RESEARCH ARTICLE OPEN ACCESS

An Improved Collaborative Filtering Algorithm Based on Tags and User Ratings

CaiyunGuo[1], HuijinWang[2]

[1,2](College of Information Science and Technology, Jinan University, Guang dong, China)

--************************************----------------------------------

Abstract:

Aiming at the problem that the existing social tags recommendation system in building user interest model does not fully reflect the genuine interests, this paper proposes aim proved recommended algorithm (TARBCF) based on tags and user ratings. Since the rating data often sparse, to make the best use of the both advantages of ratings and tags, a rating predicts algorithm based on item category is introduced to predict the ratings. In this paper, user's ratings can be incorporated to calculate the weight of tags. Considering the user interest has the time characteristic, time window is used to capture the current interests of user. Thus, by analyzing the traditional collaborative filtering thought, considering the relationship between user ratings and tags as well as the influence of user's current interest, this paper set up an user-tag correlation matrix, which can calculate the target user's nearest neighbors. Then according to the neighbor users predict the target user's preferences of candidate items. Finally, taking the top-N scores items recommend to the target user. Simulation experimental results show that the improved algorithm can better reflect the user's preferences, and the quality of its recommendations were superior to the traditional scheme.

Keywords **—Collaborative filtering algorithm, Tags, Time window, User ratings**

--************************************----------------------------------

I. INTRODUCTION

With the huge amount of information is shared through network platforms, it is becoming more difficult for people to find the useful information in timely. Fortunately, Recommender System as an effective tool to solve the problem of "information overload" [1], which has been widely applied to various types of network platforms and e-commerce platforms(e.g., Amazon, eBay, and Taobao) [2].It is changing the way of people find information, which take the initiative to provide valuable information for the user. At present, how to build a good personalized recommendation system is still the focus of researchers, although there are a lot of works has been done in recent years.

Obviously, recommendation algorithm as the core part of personalized recommendation system, its accuracy closely related to the quality of recommendation system. Over the years, various approaches for building recommender systems have been developed that utilize either demographic, content, or historical information [3].Among them, collaborative filtering (CF) is the most successful technique in recommendation systems[4][5],it is a kind of recommendation algorithms that based on user's behavior data .The basic idea of CF is that if users have the similar behaviors in some items, they will rate or act on other items similarly, so it can through the neighbor users to decide whether to recommend the items to the target user.Compared with Content-based Recommendations (CB),CF is more suitable for recommend non-text items, such as music, movies, pictures, etc., that is also one of the reasons why it has been widely used.However, with the network and user scale expands unceasingly, item quantity increased dramatically, sparse matrix and cold start problem become more and more serious, the recommend quality of the collaborative filtering was greatly reduced.

To alleviate the impact of data sparsity in collaborative recommendation system, a series of improved methods have been put forward by many researchers. An rating prediction scheme is applied tothe typical CF algorithms, which is confirmed effectively to alleviate the data sparsity[6]. Singular value decomposition (SVD) model also be proposed to reduce the space dimension of matrix.This method can significantly improve the system expansion ability, but the dimension reduction will lead to information loss [7]. Considering the user's interest would be migrated over time, More and more researchers take time factor into consideration,which make the similarity calculation more accurate and a better recommendation result have been achieved [8].

These methods above are based on the rating data to improve the quality of recommendations. With the rapid development of the web2.0, social tag is widely applied to the recommendation system, users can choose or mark tags according to their own understanding and preferences, which contains a lot of user interests information. Thus, taking tags into recommendation algorithm can help us to improve the quality of the recommendations.There are several studies that exploit various aspects of tags to build user interest modeling.Au Yeung,Cibbins, and Shadbolt constructed a user's model which can represent multiple interests of the user by forming a set of frequent tag patterns [9]. Nakamoto al.proposed Reasonable Tag-based CF(RCF) that first clusters tags into topics by using an expectation-maximization (EM) algorithm[10].More recently, Wang, Clements, and Reindersintroduced a collaborative tagging model, a collaborative browsing model, and a collaborative item search model into building personalization models [11]. Thus, the usage of tags allows us to capture valuable information for understanding user interests and can build better interest models.

Thus, on the basis of these studies, inspired by the idea of combining rating information with tags, an improved collaborative filtering algorithm based on tags and user ratings is proposed in this paper, which is improved through two ways: One is improving user similarity calculating method, and the other is catching the target user interest in recommendation generation phase.

II. DEFINITIONS

To better describe the proposed method in this paper, we define some key concepts and used in this paper as below:

- Users: $U = \{u_1, u_2, \dots, u_{|U|}\}$ contains all users who have used tags or rating to evaluate items.
- Items(i.e.,Resources): $I = \{i_1, i_2, \dots, i_{|I|}\}$ contains all items evaluated by users in U. It could be any type of resource or products that come from our daily life, such as videos, music, movies etc.
- Tags: $T = \{t_1, t_2, \dots, t_{|T|}\}$ contains all tags used by users in U. A tag is a piece of textural information and can be used by users to label multiple items.
- Item Category: $C = C_1 \cup C_2 \cup \cdots \cup C_k$ contains all item category information in the system. An item may belong to several categories,

where $C_j = \{I_{j,1}, I_{j,2}, \cdots, I_{j,k}\}$ represents the item set that belong to the j-th category.

- User-Item Rating Matrix: Described as a $|U| \times |I|$ ratings matrix $R_{|U|\times|I|} = (R_{i,j})_{|U|\times|I|}$. The row represents $|U|$ users and column represents $|I|$ items. The element of matrix $R_{i,j}$ means the rating rated to the user i on the item j, which is acquired with the rate of user's interest. Usually, the ratings is on a 1-5 scale or unknown.

III. THE PROPOSED APPROACHES

As we all know, tags are popularly used in various kinds of application areas, it is becoming another important implicit rating information used to profile users' interests. However, the tags used by users are free-formed and contain semantic ambiguities and tag synonyms [12].

In this paper, we proposed a new method to integrate tags and ratings to improve the accuracy of predictions based on the traditional CF. Different from the earlier work, we focus on what a tagged item is about and how much a user prefers the item, rather than capturing what tags are used by the user.

Since the explicit ratings are rare or not available in real life, we should use the rating predict algorithm to predict the rating of no rating items. Then, using those ratings and tags information to build User-Tag Correlation Matrix, which can integrate the advantages of ratings and tags. Finally, according the neighbors of target user, we can make recommendation by using the similarity information of items. The specific steps are as follows:

A. *Rating Predict Algorithm Based on Item Category*

First, classifying the items by different categories, then using the Item-based collaborative filtering algorithm (IBCF) to predict the rating of the no rating items in every category. Since an item may belong to several categories, we should compute the average rating finally. The basic steps of rating predict algorithm are as follows:

Inputs: User-Itemrating matrix R, Item Category

Outputs: The predicted ratings for no rating items

Step1: According to the Item Category C, the User-Itemrating matrix R can be divided into k part, that is: $R_{|U|\times|I|} = R_1 \cup R_2 \cup \cdots \cup R_k$. R_j represents the j-th rating matrix.

Step2: On the basis of the above, using the Cosine-based similarity to compute the similarity between items i and j like this formula.

$$sim(i,j) = \cos(i,j) = \frac{\sum_{c=1}^{|U|} R_{c,i} \cdot R_{c,j}}{\sqrt{\sum_{c=1}^{|U|} R_{c,i}^2}\sqrt{\sum_{c=1}^{|U|} R_{c,j}^2}} \quad (1)$$

where $R_{c,i}$, $R_{c,j}$ represents the rating of the user c for the item i and j ,respectively.

Step3: Assuming that j is an item has no rating of user u, in order to predict the rating of its, we should generate a neighbor items set $M_j = \{I_1, I_2, \cdots, I_v\}$, $j \notin M_j$ from Step 2by descending order.

Step4: Using the items in the nearest neighbor set M_j and the rating values in user-item rating matrix R, to predict the ratings of the user u for item j, denoted by $P_{u,j}$ is give by

$$P_{u,j} = \frac{\sum_{i\in M_j} sim(i,j) \times R_{u,i}}{\sum_{i\in M_j} |\, sim(i,j)\,|} \quad (2)$$

where $R_{u,i}$ presents the rating of the user u for the item i in M_j.

Step5: If item j belong to multiple categories, repeat Step 2to Step 4, then taking the average ratings into $P_{u,j}$ and save it to the rating matrix R.

B. User-Item Rating Matrix Expanded by Prediction Rating

Based on the discussed in above, we can build a new rating matrix $R'_{|U|\times|I|} = (r_{u,j})_{|U|\times|I|}$ by using the prediction ratings like that.

$$r_{u,j} = \begin{cases} R_{u,j} & if\ User\ u\ Rated\ Item\ j \\ P_{u,j} & if\ User\ u\ Not\ Rated\ Item\ j \end{cases}$$

C. Build User-Tag Correlation Matrix

Before going into further detail, the another notation and definitions required for understanding our approach are introduced as follows.

1) Positive and Negative Items

In general,ratings of a user for items can reflect the user's interest more accurately.The rating scale is fixed as numerical values (e.g., a scale of 1-5). Since each user would have his/her own rating behavior, we can classify the items into two parts: a set of positive items and a set of negative items [13].

In this paper, we use the ratings of R' to form the set of positive and negative items for user u ,which are defined as $Pos(u)$ and $Neg(u)$, respectively such that :

$$Pos(u) = \{i \in I \mid r_{u,i} \geq \overline{r_u}\},$$

$$Neg(u) = \{i \in I \mid r_{u,i} < \overline{r_u}\},$$

in which $\overline{r_u}$ represents the average rating of user u .

2) Calculating weights of tags

In our study, we associate a weight of tags with a user's rating, rather than the well-known $tf-idf$ weight.The weight of tag t for user u can be measured by the related items rating of user u . Formally, the weight of tag t annotated in item i for user u , can denoted as $w_{u,i}(t)$, is computed by:

$$w_{u,i}(t) = \frac{r_{u,i}}{\sqrt{\sum_{j=1}^{|I|} r_{u,j}^2}} \qquad (3)$$

Because a tag may appear in several items with different weights, we compute the mean weight of the tag in the set of positive items and negative items, respectively:

$$\omega_{u,t}^{pos} = \frac{1}{|I_u^{pos}(t)|} \times \sum_{j \in I_u^{pos}(t)} w_{u,j}(t) \qquad (4)$$

$$\omega_{u,t}^{neg} = \frac{1}{|I_u^{neg}(t)|} \times \sum_{j \in I_u^{neg}(t)} w_{u,j}(t) \qquad (5)$$

where $I_u^{pos}(t)$ and $I_u^{neg}(t)$ are respectively the set of positive and negative items rated by user u containing tag t. Finally, the weight of tag t for user u, denoted as $\omega_{u,t}$, can be illustrated by the following formula .

$$\omega_{u,t} = \begin{cases} (\omega_{u,t}^{pos} + \omega_{u,t}^{neg})/2, & if\ t \in I_u^{pos}(t), t \in I_u^{neg}(t) \\ \omega_{u,t}^{pos}, & if\ t \in I_u^{pos}(t), t \notin I_u^{neg}(t) \\ \omega_{u,t}^{neg}, & if\ t \in I_u^{neg}(t), t \notin I_u^{pos}(t) \end{cases} \qquad (6)$$

3) User-Tag Correlation Matrix

Based on the mentioned in above, let $UT_{|U|\times|T|} = (\omega_{u,t})_{|U|\times|T|}$ be a user-tag correlation matrix, in which $\omega_{u,t}$ represents the weight of tag t for user u , that can been computed in the equation (6).

D. Neighborhood Forming

Neighborhood forming is to generate a set of like-minded peers for a target user $u_i \in U$ or a set of similar peer items for an item $p_i \in P$ [12]. In our study, we identify the best neighbors based on the weights for tags, that is why we build the user-tag correlation matrix UT. Differing from the previous work, rating information is embedded into the tags when we compute the similarities between users, rather than frequency based weights for the tags.

In order to find K similar neighbors, various kinds of proximity computing approaches such as cosine similarity and Pearson correlation can be used [12]. Cosine similarity is popularly used to calculate the similarity of two vectors, it also be used in our approaches. Since the vector of tags with weights in UT is used to represent each item and the preferences of each user, the similarity between users can be measured through calculating the similarity of their weighted tag vectors.

In our method, the similarity between two users u_i and u_j is measured by the cosine similarity, that is defined as:

$$sim(u_i,u_j)=\frac{\sum_{t\in T_{u_i,u_j}}\omega_{u_i,t}\times\omega_{u_j,t}}{\sqrt{\sum_{t\in T_{u_i,u_j}}\omega_{u_i,t}^{2}}\sqrt{\sum_{t\in T_{u_i,u_j}}\omega_{u_j,t}^{2}}} \quad (7)$$

where T_{u_i,u_j} refer to the set of tags both in relevant to user u_i and u_j. $\omega_{u_i,t}$ and $\omega_{u_j,t}$ are respectively the weights of tag t for user u_i and u_j.

Using the similarity measure approach, we can generate the neighborhood of the target user u by sorting the similarity value in descending order. Formally, the neighborhood of user u_i, is denoted as:

$$Neigh(u_i)=\{u_j \mid u_j\in \max K\{sim(u_i,u_j)\}\}$$

where $\max K\{\}$ is used to get the top K values.

E. Recommendation Generation

After generating the set of neighbors, we can learn the historical behavior of neighbor users to predict the target user's fond items. Generally speaking， a set of items that are most frequently rated or tagged by the neighbors of the target user or the most similar to the target user's rated items will be recommended to the target user [12].

1) The Generation of Candidate Item Set

We assume that the target user u prefer the items that his neighbors $Neigh(u)$ prefer, so we generate the candidate item set as follows.

a. For each user $c\in Neigh(u)$, find the rated items in set $Pos(c)$, which contains the fond items of neighbor user c.

b. Delete the items that user u has rated from $Pos(c)$, form the candidate item set, denoted as $item(u,c,I)$.

c. Combine all the candidate item set of user u, denoted as $Can(u,I)=\cup_{c\in Neigh(u)} item(u,c,I)$.

2) Calculating the Similarity between Items

Define $I_{u,T}$ is the set of items that u rated in the past T, which can reflect the user's current interest to some extent. To ensure the number of items in $I_{u,T}$ not too small, we can adjust the T like that.If the number of items in $I_{u,T}$ is less than 10,we can let $T=2*T$. Finally, return the set $I_{u,T}$ and T.

So, computing the item similarity between the set $I_{u,T}$ and $Can(u,T)$ can find the items that the target user u may prefer. Since an item i may similar to several items, and has different similarity value. Thus, we can compute the average similarity value as the item's weights for the target user. Formally, it can be computed by :

$$w(u,i)=\overline{sim(i,I_{u,T})}=\frac{\sum_{i\in Can(u,I),\, j\in I_{u,T}} sim(i,j)}{|I_{u,T}|} \quad (8)$$

where $sim(i, j)$ represents the similarity between item i and j. Differing from the similarity calculation in formula (1), the item i and j may belong different category, we can use Jaccard formulato compute it as follows:

$$sim(i, j) = \frac{|T(i) \cap T(j)|}{|T(i) \cup T(j)|} \qquad (9)$$

where $T(i)$ and $T(j)$ represent the number of tags that related to the item i and j, respectively.

The top-N items with high similarity scores in formula (8) will be recommended to the target user.

IV. EXPERIMENT DESIGN

A. Description of the Data Set

The MovieLens dataset is used in this experiment, which is the most popular dataset used by many scholares to do the collaborative filtering research. It is publicly provided by the GroupLens site (*http://grouplens.org/datasets/movielens/*).

In our experiment, we select the MovieLens 10M as our dataset, it contains 10 million ratings and 100,000 tag applications applied to 10,000 movies by 72,000 users. The rating scale ranges from 1 to 5 in which higher ratings indicate greater preference. All movies belong to 19 classes.

B. Experiments Setup

To evaluate the proposed approaches, we divided the dataset into a training set and a test set randomly. In the dataset, 80% data were randomly used as the training set while rest 20% data were selected as the test set. To ensure the accuracy of experimental results and eliminate the impact of accidental factors, the experiments were repeated five times with different the training/test set. Finally, the result values of this experiment are the averages of the five runs results.

C. Evaluation Metrics

In this paper, the prediction accuracy of top-N recommendations is evaluated by precision and hit-rank.

- *Precision*: It is used to assess the ratio of the recommended list of items that were also contained in the test set. It is defined as follows:

$$precision = \frac{\sum_{u \in U} |TopN(u) \cap Test(u)|}{\sum_{u \in U} |TopN(u)|}$$

where $TopN(u)$ is the set of top-N items that recommended to user u, and the $Test(u)$ is the set of items rated by user u in the test data.

To computing the overallprecision for all users in the test set, we compute it by averaging the personal precision of each user.

- *Average Reciprocal Hit-Rank* (ARHR) : It was introduced by Deshpande and Karypis (2004) [3], which can be used to assess the hit item's position in the recommended list. Generally speaking, a hit that occurs in the first position is better than a hit that occurs in the N-th position. So, we can give the evaluation according to the hits positions. If h is the number of hits that occurred at positions $p_1, p_2, \cdots, p_h$ with in the top-N lists (i.e., $1 \le p_i \le N$), then the average reciprocal hit-rank is defined as :

$$ARHR = \frac{1}{n} \sum_{i=1}^{h} \frac{1}{p_i}$$

That is, hits that occur earlier in the top-N lists are weighted higher than hits that occur later in the list [3].

D. Results and Discussions

In our proposed approaches, we have the parameters T . To test the value of T how to impact on the quality of recommendation, we set the rang of parameters T from 5 to 30 days, and the number of recommend items N is 10.

As can be seen from Fig. 1, it illustrated that the time window$T$between10to 15 dayshas the higher precision, meaning that the recommended effect is good. Also, we can see the precision is decreased when the value of T is larger than 15, which reflect the large time window will not be able to catch the user's current interest, thus a lowprecision may be caused.

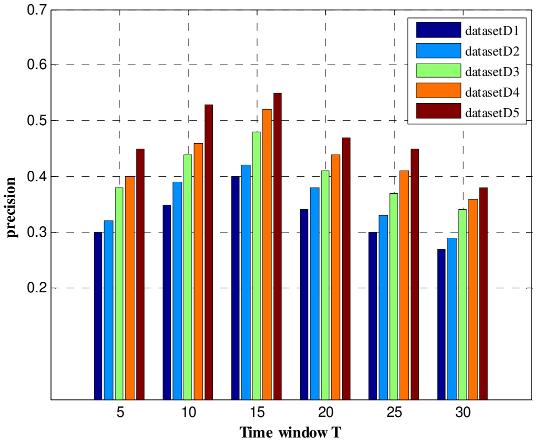

Fig. 1The precision change with T ($N = 10$) for each testing dataset

To verify the effectiveness of the algorithm proposed in this paper, we compare the performance of the algorithm with the User-based collaborative filtering algorithm (UBCF), Item-based collaborative filtering algorithm (IBCF) and Tag-based collaborative filtering algorithm (TCF). The experimental results are shown in Fig. 2 and Fig. 3.

Fig. 2 shows us that the relationship between the number of recommend items and the recommend precision. Obviously, different algorithms have the different performances, but the overall trend is same.With the recommended number N is rising, the precision of the algorithm is decreasing slowly.

The recommended number of item for 5 ~ 10 has the higher recommendation precision, which means the more recommend items be hit in the test set. Compared with the TCF, IBCF, UBCF algorithm, our algorithm has the highest precision, and the drop speed isalso slow.Therefore, theproposed algorithm TARBCF in this paper can capture the user's interests more sensitive, and has more higher credibility.

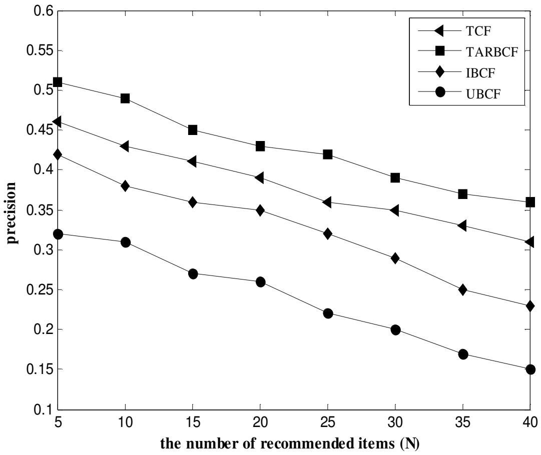

Fig. 1The average precision comparison of the algorithms

Fig. 3 shows us the performance of different algorithms in the ARHR. From the curve in Fig. 3,

we can see that with the increase of number of recommended resources N, the ARHR of each algorithm are improved, and the proposed algorithm has the highest ARHR than other schemes.This fully shows that the items list recommended by our algorithm is more fit the needs of the target user.

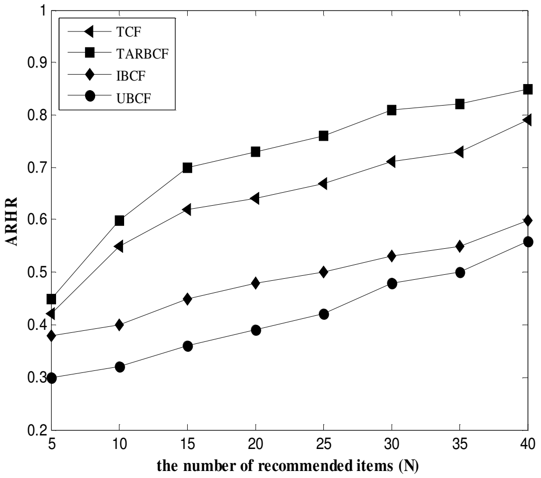

Fig. 3The average ARHR comparison of the algorithms

In a word, theproposed algorithm in this paper has a better performance than others both in the precision and ARHR evaluation metrics. It not only can find the users love items, but also can grasp the degree of user's love. The quality and effect of its recommendations were superior to other solutions.

V. CONCLUSIONS

In this paper, we presented an improved CF algorithm. To weak the effect of data sparsity in the dataset and make the best use of ratings, a rating predict algorithm based on item categoty is introduced to predict the ratings of the no rated items, which can reflect the user's hidden interest to some extent. To improve the quality of recommendations, we further build a user-tag correlation matrix by incorporating with ratings and tags, and use it to generate neighborhoods of the target user. To make the recommend items meet the target user's current interest, we also introduce the time window into the recommendation generation phase. To evaluate the proposed algorithm, an experiments based on MovieLens dataset has been conducted. The experiment results show that the proposed algorithm outperforms other traditionalalgorithm, and has a better quality of recommendation.

In the future, the timestamp of records can be taken into account to track the change of user interests, which can make the recommendation algorithm obtain a higher performance.

ACKNOWLEDGMENT

I would like to express my gratitude to all those who helped me to complete this research, especially Professor Wang Huijin, my supervisor, for his constant encouragement and guidance. I would also like to thank other members in our Computer Science Lab for their supports.

REFERENCES

[1] Adomavicius, Gediminas, and Alexander Tuzhilin. "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions." IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING2005:734--749.

[2] Zhang, et al. "An Improved Collaborative Filtering Algorithm Based on User Interest." Journal of Software9.4(2014).

[3] Deshpande, M. "Karypis G: Item-Based Top-N Recommendation Algorithms."AcmTransactions on Information Systems22.1(2004):143--177.

[4] Hu, Jinming. "Application and research of collaborative filtering in e-commerce recommendation system." Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference onIEEE, 2010:686-689.

[5] Su, Xiaoyuan, and T. M. Khoshgoftaar. "A Survey of Collaborative Filtering Techniques." Advances in Artificial Intelligence2009.2009(2009).

[6] Deng AL, Zhu YY, Shi BL. A collaborative filtering recommendation algorithm based on item ratingprediction. Journal of Software, 2003,14(9):1621~1628. http://www.jos.org.cn/ 1000-9825/14/1621.htm.

[7] Ba, Qilong, X. Li, and Z. Bai. "Clustering collaborative filtering recommendation system based on SVD algorithm." Software Engineering and Service Science (ICSESS), 2013 4th IEEE International Conference onIEEE, 2013:963-967.

[8] Chen, Yongping, S. Yang, and Y. Liu. "Time-weighted collaborative filtering algorithm based on item content and rating." Journal of Suzhou University of Science & Technology(2013).

[9] Ching-man Au Yeung ,, Nicholas Gibbins ,, and N. Shadbolt. "Multiple Interests of Users in Collaborative Tagging Systems." Weaving Services & People on the World Wide Web(2009):115 - 118.

[10] Nakamoto, Reyn Y., et al. "Reasonable tag-based collaborative filtering for social tagging systems." Proceedings of the 2nd ACM workshop on Information credibility on the webACM, 2008.

[11] Wang, J., Clements, M., Yang, J., Vries, A. P. D., & Reinders, M. J. T. (2010). Personalization of tagging systems. Information Processing & Management An International Journal, 46(1), 58-70.

[12] Liang, Huizhi, et al. "Connecting users and items with weighted tags for personalized item recommendations.." Proc of Ht'(2010):51-60.

[13] Kim, Heung Nam, et al. "Collaborative user modeling with user-generated tags for social recommender systems." Expert Systems with Applications38.7(2011):8488–8496.