Rapport de projet de C/SD

par

Nathan BARLOY, Valentin CRÔNE,
Johan TOMBRE

1 Avenue Paul Muller
54600 Villers-lès-Nancy
France

Projet de C/SD

Table des matières

Chapitre 1

Etat de l'art

1.1 Introduction

1.1.1 Qu'est ce qu'un système de recommandation ?

Un système de recommandation a pour but de founir des recommendation de résultats pertinentes à un utilisateur, en fonction de différents paramètres comme ses préférences, son historique, le temps passé sur le contenu, etc.., afin de lui proposer directement un contenu ciblé sans efforts de sa part.

Un algorithme de recommandation peut de ce fait être utilisé pour faciliter les recherches de contenus sur un thème donné, ce qui est bénéfique pour l'utilisateur car il aura plus d'informations pertinentes à regarder, et permettre aux gérants de la plateforme de fidéliser la clientèle.

Prenons un exemple, un site de e-commerce sur lequel on trouve facile des articles en rapport avec l'article que l'on est en train de consulter rendra plus facile la recherche, et donc aidera le client a trouver rapidement le produit adéquat, ce qui réduira le temps nécessaire à l'achat, et évitera la perte du client. Un autre exemple est une plateforme de visionnage de vidéos en ligne, qui génère des bénéfices grâce a la publicité, un tel système proposera du contenu ciblé interessant pour l'utilisateur, qui consultera un plus grand nombre de vidéos sur le site, ce qui permet à la fois de satisfaire l'utilisateur et améliorer la génération de revenus liés à la publicité. Ces effets sont liés au fait que l'utilisateur reçoit des suggestions pertinentes de la part du système de recommendation auxquelles il n'aurait pas forcément pu prêter attention autrement.

1.1.2 La place des systèmes de recommandation dans la recherche d'information

La recherche d'information sur internet est fondée sur un principe d'indexation des données. Afin de pouvoir répondre aux requêtes des utilisateurs dans un temps raisonnable, on indexe les données dans une base de données, et on les filtre en fonction de la requête des utilisateurs. Les requêtes sont entrées à partir de mots clés (requêtes ad hoc), et le moteur de recherche va alors retourner les résultats qui correspondent. Le nombre conséquent de résultats obligera le moteur de recherche a limiter la quantité d'informations retournée en limitant un certain nombre de résultats par page. Ce système fonctionne plutôt bien, mais nécessite que l'utilisateur traite lui même un grand nombre de résultats, ce qui peut se limiter a la ou les premières pages car il lui est impossible de tout traiter. On cherche donc à rendre les résultats de la première page plus pertinents. On fait alors immédiatement le lien avec les systèmes de recommandation, car au lieu de filtrer par mots clés, on va pouvoir rechercher des contenus à thème proche, date proche, et personnaliser les résultats en fonction des habitudes de l'utilisateur.

Ces exemples montrent un certain nombre de cas où l'utilisation d'un système de recommandation s'avère utile. Cependant il en existe de nombreux autres, car leur champ d'application est immense, c'est pourquoi il est interessant de s'interesser aux systèmes existants pour tenter de les reproduire et les améliorer.

1.1.3 Historique

A mettre ou pas ?

1.1.4 Classer les différents systèmes de recommandation

Il existe différents types de systèmes de recommandation, en fonction du type de données à classer, de la puissance disponible, de activités utilisateurs, des besoins de l'entreprise, et de nombreux autres facteurs. Les facteurs principaux à retenir sont :

- La connaissance des préférences utilisateur.
- La similarité entre les utilisateurs, les uns par rapport aux autres. (Métrique)
- La connaissance d'informations sur des données à recommander
- La connaissance d'informations de regroupement sur des données à recommander

A partir de ces facteurs on produit différents types de recommandations. Les systèmes les plus utilisés sont le filtrage basé sur le contenu, et le filtrage collaboratif.

Le filtrage basé sur le contenu

Le filtrage basé sur le contenu va essayer d'associer un utilisateur à un ensemble de données proche de ses préférences. On va alors rechercher des données qui pourraient l'interesser en tenant compte uniquement de lui-même, et de ce qu'il a consulté. Pour cela, il faut dresser des liens entres les différents éléments de la base de donnée, pour pouvoir déterminé si un élément est proche ou non de ce qu'a consulté l'utilisateur. L'avantage de ce filtrage est qu'il ne nécéssite pas d'avoir d'autres utilisateurs pour fonctionner, puisqu'on ne compare pas les utilisateurs entre eux. Cela est donc très pratique quand la base de donnée des utilisateur est petite.

Le filtrage collaboratif

Le filtrage collaboratif va mesurer une certaine distance entre un utilisateur donné, et tous les autres utilisateurs pour selectionner les utilisateurs les plus proches uniquement. On utilisera alors le profil de ces utilisateurs proches pour regarder les données qui les ont interessés pour les recommander à notre utilisateur initial. L'avantage de ce filtrage est qu'il ne nécéssite pas de créer un algorithme qui détermine si deux données sont proches ou non. Cependant, si la base de donnée des utilisateurs n'est pas assez fournie, ce filtrage ne sera que très peu efficace.

Evidemment, les filtrages les plus efficaces sont ceux qui mélangent les méthodes décrites précédement, de manière à en tirer les avantages.

1.2 Les données

Pour pouvoir lier des produits avec les utilisateurs, il est indispensable de mettre au point un système de notation afin de déterminer si un utilisateur apprécie ou non un produit. On distingue deux approches différentes :

— l'approche "réactive", où le système de recommandation deamande à l'utilisateur des informations / avis sur certains produits afin d'affiner ces recommandations. Cette approche présente cependant l'inconvénient de solliciter l'utilisateur.
— l'approche "proactive", qui anticipe plus les goûts de l'utilisateur.

1.2.1 L'approche réactive

Cette approche consiste à déterminer les goûts de l'utilisateur à travers un processus conversationnel. Le système demande régulièrement des informations à l'utilisateur sur ses préférences sur ses goûts... puis va proposer des recommandations que l'utilisateur va accepter ou alors critiquer les résultats afin d'affiner le système. L'avantage de ce système de critique est qu'il est simple à appliquer et ne requiert pas de connaissance de l'utilisateur dans ce domaine. Toutefois, l'inconvénient majeur est qu'il demande directement à l'utilisateur de faire un effort pour exprimer son avis et donner un retour.

1.2.2 L'approche proactive

L'approche proactive ce base davantage sur la déduction des appréciations pour fournir ensuite une recommandation. Ainsi l'utilisateur n'a plus à guider le système avec des retours sur les recommandations proposées mais va observer les interactions de l'utilisateur pour déterminer les goûts de celui-ci. Ces observations peuvent être directes ou indirectes.

Observation directe

Ce sont toutes les informations que l'utilisateur donne de manière explicite en notant, postant un commentaire sur un produit par exemple ou encore les informations du profil de l'utilisateur comme son âge, son sexe, sa localité géographique... La notation des produits est un élément centrale dans un système de recommandations. Ces notes sont principalement numériques peuvent avoir différentes formes qui délivrent plus ou moins d'informations.

Il y tout d'abord la notation unaire : cette notation ne délivre qu'une seule information, si l'utilisateur a aimé un produit. On peut donner par l'exemple de réseau social Instagram qui donne la possibilité aux utilisateurs de "liker" une photo uniquement, il est impossible de mettre un avis négatif.

Il y a ensuite la notation binaire qui prend en compte l'avis négatif de l'utilsateur. L'information transmise est donc binaire : "J'aime" / "Je naime pas". C'est par exemple ce qu'utilise Facebook ou Youtube. L'avantage de cette notation est facilite le choix de l'utilisateur.

Enfin, la notation ordonnée est celle qu'on retrouve dans la majorité des site de e-commerce (Amazon, C Discount et autres). L'utilisateur donne une note à un produit suivant une échelle de notation comprise entre deux valeurs (entre 1 et 5 par exemple). Plus ce chiffre est élevé, plus l'utilisateur a aimé le

produit. Cette notation permet plus de nuances sur l'appréciation des produits et donc améliore les recommandations qui en découle. Cependant, le coût en calcul en est impact car l'information n'est plus binaire. De plus cette notation est sensible à la manière de noter de chaque utilisateur : une note de 3/5 sera considéré comme une mauvaise note par certains utilisateurs alors que pour d'autres, ce sera un note neutre / moyenne. Cela peut donc impacter la qualité des recommandations.

Partie sur l'analyse des commentaires pour en extraire des tags + utilisations des données profil utilisateur

Observation indirecte

Ce sont l'ensemble des informations immplicites données par l'utilisateur ou déduites par le système. Il peut s'agir par exemple d'actions réalisées sur une page web comme une recherche, un ajout dans un panier d'achat ou tout simplement de la fréquence de consultation d'une page. Il est ensuite possible à partir de ces informations d'estimer les préférences de l'utilisateur. Par exemple, si on utilisateur ajoute un produit dans son panier ou tout simplement le consulte fréquemment, on peut conclure que ce type de produit l'intéresse puis intégrer cette information à notre système.

1.2.3 Conclusion sur les données

Nous avons vu différentes manières de collecter des informations relative à l'utilisateur dans le but d'alimenter notre algorithme de recommandation. Que ce soit par une approche réactive ou proactive, ces techniques peuvent soulever un débat sur la collecte des données. Il faut donc mener une réflexion pour trouver un bon compromis entre efficacité du système et collecte de données "abusive".

1.3 La matrice d'usage

Egalement appelé le modèle utilisateur, la matrice d'usage contient des données collectées sur les utilisateurs associés aux produits. On le représente généralement sous la forme d'une matrice avec les utilisateurs sur les lignes et les produits en colonne. Les scores attribués peuvent être une note, un nombre de consultation, un durée, un like... Cette matrice comporte généralement de nombreuses valeurs manquantes qu'il va falloir chercher à déterminer grâce à notre système de recommandation. Il est également intéressant de remarquer que les goûts des utilisateurs évoluent au cours du temps. Il est donc important de réajuster les données de cette matrice régulièrement.

1.4 Réduction de matrice

Reprenons la matrice d'usage tout juste présentée. Elle est de dimension mxn avec m le nombre d'utilisateurs et n le nombre de produit. Dans la pratique, cela représente habituellement des milliers / millions d'utilisateurs pour autant voire plus de produits. On obtient donc une matrice possèdant des milliards de valeurs. Il serait donc très long d'effectuer des calculs dessus pour en déduie des recommandation. La solution est donc de réduire la dimension de la matrice. Plusieurs approches sont possibles. On peut par exemple ne pas prendre en compte les utilisateurs dont on ne possède que très peu d'informations ou de la même manière retirer les produits peu populaires et donc ayant moins de chance d'intéresser les utilisateurs. Cependant, cette méthode a comme gros inconvénient de réduire le champ des recommandations proposées. En effet, comment proposer à un utilisateur un produit qui lui conviendrait parfaitement s'il est retiré de la liste car peu connu ? Cela restreint plus ou moins les produits possiblement recommandés à ceux qui sont populaires et donc déjà connu par la plupart.

Une autre approche consiste à ne conserver qu'une gamme de produit répondant à un critère spécifié. Par exemple, il est possible de supprimer les films de genre romantique si l'utilisateur ne manifeste aucun intérêt pour ce genre. Enfin, une dernière méthode consiste à ne prendre en compte que les utilisateurs jugés comme étant proche de l'utilisateur cible, c'est ce qu'on appelle le clustering (détaillé dans la section suivante).

L'avantage de cette méthode est donc de réduire le temps de calcul pour déterminer une recommandation. Cependant, la qualité de la recommandation s'en retrouve impacté. En effet cela réduit le nombre de personnes analysés par lalgorithme dans le cas dune réduction du nombre dutilisateur, et une réduction du nombre de produits réduira le champs des recommandations à un certains type.

Clustering

Lidée est décarter les items ou les utilisateurs jugés non représentatif pour un utilisateur (item) donné. On réduit ainsi la dimension de la matrice tout en minimisant la perte de données avant deffectuer les calculs de recommandation. Pour ce faire, on utilise des méthodes de partitionnementă: cela consiste à regrouper les éléments similaires. La méthode la plus connue est k-means

1.5 Calcul de similarités

Un calcul de similarité est utile pour déterminer la distance entre un utlisateur et un produit mais également entre deux utlisateurs (ou produits). Plusieurs outils existent pour nous permettre d'effectuer ce calcul. Nous verrons dans cette partie quelques unes de ces méthodes et discuterons des avantages et inconvénients qu'ils présentent.

1.5.1 La distance Euclidienne

La distance euclidienne est obtenue par la formule suivante :

$$L(\vec{x}, \vec{y}) = \sqrt{\sum_{i=1}^{m} (x_i - y_i)^2} \tag{1.1}$$

Paragraphe à terminer...

1.5.2 Distance cosinus

On considère les vecteurs ligne de x et y. Le but est de calculer la distance, langle entre ces deux vecteurs. Dans notre cas, les vecteurs x et y seront des vecteurs associés aux profils de deux utilisateurs par exemple.

$$CosSim(\vec{x}, \vec{y}) = \frac{\sum_{i=1}^{m} x_i y_i}{\sqrt{\sum_{i=1}^{m} x_i^2} \sqrt{\sum_{i=1}^{m} y_i^2}} \tag{1.2}$$

L'inconvénient majeur de cette méthode est quelle considère tout film non noté comme un zéro, qui est une mauvaise note, ce qui va réduire la distance entre deux utilisateurs qui en réalité seraient jugés différents.

1.5.3 Corrélation de Pearson

Afin de pallier le problème citer précédemment, il est possible de normaliser au préalable les notes en leur soustrayant la moyenne des notes données par l'utilisateur. Cette méthode est également appelé la distance cosinus centrée. On a alors :

$$Corr(x, y) = \frac{\sum_{i=1}^{m} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{m} (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^{m} (y_i - \bar{y})^2}} \tag{1.3}$$

Chapitre 2

Rapport

2.1 Amélioration de la base de données

2.1.1 Liens des bandes annonces

Afin d'améliorer l'expérience de l'utilisateur, nous souhaitons lui proposer pour chaque film consulté, la bande annonce de ce dernier. Comme notre interface graphique sera gérée grâce à la librairie WebKitWebView, il sera possible d'intégrer à notre page un lecteur YouTube avec la bande annonce. Il suffit donc de récupérer les liens URL des vidéos pour que l'ensemble fonctionne.

Comme ces liens seront stockés dans la base de donnée JSON des films, il n'est nécessaire d'exécuter ce programme qu'une seule fois. Nous avons donc optés pour un script python qui est plus simple et rapide à mettre en oeuvre, grâce notamment à ses nombreuses librairies disponibles en ligne. Le résultat de cette exécution sera ensuite stocké dans un fichier texte avant d'inclure les liens au fichier JSON par l'intermédiaire d'un programme C++.

Ce script se décompose en trois étapes. Tout d'abord, il faut réussir à générer une requête à partir du nom du film. Pour cela, nous nous appuyons sur la simplicité de l'URL lors d'une recherche. Cette URL possède une partie constante qui est https ://www.youtube.com/results ?search_query= suivi du contenu de la recherche avec des "+" remplaçant les espaces. Notre requête est donc composé du nom du film, récupéré à partir du fichier JSON suivi des mots "bande" et "annonce". Il faut considérer certains cas où le titre du film contient un caractère interdit dans une URL (comme le &). Ce carctère doit dans ce cas être encodé à l'aide du signe % suivi d'un nombre (dans notre cas le 26) Une fois l'URL de la requête généré, elle est exécuté pour récupérer le code HTML de la page des résultats.

La seconde étape consiste à récupérer le lien de la première vidéo proposée (il est supposé que l'algorithme de YouTube effectuer bien sa mission). Il est intéressant de remarquer qu'un lien YouTube ne se distingue que par ses 11 derniers caractères. Nous utilisons donc une expression régulière qui recherche

tous les liens des vidéos proposées puis on stocke ces séries de 11 caractères dans une liste. Pour terminer, on sauvgarde recréer le lien de la vidéo en concaténant le début du lien https ://www.youtube.com/watch ?v= et la première série de caractère trouvé. Il ne reste plus qu'à stocker ce lien à côté de l'ID du film dans un fichier texte.