User Semantic Model for Dependent Attributes to Enhance Collaborative Filtering

Sonia Ben Ticha[1,2], Azim Roussanaly[1], Anne Boyer[1] and Khaled Bsaïes[2]
[1] *LORIA-KIWI Team, Lorraine University, Nancy, France*
[2] *LIPA Lab, Tunis El Manar University, Tunis, Tunisia*

Keywords: Hybrid Recommender System, Latent Semantic Analysis, Rocchio Algorithm.

Abstract: Recommender system provides relevant items to users from huge catalogue. Collaborative filtering and content-based filtering are the most widely used techniques in personalized recommender systems. Collaborative filtering uses only the user-ratings data to make predictions, while content-based filtering relies on semantic information of items for recommendation. Hybrid recommendation system combines the two techniques. The aim of this work is to introduce a new approach for semantically enhanced collaborative filtering. 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 of items. For this, we design a new user semantic model by using Rocchio algorithm and we apply a latent semantic analysis to reduce the dimension of data. Applying our approach to real data, the MoviesLens 1M dataset, significant improvement can be noticed compared to usage only approach, and hybrid algorithm.

1 INTRODUCTION

Recommender Systems (RS) provide relevant items to users from a large number of choices. Several recommendations techniques exist in the literature. Among these techniques, there are those that provide personalized recommendations by defining a profile for each user. In this work, we are interested in personalized recommender systems where the user model is based on an analysis of usage. This model is usually described by a user-item ratings matrix, which is extremely sparse ($\geq 90\%$ of missing data).

Collaborative Filtering (CF) and Content-Based (CB) filtering are the most widely used techniques in 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 (Su and Khoshgoftaar, 2009). CB filtering assumes that each user operates independently and user will be recommended items similar to the ones he preferred in the past (Lops et al., 2011). The major difference between CF and CB recommender systems is that CF uses only the user-item ratings data to make predictions and recommendations, while CB relies on item content (semantic information) for recommendations. However, CF and CB techniques must face many challenges like the data sparsity problem, the scalability problem for large data with the increasing numbers of users and items.

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 process. In our approach, we design a new hybridization technique, which predicts user preferences for items based on their inferred preferences for latent item content; and presents a solution to the sparsity and scalability problems. Our system consists of two components: the first builds a new user model, *the user semantic model*, by inferring user preferences for item content; the second computes predictions and provides recommendations by using the user semantic model in a user-based CF algorithm (Resnick et al., 1994) to calculate the similarity between users. The originality of this work is in the building of the user semantic model. Indeed, assuming that items are represented by structured data in which each item is described by a same set of attributes, we build a *user semantic attribute model* for each relevant attribute. With this aim, we define two classes of attributes: *dependent* and *non dependent* and we propose a suited algorithm for each class. User semantic model is

Ben Ticha S., Roussanaly A., Boyer A. and Bsaïes K..
User Semantic Model for Dependent Attributes to Enhance Collaborative Filtering.
DOI: 10.5220/0004951102050212
In *Proceedings of the 10th International Conference on Web Information Systems and Technologies* (WEBIST-2014), pages 205-212
ISBN: 978-989-758-024-6
Copyright © 2014 SCITEPRESS (Science and Technology Publications, Lda.)

then deducted from the horizontal concatenation of all user semantic attribute model. In previous works (Ben Ticha et al., 2012; Ben Ticha et al., 2011) we have presented solutions based on machine learning algorithm to build a user semantic attribute model for non dependent attribute. In this work, we present a new approach for building a user semantic attribute model for dependent attribute by using Rocchio algorithm (Rocchio, 1971). Due to the high number of attribute values, and to reduce the expensiveness of user similarity computing, we apply a Latent Semantic Analysis (LSA)(Dumais, 2004) to reduce the size of the user semantic attribute model. We compare our results to the standards user-based CF, item-based CF and hybrid algorithms. Our approach results in an overall improvement in prediction accuracy.

The rest of paper is organized as follows: Section 2 summarizes the related work. User semantic model is described in Section 3. Section 4 describes our approach to build user semantic attribute model for non dependent attribute. Section 5 describes the recommendation component of our system. Experimental results are presented and discussed in Section 6. Finally, we conclude with a summary of our findings and some directions for future work.

2 RELATED WORK

Recommender System (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 (Resnick et al., 1994; Sarwar et al., 2001). In CF, user will be recommended items that people with similar tastes and preferences liked in the past (Adomavicius and Tuzhilin, 2005). CB is another important technique; it uses techniques developed in information filtering research (Pazzani and Billsus, 2007). CB assumes that each user operates independently and recommends items similar to the ones he preferred in the past. Hybrid approach consists on combining CF and CB techniques. The Fab System (Balabanovic and Shoham, 1997) counts among the first hybrid RS. Many systems have been developed since (Burke, 2007). Most of these hybrid systems do not distinguish between attributes and treat their values in a same way. Moreover, because of the huge number of items and users, calculating the similarity between users in CF algorithm became very expensive in time computing. Dimension reduction of data is one of the solution to reduce the expensiveness of users similarity computing. Mobasher et al. (Mobasher et al., 2003) combine values of all attributes and then apply a LSA to reduce dimension of data. Sen et al. (Sen et al., 2009) are inferring user preferences for only one attribute, the item' tags, without reducing dimension. Manzato (Manzato, 2012) computes a user semantic model for only the movie *genre* attribute and applies a Singular Value Decomposition (SVD) to reduce the dimension of data. In our approach, we compute a user semantic attribute model for each relevant attribute and we apply a suited reduction dimension algorithm for each attribute class.

3 USER SEMANTIC MODEL

In this paper, we are interested only to items described by structured data. According to the definition of Pazzani et al. (Pazzani and Billsus, 2007), in structured representation, item can be represented by a small number of attributes, and there is a known set of values that each attribute may have, for instance, the attributes of a movie can be *title*, *genre*, *actor* and *director*. In the following, we will use the term *feature* to refer to an attribute value, for instance *Documentary*, *Musical* and *Thriller* are features of *movie genre* attribute.

3.1 Dependent and Non Dependent Attribute

In structured representation, each attribute has a set of restricted features. However, the number of features can be related or not to the number of items. That is why we have defined two classes of attributes:

- **Dependent Attribute:** attribute, which having very variable number of features. This number is closely related to the number of items. So, when the number of items is increasing, the number of features is increasing also. For example: *directors* and *actors of movies*, *user tags*.
- **Non Dependent Attribute:** attribute, which having a very few variable number of features, and this number is not related to the number of items. Thus, the increasing number of items has no effect on the number of features. For example: *movie genre*, *movie origin* and *cuisine of restaurants*.

In addition, all 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 significant, in the evaluation criteria of user, than the *origin*. Experiments that we have conducted (see Section 6.2) confirmed this hypothesis. In this paper, we assume that relevant attributes will be provided by a human expert. Therefore, for each relevant attribute A, we build a *user semantic attribute model* that predicts the users preferences for its features (or group of features). This model is described by a matrix Q_A (users in lines and features (or group of features) of A in columns). In our approach, we design a suited algorithm for building the *user semantic attribute model* for each class of attribute. For non dependent attribute, due to the low number of features, we have used a clustering algorithm. Section 3.2 briefly described the operating principle of our solution that have been addressed in previous works (Ben Ticha et al., 2012; Ben Ticha et al., 2011). For dependent attribute, we have explored techniques issues from information retrieval (IR) research. Section 4 presents our solution for building the *user semantic attribute model* for dependent attribute that is the aim of this paper. The user semantic model for all relevant attributes, described by the matrix Q, is the result of the horizontal concatenation of all user semantic attribute models Q_A.

3.2 User Semantic Model for Non Dependent Attribute

Let us denote by S the set of items, U the set of users, s a given item $\in S$, u a given user $\in U$ and a rating value $r \in \{1,2,...,5\} \equiv R$. U_s the set of users that rating the item s, then we define the rating function for item s by $\delta_s : u \in U_s \longmapsto \delta_s(u) \in R$. We denote also by F_A the set of features of attribute A, f a given feature $\in F_A$ and S_f the set of items associated to feature f. For instance if we consider the *movie genre* attribute, S_{action} is the set of all action movies.

An item s is represented by its usage profile vector $s_{up} = (\delta_s(u) - \overline{\delta_u})_{(u=1..|U|)}$, where $\overline{\delta_u}$ is the average rating of all rated items by user u. The idea is to partition all items described by their usage profile in K clusters, each cluster is labeled by a feature $f \in F_A$ (or a set of features).

The number K of clusters and the initial center of each cluster is computed by the initialization step of the clustering algorithm. In initial step, each cluster C_k consists of items in $\bigcup_{f\ labeling\ C_k} S_f$ and labeled by the set of corresponding features; so its center is the mean of its items described by their usage profile vector s_{up}. Moreover, an attribute can be mono valued or multivalued depending on the number of features that can be associated to a given item s. For example, the attribute *movie genre* is multivalued because a movie can have several genres while *movie origin* is a mono valued attribute because a movie has only one origin. Thus, if an attribute is multivalued, s can belong to several clusters C_k, while for mono valued attribute, an item should belong only to one cluster. Therefore, for multivalued attribute, the clustering algorithm should provide non disjointed clusters (a fuzzy clustering), whereas, for mono valued attribute, the clustering algorithm should provide disjointed clusters.

After running the clustering algorithm, we obtain K cluster centers; each center k is described by a vector $c_k = (q_{k,u})_{(u=1..|U|)}$. The K centers is modeling k latent variables issued from the features of the attribute A. Thus, the user semantic attribute model is described by the matrix $Q_A = (q_{u,k})_{(u=1..|U|,\, k=1..K)}$.

With non dependent attribute, the number of associated features is low, this is why the clustering is suitable. Moreover, the user semantic attribute model allows an important reduction of dimension and so reduce the expensiveness of user similarity computing. In (Ben Ticha et al., 2011), we have used the Fuzzy CMean Algorithm on the movie *genre* attribute, we have obtained good performance because the user semantic attribute model has no missing values and all similarities between users were able to be computed. In (Ben Ticha et al., 2012), we have used the KMean clustering algorithm on the movie *origin* attribute. Because of the missing values in the user item rating matrix, we have proposed an algorithm for the initialization step of the KMean clustering using a movie origin ontology. We obtained good results compared to user-based CF but not as good as results for the *genre* attribute.

4 USER SEMANTIC MODEL FOR DEPENDENTS ATTRIBUTES

For a dependent attribute A, the set F_A of its features can be important and it augments with the increasing of the set of items S. In this paper, we present our solution to compute a user semantic attribute model for dependent attribute.

In addition to the formalism used in Section 3.2, we denote by F_{A_s} the set of features $f \in F_A$ associated to item s and by S_u the set of items $s \in S$ rated by user u. We define also, the rating function of user u as $\delta_u : s \in S_u \mapsto \delta_u(s) \in R$; and the Item Frequency Function for item $s \in S$ as $freq_s : f \in F_A \mapsto 1\, if\, f \in F_{A_s} (f\ associated\ to\ item\ s),\ 0\, otherwise$. The Frequency Item Matrix $F = (freq_s(f))_{s \in S\, and\, f \in F_A}$ is provided by computing $freq_s(f)$ for all items and all features.

4.1 Computing the TF-IDF on the Frequency Item Matrix F

One of the best-known measures for specifying keyword weights in Information Retrieval is the TF-IDF (Term Frequency/Inverse Document Frequency) (Salton, 1989). It is a numerical statistic, which reflects how important a word is to a document in a corpus. In our case, we replace document by item and term by feature and compute TF-IDF on the Frequency Item Matrix F.

$$FF(f,s) = \frac{freq_s(f)}{max_j freq_s(j)} \tag{1}$$

where the maximum is computed over the $freq_s(j)$ of all features in F_{A_s} of item s.

The measure of Inverse Document Frequency (IDF) is usually defined as:

$$IDF(f) = \log \frac{|S|}{|S_f|} \tag{2}$$

where $|S_f|$ is the number of items assigned to feature f (ie $freq_s(f) \neq 0$). Thus, the FF-IDF weight of feature f for item s is defined as:

$$\omega(s,f) = FF(f,s) \times IUF(f) \tag{3}$$

4.2 Rocchio Formula for User Semantic Attribute Model

Rocchio algorithm (Rocchio, 1971) is a relevance feedback procedure, which is used in information retrieval. It designed to produce improved query formulations following an initial retrieval operation. In a vector processing environment both the stored information document D and the requests for information R can be represented as t-dimensional vectors of the form $D = (d_1, d_2, ..., d_t)$ and $B = (b_1, b_2, ..., b_t)$. In each case, d_i and b_i represent the weight of term i in D and B, respectively. A typical query-document similarity measure can then be computed as the inner product between corresponding vectors.

Rocchio showed in (Rocchio, 1971), that in a retrieval environment that uses inner product computations to assess the similarity between query and document vectors, the best query leading to the retrieval of many relevant items from a collection of documents is:

$$B_{opt} = \frac{1}{|R|}\sum_R \frac{D_i}{|D_i|} - \frac{1}{|NR|}\sum_{NR} \frac{D_i}{|D_i|} \tag{4}$$

Where Di represent document vectors, and $|D_i|$ is the corresponding Euclidean vector length; R is the set of relevant documents and NR is the set of non relevant documents.

We use the Rocchio formula (4) for computing the user semantic profile of user u. In our case we replace D by S the collection of items and term by feature. Thus, the user semantic model $Q_A(u)$ for user u and attribute A is equal to Q_{opt} in formula (4). The set of relevant items R for user u is composed of all items in S having $\delta_u(s) \geq \overline{\delta_u}$. The set of non relevant items NR for user u is composed of all items in S having $\delta_u(s) < \overline{\delta_u}$.

4.3 Latent Semantic Analysis for Dimension Reduction

Latent Semantic Analysis (LSA) (Dumais, 2004) is a dimensionality reduction technique which is widely used in information retrieval. Given a term-document frequency matrix, LSA is used to decompose it into two matrices of reduced dimensions and a diagonal matrix of singular values. Each dimension in the reduced space is a latent factor representing groups of highly correlated index terms. Here, we apply this technique to create a reduced dimension space for the user semantic attribute model. In fact, for dependent attribute, the number of feature is correlated to the number of items, and so it can be very elevated and even higher than the number of items. Thus, the semantic user attribute model can have dimension greater than the user rating matrix thereby aggravating the scalability problem.

Singular Value Decomposition (SVD) is a well known technique used in LSA to perform matrix decomposition. In our case, we perform SVD on the frequency item matrix $F_{|S|\times|F_A|}$ by decomposing it into three matrices:

$$F = I_{|S|,r} * \Sigma_{r,r} * V^t_{r,|F_A|} \tag{5}$$

where I and V are two orthogonal matrices; r is the rank of matrix F, and Σ is a diagonal matrix, where its diagonal entries contain all singular values of matrix F and are stored in decreasing order. I and V matrices are the left and right singular vectors, corresponding to item and feature vectors in our case. LSA uses a truncated SVD, keeping only the k largest singular values and their associated vectors, so

$$F' = I_k * \Sigma_k * V^t_k \tag{6}$$

F' is the rank-k approximation to F, and is what LSA uses for its semantic space. The rows in I_k are the item vectors in LSA space and the rows in V, are the feature vectors in LSA space. In the resulting Frequency Item Matrix, F', each item is, thus, represented by a set of k latent variables, instead of the original features. This results in a much less sparse matrix, improving the results of users similarity computations in CF process. Furthermore, the generated latent variables represent groups of highly correlated features in the original data, thus potentially reducing the amount of noise associated with the semantic information.

In summary, for building the user semantic attribute matrix Q_A for a dependent attribute A; first, we apply a TF-IDF measure on Frequency Item Matrix F; second, we reduce the dimension of Frequency Item Matrix F by applying a LSA; third, we compute the user semantic attribute model by using Rocchio formula (4).

5 RECOMMENDATION

To compute predictions for the active user u_a, we use the user-based CF algorithm (Resnick et al., 1994). User-Based CF predicts the rating value of active user u_a on non rated item $s \in S$, it is based on the k-Nearest-Neighbors algorithm. A subset of nearest neighbors of u_a are chosen based on their similarity with him or her, and a weighted aggregate of their ratings is used to generate predictions for u_a. Equation 7 provides formula for computing predictions.

$$p(u_a, s) = \overline{\delta_{u_a}} + L \sum_{v \in V} sim(u_a, v)(\delta_v(s) - \overline{\delta_v}) \qquad (7)$$

where $L = \frac{1}{\sum_{v \in V} |sim(u_a, v)|}$ and V is the set of the nearest neighbors (most similar users) to u_a that have rated item s. V can range anywhere from 1 to the number of all users.

$$sim_(u, v) = \frac{\sum_k (q_{u,k} - \overline{q_u})(q_{v,k} - \overline{q_v})}{\sqrt{\sum_k (q_{u,k} - \overline{q_u})^2} \sqrt{\sum_k (q_{v,k} - \overline{q_v})^2}} \qquad (8)$$

The function $sim(u, v)$ provides the similarity between users u and v and is computed by using the Pearson Correlation (8). In the standard user-based CF algorithm, the users-items rating matrix $(\delta_u(s)_{(u \in U, s \in S)})$ is used to compute users' similarities. In our algorithm, for computing the similarities between users we use instead the user semantic matrix Q. As we have already mentioned, the matrix Q is the horizontal concatenation of user semantic attribute model Q_A for each relevant attribute A.

Although we apply a user-based CF for recommendation, our approach is also a model-based method because it is based on a new user model to provide ratings of active user on non rated items. Our approach resolves the scalability problem for several reasons. First, the building process of user semantic model is fully parallelizable (because the computing of user semantic attribute model is done in independent way for each other) and can be done off line. Second, this model allows a dimension reduction since the number of columns in the user semantic model is much lower than those of user item rating matrix, so, the computing of similarities between users is less expensive than in the standard user-based CF. In addition, our approach allows inferring similarity between two users even when they have any corated items because the users-semantic matrix has less missing values than user item ratings matrix. Thus, our approach provides solution to the neighbor transitivity problem emanates from the sparse nature of the underlying data sets. In this problem, users with similar preferences may not be identified as such if they haven't any items rated in common.

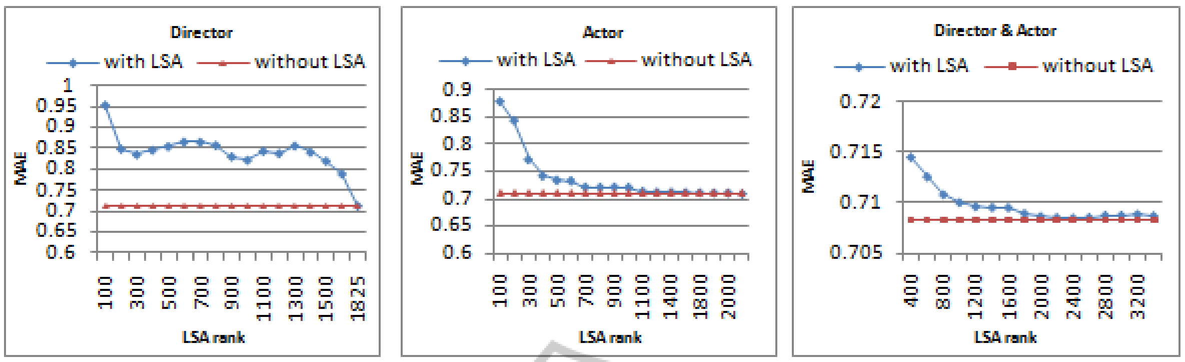
6 PERFORMANCE STUDY

In this section, we study the performance of our approach, User Semantic Collaborative Filtering (USCF in plots), against the standards CF algorithms: User-Based CF(UBCF) (Resnick et al., 1994), and Item-Based CF(IBCF) (Sarwar et al., 2001) and an hybrid algorithm. We evaluate these algorithms in terms of predictions accuracy by using the Mean Absolute Error (MAE) (Herlocker et al., 2004), which is the most widely used metric in CF research literature. It computes the average of the absolute difference between the predictions and true ratings in the test data set, lower the MAE is, better is the prediction.

We have experimented our approach on real data from the MovieLens1M dataset of the MovieLens recommender system[1]. The MovieLens1M provides the usage data set and contains 1,000,209 explicit ratings of approximately 3,900 movies made by 6,040 users. For the semantic information of items, we use the HetRec 2011 dataset (HetRec2011, 2011) that links the movies of MovieLens dataset with their corresponding web pages at Internet Movie Database (IMDb) and Rotten Tomatoes movie review systems. We use *movie genre* and *movie origin* as non dependent attributes, *movie director* and *movie actor* as dependent attributes.

We have filtered the data by maintaining only users with at least 20 ratings, and available features for all movies. After the filtering process, we obtain a data set with 6020 users, 3552 movies, 19 genres, 44 origins, 1825 directors and 4237 actors. 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% of all ratings). Thus, ratings of each user in

[1]http://www.movielens.org



(a) (b) (c)

Figure 1: Impact of LSA on prediction accuracy of Rocchio algorithm.

test set have been assigned after those of training set. It should be noted that the building of user semantic attribute model for the non dependent attributes *genre* and *origin* have been addressed respectively in previous works (Ben Ticha et al., 2011; Ben Ticha et al., 2012). Therefore, we will not detail the experiments conducted for these attributes in this paper. If it is not specified, the number of nearest neighbors is equal to 60.

6.1 Impact of LSA on Prediction Accuracy

In Figure 1, the MAE has been plotted with respect to the LSA rank. It compares the Rocchio approach *with* and *without* applying LSA (dimension reduction) on *director* attribute (Figure 1(a)), *actor* attribute (Figure 1(b)) and combined attribute *director_actor* (Figure 1(c)). In all cases, the plots have the same look, the MAE of Rocchio with LSA decreases until it reaches the MAE value of Rocchio without LSA. So, LSA dimension reduction has no effect on improving the accuracy. This can be explained by the fact that features are not highly correlated, which is understandable especially for attributes *director* and *actor*, hence their poor performance. Indeed, for the *director* attribute, for instance, the MAE without reduction (1825 features) is equal to 0.7122 while the best value with LSA is equal to 0.7884. However, for combined attributes *director_actor* (6062 features), the best value is equal to 0.7083 (obtained for Rocchio without LSA) while the worst value is equal to 0.7145 (Rocchio with LSA, rank=400). For rank equal to 1200, MAE= 0.7096, so a dimension reduction about 80% for a loss of accuracy about 0.18%. In this case, features of combined attribute, *actor_director*, are more correlated than the features of each attribute taken alone hence, the best performance. Although the LSA doesń improve the accuracy, dimension reduction is significant. Thus, it allows to reduce the cost of users similarity computing, specially when the number of features is very high, as is the case of combined attributes *director_actor*.

6.2 Impact of Attribute Class on Prediction Accuracy

Figure 2 compares algorithms for building user semantic attribute model in term of MAE. The *Average* algorithm (Average in plot) is building user semantic attribute model by computing the average of user ratings by feature ($q_{(u,f)} = AVG\{\delta_u(s)/s \in S_u \, and \, f \in F_{A_s}\}$). *Fuzzy C Mean* algorithm (FuzzyCM in plot) is a fuzzy clustering used for non dependent and multivalued attribute (here *genre*) and *KMean* algorithm (KMean in plot) is used on non dependent and mono valued attribute (here *origin*). Moreover, Rocchio algorithm (Rocchio in plot) is applied here for all attributes dependent and non dependent. For *genre*, *origin* and *director* attributes, Rocchio without LSA provides best results than with dimension reduction. For *actor* attribute, LSA with rank equal to 1100 is applied (Rocchio+LSA in plot). When analyzing this figure we note first, that *Average* algorithm provides, for all attributes, the worst performance compared to all other algorithms. Second, if we applied the *Rocchio* algorithm to non dependent attribute the performance compares unfavorably against the dependent attribute, while the best performance is attained by *FuzzyCM* algorithm on *genre* attribute and the difference is important (0.7079 for FuzzyCM and 0.7274 for Rocchio). This allows to deduct that, using a suited algorithm for each attribute class provides best performance than applying the same algorithm for all attributes. Third, the *origin* attribute has the worst performance compared to the other three attributes and this for all algorithms; this is confirm our hypothe-

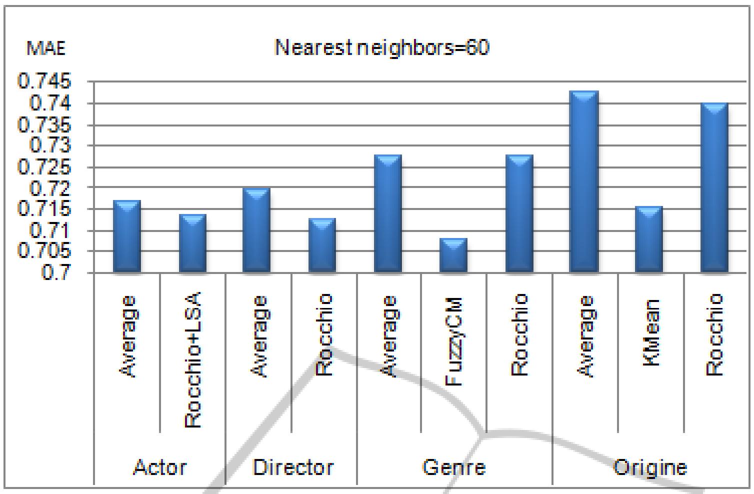


Figure 2: Impact of user semantic attribute algorithm on prediction accuracy.

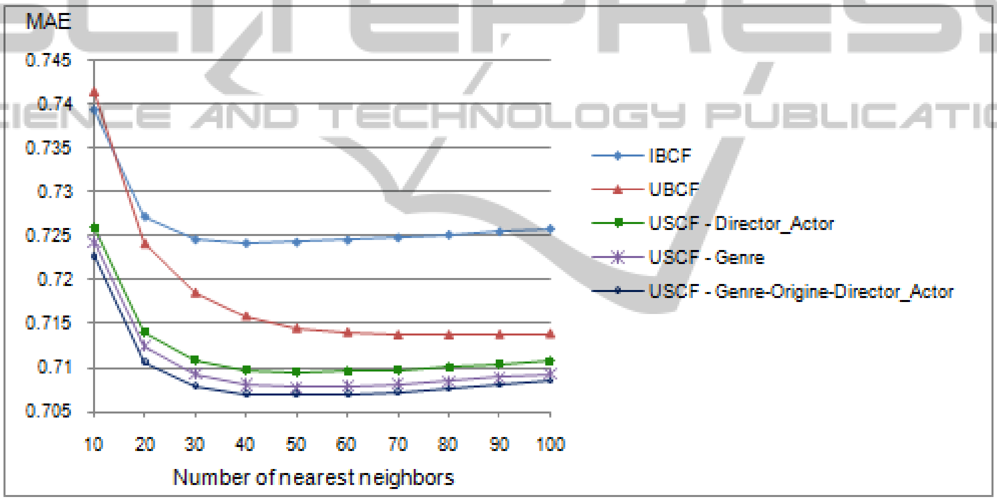


Figure 3: Evaluation of USCF against standards CF.

sis that all attributes don't have the same relevance to users. The attribute *origin* can be less significant in the choice of users than the *genre*, *actor* or *director*, which is intuitively understandable.

6.3 Comparative Results of USCF against Standard CF Systems

Figure 3 depicts the recommendation accuracy of User Semantic Collaborative Filtering (USCF) in contrast to standard Item-Based CF (IBCF) and User-Based CF (UBCF). USCF-<*Attributes*> in plot means the list of relevant attributes involved in building the user semantic model Q. For each relevant attribute, the suited algorithm is applied. So, Fuzzy CMean for *genre*, KMean for *origin*, and Rocchio with LSA (rank=1200) for combined attribute *director_actor*. Furthermore, MAE has been plotted with respect to the number of neighbors (similar users) in the k-nearest-neighbor algorithm. In all cases, the MAE converges between 60 and 70 neighbors. Our approach, USCF (in plot) results in an overall improvement in accuracy for all attributes. In addition, the best performance is achieved by the combination *genre-origin-director_actor*. This improvement can be explained by many reasons. First, taking into account the semantic profile of items in a CF recommendation process. Second, for non dependent attribute, 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 *Average* 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, users semantic model Q has few missing values, so, it allows inferring similarity between two given users even when they have any items rated in common.

7 CONCLUSION AND FUTURE WORK

The approach presented in this paper is a component of a global work, which the aim, is to semantically enhanced collaborative Filtering recommendation and to resolve the scalability problem by reducing the dimension. For this purpose, 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: *dependent* and *non dependent* attribute, and presented a suited algorithm for each class for building user semantic attribute model. The aim of this paper is to present our approach for building user semantic attribute model for dependent attribute. We have defined an algorithm based on Rocchio algorithm and have applied Latent Semantic Analysis (LSA) for dimension reduction. 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 USCF algorithm improves the prediction accuracy compared to usage only approach (UBCF and IBCF) and hybrid algorithm (Average). In addition, we have shown that applying Rocchio formula on non dependent attribute, decreases significantly the prediction accuracy compared to results obtained with machine learning algorithms. Furthermore, we have experimentally shown that all attributes don't have the same importance to users. Finally, experiments have shown that the combination of relevant attributes enhances the recommendations.

An interesting area of future work is to use machine learning techniques to infer relevant attributes. We will also study the extension of the user semantic model to non structured data in witch items are described by free text. Lastly, study how our approach can provide solution to the cold start problem in which new user has few ratings. Indeed, CF cannot provide recommendation because similarities with others users cannot be computed.

REFERENCES

Adomavicius, G. and Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng.*, 17(6):734–749.

Balabanovic, M. and Shoham, Y. (1997). Fab: Content-based, collaborative recommendation. *Commun. ACM*, 40(3):66–72.

Ben Ticha, S., Roussanaly, A., and Boyer, A. (2011). User semantic model for hybrid recommender systems. In *The 1st Int. Conf. on Social Eco-Informatics - SOTICS*, Barcelona, Espagne. IARIA.

Ben Ticha, S., Roussanaly, A., Boyer, A., and Bsaïes, K. (2012). User semantic preferences for collaborative recommendations. In *13th Int. Conf. on E-Commerce and Web Technologies - EC-Web*, pages 203–211, Vienna, Austria. Springer.

Burke, R. D. (2007). Hybrid web recommender systems. In *The Adaptive Web*, volume 4321 of *Lecture Notes in Computer Science*, pages 377–408. Springer.

Dumais, S. T. (2004). Latent semantic analysis. *Annual Review of Information Science and Technology*, 38(1):188–230.

Herlocker, J. L., Konstan, J. A., Terveen, L. G., and Riedl, J. (2004). Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5–53.

HetRec2011 (2011). In *2nd Int Workshop on Information Heterogeneity and Fusion in Recommender Systems*. The 5th ACM Conf. RecSys.

Lops, P., de Gemmis, M., and Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. In *Recommender Systems Handbook*, pages 73–105. Springer US.

Manzato, M. G. (2012). Discovering latent factors from movies genres for enhanced recommendation. In *The 6th ACM conf. on Recommender systems - RecSys*, pages 249–252, Dublin, Ireland.

Mobasher, B., Jin, X., and Zhou, Y. (2003). Semantically enhanced collaborative filtering on the web. In *1st European Web Mining Forum*, volume 3209, pages 57–76, Cavtat-Dubrovnik, Croatia.

Pazzani, M. and Billsus, D. (2007). Content-based recommendation systems. In *The Adaptive Web*, volume 4321 of *Lecture Notes in Computer Science*, pages 325–341. Springer Berlin Heidelberg.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J. (1994). Grouplens: an open architecture for collaborative filtering of netnews. In *The 1994 ACM conf. on Computer supported cooperative work*, pages 175–186, Chapel Hill, North Carolina, USA.

Rocchio, J. (1971). Relevance feedback in information retrieval. In Salton, G., editor, *The Smart Retrieval System - Experiments in Automatic Document Processing*, chapter 14, pages 313–323. Prentice-Hall, Inc.

Salton, G. (1989). *Automatic Text Processing*. Addison-Wesley.

Sarwar, B. M., Karypis, G., Konstan, J. A., and Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In *The 10 Int. WWW Conf.*, pages 285–295, Hong Kong, China.

Sen, S., Vig, J., and Riedl, J. (2009). Tagommenders: connecting users to items through tags. In *The 18th Int Conf. on WWW*, pages 671–680, Madrid, Spain.

Su, X. and Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. *Adv. Artificial Intellegence*.