Personalized Recommendation via Relevance Propagation on Social Tagging Graph

Huiming Li, Hao Li, Zimu Zhang, and Hao Wu(✉)

School of Information Science and Engineering, Yunnan University,
No.2, North Green Lake Road, Kunming 650091,
People's Republic of China
haowu@ynu.edu.cn

Abstract. This paper presents a novel random walk based relevance propagation model for personalized recommendation in social tagging systems. In the model, the tags are used to express the profiles of both users and resources, and then candidates of resources are recommended to the users based on the profile relevance between them. In particular, how the users to find the resources of interest is modeled as a random walk by which the relevance spreads in User-Resource-Tag relation graph. Experimental results on two real datasets collected from social media systems show the merits of the proposed approach.

1 Introduction

Collaborative tagging systems [1], such as Delicious, Flickr, Youtube, Lastfm, Connotea, CiteUlike and MovieLens, have become a kind of booming business on the Internet. These systems provide a wealth of information, where any persons can freely find, annotate, organize various resources of interest and share their findings (this practice is coined as Folksonomy by Thomas Vander Wal). As an information carrier, the tags play a key role in such systems. Since they cannot only express the main features of the resources, but also cover relationships of users-resources/items(we use them alternatively) and items-items.

The size and complexity of folksonomy-based systems can unfortunately lead to information overload and reduced utility for users. Too many resources can make users helpless in their process of finding useful contents. Consequentially, the increasing need for recommender services from users has arisen. For these reasons, researchers have sought to apply the techniques of recommender systems to deliver personalized views. The current researches of personalization in such systems can be classified into tag recommendation and item/resource recommendation. Given a user and a resource, the former predicts what and how tags will be adopted by the user to explain the resource, whereas the latter emphasizes suggesting unseen items of interest to the user. Compared to tag-oriented recommendation research, how to develop tag-aware personalized recommendation technologies to come forward with the application needs remains many issues [2]. To this end, this paper presents a novel random walk based relevance propagation model for personalized recommendation in social tagging systems. In the

W.-S. Han et al. (Eds.): DASFAA 2014, LNCS 8505, pp. 192–203, 2014.
DOI: 10.1007/978-3-662-43984-5_14,

model, the tags are used to express the profiles of both users and resources, and then candidates of resources are recommended to the users based on the profile relevance between them. In particular, how the users to find the resources of interest is modeled as a random walk by which the relevance spreads in User-Resource-Tag relation graph. Experimental results on two real datasets collected from social media systems, show that our model can improve the accuracy of resource discovery, and thus enhance the personalized recommendation in social tagging systems.

The rest of paper is organized as follows. Section 2 presents our models in detail. How to assess the relevance of user-user and user-resource, how to extract and build neighborhood-based social tagging graph and how the relevance spreads with a random walk on extracted graph are proposed. Next, in Sect. 3, the solid experiments are conducted to watch the effectiveness of our method. Then, we review some works most akin to us, and make some discussions. Finally, we conclude the works and point to future directions.

2 Methods

2.1 Neighborhood-Based Social Tagging Graph

One of the most commonly used algorithms in personalized recommendation is a neighborhood based approach [3], which works by first computing similarities between all pairs of users, and then to predict by integrating ratings of neighbors. Here, we follow this common idea to create a neighborhood-based tripartite graph G_{URT} for personal resource recommendation in social tagging systems.

Given a random user u_i, we first define a user profile as: $\boldsymbol{u_i} = (r_{i,1} : w_{i,1}, r_{i,k} : w_{i,k}, ..., r_{i,n} : w_{i,|R|})$, where $r_{i,k}$ is the k-th resource collected by u_i, $|R|$ is the cardinality of resource collection, $w_{i,k}$ is the preference degree of u_i on resource $r_{i,k}$. Then, we estimate the pairwise relevances of users using the cosine similarity (Eq. 1),

$$R(u, u_i) = cosine(\boldsymbol{u}, \boldsymbol{u_i}) = \frac{\sum_{k=1}^{|R|} w_k \times w_{i,k}}{\sqrt{\sum_{k=1}^{|R|} {w_k}^2}\sqrt{\sum_{k=1}^{|R|} {w_{i,k}}^2}} \quad (1)$$

Both w_k and $w_{i,k}$ in user profile $\boldsymbol{u}$ and $\boldsymbol{u_i}$ can be obtained by TF-IUF(Term Frequency-Inverse User Frequency) as followings,

$$w_{i,k} = tf_{i,k} \cdot log\frac{|U|}{uf_{i,k}} \quad (2)$$

where, $tf_{i,k}$ is the normalized occurrence frequency of the k-th resource in the user profile $\boldsymbol{u_i}$, $|U|$ is the cardinality of user collection, $uf_{i,k}$ is the total number of the user profiles in which the k-th resource occurred.

In the same way, we can represent a profile of the resource as a tag-aware vector, and estimate the relevance $R(u, r)$ between a user u and a resource r.

Given a target user u, the top-n users most similar to u are firstly found using metric (Eq. 1), these users together with u forms a virtual community $C(u)$. $C(u)$

is added to G_{URT}. Then, the resources and the tags used by $C(u)$ to annotate them are added to G_{URT}. Finally, the User-Resource-Tag relations are created according to social annotation traces. Built on the tripartite graph, if we want to automatically point a user to the most interested resources, we should imagine how the user searches resources of interest as he/she gradually surfs on a social tagging system. For simplicity, we present two independent and repeated surfing processes. In the first case, we assume that users preferably consult with community members as they search resources:

- At any time: (a) randomly visit a resource, or just pick a random community member;
- After consulting with a community member: (a) pick a resource tagged by this member, or (b) consult with another member recommended by current member;
- After visiting a resource: (a) consult with a community member who tagged this resource, or (b) visit another resource linked to this resource.

In the second case, we assume that users search resources by prefer to exploit tags:

- At any time: (a) randomly visit a resource, or just pick a random tag;
- After viewing a tag, pick a resource annotated by this tag;
- After visiting a resource, pick an interested tag to further view.

To model resource gathering process as well as to reduce the complexity of our proposed model, we separate the tripartite graph G_{URT} into two bipartite subgraphs as User-Resource graph G_{UR} and Resource-Tag graph G_{RT} (shown in Fig. 1), to respectively address the two surfing processes. And then, in our method described further, we try to overcome the limitations of the state-of-the-art works by modeling resources search as two infinite random walks on these two graphs. The results of these two processes are finally integrated to reach the original purpose of our model, as exploiting rich semantics within G_{URT} as far as possible to recommend resources to users.

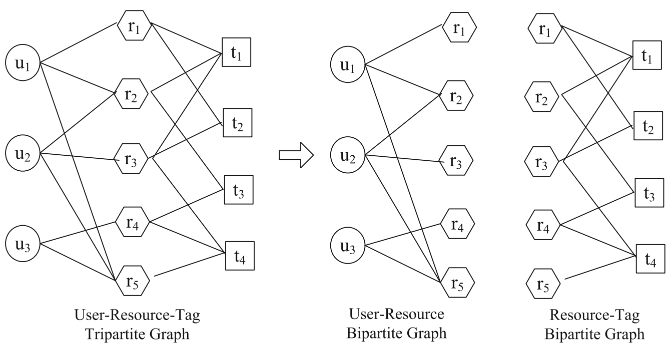

Fig. 1. An example of social tagging graph.

2.2 Relevance Propagation Based on Random Walk

We suppose that the walk in finding resources for a current user u is a non-stop process. That is, u visits the nodes in User-Resource graph or Resource-Tag graph over and over again. During this infinite walk (also, a discrete Markov process), the resources visited more often are considered more beneficial for u. However, the stationary distribution of such a random walk does not depend on the state of the initial probability distribution. To assure the existence of a stationary distribution, also retain the importance of a candidate resource to stay close to relevant tags or users, the jump transition need to be added to the graph nodes.

We first introduce the possibility to return regularly to the resource nodes from any node of the bipartite graph and to start the walk through mutual resource-user or tag-resource links again. The likelihood of jumping to the specific resource $P_J(r_j)$ (shown as Eq. 3) is considered to equal its normalized probability to be relevant to the current user u. This assumption makes candidates situated closer to u, and the more likely that the candidate is known to u, the more it can be selected for a random jump.

$$P_J(r_j) = \frac{R(u, r_j)}{\sum_{r_k \in C(u)} R(u, r_k)} \tag{3}$$

Then, the probability to jump to a user $P_J(u_i)$(in G_{UR}) and a tag $P_J(t_i)$(in G_{RT}) is added, respectively. We consider that the taste of a community member $u_j \in C(u)$ is more close to the current user, or a tag t_i is more popular in the community, it is visited more often by the current user during consecutive walk steps. So, we make $P_J(u_i)$ equal to the normalized similarity of u_i to u, and let $P_J(t_i)$ equal to the probability to find the tag t_i in the community. These two measures of jump transitions are shown as followings,

$$P_J(u_i) = \frac{R(u, u_i)}{\sum_{u_k \in C(u)} R(u, u_k)} P_J(t_i) = \frac{cf(t_i)}{\sum_{t_k \in C(u)} cf(t_k)} \tag{4}$$

where, $cf(t_i)$ is the occurrence frequency of the tag t_i in the community, $R(u, u_i)$ and $R(u, r_j)$ are same to the above-mentioned definition. For relevance propagation on User-Resource graph, the following HITS-like equations are used for iterations until convergence:

$$P^n(u_i) = dP_J(u_i) + (1-d) \sum_{r_j} P(u_i|r_j) P^{n-1}(r_j) \tag{5}$$

$$P^n(r_j) = dP_J(r_j) + (1-d) \sum_{u_i} P(r_j|u_i) P^{n-1}(u_i) \tag{6}$$

For relevance propagation on Resource-Tag graph, the following equations are used for iterations until convergence:

$$P^n(t_i) = dP_J(t_i) + (1-d) \sum_{r_j} P(t_i|r_j) P^{n-1}(r_j) \tag{7}$$

$$P^n(r_j) = dP_J(r_j) + (1-d)\sum_{t_i} P(r_j|t_i)P^{n-1}(t_i) \quad (8)$$

where d is the probability that at any step the user decides to make a jump and not to follow outgoing links anymore. According to our test, setting $d \in [0.1, 0.2]$ is a good choice for the most cases. The convergence condition of iteration is given by $|P^n(\cdot) - P^{n-1}(\cdot)| \leq \epsilon$. The described Markov process is aperiodic and irreducible, and hence has a stationary distribution. Consequently, we consider to integrate the two stationary probabilities $P_{UR}(r_j)$ and $P_{RT}(r_j)$ (shown as Eq. 9) as the final relevance of the resource r_j to the target user u.

$$R(u, r_j) = \lambda P_{UR}(r_j) + (1-\lambda)P_{RT}(r_j) \quad (9)$$

Algorithm 1. *RPRW:Relevance Propagation with Random Walk*

Require. A target user u, three parameters k, d and λ, convergence threshold ϵ.
Ensure. A ranked list of resource set I.

```
for each u_i ∈ U do
    estimate the relevance R(u, u_i);
end for
retrieve the top-k similar neighbors of u;
create C(u), and poll all resources in C(u) as the candidate set I;
create bipartite graphs G_UT and G_RT based on C(u) and I;
for each r_j ∈ I do
    estimate the relevance R(u, r_j);
end for
normalize R(u, u_i) and R(u, r_j);
repeat
    update P(u_i) and P(r_j) with d on G_UT according to Eq. 5 and Eq. 6;
until converged
repeat
    update P(t_i) and P(r_j) with d on G_RT according to Eq. 7 and Eq. 8;
until converged
get the final relevance R(u, r_j) between u and r_j, using Eq. 9;
return A ranked list of I;
```

The whole process of our proposed method is explained as Algorithm 1. A main part of the algorithm is to calculate the relevance $R(u, u_i)$ and $R(u, r_j)$, however, such a computational overhead can be controlled in an acceptable range by pre-clustering users. Another main part of the algorithm is concerning the relevance propagation with random walk. In each iteration, the random walk probability is updated from the neighbor nodes of u, so the complexity of the algorithm is $O(f(k))$, where $f(k)$ marks the scale of nodes surrounding to u. Also, in experiments, the calculation converges fast (after 20–40 iterations) by assigning ideal setting to d and ϵ. Therefore, the whole time cost for each recommendation is acceptable in realtime scenario.

3 Experiments

3.1 Datasets

For experiments, we use the actual datasets collected from two well-known social media systems-Lastfm and Movielens. Lastfm[1] is the world's largest online music catalogue, and allows user tagging music tracks and artists. In this dataset, we take artists as resources. MovieLens[2] is a recommender system and virtual community website that recommends films for its users to watch, based on their film preferences and using collaborative filtering. The website is kept by the lab of GroupLens Research. The collaborative tagging function had been included in the website, thus researchers can gather tag-aware data for research purpose. For these three systems, we use their data collections released in the framework of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems [4] to make an evaluation. Statistics of datasets are listed in Table 1, and more detailed descriptions of these datasets can be found in [5].

Table 1. The basic statistics of the datasets.

Dataset	Users	Resources	Tags	Tas(UR)	Density(RT)	Density(u)	Training(u)	Test(u)
Lastfm	1,892	12,523	9,749	186,479	$3.0 * 10^{-3}$	$2.2 * 10^{-3}$	1,821	337
MovieLens	2,113	5,908	9,079	47,957	$9.0 * 10^{-4}$	$7.0 * 10^{-4}$	1,598	135

To test the algorithmic performance, the Lastfm dataset is divided into two parts according to the tag assignment(tas) timestamp: the training set contains 90 % past entries and the remaining 10 % future entries make up the testing set. Because test cases for the Movielens dataset are relatively small, we separate this dataset by the ratio of 80 %:20 %. This policy follows the universal observation as known information used for recommending, while no information in the testing set is allowed to be used for recommending. Also, it meets the online operation principle of recommender systems, that is, the recommender periodically provides active users with resources of interest, at a certain point of time, using the historical data of the systems. Note that, since we do not concentrate on the cold-start problem in this paper, new users and new resources are eliminated from the testing dataset. The finally selected test cases are also presented in Table 1. Also, when generating the recommendation candidate list for a specified user, the resources already collected by the user are excluded from the list.

3.2 Evaluation Metrics and Baseline Methods

To give solid and comprehensive evaluation of the proposed algorithm, we employ three well-known metrics: Precision at top-K($P@K$), Recall at top-K($R@K$),

[1] http://www.lastfm.com
[2] http://www.imdb.com, http://www.rottentomatoes.com

and their harmonic mean-F1 metric at top-K($F1@K$), to characterize the accuracy of recommendations. In addition, Hamming Distance is selected to measure the diversity of recommendation. It examines the uniqueness of recommendation lists to separate users. Given two users i and j, the hamming distance between their recommendation lists can be calculated by Eq. 10.

$$HD_{ij}(k) = 1 - \frac{overlap_{ij}(k)}{k} \tag{10}$$

where $overlap_{ij}(k)$ is the number of shared items in the top-k places of the two recommendation lists. Averaging over all pairs of users, we can obtain the aggregate diversity of the system. Clearly, higher diversity means higher personalization of users' recommendation lists, $HD(k) = 1$ points to the fact that every user receives his/her own unique top-k items.

As far as the baseline method concerned, the approaches that recommend tags or use explicit ratings or other kinds of implicit information to make recommendations are not listed, considering that we focus on recommending items based on tag information.

ProbS [6]: similar to our work, a hybrid mass diffusion based algorithm using both User-Item graph and Item-Tag graph was proposed to fulfill personalized recommendation. Although mass diffusion can also work in multi-steps, we use the default two-steps diffusion in our experiments.

UserCF [7]: In this approach, the tag-based profiles were used to represent users' topic preferences as Eq. 1. The recommendation $rec(u, r_j)$ for a certain item r_j aggregates the votes of all neighbors of u using a similarity-weighting approach as Eq. 11,

$$rec(u, r_j) = \frac{\sum_{u_i \in C(u)} v_i(r_j) R(u, u_i)}{|C(u)|} \tag{11}$$

where, $v_i(r_j)$ is the normalized 'vote' of u_i to r_j. The neighborhood $C(u)$ for u and $R(u, u_i)$ are same as our above-mentioned definition.

Random Walk with Restart (RWR): RWR has recently attracted much attentions in various recommendation scenarios [8,9]. Here, we perform the RWR model on neighborhood-based tripartite graph, and set the personalized vector of the PageRank to bias the node representing the current user.

3.3 Experimental Results

We implemented our model and baseline methods using Java on a computer set to 4 GB memory and 3.1 GHz processors. We run tests extensively to find the optimal parameter settings for two datasets. The settings of the main parameters are shown in Table 2. Also, we set the convergence thresholds of iterative computation as a unified value $\epsilon < 0.001$, and perform experiments to investigate the computational efficiency of the proposed method (RPRW). Depending on Table 2, it takes only 20–40 iterations or several hundred milliseconds for our method to make a recommendation. This suggests our model can meet the

Table 2. The parameter settings for two datasets and the corresponding computational costs, where $|C(u)|$ is the community size to the user u, Iter4UR and Iter4RT are respectively the iteration times of RPRW on G_{UR} and G_{RT}.

Dataset	$\|C(u)\|$	d	λ	Iter4UR	Iter4RT	TimeCost (ms)
Lastfm	20	0.15	[0.6,0.9]	19	33	150
MovieLens	20	0.15	[0.5,0.7]	21	39	30

demands of real-time application. In addition, we find that the best setting of λ in our model is consistent with the ProbS model, since they share the basic principle as combing the relevance score of r_j to u both on G_{UR} and G_{RT}.

We first compare our RPRW model with the baseline methods against the Lastfm dataset. According to Fig. 2, the ProbS performs best from the first position to the 15th position of recommendation list in all performance indicators, however, the RPRW model basically ranks in the first class after the 15th position in term of accuracy. In particular, the RPRW model outperforms all baseline methods in the F1 measure, where it improves the baseline methods by around 7 % from F1@15 to F1@30. We next study the recommendation performance of selected methods based on the Movielens dataset. Similar to the recommendation results achieved on the Lastfm dataset, our RPRW model does best after

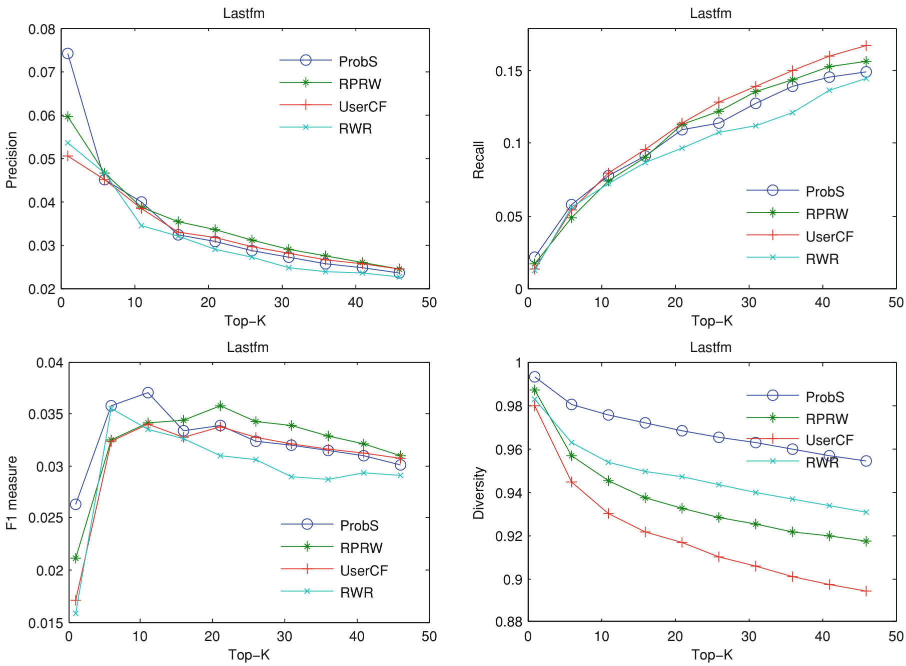

Fig. 2. Performance of recommendation based on the Lastfm dataset.

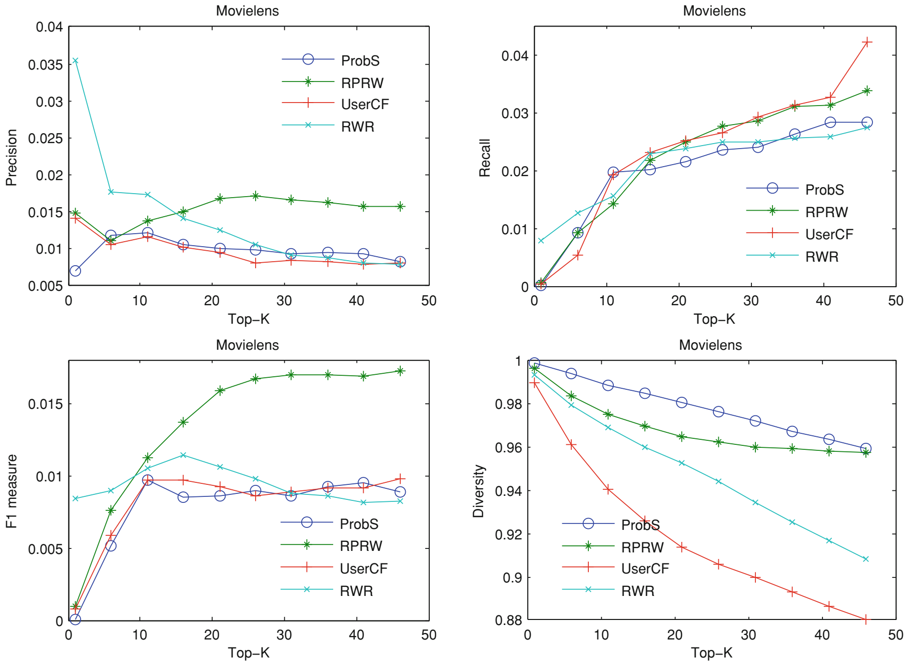

Fig. 3. Performance of recommendation based on the MovieLens dataset.

the 15th position in the recommendation list by accuracy metrics (see Fig. 3). Particularly, it significantly improves the F1 metric compared to all baseline methods. Different from the preceding experimental results where the ProbS achieves superior performance in P@10 and F1@10, instead, the RWR does best in P@10 and F1@10. By both Figs. 2 and 3, the UserCF performs best in the recall while slightly better than the RPRW model.

When examining the diversity of recommendation, the ProbS outperforms all other methods on both datasets. Both the RWR and the RPRW rank at the second place. A major reason caused this is that our model favors to popular items in the user community, i.e., the infinite random walk preferred to those nodes with higher degrees in social tagging graph. Such a situation is always observed in the classic random walk models, such as PageRank [10] and HITS [11]. Recommending commonly popular resources to users can obviously improve the accuracy while degrade the diversity. It is also a presentation of the well-known diversity-accuracy dilemma [9]. However, the RPRW model can still outperform some baseline methods by either the accuracy metric or the diversity metric, and achieve a better balance between the accuracy and the diversity of recommendation. This points to that the RPRW model has its own merits in recommendation situations. To further improve the diversification of recommendation results, the RPRW model can make use of a simple method to discount the popularity of resources [9].

4 Related Works and Discussion

There have been many technical advances in collaborative filtering models [3], topic-based models, and tensor-based models [6] for personalized recommendation. However, these models are distinctly different from our method, so we do not repeat them here. Instead, we concentrate some typical studies in applying random walks on personalized recommendation.

Hotho et al. proposed the FolkRank algorithm [12], an adaptation of the PageRank algorithm to the folksonomy structure. FolkRank performs a weight-spreading ranking scheme on folksonomies. It transforms the hypergraph between the sets of users, tags and resources into an undirected, weighted, tripartite graph. On this graph, it applies a version of PageRank that takes into account the obtained edge weights. Among applications, FolkRank provides a popularity measure of a document that seems to be better than PageRank, as it exploits the user produced folksonomy, rather than the Web links. FolkRank performs well in tag recommendation, however, it does not do well as other models on resource recommendation(for this, we omit the experimental results with respect to the FolkRank). ItemRank [13] proposed by Marco and Augusto, is used to rank products according to expected user preferences. It employs the naive PageRank on item-based graph to rank the item node. Then the PageRank and the preference to the expected user are integrated together to propose products. Similar to ItemRank, Yildirim and Krishnamoorthy [14] proposed a novel recommendation algorithm which performs random walks on a graph that stands for similarity measures between items. They evaluate their system using data from MovieLens. Although, the use of the random walk model performs well for recommendation, their use of an Item-Item similarity matrix raises some issues on the ability of the system to extend when other similarities are introduced based on social tagging. Konstas et al. [8] consider both the social annotation and the friendships inherent in the social graph established among users, items and tags. They adopt the generic framework of the RWR to provide with a more natural and efficient way to represent social networks. Their method is experimented with a self-collected Lastfm dataset and significantly outperforms the collaborative filtering method. However, their method utilizes all the training information to predict resources of interest, and seriously differs with our neighborhood-based training method. Hybrid ProbS [6] is recently introduced to item recommendation using tagging information. It applies respectively two mass diffusions in a user-resource and a resource-tag network to make recommendations. An item's preference is defined as a linear combination (similar to us) of its ranks in the two graphs. However, our approach can outperform this method by trustworthy experiments.

Regardless of the fact that these studies are close to our approach, we create a unique model, in which neighborhood-based method is first used to extract a dense social tagging subgraph, and then user-item preferences are propagated through infinite random walk. This strategy makes our model scalable even facing a huge amount of tagging data. Besides, our model has a better extendability:

– Advanced user/resource profiling methods (e.g. [15]) can be employed to strengthen relevance estimation;

- Explicit relations, such as friendships among users and inter-resources links, can be added to enrich the semantics;
- Tag-aware personalized search ([16]) can also be built on our model by making a neighborhood-based subgraph with an adhoc retrieval model.

5 Conclusion and Future Works

We have presented a relevance propagation model with random walk for a tag-aware personalized recommendation. According to solid experiments, our model performs effectively and efficiently in personalized recommendation, and achieves a better balance between accuracy and diversity metric. In future, we would consider developing advanced profiling methods to further strengthen relevance estimation in our model. Also, extending the model to cope with the cold start problem or tag-aware personalized search would also be interested.

Acknowledgments. This work is supported by the Applied Basic Research Project of Yunnan Province(2013FB009).

References

1. Golder, S.A., Huberman, B.A.: Usage patterns of collaborative tagging systems. J. Inf. Sci. **32**(2), 198–208 (2006)
2. Gupta, M., Li, R., Yin, Z., Han, J.: Survey on social tagging techniques. ACM SIGKDD Explor. **12**(1), 58–72 (2010)
3. Su, X., Khoshgoftaar, T.M.: A survey of collaborative filtering techniques. Adv. Artif. Intell. **2009**(4), 1–1 (2009)
4. Cantador, I., Brusilovsky, P., Kuflik, T.: 2nd workshop on information heterogeneity and fusion in recommender systems (hetrec 2011). In: Proceedings of the 5th ACM Conference on Recommender Systems. RecSys 2011, New York, NY, USA, ACM (2011)
5. Bellogin, A., Cantador, I., Castells, P.: A comparative study of heterogeneous item recommendations in social systems. Inf. Sci. **221**(1), 142–169 (2013)
6. Zhang, Z.K., Zhou, T., Zhang, Y.C.: Tag-aware recommender systems: a state-of-the-art survey. J. Comput. Sci. Technol. **26**, 767–777 (2011)
7. Diederich, J., Iofciu, T.: Finding communities of practice from user profiles based on folksonomies. In: Tomadaki, E., Scott, P. (eds.) EC-TEL 2006 Workshops Proceedings Innovative Approaches for Learning and Knowledge Sharing, pp. 288–297 (2006)
8. Konstas, I., Stathopoulos, V., Jose, J.M.: On social networks and collaborative recommendation. In: Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 195–202. ACM (2009)
9. Wu, H., Cui, X., He, J., Li, B., Pei, Y.: On improving aggregate recommendation diversity and novelty in folksonomy-based social systems. Submission to Personal and Ubiquitous Computing (2014).
10. Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. Comput. Netw. ISDN Syst. **30**(1–7), 107–117 (1998)

11. Kleinberg, J.M.: Authoritative sources in a hyperlinked environment. J. ACM **46**(5), 604–632 (1999)
12. Hotho, A., Jäschke, R., Schmitz, Ch., Stumme, G.: Information retrieval in folksonomies: search and ranking. In: Sure, Y., Domingue, J. (eds.) ESWC 2006. LNCS, vol. 4011, pp. 411–426. Springer, Heidelberg (2006)
13. Gori, M., Pucci, A.: Itemrank: a random-walk based scoring algorithm for recommender engines. In: Proceedings of the 20th International Joint Conference on Artificial Intelligence, pp. 2766–2771 (2007)
14. Yildirim, H., Krishnamoorthy, M.S.: A random walk method for alleviating the sparsity problem in collaborative filtering. In: Proceedings of the 2008 ACM Conference on Recommender Systems, pp. 131–138. ACM (2008)
15. Liang, H., Xu, Y., Li, Y., Nayak, R.: Connecting users and items with weighted tags for personalized item recommendations. In: Proceedings of 21st ACM Conference on HyperText and Hyper-Media, Toronto, pp. 51–60. ACM (2010)
16. Xie, H.R., Li, Q., Cai, Y.: Community-aware resource profiling for personalized search in folksonomy. J. Comput. Sci. Technol. **27**(3), 599–610 (2012)