

# Personalized recommendation via integrated diffusion on user–item–tag tripartite graphs

Zi-Ke Zhang<sup>a</sup>, Tao Zhou<sup>a,b,\*</sup>, Yi-Cheng Zhang<sup>a</sup>

<sup>a</sup> Department of Physics, University of Fribourg, Chemin du Musée 3, 1700 Fribourg, Switzerland

<sup>b</sup> Department of Modern Physics, University of Science and Technology of China, Hefei 230026, PR China

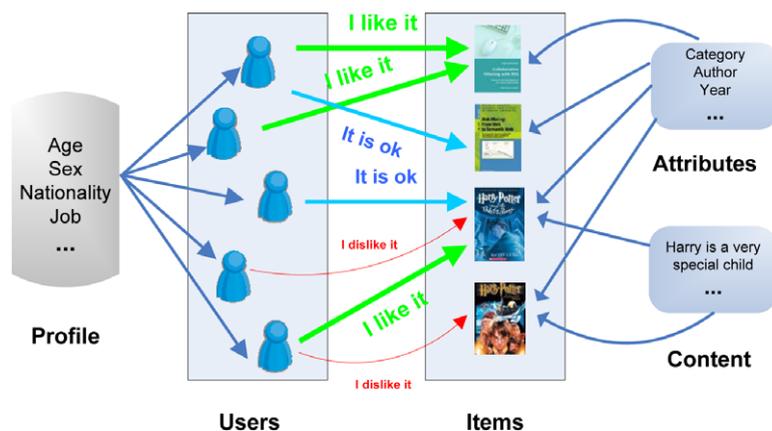
Personalized recommender systems are confronting great challenges of accuracy, diversification and novelty, especially when the data set is sparse and lacks accessorial information, such as user profiles, item attributes and explicit ratings. Collaborative tags contain rich information about personalized preferences and item contents, and are therefore potential to help in providing better recommendations. In this article, we propose a recommendation algorithm based on an integrated diffusion on user–item–tag tripartite graphs. We use three benchmark data sets, *Del.icio.us*, *MovieLens* and *BibSonomy*, to evaluate our algorithm. Experimental results demonstrate that the usage of tag information can significantly improve accuracy, diversification and novelty of recommendations.

## 1. Introduction

The last few years have witnessed an explosion of information that the exponential growth of the Internet [1] and World Wide Web [2] confronts us with an information overload: we face too much data and sources to be able to find out those most relevant for us. Indeed, we have to make choices from thousands of movies, millions of books, billions of web pages, and so on. Evaluating all these alternatives by ourselves is not feasible at all. As a consequence, a pressing problem is how to automatically find out the relevant items for us. Internet search engine [3], with the help of keyword-based queries, is an essential tool in getting what we want from the web. However, the search engine does not take into account personalization and returns the same results for people with far different habits. In addition, not all needs or tastes can be easily presented by keywords. Comparatively, *recommender system* [4], which adopts knowledge discovery techniques to provide personalized recommendations, is now considered to be the most promising way to efficiently filter out the overload of information. Thus far, recommender systems have successfully found applications in e-commerce [5], such as book recommendations in *Amazon.com* [6], movie recommendations in *Netflix.com* [7], video recommendations in *TiVo.com* [8], and so on.

A recommender system is able to automatically provide personalized recommendations based on the historical record of users' activities. These activities are usually represented by the connections in a user–item bipartite graph [9,10]. Fig. 1 illustrates such a graph consisted of five users and four books, where users can give ratings to those books. So far, collaborative filtering (CF) is the most successful technique in the design of recommender systems [11], where a user will be recommended with items that people with similar tastes and preferences liked in the past. Despite its success, the performance of CF is strongly limited by the sparsity of data resulted from: (i) the huge number of items that is far beyond user's ability to evaluate even a small fraction of them; (ii) users do not incentively wish to rate the purchased/viewed items [12]. As shown in Fig. 1, besides the fundamental user–item relations, some accessorial information can be exploited

\* Corresponding author at: Department of Physics, University of Fribourg, Chemin du Musée 3, 1700 Fribourg, Switzerland.  
E-mail address: zhutou@ustc.edu (T. Zhou).



**Fig. 1.** (Color online) Illustration of a recommender system consisted of five users and four books. The basic information contained by every recommender system is the relations between users and items that can be represented by a bipartite graph. This illustration also exhibits some additional information that are frequently exploited in the design of recommendation algorithms, including user profiles, item attributes and item content.

to improve the algorithmic accuracy [13]. User profiles, usually including age, sex, nationality, job, etc., can be treated as prior known information to filter out possibly irrelevant recommendations [14], however, the applications are mostly forbidden or strongly restricted to respect personal privacy. Attribute-aware method [15] takes into account item attributes, which are defined by domain experts. Yet, it is limited to the attribute vocabulary, and, on the other hand, attributes describe global properties of items which are essentially not helpful to generate personalized recommendations. In addition, content-based algorithms can provide very accurate recommendations [16], however, they are only effective if the items contain rich content of information that can be automatically extracted out. For example, these methods are suitable for recommending books and articles, but not for videos or pictures.

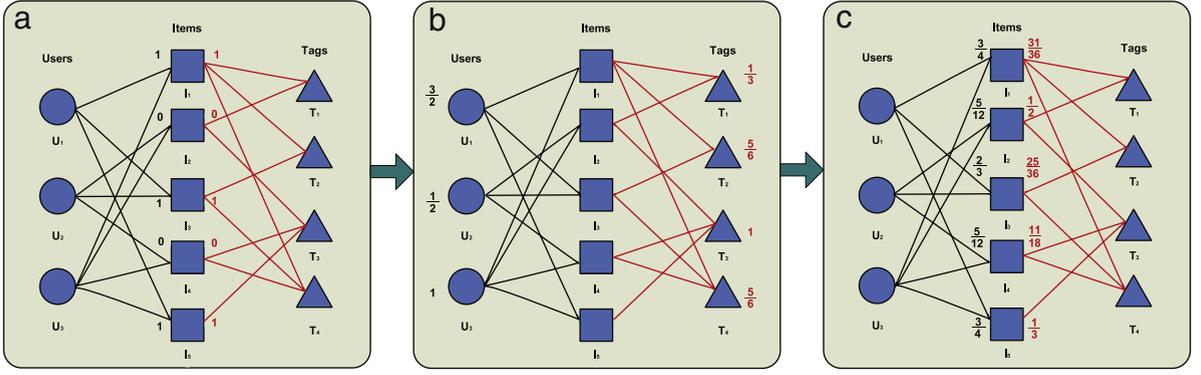
Collaborative tagging systems (CTSeS), allowing users to freely assign tags to their collections, provide promising possibility to better address the above issues. CTSeS require no specific skills for user to participate, thus can overcome the limitation of vocabulary domains and size, widen the semantic relations among items and eventually facilitate the emergence of *folksonomy* [17].

The CTSeS have recently attracted much attention from scientific communities. Firstly, many efforts have been addressed in understanding the structure and evolution of CTSeS [18,19], as well as the usage patterns in folksonomies [20]. For example, Cattuto et al. [18] investigated the evolution of an open-ended system *Del.icio.us*, while Zhang et al. [19] empirically studied the semiotic structure of a number of seriously edited scientific journals. An interesting result is that both the open-ended and seriously edited systems share many common features [18,19].

Secondly, some studies concern the applications of CTSeS. A considerable number of algorithms are designed to recommend tags to users, which may be helpful for better organizing, discovering and retrieving items [17,21,22]. The current work focuses on a relevant yet different application of CTSeS, that is, to provide personalized item recommendations with the help of tag information. Schenkel et al. [23] proposed an incremental threshold algorithm taking into account both the social ties among users and semantic relatedness of different tags, which performs remarkably better than the algorithm without tag expansion. Nakamoto et al. [24] created a tag-based contextual CF model, where the tag information is treated as the users' profiles. Tso-Sutter et al. [25] proposed a generic method that allows tags to be incorporated to the standard CF, via reducing the ternary correlations to three binary correlations and then applying a fusion method to re-associate these correlations.

In this article, instead of the well-known CF, we will apply a diffusion-based recommendation algorithm on user-item-tag tripartite graphs. Huang et al. [9] proposed a *spreading activation-based CF*, which is essentially an iterative diffusion process. This algorithm can provide relatively accurate recommendations for very sparse systems that are hard to be managed by the standard CF. Zhou et al. [10] proposed an extremely fast algorithm considering only a two-step diffusion in user-item bipartite networks, which can still give slightly more accurate recommendations than the standard CF. Further more, some refinement taking into account the initial difference of activations for different items can simultaneously enhance the recommendation accuracy and diversification [26]. Zhang et al. [27] proposed an iterative opinion diffusion algorithm to predict ratings in *Netflix.com*. All the above diffusion-based algorithms obey the *conservation law*, in contrast to which Zhang et al. [28] proposed a non-conservation diffusion-based recommendation algorithm mimicking the heat conduction in networks, which is very efficient to dig out the unpopular yet relevant items. Song et al. [29] proposed a so-called *DiffusionRank* algorithm, where the prediction score is given by the the likelihood that information can propagate from a given user to a given item within a certain time period. Liu et al. [30] proposed a *diffusion-based top-k collaborative filtering*, which performs better than the pure top-k collaborative filtering and pure diffusion-based algorithm.

This article makes the following three contributions: (i) We propose an integrated diffusion-based algorithm making use of both the user-item relations and the collaborative tagging information. (ii) We test our algorithm in three real



**Fig. 2.** (Color online) Illustration of the integrated diffusion process on a tripartite graph consisted of three users, five items and four tags. Plot (a) shows the initial condition given  $U_1$  as the target user and plot (b) describes the result after first-step diffusion, during which the resources are transferred from items to users and tags. Eventually, the resources flow back to items, the result of which is shown in plot (c).

date sets: *Delicio.us*, *MovieLens* and *BibSonomy*. Experimental results show that our algorithm can provide more accurate recommendations than the diffusion-based algorithm without tag information. (iii) Experimental results indicate that diversification and novelty of recommendations are significantly enhanced by exploiting the tag information.

The rest of this article is organized as follows. In Section 2, we introduce the diffusion process on bipartite graphs, as well as our proposed integrated algorithm on tripartite graphs. Section 3 describes the data sets and reports the experimental results, including the area under the receiver operating characteristic (ROC) curve [31], Recall [11], diversification [26] and novelty [26]. Finally, we summarize this paper and outline some open issues for future research in Section 4.

## 2. Method

We start by introducing a diffusion-based recommendation algorithm on bipartite graphs, and then show the integrated algorithm on tripartite graphs. A simple example will be given later to help the readers well in truly understanding the algorithmic procedure. A recommender system considered in this paper consists of three sets, respectively, of users  $U = \{U_1, U_2, \dots, U_n\}$ , items  $I = \{I_1, I_2, \dots, I_m\}$ , and tags  $T = \{T_1, T_2, \dots, T_r\}$ . The tripartite graph representation can be described by two adjacent matrices,  $A$  and  $A'$ , for user-item and item-tag relations. If  $U_j$  has collected  $I_j$ , we set  $a_{ij} = 1$ , otherwise  $a_{ij} = 0$ . Analogously, we set  $a'_{jk} = 1$  if  $I_j$  has been assigned by the tag  $T_k$ , and  $a'_{jk} = 0$  otherwise. Fig. 2 shows an illustration consisted of three users, five items and four tags.

### 2.1. Diffusion on bipartite graphs

Considering a bipartite graph  $G(U, I, E)$ , where  $U$  and  $I$  are user set and item set, and  $E$  is the set of edges connecting users and items. Supposing that a kind of resource is initially located on items, each item will averagely distribute its resource to all neighboring users, and then each user will redistribute the received resource to all his/her collected items. Denoting  $\vec{f}$  the initial resource vector on items (i.e.,  $f_j$  is the amount of resource located on  $I_j$ ), then the final resource vector,  $\vec{f}'$  after the two-step diffusion is:

$$f'_j = \sum_{i=1}^n \frac{a_{ij}}{k(U_i)} \sum_{s=1}^m \frac{f_s}{k(I_s)}, \quad j = 1, 2, \dots, m, \quad (1)$$

where  $k(U_i) = \sum_{j=1}^m a_{ij}$  is the number of collected items for user  $U_i$ , and  $k(I_s) = \sum_{i=1}^n a_{is}$  is the number of neighboring users for item  $I_s$ .

Given a target user  $U_i$ , we set the initial resource vector,  $\vec{f}(U_i)$ , as

$$f_j(U_i) = a_{ij}, \quad j = 1, 2, \dots, m. \quad (2)$$

In this case, the initial resource can be understood as giving a unit recommending capacity to each collected item, and the different initial resource vectors for different users have captured the personalized preferences. The final resource vector,  $\vec{f}'(U_i)$ , is obtained by Eq. (1). All  $U_i$ 's uncollected items are sorted in the descending order of the final resource, and those items with highest values of resource are recommended. This algorithm is originally motivated by the resource-allocation process on graphs [32], and has been shown to be slightly more accurate than the collaborative filtering based on *MovieLens* data [10].

### 2.2. Integrated diffusion on tripartite graphs

As mentioned in Section 2.1, the different collections of items reflect personalized preferences. Further more, even for exactly the same items, different users may assign far different tags to them. Therefore, the significance of collaborative tags

**Table 1**

Basic statistics of three data sets.  $n$ ,  $m$ ,  $r$  are the total numbers of users, items and tags, respectively.  $\langle k_u \rangle$  is the average number of items collected by a user,  $\langle k_i \rangle$  denotes the average number of users having collected an item,  $\langle k'_i \rangle$  refers to the average number of tags assigned to an item, and  $\langle k_t \rangle$  is the average number of items relevant to a tag. In the last column, we report the average running time for a given  $\lambda$ . Here, the experiments are implemented in a desktop computer under Linux OS (CPU: Intel Pentium 4, 3 GHz; Memory: 1 GB).

Data set	$n$	$m$	$r$	$\langle k_u \rangle$	$\langle k_i \rangle$	$\langle k'_i \rangle$	$\langle k_t \rangle$	Time (s)
<i>Delicio.us</i>	11 999	285 804	119 392	132.91	5.58	9.46	22.65	97.20
<i>MovieLens</i>	3 549	6 054	5 828	14.41	8.45	10.00	10.39	26.84
<i>BibSonomy</i>	794	11 211	5 986	38.69	2.74	7.91	14.81	6.74

in a recommender system is twofold. Firstly, tags enrich the item information and two items sharing many common tags are probably closely related in content. Secondly, the personalized preferences are naturally embedded in the different usages of tags. In this subsection, we introduce a simple way that utilizes the tag information to provide better recommendations.

We first consider the diffusion on an item–tag bipartite graph. Analogously, supposing that a kind of resource is initially located on items, each item will equally distribute its resource to all neighboring tags, and then each tag will redistribute the received resource to all its neighboring items. Thus, given the initial resource vector,  $\vec{f}$ , the final resource vector,  $\vec{f}''$ , is:

$$f_j'' = \sum_{l=1}^r \frac{a_{jl}'}{k(T_l)} \sum_{s=1}^m \frac{f_s}{k'(I_s)}, \quad j = 1, 2, \dots, m, \quad (3)$$

where  $k(T_l) = \sum_{j=1}^m a_{jl}'$  is the number of neighboring items for tag  $T_l$  and  $k'(I_s) = \sum_{l=1}^r a_{sl}'$  is the number of neighboring tags for item  $I_s$ .

Given a target user  $U_i$ , the initial resource is also set according to Eq. (2). As a start point, in this article, we adopt the simplest way to integrate the diffusions on user–item and item–tag bipartite graphs, that is, to define the final resource as a linear superposition of  $\vec{f}'$  and  $\vec{f}''$ :

$$\vec{f}^* = \lambda \vec{f}' + (1 - \lambda) \vec{f}'', \quad (4)$$

where  $\lambda \in [0, 1]$  is a tunable parameter,  $\vec{f}'$  is the vector obtained from Eq. (1),  $\vec{f}''$  is the vector derived from Eq. (3), and both of the two vectors are attained for the same target user. In the extremal cases  $\lambda = 0$  and  $\lambda = 1$ , the integrated algorithm degenerates to the pure diffusions on item–tag and user–item bipartite graphs, respectively.

Given the parameter  $\lambda$ , ignoring the degree–degree correlation in user–item and item–tag relations, the algorithmic complexity is  $\mathcal{O}(n \langle k_u \rangle (\langle k_u \rangle \langle k_i \rangle + \langle k'_i \rangle \langle k_t \rangle))$ , where  $\langle k_u \rangle$  is the average number of items collected by a user,  $\langle k_i \rangle$  denotes the average number of users having collected an item,  $\langle k'_i \rangle$  refers to the average number of tags assigned to an item, and  $\langle k_t \rangle$  is the average number of items relevant to a tag. Since the real networks are usually sparse, the complexity is mainly determined by the number of users,  $n$ . Approximately, the proposed algorithm has a linear complexity versus the number of users, which is very efficient and practical. The computational time for real networks are reported in Table 1.

### 2.3. Example

In this subsection, we give a simple example to concretely describe the above integrated diffusion process. As shown in Fig. 2,  $U_1$  is chosen to be the target user. Firstly, we highlight all the items collected by  $U_1$  and assign each of them a unit resource. Thus the initial vector, as shown in Fig. 2(a), is:

$$\vec{f} = (1, 0, 1, 0, 1). \quad (5)$$

Secondly, the initial resource will be transferred to users and tags (treated as two independent processes), resulting in two medi-distributions of resources on users and tags, as shown in Fig. 2(b). Finally, the resources located on users and tags will be transferred back to items. As shown in Fig. 2(c), the final resource vectors, corresponding to Eqs. (1) and (3), are:

$$\vec{f}' = (3/4, 5/12, 2/3, 5/12, 3/4), \quad \vec{f}'' = (31/36, 1/2, 25/36, 11/36, 1/3). \quad (6)$$

Given  $\lambda$ , the recommendation scores on items of  $U_1$  can be obtained by Eq. (4). The items with highest scores are then recommended to  $U_1$ .

## 3. Experiments

### 3.1. Data sets

In this article, three representative data sets, *Delicio.us*,<sup>1</sup> *MovieLens*<sup>2</sup> and *BibSonomy*,<sup>3</sup> are used to evaluate our proposed algorithm. *Delicio.us* is one of the most popular social bookmarking web sites, which allows users not only to store and

<sup>1</sup> <http://delicio.us/>.

<sup>2</sup> <http://www.movielens.org/>.

<sup>3</sup> <http://www.bibsonomy.com/>.

organize personal bookmarks (URLs), but also to look into other users' collections and find what they might be interested in by simply keeping track of the baskets with tags or items. The data used in this paper are a random sampling of about 12 000 uses in May 2008. *MovieLens* is a movie rating system, where each user votes movies in five discrete ratings 1–5. A tagging function is added in from January 2006. *BibSonomy* is a collaborative tagging system mixing with social bookmarks and scientific publications [33], and we only choose the data of bookmarks. In all the three data sets, we remove the isolated nodes and guarantee that each user has collected at least one item, each item has been collected by at least two users, and assigned by at least one tag. *MovieLens* and *BibSonomy* allow users freely to assign tags without any grammatical restrictions, resulting in many strange and/or extremely long tags, some of which are phrases, and others may contain rarely used symbols. Therefore, for *MovieLens* and *BibSonomy*, the tags appearing only once are removed. Table 1 summarizes the basic statistics of the three purified data sets.

### 3.2. Metrics for algorithmic performance

Every data set is consisted of many entries, each of which follows the form { user, item, tag<sub>1</sub>, tag<sub>2</sub>, . . . , tag<sub>h</sub>}, where  $h$  is the number of tags assigned to the relevant item by the very user. To test the algorithmic performance, each data set is randomly divided into two parts: The training set contains 95% of entries and the remaining 5% of entries constitutes the testing set. The training set is treated as known information used for generating recommendations, while no information in the testing set is allowed to be used for recommendations. To give solid and comprehensive evaluation of the proposed algorithm, we employ four different metrics that characterize not only the accuracy of recommendations, but also the diversification and novelty.

- (1) *The area under the ROC curve* [31]—In the present case, the area under the ROC curve, abbreviated as AUC, for a particular user is the probability that a randomly selected removed item for this user (i.e., an item in the testing set and being collected by this user) is given a higher score by our algorithm than a randomly selected uncollected item (i.e, an item irrelevant to this user in neither the training set nor the testing set). The AUC for the whole system is the average over all users. If all the scores are generated from an independent and identical distribution,  $AUC \approx 0.5$ . Therefore, the degree to which the AUC exceeds 0.5 indicates how better the algorithm performs than pure chance.
- (2) *Recall* [11]—Note that, the AUC takes into account the order of all uncollected items, however, in the real applications, user might only care about the recommended items, that is, the items with the highest scores. Therefore, as a complementary measure, recall is employed to quantify the accuracy of recommended items, which is defined as:

$$\text{Recall} = \frac{1}{n} \sum_{i=1}^n N_r^i / N_p^i, \quad (7)$$

where  $N_p^i$  is the number of items collected by  $U_i$  in the testing set and  $N_r^i$  is the number of recovered items in the recommendations for  $U_i$ . We use the averaged recall instead of simply counting  $N_r/N_p$  with  $N_r = \sum_i N_r^i$  and  $N_p = \sum_i N_p^i$ , since it is fair to provide equal weightage for every user in the evaluation of algorithm. Assuming the length of recommendation list,  $L$ , is fixed for every user, recall is very sensitive to  $L$  and a larger  $L$  generally gives a higher recall.

- (3) *Diversification* [26]—This measure considers the uniqueness of different users' recommendation lists, thus can be understood as the inter-user diversity.<sup>4</sup> Denote  $I_R^i$  the set of recommended items for user  $U_i$ , then

$$\text{Diversification} = \frac{2}{n(n-1)} \sum_{i \neq j} \left( 1 - \frac{|I_R^i \cap I_R^j|}{L} \right), \quad (8)$$

where  $L = |I_R^i|$  for any  $i$  is the length of recommendation list. Greater or lesser values of diversification mean, respectively, greater or lesser personalization of users' recommendation lists.

- (4) *Novelty* [26]—This measure quantifies the capacity of an algorithm to generate novel and unexpected results, that is to say, to recommend less popular items unlikely to be already known about. The simplest way is to use the average collected times over all the recommended items, as:

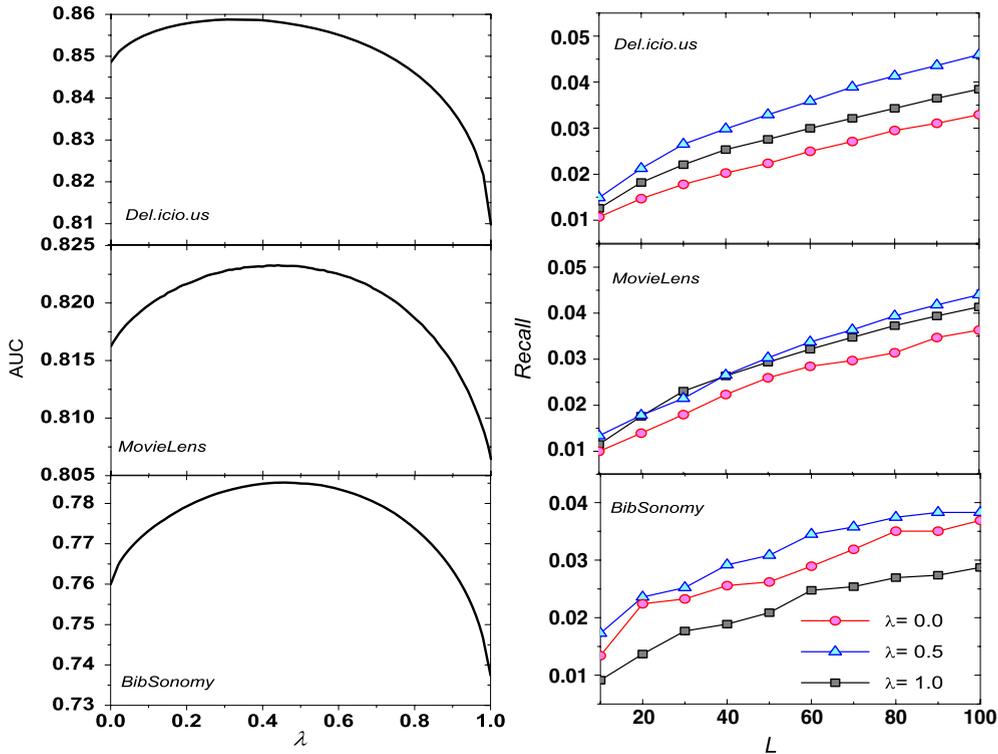
$$\text{Novelty} = \frac{1}{nL} \sum_{i=1}^n \sum_{I_r \in I_R^i} k(I_r), \quad (9)$$

<sup>4</sup> To evaluate the diversification of recommendations, Ziegler et al. [34] proposed a metric for intra-list diversity, which requires a similarity measure for item pairs in the recommendation list. This similarity can be well defined with the help of item attributes or item contents. However, this information is not available in general. Of course, one can define a kind of similarity between two items by counting the number of users collected both of them and/or the number of tags shared by both of them. However, how to weight the contributions to similarity from the user-item and item-tag relations is a problem, and the similarity measure itself has latent correlation with the algorithm (for that they both have used the same information) that may lead to systematic bias. We, therefore, employ the inter-user diversification, which is simple and meaningful for any kind of data.

**Table 2**

Comparison of algorithmic accuracy, measured by the AUC. *Pure U-I* and *Pure I-T* denote the pure diffusions on user–item and item–tag bipartite graphs, respectively, corresponding to  $\lambda = 1$  and  $\lambda = 0$ . The optimal values of  $\lambda$  as well as the corresponding optima of AUC are presented for comparison.

Data set	Pure U-I	Pure I-T	Optimum	$\lambda_{opt}$
<i>Del.icio.us</i>	0.8098	0.8486	0.8588	0.32
<i>MovieLens</i>	0.8065	0.8163	0.8233	0.44
<i>BibSonomy</i>	0.7374	0.7600	0.7852	0.44



**Fig. 3.** (Color online) AUC versus  $\lambda$  (left) and Recall versus the length of recommendation list for  $\lambda = 0$ ,  $\lambda = 0.5$  and  $\lambda = 1.0$  (right). The results reported here are averaged over 50 independent runs, each of which corresponds to a random division of training set and testing set.

with  $k(I_r)$  the number of users having collected the item  $I_r$  (see also previous definition in Eq. (1)). Readers are warned that the less value obtained by Eq. (9) actually means higher novelty and surprisal.<sup>5</sup>

### 3.3. Experimental results

Fig. 3 shows the AUC statistics of the three data sets for  $\lambda \in [0, 1]$ . It can be seen that all the three curves have qualitatively the same trend: (i) Pure diffusion on item–tag bipartite graphs ( $\lambda = 0$ ) performs better than on user–item bipartite graphs ( $\lambda = 1$ ) and (ii) integrated algorithm can provide more accurate recommendations than the pure cases. In Table 2, we compare the AUC for pure diffusions on bipartite graphs and for the optimum by integrated algorithm. Comparing with the algorithm without tag information [10], at the optimal values, the improvements for *Del.icio.us*, *MovieLens* and *BibSonomy* are 6.1%, 2.1% and 6.5%, respectively. Since AUC is not a very sensitive index [35,36], 6% is indeed a remarkable improvement.

We also report the values of recall in Fig. 2. Since the typical length for recommendation list is tens, our experimental study focuses on the interval  $L \in [10, 100]$ . For better presentation, we only show two pure diffusions with  $\lambda = 0$  and  $\lambda = 1$ , as well as a naive integration with  $\lambda = 0.5$ . Different from the AUC statistics, the diffusion on item–tag bipartite graphs does not always perform better than the one on user–item bipartite graphs. However, the integrated algorithm can always beat the two pure algorithms in certain ranges of  $\lambda$ , which again indicates the advantage of integration.

Fig. 4 reports the diversification and novelty of recommendations versus the parameter  $\lambda$ , respectively. All curves are monotone, and the algorithm depending more on tags can provide more personalized and higher novel recommendations. That is to say, the collaborative tagging systems can be used to simultaneously enhance the serendipitous discovering of individuals and the diversity among individuals.

<sup>5</sup> An alternative is to use  $k^{-1}(I_r)$  instead of  $k(I_r)$  in Eq. (9), then the larger value corresponds to higher novelty. We employ the current definition for the advantage to directly exhibit the average collected times of recommended items.

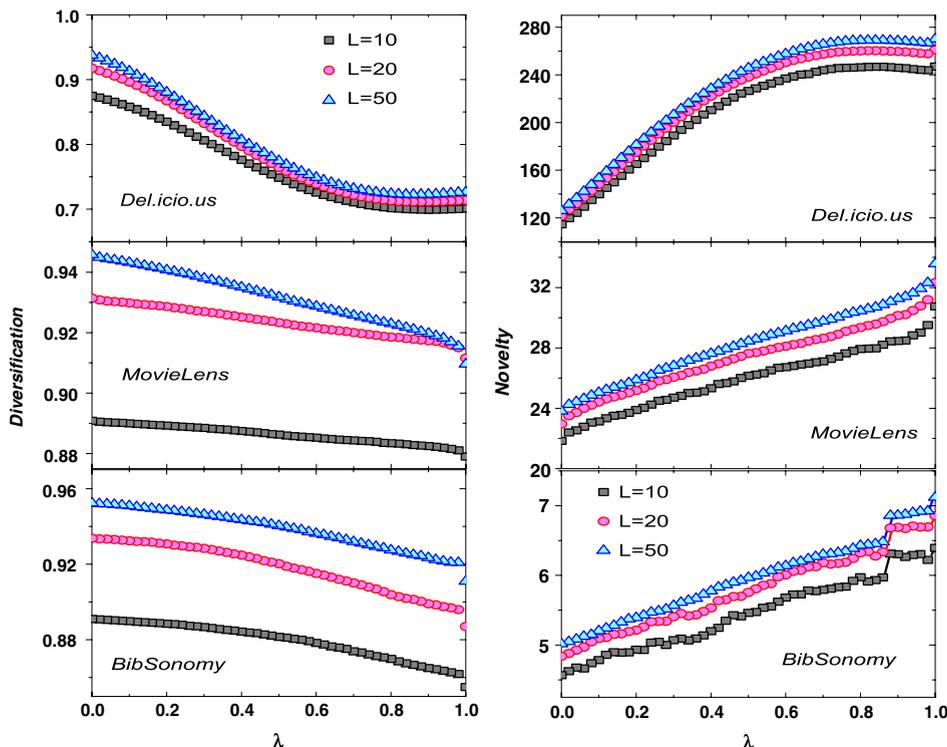


Fig. 4. (Color online) *Diversification* versus  $\lambda$  (left) and *Novelty* versus  $\lambda$  (right) for typical recommendation list lengths:  $L = 10$ ,  $L = 20$  and  $L = 50$ .

#### 4. Conclusion and discussion

In this article, we proposed an integrated diffusion-based algorithm with the help of collaborative tagging information. Experimental results demonstrate that the usage of tag information can significantly improve accuracy, diversification and novelty of recommendations. It is worthwhile to emphasize the twofold significance of the collaborative tags, as an assistant part of a recommender system. Firstly, tags assigned to a certain item can be considered as the highly abstracted content of that item. Secondly, even for the same item, different users may assign completely different tags, and thus the personalized preferences are naturally embedded in the different usages of tags. A user assigns two tags, *action* and *Nicolas Cage*, to the movie *Lord of War* may get a recommendation *Gone in Sixty Seconds*, while another user sticks *drama film* and *peace* onto the same movie may be recommended *Crash*.

The CTSES play more and more important role in the Internet world, and we must be aware of their significance. Experimental results in this article strongly suggest using the tag information to improve the quality of recommendations. Indeed, we should do even beyond, that is, for the already existing recommender systems, to add tagging functions into them and encourage users to organize their collections by using tags.

This article only provides a simple start point for the design of hybrid algorithms making use of tag information, while a couple of issues remain open for future study. First, we lack quantitative understanding of the structure and evolution of CTSES as well as the performance of folksonomy. Although the relation between folksonomy and recommender systems is not clear thus far, we deem that an in-depth understanding of tagging systems should be helpful for better recommendations. Second, the current algorithm focuses on the unweighted graphs, and a refined algorithm with properly defined weights on user-item and item-tag relations may further improve the algorithmic performance. Third, it is significant to design an online algorithm that can give real-time response to the user activities, such as selecting of new items and changing of tags. Finally, the tag information can also be exploited under the frameworks of collaborative filtering [11] and iterative diffusion algorithm [9], as well as for some more complicated methods such as *Probabilistic Latent Semantic Analysis* [37], *Latent Dirichlet Allocation* [38] and *Iterative Latent Semantic Analysis* [39]. Systematic investigation on tag-aware recommendation algorithms must be very helpful in the future design of recommender systems.

#### Acknowledgements

We acknowledge the *GroupLens Research Group* for *MovieLens* data, and the *Knowledge and Data Engineering Group* in the *University of Kassel* for *BibSonomy* data. This work is partially supported by the Swiss National Science Foundation (Project 205120-113842 and 200020-121848). T.Z. acknowledges the National Natural Science Foundation of China under Grant Nos. 60744003 and 10635040. Z.-K.Z. acknowledges the National Natural Science Foundation of China under Grant No. 60973069.

## References

- [1] G.-Q. Zhang, G.-Q. Zhang, Q.-F. Yang, S.-Q. Cheng, T. Zhou, *New J. Phys.* 10 (2008) 123027.
- [2] D. Fensel, J. Hendler, H. Lieberman, *Spinning the Semantic Web: Bringing the World Wide Web to its Full Potential*, Cambridge University Press, US, 2003.
- [3] S. Brin, L. Page, *Proc. 7th Intl. Conf. WWW*, 1998, p. 107.
- [4] P. Resnick, H.R. Varian, *Comm. ACM* 40 (1997) 56.
- [5] J.B. Schafer, J. Konstan, J. Riedi, *IEEE Internet Comput.* 5 (2001) 115.
- [6] G. Linden, B. Smith, J. York, *IEEE Internet Comput.* 7 (2003) 76.
- [7] J. Bennett, S. Lanning, *Proc. KDD Cup Workshop 2007*.
- [8] K. Ali, W. van Stam, *Proc. 10th ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, 2008, p. 394.
- [9] Z. Huang, H. Chen, D. Zeng, *ACM Trans. Inf. Syst.* 22 (2004) 116.
- [10] T. Zhou, J. Ren, M. Medo, Y.-C. Zhang, *Phys. Rev. E* 76 (2007) 046115.
- [11] J.L. Herlocker, J.A. Konstan, L.G. Terveen, J.T. Riedl, *ACM Trans. Inf. Syst.* 22 (2004) 5.
- [12] P. Resnick, K. Kuwabara, R. Zeckhauser, E. Friedman, *Comm. ACM.* 43 (2000) 45.
- [13] G. Adomavicius, A. Tuzhilin, *IEEE Trans. Knowl. Data Eng.* 17 (2005) 734.
- [14] P. Kazienko, M. Adamski, *Inform. Sci.* 177 (2007) 2269.
- [15] K. Tso, L. Schmidt-Thieme, *Proc. 29th Annual Conf. German Classification Society*, Magdeburg, Germany, 2005.
- [16] M.J. Pazzani, D. Billsus, *Lecture Notes in Comput. Sci.* 4321 (2007) 325.
- [17] A. Hotho, R. Jäschke, C. Schmitz, G. Stumme, *Lecture Notes in Comput. Sci.* 4011 (2006) 123027.
- [18] C. Cattuto, V. Loreto, L. Pietronero, *Proc. Natl. Acad. Sci. USA* 104 (2007) 1461.
- [19] Z.-K. Zhang, L. Lü, J.-G. Liu, T. Zhou, *Eur. Phys. J. B* 66 (2008) 557.
- [20] S.A. Golder, B.A. Huberman, *J. Inf. Sci.* 32 (2006) 198.
- [21] G. Mishne, *Proc. 15th Intl. Conf. WWW*, 2006, p. 953.
- [22] B. Sigurbjörnsson, R. Van Zwol, *Proc. 17th Intl. Conf. WWW*, 2008, p. 327.
- [23] R. Schenkel, T. Crecelius, M. Kacimi, S. Michel, T. Neumann, J.X. Parreira, G. Weikum, *Proc. 31st Annual Intl. ACM SIGIR Conf. Res. Dev. Inf. Retr.*, 2008, p. 523.
- [24] R.Y. Nakamoto, S. Nakajima, J. Miyazaki, S. Uemura, *IAENG Int. J. Comput. Sci.* 34 (2007) 214.
- [25] K.H.L. Tso-Sutter, L.B. Marinho, L. Schmidt-Thieme, *Proc. 2008 ACM Symp. Applied Comput.*, 2008, p. 1995.
- [26] T. Zhou, L.-L. Jiang, R.-Q. Su, Y.-C. Zhang, *Europhys. Lett.* 81 (2008) 58004.
- [27] Y.-C. Zhang, M. Medo, J. Ren, T. Zhou, T. Li, F. Yang, *Europhys. Lett.* 80 (2007) 68003.
- [28] Y.-C. Zhang, M. Blattner, Y.-K. Yu, *Phys. Rev. Lett.* 99 (2007) 154301.
- [29] X.D. Song, Y. Chi, K. Hino, B.L. Tseng, *Proc. 16th Intl. Conf. WWW*, 2007, p. 191.
- [30] J.G. Liu, B.H. Wang, Q. Guo, *Internat. J. Modern Phys. C* 20 (2009) 285.
- [31] J.A. Hanley, B.J. McNeil, *Radiology* 143 (1982) 29.
- [32] Q. Ou, Y.-D. Jin, T. Zhou, B.-H. Wang, B.-Q. Yin, *Phys. Rev. E* 75 (2007) 021102.
- [33] A. Hotho, R. Jäschke, C. Schmitz, G. Stumme, *Proc. Conceptual Structures Tool Interoperability Workshop 14th Intl. Conf. Conceptual Structures*, 2006, p. 87.
- [34] C.N. Ziegler, S.M. McNee, J.A. Konstan, G. Lausen, *Proc. 14th Intl. Conf. WWW*, 2005, p. 22.
- [35] A. Clauset, C. Moore, M.E.J. Newman, *Nature* 453 (2008) 98.
- [36] T. Zhou, L. Lü, Y.-C. Zhang, *Eur. Phys. J. B* (in press).
- [37] T. Hofmann, *Proc. 22nd Annual Intl ACM SIGIR Conf. Res. Dev. Inf. Retr.*, 1999, p. 50.
- [38] D.M. Blei, A.Y. Ng, M.I. Jordan, *J. Mach. Learn. Res.* 3 (2003) 993.
- [39] J. Ren, T. Zhou, Y.-C. Zhang, *Europhys. Lett.* 82 (2008) 58007.