Multi-Criteria Collaborative Recommender

Najma Hamzaoui, Abdelfettah Sedqui, Abdelouahid Lyhyaoui
Abdelmalek Essaâdi University
LTI Lab, ENSA of Tangier
BP: 1818, Tanger Principal
Tanger. Morocco
{nejmahamzaoui, abdelfettah.sedqui, lyhyaoui}@gmail.com

ABSTRACT: *Collaborative filtering algorithm (CF) is a personalized recommendation algorithm that is the most widely used in e-commerce. CF still needs to be improved so that it can make adequate recommendations and solve the problems such as scalability, smoothing the rating estimation. In this paper, we provide an approach of an item based collaborative filtering using item clustering prediction and including a new enhanced correlation similarity. Firstly, we cluster the items in some groups. Then, in the process of collaborative filtering recommendation, we need to calculate the similarity between the targeted item and items in the selected center. For this aim, an enhanced similarity measure based on multi criteria is proposed instead of the similarity based just on ratings' items. The objective is to consider when we calculate similarity, the integration of item rating information, the background of the item and the time-weight as criteria of the item evaluation into a convex model. In so doing, the amelioration of the similarity between items performs the recommendation. This proposed CF algorithm is showing to reduce also the influence of the former evaluation of the item.*

Keywords: Item-based Collaborative Filtering, Time Weighted, Item Information, Item Evaluation

Received: 19 June 2012, Revised 1 August 2012, Accepted 4 August 2012

1. Introduction

With the enormous information that exists in the web [1, 2, 3], the recommendation system offers users relevant information that will answer their satisfaction and their real needs. The recommendation system is a system that collects, filters, and recommends information. Automated CF is a technique to reduce information overload. CF is based on making predictions about a user's preferences or tastes based on the preferences of a group of users that are considered similar to this user. CF uses the user item ratings data to make predictions and recommendations. According to [4], CF algorithms can be grouped into memory-based and model-based approaches. The collaborative filtering algorithm is considered as the most successful and most widely used technical recommendation in the e-commerce recommendation system. It is a technique that recommends an item to a user based on his interests, while trying not to disturb him too much.

There are two types of collaborative filtering algorithms: user-based and item-based collaborative filtering [5]. Both of them use

the rating, which represents the evaluation of the user on an item, and it rarely introduces other criteria. During the two latest decades, there have been many advances in recommender systems research, an extensive review of the different approaches used in recommender systems is presented in [5,6]. CF still needs to be improved so that it can make adequate recommendations and solve the shortcomings such as sparcity, scalability, smoothing the rating estimation.

Aiming at this, we attempt in this paper to present an item based CF algorithm in which we use clustering techniques and multi-criteria to predict the user future preference for an item. When we compute the similarity between target item and it neighborhoods, we consider not only item rating information, but also the integration of the background of the item and the time-weight to make pertinent recommendation. The proposed CF algorithm improves the performance; by taking into account multi-criteria for item evaluation.

The rest of the paper is structured as follows: Section 2 presents our proposed approach in detail. Section 3 includes dataset and measurement, and finally we conclude in section 4 with planned work about further research.

2. Related Work

Automated CF is a technique to reduce information overload. According to [4], CF algorithms can be grouped into memory-based and model-based approaches. Memory-based CF (user-based or item-based) is based on the fact that users often like the items which are preferred by other users who have agreed with them in the past. Memory-based CF uses the entire user-item rating database to generate recommendations. A typical CF algorithm proceeds in three steps:

• Calculating the similarity between active user/item i and user/item j
Similarity computation is a crucial step of CF algorithms. The basic idea of similarity computation is the co-rating. For item-based CF, similarity between item i and item j is computed by working on the users who have rated both of these items. Among many methods to compute similarity, Pearson correlation and vector Cosine are the most popular and widely used.

• Neighborhood selection: select k similar users/items
We select neighbors, we find up to k items most similar to the target item. After the similarity computation, CF algorithms have to select the most similar users for the active user. This is an important step since the recommendations are generated using the ratings of neighbors, and therefore neighborhood has an impact on the recommendation quality. There are strategies for neighbors selection: Baseline strategy, Baseline strategy with overlap threshold, Similarity strategy, Combination strategy, etc. A neighborhood selection strategy is chosen depending on the similarity measures and the application domains.

• Generating prediction
In order to produce the rating prediction, we need to have the neighbors of items, and then we compute the prediction in the traditional way. In the user-based algorithms, when a subset of nearest neighbors of the active user is chosen, predictions are generated based on a weighted aggregation of their ratings. The Mostly used aggregation functions are *weighted sum and simple weighted average*.

A memory based algorithm is unsuitable in large system. To overcome the shortcomings of memory-based CF algorithms and achieve better prediction performance, model-based CF approaches have been investigated in study stream of data scalability. Model-based CF techniques, such as clustering CF algorithms, address better the scalability problem than typical CF methods, by seeking users for recommendation within smaller and highly similar clusters, instead of the entire database [7, 8, 9, 10]. Numerous approaches have been made in the literature. Each approach produces several methods. Most of these collaborative methods can be grouped to monofiltering or bifiltering approaches. In the case of monofiltering, user's rating out of item distribution [11] is one important work in this area. One other efficient method is based on building a collaborative filtering system with sparse ratings [12]. In his paper, the author introduces the concept of hierarchical clustering and recommends for new rated new user. When incomplete user's information problem has been solved by others authors introducing a fuzzy principal component analysis [13], among many others. The efficiency of collaborative recommenders is highly related also to scalability, cardinality despite of sparsity. [10] has proposed an innovative approach by introducing a smoothing based method that clusters by smoothing information data and neighborhood selection.

In the case of bifiltering approach, we consider a novel collaborative filtering that involves simultaneously the clustering both

users and items. The algorithm used of this parallel clustering is performed in real time collaborative filtering [14]. [15, 16] used also a biclustering method, but introduce a duality analysis between users and items. He proposes after an algorithm based on the neighborhood of a bicluster, innovates a new similarity measure. The most important CF challenges are the scalability, despite of sparsity. Many researchers have found that using clustering techniques is a viable way to increase scalability while maintaining good recommendations quality [7, 9, 17]. For example, when the dataset is large, it's more suitable to use the ClustKnn presented by [18, 19]. They first compress data by building an efficient clustering model. Recommendations are then generated quickly by using a simple nearest neighbor based approach, despite the fact that is intuitive; it's efficient analytically and empirically**.** A depth summery of previous work in applying clustering to analyze CF can be found in [20].

3. The Proposed Method

In this work, we propose a collaborative filtering recommendation algorithm based on clustering, collaboration techniques, and time-weighted similarity. First, we use the item clustering prediction techniques to address better the scalability problem and to achieve better prediction performance. Contrary to the traditional CF methods, the use of this approach avoids using the whole entire database but just the neighbors of the active item. In addition, we use a new enhanced correlation similarity. Traditional CF methods use only the rating to have the neighbors of the active item to compute the similarity between items. In our approach, we use the correlation similarity that integrates not only the item rating, but also the information of item [21, 22, 23, 24]. Combining these two similarities based on a convex model includes also time-weight to reduce the influence of the former evaluation of the item, enhance the recommendation process and improve predictions.

We will now give a detailed description of how the approach can be applied. We begin by introducing the used data structure, then, a quick overview of item clustering prediction process step of a step manner, enhancing its predictions by using a new formulation of similarity for the case of item-based filtering.

3.1 Data Structure

In this section, we provide details about data structure for the proposed CF algorithm.

3.1.1 Item Assessment Data

The data representation for the traditional collaborative filtering specially for item-based CF is based on the construction of an $N \times M$ item-user matrix U, showed in I. $R_{i,j}$ in the ith row and jth column of the matrix means the rating value for item i of user j.

Items from 1 to 1682, users are enumerated from 1 to 943, while ratings item of the users to an item take values between 1 and 5.

	u_1	u_2	…	u_M
I_1	R_{11}	R_{12}	…	R_{1M}
I_2	R_{21}	R_{22}	…	R_{2M}
⋮	⋮	⋮	⋮	⋮
I_N	R_{N1}	R_{N2}	…	R_{NM}

Table 1. Item-User Rating Matrix

3.1.2 Attributes of the item

Except for ratings awarded by users on items, The items, which in the case of the MovieLens data set correspond to movies .The item descriptions reflect the interests of a user when a user reads or downloads items [21,22]. Item background can be title, category, subject, authors, and time published. MovieLens data set includes information regarding the items specifically; there is list, with 1682 item attributes vectors of the following form:

movie id | movie title | release date | video release date |
IMDb URL | unknown | Action | Adventure | Animation |
Children's | Comedy | Crime | Documentary | Drama |
Fantasy | Film-Noir | Horror | Musical | Mystery |Romance | Sci-Fi | Thriller | War | Western.

The movie ids are the ones used in the main data set. The movie title is a string with the title of the movie. The release dates are

of the form dd-mmm-yyyy, e.g. 14-Jan-1967. The IMDb URL is a web link leading to the Internet Movie Database page of the corresponding movie. The last 19 fields are the film genres. Items can belong to more than one genre in the same time.

3.1.3 Item Assessment Time Data

Table 2 illustrates the data used in our algorithm .in addition of the construction of an n × m user-item matrix U, showed in table II. $T_{i,j}$ in the *ith* row and j^{th} column of the matrix means the time evaluation value for item i of user j. We have

	u_1	u_2	…	u_M
I_1	T_{11}	T_{12}	…	T_{1M}
I_2	T_{21}	T_{22}	…	T_{2M}
⋮	⋮	⋮	⋮	⋮
I_N	T_{N1}	T_{N2}	…	T_{NM}

Table 2. Item-User Time Evaluation Matrix

Where $T_{i,j}$ the time stamps are unix seconds since 1/1/1970 UTC.

3.2 Item clustering prediction process

3.2.1 Item clustering

In this step, we choose the famous k means algorithm as the basic clustering algorithm, knowing that there are many others algorithms that can be used to create item clustering. K-means is used in many previous researches. The algorithm takes randomly the first k items as the centers of k unique clusters [10]. Each of the remaining items is then compared to the closest center and we set the items to their nearest cluster and recalculate the cluster center. The cluster centers in the following passes are re-computed based on cluster centers formed in the previous pass and the cluster membership is re-evaluated. We repeat the process until the centers change in a small region. For giving the desired number of clusters, a number k is considered as an input to the algorithm.

Though, we should predefine the cluster number, the cluster sizes are similar. The implement is simple and the performance of k-means is well but k is difficult to choose from one domain to another as we know.

Table 3. illustrates the process of Collaborative filtering recommendation algorithm based on the item clustering.

	i_1	i_2	…	i_n
u_1	R_{11}	R_{12}		R_{1n}
u_2	R_{21}	R_{22}		R_{2n}
…				
u_m	R_{m1}			R_{mn}

Iterm clustering

	c_1	c_2	…	c_n
u_1	R_{11}	R_{12}		R_{1n}
u_2	R_{21}	R_{22}		R_{2n}
…				
u_m	R_{m1}			R_{mn}

Table 3. CF- based on item clustering

3.2.2 Selecting clustering centers

In order to compute the rating prediction $P_{u,i}$, for the target (user, item) pair (u, i), the following steps are taken:

We select clustering centers. When we cluster the items, we get the items centers. This center is represented as an average rating over all items in the cluster. So we can choose the target item neighbors in some of the item center clustering, unlike to the Traditional memory based collaborative filtering that consist on searching the whole ratings database and it suffers from poor scalability when the number of users and items in ratings database enhance. Before searching neighbors of the target item which is considered as an essential step of item based collaborative filtering algorithm, we use the Pearson's correlation to the similarity between the target item and the items centers and we choose the best similar one of the active item.

3.2.3 Selecting neighbors

We select neighbors, we find up to k items most similar to the target item. We need to select the Top most similar k items. Normally we can use the Pearson's correlation to calculate the similarity between the two vectors of ratings. Equation 1 measures the linear correlation between two vectors of ratings as follows:

$$\text{sim}(i, j) = \frac{\sum_{u=1}^{M} R_{u,i} R_{u,j}}{\sqrt{\sum_{u=1}^{M} R_{u,i}^{2} \sum_{u=1}^{M} R_{u,j}^{2}}} \quad (1)$$

Where $R_{u,i}$ is the rating of the item i by user u , $R_{u,j}$ is the rating of the remaining item j by u, and m the number of all rating users to item i and item j.

Our basic idea is to smooth rating estimation. By this way, we present an item-based collaborative filtering using a novel enhanced similarity measure based on multi-criteria. The objective is to consider the correlation similarity that integrates the item rating in the addition to the information of item in order to enhance the similarity. So, to take into account the advantage of these two similarities, we propose a convex combination of the two measures. The weighting of the combination is a function which parameter is a time weight. In so doing, we can reduce the influence of the former evaluation of the item. This enhances the recommendation process and improves predictions. With a lot of information that exists in the web, users tend to choose the topical information. So, we integrate time evaluation into a time function in the model in order to penalize the item assessment that has been done a long time ago. By this way, the smaller impact of historical data. Otherwise, the better importance for the sooner item evaluation. The exponential time function is widely used in many applications in which it is desirable to gradually decay the history of past behavior as time goes [25]. Authors in [26], use time weight for collaborative filtering just with the rating of the item on prediction phase. They modified the common rating prediction computation incorporating a time weighting factor. Our method is to use the time weight relatively with enhanced similarity on prediction stage. We use time weighting function as follows:

$$\lambda = e^{-\alpha . \Delta T} \quad (2)$$

where $\alpha = 1 / T_0$ is a parameter to control the user specific decay rate and should be learnt from data.

The exponential function satisfies our needs well. From equation 1, the value of the time function is in the range (0,1), and it reduces with time. We emphase the user's latest purchase interest and we focus on the most recent data and we analyze the half-life parameter T_0 to define the rate decay of the weight assigned to each data.

We can observe that T_0 is inversely proportional to α. The value of parameter decides the decay rate of old data. Changes in user purchase interest are taken into consideration. So, if the user preference for the type of items changes frequently, we should assign a low value to T_0.The lower value of T_0, the higher value of α. The higher value of α, the faster old data decays and the lower the importance of the historical information compared to more recent data. If the user preference for the type of items is consistent, we should decay the old data slowly and assign a high value to T_0.

The users purchase habits vary; it's not feasible providing an appropriating value of parameter T_0 for each user and each item. We assume that the same user has the similar references and similar purchase interest changes for the similar items. It means to one user, the similar items have the similar decay rates of old data. So, we propose to compute the corresponding values of

parameter T_0 for each user and each cluster of items according to the user personalized purchase history. To realize the goal, we use simple clustering approach to summarize items into different clusters through the rating information [25,26]. There exists strong relevancy between these items belonging to the same cluster. Then to each item cluster, we take the leave-one-out approach to compute automatically the personalized value of parameter T_0 in [26].

The advantage of our strategy is that in addition to using the item's rating for item based filtering, we can enrich the process of recommendation by other criteria, for example by taking into account the demographic information of items that are similar to the target item. Items' information can be presented by their attributes, such as items' gender, release date, etc. For this, we have to use the enhanced similarity measure which combines items' ratings with these attributes to get better correlation and find these items. There are methods which show the importance of demographic information, including the contributions from both ratings and demographic correlations [21, 22, 23, 24]. [23] proposes a hybrid method that groups items by integrating the item rating similarity and item attribute information similarity. The relative weighting is adopted to adjust the importance of rating similarity and attribute similarity. λ and 1- λrepresent the relative importance of the item rating similarity and item attribute similarity, respectively. If $\lambda = 0$, then the method becomes item Information based method. If $\lambda = 1$, then the method becomes traditional CF method. But our approach uses λ in the combination as a time weighting function, noted by (1), as a weighted parameter that is used to adjust the similarity based on item assessment and item attributes to predict user future preference automatically as described in (3):

$$\text{Improve}_{\text{sim}(i,j)} = \lambda \text{sim}_{\text{rating}}(i,j) + (1-\lambda)\text{sim}_{\text{Attribute}}(i,j) \quad (3)$$

Where $\text{sim}_{\text{rating}}(i,j)$ denotes the similarity of ratings between item i and j. $\text{sim}_{\text{Attribute}}(i,j)$ denotes the similarity of information attributes between items i and j.

Evidently, $\text{sim}_{\text{rating}}$ plays a principal role in (3) and in the basic algorithm for item-based filtering. However, when the number of users whose co-rated items i and j is too less, $\text{sim}_{\text{rating}}(i,j)$ is not accurate. So, including the enhancing correlation by demographic information lead to accurate the similarity. In addition to the reduction of the influence of the former evaluation of the item, we use λ time weighting function, as a weighted parameter that is used to adjust the similarity based on item assessment, and item attributes to predict user future preference automatically. We favor neighbors of the active item, those which are evaluated recently; so, computing similarity of the target item with neighbor's item is relatively linked to this time weighted function.

The similarity of an item that was rated recently by a user with others items based on their rating should have a bigger impact on the prediction of future user behavior. Otherwise, the similarity based on the attribute of the item plays a more important role of an item that was rated a long time ago than time weight because the user preference for the type of items consistent.

Changes in user purchase interest are also taken into consideration; we compute the time weights for different items in a manner that will assign a decreasing weight to old data.

After the similarity between i and any item that can be got with this similarity measure, there is a nearest neighbors set most similar to the active item i, named nearest item neighbor set S(i). Now, we can produce the recommendations.

3.2.4 Producing recommendations

In order to produce the rating prediction $P_{u,i}$, we need to have the neighbors of items, and then we compute the prediction in the traditional way using the following equation:

$$P_{u,i} = \bar{R}_u + \frac{\sum_{j \in S(i)} (R_{u,j} - \bar{R}_j)\, \text{sim}_{\text{rating}}(i,j)}{\sum_{j \in S(i)} \text{sim}_{\text{rating}}(i,j)} \quad (4)$$

We modify the prediction generation formula as follows:

$$P_{u,i} = \bar{R}_u + \frac{\sum_{j \in S(i)} \text{Improve}_{\text{sim}}(i,j)\, (R_{u,j} - \bar{R}_j)}{\sum_{j \in S(i)} \text{Improve}_{\text{rating}}(i,j)} \quad (5)$$

$\overline{R}_u$ is the average ratings of the target user u to the items, $R_{u,j}$ is the rating of the target user u to the neighbor item j, $\overline{R}_j$ is the average ratings of the remaining item j for users.

Obviously, at this point prediction generation formula depends on the use of the improved correlation with time weight, instead of the ratings-based correlation only. The enhanced correlation, as defined in (4), includes the contributions from both ratings-based and demographic correlations into a model convex with a weighting factor of time function.

4. Data and Measurement

4.1 DataSet

In order to measure the prediction algorithm performance and to compare the results of different neighborhood based prediction algorithms. We use Dataset that comes from the Movielens dataset at Grouplens Research Project in Minnesota University. MovieLens is a movie recommending application whereby users initially rate a subset of movies that they have already seen [27]. (http://MovieLens.umn.edu/) captures the user's rating for the movies and provides a list for movie recommendation. The historical dataset consists of 100,000 ratings from 943 users on 1682 movies with every user having at least 20 ratings and simple demographic information for the users is included. The site now has over 45000 users who have expressed opinions on 6600 different movies. The ratings are on a numeric five-point scale with 1 and 2 representing negative ratings, 4 and 5 representing positive ratings, and 3 indicating ambivalence. We need to extract the demographic information (gender of the movie for example) of the item existing in the MovieLens data to construct demographic vectors of the 19 features for the item.

4.2 Evaluation Metric

Many metrics have been proposed to measure prediction engine performance such as correlation between ratings and prediction, root mean squared error, etc, providing the same conclusions, we present only the Mean Absolute Errors (MAE) that is a measure of deviation of predicting ratings of recommendations from their true user specified values. Assuming that the actual user-specified value set is $\{q_1, \ldots, q_n\}$, and the predicting rating set by recommending algorithm is $\{p_1, p_2, \ldots, p\}$. Formally,

$$\text{MAE} = \frac{\sum_{i=1}^{n} |p_i - q_i|}{n}$$

Where n represents the total number of predictions computed for all users. According to this metric, a better CF algorithm has a lower MAE. Other similar metrics such as Root Mean Squared Error (RMSE) and ROC sensitivity are sometimes used for CF diagnostic power and evaluation as well.

4.3 Exprimentation

We can observe from equation 1 that T_0 is inversely proportional to α.The lower value of T_0, the higher value of α. The higher value of α, the faster old data decays and the lower the importance of the historical information compared to more recent data. The value of parameter T_0 decides the decay rate of old data.

The results of using different constant of parameter T_0 are demonstrated in figure I. for example, we vary the parameter from 50, 100, and 200. The parameter T_0 controls the decay rate of historical information. We have to assign different values to the parameter to obtain the best performance.

We also compare the accuracy of item based CF algorithm based on clustering method kmeans with different number of clusters c. We explore the number of clusters c, c is set to 3, 4 and 5, we choose the best one with cross validation. As shown in table 4, given the MAE to measure the prediction algorithm performance with the variation of the number of the neighborhoods [10,15,20,25,30,35,40]. We conduct experiments to find suitable parameter c for clustering that achieve the best results.

In figure 2 as observed, we use the MAE to measure the prediction algorithm performance. In order to verify the effect of ours algorithms CF with enhanced similarity (CFABES) and clustering algorithm based on enhanced similarity (CFABBESWC), the accuracy of those algorithms are compared with others existing algorithms; classic item based algorithm, clustering algorithm, time weight algorithm in literature [26], we observe in II that our CF algorithm including enhanced similarity based on clustering method is better and we find that the variation of the number of the neighborhoods has a significant effect on the prediction quality.

	Clustering with kmeans for Collaborative Filtering						
	10	15	20	25	30	35	40
c = 2	0.1864	0.1766	0.1710	0.1668	0.1646	0.1624	0.1652
c = 3	0.1896	0.2852	0.1994	0.1651	0.1616	0.1595	0.1578
c = 4	0.1843	0.1736	0.1674	0.1637	0.1631	0.1577	0.1547
c = 5	0.1841	0.1736	0.1712	0.1674	0.1596	0.1605	0.1541

Table 4. MAE of Item based Collaborative Filtering with different c

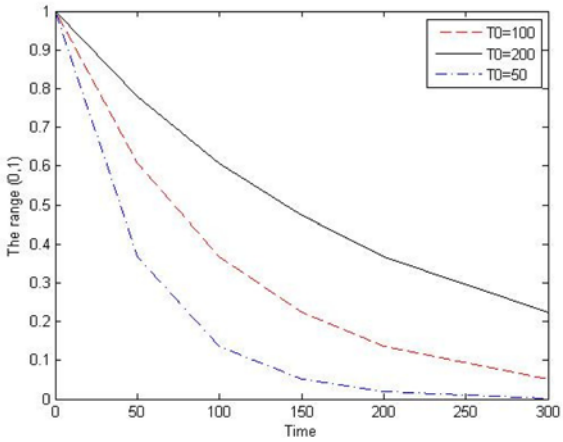

Figure 1. The Curves of Time Function Using Different T_0

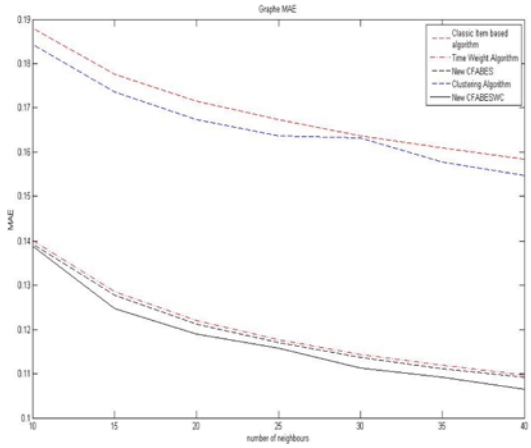

Figure 2. Comparaison of Recommended Precision

MAE is fluctuated with the number of items. It's clear that the proposed algorithm has better performance with increasing of the number of similar items. This is because the most similar items of target items distributes within the clusters that have the highest similarity with target item. It is enough to search during the clusters that similar with target item mostly. So the recommended precision is improved. Experimental results show that is has perfect effect both on recommend speed and recommend precision because of taking into account the clustering and multicriteria.

5. Conclusion

CF algorithm has been shown to be the most successful for recommendations. This paper attempts to address the improvement needs in CF to smooth prediction for truly accurate recommender systems. By this way, a new approach of the item-based collaborative filtering was proposed. First, we applied the clustering technology. We have used the item clustering prediction techniques to address better the scalability problem to avoid higher order computational complexity. Second, a new similarity measure based on multi-criteria was included to achieve better prediction performance. This has a big impact on the prediction

of target user to an item and can enhance recommendations. We consider this study as a start of a line of work that can be realized in the future to spotlight the use of the item multi-criteria with the clustering techniques in the recommendation process.

References

[1] Janner, T., Schroth, C. (2007). Web 2.0 and SOA: converging concepts enabling the internet of services, *In*: IT Professional, p. 36–41.

[2] Knights, M. (2007). Web 2.0, *IET Communications Engineer*, p. 30–35.

[3] Lin, K. J. (2007). Building Web 2.0, Computer , p. 101–102.

[4] Breese, J. S., Heckerman, D., Kadie, C. (1998). Empirical Analysis of Predictive Algorithms for Collaborative Flitering, *In*: Proc.14 th Conf. Uncertainty in Artificial Intelligence, July.

[5] R. Burke, Hybrid recommender systems: survey and experiments, User Model, *User-Adapted Interaction* 12 (4) (2002) 331–370.

[6] Adomavicius, G., Tuzhilin, A. (2005). Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions, *IEEE Trans Knowledge Data Eng*. 17 (6) 734–749. Chee, S. H. S., Han, J., Wang, K. (2001). RecTree: an efficient collaborative filtering method, *In*: Proceedings of the 3rd International Conference on Data Warehousing and Knowledge Discovery, p. 141–151.

[7] O'Connor, M., Herlocker, J. (1999). Clustering items for collaborative filtering, *In*: Proceedings of the ACM SIGIR Workshop on Recommender Systems (SIGIR '99).

[8] Sarwar, B. M., Karypis, G., Konstan, J. A., Riedl, J. (2002). Recommender systems for large-scale E-commerce: scalable neighborhood formation using clustering, *In*: Proceedings of the 5th International Conference on Computer and Information Technology (ICCIT '02), December.

[9] Xue, G.-R., Lin, C., Yang, Q. et al. (2005). Scalable collaborative filtering using cluster-based smoothing, *In*: Proceedings of the ACM SIGIR Conference, p. 114–121, Salvador, Brazil.

[10] Lee, W. S. (2000). Online clustering for collaborative filtering. School of Computing Technical Report TRA8/00.

[11] Kohrs, Merialdo, B. (1999). Clustering for Collaborative Filtering Applications. *In*: Proceedings of CIMCA'99. IOS Press.

[12] Honda, K., Sugiura, N., Ichihashi, H., Araki, S. (2001). Collaborative Filtering Using Principal Component Analysis and Fuzzy Clustering, *Lecture Notes in Computer Science*.

[13] Manolis Vozalis, Konstantinos, Margaritis, G. (2004). Enhancing Collaborative Filtering with Demographic Data:The Case of Item-based Filtering, *In*: Proc. of AIAI, p. 393-402.

[14] George, T., Merugu, S. (2005). A scalable collaborative filteringframework based on co-clustering. *In*: Proceedings of the IEEE ICDM Conference.

[15] Panagiotis Symeonidis, Alexandros Nanopoulos, Apostolos Papadopoulos, Yannis Manolopoulos, (2006). Nearest-Biclusters Collaborative Filtering,WEBKDD.

[16] Panagiotis Symeonidis, Alexandros Nanopoulos, Apostolos N. Papadopoulos

[17] Yannis Manolopoulos, (2007). Nearest-biclusters collaborative filtering based on constantand coherent values. *Information Retrieval*.

[18] Kelleher, J., Bridge, D. (2003). Rectree centroid: An accurate, scalable collaborative recommender. *In*: Procs. of the Fourteenth Irish *conference on artificial Intelligence* and Cognitive Science, p. 89-94.

[19] Rashid, A. M., Lam, S.K., Karypis, G., Riedl, J. (2006). ClustKNN: A HighlyScalable Hybrid Model- & Memory-Based CF Algorithm. WEBKDD.

[20] Rashid, A. M., Lam, S. K., LaPitz, A., Karypis, G., Riedl, J. (2007). Towards a Scalable kNN CF Algorithm:Exploring Effective Applications of Clustering. LNCS vol 4811/2007: Advances in Web Mining and Web Usage Analysis, p. 147-166, Springer.

[21] RuLong Zhu, SongJie Gong. (2009). Analyzing of Collaborative Filtering Using Clustering Technology, *In*: Procs of ISECS International Colloquium on Computing, Communication, Control, and Management.

[22] YaE Dai, SongJie Gong, Personalized Recommendation Algorithm using User Demography Information, WKDD2009, *IEEE Computer Society Press*.

[23] SongJie Gong, XiaoYan Shi, A Collaborative Recommender Combining Item Rating Similarity and Item Attribute Similarity, ISBIM 2008, *IEEE Computer Society Press*.

[24] SongJie Gong, A Collaborative Recommender Based on User,Information and Item Information, *In*: Proceedings of the 2009, *International Symposium on Information Processing*, p. 001-004

[25] Long Yun, Yan Yang, Jing wang, Ge Zhu. (2011). Improving Rating Estimation in recommender Using Demographic data and expert Opinions, Software Engineering and Service Science (ICSESS), IEEE

[26] Aggarwal, C. C., Han, J., Wang, J., Yu, P. S. (2003). A conference on Research and development in informaion retrieval, p. 259-266, Toronto,Canada.

[27] Ding, Y., Xue Li (2005). Time Weight Collaborative Filtering, *In*: Proceedings of the 14th ACM International Conference on Information and Knowledge Management, p. 485 – 492

[28] Miller, B., Albert, I., Lam, S., Konstan, J., Rield, J. (2002). Movielens unplugged: experiences with an occasionally connected recommender systems, *In*: Proc. Internat. Conf. Intelligent User Interfaces, p. 263–266.