Gan MX, Sun L, Jiang R. Trinity: Walking on a user-object-tag heterogeneous network for personalised recommendations. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 31(3): 577–594 May 2016. DOI 10.1007/s11390-016-1648-0

Trinity: Walking on a User-Object-Tag Heterogeneous Network for Personalised Recommendations

Ming-Xin Gan [1,2,*], Lily Sun [3], and Rui Jiang [4,*]

[1] *Department of Management Science and Engineering, Donlinks School of Economics and Management University of Science and Technology Beijing, Beijing 100083, China*

[2] *Department of Statistics, University of California Berkeley, Berkeley, CA 94720, U.S.A.*

[3] *School of Systems Engineering, University of Reading, Reading, RG6 6UR, U.K.*

[4] *Department of Automation, Tsinghua University, Beijing 100084, China*

E-mail: ganmx@ustb.edu.cn; lily.sun@reading.ac.uk; ruijiang@tsinghua.edu.cn

Received November 15, 2014; revised January 29, 2016.

Abstract The rapid evolution of the Internet has been appealing for effective recommender systems to pinpoint useful information from online resources. Although historical rating data has been widely used as the most important information in recommendation methods, recent advancements have been demonstrating the improvement in recommendation performance with the incorporation of tag information. Furthermore, the availability of tag annotations has been well addressed by such fruitful online social tagging applications as CiteULike, MovieLens and BibSonomy, which allow users to express their preferences, upload resources and assign their own tags. Nevertheless, most existing tag-aware recommendation approaches model relationships among users, objects and tags using a tripartite graph, and hence overlook relationships within the same types of nodes. To overcome this limitation, we propose a novel approach, Trinity, to integrate historical data and tag information towards personalised recommendation. Trinity constructs a three-layered object-user-tag network that considers not only interconnections between different types of nodes but also relationships within the same types of nodes. Based on this heterogeneous network, Trinity adopts a random walk with restart model to assign the strength of associations to candidate objects, thereby providing a means of prioritizing the objects for a query user. We validate our approach via a series of large-scale 10-fold cross-validation experiments and evaluate its performance using three comprehensive criteria. Results show that our method outperforms several existing methods, including supervised random walk with restart, simulation of resource allocating processes, and traditional collaborative filtering.

Keywords recommender system, tag-aware recommendation, random walk with restart, heterogeneous network, social tag

1 Introduction

Exponential growth of data in the Internet challenges what data are perceived as useful[1], where the useful data can be discovered from the emerging sources[2-3], and how the data are processed for recognizing personalised interests and improving user experiences. To achieve these goals, various recommender systems have been proposed to provide personalized nomination of resources, with successful examples including online recommendation of books[4], movies[5-6], commodities[7], bookmarks[8], news[9], TV programmes[10], microblogs[11] and many others.

A recommender system is typically designed based on the collaborative filtering strategy. For example, a user-based method works around historical data to calculate discriminant scores of candidate objects from similarities between users[12-13]. An item-based design is rendered formally equivalent to its user-based counterpart by exchanging roles of users and objects[10]. A

Regular Paper

This work was partially supported by the National Natural Science Foundation of China under Grant Nos. 71101010 and 71471016.

*Corresponding Author

more information intensive approach, i.e., the content-based method, uses properties of items to characterize their similarities[14]. To advance these approaches, hybrid methods have also been proposed[15]. Recently, model-based approaches, such as non-negative matrix factorisation[16], singular value decomposition[17] and their variants[18], have been well recognized for their ability to capture hidden relationships between users and objects. Network-based formulations have also been widely adopted to design recommendation algorithms, with the consideration of solving the accuracy and diversity dilemma[5,19-20].

Although historical rating data has been widely used as the most important information in recommendation approaches, there are certainly other types of resources available for characterizing users and objects. For example, online social tagging applications, such as CiteULike, MovieLens, BibSonomy, Folksonomy, Del.icio.us, Flickr and Last.fm[21-22], allow users to easily express their personalized preferences for objects, thereby exposing potential relationships between users and objects. Incorporating such valuable tag information into recommender systems, a variety of tag-aware or social tagging approaches have been proposed[23-24]. Among these approaches, network-based methods have been developed successfully and expanded flexibly[25-34]. For example, network-based FolkRank[25] and tensor decomposition[31-34] have demonstrated their superior performance over traditional methods[26]. By adopting a tripartite graph to represent relationships among users, objects and tags, random walk models have also been successfully adopted for tag-aware recommendations[25-30].

However, the tripartite graph formulation overlooks relationships within nodes of the same type[30], thereby potentially impairing the recommendation performance. Intuitively, in a tripartite graph, connections only exist between different types of nodes. Therefore, a random walker can only jump from a node of a certain type (e.g., user) to a node of another type (e.g., object). If there exist connections between the nodes of the same type, the random walker can then take such shortcuts to perform a more effective journey. With this understanding, it is natural to ask the question of how to construct a network that includes the information between the nodes of the same type and how to enhance recommendation performance based on the resulting object-user-tag network.

To answer these research questions, we propose a novel approach, Trinity, to integrate tag annotations with historical data using a three-layered random walk with restart process. Our method considers not only relationships between heterogeneous nodes of different types but also those between homogeneous nodes of the same type. There are three steps in modelling this approach: 1) construct an object-user-tag heterogeneous network using both historical and tag data; 2) design a random walk with restart model by simulating the process that a random walker wanders on the constructed network; 3) take the steady-state probabilities that the walker stays at candidate objects to measure the strength of associations between a query user and the candidate objects. The Trinity method is validated through a series of experiments by employing large-scale 10-fold cross-validation. The results show that our method outperforms many existing methods in terms of three criteria for evaluating recommendation performance.

The main contributions of our paper are summarized as follows.

1) We propose to integrate several types of relationship in a single object-user-tag network. Connections in such a network correspond to relationships between i) objects, ii) users, iii) tags, iv) objects and users, v) objects and tags, and vi) users and tags. To the best of our knowledge, this work is the first one that explores a recommendation method with such a comprehensive formulation.

2) We propose a k-nearest neighbour strategy to construct a reliable object-user-tag network. This strategy effectively removes weak relationships between the nodes of the same type (objects, users and tags) and hence significantly improves the quality of the resulting heterogeneous network.

3) We propose a three-layered random walk with restart model for characterizing the strength of associations between objects and users in the constructed heterogeneous network. This model provides more accurate predictions for unknown object-user relationships than many existing methods, including supervised random walk, simulation of resource allocating processes, and traditional collaborative filtering.

The rest of the paper is organized as follows. Section 2 provides a brief review of prior work relevant to personalized recommendation methods, especially on network-based and tag-aware recommendation. In Section 3, we give an overview of the proposed method, the network construction, similarity computation, and validation and evaluation methods as well as methods for comparison. We then provide experimental results in Section 4, including the similarity correlation, performance improvement, contributions of individual elements and robustness of parameters across different datasets. Finally, we conclude the paper in Section 5.

2 Related Work

2.1 Ordinary Collaborative Filtering

A personalized recommendation method targets ranking a set of candidate objects for a given query user such that objects preferred by the user appear among top positions in the ranking list. In mathematics, this is typically done by calculating discriminant scores for candidate objects and then sorting the objects in non-ascending order according to their scores. In particular, as the most widely used recommendation method, a user-based collaborative filtering method (i.e., USim) is formulated as follows. Given the historical preference of u users on o objects, represented as a binary matrix $\boldsymbol{A} = (a_{ij})_{o\times u}$, where $a_{ij} = 1$ means that the i-th object is preferred by the j-th user in history and 0 otherwise, one calculates a user similarity matrix $\boldsymbol{S} = (s_{ij})_{u\times u}$, where

$$s_{ij} = \frac{\sum_{1\leqslant k\leqslant o} a_{ki}a_{kj}}{\sqrt{\sum_{1\leqslant k\leqslant o} a_{ki}^2}\sqrt{\sum_{1\leqslant k\leqslant o} a_{kj}^2}}$$

denotes the cosine measure, and

$$s_{ij} = \frac{\sum_{1\leqslant k\leqslant o} a_{ki} \wedge a_{kj}}{\sum_{1\leqslant k\leqslant o} a_{ki} \vee a_{kj}}$$

represents the Jaccard index measure. With this matrix, one assigns a score to a candidate object indexed by i as the weighted average of similarities for users who prefer the object, as

$$v_{ij} = \frac{\sum_{1\leqslant k\leqslant u} a_{ik}s_{kj}}{\sum_{1\leqslant k\leqslant u} s_{kj}}.$$

In a similar way, an object-based collaborative filtering method (i.e., OSim) initially calculates an object similarity matrix $\boldsymbol{R} = (r_{ij})_{o\times o}$ and then assigns a score to a candidate object indexed by i as the weighted average of similarities for objects that are preferred by the query user, as

$$v_{ij} = \frac{\sum_{1\leqslant k\leqslant o} a_{kj}r_{ik}}{\sum_{1\leqslant k\leqslant o} r_{ik}}.$$

2.2 Network-Based Approaches

It is possible that the existence of popular objects may adversely influence the correct scoring of candidate objects and further yield unreasonable recommendation results[35-36]. With this understanding, we have previously proposed a network-based collaborative filtering approach that constructs a nearest neighbour user similarity network and then calculates discriminant scores for candidate objects based on this network. We have demonstrated that the performance of USim is greatly enhanced by filtering out weak similarities in this way[37].

In mathematics, for a user indexed by j, one introduces a parameter α, filters out unreliable user similarity scores by sorting the j-th column of the user similarity matrix $\boldsymbol{S}$ in non-ascending order and then sets similarity scores for users whose ranks are greater than $\alpha\times u$ to zeros, obtaining a weight matrix $\boldsymbol{U} = (u_{ij})_{u\times u}$. With this matrix, the discriminant score for a candidate object indexed by i is calculated as

$$v_{ij} = \frac{\sum_{1\leqslant k\leqslant u} a_{ik}u_{kj}}{\sum_{1\leqslant k\leqslant u} u_{kj}}.$$

We refer to this method as USnn in this paper. In a similar way, a nearest neighbour object-similarity network can also be constructed by filtering out unreliable similarity scores in the object similarity matrix $\boldsymbol{R}$, obtaining the corresponding weight matrix $\boldsymbol{O} = (o_{ij})_{o\times o}$. The resulting object-based approach, referred to as OSnn in this paper, then calculates the discriminant score for a candidate object as

$$v_{ij} = \frac{\sum_{1\leqslant k\leqslant u} a_{ik}o_{jk}}{\sum_{1\leqslant k\leqslant u} o_{jk}}.$$

Another branch of network-based approaches is designed according to the resource redistribution process based on the user-object bipartite network. As a representative method, ProbS[19] derives a resource redistribution matrix for objects, $\boldsymbol{D} = (d_{ij})_{o\times o}$, as

$$d_{ij} = \sum_{k=1}^{u} \frac{a_{ik}a_{jk}}{\sum_{1\leqslant l\leqslant u} a_{jl} \sum_{1\leqslant l\leqslant o} a_{lk}},$$

and then calculates for a query user j the discriminant score of a candidate object i as

$$v_{ij} = \sum_{k=1}^{o} d_{ik}a_{kj}.$$

The above formulation can also be done in a user-based manner by simply exchanging the roles of users and objects. As a result, a resource redistribution matrix for users $\boldsymbol{E} = (e_{ij})_{u\times u}$ can be derived as

$$e_{ij} = \sum_{k=1}^{o} \frac{a_{ki}a_{kj}}{\sum_{1\leqslant l\leqslant o} a_{lj} \sum_{1\leqslant l\leqslant u} a_{kl}},$$

and the discriminant score of a candidate object i for user j can then be calculated as

$$v_{ij} = \sum_{k=1}^{u} e_{kj}a_{ik}.$$

Random walk models have been widely used and demonstrated successfully in the fields of machine learning[38], bioinformatics[39], and recommender systems[29]. Most recently, random walk with restart model[40] has been successfully used with bipartite graphs in information retrieval[41] and social recommendations[42]. In one of our previous papers, we have proposed a random walk with restart model on a constructed user similarity network to achieve effective personalised recommendations[43].

To take advantage of additional properties of users, objects and their relationships, optimization techniques have been incorporated into random walk models to refine weights of edges in a bipartite network. For example, the supervised random walk method[44] uses an edge strength function to convert properties of an edge to a numeric quantity, i.e., the strength of the edge. Particularly, when there exist multiple types of properties, the linear combination of the properties is considered, and an optimization technique is used to determine the weights of the properties in the linear function.

Borrowing this idea, we propose the following method, referred to as SWalk in this paper, as a baseline for comparison. Let $\boldsymbol{B} = (b_{ij})_{o\times t}$ be annotations of t tags for o objects, where $b_{ij} = 1$ if object i is annotated with tag j and 0 otherwise. Let $\boldsymbol{C} = (c_{ij})_{u\times t}$ represent associations of the tags and u users, where $c_{ij} = 1$ if user i has used tag j to annotate some object and 0 otherwise. We calculate the correlation between object i and user j as

$$w_{ij} = \frac{\sum_{1\leqslant k\leqslant t} b_{ik}c_{jk}}{\sqrt{\sum_{1\leqslant k\leqslant t} b_{ik}^2}\sqrt{\sum_{1\leqslant k\leqslant t} c_{jk}^2}},$$

say, the cosine similarity between the vectors corresponding to the object and the user. Repeating this calculation for every known relationship between objects and users, we are able to obtain a weighted object-user bipartite network. Random walk with restart on this network can then be performed.

2.3 Tag-Aware Recommendations Based on Networks

Among various approaches, network-based methods have been developed successfully and expanded flexibly for tag-aware recommendations[25-30]. In general, a tag-based network can be viewed as a tripartite graph that consists of three bipartite graphs[25-27]. Tag-aware recommendation methods are then designed by extending diffusion-based recommendation approaches[45] or utilizing diffusion kernels, probabilistic graphs and hypergraph models[46-49] to such a tripartite network[50].

With the understanding that tag information may contribute to the correct recommendation of candidate objects, ProbS has been extended to incorporate the tag information[30], resulting in a method referred to as ODiff in this paper. Briefly, ODiff performs two resource redistribution procedures, one for the user-object bipartite network ($\boldsymbol{V}'$) and the other for the object-tag bipartite network ($\boldsymbol{V}''$). Then, this method calculates the linear combination of the two resource distribution results to obtain the final discriminant scores matrix for candidate objects, as

$$\boldsymbol{V} = \lambda\boldsymbol{V}' + (1-\lambda)\boldsymbol{V}'',$$

where $\lambda \in [0, 1]$ is a free parameter.

The above linear combination strategy can also be conducted in a user-based manner by simply exchanging the roles of users and objects[45]. The resulting method, referred to as UDiff in this paper, derives two resource redistribution matrices for users from the user-object bipartite network ($\boldsymbol{E}'$) and user-tag bipartite network ($\boldsymbol{E}''$), respectively, calculates their linear combinations to obtain a final redistribution matrix as

$$\boldsymbol{E} = \lambda\boldsymbol{E}' + (1-\lambda)\boldsymbol{E}'',$$

and then derives discriminant scores for candidate objects accordingly. Here $\lambda \in [0, 1]$ is a free parameter.

3 Methods

3.1 Overview of Trinity

The proposed method, Trinity, is designed based on the notion that different correlations between users, objects and tags could contribute to the recommendation of candidate objects to a target user. As shown in Fig.1, Trinity is composed of three sequential steps.

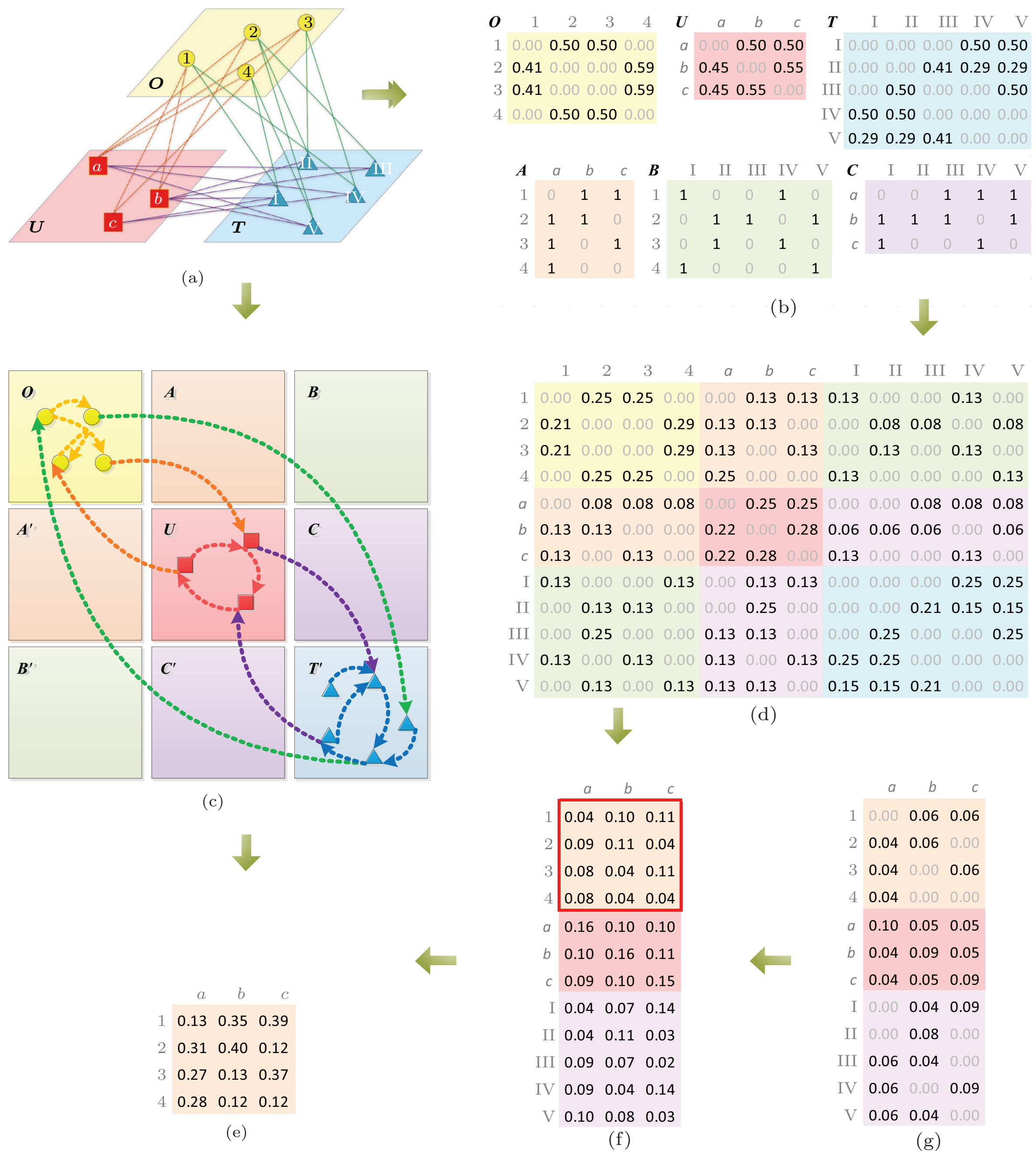

Fig.1. Overview of Trinity. (a) Network construction. (b) Corresponding weight/adjacency matrices. (c) Random walk simulation. (d) Transition matrix $\boldsymbol{W} = (w_{ij})_{12\times 12}$. (e) Ranking scores extraction. (f) Steady-state probability $\boldsymbol{p}^{(\infty)}$. (g) Initial probability $\boldsymbol{p}^{(0)}$.

Firstly, in the network construction step, we rely on historical data and tag annotations to construct an object-user-tag network composed of a user layer, an object layer, a tag layer and interconnections between every two layers. Specifically, as shown in Fig.1(a), the object layer ($\boldsymbol{O}$) is a sub-network composed of objects as nodes and their relationships as edges. The user layer ($\boldsymbol{U}$) is a sub-network composed of users as nodes and their relationships as edges. The tag layer ($\boldsymbol{T}$) is a sub-network composed of tags as nodes and their relationships as edges. As shown in Fig.1(b), the interconnections between the user and the object layers ($\boldsymbol{A}$) reflect preference of users on objects. The interconnections between the object and the tag layers ($\boldsymbol{B}$)

characterize tags used to annotate objects. The interconnections between the user and the tag layers ($\boldsymbol{C}$) reflect the preference of users in using the tags. It is clear that such an object-user-tag network is heterogeneous, not only in the sense that it contains three types of nodes (objects, users and tags), but also because edges in such a network can be classified into six different types according to the nodes connecting to the edges.

Secondly, in the random walk simulation step, we simulate the process that a random walker wanders in the constructed object-user-tag network. As illustrated in Fig.1(c), the walker can either move from one node to another in the same layer, or jump from one layer to another. Specifically, this process is determined by a transition matrix and a set of initial probabilities (a vector for a single user and a matrix for multiple users). The transition matrix can be partitioned into nine blocks, corresponding to the three layers ($\boldsymbol{O}$, $\boldsymbol{U}$ and $\boldsymbol{T}$) and their interconnections ($\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$ and their transposes) as illustrated in Fig.1(d). The initial probability vector/matrix reflects probabilities that the walker starts its journey and can be partitioned into three blocks, corresponding to the three layers (object, user and tag) from the top to the bottom, as illustrated in Fig.1(g).

Thirdly, upon finishing simulation, we extract steady-state probabilities that the walker stays in candidate objects and prioritize these objects accordingly. Specifically, as depicted in Fig.1(f), the steady-state probability vector can also be partitioned into three blocks. From top to bottom, these blocks correspond to probabilities that the walker stays in the object, user and tag layers, respectively. By extracting the upper block and performing normalization, we are able to obtain discriminant scores for candidate objects, which provide a means of prioritizing the objects, as illustrated in Fig.1(e).

Detailed descriptions of these steps will be further given in Subsection 3.2 and Subsection 3.3. The notations for Trinity and its computation procedure can be found in Table 1.

3.2 Construction of the Object-User-Tag Heterogeneous Network

An object-user-tag network is a three-layered network composed of an object layer, a user layer, a tag layer and interconnections between these layers. The object layer is constructed as a nearest neighbour network according to object similarities derived from known associations between objects and users. More specifically, given the preference of u users on o objects, represented as an association matrix $\boldsymbol{A} = (a_{ij})_{o\times u}$, where $a_{ij} = 1$ if object i is preferred by user j and 0 otherwise, we calculate the similarity between two objects i and j as the cosine of the angle between the corresponding row vectors. With all pairwise similarities calculated, we remove for each object a fraction of the weakest relationships and obtain a nearest neighbour network of objects. This is done by sorting each column of the object similarity matrix and setting similarities for objects ranked lower than $\alpha \times o$ to zero. Here α is a free parameter. We denote such a filtered object similarity matrix as $\boldsymbol{O} = (o_{ij})_{o\times o}$.

The user layer is also constructed as a nearest neighbour network according to user similarities derived from known associations between objects and users. With the association matrix $\boldsymbol{A} = (a_{ij})_{o\times u}$, we calculate the similarity between two users i and j as the cosine of the angle between the corresponding column vectors in this matrix. With all pairwise similarities calculated, we also keep for each user a fraction of α strongest relationships and obtain a nearest neighbour network of users. We denote the weight matrix for this network as $\boldsymbol{U} = (u_{ij})_{u\times u}$.

The tag layer is again constructed as a nearest neighbour network according to tag similarities derived from known associations between objects and tags. Given the annotations of o objects in terms of t tags, represented as an association matrix $\boldsymbol{B} = (b_{ij})_{o\times t}$, where $b_{ij} = 1$ if object i is annotated with tag j and 0 otherwise, we calculate the similarity between two tags i and j as the cosine of the angle between the corresponding column vectors in this matrix. With all pairwise similarities calculated, we again keep for each tag a fraction of α strongest relationships to obtain the tag network and denote the weight matrix for this network as $\boldsymbol{T} = (t_{ij})_{t\times t}$.

The interconnections between the object and the user layers are represented as the association matrix $\boldsymbol{A} = (a_{ij})_{o\times u}$. The interconnections between the object and the tag layers are represented as the association matrix $\boldsymbol{B} = (b_{ij})_{o\times t}$. The interconnections between the user and the tag layers are represented as the association matrix $\boldsymbol{C} = (c_{ij})_{u\times t}$, where $c_{ij} = 1$ if user i has used tag j to annotate some object and 0 otherwise.

Putting the above matrices together, we describe an object-user-tag network using a 6-tuple $\boldsymbol{H} = (\boldsymbol{O}, \boldsymbol{U}, \boldsymbol{T}, \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C})$, where $\boldsymbol{O}$, $\boldsymbol{U}$ and $\boldsymbol{T}$ are weight matrices of the object, user and tag layers, respectively, and $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$ adjacency matrices for the object-user, object-tag and user-tag relationships, respectively.

3.3 Random Walk with Restart on an Object-User-Tag Network

We propose a random walk with restart model on the constructed object-user-tag network to facilitate the recommendation of candidate objects in an attempt to simulate the process that a random walker wanders in such a heterogeneous network. Given a query user, a random walker starts a journey in the object-user-tag network with some initial probability $\boldsymbol{p}^{(0)}$. Then, in each step of the journey, the walker may select to start a new journey with probability λ or move on with probability $1-\lambda$. When moving on, the walker may select to stay in the same layer with probability β or jump to one of the other two layers with probability $(1-\beta)/2$. When wandering about, the walker moves to one of its direct neighbours in the same layer. After a number of steps, the probability that the walker stays in each node of the object-user-tag network will reach a steady state $\boldsymbol{p}^{(\infty)}$, from which we can extract the strength of the association between the query user and candidate objects.

Formally, let $\boldsymbol{\pi} = (\pi_{ij})_{3\times 3}$ be the transition matrix when the random walker jumps between different layers of the object-user-tag network, where π_{ij} is the probability of jumping from the i-th layer to the j-th layer ($i = 1, 2, 3$ for object, user and tag layers, respectively). To simplify the problem, we put two constraints on this matrix: 1) probabilities of staying in the same layer should be equal ($\pi_{ii} = \beta$ for $i = 1, 2, 3$), and 2) probabilities of jumping from one layer to another should be equal ($\pi_{ij} = \gamma$ for $i, j = 1, 2, 3$, $i \neq j$). It is then evident that $\gamma = (1-\beta)/2$, and hence β is the only free parameter in this matrix. For a matrix $\boldsymbol{X} = (x_{ij})_{m\times n}$, we define the row-normalised matrix as $\tilde{\boldsymbol{X}} = (\tilde{x}_{ij})_{m\times n}$ with $\tilde{x}_{ij} = x_{ij}/\sum_{k=1}^{n} x_{ik}$ and the column-normalised matrix as $\underset{\sim}{\boldsymbol{X}} = (\underset{\sim}{x}_{ij})_{m\times n}$ with $\underset{\sim}{x}_{ij} = x_{ij}/\sum_{k=1}^{m} x_{kj}$. We can then define the un-normalized transition matrix $\hat{\boldsymbol{W}}$ as

$$\hat{\boldsymbol{W}} = \begin{pmatrix} \beta\tilde{\boldsymbol{O}} & \gamma\tilde{\boldsymbol{A}} & \gamma\tilde{\boldsymbol{B}} \\ \gamma\underset{\sim}{\boldsymbol{A}}^{\mathrm{T}} & \beta\tilde{\boldsymbol{U}} & \gamma\tilde{\boldsymbol{C}} \\ \gamma\underset{\sim}{\boldsymbol{B}}^{\mathrm{T}} & \gamma\underset{\sim}{\boldsymbol{C}}^{\mathrm{T}} & \beta\tilde{\boldsymbol{T}} \end{pmatrix},$$

and perform row-normalization to obtain the transition matrix $\boldsymbol{W} = (w_{ij})_{(o+u+t)\times(o+u+t)}$, where $w_{ij} = \hat{w}_{ij}/\sum_{j=1}^{o+u+t} \hat{w}_{ij}$.

Given a query user indexed by q, we denote the initial probabilities for the object layer as $\boldsymbol{o}^{(0)} = (o_i^{(0)})_{o\times 1}$ and obtain $\boldsymbol{o}^{(0)}$ by assigning equal probabilities to objects that have been selected by the query user historically. It is then evident that $\boldsymbol{o}^{(0)}$ is equivalent to the q-th column of the column-normalized association matrix ($\underset{\sim}{\boldsymbol{A}}$). Let $\boldsymbol{u}^{(0)} = (u_i^{(0)})_{u\times 1}$ be the initial probability for the user layer. We assign probabilities to direct

Table 1. Notations for Trinity

Notation	Meaning
$\boldsymbol{H}$	Object-user-tag network as a 6-tuple $\boldsymbol{H} = (\boldsymbol{O}, \boldsymbol{U}, \boldsymbol{T}, \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C})$
$\boldsymbol{O}$, $\boldsymbol{U}$, $\boldsymbol{T}$	Weight matrices of the object layer, the user layer and the tag layer, respectively
$\tilde{\boldsymbol{O}}, \tilde{\boldsymbol{U}}, \tilde{\boldsymbol{T}}$	Row-normalized matrix of $\boldsymbol{O}$, $\boldsymbol{U}$ and $\boldsymbol{T}$
$\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$	Association matrices between objects and users, between objects and tags and between users and tags, respectively
$\tilde{\boldsymbol{A}}, \tilde{\boldsymbol{B}}, \tilde{\boldsymbol{C}}$	Row-normalized matrix of $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$, respectively
$\underset{\sim}{\boldsymbol{A}}$, $\underset{\sim}{\boldsymbol{B}}$, $\underset{\sim}{\boldsymbol{C}}$	Column-normalized matrix of $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$, respectively
α	Fraction of the nearest neighbours for objects, users and tags
β	Probability that the random walker is moving on but stays in the same layer ($\boldsymbol{U}$, $\boldsymbol{O}$ or $\boldsymbol{T}$), in each step in $\boldsymbol{H}$
γ	Probability that the random walker is moving on and jumps to one of the other two layers in $\boldsymbol{H}$ and $\gamma = (1-\beta)/2$
λ	Probability that the random walker starts a new journey, and 1-λ is the probability that the random walker moves on, in each step in $\boldsymbol{H}$
$\hat{\boldsymbol{W}}$	Un-normalized transition matrix, where $\hat{\boldsymbol{W}} = \begin{pmatrix} \beta\tilde{\boldsymbol{O}} & \gamma\tilde{\boldsymbol{A}} & \gamma\tilde{\boldsymbol{B}} \\ \gamma\underset{\sim}{\boldsymbol{A}}^{\mathrm{T}} & \beta\tilde{\boldsymbol{U}} & \gamma\tilde{\boldsymbol{C}} \\ \gamma\underset{\sim}{\boldsymbol{B}}^{\mathrm{T}} & \gamma\underset{\sim}{\boldsymbol{C}}^{\mathrm{T}} & \beta\tilde{\boldsymbol{T}} \end{pmatrix}$
$\boldsymbol{W}$	Row-normalized transition matrix of $\hat{\boldsymbol{W}}$
$\boldsymbol{\pi}$	Transition matrix when the random walker jumps between different layers of the object-user-tag network, whose element π_{ij} is the probability of jumping from the i-th layer to the j-th layer
$\boldsymbol{o}^{(0)}$, $\boldsymbol{u}^{(0)}$, $\boldsymbol{t}^{(0)}$	Initial probabilities for the object layer, the user layer and the tag layer, respectively
$\tilde{\boldsymbol{p}}^{(0)}$	Un-normalized initial probability vector, $\tilde{\boldsymbol{p}}^{(0)} = ((\boldsymbol{o}^{(0)})^{\mathrm{T}}, (\boldsymbol{u}^{(0)})^{\mathrm{T}}, (\boldsymbol{t}^{(0)})^{\mathrm{T}})^{\mathrm{T}}$
$\boldsymbol{p}^{(0)}$	Normalized initial probability vector
$\boldsymbol{p}^{(\infty)}$	Steady-state probability which can be decomposed into an object part $\boldsymbol{o}^{(\infty)}$, a user part $\boldsymbol{u}^{(\infty)}$ and a tag part $\boldsymbol{t}^{(\infty)}$, say, $\boldsymbol{p}^{(\infty)} = ((\boldsymbol{o}^{(\infty)})^{\mathrm{T}}, (\boldsymbol{u}^{(\infty)})^{\mathrm{T}}, (\boldsymbol{t}^{(\infty)})^{\mathrm{T}})^{\mathrm{T}}$

neighbours of the query user according to their similarities. It then follows that $\boldsymbol{u}^{(0)}$ is equivalent to the q-th column of the column-normalized matrix ($\underset{\sim}{\boldsymbol{U}}$). Let $\boldsymbol{t}^{(0)} = (t_i^{(0)})_{t\times 1}$ be the initial probability for the tag layer. We assign equal probabilities to tags that have been used by the query user. It then follows that $\boldsymbol{t}^{(0)}$ is equivalent to the q-th column of the transposition of the column-normalized association matrix ($\underset{\sim}{\boldsymbol{C}}^{\mathrm{T}}$). We then write the un-normalized initial probability vector as $\tilde{\boldsymbol{p}}^{(0)} = ((\boldsymbol{o}^{(0)})^{\mathrm{T}}, (\boldsymbol{u}^{(0)})^{\mathrm{T}}, (\boldsymbol{t}^{(0)})^{\mathrm{T}})^{\mathrm{T}}$ and further normalize it to obtain the initial probability vector $\boldsymbol{p}^{(0)}$.

Let $\boldsymbol{p}^{(t)}$ contain probabilities that the walker stays at each node at time t. We have the iterative formula:

$$\boldsymbol{p}^{(t+1)} = (1-\lambda)\boldsymbol{W}^{\mathrm{T}}\boldsymbol{p}^{(t)} + \lambda\boldsymbol{p}^{(0)}.$$

Repeating the iteration for a number of steps until $\boldsymbol{p}^{(t)}$ is stable (e.g., the L_1 norm of $\Delta\boldsymbol{p} = \boldsymbol{p}^{(t+1)} - \boldsymbol{p}^{(t)}$ is less than a small positive number ε), we obtain the steady-state probability $\boldsymbol{p}^{(\infty)}$, which can be decomposed into an object part $\boldsymbol{o}^{(\infty)}$, a user part $\boldsymbol{u}^{(\infty)}$ and a tag part $\boldsymbol{t}^{(\infty)}$, say $\boldsymbol{p}^{(\infty)} = ((\boldsymbol{o}^{(\infty)})^{\mathrm{T}}, (\boldsymbol{u}^{(\infty)})^{\mathrm{T}}, (\boldsymbol{t}^{(\infty)})^{\mathrm{T}})^{\mathrm{T}}$. The object part, $\boldsymbol{o}^{(\infty)}$, after normalization, can then be used to score the strength of association between the user and the candidate objects. It has been shown that the random walk model is not sensitive to the parameters involved in the model. We therefore, for the sake of simplicity, set default values for the parameters as $\beta = 0.5$, $\lambda = 0.5$ and $\varepsilon = 10^{-4}$.

An alternative approach is to obtain steady-state probabilities directly. Since $\boldsymbol{p}^{(\infty)} = (1-\lambda)\boldsymbol{W}^{\mathrm{T}}\boldsymbol{p}^{(\infty)} + \lambda\boldsymbol{p}^{(0)}$ represents the steady state, we solve this linear equation and obtain $\boldsymbol{p}^{(\infty)} = \lambda(\boldsymbol{I} - (1-\lambda)\boldsymbol{W}^{\mathrm{T}})^{-1}\boldsymbol{p}^{(0)}$ where $\boldsymbol{I}$ is the identity matrix. In literature, the simulation method is more frequently used. However, we do not see any distinguishable difference between the results obtained by these two strategies in most situations, except that the method of matrix inversion is typically two or three times faster.

3.4 Validation Methods and Evaluation Criteria

We perform 10-fold cross-validation experiments to validate our method and evaluate its performance against three criteria. For this purpose, we partition known associations between objects and users at random into 10 subsets of almost equal size. In each validation run, we use nine subsets as training data to calculate discriminant scores and adopt the remaining one as test data for assessment. Repeating such a validation run 10 times until every subset is served as the test data once, we collect the results and derive three criteria to measure recommendation performance, as shown below.

The first criterion is the mean rank ratio (MRR). In the validation experiment, we collect for each user a set of test objects that are preferred by the user in the test data and a set of control objects that are not preferred by the user in either training or test data. We then repeatedly sort a test object against the control ones to obtain its rank. In the case that multiple objects have equal scores, we break the tie by assigning equal ranks to these objects. With ranks of all test objects collected, we divide them by the total number of test and control objects to obtain rank ratios and the average over all ratios to obtain the mean rank ratio.

The second criterion is the area under the rank receiver operating characteristic curve (AUC). At a certain threshold of the rank ratio, we define the sensitivity as the fraction of test objects ranked above the threshold and the specificity as the fraction of control objects ranked below the threshold. Varying the threshold from 0 to 1, we are able to plot a rank receiver operating characteristic (ROC) curve (sensitivity versus 1-specificity) and further calculate the area under this curve as the AUC score.

The third criterion is the hit rate (HIT) at a rank threshold T (with default value 10 in this paper). With all test objects ranked, we count the number of test objects ranked among top T and divide this number by the total number of test objects to obtain the hit rate at threshold T. Clearly, a method with high recommendation performance tends to have a small mean rank ratio, a large AUC score and a large hit rate.

4 Experimental Results

4.1 Data Sources

Three large-scale datasets from the Koblenz Network Collection[1] were obtained for the validation of our approach.

The first dataset of CiteULike contains 153 277 tags assigned to 731 769 publications by 22 715 users[51]. Focusing on publications annotated by five or more tags, users having annotated 10 or more publications, and tags being used 10 or more times, we obtained 4 633 publications that were annotated by 7 463 users with the use of 6 703 tags. We further identified 35 532 links

[1] http://konect.uni-koblenz.de, April 2016.

between the publications and the users, 45 679 links between the publications and the tags, and 266 841 links between the users and the tags.

The second dataset of MovieLens contains 16 528 tags assigned to 7 601 movies by 4 009 users[52]. Focusing on movies annotated by two or more tags, users having annotated two or more movies, and tags being used two or more times, we obtained 3 856 movies that were annotated by 1 165 users with the use of 2 785 tags. We further identified 24 754 links between the movies and the users, 23 906 links between the movies and the tags, and 14 137 links between the users and the tags.

The third dataset of BibSonomy contains 204 673 tags assigned to 771 290 publications by 5 794 users[53]. Focusing on publications annotated by five or more tags, users having annotated 10 or more movies, and tags being used 10 or more times, we obtained 1 442 publications that were annotated by 1 968 users with the use of 4 972 tags. We further identified 11 152 links between the publications and the users, 23 300 links between the publications and the tags, and 139 399 links between the users and the tags.

4.2 Correlation Between Similarities

We validated whether the derived user similarity was correlated with object similarity according to annotated associations between objects and users. Focusing on the CiteULike dataset, we derived a user similarity matrix and an object similarity matrix using the cosine measure. We then derived a quantity named mean object similarity by calculating for each pair of users the average pairwise similarity between their associated objects. Next, we analyzed the relationship between user similarity and mean object similarity by partitioning mean object similarity values into 30 bins of equal size and averaging over both mean object similarity values and corresponding user similarity values in each bin.

As summarized in Fig.2(a), the results show the positive correlation between user similarity and mean object similarity. For example, for user pairs with weak mean object similarities (e.g., no more than 0.10), the corresponding user similarities are also low (i.e., no more than 0.05). For user pairs with relatively strong mean object similarities (e.g., about 0.2), the corresponding user similarities are also relatively strong (i.e., about 0.1). Furthermore, it is clear that, with an increase in mean object similarity, user similarity also increases, suggesting that users having selected similar objects also tend to be similar. We then derived two vectors, one composed of mean similarities of user pairs in the bins and the other consisting of corresponding mean object similarities, and calculated the correlation coefficient of these two vectors to quantitatively measure their relationship. Results show that the Pearson's and Spearman's correlation coefficients are 0.951 1 (p-value $= 8.88 \times 10^{-16}$) and 0.936 4 (p-value $= 3.7 \times 10^{-8}$), respectively, revealing that user similarity positively correlates with mean object similarity with strong statistical significance.

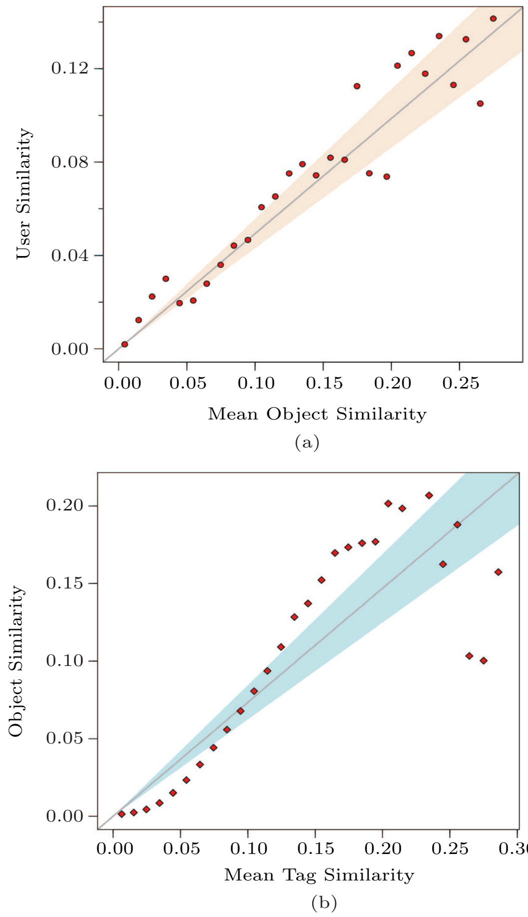

Fig.2. Correlations between similarities. (a) The user similarity is positively correlated with the mean object similarity and can be explained using a regression through the origin model. (b) The object similarity is positively correlated with the mean tag similarity and can also be explained using a regression through the origin model. The grey line across the diagonal is the estimated regression line, and shadow areas are 99% confidence bonds of the estimated means at different values of the predictor variables.

We further performed a regression analysis using the user similarity as the response and the mean object similarity as the predictor. Results show that the resulting model is a good fit ($r^2 = 0.901\,1$). The slope coefficient is estimated as 0.455 7 and is statistically significant (p-value $= 8.15\times10^{-16}$ by one-sided t test), while the intercept coefficient is almost zero (0.007 5) and is not significant (p-value $= 0.132$ by one-sided t test). We therefore discarded the intercept and fitted a regression through the origin model. Results show that the resulting model is a good fit ($r^2 = 0.975\,2$), and the slope coefficient is estimated as 0.493 3 (p-value $< 2.20\times10^{-16}$ by one-sided t test). These results demonstrate that mean object similarity implies user similarity.

In a similar way, we validated whether the derived object similarity was correlated with tag similarity according to annotated associations between objects and tags. To achieve this goal, we calculated a tag similarity matrix using the cosine measure and derived a quantity named mean tag similarity by calculating for each pair of objects the average pairwise similarity between their associated tags. We then analyzed the relationship between object similarity and mean tag similarity using the aforementioned method. Results, as summarized in Fig.2(b), show that the object similarity increases with the increase in the mean tag similarity, and the Pearson's and Spearman's correlation coefficients between them are 0.844 2 (p-value $= 2.41\times10^{-9}$) and 0.819 0 (p-value $= 7.76\times10^{-7}$), respectively. Furthermore, a regression through the origin model fits the data well ($r^2 = 0.914\,9$) with the slope coefficient estimated as 0.7346 (p-value $< 2.20\times10^{-16}$ by one-sided t test). These results demonstrate that mean tag similarity implies object similarity.

4.3 Improvement of Recommendation Performance

We investigated proportions of test objects that were ranked among top positions in the cross-validation experiment on the CiteULike dataset (with $\alpha = 1.5\%$, $\beta = 0.5$, $\gamma = 0.5$, $\lambda = 0.5$). As shown in Fig.3(a), our approach is able to rank 4.42% test objects at the top on average. Since a random guess procedure can only rank $1/(4\,633 - 35\,532/7\,463) \times 100\% \approx 0.021\,6\%$ test objects at the top on average, the effectiveness of our method is strongly supported.

We then compared the ranking performance of our method with that of SWalk, ODiff (with $\lambda = 0.4$), UDiff (with $\lambda = 0.7$), OSnn (with $\alpha = 10\%$) and USnn (with $\alpha = 10\%$). Note that the parameters involved in these methods are determined by grid search procedures to maximize the performance of these methods. We observe that SWalk ranks 4.19% of objects at the top, 0.23% less than our method. ODiff ranks 3.93% of objects at the top, 0.49% less than our approach. One-sided chi-squared tests suggest that the proportion of test objects ranked at the top by our method is significantly larger than those by SWalk (p-value $= 5.92\times10^{-4}$) and ODiff (p-value $= 5.92\times10^{-4}$). We also observe that UDiff, OSnn, USnn rank 3.39%, 2.49% and 2.89% of objects at the top, respectively, and 1.03%, 1.93% and 1.53% less than our approach, respectively. One-sided chi-squared tests also suggest that our method ranks more test objects at the top than other methods after the Bonferroni correction for multiple comparison (p-values $< 2.27\times10^{-9}$).

We calculated the proportion of test objects ranked higher than or equal to a position to obtain the empirical cumulative distribution of top ranked test objects. As shown in Fig.3(b), the cumulative distribution curve of our method clearly stays above those of the other methods, suggesting the superior performance of this approach. In detail, our method ranks 22.47% of test objects among the top 10, while SWalk, ODiff, UDiff, OSnn and USnn rank 19.33%, 21.82%, 16.94%, 12.23%

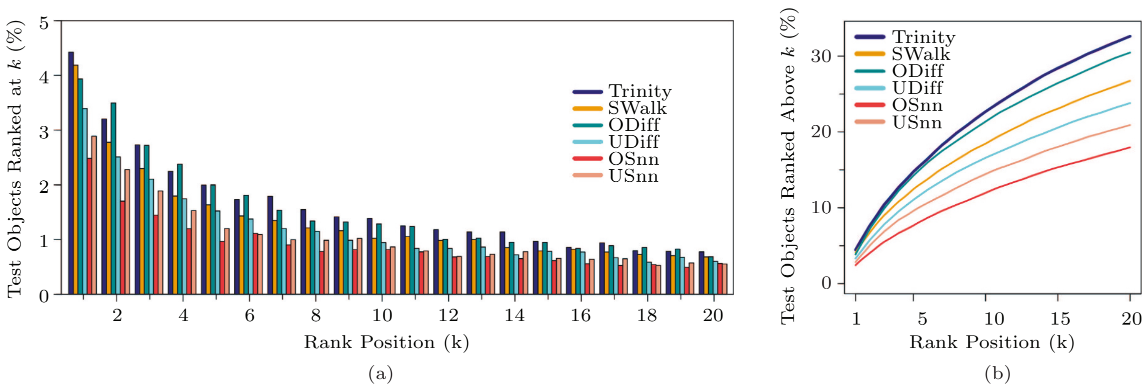

Fig.3. Improvement of ranking performance. (a) Proportions of test objects ranked at a position (probability mass). (b) Proportions of test objects ranked above a position (cumulative distribution).

and 14.76% of test objects among top 10, respectively. Pairwise one-sided chi-squared tests suggest that the proportion of test objects ranked among the top 10 by our method is significantly larger than those by the others (p-values < 0.05).

We then performed an object-wise comparison of ranking performance for different methods by testing whether rank positions of test objects produced by one method are significantly higher than those yielded by another through a one-sided Wilcoxon rank sum test. Results suggest that our method outperforms all the other methods in comparison (p-values $< 2.20\times10^{-16}$). We further performed a user-wise comparison of ranking performance via a binomial exact test. For a certain user, we claimed that method A outperforms method B if rank positions of more than half test objects generated by the former are ahead of those provided by the latter. We then counted the number of users for whom method A outperforms B and tested whether the relative frequency of such users is greater than 0.5 using a one-sided binomial exact test. Results suggest that our method outperforms all the other methods in comparison at the statistical significance level of 10^{-8} (with the Bonferroni correction applied for multiple comparison), consistent with the results obtained from the object-wise comparison.

We further assessed the performance of each method using the criteria defined in the method section, and summarized the results in Table 2 and Fig.4, from which we observe that our approach in general outperforms all the other methods. For example, our method achieves a mean rank ratio (MRR) of 4.73%, suggesting that on average a test object can be ranked at about 4∼5 out of 100 candidates. In comparison, SWalk (SW), as the method with the second highest performance, only achieves an MRR of 6.07%. ODiff (OD), UDiff (UD), OSnn (OS) and USnn (US) achieve MRRs of 8.00%, 8.35%, 13.56% and 13.68%, respectively. A one-sided Wilcoxon rank sum test based on 10 independent repeats of the validation experiments suggests that MRR of our method is significantly smaller than that of SWalk (p-value $= 9.13\times10^{-5}$), which is in turn significantly smaller than those of the other methods.

Table 2. Performance of Different Methods

	MRR (%)	AUC (%)	HIT (%)
Trinity	**4.73 (0.13)**	**95.27 (0.13)**	**22.47 (0.49)**
SWalk	6.07 (0.14)	93.92 (0.14)	19.33 (0.54)
ODiff	8.00 (0.29)	92.00 (0.29)	21.82 (0.65)
UDiff	8.35 (0.20)	91.65 (0.20)	16.94 (0.41)
OSnn	13.56 (0.26)	86.44 (0.26)	12.23 (0.41)
USnn	13.68 (0.32)	86.31 (0.32)	14.76 (0.44)

Note: results (in percentage) are means (standard derivations) obtained by 10 independent runs of 10-fold cross-validation experiments on the CiteULike dataset.

In terms of the AUC score, our method achieves a high value of 95.27%, while SWalk, ODiff, UDiff, OSnn and USnn achieve AUCs of 93.92%, 92.00%, 91.65%, 86.44% and 86.31%, respectively. One-sided Wilcoxon rank sum tests support the statistical significance of this superiority (p-values $< 9.13\times10^{-5}$). We further plotted ROC curves of all five methods in Fig.5. From the zoomed-in plot of this figure, we clearly observe that the curve corresponding to our method climbs much faster towards the top left corner of the plot than all the other methods, indicating the high performance of our approach.

In terms of the hit rate (HIT at $T = 10$), our method achieves a high value of 22.47%, while SWalk, ODiff, UDiff, OSnn and USnn achieve HITs of 19.33%, 21.82%,

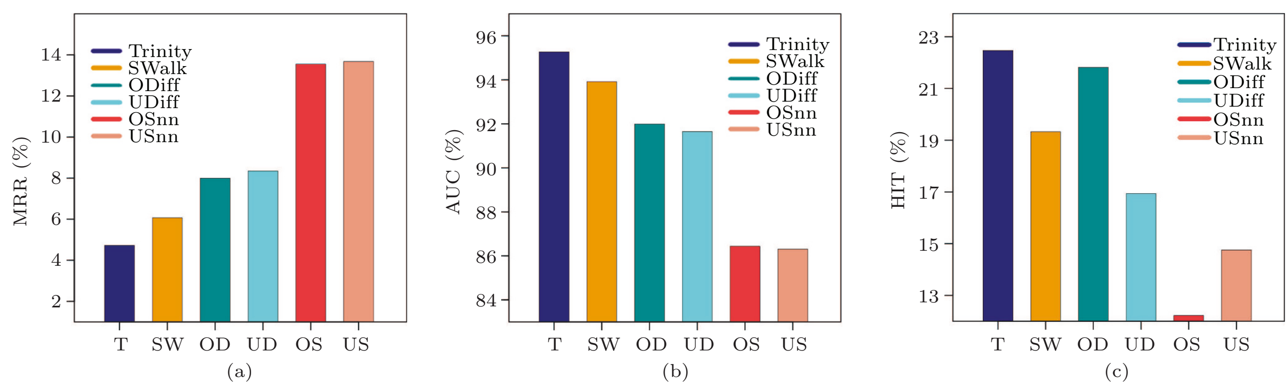

Fig.4. Comparison of recommendation performance of different methods. Trinity (T) clearly outperforms the others in terms of all the evaluation criteria.

16.94%, 12.23% and 14.76%, respectively. One-sided Wilcoxon rank sum tests again support the statistical significance of the superiority of our method (p-values $< 9.13 \times 10^{-5}$).

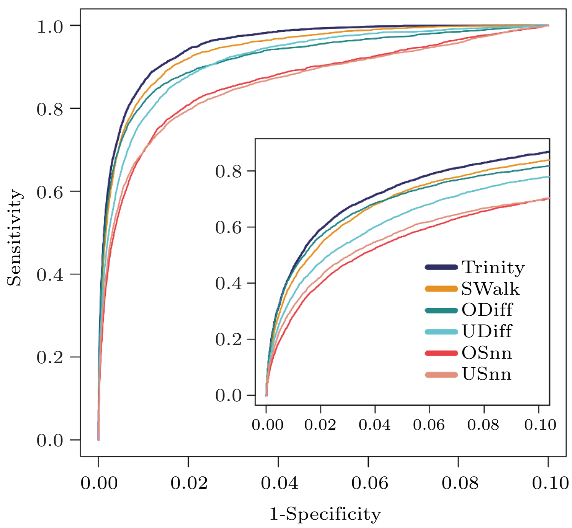

Fig.5. ROC curves of different methods. The curve of Trinity climbs much faster towards the top left corner of the plot than all the other methods, indicating the high performance of Trinity.

4.4 Contributions of Individual Components

An object-user-tag heterogeneous network $\boldsymbol{H} = (\boldsymbol{O}, \boldsymbol{U}, \boldsymbol{T}, \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C})$ is composed of an object layer $\boldsymbol{O}$, a user layer $\boldsymbol{U}$, a tag layer $\boldsymbol{T}$ and interconnections between these layers ($\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$). The initial probability in the random walk model $\tilde{\boldsymbol{p}}^{(0)} = ((\boldsymbol{o}^{(0)})^{\mathrm{T}}, (\boldsymbol{u}^{(0)})^{\mathrm{T}}, (\boldsymbol{t}^{(0)})^{\mathrm{T}})^{\mathrm{T}}$ consists of probabilities in the object layer ($\boldsymbol{o}^{(0)}$), the user layer ($\boldsymbol{u}^{(0)}$) and the tag layer ($\boldsymbol{t}^{(0)}$). It is therefore necessary to assess contributions of these components to the final performance of our method.

We first removed information regarding objects from the transition matrix by setting components $\boldsymbol{O}$, $\tilde{\boldsymbol{A}}$, $\underset{\sim}{\boldsymbol{A}}$, $\tilde{\boldsymbol{B}}$ and $\underset{\sim}{\boldsymbol{B}}$ to zero. The resulting model therefore simulated random walking on a two-layered heterogeneous network that was composed of users and tags only. Results, as shown in Table 3 and Fig.6, which have the same method index, unsurprisingly suggest the ineffectiveness of this model in making recommendations, due to the fact that the steady-state probability of staying in any object will always be zero with this model. In detail, MRR is as high as 49.98%, while AUC and HIT are as low as 50.00% and 0.17% respectively, all similar to the results of a random guess procedure.

Table 3. Contribution of Individual Components

Method	MRR (%)	AUC (%)	HIT (%)
A(Trinity)	**4.73 (0.10)**	**95.27 (0.10)**	**22.47 (0.49)**
-O	49.98 (0.46)	50.00 (0.46)	0.17 (0.06)
-U	8.46 (0.23)	91.53 (0.23)	17.70 (0.53)
-T	9.31 (0.29)	90.69 (0.29)	13.95 (0.49)
-o0	4.86 (0.11)	95.13 (0.11)	18.65 (0.58)
-u0	4.67 (0.09)	95.33 (0.09)	18.39 (0.58)
-t0	5.15 (0.13)	94.85 (0.13)	17.07 (0.56)

Note: results (in percentage) are means (standard derivations) obtained by 10 independent runs of 10-fold cross-validation experiments on the CiteULike dataset. A: with all information, -O: object information removed from the transition matrix, -U: user information removed from the transition matrix, -T: tag information removed from the transition matrix, -o0: object information removed from the initial probability vector, -u0: user information removed from the initial probability vector, -t0: tag information removed from the initial probability vector.

We next removed information regarding users from the transition matrix by setting components $\boldsymbol{U}$, $\tilde{\boldsymbol{A}}$, $\underset{\sim}{\boldsymbol{A}}$, $\tilde{\boldsymbol{C}}$ and $\underset{\sim}{\boldsymbol{C}}$ to zero. The resulting model, which simulates a random walking on a two-layered object-tag heterogeneous network, shows good recommendation performance (Table 3 and Fig.6). For example, MRR is as low

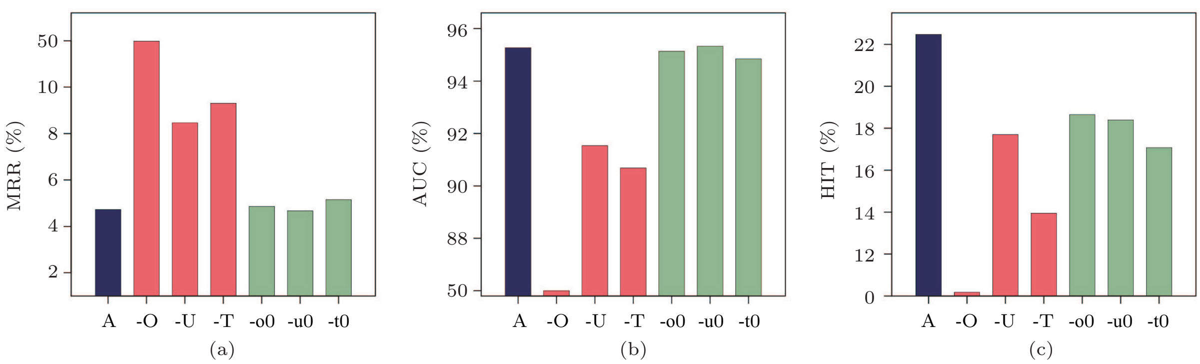

Fig.6. Contributions of individual components. Information provided by objects, users and tags all has positive contribution to the final performance of our approach.

as 8.46%, while AUC and HIT are as low as 91.53% and 17.70%, respectively. Nevertheless, we also see an obvious drop in performance when comparing these results with those of the original model (MRR = 4.73%, AUC = 95.27% and HIT = 22.47%), suggesting the importance of the tag information in making recommendations.

We then removed information regarding tags from the transition matrix by setting components $\boldsymbol{T}$, $\tilde{\boldsymbol{B}}$, $\underset{\sim}{\boldsymbol{B}}$, $\tilde{\boldsymbol{C}}$ and $\underset{\sim}{\boldsymbol{C}}$ to zero, resulting in a model that simulates random walking on a two-layered object-user heterogeneous network. Results, as shown in Table 3 and Fig.6 (MRR = 9.31%, AUC = 90.69% and HIT = 13.95%), suggest the mediocre performance of this model and verify the importance of the tag information in our three-layered random walk model.

We further removed object information from the initial probability by setting $\boldsymbol{o}^{(0)}$ to zero. Results, as shown in Table 3 and Fig.6 (MRR = 4.86%, AUC = 95.13% and HIT = 18.65%), are not significantly different from those of the original method, suggesting that, with the presence of both user and tag information in the initial probability, the importance of object information is limited. Similarly, the removal of the user information by setting $\boldsymbol{u}^{(0)}$ to zeros also results in a model (MRR = 4.67%, AUC = 95.33% and HIT = 18.39%) that is not significantly different from the original one, suggesting the limited importance of the user information to the initial probability. Furthermore, the removal of the tag information from the initial probability by setting $\boldsymbol{t}^{(0)}$ to zero also results in a model (MRR = 5.15%, AUC = 94.85% and HIT = 17.07%) that is not significantly different from the original one, again suggesting the limited importance of the tag information to the initial probability.

With the above analysis, we conclude that the information of objects, users and tags in the transition matrix has a larger contribution to the final performance of our approach than that in the initial probability vector. In the transition matrix, information regarding objects has the highest contribution, followed by information about tags and then that about users. In the initial probability vector, information about tags has the highest contribution, followed by information about objects and that about users. All these results support the usefulness of tag information in a recommender system.

4.5 Robustness to Parameters

There are three parameters in our method: the number of nearest neighbours (α), the probability of staying in the same layer (β) and the restart probability (λ). By default, these parameters α, β and λ are set to 1.5%, 0.5 and 0.5, respectively. It is therefore necessary to assess how these parameters influence the performance of our method.

We first varied the number of nearest neighbours, α, from 1.0% to 2.0% with step 0.1%, while setting the other parameters to their default values. The performance of our method at different values of this parameter, as shown in Fig.7(a), clearly suggests the robust-

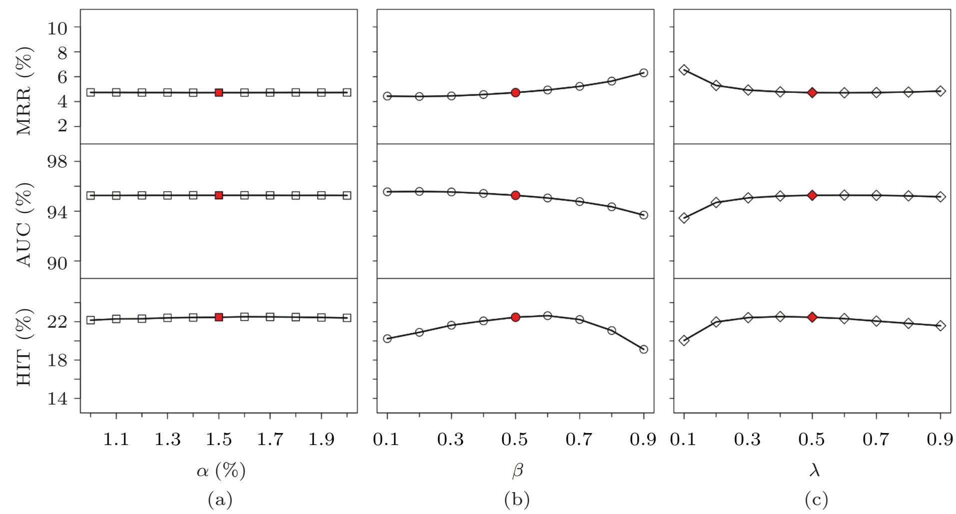

Fig.7. Robustness to related parameters. (a) Performance of our method on different numbers of the nearest neighbours (α). (b) Performance of our method on different probabilities of staying at the same layer (β). (c) Performance of our method on different restart probabilities (λ).

ness of our method to this parameter in a wide range. For example, as the parameter values increase, MRR improves slowly from 4.74% at $\alpha = 1.0\%$, to 4.73% at $\alpha = 1.5\%$, and then stabilizes afterwards. Similarly, as the parameter values increase, AUC improves slowly from 95.26% at $\alpha = 1.0\%$ to 95.27% at $\alpha = 1.5\%$, and then stabilizes afterwards. Furthermore, as the parameter values increase, HIT improves slowly from 22.16% at $\alpha = 1.0\%$ to 22.47% at $\alpha = 1.5\%$, and then stabilizes afterwards. However, when considering the variance in validation experiments, differences in HIT at different α values are not statistically significant. We therefore conclude that the performance of our method is very robust regarding this parameter. This property is crucial to a good model in that one can simply select the parameter at the default value without losing the high performance. For the sake of simplicity, we select $\alpha = 1.5\%$ by default in this paper.

We then varied probability β of walking in the same layer from 0.1 to 0.9 with step 0.1 while fixing the other parameters to their default values. The performance of our method at different values of this parameter is shown in Fig.7(b). Taking the mean rank ratio as an example, when $\beta = 0.5$, MRR is 4.73%. When β decreases towards small values, MRR decreases to 4.67% at $\beta = 0.1$, suggesting that values of β in this range do not significantly affect the performance. On the other hand, when β increases towards large values, MRR increases to 6.31% at $\beta = 0.9$, suggesting a preference for small β values. The criterion of AUC demonstrates an opposite pattern to MRR, due to the fact that these two criteria are obviously negative correlation. When β decreases towards small values, AUC increases slightly from 95.27% at $\beta = 0.5$ to 95.56% at $\beta = 0.1$, suggesting that large values of β are preferred. On the other hand, when β increases towards large values, AUC decreases slowly to 93.68% at $\beta = 0.9$, suggesting a preference for small values of β. The criterion of HIT, however, shows a different pattern. When β increases from 0.1 to 0.5, HIT increases from 20.23% to 22.47%. Afterwards, HIT drops towards 19.11% at $\beta = 0.9$. Taking all the above observations into consideration, we conclude that neither small nor large values of β are preferred. Therefore, we take the middle value $\beta = 0.5$ as the default.

We finally varied the restart probability, λ, from 0.1 to 0.9 with step 0.1 while setting the other parameters to their default values. The performance of our method at different values of this parameter, as shown in Fig.7(c), again suggests the robustness of our method to this parameter in a wide range at large values. Taking the mean rank ratio as an example, MRR is 4.73% at $\lambda = 0.5$. When λ decreases towards small values, MRR increases to 6.55% at $\lambda = 0.1$, suggesting that small values of this parameter are not preferred. On the other hand, when λ increases towards large values, MRR shows only fluctuations, suggesting the stability of this parameter in this region. AUC demonstrates an opposite pattern to that of MRR. AUC is 95.27% at $\lambda = 0.5$. When λ decreases towards small values, AUC decreases to 93.45% at $\lambda = 0.1$. When λ increases towards large values, AUC exhibits a stable pattern, showing only reasonable fluctuations. Therefore, according to this criterion, we also have the conclusion that small values of this parameter are not preferred while large values do not significantly affect the performance. HIT demonstrates a similar pattern as AUC, though the highest value is achieved at $\lambda = 0.4$. In detail, HIT is 22.54% at $\lambda = 0.4$. When λ decreases towards small values, HIT decreases to 20.04% at $\lambda = 0.1$. When λ increases towards large values, HIT decreases slightly to 21.58% at $\lambda = 0.9$. Taking all the above into consideration, we have the conclusion that neither small nor large values of λ are preferred. Therefore, we take the middle value $\lambda = 0.5$ as the default.

4.6 Consistency Across Different Datasets

We finally asked the question whether the excellent recommendation performance achieved by our method is consistent between different datasets. To answer this question, we replaced the CiteULike dataset with the MovieLens dataset (1 165 users, 3 856 objects and 2 785 tags) and repeated the validation experiments. The parameters for the methods are optimized by grid search as, Trinity ($\alpha = 1.5\%$, $\beta = 0.5$, $\lambda = 0.5$), ODiff ($\lambda = 0.4$), UDiff ($\lambda = 0.9$), OSnn ($\alpha = 10\%$) and USnn ($\alpha = 10\%$). Results, as shown in Table 4 and Figs.8(a)$\sim$8(c), suggest that the excellent performance achieved by our method on the MovieLens dataset is consistent with that exhibited on the CiteULike one. More specifically, our method achieves an MRR of 13.27%, an AUC of 86.72% and an HIT of 17.66%, while ODiff, as the best existing method, only achieves an MRR of 15.15%, an AUC of 84.83% and an HIT of 14.52%. These results suggest the superiority of our method over the existing methods.

We further replaced the CiteULike dataset with the BibSonomy one (1 968 users, 1 442 objects and 4 972 tags) and repeated the validation experiments. The parameters for the methods are optimized by grid search as, Trinity ($\alpha = 1.0\%$, $\beta = 0.5$, $\lambda = 0.6$), ODiff ($\lambda = 0.4$), UDiff ($\lambda = 0.9$), OSnn ($\alpha = 10\%$) and USnn ($\alpha = 10\%$). Results, as shown in Table 5 and Figs.8(d)~8(f), again suggest that the excellent performance achieved by our method on this dataset is consistent with that exhibited on the CiteULike one. For example, our method achieves an MRR of 18.85%, an AUC of 81.13% and an HIT of 12.58%, while ODiff, as the best existing method, only achieves an MRR of 23.08%, an AUC of 76.89% and an HIT of 11.82%. These results further support the superiority of our method over the existing methods.

Table 4. Results on the MovieLens Dataset

Method	MRR (%)	AUC (%)	HIT (%)
Trinity	**13.27 (0.44)**	**86.72 (0.44)**	**17.66 (0.43)**
SWalk	23.71 (0.04)	76.28 (0.04)	4.50 (0.16)
ODiff	15.15 (0.44)	84.83 (0.44)	14.52 (0.97)
UDiff	24.27 (0.65)	75.72 (0.66)	6.93 (0.54)
OSnn	28.42 (0.64)	71.57 (0.64)	4.99 (0.48)
USnn	27.01 (0.66)	72.98 (0.66)	5.89 (0.48)

Note: results (in percentage) are means (standard derivations) obtained by 10 independent runs of 10-fold cross-validation experiments on the MovieLens dataset.

Table 5. Results on the BibSonomy Dataset

Method	MRR (%)	AUC (%)	HIT (%)
Trinity	**18.85 (0.09)**	**81.13 (0.09)**	**12.58 (1.02)**
SWalk	23.51 (0.08)	76.47 (0.00)	9.94 (1.09)
ODiff	23.08 (0.64)	76.89 (0.64)	11.82 (0.90)
UDiff	30.65 (0.71)	69.32 (0.71)	9.50 (0.97)
OSnn	29.79 (1.11)	70.18 (1.11)	9.33 (0.58)
USnn	32.34 (0.94)	67.63 (0.94)	8.80 (0.46)

Note: results (in percentage) are means (standard derivations) obtained by 10 independent runs of 10-fold cross-validation experiments on the BibSonomy dataset.

5 Conclusions

We proposed Trinity, a random walk with restart model, on a three-layered object-user-tag heterogeneous network towards personalized recommendation. We analyzed the performance of this method via a series of large-scale 10-fold cross-validation experiments. Results showed the superior performance of our method over several existing methods across three independent datasets of different sizes. We further analyzed the contribution of individual components in our model and demonstrated positive contributions of object, user and tag information to the final recommendation performance.

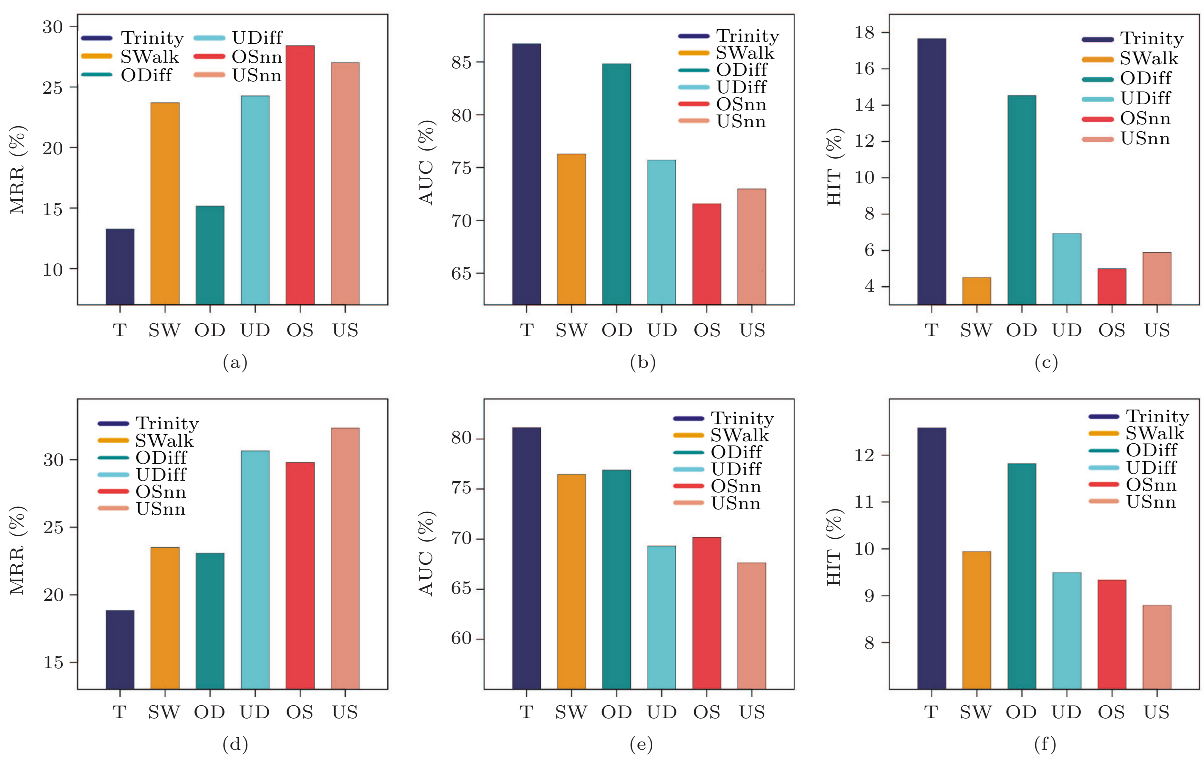

Fig.8. Performance of different methods on (a)~(c) MovieLens and (d)~(f) BibSonomy. Our method clearly outperforms the others in terms of all the evaluation criteria.

The success of the Trinity method can be attributed to a combination of several aspects. First, the nearest neighbour strategy for constructing networks in different layers of the object-user-tag network effectively removes weak relationships that may adversely affect the correct calculation of discriminant scores. Second, the random walk with restart model utilizes information in a more effective way than most of the existing approaches, such as those based on the ordinary collaborative filtering principle. Third, the combined use of not only relationships between objects but also those between users and tags in a single object-user-tag network improves the performance over methods that rely on only one or two types of relationship. As a result, Trinity achieves significant improvements in overall recommendation performance over existing methods.

Certainly, our approach can be further investigated from the following aspects. First, although we demonstrated the effectiveness of the random walk model, it is certainly not the only choice. Recently, graph algorithms such as the maximum flow and statistical models such as the regression analysis have all been successfully used in the analysis of biological networks. The possibility of introducing these methods in the design of recommender systems is then worth noting. Second, there are other types of information available for describing relationships between objects and users. For example, social networks have been shown to be a rich resource for relationships between users, while contents, annotations and comments can all be used to infer similarities between objects. How to integrate such valuable information towards to a recommender system of even higher performance will be a direction worth pursuing. Finally, the code start problem has been well recognized as one of the major obstacles to the improvement of recommendation performance. Although addressing this problem is already beyond the scope of our method, recent studies do suggest the possibility of overcoming the cold start problem from the viewpoint of semi-supervised co-training[54]. The incorporation of their ideas into our method to further enhance the performance of Trinity is one of our immediate goals.

References

[1] Huang Z, Chung W Y, Chen H C. A graph model for E-commerce recommender systems. *J. Am. Soc. Inf. Sci. Technol.*, 2004, 55(3): 259-274.

[2] Brin S, Page L. The anatomy of a large-scale hypertextual web search engine. *Comput. Networks and ISDN Systems*, 1998, 30: 107-117.

[3] Al-Masri E, Mahmoud Q H. Investigating web services on the world wide web. In *Proc. the 17th Int. Conf. World Wide Web*, April 2008, pp.795-804.

[4] Jeong B, Lee J, Cho H. Improving memory-based collaborative filtering via similarity updating and prediction modulation. *Inf. Sci.: an International Journal*, 2010, 180(5): 602-612.

[5] Adomavicius G, Tuzhilin A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng.*, 2005, 17(6): 734-749.

[6] Sarwar B, Karypis G, Konstan J, Reidl J. Item-based collaborative filtering recommendation algorithms. In *Proc. the 10th Int. Conf. World Wide Web*, May 2001, pp.285-295.

[7] Linden G, Smith B, York J. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Comput.*, 2003, 7(1): 76-80.

[8] Bogers T, van den Bosch A. Fusing recommendations for social bookmarking web sites. *Int. J. Electron. Commer.*, 2011, 15(3): 31-72.

[9] Prawesh S, Padmanabhan B. Probabilistic news recommender systems with feedback. In *Proc. the 6th ACM Conf. Recomm. Syst.*, September 2012, pp.257-260.

[10] Barragáns-Martínez A B, Costa-Montenegro E, Burguillo J C, Rey-López M, Mikic-Fonte F A, Peleteiro A. A hybrid content-based and item-based collaborative filtering approach to recommend TV programs enhanced with singular value decomposition. *Inf. Sci.: an International Journal*, 2010, 180(22): 4290-4311.

[11] Cheng J S, Sun A, Hu D N, Zeng D. An information diffusion-based recommendation framework for microblogging. *J. Assoc. Inf. Syst.*, 2011, 12(7): 463–486.

[12] Biau G, Cadre B, Rouvière L. Statistical analysis of k-nearest neighbor collaborative recommendation. *Ann. Stat.*, 2010, 38(3): 1568-1592.

[13] Georgiou O, Tsapatsoulis N. The importance of similarity metrics for representative users identification in recommender systems. In *Artif. Intell. Appl. Innov.*, Papadopoulos H, Andreou A S, Bramer M (eds.), Springer Berlin Heidelberg, 2010, pp.12-21.

[14] Pérez I, Cabrerizo F, Herrera-Viedma E. Group decision making problems in a linguistic and dynamic context. *Expert Syst. Appl.*, 2011, 38(3): 1675-1688.

[15] Burke R. Hybrid recommender systems: Survey and experiments. *User Model. User-Adapt. Interact.*, 2002, 12(4): 331-370.

[16] Koren Y, Bell R, Volinsky C. Matrix factorization techniques for recommender systems. *Computer*, 2009, 42(8): 30-37.

[17] Paterek A. Improving regularized singular value decomposition for collaborative filtering. In *Proc. the 13th KDD Cup Work.*, August 2007, pp.39-42.

[18] Wang X, Sun J T, Chen Z, Zhai C. Latent semantic analysis for multiple-type interrelated data objects. In *Proc. the 29th Annu. Int. ACM SIGIR Conf. Res. Dev. Inf. Retr.*, August 2006, pp.236-243.

[19] Zhou T, Ren J, Medo M, Zhang Y C. Bipartite network projection and personal recommendation. *Phys. Rev. E*, 2007, 76(4pt2): 046115.

[20] Gan, M. COUSIN: A network-based regression model for personalized recommendations. *Decision Support Systems*, 2016, 82: 58-68.

[21] Golder S A, Huberman B A. Usage patterns of collaborative tagging systems. *J. Inf. Sci.*, 2006, 32(2): 198-208.

[22] Vig J, Sen S, Riedl J. The tag genome: Encoding community knowledge to support novel interaction. *ACM Trans. Interact. Intell. Syst.*, 2012, 2(3): Article No. 13.

[23] Zhang Z K, Zhou T, Zhang Y C. Tag-aware recommender systems: A state-of-the-art survey. *J. Comput. Sci. Technol.*, 2011, 26(5): 767-777.

[24] Gan M. TAFFY: Incorporating tag information into a diffusion process for personalized recommendations. *World Wide Web—Internet and Web Information Systems*, 2015.

[25] Hotho A, Robert J, Schmitz C, Stumme G. Information retrieval in folksonomies: Search and ranking. In *Proc. the 3rd ESWC*, June 2006, pp.411-426.

[26] Lambiotte R, Ausloos M. Collaborative tagging as a tripartite network. In *Proc. the 6th ICCS*, May 2006, pp.1114-1117.

[27] Song Y, Zhuang Z, Li H, Zhao Q, Li J, Lee W C *et al.* Real-time automatic tag recommendation. In *Proc. the 31st Annu. Int. ACM SIGIR Conf. Res. Dev. Inf. Retr.*, July 2008, pp.515-522.

[28] Zhou X, Xu Y, Li Y, Josang A, Cox C. The state-of-the-art in personalized recommender systems for social networking. *Artif. Intell. Rev.*, 2011, 37(2): 119-132.

[29] Clements M, De Vries A P, Reinders M J T. The task-dependent effect of tags and ratings on social media access. *ACM Trans. Inf. Syst.*, 2010, 28(4): Article No. 21.

[30] Zhang Z K, Zhou T, Zhang Y C. Personalized recommendation via integrated diffusion on user-item-tag tripartite graphs. *Phys. A: Stat. Mech. Its Appl.*, 2010, 389(1): 179-186.

[31] Kolda T G, Bader B W. Tensor decompositions and applications. *SIAM Rev.*, 2009, 51(3): 455-500.

[32] Rendle S, Marinho L B, Nanopoulos A, Schmidt-Thieme L. Learning optimal ranking with tensor factorization for tag recommendation. In *Proc. the 15th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, June 28-July 1, 2009, pp.727-736.

[33] Symeonidis P, Nanopoulos A, Manolopoulos Y. Tag recommendations based on tensor dimensionality reduction. In *Proc. ACM Conf. Recomm. Syst.*, October 2008, pp.43-50.

[34] Symeonidis P, Nanopoulos A, Manolopoulos Y. A unified framework for providing recommendations in social tagging systems based on ternary semantic analysis. *IEEE Trans. Knowl. Data Eng.*, 2010, 22(2): 179-192.

[35] Zhang Q M, Zeng A, Shang M S. Extracting the information backbone in online system. *PLoS One*, 2013, 8(5): e62624.

[36] Gan M, Jiang R. Improving accuracy and diversity of personalized recommendation through power law adjustments of user similarities. *Decis. Support Syst.*, 2013, 55(3): 811-821.

[37] Gan M, Jiang R. Constructing a user similarity network to remove adverse influence of popular objects for personalized recommendation. *Expert Syst. Appl.*, 2013, 40(10): 4044-4053.

[38] Gan M, Jiang R. ROUND: Walking on an object-user heterogeneous network for personalized recommendations. *Expert Syst. Appl.*, 2015, 42(22): 8791-8804.

[39] Jiang R. Walking on multiple disease-gene networks to prioritize candidate genes. *J. Mol. Cell Biol.*, 2015, 7(3): 214-230.

[40] Yu W, Lin X. IRWR: Incremental random walk with restart. In *Proc. the 36th Int. ACM SIGIR Conf. Res. Dev. Inf. Retr.*, July 28-August 1, 2013, pp.1017-1020.

[41] Chiang M F, Liou J J, Wang J L, Peng W C, Shan M K. Exploring heterogeneous information networks and random walk with restart for academic search. *Knowl. Inf. Syst.*, 2012, 36(1): 59-82.

[42] Li J, Xia F, Wang W, Chen Z, Asabere N Y, Jiang H. ACRec: A co-authorship based random walk model for academic collaboration recommendation. In *Proc. the 23rd Int. Conf. World Wide Web Companion*, April 2014, pp.1209-1214.

[43] Gan M. Walking on a user similarity network towards personalized recommendations. *PLoS One*, 2014, 9(12): e114662.

[44] Backstrom L, Leskovec J. Supervised random walks: Predicting and recommending links in social networks. In *Proc. the 4th ACM Intl. Conf. Web Search and Data Mining*, Feb. 2011, pp.635-644.

[45] Shang M S, Zhang Z K, Zhou T, Zhang Y C. Collaborative filtering with diffusion-based similarity on tripartite graphs. *Phys. A: Stat. Mech. Its Appl.*, 2010, 389(6): 1259-1264.

[46] Tso-Sutter K H L, Marinho L B, Schmidt-Thieme L. Tag-aware recommender systems by fusion of collaborative filtering algorithms. In *Proc. the 2008 ACM Symp. Appl. Comput.*, March 2008, pp.1995-1999.

[47] Wetzker R, Umbrath W, Said A. A hybrid approach to item recommendation in folksonomies. In *Proc. WSDM 2009 Work. Exploit. Semant. Annot. Inf. Retr.*, February 2009, pp.25-29.

[48] Zlatić V, Ghoshal G, Caldarelli G. Hypergraph topological quantities for tagged social networks. *Phys. Rev. E.*, 2009, 80: 036118.

[49] Zhang Z K, Liu C. A hypergraph model of social tagging networks. *J. Stat. Mech.: Theory & Exp.*, 2010, 2010: P10005.

[50] Shang M S, Zhang Z K. Diffusion-based recommendation in collaborative tagging systems. *Chinese Phys. Lett.*, 2009, 26(11): 118903.

[51] Emamy K, Cameron R. CiteULike: A researcher's social bookmarking service. *Ariadne.*, 2007, 51(5).

[52] Grouplens Research. MovieLens datasets. http://www.grouplens.org/node/73, April 2016.

[53] Benz D, Hotho A, Jäschke R, Krause B, Mitzlaff F, Schmitz C *et al.* The social bookmark and publication management system BibSonomy. *VLDB J.*, 2010, 19(6): 849-875.

[54] Zhang M, Tang J, Zhang X, Xue X. Addressing cold start in recommender systems: A semi-supervised co-training algorithm. In *Proc. the 37th Int. ACM SIGIR Conf. Res. Dev. Inf. Retr.*, July 2014, pp.73-82.

Ming-Xin Gan received her B.S. degree in automation from Tsinghua University, Beijing, in 2001, her Ph.D. degree in management science and engineering from Beijing Institute of Technology, Beijing, in 2006. She is currently an associate professor in the Department of Management Science and Engineering, Donlinks School of Economics and Management, University of Science and Technology Beijing, Beijing. She has published more than 40 papers. Her current research interests concern recommender systems, information retrieval, complex systems, social network, big data analysis and visualization.

Lily Sun is an associate professor in information systems at School of Systems Engineering, University of Reading, Reading. She, with a background of computer science, earned her Ph.D. degree in information requirements modelling for decision support from Staffordshire University, Staffordshire, UK, in 2000. She has experience in complex systems design for commercial applications. Her research interests and recent publications are found in ontology modelling for knowledge representation, requirements engineering, and data analytics.

Rui Jiang received his B.S. and Ph.D. degrees in control science and engineering from Tsinghua University, Beijing, in 1997 and 2002, respectively. He worked as a postdoctoral research associate in Hong Kong University of Science and Technology, Hongkong, from 2002 to 2003 and then in the University of Southern California, Los Angeles, USA, from 2004 to 2007. He is currently an associate professor in the Department of Automation, Tsinghua University, Beijing. He works in the fields of intelligent information processing, knowledge discovery and data mining, information retrieval and text mining, and bioinformatics.