Ministère de l'Enseignement supérieur et de la Recherche scientifique

Université de la Manouba
École Nationale Des Sciences De L'Informatique

Rapport de Projet de Conception et de Développement

Sujet

Modélisation des utilisateurs d'un système de recommandation personnalisée en utilisant l'algorithme Expectation-Maximization

---

Réalisé par

Asma GUEMRI

Wafa OUNIS

Encadré par

Madame Sonia BEN TICHA AZZOUZ

Année universitaire : 2016-2017

Appréciations de l’encadrant

Remerciements

Nous tenons, avant de présenter notre travail, à témoigner nos profonds remerciements envers toutes les personnes ayant contribuées de près ou de loin à l'élaboration de ce travail, qui présente une phase essentielle dans notre cursus scolaire.

C'est parce que nous avons beaucoup estimé leurs conseils, critiques et soutiens, que nous tenons à leurs faire part de toute notre gratitude.

Nous avons l'honneur d'exprimer à travers ces courtes lignes, notre sincère reconnaissance à notre encadrante Madame Sonia Ben Ticha Azzouz pour ses bonnes directives, sa disponibilité et ses conseils précieux qui nous ont permis de mener à bon terme notre projet.

Nous tenons à exprimer notre respect et notre haute considération aux membres du jury pour l'honneur qu'ils nous ont fait d'avoir accepté d'examiner et évaluer cette modeste contribution, et à l'administration et tout le corps enseignant de l'ENSI pour la formation qu'ils nous ont assuré durant ces deux années.

Enfin, nous tenons à dédier cet humble travail à nos familles avec tous nos sentiments d'amour, de gratitude et de reconnaissance pour tous leurs sacrifices déployés pour nous.

Glossaire

SRP	Système de Recommandation Personnalisée
EM	Expectation-Maximization
MSU	Modèle sémantique des utilisateurs

Table des matières

Table des figures

Introduction Générale

Avec la très grande masse d'informations devenue disponible sur l'internet , il est devenu primordial aujourd'hui d'assister l'utilisateur en lui faisant parvenir continuellement l'information qui l'intéresse au lieu de le laisser dépenser son temps à chercher l'information dont il a besoin . La tendance actuelle est de concevoir et mettre en application des systèmes capables de traiter un flux d'informations au fur et à mesure de leur arrivée pour en extraire et diffuser seulement les informations pertinentes .

Parmi ces systèmes on trouve le système de recommandation personnalisée (SRP) qui est défini comme étant un ensemble d'outils et de techniques intégrés dans un e-service (un terme référant la prestation de services via Internet , exemples : site web , commerce électronique,. . .) , permettant de filtrer les ressources disponibles en ne présentant à l'utilisateur courant que celles susceptibles de l'intéresser . Toutes les ressources d'un système de recommandation sont de même type, on peut citer par exemple un système de recommandation de films , ou un système de recommandation de musique , . . .

La personnalisation de la recommandation réside dans l'apprentissage d'un profil pour chaque utilisateur (appelé aussi modèle utilisateur) sur lequel se base le système de recommandation pour proposer à chacun des usagers des ressources en relation avec ses préférences. Le profil d'un utilisateur est défini généralement à partir de l'analyse de l'historique de ses interactions avec le service. Cette technique est appelée l'analyse des usages .

Le but de notre projet est de proposer un nouveau profil utilisateur en tenant compte des informations issues de l'analyse des usages ainsi que des informations sémantiques sur les ressources à recommander en utilisant l'algorithme Expectation-Maximization . Ce projet se présente dans le cadre d'une extension d'un travail de recherche sur l'apprentissage des profils sémantiques des utilisateurs dans un système de recommandation personnalisée .

Dans notre travail nous allons utiliser le jeu de données MovieLens 100K[3] issu du système de recommandation de films MovieLens (movielens.umn.edu)[3] . Ce jeu de données contient

100000 évaluations de 943 utilisateurs sur 1682 films triés par ordre chronologique ainsi que la description des genres de chaque film .

Notre rapport est organisé en 4 grands chapitres.Le premier , qui est L'état de l'art , est consacré pour une présentation générale du système de recommandation personnalisée en se focalisant sur la technique du filtrage collaboratif et également une explication du principe de l'algorithme Expectation-Maximization . L'analyse et la spécification de notre travail sont l'objet du second chapitre . Nous détaillons ensuite la conception du projet dans le troisième chapitre et nous présentons dans le dernier la partie réalisation .

Chapitre 1

État de l'art

Introduction

Dans ce chapitre nous présentons le principe général d'un système de recommandation personnalisée en s'intéressant à la technique de recommandation basée sur le filtrage collaboratif, ainsi que le principe de l'algorithme Expectation-Maximization.

1.1 Système de recommandation personnalisée

1.1.1 Principe

Un système de recommandation personnalisée a pour objectif d'apprendre les goûts et les préférences d'un utilisateur courant à fin de lui recommander par la suite les items susceptibles de l'intéresser.
Item est le terme utilisé pour désigner une ressource du système de recommandation. Par exemple, dans un système de recommandation de musique, les items sont les chansons ou la musique à recommander. Dans un système de recommandation des films, tel est le cas de MovieLens, les items sont les films.

Trois techniques de recommandation personnalisée sont principalement utilisées :

- Le filtrage basé sur le contenu : les items recommandés sont ceux similaires aux items que l'utilisateur courant a déjà appréciés dans le passé.
- Le filtrage collaboratif : les items recommandés sont ceux appréciés par un ensemble d'utilisateurs partageant les mêmes goûts que l'utilisateur courant (ses plus proches voisins).

- Le filtrage hybride : combine les deux techniques précédentes de différentes manières.

1.1.2 Filtrage collaboratif

Le filtrage collaboratif est une technique parmi les techniques les plus utilisées dans les systèmes de recommandation personnalisée. Cette technique se base sur le principe que si deux utilisateurs notent n items de la même façon alors ils noteront d'autres items de façon similaire. La figure ci-dessous présente l'architecture générale d'un SRP basé sur le filtrage collaboratif. Une première étape de l'algorithme de filtrage collaboratif consiste à définir les plus proches voisins de l'utilisateur courant. Une deuxième étape consiste à prédire les valeurs manquantes dans la matrice des votes. Enfin une liste des items pertinents sera recommandée pour l'utilisateur courant .

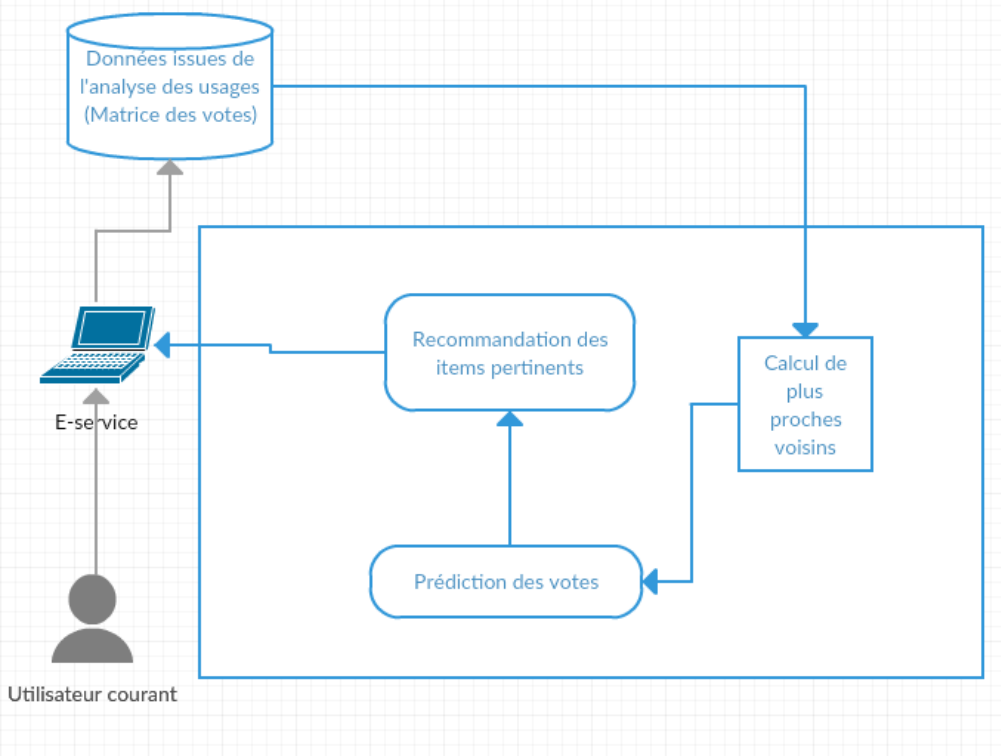

FIGURE 1.1: Architecture générale d'un système de recommandation personnalisée basé sur le filtrage collaboratif

1.1.2.1 Personnalisation dans le filtrage collaboratif

Dans la technique de filtrage collaboratif, la personnalisation est définie à travers les votes attribués par l'utilisateur aux différents items qu'il a consultés.

Un vote est une valeur numérique définie sur une échelle de valeurs spécifique et qui représente l'évaluation de l'item par l'utilisateur. Pour MoviLens par exemple, les votes sont dans l'intervalle [1..5], la valeur 1 signifie que l'utilisateur n'a pas aimé l'item et la valeur 5 montre qu'il est totalement satisfait.

Matrice des votes

Les votes attribués par les différents utilisateurs aux différents items sont représentés sous forme d'une matrice appelée matrice des votes qui résulte de l'analyse des usages.
Ci-dessous un exemple de matrice des votes de n utilisateurs sur N items avec v_{ij} représente la valeur du vote attribué par l'utilisateur U_i à l'item I_j s'il l'a consulté et ? désigne une donnée manquante (c'est à dire que l'item est inconnu pour l'utilisateur).

$$\begin{array}{c} \\ U_1 \\ \\ U_i \\ \\ \\ U_n \end{array} \begin{array}{c} \begin{array}{ccccc} I_1 & ..... & I_j & ..... & I_N \end{array} \\ \begin{pmatrix} v_{11} & .....?.....??... & ? & ....???...? & v_{1N} \\ ? & .?...??..... & ? & .....?..??....??? & ? \\ ? & ..???....?? & v_{ij} & ..??.... & v_{iN} \\ .... & ....???..?? & ? & ...??...? & ? \\ ? & ??..... & ? & .....?..??.. & .. \\ v_{n1} & ...?? & v_{nj} & ...?.....??.. & v_{nN} \end{pmatrix} \end{array}$$

La matrice des votes est une matrice creuse avec un taux très élevé de données manquantes pouvant atteindre le 98%. Movilens par exemple a un taux avoisinant les 95%.

Profil utilisateur

Chaque utilisateur, dans la technique de filtrage collaboratif, est représenté par un profil appelé profil usage. Le profil usage de l'utilisateur U_i est défini par le vecteur ligne d'indice i de la matrice des votes. Il est défini donc dans la dimension des items, représentant les votes attribués par cet utilisateur aux différents items consultés.

1.1.2.2 Recommandation

Un algorithme de filtrage collaboratif basé sur les utilisateurs consiste à recommander à l'utilisateur courant des items appréciés par ses voisins. Pour ce faire, l'algorithme doit calculer

les plus proches voisins de l'utilisateur courant pour prédire après ses votes sur les items qu'il n'a pas consultés.

Calcul des plus proches voisins

La détermination des plus proches voisins de l'utilisateur courant se fait en utilisant le coefficient de corrélation de Pearson. C'est la meilleure mesure pour calculer la similarité entre les utilisateurs d'après Herlocker et al, 1999[5]. Le coefficient de corrélation de Pearson permet de calculer la similarité entre deux utilisateurs u et w selon la formule suivante :

$$sim(u,w) = Pearson(u,w) = \frac{\sum_{i \in I_{uw}} (v_{ui} - \overline{v_u})(v_{wi} - \overline{v_w})}{\sqrt{\sum_{i \in I_{uw}} (v_{ui} - \overline{v_u})^2} \sqrt{\sum_{i \in I_{uw}} (v_{wi} - \overline{v_w})^2}}$$

Avec $\overline{v_u}$ et $\overline{v_w}$ sont les moyennes des votes attribués par l'utilisateur u, respectivement w, aux différents items de I_{uw}, I_{uw} étant l'ensemble des items notés à la fois par l'utilisateur u et w.
Le coefficient de corrélation de Pearson varie entre -1 et 1. Une valeur égale à 1 indique que les utilisateurs partagent exactement les mêmes goûts, une valeur de -1 indique qu'ils ont des goûts totalement opposés.

Prédiction des votes

Le calcul de la prédiction de la valeur du vote d'un item i non observé par l'utilisateur courant u_a se fait en appliquant la formule suivante :

$$pred(u_a, i) = \overline{v_{u_a}} + \frac{\sum_{u \in voisins(u_a) \cap U_i} sim(u_a, u)(v_{ui} - \overline{v_u})}{\sum_{u \in voisins(u_a) \cap U_i} |sim(u_a, u)|}$$

Avec U_i représente l'ensemble des utilisateurs ayant noté l'item i.
Le calcul des prédictions est exécuté en temps réel lors de la connexion de l'utilisateur courant au e-service.

1.2 Algorithme Expectation-Maximization

1.2.1 Principe général

L'algorithme EM — pour Expectation-Maximisation — est un algorithme itératif du à Dempster, Laird et Rubin (1977). Il s'agit d'une méthode d'estimation paramétrique s'inscrivant dans le cadre général du maximum de vraisemblance. Lorsque les seules données dont on dispose ne permettent pas l'estimation des paramètres, et/ou que l'expression de la vraisemblance est

analytiquement impossible à maximiser, l'algorithme EM peut être une solution. Il vise à fournir un estimateur lorsque cette impossibilité provient de la présence de données cachées ou manquantes, ou plutôt, lorsque la connaissance de ces données rendrait possible l'estimation des paramètres.
L'algorithme EM tire son nom du fait qu'à chaque itération il opère deux étapes distinctes :
— la phase « Expectation », souvent désignée comme « l'étape E », procède comme son nom le laisse supposer à l'estimation des données inconnues, sachant les données observées et la valeur des paramètres déterminée à l'itération précédente.
— la phase « Maximisation » , ou « étape M », procède donc à la maximisation de la vraisemblance, rendue désormais possible en utilisant l'estimation des données inconnues effectuée à l'étape précédente, et met à jour la valeur du ou des paramètre(s) pour la prochaine itération.
L'algorithme garantit que la vraisemblance augmente à chaque itération, ce qui conduit donc à des estimateurs de plus en plus correctes.

1.2.2 L'algorithme EM pour le clustering

Plusieurs variantes de l'algorithme EM ont été proposées. Parmi celles les plus pertinentes et les plus utilisées, on trouve sa version classifiante CEM (C pour Classification). C'est une version qui a été proposé par Celeux et Govaert, 1992, en introduisant une étape classification entre les deux étapes de l'algorithme EM.
En gardant les deux étapes E et M telle qu'elles sont, à une itération c, chaque item i est affecté à une classe dont la probabilité d'appartenance est maximale.

Conclusion

Nous avons présenté dans ce chapitre le principe du système de recommandation personnalisée et la technique de filtrage collaboratif. Par la suite nous avons présenter le principe de l'algorithme EM que l'on détaillera dans le chapitre [3] .
Dans le chapitre suivant nous allons faire l'analyse et la spécification de notre système.

Chapitre 2

Analyse et Spécification

Introduction

Dans ce chapitre nous allons présenter la spécification de notre travail.

2.1 Architecture générale de notre système de recommandation

Un système de recommandation personnalisée est un système de traitement des informations qui permet d'établir des recommandations à chaque utilisateur à partir de son profil personnel. Ce profil est généralement défini seulement par les données issues de l'analyse des usages (matrice des votes). Chaque système de recommandation personnalisée est constitué donc de deux composants. Le premier est chargé de la personnalisation et le deuxième est chargé de la recommandation.
Dans ce projet nous nous limitons à la phase de personnalisation. Notre travail consiste à faire l'apprentissage d'un nouveau profil utilisateurs à partir des données issues de l'analyse des usages ainsi que des données sémantiques sur les items à recommander en utilisant l'algorithme Expectation-Maximization. Ce profil est appelé modèle sémantique des utilisateurs (MSU).
La figure 2.1 suivante présente l'architecture générale de notre nouveau système.

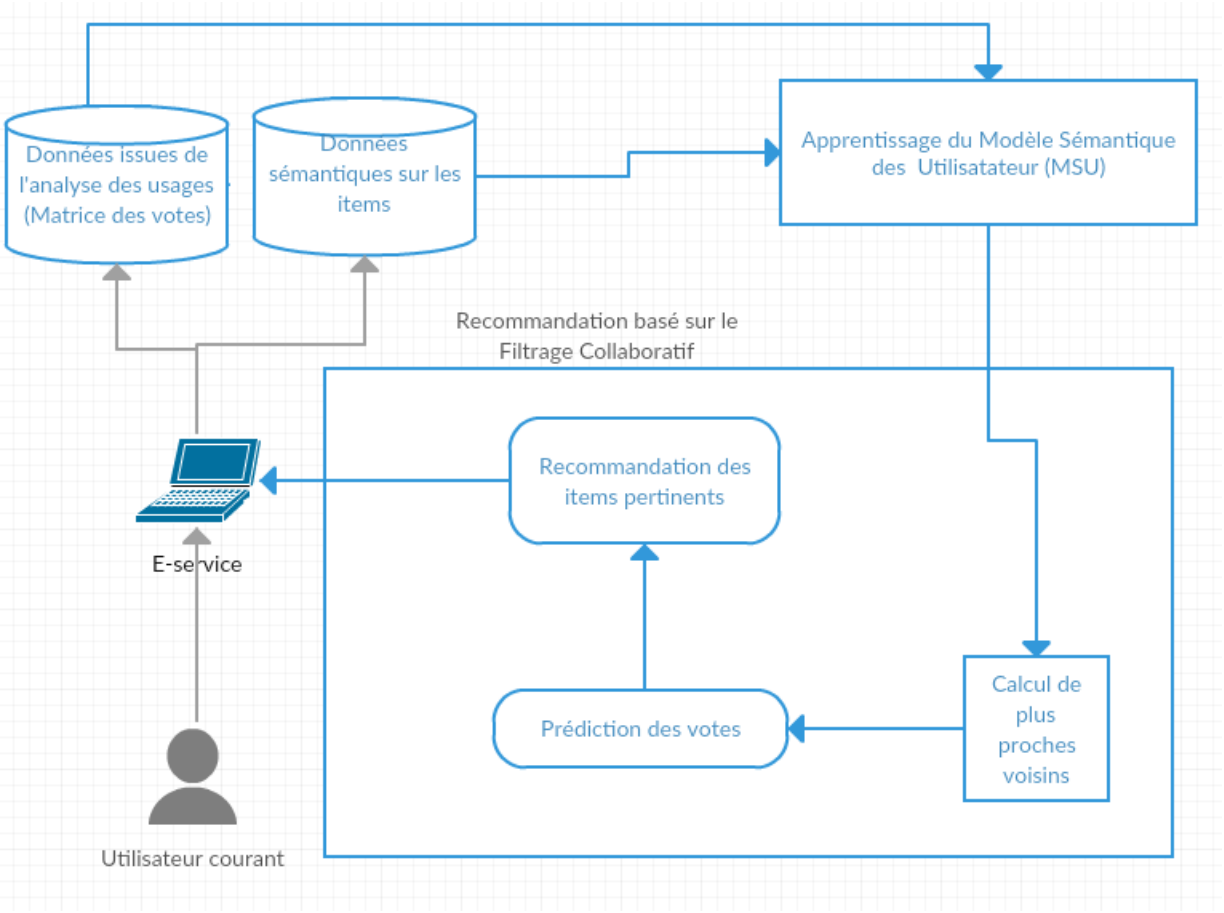

FIGURE 2.1: Architecture de notre système de recommandation personnalisée[1]

2.2 Apprentissage du modèle sémantique des utilisateurs (MSU)

Le modèle sémantique des utilisateurs est construit à partir des données sémantique sur les items et des données issues de l'analyse des usages.

Les données sémantiques sur les items

Les items dans notre cas, dans le jeu de données MovieLens, sont décrits uniquement par l'attribut genre suivant une représentation structurée. L'attribut genre est un attribut indépendant multivalué. C'est à dire que nous disposons d'un nombre limité de genres(dans le jeu de données MovieLens 100K nous considérons 18 genres) et qu'un film peut avoir plusieurs genres.

Ci-aprés un exemple d'une représentation structurés de l'attribut genre pour deux items.

Titre	Genre
Titanic	Romantique, Drame
les visiteurs	Comédie

TABLE 2.1: Représentation structurée des items

Données issues de l'analyse des usages

L'analyse des usages a pour objectif d'extraire les préférences des utilisateurs à partir des différentes interactions qu'ils ont avec le e-service. Dans notre cas, les préférences des utilisateurs sont représentées par des votes dont la formalisation est détaillée dans la section 1.1.2.1.

2.3 Filtrage collaboratif

Comme montré dans la figure 2.1, le modèle sémantique des utilisateurs est intégrer par la suite dans un système de recommandation basé sur le filtrage collaboratif pour évaluer sa performance et exactitude de prédiction en comparant les résultats obtenus à d'autres SRP utilisant différentes approches.
Le fonctionnement de l'algorithme de filtrage collaboratif est expliqué dans la section 1.1.2.

Conclusion

Nous avons fourni dans ce chapitre une analyse de notre travail. Nous essayerons dans le chapitre qui suit de concevoir clairement l'architecture de notre système.

Chapitre 3

Conception

Introduction

Après l'achèvement de l'analyse des besoins, nous entamons maintenant la conception du module d'apprentissage du nouveau profil utilisateur.

3.1 Phase d'apprentissage

3.1.1 Modélisation des items

Chaque item i , dans notre travail, est modélisé par deux profils : un profil usage qui est le vecteur colonne d'indice i de la matrice des votes représentant les votes attribués à cet item par les différents utilisateurs, et un profil sémantique(PSI_i) qui est un vecteur binaire défini dans la dimension des genres avec :

$$PSI_i(j) = \begin{cases} 1 & \text{si } i \text{ est de genre } j \\ 0 & \text{sinon} \end{cases}$$

3.1.2 L'EM pour l'apprentissage du profil sémantique des utilisateurs

comme est mentionné dans le chapitre spécification, notre travail consiste à l'apprentissage d'un nouveau profil sémantique des utilisateurs en utilisant l'algorithme EM notamment sa variante classifiante (CEM) (voir la section 1.2.2).
Nous disposons en entrée de deux matrices : d'une matrice qui représente le profil sémantique des items défini dans la base de l'attribut genre, et de la matrice des votes.

L'application de cet algorithme se fait en deux principales phases : l'initialisation, et la répétition des étapes E et M jusqu'à la convergence comme le montre le diagramme d'activité suivant.

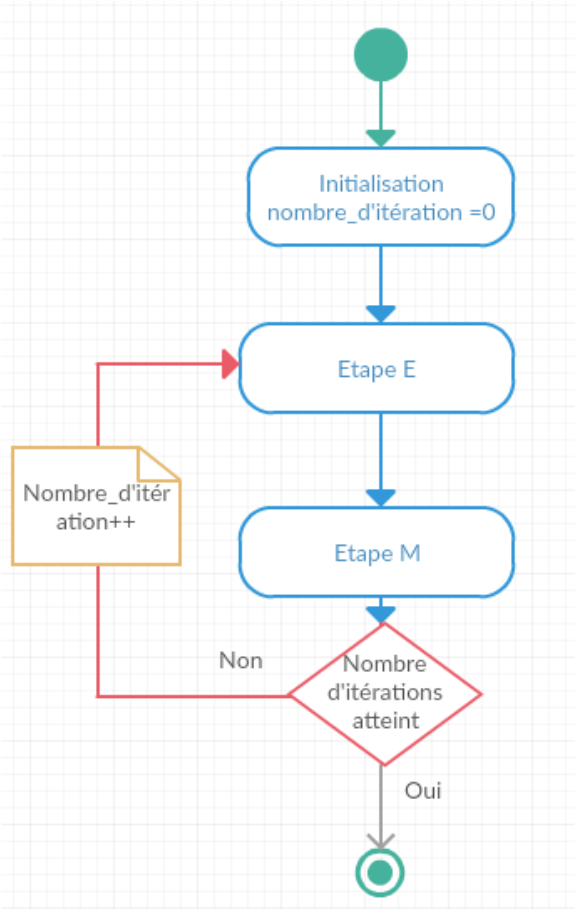

FIGURE 3.1: Diagramme d'activité de l'algorithme EM

Initialisation :

Nous considérons que les observations, qui sont les vecteurs modélisant le profil usage des items, sont issues d'une loi normale, leurs densités de probabilités sont donc des gaussiennes. L'étape d'initialisation consiste à initialiser les paramètres μ_j (la moyenne), Σ_j (la matrice variance-covariance) etπ_j (la probabilité à priori) pour chaque cluster C_j ;j=1..c ;c étant le nombre de clusters(genres).

Le problème dans l'initialisation vient du fait que nous ne connaissons pas a priori le cluster auquel appartient le vecteur X_i pour pouvoir calculer ces paramètres. Dans l'habitude, le choix de cette appartenance à priori se fait aléatoirement ce qui affecte le résultat obtenu pour la clas-

sification. Dans notre cas, l'algorithme est orienté en se basant sur le modèle sémantique des items. Ainsi, l'appartenance d'un item a chaque cluster est équiprobable entre tout les clusters représentant les genres dont il fait partie et elle sera nulle pour tout les autres.
exemple :pour un item i ayant le profil sémantique suivant :
Le tableau des probabilités à posteriori $\gamma_i^{(j)}$ d'appartenance à chaque cluster C_j, j=1..18, corres-

g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8	g_9	g_{10}	g_{11}	g_{12}	g_{13}	g_{14}	g_{15}	g_{16}	g_{17}	g_{18}
0	0	0	1	0	0	0	0	1	1	0	0	0	0	1	0	0	0

TABLE 3.1: Modèle sémantique de l'item i

pondant, est le suivant :

g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8	g_9	g_{10}	g_{11}	g_{12}	g_{13}	g_{14}	g_{15}	g_{16}	g_{17}	g_{18}
0	0	0	0.25	0	0	0	0	0.25	0.25	0	0	0	0	0.25	0	0	0

TABLE 3.2: Le tableau des probabilité à posteriori $\gamma_i(j)$ de l'item i

A partir de ce calcul des probabilités à posteriori, les autres paramètres$\mu, \Sigma et \pi$ sont initialisée selon les formules présentées ci-dessous :

$$\mu_j = \frac{\sum_{i=1}^{N} \gamma_i^{(j)} X_i}{\sum_{i=1}^{N} \gamma_i(j)}; \Sigma_j = \frac{\sum i{=}1^N \gamma_i^{(j)} (X_i - \mu_j)(X_i - \mu_j)^T}{\sum_{i=1}^{N} \gamma i^{(j)}}; \pi_j = \frac{\sum_{i=1}^{N} \gamma_i^{(j)}}{N}; \text{j=1..c}$$

Étape E (Expectation) :

consiste à calculer les probabilités à posteriori d'appartenance de chaque item I_i à chaque cluster C_j $\gamma_i^{(j)}, i = 1..N, j = 1..c$selon la formule suivante :

$$\gamma_i^{(j)} = \frac{(\pi_j N(X_i; \mu_j; \Sigma_j))}{\sum_{j=0}^{c} (\pi_j N(X_i; \mu_j; \Sigma_j))}$$

avec :

$$N(x_i; \mu_j; \Sigma_j) = \frac{1}{\sqrt{(2\pi)^d |det(\Sigma_j)|}} exp(-1/2(X_i - \mu_j)^T \Sigma_j^{-1} (X_i - \mu_j))$$

Remarque : $\sum_{j=0}^{c} \gamma_i^{(j)} = 1$; i=1..N.
Et comme nous travaillons dans un espace avec beaucoup de données manquantes, le vecteur usage de l'item contient beaucoup de valeurs nulles. Par ailleurs, on suppose que le résultat de toute opération ayant pour opérande nulle est égal à nulle.

Étape M (Maximisation) :

consiste à recalculer les paramètres μ_j,Σ_j et π_j en utilisant les nouvelles valeurs des probabilité à posteriori $\gamma_i^{(j)}$ trouvées dans l'étape E. Ce calcul se fait selon les mêmes formules présentées dans l'étape d'initialisation.

Convergence :

La convergence dans notre cas est atteinte après un certain nombre d'itérations bien déterminé. Nous disposons des paramètres et chaque item I_i sera affecté au cluster C_j tel que $\gamma_i(j)$ soit maximale.
Le résultat de l'application de l'algorithme EM est donc le modèle sémantique des utilisateurs présenté sous forme d'une matrice définie dans la dimension des utilisateurs/genres où chaque colonne j est le vecteur moyenne μ_j du cluster C_j représentant le genre j.

$$\begin{array}{c} \\ U_1 \\ U_2 \\ \\ U_i \end{array} \begin{array}{c} \begin{array}{ccccc} C_1 & .... & C_j & .... & C_{18} \end{array} \\ \begin{pmatrix} \mu_{11} & ..... & \mu_{1j} & ..?.. & \mu_{1,18} \\ \mu_{21} & ..... & \mu_{2j} & ..... & \mu_{2,18} \\ ...? & .. & .... & .. & ... \\ \mu_{i1} & ..... & \mu_{ij} & ..... & \mu_{i,18} \end{pmatrix} \end{array}$$

3.2 Phase de recommandation

Dans la partie recommandation de notre système, le résultat de l'apprentissage du modèle sémantique des utilisateurs, que nous développons, est donné en entrée pour un algorithme de filtrage collaboratif basé sur les utilisateurs.
l'algorithme de filtrage collaboratif se charge de calculer les plus proches voisins puis de prédire les votes manquantes pour enfin recommander les items pertinents à chaque utilisateur.
Le développement de l'algorithme de filtrage collaboratif ne fait pas partie de notre travail et est fourni par notre encadrante pour l'utiliser directement.

Conclusion

Tout au long de ce chapitre, nous avons décortiqué le module à réaliser progressivement. Nous nous sommes intéressés à la conception détaillée de la phase d'apprentissage du modèle

sémantique des utilisateurs. Dans le chapitre suivant, nous exposons la réalisation de notre travail ainsi que les résultats obtenus.

Chapitre 4

Réalisation

Introduction

Tenant compte des besoins fixés et des choix conceptuels effectués, nous consacrons ce chapitre à la description de l'état actuel du travail réalisé . Nous commencerons par décrire l'environnement matériel et logiciel sur lequel nous avons développé notre projet . Ensuite , nous signalerons l'état d'avancement du projet et nous mettrons en évidence le travail réalisé par la présentation de quelques captures d'écran traduisant le déroulement du projet . Enfin , nous finirons par un chronogramme qui décrit toutes les étapes de mise en œuvre du travail .

4.1 Environnement du travail

Nous décrivons dans cette section l'environnement matériel et logiciel adoptés pour l'implémentation de l'application demandée .

4.1.1 Environnement matériel

Pendant les différentes phases du travail , à savoir la documentation, la spécification des besoins , la conception et le développement , nous avons élaboré notre projet sur trois ordinateurs :

- Ordinateur1 :
  - Processeur : Intel® Core™ i7-4200M
  - Mémoire : 8 Go de RAM

— Système d'exploitation : Windows 10

- Ordinateur2 :
  — Processeur : Intel® Core™ i5-4200M
  — Mémoire : 6 Go de RAM
  — Système d'exploitation : Windows 7

- Ordinateur3 :
  — Processeur : Intel® Core™ i5-4200M
  — Mémoire : 4 Go de RAM
  — Système d'exploitation : Ubuntu 16.04

4.1.2 Environnement logiciel

Les choix techniques que nous avons adoptés sont présentés dans ce qui suit :

- Un Système d'exploitation Windows 10 intégral 64bits .
- Eclipse comme environnement de développement JAVA(IDE) .
- Creately pour le traçage des diagrammes .
- Overleaf comme éditeur Latex .

4.2 Phase d'implémentation

Pour l'implémentation de notre module d'apprentissage, nous avons cherché le code open source de l'algorithme EM[N6] que nous avons adapté aux données dont nous disposons pour remédier au problème de données manquantes.

4.2.1 Expérimentation

Nous utilisons dans notre travail le jeu de données MovieLens 100K issu du système de recommandation de films MovieLens . Ce jeu de données se compose de :
-100000 évaluations (de 1 à 5) de 943 utilisateurs sur 1682 films triés par ordre chronologique .
- Chaque utilisateur a évalué au moins 20 films .
- l'utilisateur est identifié uniquement par un id (de 1 à 943) .
-les films sont de 18 différents genres .
L'expérimentation consiste à partager le jeu de données initial en deux jeux selon la proportion 80/20 en considérant l'ordre chronologique . Le premier jeu ($jeu_apprentissage$), représentant 80% du jeu initial est utilisé dans l'apprentissage du modèle sémantique des utilisateurs (MSU). Le deuxième (jeu_test), le 20% restant du jeu initial, est utilisé comme jeu de test pour évaluer la prédiction des votes.

4.2.2 Évaluation

Dans cette sous-section nous allons évaluer notre système de recommandation en évaluant la précision des prédictions des votes.
Pour évaluer notre système de recommandation, le $jeu_{apprentissage}$ est utilisé pour l'apprentissage de la fonction de prédiction pred(u,i) et le jeu_{test} pour évaluer la précision de la prédiction. Notons par T l'ensemble des couples (u ;i) de jeu_{test} pour lesquels le système de recommandation a prédit la valeur du vote .
La mesure de précision des prédiction des votes se fait selon la formule de la Racine de l'Erreur Quadratique Moyenne , "Root Mean Squared Error" (RMSE) , suivante :

$$RMSE = \sqrt{\frac{\sum_{(u,i)\in T}(pred(u,i)-v_{ui})^2}{|T|}}$$

4.2.3 Chronogramme du travail

Ce travail a été réalisé durant une période de 12 semaines. Il a été structuré comme le décrit la figure suivante :

	Février				Mars				Avril			
	1	2	3	4	1	2	3	4	1	2	3	4
Documentation												
Conception												
Implémentation												
Rédaction du rapport												

FIGURE 4.1: Le schéma de chronogramme du travail

Conclusion

Malheureusement , durant l'implémentation de notre module d'apprentissage , nous avons eu un problème d'incapacité de la mémoire que , faute de temps , nous n'avons pas pu le résoudre.

Conclusion générale

Notre objectif était l'apprentissage d'un nouveau profil des utilisateurs d'un système de recommandation personnalisée en tenant compte des données issues de l'analyse des usages et des données sémantiques sur les items à recommander .

Afin de réaliser ce projet , nous avons suivi une démarche précise en commençant par l'état de l'art . Nous avons fait une recherche bibliographique sur le principe d'un système de recommandation personnalisée et sur l'algorithme EM . Ensuite nous avons spécifié nos besoins et nous avons expliqué comment adapter l'algorithme EM pour l'apprentissage du MSU.

Pour l'implémentation de notre module d'apprentissage, nous avons cherché une implémentation open source de l'algorithme CEM . Mais l'implémentation en java trouvée [N6] n'est pas adaptée aux données manquantes donc nous l'avons modifiée pour tenir compte de ce problème .
Nous avons utilisé pour l'apprentissage le jeu de données MovieLens 100k issu de système de recommandation des films MovieLens[3] qui contient 100000 évaluations de 943 utilisateurs sur 1682 films et fournit une description des genres de chaque film comme données sémantiques sur les items .

Mais, lors du développement, nous avons rencontré un problème de passage à l'échelle . Les données en entrée étant très volumineuses et l'algorithme est basé sur des calculs matriciels complexes ce qui entraîne des problèmes d'incapacité de mémoire .
Nous sommes en train d'étudier les solutions possibles afin de résoudre ces problèmes mais faute de temps nous ne sommes pas parvenus à les remédier .

Bibliographie

[1] Sonia Ben Ticha. Recommandation Personnalisée Hybride. 2015

[2] Gilles Gasso - Philippe LERAY , Clustering,INSA Rouen-Département ASI. Laboratoire LITIS

[3] JeuMovieLens , http ://grouplens.org/datasets/movielens/. 2014

[4] Frédéric Santos , L'algorithme EM : une courte présentation. 2015

[5] Jonathan L. Herlocker, Joseph A. Konstan, Al Borchers, and John Riedl. An algorithmic framework for performing collaborative filtering

[6] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. GroupLens : an open architecture for collaborative filtering of netnews. In The 1994 ACM conference on Computer supported cooperative work, page 175–186, 1994.

Netographie

[1] https ://www.eclipse.org/forums/. dernière consultation le 30/04/2017

[2] http ://www.pacea.u-bordeaux1.fr/IMG/pdf/algo-em.pdf. dernière consultation le 25/04/2017

[3]https ://www.overleaf.com/. dernière consultation le 2/05/2017

[4] https ://creately.com/app/. dernière consultation le 29/04/2017

[5] https ://www.developpez.com/. dernière consultation le 20/04/2017

[6] https ://github.com/nash-pwnage/Expectation-Maximization-Algorithm/blob/master/src/EM.java. consultation le 15/04/2017.

[7] http ://weka.sourceforge.net/doc.dev/weka/clusterers/EM.html dernière consultation le 30/04/2017.