire pour l'analyse du contenu est la mesure TF-IDF (Term Frequency-Inverse Document Frequency), permet d'évaluer l'importance d'un mot dans un document.

Les systèmes de recommandations basées sur le contenu ont deux types : les systèmes de recommandation basée sur les mots-clefs et les systèmes de recommandations basés sur la sémantique. [4]

1.3.1.1 Les systèmes de recommandation basée sur les mots-clefs

La majorité des systèmes de recommandation basés sur le contenu utilisent de simples modèles de recherche initialement utilisés en recherche d'informations comme la correspondance de mots-clefs ou le Modèle d'Espace Vectoriel (MEV, qui est une représentation spatiale des documents textuels), et les couples avec la pondération basique TF-IDF. [5]

Dans ce modèle chaque document est représenté par un vecteur de dimension n , ou chaque dimension correspond à un terme de l'ensemble du vocabulaire d'une collection de documents. Formellement, tout document est représenté par un vecteur poids sur des termes, ou chaque poids indique le degré d'association entre le document et le terme.

Ce type de systèmes a été développé et utilisé dans de multiples domaines d'applications. Par exemple dans le domaine de la recommandation d'actualités, nous pouvons citer yourNews.

yourNews contient un système classique fonctionnant à l'aide de vecteurs pondérées avec la mesure TF-IDF et compare à l'aide de la similarité cosinus.

1.3.1.2 Les systèmes de recommandations basés sur la sémantique

Un autre cas particulier des systèmes de recommandations basés sur le contenu sont les systèmes basés sur la sémantique Ils tendent à intégrer les nouvelles technologies du Web sémantique, afin de remédier à certains manques des systèmes basés sur le contenu classique :

Site IF (qui est un agent personnel pour un site Web de nouvelles multilingue) a été le premier système de recommandation à adopter une représentation basée sur le sens des documents pour construire un modèle des intérêts de l'utilisateur [6].

1.3.2 Filtrage Collaboratif

Le filtrage collaboratif est parmi les technologies les plus populaires dans le domaine des systèmes de recommandation [7]. Il fonctionne en recommandant des Items en fonction du comportement passé des utilisateurs similaires, en effectuant une corrélation entre des utilisateurs ayant des préférences et intérêts similaires et cette tâche est faite en

utilisant des méthodes qui collectent et analysent des données sur le comportement, les activités, les préférences des utilisateurs.

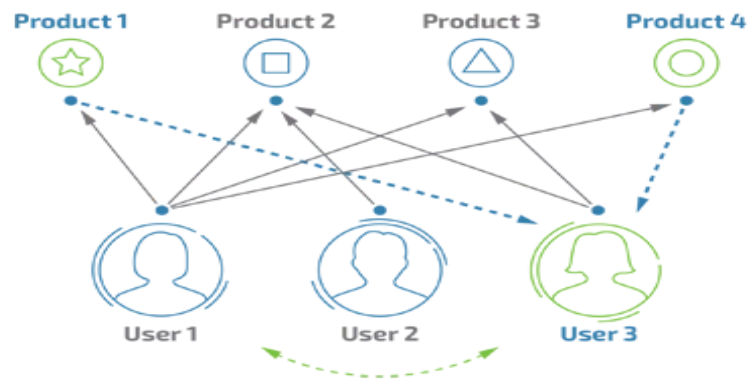


Figure 1-3: Filtrage collaboratif

Des algorithmes tentent de prédire ce que l'utilisateur aimera en cherchant d'autres utilisateurs qui ont les mêmes comportements que l'utilisateur à qui nous souhaitons faire des recommandations. L'idée de base est donc de dire que si des utilisateurs ont partagé des mêmes intérêts dans le passé, il y a de fortes chances qu'ils partagent aussi les mêmes goûts dans le futur.

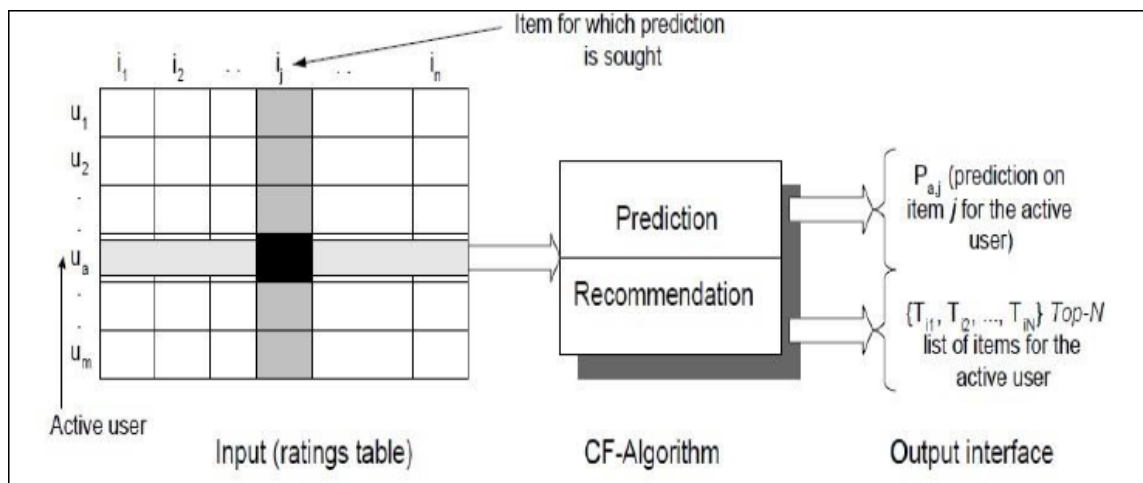


Figure 1-4: Processus du filtrage collaboratif

L'avantage principal de cette approche est qu'elle ne nécessite pas de description précise des objets à recommander. Les recommandations étant basées sur l'ensemble des interactions des utilisateurs avec les objets/produits, cette méthode permet de recommander des objets complexes sans avoir à les analyser.

Comme nous l'avons mentionné avant, pour utiliser cette approche, le système a besoin de collecter des données sur les utilisateurs et leurs préférences, il existe deux méthodologies permettant la construction d'un profil utilisateur :

- ❖ **Collecte Explicite (active):** Dans ce cas, les utilisateurs sont sollicités pour émettre leurs avis sur des produits/objets. Ils peuvent le faire via un système de notation (ex. une grille de 5 étoiles, un questionnaire de satisfaction), ou bien en publiant leurs avis sur un élément donné (ex. La fonction “*J'aime*” sur le réseau social *Facebook*).
- ❖ **Collecte Implicite (passive):** La collecte implicite s'intéresse aux interactions des utilisateurs sur le système. Les exemples de cette collecte incluent la surveillance du nombre de visites sur une page, le nombre de vues sur une vidéo, le temps passé sur une section donnée ou de l'historique des achats sur une plateforme de *e-commerce*. [8].

Le filtrage collaboratif est composé de deux méthodes :

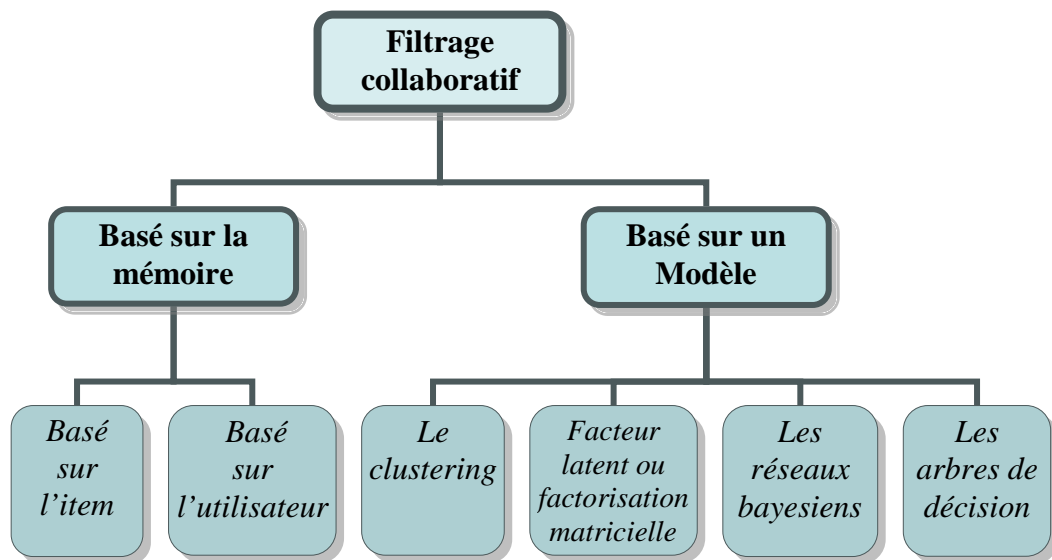


Figure 1-5: Les méthodes de filtrage collaboratif

1.3.2.1 Méthode basée sur la mémoire

Dans un système de filtrage collaboratif basé sur la mémoire, les données sont représentées sous forme d'une matrice "Utilisateur x Item" où les lignes représentent les utilisateurs $U = \{u_1, \dots, u_m\}$ et les colonnes constituent les items $I = \{i_1, \dots, i_j\}$.

Les utilisateurs fournissent leurs opinions concernant les items sous forme de notes v . Pour un utilisateur actif U_a (par exemple Sarah) n'ayant pas exprimé son avis concernant un item i_k (le film "Les visiteurs"), le système recherche les utilisateurs voisins les plus proches notés U_a (parmi Imene, Hadi et Karim ayant noté le film "Les visiteurs" et qui ont déjà co-noté le film "Pulp Fiction" avec Sarah) et utilisent leurs opinions pour prédire la note manquante $v(U_a, i_k)$ (v (Sarah, Les visiteurs)).

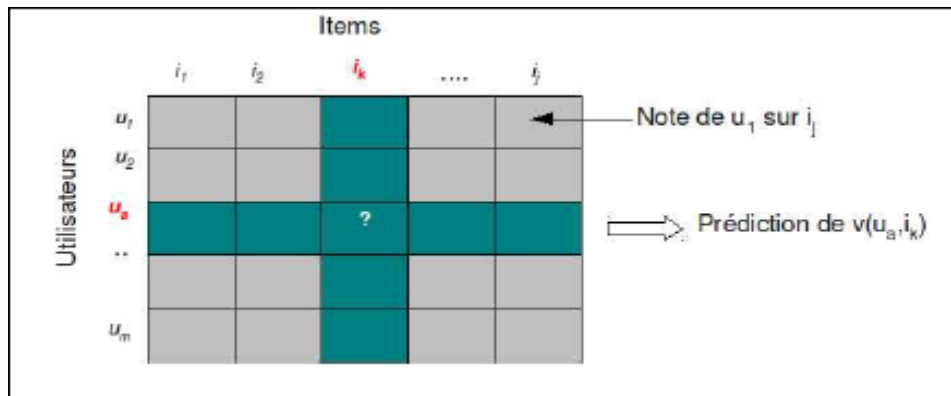


Figure 1-6: Matrice Utilisateur x Item

	Pulp Fiction	Spiderman	Les visiteurs	Shrek
Sarah	1	5	?	3
Imene	4	2	4	?
Walid	3	?	?	5
Karim	4	?	5	?
Hadi	2	?	4	1

Figure 1-7: Tableau Utilisateur x Item

L'approche basée sur la mémoire exploite les appréciations des utilisateurs sur les items (sous forme de notes par exemple), afin de générer les prédictions. Cette approche applique principalement des techniques statistiques dans le but d'identifier des utilisateurs voisins ayant, sur un même ensemble d'items, des appréciations similaires à celles de l'utilisateur actif.

Une fois les voisins identifiés, l'approche basée sur la mémoire utilise différents algorithmes afin de combiner les appréciations des voisins et générer des

recommandations à l'utilisateur actif. Cette méthode utilise alors principalement les *ratings* (*notations*) (par ex : films). [3]

Le poids donné au rating de chaque utilisateur est déterminé par le degré de corrélation entre cet utilisateur et l'utilisateur pour qui nous désirons faire la recommandation. Les systèmes doivent généralement aussi gérer un grand nombre d'utilisateurs.

Faire des recommandations à partir des ratings de millions d'utilisateurs peut avoir de sérieuses implications en termes de performance. Ainsi, quand le nombre d'utilisateurs atteint un certain seuil, une sélection des "meilleurs" voisins doit être faite. Pour déterminer quels sont les voisins les plus pertinents à sélectionner et générer des recommandations fiables, nous utilisons généralement l'algorithme du k-nearest neighbor (k-NN) qui permet de sélectionner seulement les k meilleurs voisins ayant la plus haute valeur de corrélation.

Une autre approche (corrélation-thresholding) serait de sélectionner seulement les voisins possédant une corrélation plus grande qu'un certain seuil.

Nous pouvons distinguer deux méthodes de filtrage collaboratif basé sur la mémoire : la méthode basée sur la mémoire centrée sur l'item et la méthode basée sur la mémoire centrée sur l'utilisateur. [9]

1.3.2.1 Méthode basée sur un modèle

Les méthodes basées sur un modèle ont été intégrées aux systèmes de recommandation pour améliorer et remédier aux problèmes des méthodes basées sur la mémoire.

Les algorithmes basés sur le modèle se basent aussi sur les évaluations précédentes (les profils) des usagers, mais cette méthode ne calcule pas directement les prédictions, elle classifie les usagers suivant des groupes ou d'apprendre les modèles à partir de leurs données

Pour la construction du modèle plusieurs méthodes sont utilisées. En général, les méthodes basées sur le modèle utilisent les techniques d'apprentissage automatiques, telles que le clustering, la factorisation matricielle, les réseaux bayésiens, les arbres de décision, etc. [9]

Dans notre travail, nous allons nous focaliser essentiellement sur la factorisation matricielle aussi dite décomposition matricielle. Elle consiste à décomposer une matrice en plusieurs autres matrices. Pour retrouver la matrice originale, il suffira de faire le produit de ces matrices entre elles.

La factorisation matricielle a donné de bons résultats dans les systèmes de recommandation [10].

Comparaison entre les systèmes collaboratifs et les systèmes basés sur le contenu

Chacune des approches de recommandation présentées a des avantages ainsi que des inconvénients.

Un système de recommandation basé sur le contenu ne nécessite une connaissance du domaine étudié. En effet, seul la connaissance de l'utilisateur est requise. Le caractère dynamique de ces systèmes est également un avantage car plus l'utilisateur va utiliser le système et plus la pertinence des items qui lui seront proposés sera fine mais le système ne pourra pas fournir des recommandations de qualité pour un utilisateur fraîchement inscrit. [9]

Le filtrage collaboratif se distingue par sa capacité à recommander à un utilisateur ce qui est hors du familier, c'est ce que Burke [11] appelle : cross-genre niches. En effet, un utilisateur peut se voir recommander des items de genres différents.

Par exemple, un utilisateur qui a des voisins similaires du point de vue des politiques, peut se voir recommander des articles de sport si ces voisins aiment les articles de sport, même si cet utilisateur n'a jamais exprimé ce genre de favoris.

Contrairement aux systèmes de recommandations basés sur le contenu un utilisateur ne se verra jamais proposer d'items qui n'auront pas été jugés similaires à ceux qu'il apprécie. C'est à dire si un utilisateur ne s'intéresse qu'aux articles parlant de la médecine, il ne se verra jamais proposer un article de cuisine. [12]

Un autre problème commun au filtrage basé sur le contenu et au filtrage collaboratif, est le problème de démarrage à froid c'est-à-dire pas d'informations sur un nouvel utilisateur (qui n'a pas encore accumulé suffisamment d'évaluations). Ce dernier, ne

peut pas avoir de recommandations pertinentes. D'où l'impossibilité de le classer ou de lui trouver un voisinage.

Et pour le problème de démarrage à froid, cas du nouvel item nous parlons de ce problème quand nous parlons du filtrage collaboratif et non pas le filtrage à base de contenu. Dans le cas du filtrage à base de contenu, il suffit d'introduire l'item dans le système pour que celui-ci soit analysé et rentré dans le processus de recommandation. Dans le cas du filtrage collaboratif il doit avoir suffisamment d'évaluations pour fonctionner et être pris en considération dans le processus de recommandation. [4][9]

Pour remédier aux inconvénients et profiter des avantages des deux approches, l'approche hybride a vu le jour.

1.3.3 Filtrage Hybride

Cette approche est en fait une combinaison des deux types d'approches précédemment cités [2]. Les systèmes hybrides tirent parti des avantages des deux approches citées précédemment.

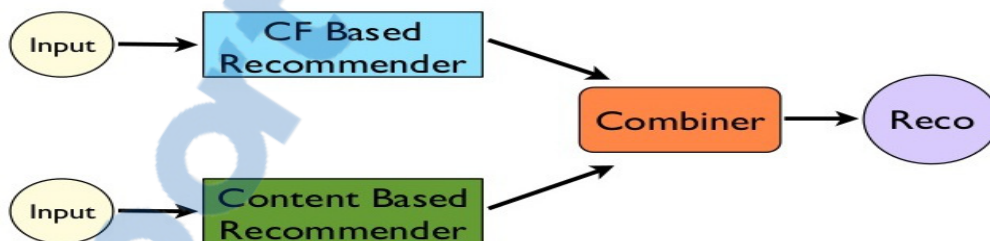


Figure 1-8: Filtrage hybride

Cette solution permet de combler les lacunes de l'une des approches sur des cas d'utilisation précis.

Il existe différentes possibilités de combinaison [9], Adomavicius et Tuzhilin [2] les ont classifiés en quatre catégories :

- Implémenter la méthode collaborative et la méthode basée sur le contenu séparément puis combiner leurs prédictions.



- Incorporer quelques caractéristiques de la méthode basée sur le contenu dans l'approche collaborative.
- Incorporer quelques caractéristiques de la méthode collaborative dans l'approche à base de contenu.
- Construction d'un modèle général unifié qui incorpore les caractéristiques des deux modèles.

1.4 Factorisation matricielle (FM)

Factorisation matricielle ou en Anglais matrix factorisation (MF) est parmi les algorithmes utilisés dans le filtrage collaboratif qui appartient aux méthodes basé sur le modèle. Nous présentons dans ce qui suit le principe et les différents modèles de FM et son rôle dans les systèmes de recommandation.

1.4.1 Principe de FM

La plupart des modèles MF sont basés sur le modèle du facteur latent. C'est l'approche la plus précise pour réduire le problème de sparsity dans la base de données d'un système de recommandation, certaines études ont utilisé des techniques de réduction de la dimensionnalité.

La décomposition de la valeur singulière (SVD) et l'analyse des composantes principales (PCA) sont des techniques bien établies pour identifier les facteurs latents dans le domaine de la récupération d'information pour faire face aux défis des filtres collaboratifs (FC).

Ces méthodes sont devenues populaires récemment en combinant une bonne évolutivité et une précision prédictive. Ils offrent beaucoup de flexibilité pour la modélisation de diverses applications de la vie réelle.

Tout d'abord, nous avons un ensemble d'utilisateurs U et un ensemble d'articles I . Soit X la matrice de taille $(m \times n)$ qui contient toutes les notes que les utilisateurs ont attribuées aux articles (rating matrix). Maintenant, les caractéristiques latentes seraient découvertes.

Notre tâche est donc de trouver deux matrices, P ($m \times f$) et Q ($f \times n$) f est le factor latent (un entier positif tel que $f < m \times n$), de sorte que leur produit équivaut approximativement à R est donné par [13]:

$$X_{ui} \approx Q_i^T * P_u$$

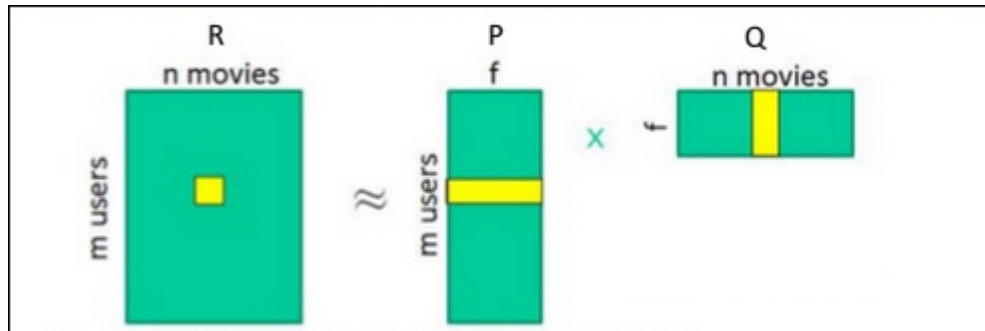


Figure 1-9: Factorisation matricielle

De cette façon, les modèles de factorisation matricielle mettent à la carte les utilisateurs et les éléments dans un espace de facteur latent conjoint de la dimensionnalité f , les interactions avec les éléments de l'utilisateur sont modelées comme des produits internes dans cet espace [13].

En conséquence, chaque élément i est associé à un vecteur $Q_i \in R_f$, et chaque utilisateur u est associé à un vecteur $P_u \in R_f$. Pour un élément donné i , les éléments de Q_i donnent la mesure dans laquelle l'élément possède ces facteurs positifs ou négatifs. Le produit de point résultant $Q_i^T P_u$ capture l'interaction entre l'utilisateur U et l'élément I , l'intérêt général des utilisateurs pour les caractéristiques de l'élément.

1.4.2 Les modèles de la factorisation matricielle

Il existe plusieurs modèles de factorisation matricielle, les plus utilisés sont :

1.4.2.1 La décomposition de la valeur singulière SVD

La décomposition de la valeur singulière, en Anglais (Singular Value Decomposition SVD) est la technique la plus puissante de réduction de la dimensionnalité. Il réduit d'abord la dimensionnalité d'une matrice, avant d'utiliser le résultat de ce processus pour rapprocher les scores originaux. SVD réduit une matrice ($m \times n$) en trois matrices, U , S et V .

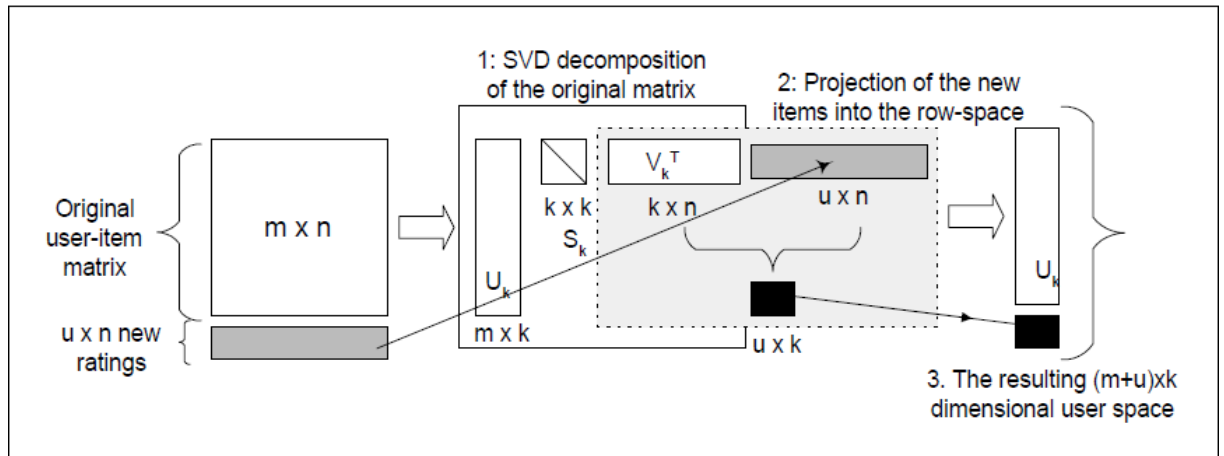


Figure 1-10: Décomposition de la valeur singulière SVD [14]

U et V sont deux matrices orthogonales de taille $(m \times k)$ et $(n \times k)$ où k est le rang de la matrice R . S est une matrice $k \times k$ et contient toutes les valeurs singulières de R comme ses entrées diagonales.

La dimensionnalité des matrices peut être réduite en choisissant S pour ne contenir que les k plus grandes valeurs singulières de R . Cela fait de S_k une matrice $k \times k$, et puisque la réduction peut également être appliquée aux matrices U et V en supprimant les colonnes (r_k) des lignes U et (r_k) de V , la dimensionnalité de USV' peut être considérablement réduite en $U_k S_k V_k'$, ce qui permet d'économiser de l'espace et de la puissance de calcul en plus de supprimer les données de bruit de R .

Le résultat de l'exécution de SVD sur R est donc R_k , une version de dimension réduite de R . [10]

Ces matrices peuvent ensuite être utilisées pour prédire une note d'un utilisateur à un élément i . et c'est la technique que nous allons utiliser dans notre application.

1.4.2.2 Analyse des composants principaux PCA

L'analyse des composantes principales (ACP) (en Anglais Principal Component Analysis PCA) est une réalisation particulière de l'approche de la factorisation de la matrice (MF) [15].

L'ACP est une procédure statistique qui utilise une transformation orthogonale pour convertir un ensemble d'observations de variables éventuellement corrélées en un

ensemble de valeurs de variables linéairement non corrélées appelées composantes principales. Le nombre de variables d'origine est supérieur ou égal aux composants principaux.

Cette transformation est définie de telle sorte qu'une projection linéaire de données dimensionnelles élevées dans un sous-espace dimensionnel inférieur tel que la variance retenue est maximisée et l'erreur de reconstruction de moindre carré est minimisée. Les principaux composants sont orthogonaux car ils sont les vecteurs propres de la matrice de covariance.

PCA est sensible à la mise à l'échelle relative des variables d'origine. PCA permet d'obtenir une liste ordonnée de composants qui représentent la plus grande quantité de variance à partir des données en termes d'erreurs de moindres carrés.

La quantité de variance saisie par le premier composant est supérieure à la quantité de variance sur le deuxième composant et ainsi de suite. Nous pouvons réduire la dimensionnalité des données en négligeant ces composants. [10]

1.4.2.3 La factorisation probabiliste matricielle (PMF)

La factorisation probabiliste de la matrice (en Anglais Probabilistic Matrix Factorization PMF) est un modèle linéaire probabiliste avec un bruit d'observation gaussien. La matrice de préférence de l'utilisateur est représentée comme le produit des matrices d'objets et d'éléments de base inférieurs de filature dans la factorisation probabiliste de la matrice (PMF).

Supposons que nous avons N utilisateurs et M films. R_{ij} la valeur nominale de l'utilisateur i pour le film j , U_i et V_j représentent respectivement des vecteurs de caractéristique latente spécifiques à l'utilisateur spécifiques à la D -dimension et au film. Ensuite, la distribution conditionnelle sur les notes observées $R \in \mathbb{R}^{n \times m}$ et les distributions antérieures sur $U \in \mathbb{R}^{n \times d}$ et $V \in \mathbb{R}^{d \times m}$ sont donnés par [16] dans l'ensemble des équations présentés ci-dessous :

$$\begin{aligned}
 1/ \quad p(R|U, V, \sigma^2) &= \prod_{i=1}^N \prod_{j=1}^M [\mathcal{N}(R_{ij}|U_i^T V_j, \sigma^2)]^{I_{ij}} \\
 2/ \quad p(U|\sigma_u^2) &= \prod_{i=1}^N \mathcal{N}(U_i|0, \sigma_u^2 I) \\
 3/ \quad p(V|\sigma_v^2) &= \prod_{j=1}^M \mathcal{N}(V_j|0, \sigma_v^2 I)
 \end{aligned}$$

$\mathcal{N}(x|\mu, Q^2)$ désigne la distribution gaussienne avec la moyenne μ et la variance. I_{ij} est la variable indicatrice égale à 1 si l'utilisateur a évalué le film j et égal à 0 sinon.

1.4.3 Le rôle de la factorisation matricielle dans les SR

Nous avons d'abord analysé la matrice d'évaluation de l'élément utilisateur pour identifier la relation entre différents éléments, puis nous avons utilisé ces relations pour calculer directement les recommandations pour les utilisateurs en utilisant la technique basée sur l'objet.

Nous avons appliqué le modèle de factorisation matricielle SVD pour réduire la dimensionnalité d'une matrice d'évaluation. À l'aide de l'ensemble de données MovieLens, nous avons sélectionné 100 utilisateurs pour former une matrice (100 × 3500) chaque utilisateur en moyenne 5% des 3500 films.

Nous remplissons d'abord les valeurs manquantes en utilisant les moyennes d'évaluation des utilisateurs et des films, puis postulez SVD. Pour cette grande base de données, la technique basée sur l'objet fournit une précision optimale avec des recommandations beaucoup plus rapides et de haute qualité.

1.5 Les Inconvénients des Systèmes de Recommandation

Malgré le succès des systèmes de recommandation, ils ont subi quelques ratés. Nous allons citer quelques problèmes.

- **Démarrage à froid** : Un problème des plus connus dans les systèmes de recommandation est lorsque l'on ne détient aucune ou peu d'informations et/ou sur un utilisateur, il y a 3 grands types de ce problème : [17]

- Cas du nouvel utilisateur : est qu'un nouvel utilisateur qui n'a pas encore accumulé suffisamment d'évaluations ne peut pas avoir de recommandations pertinentes.
- Cas du nouvel item : le système doit avoir suffisamment d'évaluations pour que l'item soit pris en considération dans le processus de recommandation.
- Cas du système débutant : Le cas du système débutant provient lors du lancement d'un nouveau service de recommandation. Le système ne possède alors aucune information sur les utilisateurs et sur les items.
- **Le shilling** : C'est l'action malveillante d'influencer la recommandation, en créant de faux profils pour voter et favoriser/défavoriser certains items. Par exemple, un vendeur qui, à travers des faux profils, valorise ses produits pour que le système les recommande pour les clients en ligne. [4]
- **Rareté (Sparsity) Le gray Sheep** : Quand des utilisateurs ont des goûts atypiques (qui varient de la norme), ils n'auront pas beaucoup d'utilisateurs en tant que voisins. Cela mènera à des recommandations pauvres. Ce problème est également connu comme gray sheep. La popularité est un problème qui se produit fréquemment dans le filtrage collaboratif. [12]
- **Manque de données** : l'identification des appréciations des utilisateurs est l'un des piliers de base du processus de recommandation. Ces appréciations sont soit renseignées explicitement par les utilisateurs eux-mêmes ou bien induites par le système sur la base de l'analyse des interactions de ces utilisateurs avec le système. Dans les deux cas, souvent les données relatives aux appréciations des utilisateurs manquent et s'avèrent insuffisantes pour le bon fonctionnement du système de recommandation. En effet, la quantité de données ou de notes disponible demeure toujours insuffisante pour pouvoir prédire correctement les notes manquantes. Donc, en raison de ce manque de données, la modélisation des utilisateurs devient complexe. Les modèles utilisateurs deviennent ainsi peu fiables. [9]
- **Protection de la vie privée** : Un autre inconvénient des systèmes de recommandation est la protection des informations sensibles constituant le profil de l'utilisateur (informations personnelles, intérêts, goûts, habitudes, etc.). Vu la

nature de l'information, ces systèmes doivent assurer une telle protection. Ainsi, des moyens, pour préserver l'anonymat des utilisateurs et chiffrer les données transmises, sont nécessaires. [12]

1.6 L'avenir des Systèmes de Recommandation

La recherche dans le domaine des systèmes de recommandation n'en est qu'à ses débuts, l'engouement des chercheurs au cours de ces vingt dernières années a fait grandement avancer les choses. Il existe plusieurs livres qui présentent un bilan exhaustif des systèmes de recommandation, montrent déjà une certaine maturité du domaine de recherche, comme par exemple le livre *Recommender Systems Handbook*.

Des workshops et des thèmes de recherche consacrés aux systèmes de recommandation ont fait leur apparition dans les principalement conférences internationales en apprentissage artificiel et en fouille de données ou dans des conférences spécialisées dans le domaine.

L'évolution constante de sociétés sur Internet ayant de réels besoins en termes de recommandation assure un avenir florissant à ce domaine de recherche.

1.7 Conclusion

Dans ce chapitre, nous avons donné une définition des systèmes de recommandation en citant les différentes approches les plus utilisées et les détaillées en donnant des exemples pour mieux comprendre la notion de ces systèmes.

Après nous avons parlé de la factorisation matricielle et ses modèles. Enfin, nous avons cité les problèmes de ces systèmes. Un de ces problèmes est le fait que ces systèmes restent limités car ils ne sont pas capables de s'adapter à leur environnement, disent autrement ils ne prennent pas en compte le contexte (c'est-à-dire : l'ensemble des éléments qui peuvent influencer la compréhension d'une situation particulière).

Delà, les systèmes de recommandation sensible au contexte (les CARS : Context Aware Recommender Systems) sont nés et leur qualité est étroitement liée à leur capacité à prendre en compte le contexte dans lequel se trouve l'utilisateur lorsqu'il désire une recommandation. Nous allons le détailler dans le chapitre 2.

2. Les Systèmes de Recommandation Sensibles au Contexte

2.1 Introduction

Les systèmes de recommandation ont donné de bons résultats en ce qui concerne la manipulation des données mais plus la technologie se développa plus les exigences étaient fortes. C'est là qu'apparue la notion de contexte qui a son tour à révolutionner l'air des systèmes de recommandation.

Dans la première partie, nous présenterons les différentes visions de la définition du contexte donnée par quelques chercheurs. Nous enchaînerons ensuite par donner les caractéristiques du contexte et ses différentes dimensions.

La deuxième partie du chapitre sera consacré aux systèmes de recommandation sensible au contexte "Context-Aware Recommender System - CARS". Nous donnerons quelques définitions et les différents paradigmes pour intégrer le contexte au CARS.

2.2 Définition du Contexte

Chacun de nous peut avoir une idée générale de ce qu'est le contexte, par contre il est beaucoup plus difficile de donner une définition précise à la notion de contexte.

Littérairement parlant et plus précisément dans la Rouse, la définition du contexte est donnée comme suit : « *Ensemble des conditions naturelles, sociales, culturelles dans lesquelles se situe un énoncé, un discours* »

En informatique, plusieurs définitions ont vu le jour par plusieurs chercheurs. Dans notre travail, nous avons sélectionné quelques une qui ont été marquées au fil des années.

En premier, les chercheurs ont débuté par définir le contexte en énumérant tous les éléments (ces éléments constituent des informations non fonctionnelles de l'application, mais qui peuvent apporter un impact sur cette dernière) qui peuvent être considérés comme une information du contexte. Parmi ces définitions nous trouvons :

Schilit [18] qui considère que le contexte est la localisation de l'utilisateur, les identités et les états des personnes et des objets qui l'entourent.

Rayn et al. [19] à leur tour présentent le contexte par la localisation, l'environnement physique, l'identité et le temps de l'utilisateur.

Dans d'autres recherches,

Gaetan Rey et Joelle Coutaz proposent [20] quatre axes de définitions du contexte sur lesquels s'accordent une majorité des chercheurs :

- Il n'y a pas de contexte sans contexte : La notion de contexte doit se définir en fonction d'une finalité. Par exemple, nous cherchons à adapter dynamiquement les capacités interactives d'un système.
- Le contexte est un espace d'information qui sert l'interprétation : la capture du contexte n'est pas une fin en soi mais les données capturées doivent servir un objectif.
- Le contexte est un espace d'informations partagé par plusieurs acteurs : ici, il s'agit de l'utilisateur et du système.
- Le contexte est un espace d'information infini et évolutif : le contexte n'est pas figé une fois pour toutes mais se construit au cours du temps.

Selon Anind K. Dey et Gregory D. Abowd [21], cette définition est trop vague pour donner un cadre formel au contexte. Leur étude vise à donner la définition la plus précise possible.

Pour eux, « *le contexte est l'ensemble de toutes les informations qui peuvent être utilisées pour caractériser la situation d'une entité* ». Une entité pouvant être un acteur, un lieu, un objet de l'environnement considéré comme utile à l'interaction entre la personne et le système.

Au final, Chaari dans [22] considère toujours que la définition de K. Dey et Gregory D. Abowd est vague car elle ne donne pas une limite entre les informations du contexte et les données de l'application. Il propose alors une nouvelle définition pour apporter plus de précision.

« le contexte est l'ensemble des paramètres externes à l'application qui peuvent influencer sur son comportement en définissant de nouvelles vues sur ses données et ses fonctionnalités. Ces paramètres ont un aspect dynamique qui leur permet d'évoluer durant le temps d'exécution »
[23]

2.3 Caractéristiques du Contexte

Dans ce qui suit, nous allons donner des caractéristiques techniques de l'information du contexte :

- **La Continuité** : Pendant que l'utilisateur se déplace d'une localisation à une autre ce qui implique un changement dans la valeur du contexte, ou bien quand l'utilisateur obtient de nouvelles ressources, le contexte évolue. Cette évolution doit s'accompagner d'un fonctionnement continu des systèmes malgré les changements incessants du contexte.
- **L'hétérogénéité** : Une large hétérogénéité en termes de modélisation, de traitement et de qualité provient du fait que le contexte est capturé à partir d'une variété de sources [24]. Par exemple, en informatique, les ordinateurs portables, les smartphones, les tablettes sont différentes à tous niveaux (Matériels, logiciels, communications...etc.).
- **L'imprévisibilité** : L'une des questions qui se pose est de savoir comment un système doit prendre en compte l'imprévisibilité des changements de l'environnement et les interactions de l'utilisateur. Car certains éléments du contexte peuvent ne pas être connus à l'avance dans le système.
- **L'interdépendance** : L'information et/ou changement de valeur de l'information de contexte, peut être dépendante d'une autre information et/ou changement de valeur de l'information de contexte [25].
- **Contexte imparfait** : cette caractéristique est déterminée selon la source de l'information. Le contexte peut donc être imprécis ou bien le contexte peut être erroné ou même inconnu.

2.4 Dimension du Contexte

Pour de raisons d'adaptation, les informations contextuelles doivent être collectées et présentées au système, et vu l'hétérogénéité et la diversité de ses informations, Il est donc préférable de faire une catégorisation.

Nous présentons ici la plupart des dimensions couramment utilisées :

Schilit et al. dans [18] donne une catégorisation du contexte en deux dimensions.

Contexte primaire : qui contient les informations sur la localisation, temps, l'identité et l'activité de l'utilisateur.

Contexte secondaire : qui contient des informations pouvant être déduite du contexte primaire.

Dans des travaux plus avancés et spécialement de Chen and Kotz [26] qui proposent le contexte actif : influence sur le comportement d'une application, et contexte passif : qui est nécessaire mais pas critique pour l'application.

En se basant sur les deux recherches précédentes, Petrelli et al. [27] ont ajouté deux nouvelles dimensions du contexte :

Contexte matériel (device, plateforme existante), Contexte social (être seul ou pas, qui sont les autres, les relations entre ces individus, etc.)

Le travail de [28] donne une catégorisation de contexte selon un contexte utilisateur, un contexte physique et un contexte social.

- Le contexte utilisateur englobe les préférences de l'utilisateur.
- Le contexte physique définit les caractéristiques des dispositifs physiques (la puissance du CPU, la bande passante).
- Le contexte social décrit les interactions entre les utilisateurs existants et les nouveaux arrivants.

Enfin Dey [29] propose de classer toute variable contextuelle dans des dimensions répondant chacune aux questions, *quand ?*, *où ?*, *quoi ?* Et *qui ?*

Ainsi afin de donner une catégorisation synthétisé et globale, Pham [30] propose les dimensions suivantes :

- **Dimensions Temporelle** : indique une référence temporelle aux actions courantes de l'utilisateur et garde une trace pour les actions passées.
- **Dimension Spatial** : inclut l'information de l'emplacement de l'utilisateur, terminaux disponibles à proximité et d'autres utilisateurs.
- **Dimension Utilisateur** : décrit les participants humains du contexte dans le système et les relations entre eux.
- **Dimension Dispositif** : caractérise les dispositifs d'apprentissage mobile, les ressources informatiques et les outils pédagogiques.

- **Dimension Environnement** : décrit les caractéristiques de l'environnement autour de l'utilisateur.
- **Dimension du Scénario/Activité** : détermine l'intention de l'apprenant, les informations, les connaissances... nécessaire à l'accomplissement d'une activité.

Dans la section suivante, nous entamant les systèmes de recommandation sensible au contexte.

2.5 Système de Recommandation Sensible au Contexte

2.5.1 Définition de la sensibilité au contexte

Le terme sensibilité au contexte est de bien réagir en prenant en considération l'information de contexte.

La première définition a été introduite par Schilit et al. [18] pour eux la sensibilité au contexte est la capacité d'un système à s'adapter au contexte de son exécution en fonction de sa localisation, l'ensemble des personnes à proximité, des équipements accessibles, etc.

Ryan et al. [19] ajoutent ensuite que la sensibilité au contexte est l'aptitude de capturer, interpréter et répondre aux aspects de l'environnement local de l'utilisateur et de terminal.

Une condition a été proposée par Dey et al. [21] pour savoir si un système est sensible au contexte. Ils disent que si un système 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), dans ce cas-là le système est beau et bien sensible au contexte.

Pour finir, Une définition simple est complète à capter notre attention et nous a permis de bien comprendre cette notion, elle est définie comme suit : « *Un système sensible au contexte est un ensemble de mécanismes destinés pour, d'une part, la collection et la gestion des informations de contexte et, d'une autre part, la gestion et le contrôle des actions du système (son comportement) en fonction de ces informations de contexte.* » [31]

Dans ce qui suit, nous allons montrer comment intégrer du contexte dans les systèmes de recommandations.

2.5.2 Approches d'intégration du contexte dans les systèmes de recommandation

Selon U.Panniello et al. [32], trois différents paradigmes algorithmiques d'intégration de contexte sont présentés :

2.5.2.1 Pré-filtrage Contextuel

Avant de lancer la méthode principale de recommandation sur les données sélectionnées, l'approche de pré filtrage consiste à utiliser des informations contextuelles pour sélectionner ou construire les données 2D (Utilisateur x Article) les plus pertinentes pour générer des recommandations.

Ces informations sont considérées comme une étiquette qui permet de filtrer les évaluations qui ne correspondent pas à l'information contextuelle spécifiée. La figure ci-dessous montre l'intégration du contexte dans un SR à deux dimensions.

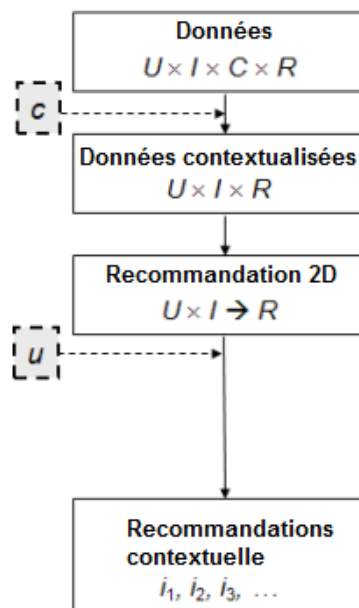


Figure 2-1: Pré-filtrage contextuel [33]

2.5.2.2 Post-filtrage Contextuel

L'approche post-filtrage contextuelle ignore les informations de contexte dans les données d'entrée lors de la génération de recommandations.

Ensuite, l'approche post-filtrage contextuelle ajuste la liste de recommandations obtenue pour chaque utilisateur en utilisant des informations contextuelles. Les ajustements de la liste de recommandations peuvent être effectués par :

- Filtrer les recommandations qui ne sont pas pertinentes (dans un contexte donné)
- Ajuster le classement des recommandations sur la liste (en fonction d'un contexte donné).

La figure montre comment utiliser le contexte après la recommandation (2D).

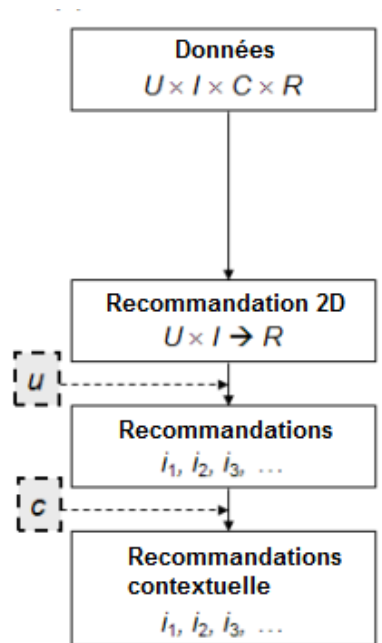


Figure 2-2: Post-filtrage contextuel [33]

Une comparaison a été faite par U.Panniello et al. [32] entre les approches pré filtrage et post-filtrage pour générer des recommandations contextuelles. Ils ont utilisé la méthode Exact Pre-Filtrig (EPF) et les méthodes de Post Filtering Weight et Filter. Après des études, les expérimentations ont montré que la méthode du Filter domine EPF et que l'EPF domine la méthode du Weight.

Au final, la comparaison entre les deux approches dépend de manière significative du type de la méthode de post filtrage utilisée.

2.5.2.3 Modélisation contextuelle

Une troisième méthode qui est la modélisation contextuelle, utilise directement des informations contextuelles dans la fonction de recommandation comme prédicteur explicite de la note d'un utilisateur pour un article.

Contrairement aux approches contextuelles de Pré filtrage et Post filtrage qui utilisent des fonctions de recommandation 2D traditionnelles, l'approche de modélisation contextuelles

donne lieu à des fonctions de recommandation multidimensionnelles, qui représentent essentiellement des modèles prédictifs (construits à l'aide d'un arbre de décision, d'une régression, d'un modèle probabiliste etc.) ou des calculs heuristiques qui intègrent des informations contextuelles. La figure montre cette intégration.

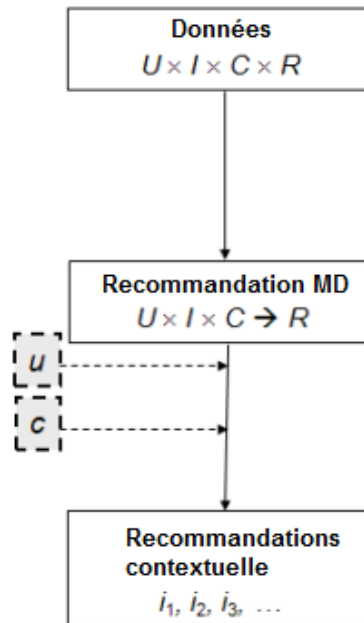


Figure 2-3: Modélisation contextuelle [33]

2.6 Conclusion

Ce chapitre présente une nouvelle famille de systèmes de recommandation appelés systèmes de recommandation sensible au contexte. Nous avons présenté des définitions du contexte, la sensibilité au contexte et aussi parler des différents paradigmes utilisés dans les systèmes de recommandation sensible au contexte.

Vu l'utilité de la sensibilité au contexte et son développement qui grandit chaque année, la majorité des travaux récent cité auparavant, se basent sur le contexte comme un élément primordial ce qui engendre une importance qui ne fera que croître.

Dans le chapitre suivant, nous donnerons un état d'art des systèmes de recommandation sensible au contexte.

3. Etat de l'Art des Systèmes de Recommandation Sensibles au Contexte

3.1 Introduction

Afin que les systèmes de recommandation sensible au contexte puissent bien introduire le contexte, il existe des algorithmes conçus spécialement pour ce travail.

Nous présentons en bref dans ce chapitre quelques algorithmes proposés par des chercheurs pour chaque paradigme d'intégration de contexte.

3.2 Les algorithmes utilisés pour chaque paradigme

Comme nous venons de définir dans le chapitre précédent, les CARS sont classifiés en trois paradigmes (*Contextual pre-filtering*, *Contextual post-filtering*, *Contextual modeling*), et pour chaque paradigme, des algorithmes sont proposés pour améliorer la qualité de recommandations. Nous allons présenter juste quelques algorithmes utilisés dans l'état de l'art.

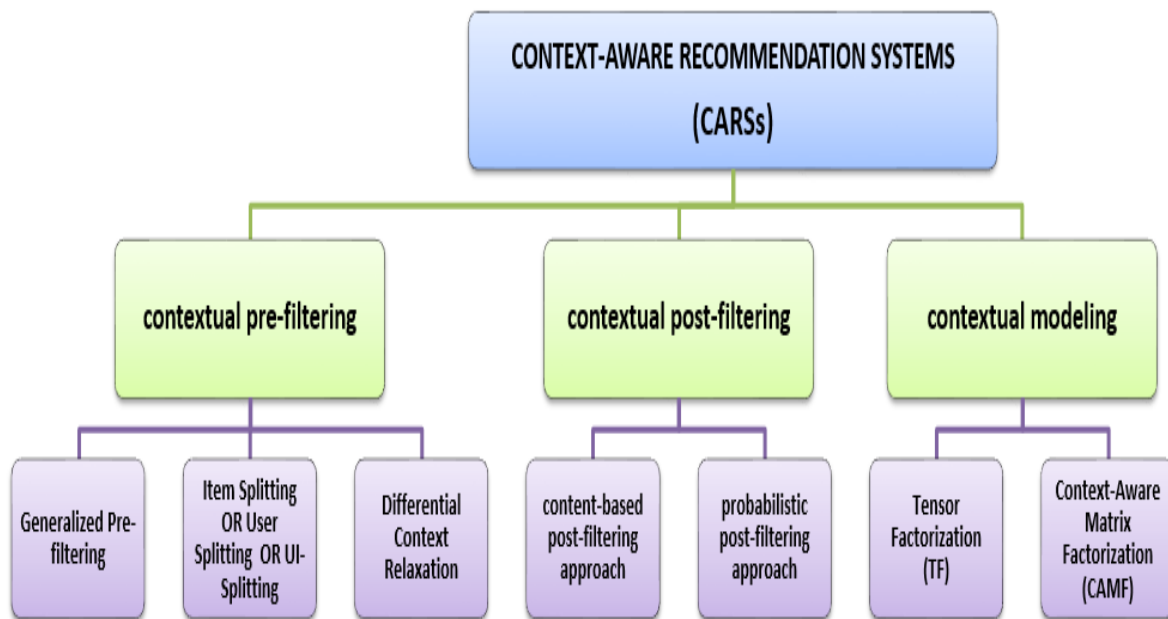


Figure 3-1: Les approches utilisées pour chaque approche

3.2.1 Approche Pré-filtrage contextuel

Le principe de cette approche est d'éliminer les données d'évaluation (rating data) qui ne sont pas pertinentes pour la situation ciblée. Les évaluations restantes sont utilisées pour apprendre un modèle local de prédiction d'évaluation et de recommandation.

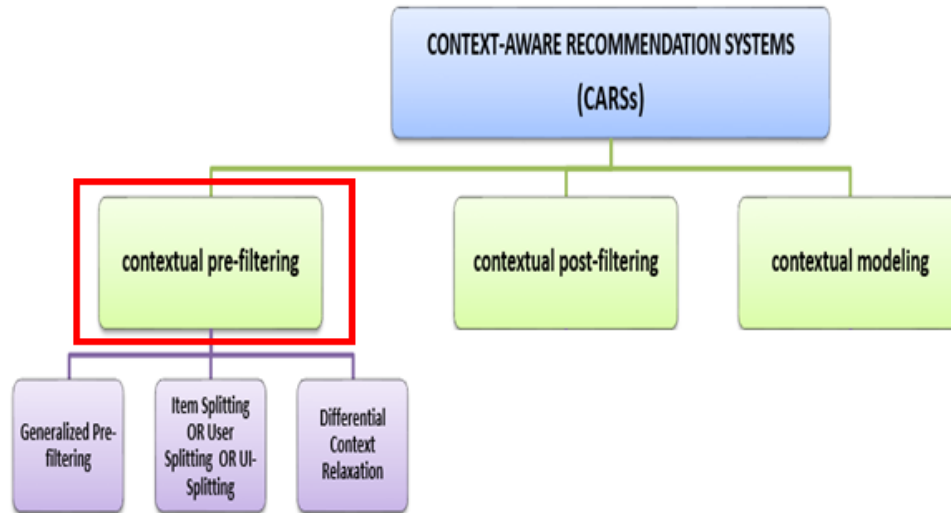


Figure 3-2: Les approches du pré-filtrage

Adomavicius et al. [34] ont proposé au début une première approche : *Exact Pre Filtering* qui consiste à appliquer strictement l'idée de pré-filtrage et construit un modèle de contexte local pour chaque situation ciblée, mais qui s'avère avoir des inconvénients, c'est à partir de là qu'est apparue l'approche du :

- * *Generalized Pre Filtering*: qui en premier, détermine d'abord l'agrégation optimale des évaluations marquées avec des situations contextuelles liées hiérarchiquement (c'est-à-dire des segments optimaux), puis construit une collection de modèles de prédiction en utilisant les évaluations appartenant à chaque segment.

Cependant, la performance de cette solution dépend de la taxonomie hiérarchique du contexte prédéfinie, qui peut ne pas diviser de manière optimale les situations contextuelles dans une collection de modèles locaux (un pour chaque segment) ayant conjointement la plus haute précision de prédiction de l'évaluation.

Une différente approche connue sous le nom d' :

- * *Item Splitting* : a été introduite par Baltrunas et Ricci [35]. L'idée est de diviser le vecteur d'évaluation (notation) d'un item donné en deux vecteurs d'items virtuels en utilisant un facteur contextuel spécifique.

Par exemple, l'ensemble complet des évaluations d'une piste musicale peut être divisé en deux ensembles, l'ensemble des évaluations pour la piste collectée lorsque

l'utilisateur était heureux et un autre ensemble qui a été acquis lorsque l'utilisateur n'était pas content (en supposant dans cet exemple que le bonheur est le Facteur contextuel).

Baltrunas et Amatriain [36] ont proposé une variante de l'approche précédente appelée :

- * **User Splitting** : qui au lieu de diviser les items, divise les utilisateurs en plusieurs sous-profils, chacun représentant l'utilisateur dans un contexte particulier.

Enfin Zheng et al. [37] ont combiné les deux approches pour donner le :

- * **UI Splitting** : qui a donné une meilleure précision de prédiction dans un problème de recommandation de film.

Ils ont aussi proposé le :

- * **Differential Context Relaxation** : qui supprime les facteurs contextuels de faible pertinence pour un composant de fonction donné, en combinaison avec le traditionnel filtrage collaboratif.

3.2.2 Approche Post-filtrage contextuel

Le post-filtrage contextuel exploite l'information contextuelle pour rejeter certaines recommandations (non pertinentes), après qu'un ensemble de recommandations initiales est déterminé par un modèle prédictif sans contexte.

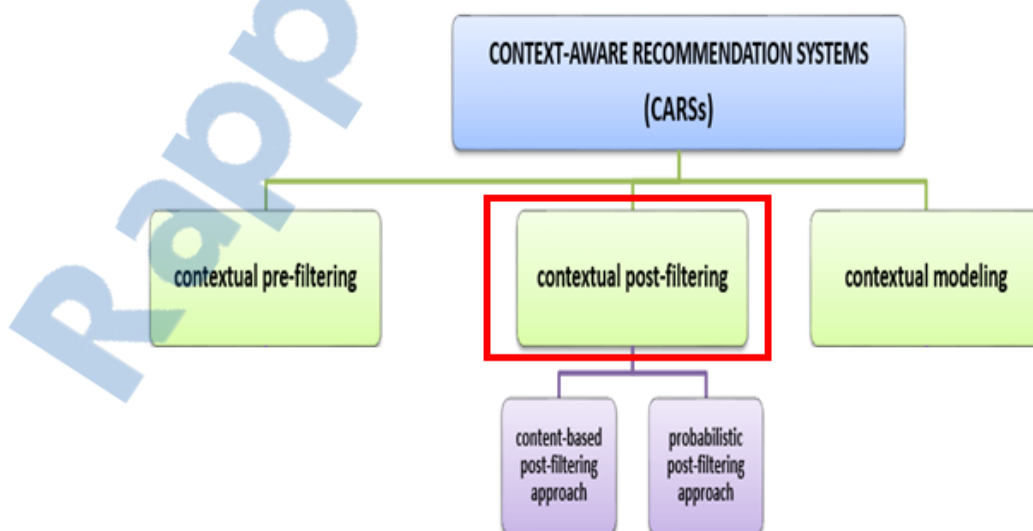


Figure 3-3: Les approches du post-filtrage

L'approche optée par Panniello et al. [32] est

- * **Probabilistic post-filtering** : qui estime d'abord la probabilité qu'un item soit pertinent pour l'utilisateur dans un contexte donné, puis utilise ces probabilités pour pénaliser les items estimés non pertinents pour le contexte cible.

Hayes et Cunningham [38] ont à leur tour présenté l'approche

- * **Content-based post-filtering** qui se concentre sur la recherche de fonctionnalités communes des items (par exemple, les acteurs préférés à surveiller) pour un utilisateur donné dans un contexte donné, puis utilise ces fonctionnalités pour ajuster les recommandations.

3.2.3 Modélisation contextuelle

Les approches basées sur la modélisation contextuelle étendent les modèles prédictifs sans contexte en ajoutant généralement de nouveaux paramètres de modèle qui représentent l'information contextuelle.

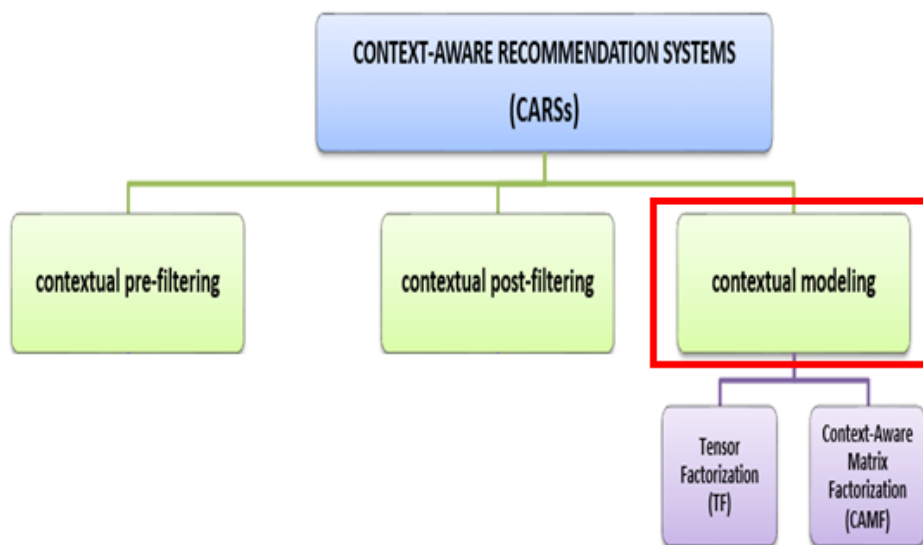


Figure 3-4: Les approches de la modélisation contextuelle

Deux approches majeures basées sur l'extension des techniques de factorisation matricielle (MF) ont été proposées :

- * **Tensor Factorisation (TF)** : Elle consiste à étendre le problème MF bidimensionnel en une version multidimensionnelle, où le tenseur de notation est pris en compte dans un espace vectoriel de dimension inférieure. De cette façon, les interactions entre les utilisateurs, les items et les facteurs contextuels sont représentés sous forme de vecteurs de facteurs latents.

Plusieurs auteurs ont proposé des variantes de cette approche : certaines d'entre elles sont optimisées pour la prédiction de l'évaluation, telles que la Recommandation Multiverse (Karatzoglou et al. [39]) et les Machines de factorisation (Rendle et al. [40]), tandis que d'autres sont optimisés pour une recommandation fondée sur le classement, Tels que iTALS (Hidasi et Tikk [41]), et TFMAP (Shi et al. [42]). La principale limitation de TF est sa complexité de calcul.

En fait, le nombre de paramètres de modèle à apprendre augmente de façon exponentielle avec le nombre de facteurs contextuels.

- * **Context-Aware Matrix Factorization (CAMF)** : Est une approche de modélisation contextuelle plus évolutive proposée par Baltrunas et al. [43] CAMF étend MF en utilisant des prédicteurs de base contextuels pour représenter les interactions de l'information contextuelle avec les éléments ou les utilisateurs.

De cette façon, un plus petit nombre de paramètres, par rapport à TF, est utilisé. Il généralise le prédicteur de référence qui a été initialement proposé par Koren [44] pour intégrer la dynamique temporelle associée aux données de notation. Cette technique s'est révélée efficace lorsque combiné avec MF sur l'ensemble de données Netflix (Koren et Bell [45]; Campos et al. [46]).

3.3 Conclusion

Nous avons présenté quelques algorithmes incorporant le contexte dans le processus de la recommandation.

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

4. Conception et Réalisation du Système MovieRec_FM

4.1 Introduction

Après la réalisation de notre état de l'art, nous avons remarqué que la majorité des travaux de recherche utilisent l'algorithme de la factorisation matricielle de l'approche filtrage collaboratif dans l'étape d'expérimentation.

Dans ce chapitre, nous allons présenter l'avantage de cet algorithme, puis nous passons à la conception de notre application « MovieRec_FM » en utilisant le langage de modélisation UML.

Nous allons appliquer cet algorithme sur une base de données réelle sur les films comportant 700 utilisateurs et 9000 Items.

Nous terminons par la description des différents outils utilisés et par une description de notre application.

4.2 Architecture du système MovieRec FM

Pour la réalisation de notre système, nous nous sommes basé sur la factorisation matricielle de l'approche basée sur les modèles des systèmes de recommandations collaboratifs. Elle a fait ces preuves après le prix donné par Netflix¹ pour avoir réussi à améliorer son système de recommandation [47].

Nous citons dans ce qui suit quelques travaux de recherche du domaine des systèmes de recommandation, qui affirme l'avantage de factorisation matricielle lors de l'étape d'expérimentation à savoir :

- ✓ Koren et al. [13] disent que c'est une méthode qui permettra d'accélérer les divers processus mis en place au sein des systèmes de recommandations.
- ✓ [48,49] affirment que FM est la méthode qui fournies l'une des approches les plus simples et les plus efficaces pour les systèmes de recommandation.
- ✓ Zheng et al. dans [50] disent que la FM est l'algorithme de recommandation le plus performant, car les approches du facteur latent sert à bien pallier le problème de rareté (sparsity) mieux que les approches du filtrage collaboratif basées sur la mémoire (plus proche voisin).

¹ <https://www.netflix.com>

Notre objectif est de tirer les avantages de cette méthode en prenant en considération l'information contextuelle dans le processus de recommandation.

L'architecture de notre système est la suivante :

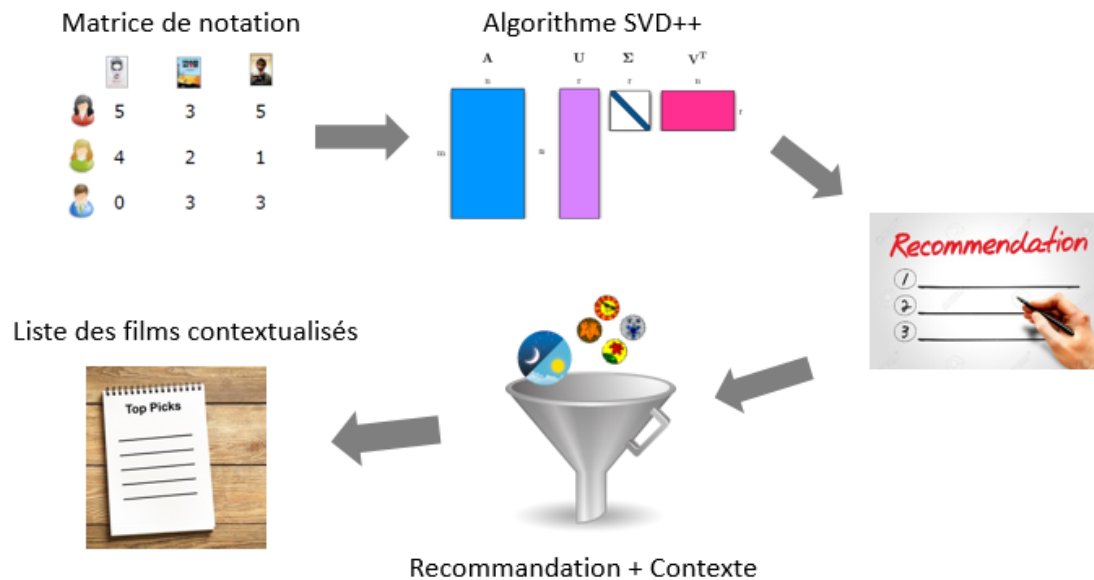


Figure 4-1: Architecture du système MovieRec_FM

4.3 Conception du Système MovieRec_FM

Pour bien décrire les différentes étapes de conception de notre système, nous avons choisi le langage de modélisation UML. [51]

UML : c'est l'acronyme Anglais pour « Unified Modeling Language ». Traduit par « Langage de modélisation unifié ».

UML est constitué d'un ensemble de boîte à outils appelée diagrammes, chacun permet de représenter une vision du système à implémenter.

Parmi les diagrammes d'UML, nous allons utiliser le diagramme de cas d'utilisation et le diagramme de séquence.

4.3.1 Diagramme de cas d'utilisation

Un diagramme de cas d'utilisation capture le comportement d'un système, d'un sous-système, d'une classe ou d'un composant tel qu'un utilisateur extérieur le voit.

L'utilisateur va se connecter pour demander une proposition de liste de films à voir. Cette liste va prendre en considération le contexte de l'utilisateur qu'elle va être extraite d'une manière implicite à savoir (heure, jour/nuit, saison) pour lui recommander des films qui correspondent à sa situation

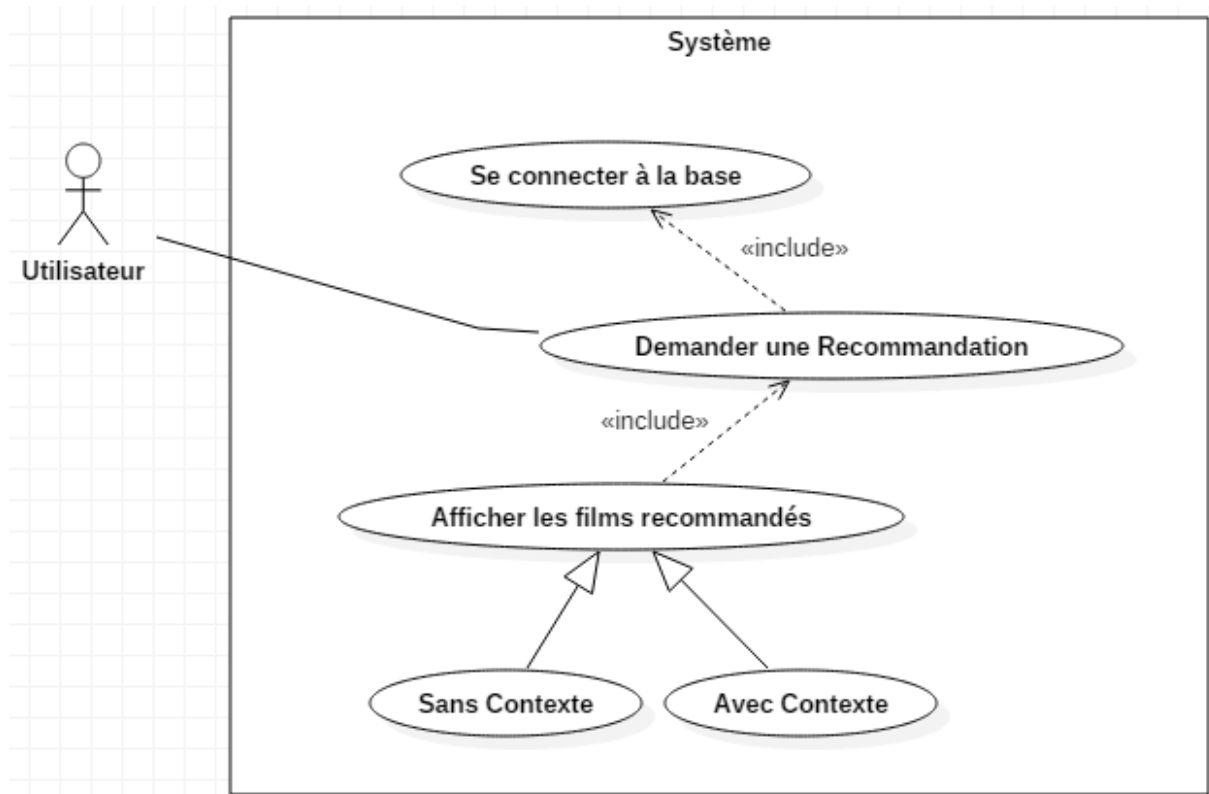


Figure 4-2: Diagramme de cas d'utilisation

4.3.2 Diagramme de séquence

Le diagramme de séquence permet de montrer les interactions d'objets dans le cadre d'un scénario d'un diagramme des cas d'utilisation.

La figure ci-dessous montre les différentes interactions entre l'utilisateur, le système MovieRec_FM et la base de données.

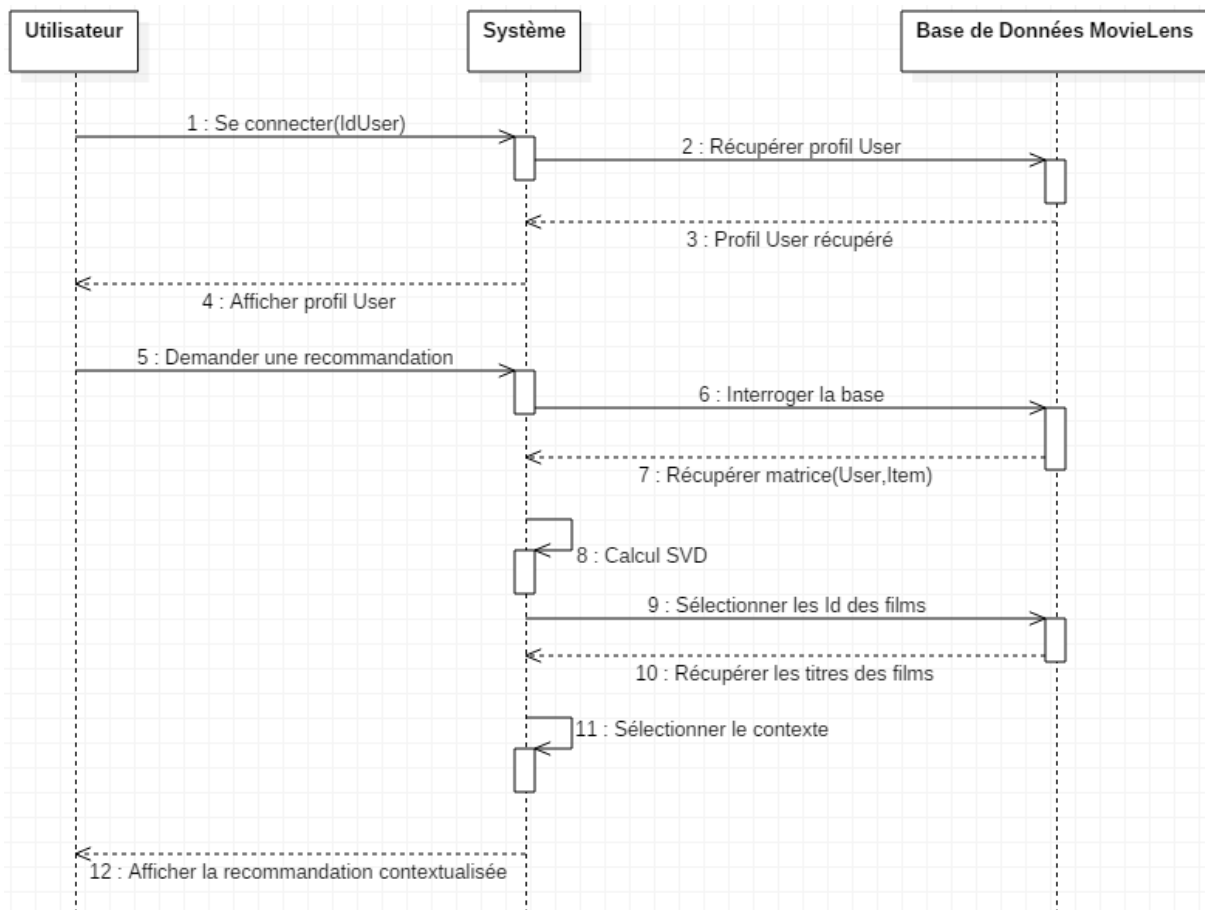


Figure 4-3: Diagramme de séquence

4.4 Collection de Données

Afin de pouvoir appliquer notre démarche, nous avons besoin d'un jeu de données contenant les utilisateurs, les items à recommander et les ratings donnés pour ces items. Pour cela nous allons utiliser la base MovieLens² ([ml-latest-small](http://www.movielens.org/)) qui a été collectée par le projet de recherche GroupLens à l'Université de Minnesota. Cette base contient les informations de 9000 films, 700 utilisateurs et 100000 votes.

Chaque utilisateur a voté au moins 5 films. La valeur d'un vote est entre 1 et 5 et chaque utilisateur a noté au moins 20 films.

Les fichiers principaux de la base MovieLens sont les suivants : rating.csv et movies.csv.

²<http://www.movielens.org/>

Les informations des films (items) sont contenues dans le fichier movies.csv comme suit :

Champs	Description
movieId	C'est l'identifiant du film (Il est unique pour chaque film)
title	Contient le titre de chaque film.
genres	Comporte le genre du film (ex : Drama, Comédie, Aventure...)

Table 4-1: Tableau de description des champs du fichier movies

Le fichier rating.csv contient des votes créés par des utilisateurs sur des films (items). C'est une liste dont les champs sont :

Champs	Description
userId	L'identifiant de chaque utilisateur.
movieId	C'est l'identifiant du film.
rating	La note donnée par un utilisateur à un film.
timestamp	Représente le nombre de secondes qui sont écoulé depuis le 1 ^{er} Janvier 1970 (ex : 26 mai 2016 à 22 :35 :45, le timestamp est "1464294945").

Table 4-2: Tableau de description des champs du fichier ratings

4.5 Exemple d'application de FM

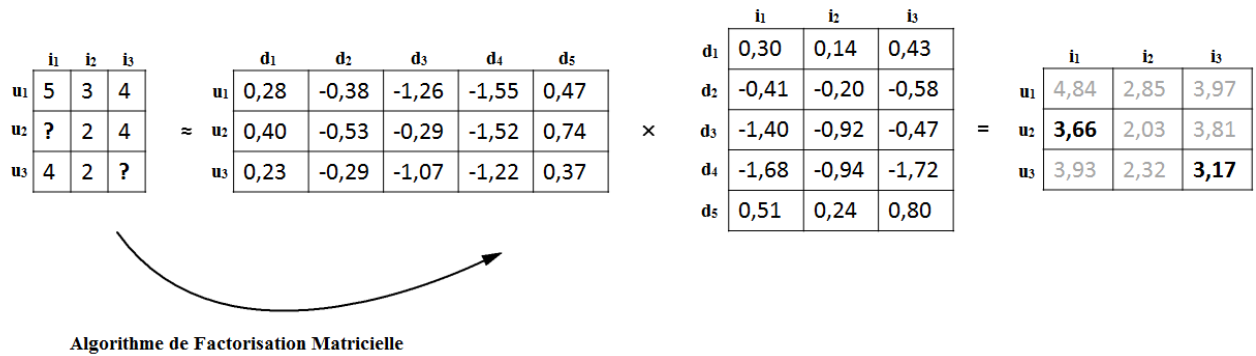


Figure 4-4: Exemple de la factorisation matricielle

Nous proposons un exemple simple de la matrice de notes qui représente trois utilisateurs pouvant voter trois films. Cette matrice U (3*3) qui contient toutes les notes que les utilisateurs (u₁,u₂,u₃) ont attribuées aux articles (i₁,i₂,i₃).

Nous remarquons qu'il y a deux valeurs manquantes dans la matrice. Donc après l'application de l'algorithme de la factorisation matricielle sur la matrice U. nous obtenons les deux matrices P (3* 5) et Q (5*3) et après avoir fait le produit entre ces derniers nous obtenons comme output les ratings prédictions donc le rating P (u₂, i₁) ≈4 et P (u₃, i₃) ≈3.

A travers cet exemple nous avons pu montrer comment nous pouvons calculer le rating d'un utilisateur pour un film donné en utilisant la factorisation matricielle.

4.6 Outils d'Implémentation

Pour le développement de notre application nous avons utilisé plusieurs outils :

Java : est un langage de programmation orienté objet, développé par Sun Microsystems. Il permet de créer des logiciels compatibles avec de nombreux systèmes d'exploitation (Windows, Linux, Macintosh, Solaris). Java donne aussi la possibilité de développer des programmes pour téléphones portables et assistants personnels.

Pour supporter le langage Java, nous avons choisi l'outil NetBeans.

NetBeans : est un environnement de développement intégré (IDE) open source pour Java, PHP, C ++ et d'autres langages de programmation.

NetBeans est également appelé une plate-forme de composants modulaires utilisés pour développer des applications de bureau Java. Il utilise des composants, également appelés modules, pour permettre le développement de logiciels.

NetBeans installe dynamiquement des modules et permet aux utilisateurs de télécharger des fonctionnalités mises à jour et des mises à niveau authentifiées numériquement.

Les modules NetBeans IDE comprennent NetBeans Profiler, un outil de conception d'interface utilisateur graphique (GUI). [52]

Et enfin pour travailler avec l'algorithme de SVD, nous avons utilisé la bibliothèque EJML.

Efficace Java Matrix Library (EJML) [53] : est une bibliothèque d'algèbre linéaire Java pour manipuler des matrices denses. Ses objectifs de conception sont :

- Etre aussi efficace que possible pour les matrices petites et grandes
- Etre accessible aux novices et aux experts. EJML est gratuit, écrit en 100% Java et a été publié sous la licence Apache v2.0.

EJML offre les fonctionnalités suivantes pour les matrices denses :

- Manipulation de matrice (extraire, insérer, combiner, ...)
- Résolveurs linéaires (linéaires, les moindres carrés, incrémentaux, ...)
- Décomposition (LU, QR, Cholesky, SVD, Valeur propre, ...)
- Caractéristiques de la matrice (rang, symétrique, définitive, ...)
- Matrices aléatoires (covariance, orthogonal, symétrique, ...)
- Différents formats internes (row-major, block)
- Test unitaire

4.7 Description de l'Application

Dans cette section, nous allons présenter l'application qui a été implémentée afin de bien étudier le rôle de la factorisation dans les systèmes de recommandation. Elle est dotée d'une interface graphique facile à utiliser.

La figure ci-dessous présente l'interface graphique et la fenêtre principale de notre application. Le bouton login permet à un utilisateur donné de se connecter au système après avoir saisi son UserID.



Figure 4-5: L'interface de l'application MovieRec_FM

Pour illustrer un exemple, nous allons prendre l'utilisateur 20, après avoir entré le login, l'interface profil User va s'afficher.

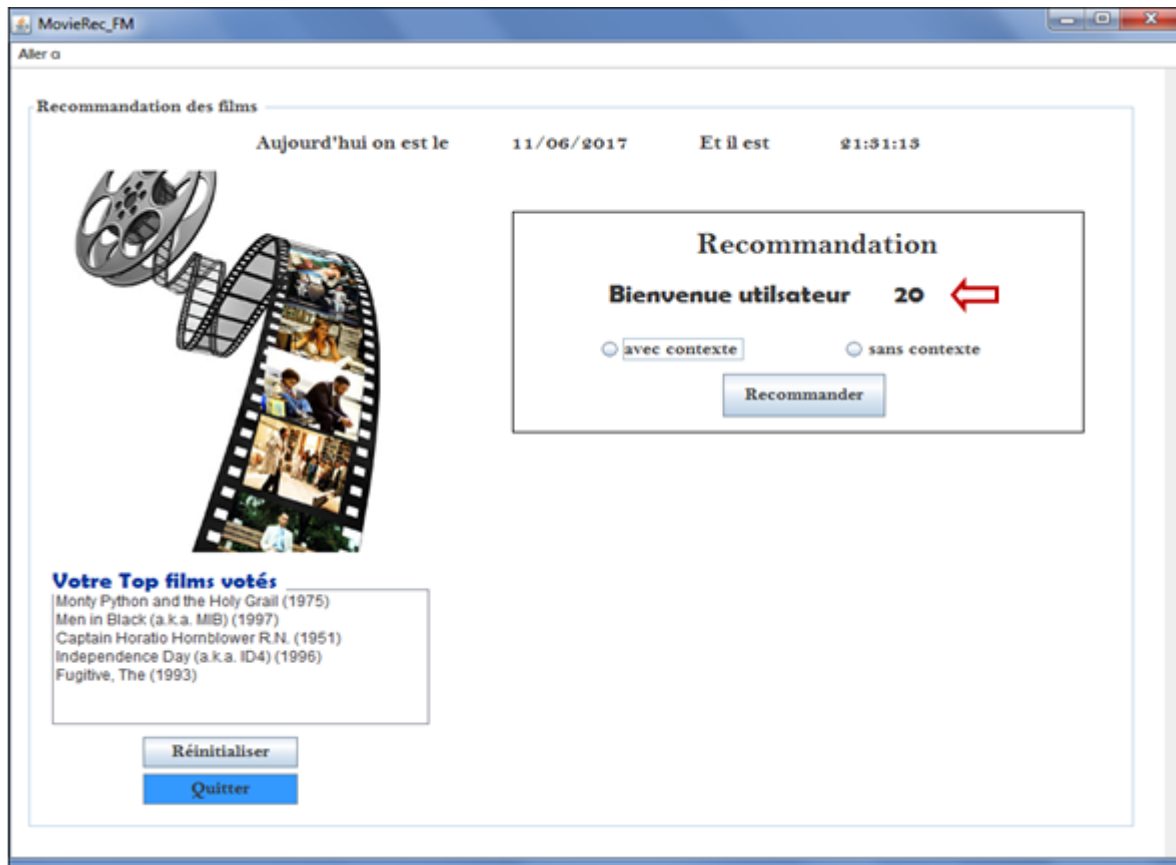


Figure 4-6: Profil User de l'utilisateur 20

Dans l'interface profil User les cinq tops films déjà votés par l'utilisateur 20 vont être affichés en bas à gauche. Après avoir faire le choix entre une recommandation avec contexte ou sans contexte le bouton Recommander permet de retourner la liste des films recommandés après avoir calculé les ratings en utilisant l'algorithme SVD.

Si nous choisissons une recommandation avec contexte le système retournera ensuite une liste des films pour l'utilisateur selon l'heure (**jour, nuit**) et aussi selon la saison (**printemps, été, automne, hiver**). Sinon le système retournera une recommandation sans contexte.

Recommandation avec contexte :

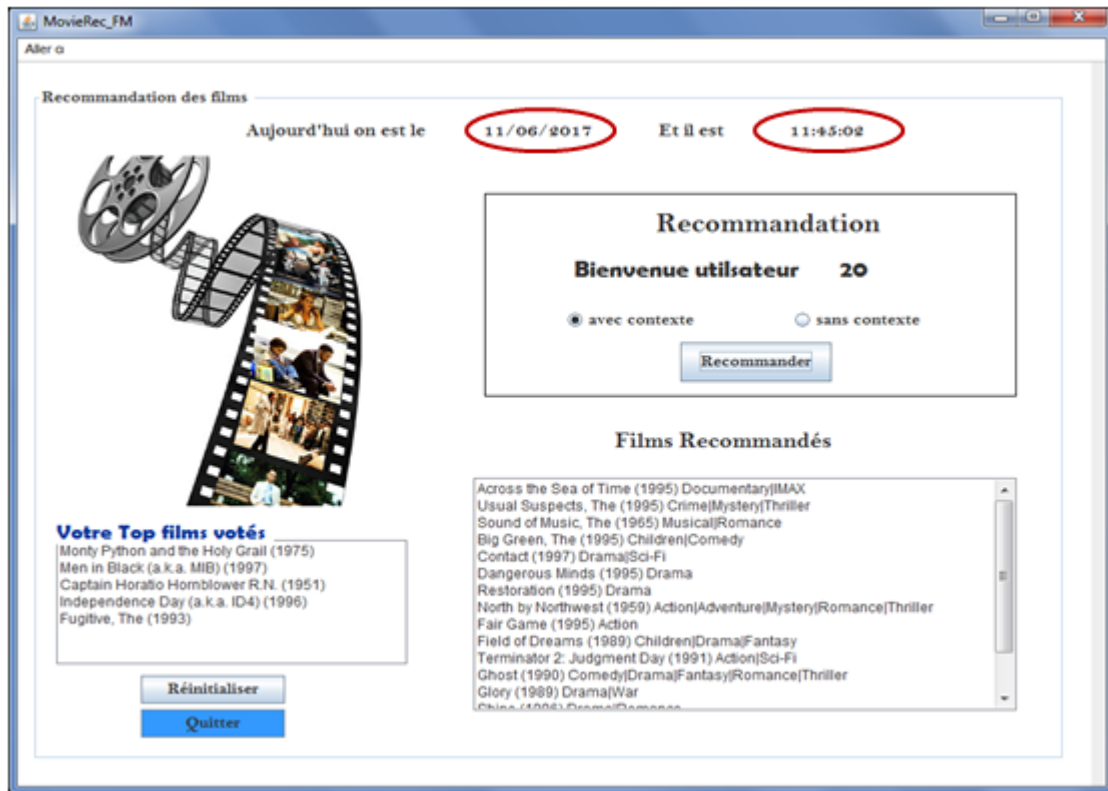


Figure 4-7: Recommandation jour/printemps

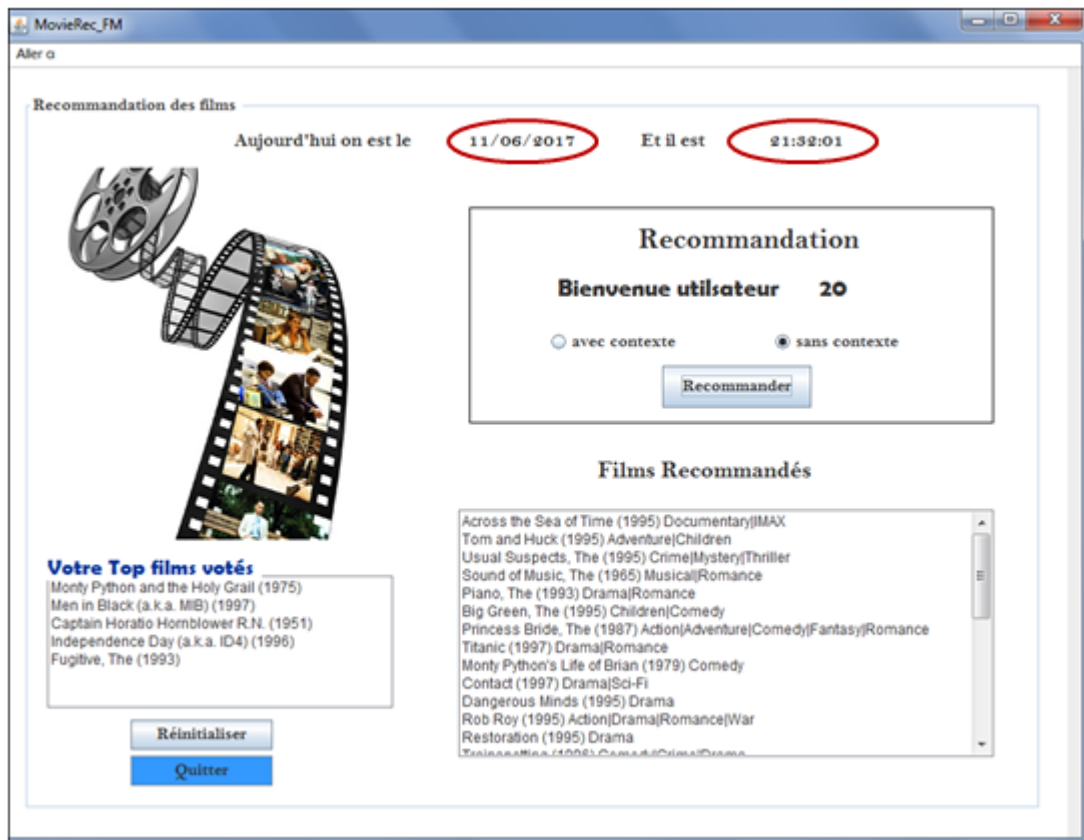


Figure 4-8: Recommandation nuit/printemps

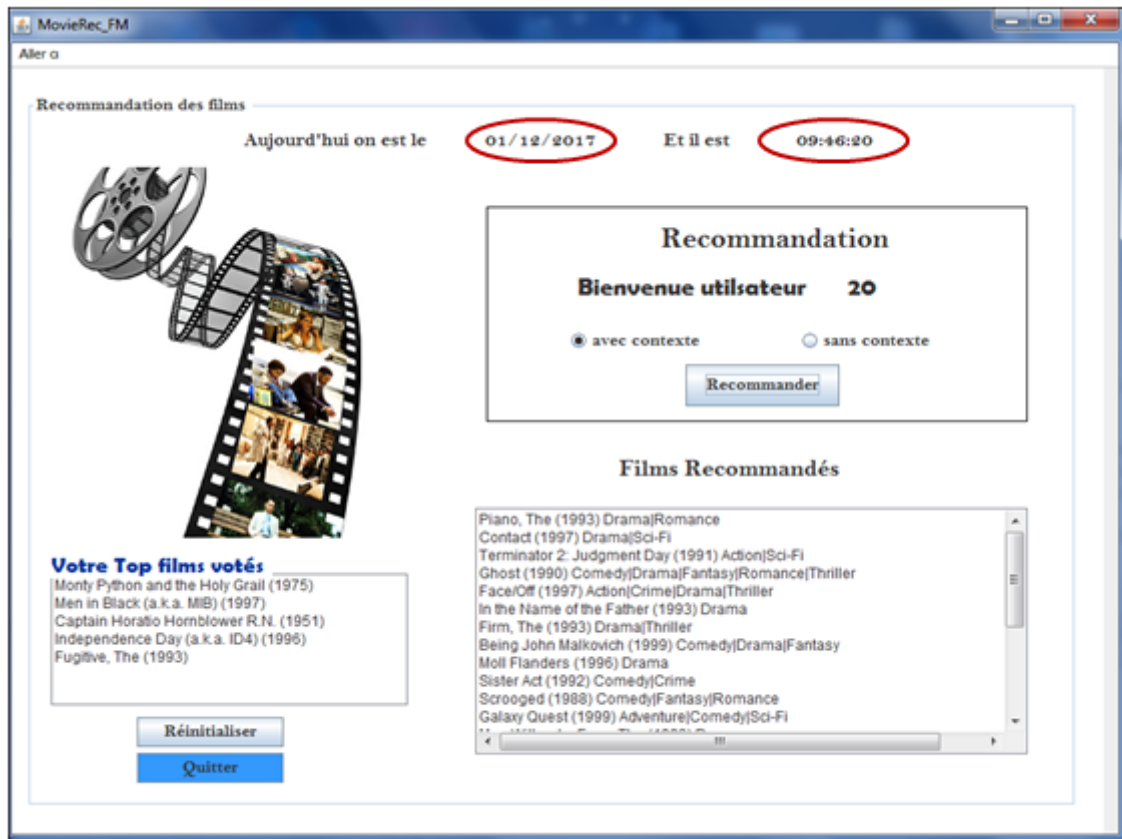


Figure 4-9: Recommandation jour/hiver

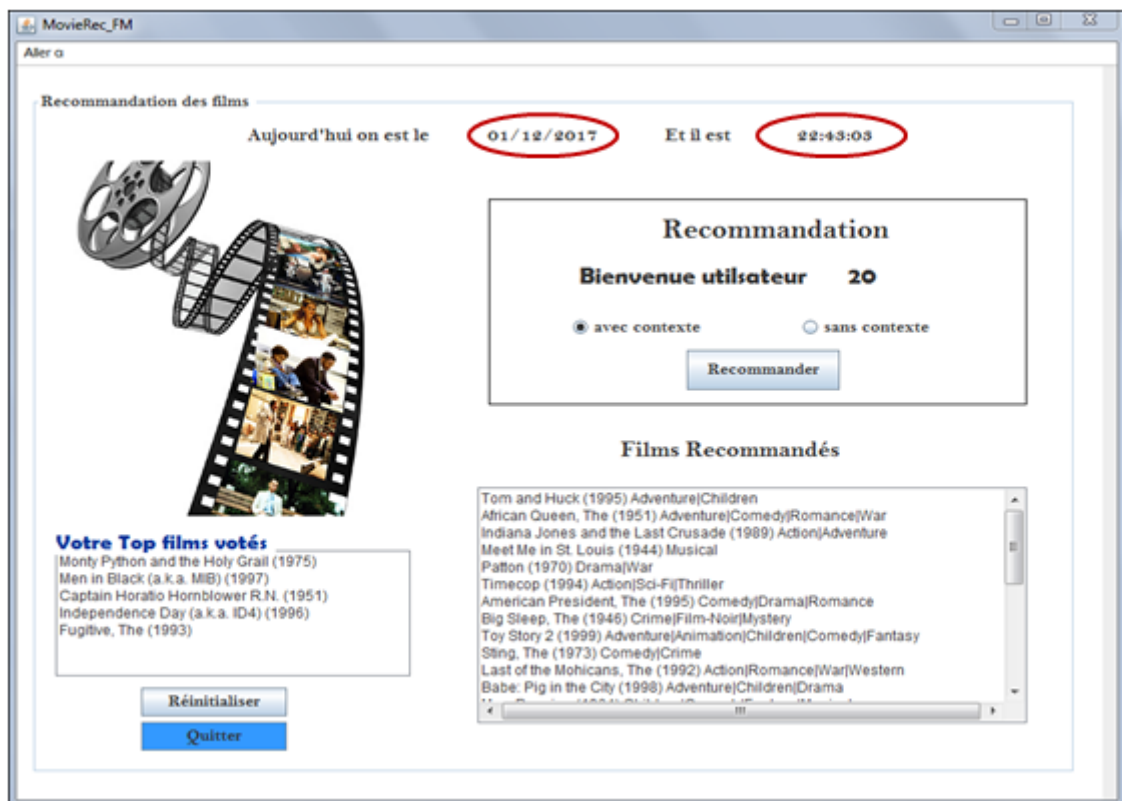
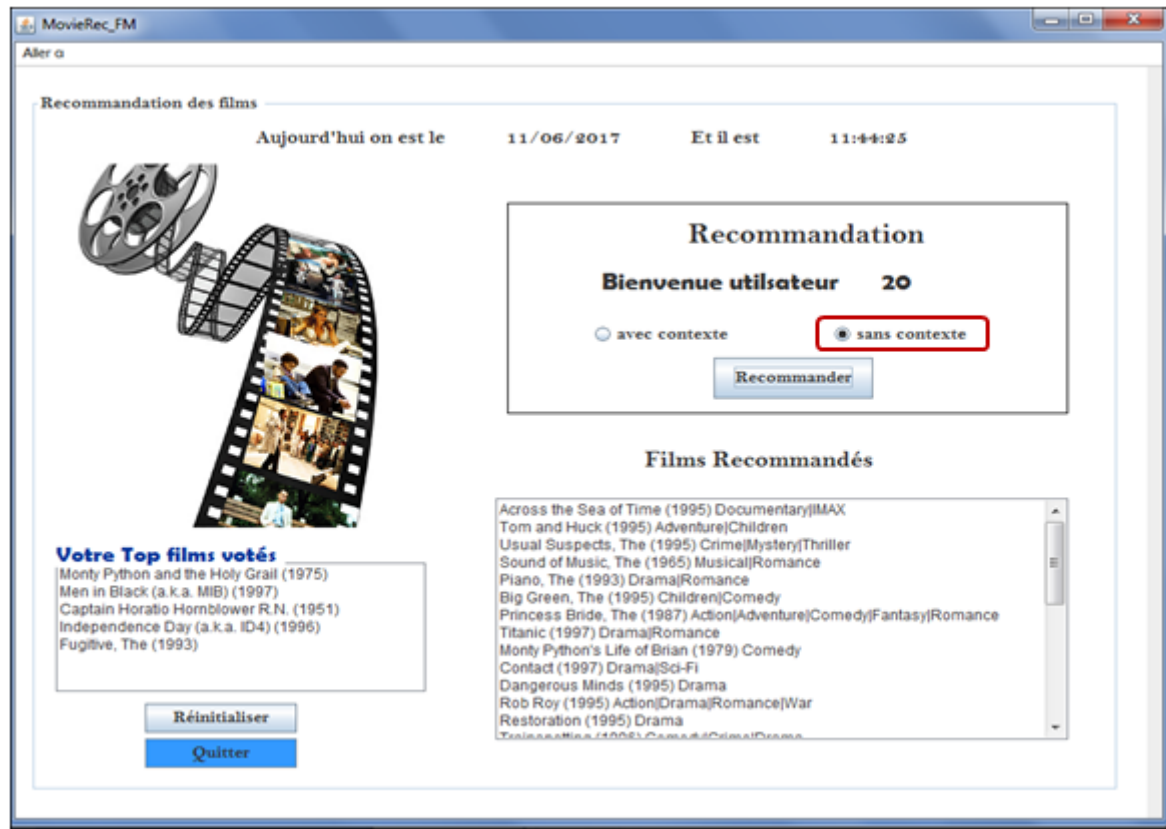


Figure 4-10: Recommandation nuit/hiver

Recommandation sans contexte :**Figure 4-11: Recommandation sans contexte**

Nous avons fait plusieurs expérimentations en effectuant des changements sur l'heure pour voir la différence entre les films recommandés, ensuite nous avons changé le mois pour voir la liste recommandés pour le même utilisateur dans une autre saison ici nous avons choisi les deux saisons printemps et l'hiver. Et pour finir nous avons fait une recommandation sans contexte pour faire la comparaison entre les deux résultats.

4.8 Conclusion

Dans ce dernier chapitre, nous avons exposé et expliqué les différentes phases pour la conception et la réalisation de notre système de recommandation sensible au contexte, qui consiste à recommander une liste pertinente de films à un utilisateur donné, en utilisant le filtrage collaboratif et le modèle de factorisation matricielle « La décomposition de la valeur singulière (SVD) ».

Donc nous pouvons conclure que SVD est capable de traiter efficacement un grand ensemble de données, et augmenter la précision et la performance de prédiction.

*CONCLUSION
GENERALE
ET PERSPECTIVES*

Conclusion générale

Le domaine des systèmes de recommandation a été révolutionné en passant des systèmes de recommandation classiques aux systèmes de recommandations sensibles au contexte. La particularité de ces derniers est leur adaptabilité à l'environnement. Cela est fait par l'intégration du contexte qui prend en compte l'information contextuelle (L'humeur, le temps, l'emplacement... d'un utilisateur) pour effectuer une recommandation.

Par la suite, l'ajout de la factorisation matricielle a ses systèmes a donné de meilleurs et précis résultats. C'est ce qui a été montré dans le Netflix Prize.

Dans notre travail, nous avons mis en place une application de recommandation des films basé premièrement sur l'intégration du contexte, et plus précisément la seconde approche d'intégration : le post filtrage contextuel qui est exécuter après le module principal de recommandation, et deuxièmement sur la factorisation matricielle et plus précisément en utilisant une de ses méthodes qui est la décomposition en valeur singulière « SVD »

Nous pouvons conclure que le contexte et la factorisation matricielle ont joué un rôle important dans les systèmes de recommandation sensibles au contexte. De plus, la factorisation matricielle a non seulement réduit la taille de la matrice ce qui implique un temps de réponse réduit, mais aussi amélioré le rendement de la recommandation.

Comme perspectives nous envisageons d'apporter quelques améliorations à savoir :

- Nous souhaitons combiner la méthode de factorisation matricielle et l'approche "kNN" (k Nearest Neighbors) dans le but d'avoir de meilleurs résultats et d'améliorer encore plus la performance de recommandations.
- Intégrer le contexte « localisation » de l'utilisateur (à la maison, au travail) pour but de lui fournir des suggestions pertinentes selon l'endroit où il se trouve.

Références Bibliographiques

- [1] Ricci, F., Rokach, L., Shapira, B., and Kantor, P. B., *Recommender Systems Handbook*. Springer, 2011.
- [2] Adomavicius, G. Tuzhilin, A., *Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions*. IEEE Trans. on Knowl. and Data Eng., 2005.
- [4] Soualah-Alila, F., *CAMLearn: Une Architecture de Système de Recommandation Sémantique Sensible au Contexte*. Application au Domaine du MLearning. Informatique [cs]. 2015.
- [5] Pazzani, M. and Billsus, D., *Content-based recommendation systems*. In *The Adaptive Web: Lecture Notes in Computer Science*, 2007.
- [6] Magnini, B. and Strapparava, C., *Experiments in word domain disambiguation for parallel texts*. In *Proceedings of the ACL-2000 Workshop on Word Senses and Multi-linguality*, 2000.
- [7] Herlocker, J. L., Konstan, J. A., and Riedl, J., *Explaining collaborative filtering recommendations*. In *Proceedings of the 2000 Conference on Computer Supported Cooperative Work*, 2000.
- [8] Dahimene, M. R., *Filtrage et Recommandation sur les Réseaux Sociaux*, Thèse de Doctorat, 2014.
- [9] Esslimani, I., *Vers une approche comportementale de recommandation : apport de l'analyse des usages dans un processus de personnalisation*. Interface homme-machine [cs.HC], 2010.
- [10] Bokde D., Girase S., Mukhopadhyay D., *Role of Matrix Factorization Model in Collaborative Filtering Algorithm: A Survey*. IJAFRC, 2014.
- [11] Burke, R., *Hybrid Recommender Systems: Survey and Experiments*. User Modeling and User-Adapted Interaction, 2002.
- [12] Naak, A., *Papyrus : Un système de gestion et de recommandation d'articles de recherche*. Département d'Informatique et de Recherche Opérationnelle Faculté des arts et des sciences, 2009.
- [13] Koren, Y., *Matrix Factorization Techniques for Recommender Systems*. Published by the IEEE Computer Society, 2009.
- [14] Sarwar B., Karypis G., Konstan J. and Riedl J., *Incremental Singular Value Decomposition Algorithms for Highly Scalable Recommender Systems*. GroupLens Research Group, 2000.

- [15] Ricci, F., Lior, R., Bracha, S. and Paul B. K., *Recommender Systems Handbook*. Springer, 2011.
- [16] Salakhutdinov, R., Andriy, M., *Probabilistic Matrix Factorization*. Published in the Proceedings of Neural Information Processing Systems Foundation (NIPS'07), 2007.
- [18] Schilit B., Adams N. and Want R., *Context-aware computing applications*. Workshop on Mobile Computing Systems and Applications, 1994.
- [19] Ryan N. S, Pascoe J. and Morse D. R., *Enhanced reality fieldwork: the context-aware archeological assistant*. Gaffney, Leusen and Exxon edition, 1997.
- [20] Rey, G., Coutaz, J., *Le Contexteur : une abstraction logicielle pour la réalisation de systèmes interactifs sensibles au contexte*. In Proceedings of Interaction Homme Machine (IHM2002), 2003.
- [21] Dey, K. A., Abowd, D. G., *Towards a better understanding of context and contextawareness*. In Computer Human Interactions (CHI2000) Workshop on the What, Who, Where and How of ContextAwareness. 2000.
- [22] Chaari T., *Adaptation d'applications pervasives dans des environnements multi-contextes*. Thèse de doctorat, 2007.
- [23] Chaari, T., Lafort, F., and Flory, A., *Adaptation des applications au contexte en utilisant les services web : Le projet secas*. In La Deuxième Journées Francophones Mobilité et Ubiquit, 2005.
- [24] Henricksen K. and Indulska J., *Developing context-aware pervasive computing applications: Models and approach*. Journal of Pervasive and Mobile Computing, 2006.
- [25] Bettini C., Brdiczka O., Henricksen K., Indulska J., Nicklas D., Ranganathan A., and Riboni D., *A survey of context modelling and reasoning techniques*. Journal of Pervasive and Mobile Computing, 2010.
- [26] Chen, G. and Kotz, D., *A survey of context-aware mobile computing research*. Technical report, 2000.
- [27] Petrelli, D., Not, E., Strapparava, C., Stock, O., and Zancanaro, M., *Modeling context is like taking pictures*. In Conference on Human Factors in Computers, Workshop "The What, Who, Where, When, Why and How of Context-Awareness". 2000.
- [28] Amara-Hachimi N. and El Fallah-Seghrouchni A., *Modélisation d'informations contextuelles pour des agents mobiles sensibles au contexte*. Villemur T., Saqui-Sannes P., and Drira K. Eds , 2006.
- [29] Dey, A., *Providing Architectural Support for Building Context-aware Applications*. PhD thesis, 2000.
- [30] Pham, C. *Conception d'un système d'apprentissage et de travail pervasif et adaptatif fondé sur un modèle de scénario*. PhD thesis, 2010.

- [31] El Ghayam, Y., *La sensibilité au Contexte dans un Environnement Mobile*. Thèse de doctorat, 2011.
- [32] Panniello, U., Tuzhilin, A., Gorgoglione, M., Palmisano, C., Pedone, A., *Experimental comparison of prevs. post-filtering approaches in context-aware recommender systems*. In: Proceedings of the 3th ACM conference on Recommender systems (RecSys'09), 2009.
- [33] Adomavicius, G., Tuzhilin, A., *Context-aware recommender systems*. In: Recommender Systems Handbook, 2011.
- [34] Adomavicius, G., Sankaranarayanan, R., Sen, S., Tuzhilin, A., *Incorporating contextual information in recommender systems using a multidimensional approach*. ACM Trans. Inf. Syst. 23(1), 2005.
- [35] Baltrunas, L., Ricci, F., *Context-dependent items generation in collaborative filtering*. In: Proceedings of the 3th ACM conference on Recommender system (RecSys'09), 2009.
- [36] Baltrunas, L., Amatriain, X., *Towards time-dependant recommendation based on implicit feedback*. In: Proceedings of the 1st Workshop on Context-Aware Recommender Systems (CARS'09), 2009.
- [37] Zheng, Y., Burke, R., Mobasher, B., *Recommendation with differential context weighting*. In: Proceedings of the 21th international conference on user modeling, adaptation, and personalization (UMAP'13), 2013a.
- [38] Hayes, C., Cunningham, P., *Context boosting collaborative recommendations*. Knowl. Based Syst, 2004.
- [39] Karatzoglou, A., Amatriain, X., Baltrunas, L., Oliver, N., *Multiverse recommendation: N-dimensional tensor factorization for context-aware collaborative filtering*. In: Proceedings of the 4th ACM conference on Recommender systems (RecSys'10), 2010.
- [40] Rendle, S., Gantner, Z., Freudenthaler, C., Schmidt-Thieme, L., *Fast context-aware recommendations with factorization machines*. In: Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval (SIGIR '11), 2011.
- [41] Hidasi, B., Tikk, D., *Fast ALS-Based Tensor Factorization for Context-Aware Recommendation*. In: Proceedings of the 2012 European conference on Machine Learning and Knowledge Discovery in Databases (KDD'12), 2012.
- [42] Shi, Y., Karatzoglou, A., Baltrunas, L., Larson, M., *TFMAP: optimizing MAP for top-N context-aware recommendation*. In: Proceedings of the 35th international ACM SIGIR conference on Research and development in Information Retrieval (SIGIR '12), 2012.
- [43] Baltrunas, L., Ludwig, B., Ricci, F., *Matrix Factorization Techniques for Context Aware*. In: Proceedings of the 5th ACM conference on Recommender systems (RecSys'11), 2011.
- [44] Koren, Y., *Collaborative filtering with temporal dynamics*. Commun. ACM 53(4), 2010.

- [45] Koren, Y., Bell, R., *Advances in collaborative filtering*. In: Recommender Systems Handbook, 2011.
- [46] Campos, P., Díez, F., Cantador, I., *Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols*. User Model User-Adapt. Interact, 2014.
- [47] Bell, R. M. and Koren, Y., *Lessons From the Netflix Prize Challenge*. SIGKDD Explor, 2007.
- [48] Salakhutdinov, R. and Mnih, A., *Bayesian probabilistic matrix factorization using markov chain monte carlo*. In Proceedings of the 25th international conference on Machine learning, ICML 08, 2008.
- [49] Salakhutdinov, R., and Mnih, A., *Probabilistic matrix factorization*. In NIPS, 2007.
- [50] Zheng Y., Burke R., Mobasher B., *Splitting approaches for context-aware recommendation: an empirical study*. Proceedings of the 29th Annual ACM Symposium on Applied Computing, 2014

Références Webographiques

- [3] Mathieu, *Les algorithmes de recommandation*,
<http://www.podcastscience.fm/dossiers/2012/04/25/les-algorithmes-de-recommandation/>
Consulté le 11 mars 2017.
- [17] Park S., Chu W., *Pairwise Preference Regression for Cold-start Recommendation*.
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.211.9762&rep=rep1&type=pdf>,
Consulté le 20 mars 2017.
- [51] Roels, C., *UML, c'est quoi ?*. <https://openclassrooms.com/courses/debutez-l-analyse-logicielle-avec-uml/uml-c-est-quoi#/id/r-2048966>. Consulté le 12 avril 2017.
- [52] Technopedia Inc. *NetBeans*. <https://www.techopedia.com/definition/24735/netbeans>.
Consulté le 12 avril 2017.
- [53] EJML. *Main page*. http://ejml.org/wiki/index.php?title=Main_Page. Consulté le 27 mai 2017.

Liste des figures

Figure 1-1: Les étapes d'un système de recommandation	14
Figure 1-2: Filtrage basé sur le contenu	15
Figure 1-3: Filtrage collaboratif	17
Figure 1-4: Processus du filtrage collaboratif	17
Figure 1-5: Les méthodes de filtrage collaboratif	18
Figure 1-6: Matrice Utilisateur x Item	19
Figure 1-7: Tableau Utilisateur x Item	19
Figure 1-8: Filtrage hybride	22
Figure 1-9: Factorisation matricielle	24
Figure 1-10: Décomposition de la valeur singulière SVD [14].....	25
Figure 2-1: Pré-filtrage contextuel [33]	36
Figure 2-2: Post-filtrage contextuel [33].....	37
Figure 2-3: Modélisation contextuelle [33].....	38
Figure 3-1: Les approches utilisées pour chaque approche.....	40
Figure 3-2: Les approches du pré-filtrage	41
Figure 3-3: Les approches du post-filtrage	42
Figure 3-4: Les approches de la modélisation contextuelle	43
Figure 4-1: Architecture du système MovieRec_FM.....	47
Figure 4-2: Diagramme de cas d'utilisation.....	48
Figure 4-3: Diagramme de séquence.....	49
Figure 4-4: Exemple de la factorisation matricielle	51
Figure 4-5: L'interface de l'application MovieRec_FM	53
Figure 4-6: Profil User de l'utilisateur 20.....	54
Figure 4-7: Recommandation jour/printemps	55
Figure 4-8: Recommandation nuit/printemps	55
Figure 4-9: Recommandation jour/hiver	56
Figure 4-10: Recommandation nuit/hiver	56
Figure 4-11: Recommandation sans contexte	57

Liste des tableaux

Table 4-1: Tableau de description des champs du fichier movies	50
Table 4-2: Tableau de description des champs du fichier ratings	50

