2010 12th International Asia-Pacific Web Conference

Graph-based Recommendation on Social Networks

Ziqi Wang, Yuwei Tan, Ming Zhang
School of Electronics Engineering and Computer Science
Peking University
Beijing, P.R.China
zikkiwang@gmail.com, tanyuwei715@gmail.com, mzhang@net.pku.edu.cn

Abstract—**Recommender systems have emerged as an essential response to the rapidly growing digital information phenomenon in which users are finding it more and more difficult to locate the right information at the right time. Systems under Web2.0 allow users not only to give resources-ratings but also to assign tags to them. Tags play a significant role in Web 2.0. They can be used for navigation, browsing, recommendation and so on. In this paper, we propose a novel recommendation algorithm, which is based on social networks. The social network is established among users and items, taking into account both the information of ratings and tags. We consider users' co-tagging behaviors and add the similarity relationship to the graph to enhance the performance. Our algorithm is based on the Random Walk with Restarts but provides a more natural and efficient way to represent social networks. Having considered the influence of tags, the transition matrix is denser and the recommendation is more accurate. By evaluating our new algorithm and comparing it to the baseline algorithm which is used in many real world recommender systems on a real life dataset, we make the conclusion that our method performs better than the baseline method.**

Keywords-Recommendation; Random Walk with Restarts; Social Networks; Graph-based Algorithm

I. INTRODUCTION

Development of network technology has accelerated the progress of e-commerce, and commercial interests require web sites to meet the user's interest and to explore their potential needs by providing personalized services. In this situation, the recommendation system came into being [1]. Recommender system is a system that produces individualized recommendations as output or has the effect of guiding the user in a personalized way to interesting or useful objects in a large space of possible options [2]. Recommender systems are used in many fields including books, music and news. Using *Amazon*[1] as an example, a user buys a book from the site, and the system automatically recommends some similar books bought by other users. It is very likely that the user would like these recommended books, and buy more, which brings more profits to the site [3].

With the rapid development of Web2.0 technologies, social networks began to flourish. Social network is a platform on which people can share their views, opinions, experiences and perspective. Such views and experiences can be presented in a variety of forms, including text, images, music and video. Major social networks include blog, online forums, podcast, wiki, folksonomy and so on. Social network becomes the key feature of Web2.0, and tagging is one of t he most important applications in social networks [4] [5] [6].

In a tagging system, users can assign any tag to any resource (liked or disliked). Tags can be seen by everyone. Some of the websites also offer tag recommendation service; a few tags will be recommended when users are tagging. Carmagnola et al. presented that the action of tagging shows user's interactivity level, organization level and interest in a content [7]. Researchers are paying more and more attention to tags recently. *Flickr*[2], which is a typical social networking system, provides users with a platform to upload photos: after registration and login, one user can browse, rate and tag other users' photos. *CiteULike*[3] is a platform for sharing papers and it also allows users to tag their collected papers.

The function of a recommender system is to filter the huge amount of information and pick potential interesting content to the users. It provides these personalized recommendations based on users' browsing histories [8]. However, recommender systems in social networks should be different from the traditional one because a user's tagging behavior could be an important fact for analyzing his personalized interests [9]. In this paper, we explore how to use the tagging information to construct the relationship between users and resources in the social networking system, based on the generic framework of Random Walk with Restarts. We also design two algorithms. The first one treats the tagging behavior as a direct relationship between users and resources and adds the relationship to improve the performance. The second one measures the similarity between users depending on their co-tagging behavior and adds the similarity relationship to promote the performance. Experiments on real data show that, compared to the baseline Random Walk and Restarts method, the second algorithm has better performance and the first one has worse performance.

Within this context we make the following contributions:

[1] http://www.amazon.com/

[2] http://www.flickr.com/

[3] http://www.citeulike.org/

978-0-7695-4012-2/10 $26.00 © 2010 IEEE
DOI 10.1109/APWeb.2010.60

- Our adapted algorithm significantly promotes the performance, compared to the baseline Random Walk with Restart method.
- We show that the relationship between users based on the common tagging behaviors can improve the performance on sparse data of an item recommendation system.
- We successfully combine the user's rating and tagging information of the resources for recommendation.

The rest of the paper is organized as follows. In Section II we review related work and provide useful background for the Random Walk methods used in this paper. The baseline method, Random Walk with Restarts algorithm, and two tag-based algorithms are described in Section III. In Section IV we present the evaluation protocol and analyze the results of the experiments. Finally we conclude this paper in Section V.

II. Related Work

In this section, we review related work in two parts: recommender algorithm and Random Walk.

A. Recommender Algorithm

Recommender algorithm is the core of the recommender system, which determines the performance of the system. There are three popular methods adopted by contemporary recommender systems.

Firstly, the content-based recommendation, which recommends resources based on their content and not on user's rating and opinion [10]. The objects are defined by their associated features of content in the content-based system. The disadvantage of this method is that the resources need to be structural and the taste of users should be described in the features of the content. For examples, in the case of news recommendation, news cannot be characterized by the terms present in a document [11].

Secondly, the collaborative filtering recommendation is one of the earliest and the most successful technologies. It is based on the assumption that similar users express similar interests on similar resources [12] [13]. These collaborative filtering systems can be classified into user-based and item-based systems [14] [15]. User-based system predicts the score of resources by finding the nearest neighbors. We often use cosine-based similarity to get the neighbor set, which can be identified by a threshold or selecting top-N [16]. Item-based system makes prediction by finding similar resources and then takes a weighted combination of their ratings [17]. The strong advantage of this method is that it has no requirements of the recommended object. However, it suffers from data sparseness.

Thirdly, the graph based recommendation which uses transitive associations between users and resources in the bipartite user-resource graph. In graph based algorithm, recommendation problem turns into a node selection problem on a graph. Such graph based recommendation algorithms are getting more attention recently. The Random Walk we will review in the next part is one of the graph based algorithms.

B. Random Walk

The phrase "Random Walk" was first used by Karl Pearson in 1905. It translated from single dimension to higher dimensions in later developments. It is often applied on bipartite graphs in computer science [18] [19]. Konstas et al. recommended items based on the Random Walk in a click-through data graph [17]. Jaschke et al. evaluated adapted PageRank on various types of social networks to recommend tags in folksonomies. They evaluate their system using data from del.icio.us, last.fm and BibSonomy [20].

Random Walk with Restarts is an improvement on the Random Walk technique. It adds the "restart" part in the walk process, which is more suitable for personalization system. Several studies exist in this field of applying Random Walks with Restarts (RWR) on social graph. Some of them compare the algorithm based on RWR model to collaborative filtering and prove that RWR model performs better than collaborative filtering model. Yildirim et al. designed a novel recommendation algorithm which performs Random Walk on a graph that makes use of the similarity measures between items [21]. Konstas et al. took into account both the social annotation and friendships inherent in the social graph to recommend music based on Random Walk with Restarts framework [17]. They evaluated their method with adapted collaborative filtering and proved that RWR model performs better in the area of social networks. Although both studies are close to our approach, they did not take into account the user's similarity in tagging behavior. We make use of this relationship of users to make the transition matrix denser and improve the performance.

III. Tag-based Algorithms

In this section, we describe details of the baseline RWR algorithm and our two algorithms.

A. Random Walk with Restarts(RWR)

The structure of a social network can be described by a graph, *G=(V,E)*, where *V* represents the set of nodes and *E* denotes the set of edges. Users and resources become nodes and the links between them become edges. It is a weighted undirected graph. The edge with larger weight value represents a closer relationship between two nodes. Random Walk with Restarts can be used to calculate the correlation between two nodes.

According to Konstas et al., starting from a node x, a RWR is performed by randomly following a link to another node at each step [17]. Additionally, each step can restart from x with a probability of a. The basic idea of RWR can be expressed as:

$$p^{(t+1)} = (1-a)Sp^{(t)} + aq \ . \qquad (1)$$

$p^{(t)}$, $p^{(t+1)}$ and q are vectors. $p_i^{(t)}$ represents the probability that the random walk at step *t* visits node *i*. *S* is the transition probability matrix and its element, $S_{i,j}$,

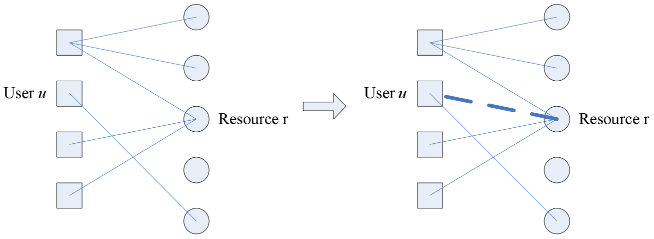


Figure 1. Add an edge from u to r

denotes the transition probability from node *i* to node *j* in one step.

If the graph *G* is connected and the Greatest Common Divisor of the length of all closed walks on *G* is 1, the probability of reaching any node in the graph should reach a stationary distribution after *n* steps and re-iteration does not change the distribution [22]. The probability distribution is calculated using equation (1), which is carried out in a process of random walk in the graph. Repeating the iteration of the equation until p converges, we can get a stationary distribution of the nodes.

Let us consider a movie data set. There are three important parts in this social website: information about the users and resources, information about ratings which users assigned to movies they have watched and information about tags which users assigned to movies. We assume that we are given a set of users *U*, a set of resources *R* and a set of tags *T*. We define a weighted and undirected graph *G* as we described before. *V* consists of two parts: users and resources, which means $V = U \cup R$. If user *i* rates resource *j*, there is an edge between node *i* and node *j*. The weight of the edge is the rating. We define *A* as the weighted adjacency matrix of *G*:

$$A = (ai_j)_{i,j \in V}. \quad ai_j = \begin{cases} rating & (i,j) \in E \\ 0 & otherwise \end{cases}. \tag{2}$$

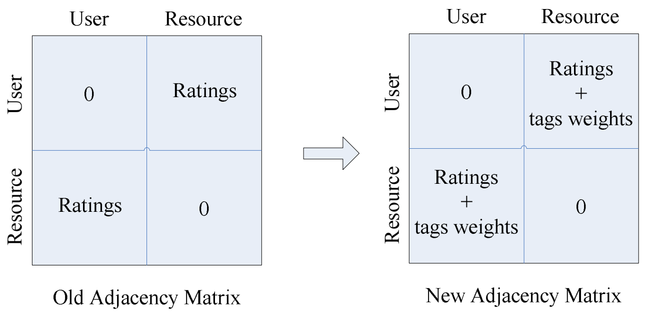
Apparently the matrix *A* is symmetric and the element of *A*, a_{ij}, presents the strength of the relationship between node *i* and node *j*. If $a_{ij} > a_{ij'}$, the relationship between *i* and *j* is closer than that between *i* and *j'*.

We define *S* as the normalized adjacency matrix of the graph [23], which is derived from *A*:

$$S = D^{-1/2} A D^{-1/2}. \tag{3}$$

where *D* is the degree matrix of *A*.

$p^{(0)}$ represents the initial distribution of the nodes of the graph. In our algorithm, the elements of $p^{(0)}$ are all equal and the sum is 1. If we want to recommend resources to user *i*, we will mark node *i* as the starting node. q is a vector in which the element corresponding to the starting node is set to 1. So it should be easily accessible to the node which is closer to the starting node that they would get greater stationary probabilities. Therefore, after the iteration converges, we sort the nodes with their stationary probabilities. The node with larger stationary probability is closer to the target node.



	User	Resource
User	0	Ratings
Resource	Ratings	0

Old Adjacency Matrix

	User	Resource
User	0	Ratings + tags weights
Resource	Ratings + tags weights	0

New Adjacency Matrix

Figure 2. Change in Adjacency Matrix with tags weights

B. *Tag-based Promotion Algorithms*

Here, we propose two algorithms based on tagging information. The first one treats tagging behaviour directly as another form of rating. The second one measures the user's similarity based on their tagging information. We discuss them one by one in this subsection.

1) Treating tagging behaviour directly as another form of rating: By investigating and analysing the real data, we found that a large number of users assign tags on the resources, but do not do the rating. Our first algorithm takes this part of information into account.

Suppose a user *u* assigns a tag *t* on resource *r*, but he didn't rate this resource. So there is no connection between them in the original graph *G*. In our first algorithm, we add an edge from *u* to *r* in the graph by considering users' tagging behaviours as shown in Fig. 1. The adjacency matrix of the graph is also changed.

Fig. 2 shows the changed adjacency matrix of the graph and "tags weights" denotes the new edges. The change in the adjacency matrix makes the transition matrix denser. However, the weight assignment metric of newly added edges is different from the existing edges which directly use the value of user rating. Here, we design three methods to determine the weight of new edges and we will choose the best one in the experiment:

- Assigning the minimum value of user rating to be the weight of each new edge.
- Assigning the maximum value of user rating to be the weight of each new edge.
- Assigning the average rating of the corresponding user to be the weight of the new edge.

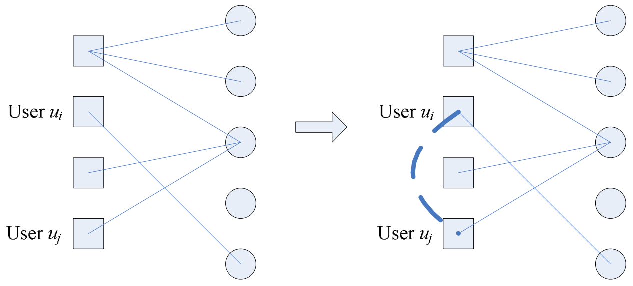
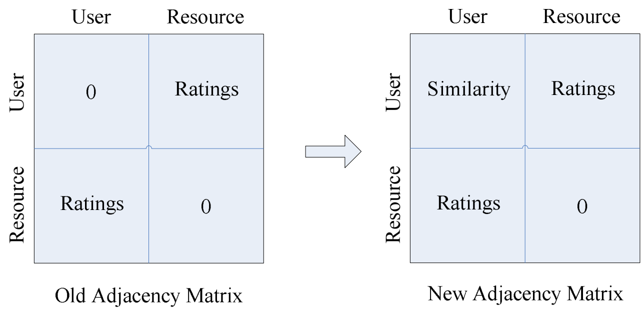


Figure 3. Add an edge from u_i to u_j



Old Adjacency Matrix

	User	Resource
User	0	Ratings
Resource	Ratings	0

New Adjacency Matrix

	User	Resource
User	Similarity	Ratings
Resource	Ratings	0

Figure 4. Change in Adjacency Matrix with Similarity

2) Measuring the user's similarity based on their tagging information: Suppose there are two users, u_i and u_j, u_i assigned tag t to resource r_i and u_j assigned the same tag t to resource r_j. If these two users did not browse the same resource, then they will not have the connection in the original graph. However, we notice that they are actually using the same tag, in other words they are both interested in the content related to tag t. Therefore, the second promotion algorithm uses the tags' similarity between the users to improve the performance.

Assume that n_i is the number of tags user u_i assigned, $t_i^{(k)}$ is the k^{th} tag made by user u_i and the frequency of tag $t_i^{(k)}$ is $c_i^{(k)}$. The pair *(tag, frequency)* is used to describe the interest of user u_i, which should be: $(t_i^{(1)}, c_i^{(1)})$, $(t_i^{(2)}, c_i^{(2)})$, … ,$(t_i^{(n_i)}, c_i^{(n_i)})$. Similarly, the interest of user u_j is $(t_j^{(1)}, c_j^{(1)})$, $(t_j^{(2)}, c_j^{(2)})$, … ,$(t_j^{(n_j)}, c_j^{(n_j)})$. Assume that set T is the set of tags they both assigned and we are here to use the cosine-based similarity to calculate the similarity between the two users:

$$sim(u_i, u_j) = \cos(\vec{u}_i, \vec{u}_j) = \frac{\vec{u}_i \cdot \vec{u}_j}{|\vec{u}_i| \times |\vec{u}_j|} = \frac{\sum_{t \in T} c_i^{(t)} c_j^{(t)}}{\sqrt{\sum_{k=1}^{n_i} c_i^{(k)2} \sum_{k=1}^{n_j} c_j^{(k)2}}} . \qquad (4)$$

We connect two users who assigned the same tags as shown in Fig. 3. The weight of the edge should be proportional to the similarity and we assume it is w:

$$w = k * sim(u_i, u_j) . \qquad (5)$$

where k is a parameter that we will test it in the experiment.

Fig. 4 shows the difference between the old adjacency matrix and new one. The change in the adjacency matrix makes the transition matrix denser.

IV. Evaluation and Experiment

A. Dataset and Evaluation Protocol

We take the experiment on the real movie-rating data set *MovieLen* [4]. MovieLens data set was collected by *GroupLens*[5] Research Project at the University of Minnesota.

We removed some of the users with no rating or no tagging information in the network. After this process, we finally chose the giant connected component to be our experiment data set, which consists of 561 users, 1688 movies, 158812 ratings and 95580 tag assigning information. We split the dataset into training set and test set with the ratio of 90%:10%. While 141,982 ratings were used as training set, the remaining 16,830 ratings served as the test set.

With this datasets, we constructed a graph following the rules described in Section III. We chose positive ratings (3.5, 4, 4.5 and 5) to construct the graph. According to Onuma et al., dealing with negative ratings requires another model and we would like to focus on the evaluation of the performance of our algorithms [23].

The graph is a connected component and we calculate the Greatest Common Divisor of the length of all closed walks on graph, which is 1. So as we discussed in Section III, the algorithm can converges to a stationary distribution.

We evaluate the performance over two widely used information retrieval metrics [11], which are:

- Precision at rank K (P@k): At the most metrics in Information Retrieval field, P@k reports the proportion of resources ranked in the top K results that is relevant. In our experiment, we report the precision at k for three values of k: P@2, P@5, P@10.
- Success at rank K (S@k): This metrics reports the probability of finding a good resource among the top

[4] http://www.movielens.org/
[5] http://www.grouplens.org/

TABLE 1
PERFORMANCE OF RWR WITH DIFFERENT a VALUES

	S@1	S@5	P@2	P@5	P@10
RWR:					
a=0.9	0.1871	0.5365	0.2040	0.2017	0.1942
a=0.8	0.1942	0.5347	0.2023	0.2057	0.1925
a=0.7	0.1818	0.5401	0.1827	0.2053	0.1871
a=0.3	0.1782	0.5204	0.1862	0.1985	0.1823

TABLE 2
PERFORMANCE OF RWUR WITH DIFFERENT METHODS

	S@1	S@5	P@2	P@5	P@10
RWUR:					
Method 1	0.1782	0.5294	0.1818	0.2035	0.1871
Method 2	0.1764	0.5294	0.1827	0.2032	0.1864
Method 3	0.1782	0.5294	0.1827	0.2028	0.1864

K results. We report the success at rank 1 (S@1) and the success at rank 5 (S@5).

The reason why we do not use recall as one of the metrics is that recall and precision actually measure the same thing in this problem. We can see their definitions below:

$$recall = \frac{RelevantNum}{|TS|}. \tag{6}$$

$$precision = \frac{RelevantNum}{|U| * RecommendLength}. \tag{7}$$

where *TS* stands for test set, *U* stands for users set, *RelevantNum* denotes the number of relevant resources in the results and *RecommendLength* represents the number of resources that are recommended to a user. Recall and precision are both determined by *RelevantNum* so that an algorithm with higher precision at rank k must have higher recall at rank *k*.

In addition, users always care about the relevance of the results with high ranking. They will not notice some recommended resources given by system with low ranking, especially after 20. So the performance of the high ranking results should be the most important and we test precision at top 5 and top 10 instead of top 20 and top 50.

B. Experiment

We test the following three algorithms on the dataset:

- Random Walk with Restart(RWR)
- User-Resource Tag-based Algorithm(RWUR)
- User-User Tag-based Algorithm(RWUU)

The algorithms were implemented in the Java language and run on a 1.66GHz Intel Core 2 Duo CPU with 2GB memory running Windows Vista.

Firstly, we test the parameter a in the RWR. The results are showed in Table 1.

a is the probability of re-start and *a=0.8* and *a=0.9* are better then others, which means higher level of personalization can get more precision. So we use *a=0.8* for RWR and we also use it in RWUR and RWUU similarly.

TABLE 3
PERFORMANCE OF RWUU WITH DIFFERENT VALUES

	S@1	S@5	P@2	P@5	P@10
RWUU:					
k=1	0.1889	0.5365	0.1916	0.2078	0.1875
k=3	0.1978	0.5258	0.1996	0.2071	0.1914
k=5	0.2014	0.5418	0.2085	0.2089	0.1934
k=7	0.1942	0.5347	0.2023	0.2057	0.1925

TABLE 4
EVALUATION RESULTS OF THREE ALGORITHMS

	S@1	S@5	P@2	P@5	P@10
Baseline Algorithm:					
RWR	0.1942	0.5347	0.2023	0.2057	0.1925
Promotion Algorithm:					
RWUR	0.1782	0.5294	0.1827	0.2028	0.1864
RWUU	0.2014	0.5418	0.2085	0.2089	0.1934
Improvement:					
RWUR	-8.2%	-0.9%	-9.6%	-1.4%	-3.3%
RWUU	3.7%	1.32%	3.1%	1.55%	0.4%

Secondly, we should choose a method to determine the weight of the new edges in the RWUR. The performance of RWUR with different methods is presented in Table 2.

Because we only choose positive rating, the average rating should between 3.5 and 5, while the minimum and maximum of rating are 3.5 and 5 respectively. So the performances have negligible difference in the three methods, which is listed in Table 2. Finally, we choose the third method: assigning the average rating of the corresponding user to be the weight of the new edge.

Thirdly, we test the parameter *k* in the RWUU. The results are presented in Table 3.

The average value of user's similarity was about 0.07 and the average rating score was 4.07. k determines the degree of importance of the link between users to users' rating information. From the experimental results, when k=5, the various indicators are optimal. At this point, the average similarity is 0.35, so their ratio is about 1:10, which comes to the global optimum. From this experiment, we can see that rating information is still the most important feature in building user's interest model, but user's tagging behaviours should still be considered to make the model perfect.

C. Results Analysis

We carried out evaluation on the baseline, RWR and our two algorithms together and make comparisons. The results are listed in Table 4.

From the above results, we can conclude that the performance of RWUR is lower compared to baseline algorithm RWR. The reason is that assigning a tag to a resource doesn't mean that the user likes this resource. Users may simply feel that the film is a representative one, or simply to be marked, but have never watched it. If the user has really watched the movie, he would usually rate it, rather than just tag it.

TABLE 5
THE RANGE OF RATED RESOURCES NUMBER IN EACH GROUP

	Rated Resources Number
Group 1	<=100
Group 2	(100,200]

TABLE 6
EVALUATION RESULTS OF RWR AND RWUU IN DIFFERENT GROUPS

	S@1	**S@5**	**P@2**	**P@5**	**P@10**
Baseline Algorithm RWR:					
Group 1	0.2210	0.5052	0.2105	0.2042	0.1852
Group 2	0.2795	0.7763	0.2981	0.2658	0.2167
Promotion Algorithm RWUU:					
Group 1	0.2315	0.5473	0.2263	0.2189	0.1926
Group 2	0.2981	0.7826	0.3074	0.2658	0.2204
Improvement:					
Group 1	4.75%	8.3%	7.5%	7.1%	3.9%
Group 2	6.6%	0%	3.1%	0%	1.7%

The performance of RWUU is higher compared to the baseline. RWUU algorithm increases 3.1% in P@2 and 1.55% in P@5. In other words, it makes the relevant results rank higher. It also increases in metrics S@1 and S@5 with 3.7% and 1.32%, which means that more users can get their satisfactory results in the top 5. However, we notice that the extent of the increase of S@5 and P@5 is smaller than S@1 and P@2 and we think the reason is that not enough tagging information can be used in this dataset, so the similarity of users can't be captured and expressed well. In our training test, the amount of tagging information is just one tenth of rating information. In this respect, the system should give users some tagging recommendation when they are assigning tags, so as to encourage the user tagging behaviour. Besides, giving bonus to the users who mark more tags can increase their enthusiasm. In summary, adding the information based on users' co-tagging behaviours is beneficial to the recommendation results.

In addition, we choose two groups of users based on the number of resources they have seen. The group information is listed in Table 5.

We repeat the experiments in these two groups and see the performance of our promotion algorithm under the different groups of sparse data. The results are listed in Table 6.

We can see from the results that the algorithm shows better performance and higher increase in the sparser data set. Group 2 is comparatively denser. Because the number of users is smaller in this group, two algorithms are nominated for the same precision on two metrics. But we still can see the promotion of performance in the first group, which shows that the improved algorithm is more practical in the sparse data sets. The explanation for this phenomenon is that in the sparse data sets we can only get little information about the user, so the user model is not entirely correct. Here we add the tagging information in it and more relationships are built between users, so the user interest model becomes more complete.

If there are a lot of resources in the system, it is likely that different users browse diverse resources. So less common resources are shared between users and it will be hard for the current recommendation algorithm to obtain a good effect. However, we notice that the use of tags would be general that people usually use some common tags to describe diverse resources. For example, if a user looked some horror movies which almost no other user has seen, the user can not establish any connection with someone else, but if the user assigns some tags to the movies, he may choose tag "horror", which will set up a link to other users, and that can still makes our recommendation algorithm recommend some relevant results.

V. CONCLUSIONS

Social network plays a fundamental role in the information time. As an important element of social network in Web2.0, tags are getting more attention from websites. It is critical to analyze users' tagging behaviors to understand their interests. Tags can be viewed as a feature of user's model and two people using the same tag means they have some connections in a social website.

We designed two algorithms in this paper based on the framework of Random Walk with Restarts. The two methods use tagging information in different ways. Our experimental results prove that adding the similarities among users based on their own tagging information makes the transition matrix denser and enhances performance. We choose two groups of users with different numbers of rated resources and performed the experiments again; this proves that our promotion algorithm performs better on sparse data sets.

Our future work will focus on recommendation on large-scale data with better performance and lower time cost. As the efficiency of RWR is not as good as collaborative filtering algorithm, it cannot be applied to online recommendation. We should improve RWR algorithm for large-scale data to cope with the growing size of social network. Tong et al. designed a fast algorithm based on the framework of RWR, which can maintain 90% precision rate and increase the efficiency [24]. Das et al. introduced a distribution method to solve the scalable online collaborative filtering with "MapReduce" [11]. So our future work has two directions. The first one is to design an efficient algorithm without a significant reduction in precision. This algorithm should have better performance than collaborative filtering and can be applied to online recommendation. The second direction is the combination of Random Walk with Restarts framework and distribution method, which is likely to give a more robust performance.

ACKNOWLEDGMENT

This study is partially supported by the National High Technology Research and Development Program of China (863 Program Nos. 2009AA01Z143 and 2009AA01Z136), and the Special Fund for Fast Sharing of Science Paper in Net Era by CSTD ("FSSP" Grant No. 20090001110106). We thank Fei Yan, Jian Tang, Xiaolong Wang, Zhenyun Huang and Ping Yin for their invaluable discussions, as well as Bolanle Ojokoh and other reviewers.

REFERENCES

[1] G. Adomavicius and A. Tuzhilin. "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions." In IEEE Transactions on Knowledge and Data Engineering, Volume 17, Issue 6, 734-749, 2005.

[2] R. Burke. "Hybrid Recommender Systems: Survey and Experiments." User Modeling and User-Adapted Interaction 12: 331-370, 2002. Kluwer Academic Publishers.

[3] B. Sarwa, G. Karypis, J. Konstan and J. Riedl. "Analysis of recommendation algorithms for E-commerce." In ACM Conference on Electronic Commerce, 2000, pp. 158-167.

[4] G. Das, N. Koudas, M. Papagelis and S. Puttaswamy. "Efficient sampling of information in social networks." In SSM '08: Proceeding of the 2008 ACM Workshop on Search in Social Media, Napa Valley, California, USA.

[5] P. Yin, M. Zhang and X. Li. "Recommending Scientific Literatures in a Collaborative Tagging Environment." ICADL '07: 10th International Conference on Asian Digital Libraries. Hanoi, Vietnam, December 10-13, 2007, Proceedings.

[6] E. Michlmayr and S. Cayzer. "Learning User Profiles from Tagging Data and Leveraging them for Personalized Information Access". In WWW '07: Proceedings of the 18th International Conference on World Wide Web, Madrid, Spain, April 20-24, 2009.

[7] F. Carmagnola, F. Cena, O. Cortassa, C. Gena and I. Torre, "Towards a tag-based user model: how can user model benefit from tags?" Available:http://my-svn.assembla.com/svn/armin-da/literatur/HowCanUserModelBenefitFromTags.pdf

[8] Y. Wu, Y. Chen and A.L.P. Chen. "Enabling Personalized Recommendation on the Web Based on User Interests and Behaviors". Proceedings of the 11th International Workshop on research Issues in Data Engineering, page: 17.

[9] . Shardanand and P. Maes. "Social Information Filtering: Algorithms for Automating 'Word of Mouth'". Proceedings of the SIGCHI conference on Human factors in computing systems, pages: 210-217.

[10] A.M. Ferman, J.H. Errico, P. van Beek and M.I. Sezan. "Content-based filtering and personalization using structured metadata." In JCDL '02: Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries, pages 393-393, New York, NY, USA, 2002.ACM.

[11] A. Das, M. Datar, A. Garg and S. Rajarm. "Google News Personalization: Scalable Online Collaborative Filtering." In WWW '07: Proceedings of the 16th international conference on World Wide Web, Pages: 271-280.

[12] D. Zhou, S. Zhou, K. Yu, X. Song, B.L. Tseng, H. Zha and C.L. Giles. "Learning Multiple Graphs for Document Recommendations". In WWW '08: Proceeding of the 17th international conference on World Wide Web, Pages 141-150.

[13] H. Halpin, V. Robu and H. Shepherd. "The complex dynamics of collaborative tagging." In WWW '07: Proceedings of the 16th international conference on World Wide Web, pages 211-220, New York, NY, USA, 2007. ACM Press.

[14] W. Chen, D.Z. Edward and Y. Chang. "Combinational Collaborative Filtering for Personalized Community Recommendation". In KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, page: 115-123.

[15] A.S. Harpale and Y. Yang. "Personalized Active Learning for Collaborative Filtering". In SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, page 91-98.

[16] J. Wang, A.P. de Vries and M.J.T. Reinders. "Unifying User-based and Item-based Collaborative Filtering Approaches by Similarity Fusion." In SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, Pages: 501-508.

[17] I. Konstas, V. Stathopoulos, J.M. Jose, "On Social Networks and Collaborative Recommendation" In SIGIR '09: Proceedings of the 32nd Annual International ACM SIGIR, Boston, MA, USA.

[18] M. Clements, A.P. de Vries and M.J.T. Reinders. "Optimizing single term queries using a personalized markov random walk over the social graph." In Workship on Exploiting Semantic Annotations in Information Retrieval (ESAIR), March 2008.

[19] L. Lovasz. "Random walks on graphs: A survey." In Combinatronics, volume 2, pages 1{46. Bolyai Society for Mathematical Studies, 1996.

[20] R. Jaschke, L.Marinho, A. Hotho, L.Schmidt-Thieme and G. Stumme. "Tag Recommendation in Folksonomies." In PKDD' 07: Proceedings of the 11th European conference on Principles and Practice of Knowledge Discovery in Databases, Pages: 506-514.

[21] H. Yildirim and M.S. Krishnamoorthy. "A random walk method for alleviating the sparsity problem in collaborative filtering." In RecSys '08: Proceedings of the 2008 ACM conference on Recommender systems, pages 131-138, New York, NY, USA, 2008. ACM.

[22] P. SarkarRandom. "Walk on Graphs: An Overview." Available: http://www.cs.cmu.edu/~learning/talks-2007-fall/slides/2007-10-01.sarkar.ppt

[23] K. Onuma, H. Tong and C. Faloutsos. "TANGENT: A Novel, 'Surprise-me', Recommendation Algorithm." In KDD'09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, Pages 657-666.

[24] H.Tong, C.Faloutsos and J. Pan. "Fast random walk with restart and its applications." In ICDM '06: Proceedings of the Sixth International Conference on Data Mining, pages 613-622, Washington, DC, USA, 2006. IEEE Computer Science.