User Semantic Preferences for Collaborative Recommendations

Sonia Ben Ticha[1,2], Azim Roussanaly[2], Anne Boyer[2], and Khaled Bsaies[1]

[1] URPAH Teams Tunis, Tunisia
[2] KIWI Team LORIA laboratory Nancy, France

Abstract. Personalized recommender systems provide relevant items to users from huge catalogue. Collaborative filtering (CF) and content-based (CB) filtering are the most widely used techniques in personalized recommender systems. CF uses only the user-rating data to make predictions, while CB filtering relies on semantic information of items for recommendation. In this paper we present a new approach taking into account the semantic information of items in a CF system. Many works have addressed this problem by proposing hybrid solutions. In this paper, we present another hybridization technique that predicts users 'preferences for items based on their inferred preferences for semantic information. With this aim, we propose a new approach to build user semantic profile to model users' preferences for semantic information of items. Then, we use this model in a user-based CF algorithm to calculate the similarity between users. We apply our approach to real data, the MoviesLens dataset, and compare our results to standards user-based and item-based CF algorithms.

Keywords: Recommender systems, collaborative filtering, semantic information, user modeling.

1 Introduction

Collaborative filtering (CF) and content-based (CB) filtering are the most widely used techniques in Personalized Recommender Systems (RS). The fundamental assumption of CF is that if users X and Y rate n items similarly and hence will rate or act on other items similarly [7]. In CB, user will be recommended items similar to the ones he preferred in the past. However, CF and CB techniques must face many challenges [9] like the data sparsity problem, the scalability problem for big database with the increasing numbers of users and items and the cold start problem. To overcome the disadvantages of both techniques and benefit from their strengths, hybrid solutions have emerged. In this paper, we present a new approach taking into account the semantic information of items in a CF system. In our approach, we design a new hybridization technique, called User Semantic CF (USCF), which predicts user preferences for items based on their inferred preferences for semantic information.

Our contribution is summarized as follows: (i) we propose a new approach for building *user semantic model*, that inferred the user preferences for semantic information of items, (ii) we define a classification of attributes and propose a suited algorithm for each class, (iii) for each relevant attribute, we build the *user semantic*

C. Huemer and P. Lops (Eds.): EC-Web 2012, LNBIP 123, pp. 203–211, 2012.

attribute model using the suited algorithm, (iv) we provide predictions and recommendations by using the user semantic model in a user-based CF algorithm [5], (iv) we perform several experiments with real data from the MoviesLens data sets which showed improvement in the quality of predictions compared to only usage CF.

2 Related Work

RS have become an independent research area in the middle 1990s. CF is the most widespread used technique in RS, it was the subject of several researches [5][6][**7**]. In CF user will be recommended items that people with similar tastes and preferences liked in the past. CB is another important technique; it uses techniques developed in information filtering research [15][16]. CB assumes that each user operates independently and recommends items similar to the ones he preferred in the past. The major difference between them is that CF only uses the user-item ratings data to make recommendations, while CB rely on the features of items for predictions.

To overcome the disadvantages of both techniques and benefit from their strengths several RS use a hybrid approach by combining CF and CB techniques. The Fab System [1] counts among the first hybrid RS. Many systems have been developed since [3][10]. In [2], authors integrate semantic similarities of items with item rating similarities and used it in item based CF algorithm to generate recommendations. Most of these hybrid systems ignore the dependency between users' ratings and items' features in their recommendation process; taking account of this link can improve the accuracy of recommendation. In [8], this dependency was computed by using TF/IDF measure to calculate the weight of item feature for each user. In [14], authors are inferring user preferences for item 'tags by using several measures. This work is suitable only for item 'tags and cannot be used for others kinds of attributes.

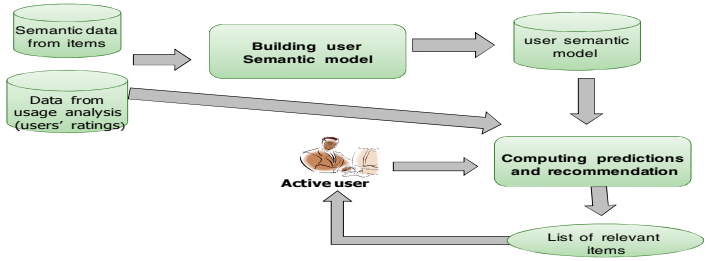

Fig. 1. Architecture of our system: USCF approach

3 User Semantic Collaborative Filtering (USCF) Approach

Our system consists of two components as shown in Fig. 1. The first builds the user semantic model by inferring user semantic preferences from user ratings and item features. The second predicts for each user a list of relevant items based on the user-based CF algorithm and using the user semantic model for computing similarities between users. USCF uses only data from usage analysis and semantic information of items. Table 1 describes all used symbols. So, we define:

- **Data from usage analysis**: the usage analysis profile of item i is given by the following ratings vector $I_i=(r_{1,i},r_{2,i},...r_{u,i},...,r_{N,i})$, $r_{u,i}$ is the rating of user u on item i; it can be either a missing value or a number on a specific scale if user u rated item i.
- **Semantic information from items**: we assume that item is represented by *structured data* [16] in which there is a small number of attributes, each item is described by the same set of attributes, and there is a known set of values that each attribute may have. In the following, we will use the terms *feature* to designate a value of an attribute. The semantic attribute based profile of item i on attribute A is given by the features vector: $F_{Ai}=(b_{Ai,1},..., b_{Ai,f}, ...,b_{Ai,LA})$, where:

$$b_{\mathrm{Ai},f} = \begin{cases} 0 \; if \; f \; is \; not \; a \; feature \; of \; item \; i \\ 1 \; if \; f \; is \; a \; feature \; of \; item \; i \end{cases} \quad (1)$$

Table 1. Description of the used symbols

Symbol	Meaning	Symbol	Meaning
N	number of users	I	item-user ratings matrix, $(r_{u,i})_{1..M,1..N}$
M	number of items	F_{Ai}	semantic attribute based profile of item i
S_I	set of items described by I_i	S_{FA}	set of items defined by F_{Ai}
I_i	usage analysis profile of item i	F_A	item semantic attribute matrix (M,L_A)
L_A	number of features of A	Q	user semantic model (matrix $(q_{u,k})_{1..N,1..L}$)
K_A	Number of clusters associated to A	U	User-item rating matrix (I transposed)
Q_A	user semantic attribute model (matrix $(q_{u,Ak})_{1..N,1..KA}$)	$q_{u,Ak}$	inferred preference of user u on feature(s) of A labeling the cluster k
A	Relevant attribute	$q_{u,k}$	inferred preference of u on feature(s) k

Otherwise, we must distinguish between two kinds of attributes: *multi-valued* and *mono-valued* attribute. For a same item, if an attribute can have many values, then it is a multi-valued attribute (a *movie* can have many *genres)*; while if it must have only one feature it is called mono-valued attribute (a *movie* has only one *director*).

Furthermore, all item attributes do not have the same degrees of importance to users. There are attributes more relevant than others. For instance, the *movie genre* can be more important, in the evaluation criteria of user, than the *release date*. Experiments that we have conducted (see section 4) confirmed this hypothesis. In this paper, we assume that relevant attributes will be provided by a human expert.

3.1 Building the Users Semantic Model

In our approach we have defined two classes of attributes: *dependent attribute* which having very variable number of features. This number is directly correlated to the number of items. Thus, when the number of items is increasing, the number of features is increasing also (*actors of movies; user tags*). *Non dependent attribute* which having a very few variable number of features that is no correlated to the number of items. Thus, the increasing number of items has no effect on the number of features (*movie genres*). For each class, we have defined a suited inferring user semantic preferences algorithm. For the dependent attribute, we propose techniques issues from information filtering research like TF/IDF. For non dependent attribute, we use machine learning algorithms. The aim of this paper is to present our solution for non dependent attributes, dependent attributes will be addressed in future works.

For each relevant attribute A, we have built the corresponding user semantic attribute model Q_A that provides the inferring user preferences for its features. The user semantic model Q is so the horizontal concatenation of all users semantic attributes models. For example, assume that we have a movies Data set with users ratings and we want to infer the preference $q_{u,action}$ of user u on the *action movies*. This means computing an aggregation overall ratings of user u on all action movies (eq. 2). The aggregation function can be a simple function like the average (AVG), or more complicated mathematical function like TF/IDF, or special user-defined function. For non dependent attribute, we choose to define a special user function, so we use a clustering algorithm to learn the user semantic attribute model.

$$q_{u,\,genre=action} = AGGR_{\,i.genre=action}\, r_{u,i} \tag{2}$$

3.2 User Semantic Attribute Model for Non Dependent Attribute

The idea is to partition S_I in K clusters; each cluster is labeled by a feature or a set of features of A ($K<=L_A$). Thus, the cluster center $C_{A,k}=(q_{1,Ak},..., q_{u,Ak},...,q_{N,Ak})$ modeled the inferred users preferences for the feature(s) associated to cluster k. For example, assume that we have a movies dataset and we want to infer users' preferences on *movie genre*. The attribute *genre* has L_{genre} features, if each cluster is labeled by a feature, then we will have L_{genre} clusters. Assume that the feature *action* is labeled the cluster l, then after running the clustering algorithm, the center of cluster l provides the *action*-users profile $C_{genre,1}=(q_{1,genre1},..,q_{u,genre1},..,q_{N,genre1})$ where $q_{u,\ genre1}$ provides the inferring preference of user u on *action movie*. Matrix Q_A is so obtained by calculating the transposed matrix of C_A. However, the question is what clustering algorithm to use? As we have already said, we have two kinds of attributes, multi-valued attribute and mono-valued attribute. For multi-valued attribute, a same item can belongs to many clusters, so the clustering algorithm must provide non disjointed clusters, while, for mono-valued attribute, an item must belong to only one cluster so the clustering must provide disjointed clusters. In previous work [11] we addressed the multi-valued attribute and we choose the Fuzzy C Mean as a fuzzy clustering algorithm. In this paper, we present our solution for mono-valued attribute.

After a study of several clustering algorithms, we have chosen the K-Mean clustering algorithm for its simplicity. The result of K-mean is depending on the number K of clusters, and the initial set of cluster centers. In this paper, we design an algorithm for the initialization step of the K-mean algorithm.

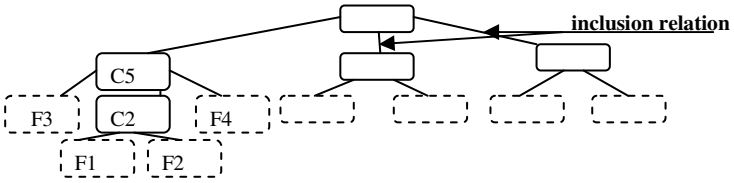

Fig. 2. Ontology of an attribute

Algorithm of the K-Mean initialization step

To determine the number of clusters and their respective initial centers, we have defined two thresholds: *MinNbRaIt* that defines the minimum number of item ratings and *MinNbItClust* which indicates the minimum number of items by cluster in the initialization step. Each cluster is labeled by a feature and is created according to formula 3, its initial center is the mean value of its items. Among all clusters created, only those checking the selection criteria described by formula 4 are preserved. Thus, user preferences cannot be inferred for features assigned to non selected clusters. Because of the data sparsity, the number of these features can be important. To solve this problem, we use an ontology describing the non dependent attribute, thus a cluster can be labeled by a single or several features.

$$C_k = \{item\ i\ \in\ S_I / ratings\_Number(\ i_{.attribute=f_k}\ ) \geq \text{MinNbRaIt}\} \quad (3)$$

$$|C_k| >= \text{MinNbItClust} \quad (4)$$

We assume having an ontology describing the attribute. The concepts of the ontology (solid line in Fig. 2) are interconnected hierarchically, and the leaf nodes describe the features of the attribute (dashed in Fig. 2). For example, features F1 and F2 are included in the concept C2; features F3 and F4 and concept C2 are included in concept C5. Each feature does not check the selection criteria defined above, will be replaced by its closest ancestor meeting the criteria in the ontology. In the example described in Table 2, F1 and F3 satisfy the selection criterion, so a cluster will be assigned to each. However, as F2 does not satisfy the criteria, it will be replaced by its father C2; Similarly, C2 does not satisfy the criteria itself, it will be replaced by C5. In addition, F4 does not check the criteria; it will also be replaced by C5. The number of items assigned to the concept C5 is equal to 8 (5+3) and it's greater than MinNbItClust. As, C5 satisfies the criterion, a cluster will be associated to it. Using this initialization algorithm, we will be able to infer user preferences for the concept C5 which groups features F2 and F4.

Table 2. Example, *MinNbItClust = 6*

Feature	Nb items with $ratings\_Number >= MinNbRaIt$
F1	10
F2	5
F3	12
F4	3

3.3 Computing Predictions and Recommendation

To compute predictions we use the user semantic model Q in a user-based CF algorithm [5] for computing similarities between users. User-based CF is based on the k-Nearest-Neighbors algorithm. Formula (5) computes similarities between two users with the Pearson correlation introduced by Resnick et al. [5]; $\bar{q}_v$ is the average of inferred preferences of user *v* on features. Then, the prediction of rating value of active user u_a on non rated item *i* was computed by formula 6; V denotes the set of the nearest neighbors that have rated item *i*.

$$sim(u_a, v) = \frac{\sum_k (q_{u_a,k} - \bar{q}_{u_a})(q_{v,k} - \bar{q}_v)}{\sqrt{\sum_k (q_{u_a,k} - \bar{q}_{u_a})^2} \sqrt{\sum_k (q_{v,k} - \bar{q}_v)^2}} \quad (5)$$

$$pr(u_a, i) = \bar{r}_{u_a} + \frac{1}{\sum_{v \in V} |sim(u_a, v)|} \sum_{v \in V} sim(u_a, v)(r_{v,i} - \bar{r}_v) \quad (6)$$

3.4 USCF Algorithm

Our approach resolves the scalability problem for several reasons. First, the building process of user-semantic model is fully parallelizable and can be done offline. Second, this model allows a dimension reduction since the number of columns in user semantic model Q is much lower than those of user rating matrix U. Third, the computing of similarities between users can be done offline. In addition, USCF alleviates the data sparsity problem by providing solution to the neighbor transitivity problem in which users with similar preferences may not be identifies as such if they haven't any items rated in common. Indeed, the number of missing values is much lower in Q than in U; thus, all similarities can be computed. This is not the case with U, because similarities between users who have no co rated items cannot be computed.

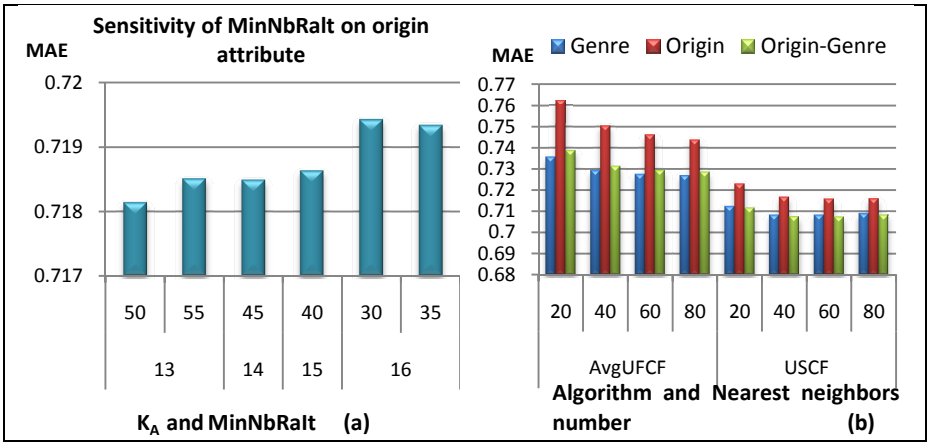

Fig. 3. Comparative results for USCF: (a) by varying the MinNbRaIt; (b) between USCF and AvgUFCF on all relevant attributes.

4 Performance Study

In this section, we study the performance of the USCF algorithm against the standard User-Based CF [5] (UBCF), the standard Item-Based CF (IBCF) [6] and Average User Feature CF algorithm (AvgUFCF). For IBCF, predictions have been computed using the Adjusted Cosine correlation measure which provides, according to [6], best prediction accuracy. In AvgUFCF user semantic model has been built by using the average (AVG) as an aggregation function (formula 2). We evaluate these algorithms in terms of predictions relevancy by using the Mean Absolute Error (MAE) (7).

$$MAE = \frac{\sum_{u,i} \left| pr_{u,i} - r_{u,i} \right|}{d} \quad (7)$$

d is the total number of ratings over all users, $pr_{u,i}$ is the predicted rating for user u on item i, and $r_{u,i}$ is the actual rating. Lower the MAE is, better is the prediction.

4.1 Experimental Datasets

We have experimented our approach on real data from the MovieLens1M dataset [4]. For the semantic information of items, we have used the HetRec 2011 dataset [12]. The *genre* and the *origin country* of movies have been used as non dependent attributes. Movie' genre is a multi-valued attribute whereas origin country is mono-valued. W3C movie ontology [13] has been used for describing the origin of movie.

We have filtered the data by maintaining only users with at least 20 ratings and the movies origins existing in the ontology. After the filtering process, we have obtained a data set with 6027 users, 3559 movies, 19 genres, 44 origins. The usage data set has been sorted by the timestamps, in ascending order, and has been divided into a training set (including the first 80% of all ratings) and a test set (the last 20%). We have tried several distance measures in the clustering algorithm; the cosines distance has provided the best result.

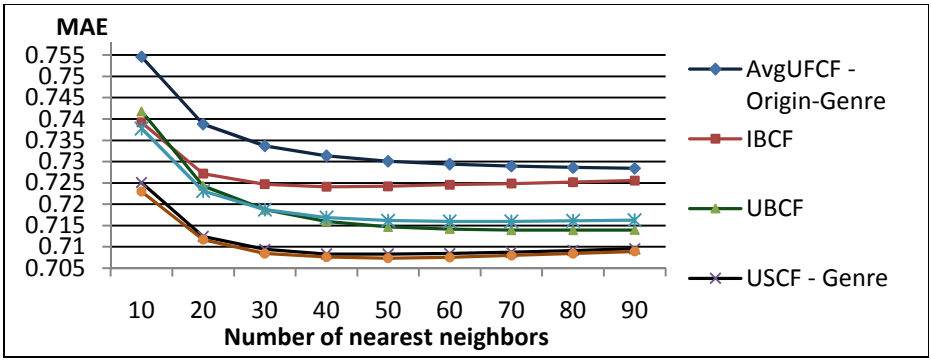

Fig. 4. Prediction accuracy for USCF v. IBCF, UBCF and AvgUFCF

4.2 Results

It should be noted that the inferring user preferences for the attribute *genre* have been addressed in a previous work [11]. Therefore, we will not detail the experiments conducted for this attribute in this paper. In Fig. 3 (a), the MAE has been plotted with respect to the MinNbRaIt parameter. It compares the K-Mean initialization algorithms on the attribute *origin* for MinNbItClust =9 and 60 neighbors. We note that the accuracy of recommendations improves with the decreasing number of clusters K_A. In addition, the MAE converges for 50 ratings; this shows the impact of MinNbRaIt on the accuracy of the recommendations. The plots in Fig. 3 (b) show that the *genre* provides better results than the attribute *origin*, for both algorithms USCF and AvgUFCF

and regardless of the number of neighbors. Therefore, we can conclude that the *genre* is more relevant than the *origin*. Fig. 4 depicts the prediction accuracy of USCF, in contrast to those produced by IBCF, UBCF and AvgUFCF using the best parameters of each algorithm. In all cases, the MAE converges between 60 and 70 neighbors, however, our algorithm results in an overall improvement in accuracy. This improvement can be explained by many reasons. First, the use of semantic information of items in CF. Second, user semantic model is built according to a collaborative principle; ratings of all users are used to compute the semantic profile of each user. It is not the case of the AvgUFCF algorithm; this may explain its results despite taking into account the semantic aspect. Third, the choice of the attribute can have significant influence on improving the accuracy. Lastly, matrix Q has few missing values, so, it allows inferring similarity between all users.

5 Conclusion and Future Work

In this paper, we have designed a new hybridization technique, which predicts users' preferences for items based on their inferred preferences for semantic information. We have defined two classes of attributes, the *dependent attribute* and the *non dependent attribute* and we have proposed an approach for inferring user semantic preferences for each class. Our approach provides solutions to the scalability problem, and alleviates the data sparsity problem by reducing the dimensionality of data. The experimental results show that the USCF algorithm improves the prediction accuracy compared to usage only approach (UBCF and IBCF) and hybrid algorithm (AvgUFCF). Furthermore, we have experimentally shown that, all the attributes don't have the same importance to users.

An interesting area of future work is to use machine learning techniques to automatically determine the relevant attributes. We will also further study the extension of the user semantic model to the dependent attribute and non structured data; study the use of this model in case-based RS to solve the cold start problem; and lastly, study the impact of using others machine learning algorithms for building the user semantic attribute model for non dependent attribute and comparing their results.

References

1. Balabanovic, M., Shoham, Y.: Fab: content-based, collaborative recommendation. Comm. ACM 40(3), 66–72 (1997)
2. Mobasher, B., Jin, X., Zhou, Y.: Semantically Enhanced Collaborative Filtering on the Web. In: Berendt, B., Hotho, A., Mladenič, D., van Someren, M., Spiliopoulou, M., Stumme, G. (eds.) EWMF 2003. LNCS (LNAI), vol. 3209, pp. 57–76. Springer, Heidelberg (2004)
3. Burke, R.: Hybrid Web Recommender Systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) The Adaptive Web. LNCS, vol. 4321, pp. 377–408. Springer, Heidelberg (2007)
4. MovieLens, `http://www.movielens.org/` (retrieved: January 2012)
5. Resnick, P., Iacovou, N., Suchak, M., Bergstorm, P., Riedl, J.: Grouplens: an open architecture for collaborative filtering of netnews. In: Proc. of Conf. on Computer Supported Cooperative Work, pp. 175–186. ACM, North Carolina (1994)

6. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: Proc. 10th Int'l WWW Conf. (2001)
7. Su, X., Khoshgoftaar, T.M.: A survey of collaborative filtering techniques. In: Adv. in Artif. Intell. (January 2009)
8. Symeonidis, P., Nanopoulos, A., Manolopoulos, Y.: Feature-Weighted User Model for Recommender Systems. In: Conati, C., McCoy, K., Paliouras, G. (eds.) UM 2007. LNCS (LNAI), vol. 4511, pp. 97–106. Springer, Heidelberg (2007)
9. 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 Knowledge and Data Engineering 17(6), 734–749 (2005)
10. Belén Barragáns-Martínez, A., Costa-Montenegro, E., Burguillo, J.C., Rey-López, M., Mikic-Fonteand, F.A., Peleteiro, A.: A hybrid content-based and item-based collaborative filtering approach to recommend TV programs enhanced with singular value decomposition. Information Sciences 15, 4290–4311 (2010)
11. Ben Ticha, S., Roussanaly, A., Boyer, A.: User Semantic Model for Hybrid Recommender System. In: The First Int'l Conf. SOTICS, Barcelona, Spain (October 2011)
12. 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems (HetRec 2011) at the 5th ACM Conf. on Recommender Systems (RecSys 2011)
13. Bouza, A.: MO – the Movie Ontology (2010), `http://movieontology.org`
14. Sen, S., Vig., J., Riedl, J.: Tagommenders: connecting users to items through tags. In: Proc. of the 18th Int'l Conf. on WWW, Madrid, Spain (April 2009)
15. Lops, P., de Gemmis, M., Semeraro, G.: Content-based Recommender Systems: State of the Art and Trends. In: Recommender Systems Handbook, pp. 73–105 (2011)
16. Pazzani, M.J., Billsus, D.: Content-Based Recommendation Systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) The Adaptive Web. LNCS, vol. 4321, pp. 325–341. Springer, Heidelberg (2007)