Extraction de l'intérêt implicite des utilisateurs dans les attributs des items pour améliorer les systèmes de recommandations

Manuel Pozo, Raja Chiky, Elisabeth Metais

LISITE-ISEP
28, rue Notre Dame Des Champs
75006 Paris
{manuel.pozo, raja.chiky}@isep.fr, elisabeth.metais@cnam.fr

Résumé. Les systèmes de recommandation ont pour objectif de sélectionner et présenter d'abord les informations susceptibles d'intéresser les utilisateurs. Ce travail expose un système de recommandation qui s'appuie sur deux concepts: des relations sémantiques sur les données et une technique de filtrage collaboratif distribué basée sur la factorisation des matrices (MF). D'une part, les techniques sémantiques peuvent extraire des relations entre les données, et par conséquent, améliorer la précision des recommandations. D'autre part, MF donne des prévisions très précises avec un algorithme facilement parralélisable. Notre proposition utilise cette technique en ajoutant des relations sémantiques au processus. En effet, nous analysons en profondeur les intérêts cachés des utilisateurs dans les attributs des items à recommander. Nous utilisons dans nos expérimentations le jeu de données MovieLens enrichi par la base de données IMDb. Nous comparons notre travail à une technique MF classique. Les résultats montrent une précision dans les recommandations, tout en préservant un niveau élevé d'abstraction du domaine. En outre, nous améliorons le passage à l'échelle du système en utilisant des techniques parallélisables.

1 Introduction

La quantité d'information dans le Web a augmenté ces dix dernières années. Ce phénomène a favorisé la progression de la recherche dans le domaine des systèmes de recommandation. Les systèmes de recommandation consistent en un filtrage de l'information dans le but de ne présenter aux utilisateurs que les éléments qui sont susceptibles de l'intéresser, quel que soit le domaine. Les éléments à recommander sont également appelés items et peuvent être de différents types : des produits, services, informations, etc. Les systèmes de recommandation se doivent de sélectionner les informations les plus intéressantes en fonction du but recherché, tout en conciliant nouveauté, surprise et pertinence. Un système de recommandation se base sur des caractéristiques de références acquises de manière automatisée selon plusieurs méthodes différentes. Les caractéristiques de références peuvent provenir de :

— L'item (l'objet à recommander) lui-même, on parle alors « d'approche basée sur le contenu » (ou content-based approach) Balabanović et Shoham (1997). Le filtrage basé sur le contenu calcule la similarité entre les objets afin de trouver l'objet le plus semblable aux goûts de l'utilisateur. Dans ce cas, l'utilisateur se voit recommander des items similaires à ceux qu'il a préférés dans le passé.
— L'utilisateur et l'environnement social, on parle alors « d'approche de filtrage collaboratif » (ou collaborative filtering). Le principe du filtrage collaboratif Breese et al. (1998) est d'implanter informatiquement le principe du « bouche-à-oreille ». Il utilise les comportements connus d'une population pour prévoir les futurs agissements d'un individu. La méthode collaborative présente des avantages par rapport au filtrage basé sur le contenu. En effet, elle est plus efficace dans la pratique et simple à mettre en oeuvre. Notamment, il a été prouvé que les techniques de factorisation de matrice fournissent des résultas précis et ont l'avantage d'être facilement parallélisable (pour la montée en charge) Koren et al. (2009).

Pour remplir leurs fonctions, les technologies de recommandation font aujourd'hui face à des défis scientifiques majeurs. Comment intégrer l'hétérogénéité des sources d'information pour modéliser les préférences, comment prendre en compte le contexte, comment traiter efficacement ces masses d'information, quels types d'interfaces faut-il considérer ? Par ailleurs, les deux approches citées précédemment présentent des inconvénients principalement liés au démarrage à froid et à la montée en charge du système d'où la nécessité de mettre en place des algorithmes performants et robustes. Ceci est l'objectif de cette étude en vue d'améliorer la qualité des systèmes de recommandation en introduisant de la sémantique aux données et en distribuant les traitements afin de minimiser les temps de calcul. La sémantique est ici représentée par une ontologie du domaine (domaines des films pour les expérimentations).

2 Architecture globale

Afin de fournir une généricité dans le domaine d'application, un passage à l'échelle et une recommandation précise, nous proposons un système à trois couches : une couche de pré-analyse, une couche sémantique et une couche de recommandation.

Le module de pré-analyse met en oeuvre un filtre de comptage afin d'étudier en profondeur l'intérêt implicite des utilisateurs : le filtre permet de compter le nombre de fois qu'une valeur d'un attribut figure parmi les items évalués par les utilisateurs, pour cela nous utilisons un filtre de Bloom. Un filtre de Bloom Broder et Mitzenmacher (2004) est un tableau de bits qui permet de tester d'une manière rapide l'appartenance d'un élément à un certain ensemble. Le FBC, décrit dans Broder et Mitzenmacher (2004) est une extension du filtre de Bloom standard qui fournit la possibilité de supprimer des éléments du filtre. Le vecteur de bits y est remplacé par un tableau d'entiers, où chaque position est utilisée comme compteur. L'insertion d'un élément est réalisée en incrémentant de 1 les entiers aux positions renvoyées par les fonctions de hachage. Le retrait est réalisé en décrémentant de 1 ces entiers. La question d'appartenance d'un élément au filtre est traitée en regardant si tous les entiers aux positions renvoyées par les k fonctions de hachage sont strictement positifs. Nous proposons de se baser sur l'ontologie du domaine afin d'extraire les attributs des items. Ensuite, nous utilisons les filtres de bloom avec compteur afin de stocker l'intérêt implicite des utilisateurs dans les attributs des éléments. Ceci se fait en suivant les étapes : (1) Pour chaque utilisateur, nous créons un filtre de bloom avec

compteur vide, (2) pour chaque élément noté par cet utilisateur, nous extrayons ses attributs et enfin (3) nous insérons ces attributs dans le filtre . Ainsi, le filtre contient tous les attributs des items précédemment notés par l'utilisateur.

Le module sémantique exploite l'ensemble des données ainsi que l'ontologie dans le but de définir les relations entre les utilisateurs, les items et les attributs. Ceci se traduit par la transformation sémantique des notes des utilisateurs. Nous nous intéressons tout d'abord au nombre d'occurrence des attributs qui ont été notés par un utilisateur. Nous appelons cette occurrence « la fréquence d'apparition » ou « coïncidence » : cette valeur correspond au nombre de fois que les valeurs des attributs se répètent dans les items notés par l'utilisateur. Cette valeur est extraite à partir des filtres de bloom avec compteurs.

La deuxième étape consiste à calculer la valeur sémantique (SV) en se basant sur la fréquence d'apparition. L'équation utilisée est la suivante (1).

$$sv_{u,i} = r_{u,i} + E[r_{u,*}] * \frac{\left|\sum_{j=1}^{F} C_j * W_j\right|}{N_u} \quad (1)$$

Avec F le nombre total des attributs, N_u le nombre total des items notés par l'utilisateur "u". C_j est la fréquence d'apparition de l'attribut j dans l'ensemble des items qui ont été notés par l'utilisateur et W_j étant le poids calculé à partir de la phase de la sélection des attributs par une analyse des composantes principales.

$E[r_{u,*}]$ est la moyenne des notes de l'utilisateur et $r_{u,i}$ est la valeur du rating initial donnée à l'item "i". L'utilisation de N permet de normaliser l'équation sémantique. Cette équation a l'intérêt de pouvoir prendre en compte des valeurs positives et/ou négatives comme note.

L'équation sémantique peut être appliquée à deux niveaux dans la recommandation. D'une part, nous pouvons appliquer cette équation à toutes les notes disponibles dans la base de données initiale, ce qui permet de mieux expliquer l'intérêt des utilisateurs pour les caractéristiques définissant les items notés (ajouter du sens à la note). D'autre part, nous pouvons faire le choix d'appliquer l'équation sémantique à la sortie de la recommandation. Supposons que le module de recommandation renvoie un résultat des top K items (les K items les plus pertinents) pour un utilisateur donné, avec une estimation de la note pour ces top K. Ces notes seront transformées en une note sémantique suivant l'équation (1) et les items proposés seront réordonnés en conséquence en top K', K' pouvant être inférieur ou égal à K.

Enfin, le module de recommandation utilise une technique de filtrage collaboratif basée sur une méthode de factorisation de la matrice pour générer des recommandations précises. Nous avons fait le choix d'utiliser la factorisation de matrice car cette technique a montré son efficacité comme méthode de filtrage collaboratif pour la recommandation Koren et Bell (2011).

3 Expérimentations

Jeux de données

Afin de pouvoir effectuer des expérimentations, nous avons besoin d'un jeu de données contenant les utilisateurs, les items à recommander et les ratings donnés pour ces items. Nous avons également besoin d'une description du domaine des items pour pouvoir appliquer le

module sémantique. Nous avons choisi le jeu de données MovieLens [1] . Toutefois, le jeu de données ne contient pas d'attributs pour les items (films). Nous avons donc décidé d'utiliser une ontologie du domaine pour extraire les relations entre les items et leurs attributs que nous allons peupler en se basant sur les données fournies par ImDB [2]. Cette base de données fusionnée est fournie également par GroupLens Cantador et al. (2011). Le jeu de données est composé de 2 113 utilisateurs et 855 598 notes sur 10 197 films. Il offre également six attributs : genre, réalisateurs, acteurs, pays, lieux et étiquettes. Le nombre total de valeurs distinctes pour ces attributs est 112881.

Afin de pouvoir réaliser nos expérimentations, nous nous sommes orientés vers l'utilisation de Mahout [3]. Nous utilisons Mahout comme une boite noire, capable de prendre un jeu de données, de l'analyser, et d'exécuter l'algorithme de recommandation. L'algorithme choisi est le SVD++ Koren et Bell (2011). Pour interagir avec des ontologies, nous avons décidé d'utiliser la bibliothèque Jena McBride (2002).

Nous avons testé le module sémantique au travers des deux approches :

— Application du module sémantique au jeu de données (Semantic dataset). Dans cette approche, l'application du module sémantique (que nous appelons "sémantisation") porte sur les données en entrée du système de recommandation. Il s'agit donc de traiter l'ensemble du jeu de données, avant son analyse par le système de recommandation et le filtrage collaboratif. Cette approche nous permet de retrouver des objets non pris en compte a priori. Cependant, puisque le module sémantique doit analyser tout le jeu de données, le temps de calcul peut être grandement augmenté par rapport à une analyse non sémantique.

— Application du module sémantique au résultat de la recommandation Top K (semantic top K). Dans cette première approche, nous cherchons à "sémantiser" les données en sortie du système de recommandation. Le système de recommandation fournit classiquement une liste de K objets, ordonnés par ordre décroissant de préférence. La sémantisation réordonne ces objets, et fournit une nouvelle liste plus pertinente. Cette approche est extrêmement légère, et permet d'améliorer la recommandation sans (trop de) perte de temps. De plus, avec cette approche, les paramètres de pondérations peuvent être personnalisés par l'utilisateur plutôt que de considérer l'ensemble du jeu de données. Néanmoins, une analyse a priori des données de l'utilisateur est nécessaire avant de personnaliser les paramètres de pondérations.

Dans la section suivante, nous présentons comment ces deux approches modifient les recommandations, et la qualité des résultats obtenus.

F-Mesure

Cette métrique est généralement utilisée dans l'évaluation des systèmes de recommandations. Cette métrique n'évalue pas la qualité de la prédiction des notes, mais la pertinence des items qui sont proposés aux utilisateurs. La F-mesure est une façon courante de combiner le rappel et la précision dans une seule métrique afin de faciliter la comparaison. le rappel étant la probabilité qu'un item choisi soit pertinent et la précision calcule la probabilité qu'un item pertinent soit choisi. tel que nous pouvons le constater dans la figure 1, nos approches donnent

1. http ://grouplens.org/datasets/movielens
2. http ://www.imdb.com/
3. https ://mahout.apache.org

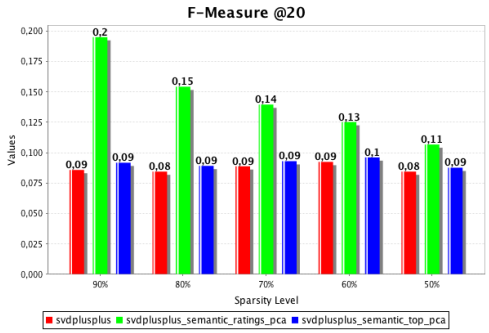

FIG. 1 – *Métrique "F-mesure".*

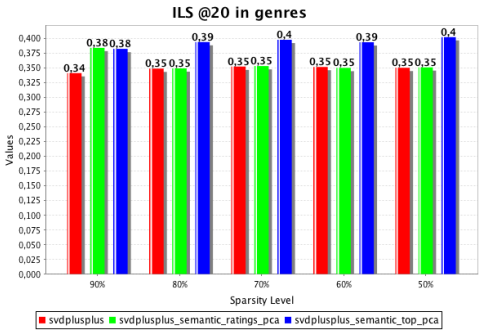

FIG. 2 – *ILS in "genre" attribute.*

de meilleurs résultats que la technique de matrice de factorisation (SVD++) sans sémantique. l'amélioration est plus prononcée pour le cas de la" sémantisation" du jeu de données.

ILS

ILS (*Intra-List Similarity*), appelée également ILD (*Intra-List Diversity*) mesure la diversité/similarité entre les items dans la liste des top-k présentée à l'utilisateur. Un bon système de recommandation doit trouver l'équilibre entre ces deux concepts diversité et similarité. En effet, des items trop diversifiés peut provoquer une confusion chez l'utilisateur, alors que recommander toujours les mêmes items peut ennuyer celui-ci. Le figure 2 représente cette mesure en se concentrant sur le attribut genres des films dans le top-k. Des valeurs élevées correspondent à une grande similarité. Ainsi, nous pouvons constater que notre approche permet de retourner des items plus similaires dans le top-k. Ceci est du au fait que nous prenons en compte l'intérêt pour les attributs afin d'identifier les items susceptibles d'intéresser l'utilisateur.

4 Conclusion

Nous avons proposé un système de recommandation qui repose sur deux concepts : relations sémantiques entre les données manipulées et un filtrage collaboratif basé sur la factorisation des matrices. Dans le but d'améliorer la pertinence des recommandations, nous avons étudié en profondeur l'intérêt implicite des utilisateurs pour les attributs des items. pour cela, nous appliquons une équation sémantique permettant de modifier les notes initiales des utilisateurs pour refléter leur intérêt pour les items.

Notre système de recommandation opère en plusieurs étapes : Nous comptons le nombre de fois qu'un attribut figure dans les items notés par les utilisateurs. Pour cela, nous nous sommes appuyés sur l'utilisation d'un filtre de bloom avec compteur. Ensuite, après passage par le module sémantique (transformation des notes des utilisateurs en appliquant l'équation sémantique) , les recommandations sont générées en utilisant une technique de matrice de factorisation (SVD++).

L'approche proposée dans ce papier a montré un intérêt pour la recommandation de films en utilisant le jeu de données MovieLens combiné à une ontologie de films, peuplé par les

données de IMDB. Nous avons fait le choix de travailler avec ce jeu de données car il est disponible et public et c est celui qui est le plus utilisé dans les expérimentations autour des systèmes de recommandation. Toutefois, nous avons conçu notre approche indépendamment du jeu de données que nous avions utilisé. Notre approche peut être utilisée comme une boite noire nécessitant de se connecter à une base de données des ratings disponibles, mais aussi à l'ontologie du domaine de l'application. Nous envisageons de tester notre approche sur des jeux de données provenant d'autres applications telles que la recommandation nutritionnelle ou de tourisme. Nous envisageons également d'intégrer dans notre approche la prise en compte des notes négatives (attributs non aimés par l'utilisateur).

Références

Balabanović, M. et Y. Shoham (1997). Fab : content-based, collaborative recommendation. *Communications of the ACM 40*(3), 66–72.

Breese, J. S., D. Heckerman, et C. Kadie (1998). Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pp. 43–52. Morgan Kaufmann Publishers Inc.

Broder, A. et M. Mitzenmacher (2004). Network applications of bloom filters : A survey. *Internet mathematics 1*(4), 485–509.

Cantador, I., P. Brusilovsky, et T. Kuflik (2011). 2nd workshop on information heterogeneity and fusion in recommender systems (hetrec 2011). In *Proceedings of the 5th ACM conference on Recommender systems*, RecSys 2011, New York, NY, USA. ACM.

Koren, Y. et R. Bell (2011). Advances in collaborative filtering. In *Recommender Systems Handbook*, pp. 145–186. Springer.

Koren, Y., R. Bell, et C. Volinsky (2009). Matrix factorization techniques for recommender systems. *Computer 42*(8), 30–37.

McBride, B. (2002). Jena : A semantic web toolkit. *IEEE Internet computing 6*(6), 55–59.

Summary

Recommender Systems aim at selecting and presenting first the information that users could be interested in. This work presents a Recommender System that relies on two concepts: semantic relations in data and a distributed collaborative filtering technique based on Matrix Factorization (MF). On the one hand, semantic technologies may increase relations among data, and thus, recommendation accuracy. On the other hand, MF grants highly accurate predictions in a parallelizable algorithm. Our proposal extends this technique by adding semantic relations to the process. Indeed, we deeply analyze the implicit user interests in the attributes of items. The experimentation phase uses MovieLens dataset and IMDb database. We compare our work against a semantic-less MF technique. Results show high accuracy in recommendations while preserving a high level of domain abstraction. Besides, we alleviate system workload by parallelizing the algorithms process.