PAPER

Social Network and Tag Sources Based Augmenting Collaborative Recommender System

Tinghuai MA^{†,††a)}, Jinjuan ZHOU^{††}, Meili TANG^{†††}, Yuan TIAN^{††††}, Abdullah AL-DHELAAN^{††††}, Mznah AL-RODHAAN^{††††}, Nonmembers, and Sungyoung LEE^{†††††}, Member

SUMMARY Recommender systems, which provide users with recommendations of content suited to their needs, have received great attention in today's online business world. However, most recommendation approaches exploit only a single source of input data and suffer from the data sparsity problem and the cold start problem. To improve recommendation accuracy in this situation, additional sources of information, such as friend relationship and user-generated tags, should be incorporated in recommendation systems. In this paper, we revise the user-based collaborative filtering (CF) technique, and propose two recommendation approaches fusing user-generated tags and social relations in a novel way. In order to evaluate the performance of our approaches, we compare experimental results with two baseline methods: user-based CF and user-based CF with weighted friendship similarity using the real datasets (Last.fm and MovieLens). Our experimental results show that our methods get higher accuracy. We also verify our methods in cold-start settings, and our methods achieve more precise recommendations than the compared approaches.

key words: recommender system, collaborative filtering, social tagging, social network

1. Introduction

Recommender systems provide users with personal recommendations of items that they would like most and have become an important research area in the past decade. In general, recommendation systems are classified into collaborative filtering (CF) and content-based filtering (CBF)[1]. CF predicts the interest of a particular user for an item based on other similar users' interests. Rating is the most common information used in CF. However, with the development of social media, the way people find information, share knowledge and communicate with each other are changing [2]. As rich information is shared through the social media sites, it

would be increasingly difficult for users to find what they are actually interested in. In this situation, rating information would be insufficient to predict the preferences of a user. Nowadays, social websites have become a major trend in the Web 2.0 environment [3]. Users in the social websites could give ratings to items, add other users as friends, join in some interest group online, and even add tags to the items, known as social tagging. Besides ratings, other available social data to some extent also indicate the preference of a particular user. Therefore, in recent years, more studies have tried to fuse the social data in recommendation systems to improve the accuracy of the recommender approaches [2], [4]–[7]. At the same time, the additional social data may alleviate the open issues of CF, such as the data sparsity problem and the cold start problem.

In this study, we introduce a new way of integrating social relationships and user-generated tags into the user-based CF and propose two novel recommendation methods cooperating with social relations and user-generated tags. We evaluate our methods on two real datasets (Last.fm and MovieLens), and compare with some basic recommender methods. We also verify our methods in cold start settings.

The rest of this paper is organized as follows. Background knowledge and related work are discussed in Sect. 2. In Sect. 3, we introduce the proposed methods and we evaluate our methods in Sect. 4. Finally, Sect. 5 gives our conclusions and discusses future work.

2. Related Works

In this section, we first review the traditional CF. We then summarize related work on recommenders fusing with tag sources and social relationships separately.

2.1 Recommendation Systems

Recommendation systems provide personalized recommendations to users [8]. One of the most successful technologies among recommender systems is CF, which predicts user preferences based on the taste of other similar users. Generally, CF approaches can be classified as either item-based or user-based techniques [1]. In this study, we focus on user-based CF, which measures the similarity between users in terms of their item rating. Many correlation metrics can be used to compute the similarity between users, for example, cosine-based distance, Pearson correlations and others [9].

Manuscript received August 20, 2014.

Manuscript revised November 10, 2014.

Manuscript publicized December 26, 2014.

[†]The author is with School of Computer, Nanjing University of Information Science & Technology, Jiangsu, Nanjing 210-044, China.

^{††}The authors are with Jiangsu Engineering Centre of Network Monitoring, Nanjing University of Information Science & Technology, Jiangsu, Nanjing 210-044, China.

^{†††}The author is with School of Public Administration, Nanjing University of Information Science & Technology, Nanjing 210044, China.

^{††††}The authors are with Computer Science Department, College of Computer and Information Science, King Saud University, Riyadh 11362, Saudi Arabia.

^{†††††}The author is with the Department of Computer Engineering, KyungHee University, Suwon, 446701, Korea.

a) E-mail: thma@nuist.edu.cn

DOI: 10.1587/transinf.2014EDP7283

The nearest neighbors can be obtained by sorting the similarities between users. By finding the nearest neighbors with similar interests, the user-based CF predicts the preference (i.e., item rating) of an un-rated item i for user u as follow:

$$P_{u,i} = \frac{\sum_{v \in U_{sim}} sim_r(u, v) * r_{v,i}}{\sum_{v \in U_{sim}} sim_r(u, v)} \quad (1)$$

where U_{sim} is the nearest neighbors of user u , $sim_r(u, v)$ is the similarity between user u and user v , $r_{v,i}$ is the rating provided by user v to item i .

A major problem with user-based CF is the data sparsity problem. We know that in traditional CF, user similarity is based on the numeric ratings. However, with the exponential growth of the number of items or users in the websites, few people rate or view few items. The sparsity of item ratings thus affects the similarity metrics and decreases the accuracy of item recommendations finally. Subhash K. Shinde and Uday Kulkarni [10] proposed a novel centering-bunching based clustering (CBBC) algorithm. The proposed system firstly collected the ratings of the users and clustered the users using CBBC. Then the recommendations are generated online for active user using similarity measures by choosing the clusters with good quality rating. The proposed method performs superiorly and alleviates problems such as cold-start, first-rater and sparsity. Joel Pinho Lucas et al. [11] made use of associative classifiers in order to alleviate typical drawbacks in recommender systems. Juntao Liu [12] integrated social relations and item contents into the framework of Bayesian Probabilistic Matrix Factorization (BPMF) in a novel way to alleviate the data sparsity problem and the cold start problem.

2.2 Social Tag Sources

A number of social websites allow users to add tags to items, such as Delicious, CiteULike, Last.fm and so on. The popularity of the usage of user-generated tags allows us to capture valuable information for understanding user interests [2]. User-generated tags have been exploited by many researchers in recommender systems. Marek Lipczak et al. [13] discussed the potential role of the three tag sources: resource content as well as resource profiles and user profiles. They considered the title of a resource as a starting point of the recommendation process and extend the set by tags related to the title as well as tags present in the profiles of resource and user to extract valuable tags from user profile. Nan Zheng et al. [7] investigated the importance and usefulness of tag and time information when predicting users' preference and better performances can be achieved if such information is integrated into CF. In social tagging systems, users and items can be assigned profiles defined in terms of weighted lists of social tags. Ivn Cantador et al. [14] presented and evaluated various content-based recommendation models that make use of the weighted tag profiles of users and items. Heung-Nam Kim et al. [2] proposed

a new collaborative approach to user modeling by leveraging user-generated tags as preference indicators. They also enriched the user models from the neighbors so that the proposed method can provide the proper recommendations even if users rated few items. We also enrich user tag vectors. However, differing from Heung-Nam Kim, our work enriches user tag vectors from his friends, not nearest neighbors. And not every test user needs to be enriched tags from his friends. We only deal with users who have no tags.

2.3 Social Network

Apart from social tags, social websites usually provide social networking functionalities. Users may have explicit friendship or membership relations with others. Recently, researchers have started to incorporate social relations to improve the performance of the standard CF. Ben-Shimon et al. [15] present a collaborative filtering strategy that estimates the rating of an item for a user based on the ratings provided by the user's friends. Fengkun Liu et al. [16] incorporated social network information into CF and evaluated CF performance with diverse neighbor groups combining groups of friends and nearest neighbors. The result showed that more accurate prediction algorithms can be produced by incorporating social network information into CF. He and Chu [4] developed a Bayesian network-based recommender system (SNRS-BN) and study the performance of SNRS-BN under different types of social relationships. Panagiotis Symeonidis et al. [5] proposed the Social-Union which combines similarity matrices derived from heterogeneous explicit or implicit SRNs. Apart from friendships, membership is another kind of social relations. Quan Yuan et al. [6] explored the role of friendship and membership while being fused with traditional CF recommender methods.

In this paper, we propose two novel recommendation methods cooperating with social relations and user-generated tags. Our methods are different from previous methods because we integrate the tags into CF in a novel way and one of our methods integrates both social relations and tags into CF to improve the accuracy of the recommendation. We also enrich user tag vector from user's friends to alleviate the cold start problem. Our methods get higher accuracy in our experimental comparisons.

3. Proposed Method

Since the user-based CF only relies on ratings. It inevitably suffers from sparse and imbalance of rating data. Alejandro Bellogn et al. [17] get the conclusions that when explicit social networks are available, incorporating characteristics of social graphs into the computation of user neighbors in memory-based CF significantly improves recommendation in terms of ranking quality and social tagging can easily be exploited to provide precise item recommendation ranking lists. According to this, we fuse social relations and user-generated tags into CF to improve the accuracy of recommendations. In this section, we will first introduce

the preliminaries for user-based CF (cf-user) and user-based CF with weighted friendship similarity: WS_{fri+ui} [6] (sn-rating). And then, we introduce the user-based CF with tags (cf-tag) and social recommender with tags (cf-sn-tag) in details.

3.1 Preliminaries

3.1.1 User-Based CF (cf-user)

As described in Sect. 2, cf-user first finds the nearest neighbors of the test user who the systems will provide recommendations for, and then predicts the preferences of un-rated items for the test user. The recommender sorts the preferences in descending order, and finally chooses the top n items [18] as the recommendations for the test user. In details, in user-item rating matrix, we view the ratings of users for rated items as vectors. If user u has a rating to item i , then the corresponding cell in rating matrix is $r_{u,i}$, or 0 if no such rating exists. We choose the cosine similarity to get the top k nearest neighbors as follows:

$$sim_{ui}(u, v) = \frac{R_u \cdot R_v}{\|R_u\| \|R_v\|} \quad (2)$$

where R_u and R_v are two vectors of ratings from user u and user v respectively. And then, we predict the rating of test items for test user by Eq. (1). We sorted the predictive ratings in descending order and choose the top n items as the final recommendations.

3.1.2 User-Based CF with Friendships (sn-rating)

As discussed in Sect. 2, many researchers have integrated social relations into CF to improve the accuracy of recommendations. We use sn-rating which fusing friendships with user-item rating matrix via weighted similarity. The friendship is represented by a user-user matrix A . If user u and user v are friends, then the value of the cell $A_{u,v}$ is set to 1, otherwise 0. Based on this user-user matrix, the friendship similarity between two users is calculated by cosine similarity, named sim_{fri} . The model combines sim_{fri} with sim_{ui} (user similarity calculated from user-item rating matrix) in a weighted approach as follows:

$$sim_{ui+fri}(u, v) = \lambda sim_{ui}(u, v) + (1 - \lambda) sim_{fri}(u, v) \quad (3)$$

The parameter λ is used to adjust the weight of sim_{ui} and sim_{fri} , the bigger the λ is, the rating matrix plays a more important role in the combined similarity and vice versa. In several cases, the distribution of the similarity values in the interval [0,1] between sim_{ui} and sim_{fri} differ significantly. For example, consider the case that the most similarity values in user-user matrix are normally distributed between 0 and 0.3, whereas the most similarity values in user-item rating matrix are normally distributed between 0.6 and 0.9. Assume that we want the similarity values based on user-user matrix to have much more impact in the final similarity values than the similarity values based on user-item

rating matrix, we may set λ to 0.4. Then we get the weight of sim_{ui} is 0.4 and the weight of sim_{fri} 0.6. Actually, after weighting, the maximum value of sim_{fri} is 0.18 whereas the minimum value of sim_{ui} is 0.24. We will find that even if λ is less than 0.5, the minimum similarity values of user-item ratings is larger than the maximum similarity values based on user-user matrix. That means only few small values of λ can be chosen for equivalence. Therefore, in this case, some transformations are needed for the two matrixes [5]. We first make transformations to all similarity values of the two matrixes by the following formula:

$$sim_x(u, v) = \frac{sim_x(u, v) - m_x}{s_x} \quad (4)$$

where X is the user-user matrix or the user-item rating matrix. m_x is the mean similarity value of the matrix. s_x is the standard deviation value of the matrix. We then normalize the derived similarity values back in the interval [0,1]:

$$sim_x(u, v) = \frac{sim_x(u, v) - min_x}{max_x - min_x} \quad (5)$$

where max_x and min_x are the maximum and the minimum derived similarity values in matrix X after the transformation of Eq. (4), respectively.

The combined similarity, after transformations, is used to find the top k nearest neighbors for test users. And then, the model predicts the preferences and do recommendations just like what the cf-user does.

3.2 User-Based CF with Tags (cf-tag)

Considering the sparsity and inaccuracy of ratings, we fuse the tags in the predictive process in cf-user.

In social tagging systems, users could add tags to items. Then user and item profiles can be defined in terms of lists of weighted tags. We define the profile of user u as a vector $u = (u_1, \dots, u_t, \dots, u_L)$, where u_t is a weight of tag t . Similarly, we define the profile of item i as a vector $i = (i_1, \dots, i_t, \dots, i_L)$, where i_t is a weight of tag t . There exist different schemes to weight the tags, and we choose the term frequency-inverse document frequency (TF-IDF) weighting scheme for assigning a weight to a particular tag in a user or item profile. Then u_t and i_t can be calculated as follows:

$$u_t = tf_u(t) \cdot iuf(t) \quad (6)$$

$$i_t = tf_i(t) \cdot iif(t) \quad (7)$$

where $tf_u(t)$ and $tf_i(t)$ are frequency of tag t , that is the number of times user u has annotated items with tag t or the number of times item i has been annotated with tag t . $iuf(t)$ and $iif(t)$ are inverse frequency factors that penalize tags that frequently appear in tag-based user and item profiles respectively. More specifically, $iuf(t)$ and $iif(t)$ can be calculated as follows:

$$iuf(t) = \log(M/n_u(t)), n_u(t) = |\{u \in U | u_t > 0\}| \quad (8)$$

Algorithm cf-tag**Input**

n : the number of recommendations
 k : the number of nearest neighbors
 u : the selected test user

Output

$N_RecList$: n item recommendations

```

01 construct the user profiles based on ratings and tags
02 find the top  $k$  nearest neighbors for  $u$  using Equation 2
03 search test items which  $u$  has not rated from the top  $k$  nearest neighbors of  $u$ 
04 for item in test items
05   if  $u$  has tags and item has tags then
06     compute the cosine similarity between  $u$  and item by user tag vector and item tag
       vector
07   else
08     predict the preference of item for  $u$  by Equation 1
09   end if
10 end for
11 if |item recommendations from tags|  $\geq n$  then
12   return  $N\_RecList$ 
13 else
14   take the item recommendations from ratings as the supplement
15 end if
16 return  $N\_RecList$ 

```

Fig. 1 Outline of the user-based CF with tags (cf-tag).

$$ii f(t) = \log(N/n_i(t)), n_i(t) = |\{i \in I | i_t > 0\}| \quad (9)$$

where U and I are respectively the set of all users and the set of all items. M and N are the number of users and items respectively [18]. In cf-tag, we find top k nearest neighbors and test items just like that in cf-user. The difference between cf-user and cf-tag is in the predictive process. Instead of the typical predictive preferences of cf-user, we predict the preferences of items for the test user by user tag vector and item tag vector. We compute the cosine similarity between test user vector and item tag vector and take this similarity as the predictive preference of the item for the test user. At last, we choose the top n items with high preferences as the final recommendations. More specifically, if the test user gets less than n item recommendations or even no recommendations, we use the typical predictive formula in cf-user based on ratings as the supplements. Figure 1 depicts the outline of the algorithmic procedure of the proposed cf-tag. We exploit tags and ratings to make sure that more test users can get precise and enough recommendations.

3.3 Social Recommender with Tags (cf-sn-tag)

For cf-sn-tag, we choose sn-rating model as baseline and integrate tags in this model. As described in Sect. 3.1.2, sn-rating fuses social friendships into cf-user and finds nearest neighbors based on both friendships and ratings. And then, test items are generated from these nearest neighbors. Like cf-tag, we fuse tags in the predictive process in sn-rating. We predict the preference of each test item using test user tag vector and item tag vector.

Considering that some users tagged few items and it will be difficult for the recommender to give them proper recommendations. So we extend the user tag vector by popular tags from user's friends. More specifically, we count

Algorithm cf-sn-tag**Input**

n : the number of recommendations
 k : the number of nearest neighbors
 u : the selected test user

λ : the weight between similarity values based on user-item rating matrix and user-user matrix

Output

$N_RecList$: n item recommendations

```

01 construct the user profiles based on ratings and tags
02 find the top  $k$  nearest neighbors using Equation 3
03 search possible items which  $u$  has not rated from the top  $k$  nearest neighbors of  $u$ 
04 if  $u$  has no tags then
05   we extend the tag vector of  $u$  from his friends by choosing the top 10 popular tags  $u$ 's
       friends have tagged
06 end if
07 for item in possible items
08   if  $u$  has tags and item has tags then
09     compute the cosine similarity between  $u$  and item by user tag vector and item tag
       vector
10   else
11     predict the preference of item for  $u$  by Equation 1
12   end if
13 end for
14 if |item recommendations from tags|  $\geq n$  then
15   return  $N\_RecList$ 
16 else
17   take the item recommendations from ratings as the supplement
18 end if
19 return  $N\_RecList$ 

```

Fig. 2 Outline of the social recommender with tags (cf-sn-tag).

the sum of TF-IDF of each tag used by user's friends and choose the top 10 popular tags as the test user tag vector if the test user has not tagged any items. After tag extending, if there still exist some users who get less or no recommendations, we produce recommendations by Eq. (1) in cf-user as the supplement.

The outline of the algorithmic procedure of the proposed cf-sn-tag can refer to Fig. 2. The differences of cf-tag and cf-sn-tag in the procedure are the selection of neighbors and the extension of user tag vectors from user's friends. Sn-rating, which fusing social relations in user-based CF, helps to find more friends or neighbors for test user and user-generated tags can easily be used to get precise recommendation. Based on social network and user-generated tags, we expect cf-sn-tag will get the best performance values. However, we have to check the above hypothesis empirically since there are some other aspects which may influence the final results.

4. Experimental Evaluation

4.1 Datasets

In order to evaluate the presented cf-tag and cf-sn-tag algorithms, we need the datasets which include ratings, user-generated tags and social relations simultaneously. However, depending on the nature of a system, it may be difficult to find users with enough information of each type [17]. So in our paper, we choose the Last.fm dataset [19] which could meets our needs after simple preprocessing. Apart from Last.fm dataset, we also use Movielens dataset [19] to test the performance of recommenders with ratings and tags.

(1) Last.fm dataset

Last.fm is a popular social music website. Last.fm dataset is released in the framework of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems (HetRec 2011)[19]. There are 1892 users, 17,632 artists (items), 92,834 ratings, 11,946 user-generated tags and 25,434 friend links in it. Explicit social relations and user-generated tags are available while explicit ratings are not. Last.fm dataset records the listening count that each user listened to artists and the listening count indicates the interest of the user to artists. We map listening counts into integer values of 1 to 5 as the implicit ratings of users. The mapping formula is defined as follows [20]:

$$r = \begin{cases} \lfloor \log_{10} l \rfloor + 1, & \text{if } \lfloor \log_{10} l \rfloor + 1 \leq 5 \\ 5, & \text{otherwise} \end{cases} \quad (10)$$

where l is the listening count, r is the implicit rating after mapping. $\lfloor \cdot \rfloor$ is the operator of rounding towards zero.

(2) MovieLens dataset

MovieLens dataset, published by the GroupLens research group at University of Minnesota, is one of the most popular datasets used in the recommender systems. MovieLens dataset is also released in the framework of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems (HetRec 2011)[19]. It contains 2113 users, 10197 movies (items), 855,598 ratings and 13,222 tags in the dataset. Friend links are not available in MovieLens dataset. So in our experiments, we only test recommenders exploiting ratings and tags in MovieLens dataset. Table 1 shows the main characteristics of the two datasets.

4.2 Evaluation Setting

In this section, we describe the methodology followed to evaluate the recommenders. To evaluate the performance of the recommendations, we randomly split the set of items rated by users into a training set and a test set. The training set contains 80% of the items for each user and the test set contains the remaining 20% of the items for each user. We perform 5-fold cross validation procedure in our evaluation. More specifically, in our experiments, we do not split tags into a training set and a test set. Tag vector of a user comes from tags the user has tagged to the items in the training set. Tag vector of an item comes from tags users in the training

set have tagged to this item.

Different from the methodologies in [21], we search test items from nearest neighbors. More specifically, by the top k nearest neighbors, we take items in neighbors' rating vectors which test user has not rated as test items for the test user.

As for the transformation of the similarity matrix, we not only do the transformation in sn-rating and cf-sn-tag algorithms, but also in cf-user and cf-tag algorithms. We find that cf-user and cf-tag algorithms can get better results when the similarity matrix is transformed.

In our experiments, we set the user neighborhood sizes to 10. We do experiments with different neighborhood sizes and find that 10 is relatively the best choice. For sn-rating and cf-sn-tag, we set λ to 0.75.

4.3 Evaluation Metrics

We measure the performance of the recommenders in terms of ranking-based metrics widely used in the area of information retrieval. In our experiments, the recommenders provide a list of n recommended items. And we choose precision, recall, F-measure and DCG to evaluate the ratio and position of relevant items in the ranked lists of recommended items. We consider the items of the test user in the test set are the relevant items of the test user. The final performance value is the average of all the performance values over the set of all test users.

(1) Precision

Precision measures the proportion of recommended items that are relevant and can be defined as follows [17]:

$$\text{precision} = \frac{|\text{relevant items hit}|}{|\text{all recommended items}|} \quad (11)$$

The relevant items hit means items in the recommended lists are also in the sets of relevant items for all test users. All recommended items are the item recommendations all the test users get. When n items are recommended, we call the ratio as precision@ n or p@ n .

(2) Recall

Recall measures the proportion of relevant items that are really recommended and the formula is defined as follows [17]:

$$\text{recall} = \frac{|\text{relevant items hit}|}{|\text{all relevant items}|} \quad (12)$$

The numerator is the same as the precision. The denominator is the sum of the number of relevant items of each test user. When n items are recommended, we call the ratio as recall@ n or r@ n .

(3) F-measure

F-measure[22] is the weighted harmonic mean of precision and recall. The equation is as follows:

$$F = \frac{2 * \text{precision} * \text{recall}}{(\text{precision} + \text{recall})} \quad (13)$$

When n items are recommended, we call the F-measure

Table 1 Description of the datasets.

	Last.fm	MovieLens
number of users	1,892	2,113
number of items	17,632	10,197
number of ratings	92,834	855,598
ratings per user	49.07	404.92
ratings per item	5.26	84.64
number of tags	11,946	13,222
tag assignments	186,479	47,957
tag assignments per user	98.56	22.67
tag assignments per item	14.89	8.12
number of friend links	25,434	0
friend links per user	13.44	0

Table 2 Precision, recall and F-measure scores of $n = 10, 20, 30$ in Movielens dataset.

algorithms	p@10	r@10	f1@10	p@20	r@20	f1@20	p@30	r@30	f1@30
cf-user	0.0196	0.0024	0.0043	0.0233	0.0058	0.0093	0.0279	0.0104	0.0151
cf-tag	0.0346	0.0043	0.0076	0.0342	0.0084	0.0135	0.0337	0.0123	0.0180

score as $f1@n$.

(4) Discounted cumulative gain

Discounted cumulative gain (DCG)[23] measures the usefulness of an item based on its position in a recommendation list. For instance, in the computation of $p@10$ or $r@10$, a relevant item at position 1 in the recommendation list is considered as useful as a relevant item at position 10. DCG penalizes relevant items appearing lower in a recommendation list and the formula is as follows:

$$DCG = \sum_{j=1}^N \frac{r_{u,i}}{\log_2(j+1)} \quad (14)$$

N is the length of recommendation list and $r_{u,i}$ is the relevance value of user u to item i . If item i in recommendation list exists in the relevant items of user u , then we set $r_{u,i}=1$, otherwise 0.

4.4 Experiment Results

In this section, we compare cf-tag and cf-sn-tag algorithms with cf-user and sn-rating algorithms in Last.fm and Movielens datasets.

To experimentally evaluate the performance of top n recommendation, we calculated $p@n$, $r@n$, $f1@n$ and $DCG@n$ of each method in the whole Last.fm and Movielens datasets. We select n as 10,20,30. Since Movielens dataset has no social relations, we only test cf-user and cf-tag in this dataset. We test all the four methods in Last.fm dataset.

Table 2 shows the results of precision, recall and f1 score of cf-user and cf-tag in Movielens dataset. We can find from the table that cf-tag which fusing tags in user-based CF outperforms cf-user in all cases. The result proves that tag sources can be exploited to provide more accurate recommendations than ratings. For the reason that precision and recall do not pay attention to the position of relevant items in recommendation list, we also give the results of DCG of cf-user and cf-tag in Movielens dataset in Table 3. From Table 3, we can see that cf-tag still outperforms cf-user with all different values of n . Considering both Table 2 and Table 3, we find cf-tag is superior to cf-user in all four metrics. We may get the conclusion that tag information is more personalized and efficient than ratings in the recommendation process.

Table 4 and Table 5 show the results of precision, recall, f1 values and DCG of the four recommenders in Last.fm dataset separately. We can see that all the precision values tend to decrease as the number of recommended items n increases, while recall and DCG increase with n . Cf-user is the only recommender which is based on single information and its performance is the worst in the four methods.

Table 3 DCG values of $n = 10, 20, 30$ in Movielens dataset.

algorithms	DCG@10	DCG@20	DCG@30
cf-user	0.1000	0.1688	0.2483
cf-tag	0.1766	0.2622	0.3293

Sn-rating fuses social friendships with ratings and gets better performance than cf-user which based on only ratings. The better results of sn-rating prove that social friendships are helpful to get more accurate recommendations.

However, comparing the results achieved by cf-tag and sn-rating, the performance of the former was found to be superior to that of the latter. The phenomenon demonstrates that in our experiments, when fusing with user-based CF, recommender based on tags performs better than that based on social relations. Combining with the comparison of cf-user and sn-rating described above, we may get that cf-tag is more efficient than sn-rating and sn-rating is better than cf-user. That is, for the three single information, tags are relative more accurate and valuable than social friendships and ratings.

Meanwhile, we can also find that the performance of cf-sn-tag is better than that of sn-rating, as well as that of cf-tag. This result manifests that although tags can be easily exploited to generate more precise recommendations than social relations, the recommender based on the combination of social relations and tags performs better than recommenders based only on social relations or tags. The best performance of cf-sn-tag indicates that tag information and social friendships focus on different points in the recommendations. Both kinds of the information are helpful to the recommendation. If only tag information is taken into consideration, like cf-tag, the relationships and interactions of users will be ignored. While taking social relationships into consideration only, like sn-rating, the tags which reflect the personality and preferences of users will be neglected. So, the experimental results show that the combination of tags and social relationships can get more accurate recommendations.

The above results give an experimental indication that recommenders exploiting more information produce more accurate recommendations for users. More specifically, social relations help to find more reliable neighbors and the recommender based on social relations performs better than user-based CF which based only on ratings. And user-generated tags are personalized content of users and recommenders based on tags provide more precise recommendations than recommenders based on social relations and ratings. Moreover, the recommender fusing both social relations and user-generated tags achieves the best results.

4.5 Cold-Start Settings

We further examine the recommendation performance for

Table 4 Precision, recall and F-measure scores of $n = 10, 20, 30$ in all Last.fm dataset.

algorithms	p@10	r@10	f1@10	p@20	r@20	f1@20	p@30	r@30	f1@30
cf-user	0.0427	0.0433	0.0430	0.0323	0.0656	0.0433	0.0286	0.0872	0.0431
sn-rating	0.0433	0.0440	0.0436	0.0329	0.0667	0.0440	0.0288	0.0878	0.0434
cf-tag	0.0472	0.0479	0.0475	0.0467	0.0944	0.0625	0.0452	0.1359	0.0678
cf-sn-tag	0.0478	0.0485	0.0481	0.0476	0.0965	0.0638	0.0462	0.1391	0.0694

Table 5 DCG values of $n = 10, 20, 30$ in all Last.fm dataset.

algorithms	DCG@10	DCG@20	DCG@30
cf-user	0.2343	0.2909	0.3365
sn-rating	0.2345	0.2946	0.3390
cf-tag	0.2368	0.3519	0.4397
cf-sn-tag	0.2369	0.3577	0.4480

Table 6 Performance values for cold-start dataset.

algorithms	p@10	r@10	f1@10	DCG@10
cf-user	0.0048	0.0224	0.0079	0.0235
sn-rating	0.0071	0.0351	0.0118	0.0340
cf-tag	0.0131	0.0636	0.0217	0.0614
cf-sn-tag	0.0160	0.0815	0.0268	0.0724

users who had few ratings, namely cold start users. We can see from Table 1 that the average number of ratings per user is about 50 in Last.fm dataset, so we consider group of users who have the number of ratings in [2], [15] from the Last.fm dataset and take them as the cold start users. Finally we get 23 cold start users. We compute the precision@10, recall@10, f1@10 and DCG@10 in the four methods in order to analyze the performance of our methods in cold start settings.

Table 6 shows the results of p@10, r@10, f1@10 and DCG@10 of the four recommenders in cold start settings. As we can see from the table, the values of the four metrics of all methods in the cold start settings are apparently worse than those of in the whole Last.fm dataset. Such results were caused by the fact that cold start users are hard to get enough or precise recommendations because they do not have enough personal information.

Specifically, comparing the results of the four methods, we still can find the same rules as those in the whole Last.fm. The values of the four metrics of cf-user are the lowest while sn-rating, which fuses social relations with ratings, gets better performance than cf-user. The two recommenders, namely cf-tag and cf-sn-tag, which exploiting tags achieve significantly better results than those of the corresponding baseline methods. The result indicates that tag sources could provide more precise recommendations than other information. By comparing cf-tag and cf-sn-tag, we find that cf-sn-tag, which incorporating social friendships and user-generated tags into user-based CF, achieves better performance than cf-tag. The reason for the result may be that cf-tag neglects the relations of users which are also very important to the recommendations.

All the results in cold start users imply that our models, fusing tags into the corresponding baseline methods, can considerably improve the recommendation performance. Particularly, cf-sn-tag, which exploiting both social relations

and tags, can help, indeed, in alleviating the problem of the cold start users and thus in improving the quality of item recommendations.

5. Conclusions and Future Work

In this paper, we introduce two hybrid recommenders to integrate social relations and user-generated tags into user-based CF: user-based CF with tags (cf-tag) and social recommender with tags (cf-sn-tag). We performed extensive experimental comparison of our methods against cf-user and sn-rating, using two real datasets (Last.fm and Movielens). The results show that cf-tag and cf-sn-tag yield more accurate recommendations than cf-user and sn-rating. And cf-sn-tag which exploiting both social relations and user-generated tags provides the best recommendations.

In our methods, we fuse tags and social relations into the basic user-based CF and get better results than the baseline methods. However, we use TF-IDF to compute the weight of tags for users and items without considering the polysemy and synonyms of tags. That is, in our methods, we think that tags with exactly the same forms have the same meanings. Actually, a tag may have different meanings when tagged by different users. And for the same item, users may add different tags with similar meanings for it. Both of the situations may bring about inaccurate recommendations and more accurate method should be researched to compute the weight of tags for users and items. As for social relationships, not all friends have the same interests with the test user. Negative impact might be produced by friends who have different interests with the test user and personalized information is needed to find friends with similar preferences.

At the same time, in our methods, we only consider a simple social recommender model: sn-rating. More precise models need to be explored and compared to get the best social recommender model and consequently improve the performance of cf-sn-tag further. In our experiments, we compare our methods with two baseline methods. In future, we will compare our methods against other hybrid recommenders and evaluate the recommenders with more metrics. In addition to friendship and user-generated tags, how to fuse more social information into recommenders is another research direction.

Acknowledgments

This work was supported in part by National Science Foundation of China (No. 61173143, No.11361066), China Postdoctoral Science Foundation(No.2012M511303), and was

also supported by PAPD.

The authors extend their appreciation to the Deanship of Scientific Research at King Saud University for funding this work through research group no. RGP-264.

References

- [1] D.H. Park, H.K. Kim, I.Y. Choi, and J.K. Kim, "A literature review and classification of recommender systems research," *Expert Systems with Applications*, vol.39, no.11, pp.10059–10072, 2012.
- [2] H.-N. Kim, A. Alkhalidi, A. El Saddik, and G.-S. Jo, "Collaborative user modeling with user-generated tags for social recommender systems," *Expert Systems with Applications*, vol.38, no.7, pp.8488–8496, 2011.
- [3] Y. Xu, X. Guo, J. Hao, J. Ma, R. Lau, and W. Xu, "Combining social network and semantic concept analysis for personalized academic researcher recommendation," *Decision Support Systems*, vol.54, no.1, pp.564–573, 2012.
- [4] J. He, *A Social Network-Based Recommender System*, Dissertation, University of California at Los Angeles, p.154, 2010.
- [5] P. Symeonidis, E. Tiakas, and Y. Manolopoulos, "Product recommendation and rating prediction based on multi-modal social networks," *Proc. Fifth ACM Conference on Recommender Systems*, pp.61–68, Chicago, Illinois, USA, 2011.
- [6] Q. Yuan, S. Zhao, S. Ding, L. Chen, X. Zhang, Y. Liu, and W. Zheng, "Augmenting collaborative recommender by fusing explicit social relationships," *ACM RecSys'09 Workshop on Recommender Systems & the Social Web*, 2009.
- [7] N. Zheng and Q. Li, "A recommender system based on tag and time information for social tagging systems," *Expert Systems with Applications* vol.38, no.4, pp.4575–4587, 2011.
- [8] C.C. Chen, Y.-H. Wan, M.-C. Chung, and Y.-C. Sun, "An effective recommendation method for cold start new users using trust and distrust networks," *Information Sciences* vol.224, pp.19–36, 2013.
- [9] L. Lü, C.H. Yueng, M. Medo, Y.-C. Zhang, Z.-K. Zhang, and T. Zhou, "Recommender systems," *Physics Reports*, vol.519, no.1, pp.1–49, 2012.
- [10] S.K. Shinde and U. Kulkarni, "Hybrid personalized recommender system using centering-bunching based clustering algorithm," *Expert Systems with Applications*, vol.39, no.1, pp.1381–1387, 2012.
- [11] J. Pinho Lucas, S. Segrera, and M.N. Moreno, "Making use of associative classifiers in order to alleviate typical drawbacks in recommender systems," *Expert Systems with Applications*, vol.39, no.1, pp.1273–1283, 2012.
- [12] J. Liu, C. Wu, and W. Liu, "Bayesian Probabilistic Matrix Factorization with Social Relations and Item Contents for recommendation," *Decision Support Systems*, vol.55, no.3, pp.838–850, 2013.
- [13] Y.H. Marek Lipczak, Y. Kollet, and E. Milios, "Tag sources for recommendation in collaborative tagging systems," *ECML PKDD Discovery Challenge 2009 DC09*, vol.497, pp.157–172, 2009.
- [14] I. Cantador, A. Belloghí, and D. Vallet, "Content-based recommendation in social tagging systems," *Proc. Fourth ACM Conference on Recommender Systems*, pp.237–240, Barcelona, Spain, 2010.
- [15] D. Ben-Shimon, A. Tsikinovsky, L. Rokach, A. Meisles, G. Shani, and L. Naamani, "Recommender system from personal social networks," K. Wegrzyn-Wolska and P. Szczepaniak (eds.), *Advances in Intelligent Web Mastering*, pp.47–55, Springer Berlin Heidelberg, 2007.
- [16] F. Liu and H.J. Lee, "Use of social network information to enhance collaborative filtering performance," *Expert Systems with Applications*, vol.37, no.7, pp.4772–4778, 2010.
- [17] A. Belloghí, I. Cantador, and P. Castells, "A comparative study of heterogeneous item recommendations in social systems," *Information Sciences*, vol.221, pp.142–169, 2013.
- [18] P. Cremonesi, Y. Koren, and R. Turrin, "Performance of recommender algorithms on top-n recommendation tasks," *Proc. Fourth ACM Conference on Recommender Systems*, pp.39–46, Barcelona, Spain, 2010.
- [19] I. Cantador, P. Brusilovsky, and T. Kuflik, "Second workshop on information heterogeneity and fusion in recommender systems (Het-Rec2011)," *Proc. Fifth ACM Conference on Recommender Systems*, pp.387–388, Chicago, Illinois, USA, 2011.
- [20] O. Koyejo and J. Ghosh, "A kernel-based approach to exploiting interaction-networks in heterogeneous information sources for improved recommender systems," *Proc. 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems*, pp.9–16, Chicago, Illinois, 2011.
- [21] A. Belloghí, P. Castells, and I. Cantador, "Precision-oriented evaluation of recommender systems: An algorithmic comparison," *Proc. Fifth ACM Conference on Recommender Systems*, pp.333–336, Chicago, Illinois, USA, 2011.
- [22] J. Bobadilla, F. Ortega, A. Hernando, and J. Bern, "A collaborative filtering approach to mitigate the new user cold start problem," *Knowledge-Based Systems*, vol.26, pp.225–238, 2012.
- [23] L. Baltrunas, T. Makcinskis, and F. Ricci, "Group recommendations with rank aggregation and collaborative filtering," *Proc. Fourth ACM Conference on Recommender Systems*, pp.119–126, Barcelona, Spain, 2010.

Tinghuai Ma is a professor in Computer Sciences at Nanjing University of Information Science & Technology, China. He received his Bachelor in 1997 (HUST, China), Master in 2000 (HUST, China), Ph.D. in 2003(CAS, China) and was Post-doctoral in 2004(AJOU University, Korea). From Feb. 2009 to Aug. 2009, he was a visiting professor in Ubiquitous computing Lab, Kyung Hee University. His research interests are data mining, grid computing, ubiquitous computing, privacy preserving etc.

Jinjuan Zhou received her Bachelor degree in Software Engineering from Nanjing University of Information Science & Technology, China in 2012. Currently, she is a candidate for the degree of Master of Computer Science in Nanjing University of Information Science & Technology. Her research interest is Recommender system.

Meili Tang is an associate professor in School of Public Administration at Nanjing University of Information Science & Technology, China. She received her master degree from Huazhong University of Science & Technology, China, 2000. Her main research interests are in the areas of e-government and data publishing.

Yuan Tian has received her master and Ph.D. degree from KyungHee University and she is currently working as Assistant Professor at College of Computer and Information Sciences, King Saud University, Kingdom of Saudi Arabia. She is member of technical committees of several international conferences. In addition, she is an active reviewer of many international journals. Her research interests are broadly divided into privacy and security, which are related to cloud computing, bioinformatics, multimedia, cryptograph, smart environment, and big data.

Abdullah Al-Dhelaan has received BS in Statistics (Hon) from King Saud University, on 1982, and the MS and Ph.D. in Computer Science from Oregon State University on 1986 and 1989 respectively. He is currently the Vice Dean for Academic Affairs, Deanship of Graduate Studies and a Professor of Computer Science, King Saud University, Riyadh, Saudi Arabia. He has guest edited several special issues for the Telecommunication Journal (Springer), and the International Journal for Computers and their applications (ISCA). Moreover, he is currently on the editorial boards of several journals and the organizing committees for several reputable international conferences. His current research interest includes: Mobile Ad Hoc Networks, Sensor Networks, Cognitive Networks, Network Security, Image Processing, and High Performance Computing.

Mznah Al-Rodhaan has received her BS in Computer Applications (Hon) and MS in Computer Science both from King Saud University on 1999 and 2003 respectively. In 2009, she received her Ph.D. in Computer Science from University of Glasgow in Scotland, UK. She is currently working as the Vice Chair of the Computer Science Department in College of Computer & Information Sciences, King Saud University, Riyadh, Saudi Arabia. Moreover, she has served in the editorial boards for some journals such as the Ad Hoc journal (Elsevier) and has participated in several international conferences. Her current research interest includes Mobile Ad Hoc Networks, Wireless Sensor Networks, Multimedia Sensor networks, Cognitive Networks, and Network Security.

Sungyoung Lee received his B.S. from Korea University, Seoul, Korea. He got his M.S. and Ph.D. degrees in Computer Science from Illinois Institute of Technology (IIT), Chicago, Illinois, USA in 1987 and 1991 respectively. He has been a professor in the Department of Computer Engineering, Kyung Hee University, Korea since 1993. He is a founding director of the Ubiquitous Computing Laboratory, and has been a director of Neo Medical ubiquitous-Life Care Information Technology Research Center, Kyung Hee University since 2006. Before joining Kyung Hee University, he was an assistant professor in the Department of Computer Science, Governors State University, Illinois, USA from 1992 to 1993. His current research focuses on Ubiquitous Computing and applications, Context-aware Middleware, Sensor Operating Systems, Real-Time.