Enhancing Collaborative Filtering Using Semantic Relations in Data

Manuel Pozo, Raja Chiky, Zakia Kazi-Aoul

Enhancing Collaborative Filtering using Semantic Relations in Data

Manuel Pozo, Raja Chiky, Zakia Kazi-Aoul

Institut Supérieur d'Eléctronique de Paris, LISITE Lab
28, rue Notre-Dame-des-Champs. 75006 Paris, France
{manuel.pozo,raja.chiky,zakia.kazi}@isep.fr
http://www.isep.fr

Abstract. Recommender Systems (RS) pre-select and filter information according to the needs and preferences of the user. Users express their interest in items by giving their opinion (explicit data) and navigating through the webpages (implicit data). In order to personalize users experience, recommender systems exploit this data by offering the items that the user could be more interested in. However, most of the RS do not deal with domain independency and scalability. In this paper, we propose a scalable and reliable recommender system based on semantic data and Matrix Factorization. The former increases the recommendations quality and domain independency. The latter offers scalability by distributing treatments over several machines. Consequently, our proposition offers quality in user's personalization in interchangeable item's environments, but also alleviates the system by balancing load among distributed machines.

Keywords: collaborative filtering, distributed systems, recommender system, semantic web technologies.

1 Introduction

The amount of information in the web has greatly increased in the past decade. This phenomenon has promoted the advance of Recommender Systems (RS) research area. The aim of these systems is to provide personalized recommendations. They help users by suggesting useful items to them, usually dealing with enormous amounts of data.

Typically, in order to create a top K items (K most relevant items) that should be presented first to the user, Recommender Systems study the interaction between users and items. For instance, users may rate items (such as films, books, etc.) using a 0-5 stars scale (explicit feedback), or user might just create a navigational path clicking links or purchasing items (implicit feedback). Hence, Recommender Systems exploit these feedbacks to set up recommendations. In the literature, Recommender Systems have been usually classified into two basic types: Content-based CB and Collaborative Filtering CF [1]. The former focus on the characteristics of the items in order to determine similarities between

them, and finally recommend similar items to the one the user liked in the past. The latter groups users according to their preferences profile and recommend items that people from the same group have already liked [2]. However, the correct exploitation of the data and the recommendation accuracy are the trend challenges of RS. On the one hand, seeking more relations between users and items may increase the recommendation quality. To this task, the RS might use semantic information about items, which enhance the data representation and help to find out the underlying reasons for which a user may or may not be interested in a particular item [3]. On the other hand, as the quantity and the variety of information constantly increase, the scalability and the domain generality of the system are two important aspects to alleviate the time processing and the heterogeneity of data.

In this paper, we propose a semantic collaborative filtering recommender system in order to improve the recommendations quality and preserve the domain genericity. In addition, we use an easily distributable collaborative filtering technique based on the well known ALS algorithm [4] to ensure the system scalability.

This article is structured as follows: in section 2, we present related work. In section 3, we explain our general approach. In section 4, we expose the experimentation phase and the comparisons. Finally, we discuss in section 5 about the results and the future work.

2 Related Work

Recommender Systems (RS) use users feedback to predict interest in items. Nowadays, the current challenges for Recommender Systems are to improve their recommendations quality, their items domain genericity (i.e the domain independency) and their scalability. In order to address these issues, a trend topic is the use of semantic knowledge. For instance, [5] and [6] propose Content Based (CB) recommender systems based on items domain descriptions, such as ontologies. The approaches associate items and keywords that characterize them, creating a bag of words for each item. Hence, the items similarity looks for common keywords in bags. Recently, [7] proposed a recommender system architecture that facilitates the integration of heterogeneous data. The system creates a semantic graph representation of data in order to weight the relations between the nodes of the graph. The system explores the graph and chooses the best weights for predictions tasks.

Nevertheless, some authors have already demonstrated the effectiveness of more extended Collaborative Filtering (CF) techniques [2]. For example, [8] proposed a CF method that took the implicit users feedback into account. Authors used Alternating Least Squares (ALS) CF technique that is based on Matrix Factorization techniques. The ALS method can be easily distributed among machines, and thus, enhances the scalability of the system. [9] focus on the relevance of items in the ranking, rather than in the items ratings prediction using a technique similar to ALS. [10] proposes another approach that improves CF

techniques using item-item similarity based on items description in Wikipedia. In cases where sparsity is too high, the system guess artificial ratings for new items regarding the last user ratings on similar items.

Differently, but also aiming to improve CF, [11] suggests a three-layer representation for data: users, interests and items. For a user, an interest is a characteristic that an item must have. For an item, an interest is one of its attributes. Thus, authors construct a correlation matrix graph containing users hidden interests. Other approaches encourage the use of multi-criteria feedback to improve recommendations quality [12][13][14]. In order to explain an overall rating in items, authors also analyze these item attributes feedbacks and also predict their rating.

In [15] and [16], authors proposed an approach for reducing dynamically the big number of item features needed for the recommendations. Both approaches construct an items-attributes matrix that they reduce by using SVD algorithm, alleviating sparsity and noise.

However, most of the approaches lack distribution environment and domain genericity [3][17][11]. Thus, we propose a general semantic CF method based on ALS [8][9] and an ontology semantic technology. ALS allows the distribution of the computation among several machines using Matrix Factorization techniques, whereas ontologies add information representation and domain independency to the system.

3 General Approach

The proposed architecture includes a collaborative filtering engine that relies on domain knowledge about the items. This knowledge is represented using an ontology that will enhance the recommendation quality. The architecture is represented in figure 1 and is composed of two main modules: a semantic module and a recommendation module.

The semantic module exploits the domain ontology to define the relations between items and their attributes. Because the number of those attributes could be huge, we apply a dimension reduction technique based on **Principal Component Analysis** (PCA) in order to select the most representative ones (pre-analysis phase). The PCA also provides weights for attributes to give them more or less importance in the recommendation process. The recommendation module is based on matrix factorization method. Both modules will be described in the following subsections.

3.1 Semantic Module

A user could be interested in an item because of one or more of its attributes. For example, a user might appreciate a set of movies only because they have in common his favorite actor. Therefore, the purpose of this module is to take into account the attributes that compose items to better serve the user. For that,

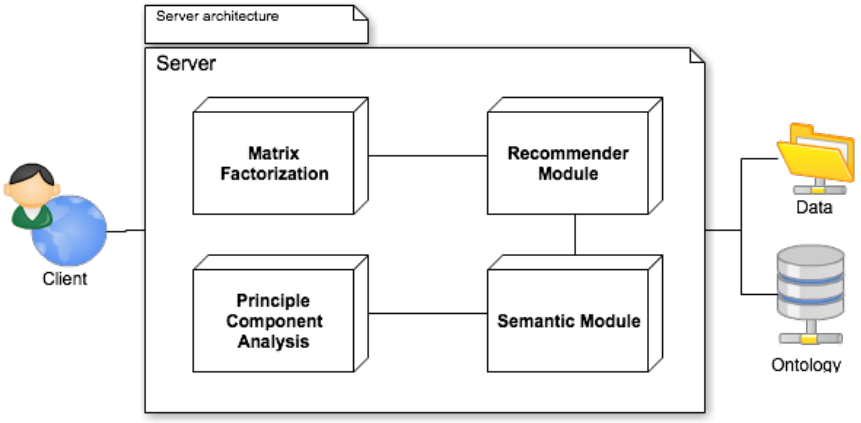

Fig. 1. Global architecture of our approach

its relies on a domain ontology (i.e. an ontology that describes the domain for which we want to implement a recommendation system).

To transform traditional ratings into "semantic rating", we focus first on the number of occurrence of attributes that were rated by a user. This occurrence that we call **occurrence frequency** or **coincidence** is the number of times the attribute values of the items rated by the user are repeated. The second step is to calculate the semantic value (sv) based on the occurrence frequency. The equation used is presented in (1):

$$sv = r + E[r] * \frac{\left|\sum_{i=1}^{F} c_i * w_i\right|}{N} \tag{1}$$

Where r is the value of the initial rating, E[r] is the average of the user ratings, F is the total number of attributes, c_i is the occurrence frequency of the attribute i in the set of items that have been rated by the user, w_i is the weight calculated from the PCA phase, and N is the total number of items rated by the user. This equation takes into account positive and/or negative ratings. It can be used at two levels in the Recommendation. On the one hand, we can apply it to all ratings available in the original database, which helps to explain the users interest for particular characteristics of the rated items. On the other hand, we can choose to apply the semantic equation to the output of the recommendation. Indeed, suppose the recommendation module returns the result as top K items for a user, with an estimation of ratings of this top K. These user ratings will be transformed into a semantic rating according to the equation (1) and will be reordered accordingly in top K'. K' can be less than or equal to K.

3.2 Matrix Factorization

As its name suggests, the matrix factorization consists on decomposing a matrix into two or several matrices, which results in the same original matrices after their multiplication.

This method allows discovering latent or hidden relationships between a user and the items. For instance, we may suppose that two users have highly rated a movie because they like its actors/actress on it, or because it is an action movie, which may be their favorite movie genre. Thus, if we are able to discover these hidden-reasons, we may be capable of predicting the interest on each item, because the associated characteristics of preferences may correspond with the ones that the items contain.

The Matrix Factorization is closely related to Singular Value Decomposition (SVD), which is widely used to identify hidden relation factors in information retrieval. Collaborative Filtering method may adapt this technique in order to work on the typical sparse rating matrix.

The Matrix Factorization models the interaction between users and items by applying a scalar product of two vectors representing the latent features in a space of dimension f. As a consequence, each item i is associated to one vector $q_i \in$, as well as each user u is associated to a vector p_u. The dot product of both vectors represents an estimation of the ratings that the user may give to the item.

$$r'_{ui} = q_i^T p_u$$

Hence, the challenge is to obtain all these vectors q_i and p_u; by solving the equation (2)

$$min_{q*,p*} \sum_{u,i \subset K} (r_{ui} - q_i^T p_u)^2 + \lambda(\parallel q_i \parallel^2 + \parallel p_u \parallel^2) \tag{2}$$

Where K is the set of pairs (u, i) in which the ratings of $r_{u,i}$ are available, and λ is a regularization parameter that allows controlling the learning model [4].

Alternating Least Squares (ALS) To solve equation (2), [18] proposes an approach called Alternating Least Squares (ALS). In this equation, the vectors q_i and p_u are unknown, and thus the equation is not convergent. However, if we are able to fix one of them, the optimization problem becomes square, and we are able to optimally solve the equation. Thus, the ALS techniques fix in each iteration q_i or p_u: when p_u is fixed, the system computes q_i by least squares, and vice versa. This fact allows an optimal convergence in the iteration of the equation. This method is highly interesting because it is easily distributable among multiple machines. In fact, the system computes each q_i independently of the other factors, as well as for each p_u.

Table 1. Experimentation: Dataset

Users	Films	Ratings	Features
100	1232	11019	9

4 Experimentation

4.1 Experimental Dataset

In order to implement the semantic module, we need a dataset and an items domain description that defines the elements. We choose MovieLens, a movies dataset, which contains a good number of items (films) and users ratings.

In addition, we need the relative information about the attributes of the items, which is not provided in the MovieLens dataset. To alleviate this lack, we use an ontology in order to associate automatically the movies with their attributes. Filling a dataset might be a hard work because we have to deal with each item, its properties and its attributes [19]. As a consequence, it may become an obstacle for the experimentation [20]. For this task, we use the IMDb database that represents the relations described in the ontology. Thus, the use of both sources (IMDb and MovieLens) facilitates the integration of the semantic module, which needs items (MovieLens) and their attributes (IMDb). Note that the titles might have different representations, and thus, we needed a clean up phase in order to merge both sources.

Moreover, the ratings dataset in MovieLens and the associated attributes from IMDb are too big and this does not help to apply the semantic equation, which may require important processing time. In order to test our approach, we reduce the number of users to the 100 users who have more rated movies and we focus on only 9 attributes (actor or actress, color, editor, director, genre, language, producer, writer and productions year). Table 1 summaries information about our experimental dataset.

4.2 Semantic Recommender System

As far as we know, there is no existing recommender system that handles or facilitates the usage of the web semantic technologies. In addition, such a system might be really complex to implement in order to validate our approach. Thus, we decided to add a semantic layer to an existing recommender system. Moreover, our objective is to implement a collaborative filtering algorithm based on Matrix Factorization in order to be able to analyze huge volume of data. This algorithm is implemented in numerous recommender systems libraries. Hence, we choose a system that better fit our work environment and our constraints: the system has to use a collaborative filtering technique and be easily distributed for handling scalability issues. For our experimentations, we use Myrrix [21]. It is a natural evolution of the widely extended Apache Mahout libraries [22]. We consider Myrrix as a black box: it takes a dataset, analyzes it and executes

the recommendation technique. To interact with the ontologies, we use the Jena library [23].

We test the semantic recommender system using two approaches:

- **Semantic Top K**: In this first approach, we aim to apply the semantic layer on the output data. Typically, recommender systems provide a list of K items ordered by user's preference. In this case, the semantic module re-orders these items and gives back a new list in an other relevance order. This approach allows better recommendations in a reduced execution time.
- **Semantic Dataset**: In this second approach, we apply the semantic layer on the input of the system. A pre-analysis is done on the dataset before it goes into the recommender system. This approach allows finding out new items that were not taken into account a priori. However, the semantic module needs to analyze the whole dataset increasing the process time in comparison with the non-semantic analysis.

In the next section, we present how these approaches have modified the recommendations, and the quality of the obtained results.

4.3 Results

We tested our approach using the recommender system Mahout/Myrrix and adding the semantic layer either at the input or the output. From the experimental dataset, we chose the user who has rated more items (the user with id 13 and 636 ratings). Then, we took out 60 ratings of this user (all of them have been rated with a 5 over 5). Next, we ask to the system a list of 60 items for this concrete user. The table 2 presents the top 10 items (over the 60 items) with the higher prediction scores in the three approaches (non-semantic, semantic dataset and semantic top K).

Semantic Output: Semantic Top K The semantic top K approach works over the 60 films asked to Mahout/Myrrix. The semantic algorithm re-evaluates the predictions associated to each film and gives them back in a new order. The table 2 shows up the modification in the top K. In concrete, we can see that the film 121 is now in the most preferable, when before it was the third one. The film 237 has gone from the 10th place to the 3rd one. Moreover, note that the half of items in the semantic top K items is absent in the non-semantic top 10; yet, they are still in the list of 60 extracted films.

Semantic Input: Semantic Dataset In this approach, the semantic layer is applied on to the whole dataset before Mahout/Myrrix analyzes it. The results in the table 2 present some small differences against the non-semantic recommendations. The similarities are more important than semantic Top K, in particular the first top 7 recommended films are the same but in different order. However, we also see the emergence of new films (202 film in position 9).

Table 2. Experimentation: Top-10 results for a request of the user 13 of MovieLens

Non-Sem.		Sem.Ratings		Sem.Top K	
ID	Prediction	ID	Prediction	ID	Prediction
56	0.720	127	0.74	121	1.12
127	0.690	56	0.73	56	1.06
121	0.610	121	0.63	237	1.049
135	0.600	100	0.60	202	1.03
100	0.570	135	0.58	127	1.00
50	0.560	50	0.579	13	0.91
234	0.550	234	0.572	423	0.902
204	0.519	237	0.53	993	0.90
181	0.517	202	0.518	161	0.894
237	0.500	181	0.514	50	0.883

In order to study in more details the impact of the semantic layer on the results, we performed some statistical comparison techniques. We use 3 evaluation methods, all of them are implemented in the Mahout/Myrrix libraries.

- The Area Under Curve (AUC) represents the probability to give a higher rating to a relevant random item than an irrelevant item. In this measure, the higher value is the better one.
- The Precision and Recall (PAR) measures the relevance of the items in the top K recommended list. It combines the fraction of all recommended items that are relevant (precision) and the fraction of all relevant items that were recommended (recall). In this case, the higher value is the better one.
- Estimated Strength (ES) measures the quality and reliability of the results. In this measure, the lower value is the better one.

The table 3 presents the obtained results using or not the semantic layer in the Top-K, and thus it may compare the approaches. The non-semantic dataset contains 0-5 scaled ratings whereas the semantic dataset ratings are between 0 and 10. In the row Semantic (0-5), the ratings have been scaled to 0-5 for a quick comparison with the non-semantic dataset.

Table 3. Comparisons: different scale datasets.

Dataset	AUC	PAR	ES
Non-Semantic (0-5 ratings)	0.6233	0.0692	0.929
Semantic (0-5 ratings)	0.6448	0.0728	0.937
Semantic (0-10 ratings)	0.7945	0.0329	0.1651

From this table, we can say that:

– Concerning semantic Dataset (0-10), we got good results for the AUC and the ES measures in comparison to the non-semantic approach.
– Semantic Dataset (0-5): In this case, we observe good results in the AUC and PAR measures. Moreover, the ES values in both semantic and non-semantic approach are very close.

Finally, note that these results are preliminary, and can be improved. Indeed, a closer analysis of the data set would improve the weightings applied to the semantic equation (equation (1)). Further, adding attributes and more users and feedbacks will improve the semantic equation, and therefore the quality of recommendations.

5 Conclusion

Recommender systems face a substantial challenge when dealing with huge amount of data. In this paper, our main goal was to help solving this problem by describing an easy development of a distributed recommender system. This was possible by using a powerful algorithm of Collaborative Filtering called ALS which can be easily distributed among several machines. In addition, we propose a semantic equation that allows to reorder the top K suggested items to the user according to his preferences (based on underlying reasons for which a user may or may not be interested in a particular item). The proposed approach in this paper has demonstrated an efficiency on the recommendation of films by using the MovieLens dataset and a film ontology, which was filled from IMDb data source. We decided to choose this domain because the MovieLens dataset is public and highly available.

Nevertheless, the design of the approach is very independent of the application domain. Our system may be used as a black box, where we can connect a database to introduce ratings as well as the application domain ontology. Our future work will focus on two main aspects: (1) To reproduce similar experimentation in different domain datasets in order to prove the genericity of our system, (2) To improve the semantic-layer in order to get better recommendation results and to reduce the time execution of the system.