J Intell Inf Syst (2018) 50:479–500
DOI 10.1007/s10844-017-0468-1

Tag recommendation method in folksonomy based on user tagging status

Hong Yu[1] · Bing Zhou[2] · Mingyao Deng[1] · Feng Hu[1]

Received: 14 December 2014 / Revised: 19 May 2017 / Accepted: 19 May 2017 /
Published online: 6 June 2017

Abstract A folksonomy consists of three basic entities, namely users, tags and resources. This kind of social tagging system is a good way to index information, facilitate searches and navigate resources. The main objective of this paper is to present a novel method to improve the quality of tag recommendation. According to the statistical analysis, we find that the total number of tags used by a user changes over time in a social tagging system. Thus, this paper introduces the concept of user tagging status, namely the growing status, the mature status and the dormant status. Then, the determining user tagging status algorithm is presented considering a user's current tagging status to be one of the three tagging status at one point. Finally, three corresponding strategies are developed to compute the tag probability distribution based on the statistical language model in order to recommend tags most likely to be used by users. Experimental results show that the proposed method is better than the compared methods at the accuracy of tag recommendation.

Keywords Social tagging · Tag recommendation · Tagging status · Probability distribution · Folksonomy

1 Introduction

A folksonomy is a system of classification derived from the practice and method of collaboratively creating and managing tags to annotate and categorize content; this practice is also known as social tagging, social classification, social indexing and collaborative tagging (Trant 2009). Social tagging is widely used in various web sites to collect, retrieve and share information. For example, the CiteULike (http://www.citeulike.org/) uses tags for sharing

✉ Hong Yu
yuhong@cqupt.edu.cn

1 Chongqing Key Laboratory of Computational Intelligence, Chongqing University of Posts and Telecommunications, Chongqing, 400065, China

2 Department of Computer Science, Sam Houston State University, Huntsville, TX 77341, USA

bibliographic references, the Delicious (https://delicious.com/) uses tags for social bookmarking, the Last.fm (http://www.last.fm/) uses tags for sharing music listening habits, and the MovieLens (http://movielens.org/) uses tags for helping users to find the right movies.

These folksonomies allow users to annotate resources with their own tags, and tagging allows users to classify and find information collectively. Especially, for multimedia resources like music, photos or videos, tagging resources is the only feasible way to organize multimedia data and to make it searchable. These tags can be freely chosen by a user and are not restricted to any taxonomy (Krestel and Fankhauser 2012). Many existing studies have investigated a variety of co-occurrence patterns between entities from a folksonomy system. The unsupervised tagging results in some benefits like flexibility, quick adaption and easy usability, but also presents some challenges; for example, the wide variety of tags assigned by users can be redundant, ambiguous or entirely idiosyncratic (Hu et al. 2012).

Tag recommendation can deal with these challenges by suggesting tags that users are most likely to use for a resource. Recommending tags can serve various purposes, such as: increasing the chances of getting a resource annotated, reminding a user what a resource about, and consolidating the vocabulary across the users (Marinho et al. 2011). Furthermore, as Sood et al. (2007) pointed out that, tag recommendations "fundamentally change the tagging process from generation to recognition", which requires less cognitive effort and time. So more researchers and Internet enterprises pay highly attentions to tag recommendation. Recently, scholars have put forward various tag recommendation approaches, which mainly include the collaborative filtering approaches, the graph-based approaches, the content-based approaches and the hybrid approaches. The related work is introduced in the next section.

The existing achievements seldom consider the fact that users' tagging behavior changes with time. However, according to the statistical analysis, we find that the total number of tags used by a user changes over time in a social tagging system. In this paper, we study the tag recommendation method by considering the phenomenon that users' tagging behavior changes with time. We first propose three types of user tagging status, namely the growing status, the mature status and the dormant status; and the determining user tagging status algorithm is also devised. After analysing the characteristics of user tagging status, we present three corresponding tag recommendation strategies by computing tag probability distribution in users' and resources' tag space, based on the statistical language model. Finally, the results of comparison experiments on the CiteULike dateset and the Last.fm dateset show that the proposed tag recommendation method is better at the accuracy than the comparative approaches as the FolkRank (Kim and El Saddik 2011), the LocalRank (Kubatz et al. 2011) and the most popular tags ρ-mix (Jäschke et al. 2008).

The remainder of the paper is structured as follows. Section 2 briefly reviews the related work. Then we introduce some basic concepts in Section 3. In Section 4, we formalize the concept of user tagging status, and present the determining user tagging status algorithm. Section 5 brings the further tag recommendation method based on different user tagging status. The comparative experiment analysis on two social tagging systems are described in Section 6. Finally, some conclusions and discussions are given in Section 7.

2 Related work

In recent years, scholars have put forward various tag recommendation approaches. Generally speaking, these approaches could be divided into four categories, namely the collaborative filtering approaches, the graph-based approaches, the content-based approaches and the hybrid approaches.

Collaborative filtering is a common technique used by recommender systems. The traditional collaborative filtering methods cannot be applied directly, unless we reduce the ternary relation to a lower dimensional space, because there exists the ternary relationships among users, resources and tags in a social tagging system. Lu et al. (2011) developed a post-based collaborative filtering framework to recommend tags based on the query user's tagging history and tags that have been associated with the query document, leveraging the ternary relationships. Liu et al. (2011) injected the social relations between users and the content similarities between resources into a graph representation of folksonomies, exploited random-walk computation of similarities, and combined both the collaborative information and the tag preferences to recommend tags. Wang et al. (2013) put forward a novel hierarchical Bayesian model, which can seamlessly integrate the item-tag matrix, item content information and social networks between items into the same principled model based on extending the collaborative filtering approaches. Ma et al. (2015) proposed a recommendation approach fusing user-generated tags and social relations into a novel way, in order to solve the data sparsity problem and improve the recommendation accuracy.

The basic idea of graph-based approaches is to construct a graph with users, resources and tags as vertices and build edges according to user's tagging behavior (Liu et al. 2010). The kind of method dose not need consider the content of resources and semantic information of tags. Kim and El Saddik (2011) introduced a new way to compute the probabilistic interpretation in FolkRank by representing it as a linear combination of the personalized PageRank vectors. However, one of the major disadvantage of FolkRank is the steep computational costs. In contrast to the previous graph-based algorithms, Kubatz et al. (2011) computed the rank weights of tags only based on the tag space of a given user and resource. Ramezani (2011) suggested to improve the existing graph-based tag recommendation techniques by introducing a new model of the folksonomy as a directed graph. Rawashdeh et al. (2013) proposed to adapt the Katz measure in social tagging systems from a graph-based perspective. Cai et al. (2016) proposed the GRETA, a novel graph-based approach to assign tags for repositories on GitHub, based on constructing an Entity-Tag Graph (ETG) for GitHub using the domain knowledge from StackOverflow, and assign tags for repositories by taking a random walk algorithm. Hmimida and Kanawati (2016) proposed a graph-coarsening approach where a community detection algorithm is applied in the diversiform networks to speed up the execution time of graph-based tag recommenders in large-scale folksonomies.

The content-based approaches usually employ content of resources and adopt machine learning technology to recommend tags. Krestel and Fankhauser (2012) thoroughly investigated the use of language models for tag recommendation, showing that simple language models built from users and resources yield competitive performance while consuming only a fraction of the computational costs compared to more sophisticated algorithms. By modeling the generating process of social tagging systems in a Latent Dirichlet Allocation approach, Zhang et al. (2012) built a fully generative model for social tagging, leveraged it to estimate the relation among users, tags and resources in order to achieve the tag recommendation tasks. To learn the weights of different types of nodes and edges represented by features, Feng and Wang (2012) proposed an optimization framework, which learnt the best feature weights by maximizing the average area under the Curve of the tag recommender. Wu et al. (2016) proposed a generative model, where they can generate the words based on the tag-word distribution as well as the tag itself. Xie et al. (2016) proposed a novel generic model SenticRank to incorporate various sentiment information to various sentiment-based information for personalized recommendation by user profiles and other information.

Generally speaking, the hybrid approaches combine two or more than two kinds of tag recommendation algorithms. Gemmell et al. (2010) proposed a weighted linear hybrid incorporating simple popularity and collaborative filtering components, and the success of the hybrid over the lower-dimensional components demonstrates clearly the importance of an integrative approach that exploits multiple dimensions of the data. Belém et al. (2014) had proposed a personalized and object-centered tag recommendation methods for Web 2.0 applications. Kim and Kim (2014) investigated association rule, bigram, tag expansion, and implicit trust relationship for providing tag and item recommendations on a social tagging recommendation system. Wei et al. (2016) proposed a hybrid movie recommendation approach based on the user's annotating information to improve the ability of fusion and give the personalized recommendation services.

Furthermore, in the past few years, we have witnessed great advances in many perception tasks by using deep learning models. Wang and Yeung (2016) proposed a general framework for Bayesian deep learning and discussed the applications of deep learning on recommender systems, topic models and control. Wang et al. (2015) proposed a hierarchical Bayesian model called collaborative deep learning, which jointly performs deep representation learning for the content information and collaborative filtering for the ratings matrix.

However, the existing achievements seldom consider users' tagging behavior changes with time. In fact, the tagging behavior varies during different time. For example, the user might tag resources frequently during a period, the user might tag resources occasionally during a period, or the user might seldom tag resources during a period. Thus, this paper studies the tag recommendation method by considering the fact that users' tagging behavior changes with time.

3 Basic concepts

3.1 Social tagging system

Folksonomy (Vander Wal 2007), a term coined by Thomas Vander Wal in 2004, is the basic data structure of the social tagging system.

Formally, a folksonomy is a quadruple $\mathbb{F} = (U, R, Tag, Y)$, where $U = \{u_1, \cdots, u_k, \cdots, u_K\}$, $R = \{r_1, \cdots, r_l, \cdots, r_L\}$ and $Tag = \{tag_1, \cdots, tag_m, \cdots, tag_M\}$ are finite sets, whose elements are called users, resources and tags, respectively; K, L, and M are the numbers of users, resources, and tags, respectively. Y is a ternary relation among them, that is $Y \subseteq U \times R \times Tag$.

The ternary relation Y can be transferred to three binary relations, and each binary relation can be described by a matrix. That is, matrices $\mathbf{UTag}_{K\times M}$, $\mathbf{RTag}_{L\times M}$, and $\mathbf{UR}_{K\times L}$ represent the user-tag, the resource-tag and the resource-tag relations, respectively. Set the element of $\mathbf{UTag}_{K\times M}$ be $w_{u_k tag_m}$, where $w_{u_k tag_m}$ represents the number of resources which are labeled as the tag tag_m by the user u_k. Set the element of $\mathbf{RTag}_{L\times M}$ be $w_{r_l tag_m}$, where $w_{r_l tag_m}$ represents the number of users who use the tag tag_m to label the resource r_l. Set the element of $\mathbf{UR}_{K\times L}$ be $w_{u_k R_l}$, where $w_{u_k R_l}$ represents the number of tags which are labeled by the user u_k to the resource r_l.

Let Tag_{u_k} be the set of tags used by the user u_k, and Tag_{r_l} be the set of tags assigned to the resource r_l. Each post a of the folksonomy consists of three parts: a user u_k, a resource r_l and all tags in $Tag(u_k, r_l)$. That is, $a = (u_k, r_l, Tag(u_k, r_l))$. $Tag(u_k, r_l)$ is a set of tags that the user u_k has assigned to the resource r_l. All posts of the social tagging system constitute the post set A.

For a given user $u_q \in U$ and a given resource $r_q \in R$ with $Tag(u_q, r_q) \neq \varnothing$, the task of a tag recommendation is to recommend a set of tags, $\widehat{Tag}(u_q, r_q)$, with a tag recommendation algorithm, where $\widehat{Tag}(u_q, r_q) \subseteq Tag$. In many cases, $\widehat{Tag}(u_q, r_q)$ is computed by generating a ranking on the set of tags according to some quality or relevance criterion, from which then the top n elements are selected and recorded in $\widehat{Tag}^n(u_q, r_q)$.

3.2 Statistical language model

Statistical language model (Ponte and Croft 1998) (abbreviated as SLM) is widely used in natural language processing fields, such as speech recognition, information retrieval and machine translation. Essentially, it is a probability distribution model, mainly describes the inherent laws of statistics and structure of natural language. The set of all strings is a language, and a language model is called the probability distribution model of strings in the language.

In the field of information retrieval, the basic idea of statistical language model is to explain the correlation between a query q and a document d to produce a probability model of query from the document, i.e. $p_{LM}(q|d) = \prod_{w \in q} p(w|d)$, where w is a word of the query, and $p(w|d)$ is the probability of querying the word w from the document d, which is calculated as follows:

$$p(w|d) = \frac{N_d}{N_d + \lambda} \times \frac{tf(w,d)}{N_d} + \left(1 - \frac{N_d}{N_d + \lambda}\right) \times \frac{tf(w,D)}{N_D}, \tag{1}$$

where N_d is the length of the document d with the word as the unit, $tf(w, d)$ is the word frequency of w in the document d, N_D is the total number of words in all the documents, $tf(w, D)$ is the word frequency of w in all the documents, λ is a Dirichlet smoothing factor whose value is set to be the average document length in the document set, i.e. $\lambda = N_d/N_D$.

4 User tagging status

4.1 Related definitions

Let us observe the change of total numbers of tags that the user owned during a period of time T. Let the start moment be T_0, and we take equal interval as observation points (in the following experiments, the period of a month is chosen as a unit of time), then the next moment is T_1. Suppose the current moment to be T_t.

The set $Tag^{u_k T_t}$ consists of different tags used by the user u_k in a unit time interval, i.e. $[T_{t-1}, T_t)$. The $f_{u_k}(T_t)$ indicates the number of tags used by the user u_k in $[T_{t-1}, T_t)$, that is:

$$f_{u_k}(T_t) = |Tag^{u_k T_t}|. \tag{2}$$

The $g_{u_k}(T_t)$ is the number of tags used by the user u_k in the time interval $[T_0, T_t)$, that is:

$$g_{u_k}(T_t) = |\bigcup_{\tau=0}^{t} Tag^{u_k T_\tau}|. \tag{3}$$

For the user u_k, the tags used before the moment T_{t-1} are called the historical tags of the user at the moment T_t. Obviously, the number of historical tags is $g_{u_k}(T_t) = |\bigcup_{\tau=0}^{t-1} Tag^{u_k T_\tau}|$.

The tags, which have not used before the moment T_{t-1} but used in $[T_{t-1}, T_t)$, are called the new tags for the user. The number of new tags is $g_{u_k}(T_t) = |\bigcup_{\tau=0}^{t} Tag^{u_k T_\tau} \setminus \bigcup_{\tau=0}^{t-1} Tag^{u_k T_\tau}|$.

Let us observe what will happen during the time period $[T_{t-1}, T_t)$, the user u_k may tag or not. Thus, we should discuss in two cases:

Case 1: the user u_k has no tagging behavior, i.e. $f_{u_k}(T_t) = 0$.

Case 2: the user u_k has tagging behavior, i.e. $f_{u_k}(T_t) \neq 0$.

In the Case 2, we need to consider three aspects:

The u_k uses both new tags and historical tags, i.e. $0 < \frac{g_{u_k}(T_t) - g_{u_k}(T_{t-1})}{f_{u_k}(T_t)} < 1$.

The u_k uses only new tags, i.e. $\frac{g_{u_k}(T_t) - g_{u_k}(T_{t-1})}{f_{u_k}(T_t)} = 1$.

The u_k uses only historical tags, i.e. $g_{u_k}(T_t) - g_{u_k}(T_{t-1}) = 0$.

During a period of time, when the total of tags which the user owns (i.e. the total of different tags used by a user to tag resources) grows slowly or rapidly, it is certainly that the user uses new tags and it is possible that the user uses historical tags. When the total of tags which the user owns remains unchanged, the user only uses historical tags or the user has no tagging behavior. In other words, the user's tagging status have three cases during a period of time: the first case is the scenario that a user's total number of tags increases rapidly; the second case is the scenario that a user's total number of tags increases slowly; the third case is the scenario that a user has no tagging behavior. Therefore, we defined the three cases as users' tagging status as the growing status, the mature status and dormant status respectively.

In the period of time $T = [T_{t-\Delta t}, T_t)$, if the total number of tags which a user owns is increased and the average growth rate of the total number of tags is no less than the threshold α, we call that the user is in the growing status. That means, during the period of time, the user is quite active, and adds many new tags into the social tagging system.

Definition 1 (**Growing Status**) Considering the period of time $[T_{t-\Delta t}, T_t)$, for a user u_k, if $\frac{g_{u_k}(T_{t-1}) - g_{u_k}(T_{t-\Delta t})}{\Delta t} \geq \alpha$, then the user tagging status of the user u_k at the time T_t is the growing status.

In the period of time $T = [T_{t-\Delta t}, T_t)$, if the total number of tags a user owned is increased and the average growth rate of the total number of tags is less than the threshold α, we call that the user is in the mature status. That means, during the period of time, the user adds a few new tags into the social tagging system and also uses many historical tags.

Definition 2 (**Mature Status**) Considering the period of time $[T_{t-\Delta t}, T_t)$, for a user u_k, if $\exists T_{t'} \in [T_{t-\Delta t}, T_t)$ brings $f_{u_k}(T_{t'}) \neq 0$ and $0 \leq \frac{g_{u_k}(T_{t-1}) - g_{u_k}(T_{t-\Delta t})}{\Delta t} < \alpha$, then the user tagging status of user u_k at the time T_t is the mature status.

In the period of time $T = [T_{t-\Delta t}, T_t)$, if a user has no tagging behavior and the total number of tags the user remains constant, we call that the user is in the dormant status.

Definition 3 (**Dormant Status**) Considering the period of time $[T_{t-\Delta t}, T_t)$, for a user u_k, if $\forall T_{t'} \in [T_{t-\Delta t}, T_t)$ brings $f_{u_k}(T_{t'}) = 0$, then the user tagging status of user u_k at the time T_t is the dormant status.

4.2 Determining user tagging status algorithm

Suppose the current time is T_t. We can determine the user tagging status at the moment T_t according to Definition 1, Definition 2 and Definition 3, by analysing the tagging history of the user u_k during the period of time Δt.

Then, the determining user tagging status algorithm, abbreviated as DUTS, is described in Algorithm 1. Here, the T_0 is the moment when the user begins to use the social tagging system. If the duration that user u_k uses the social tagging system is less than Δt, and has tagging behavior recently, we think the user is in the growing status. Because everyone is personalized, the duration that a user in different tagging status is different. In order to simplify the calculation, the determining user tagging status algorithm only backs the user's tagging history to a fixed period of time.

Algorithm 1 DUTS: determining user tagging status algorithm

Input: the current moment T_t ; the start moment T_0; the threshold value α and the length of the time period Δt;
Output: the user tagging status, s, of user u_k at the moment T_t;
Step 1: (**Initialization**)
BOOL $hastag = false$; //whether the user has tagging behavior or not
for each $mt \in [T_0, T_{t-1}]$ **do**
    $f_{u_k}(mt)$ is computed as Eq. (2);
end for;
$g_{u_k}(T_{t-1})$ and $g_{u_k}(T_{t-\Delta t})$ are computed as Eq. (3);
Step2:(**Determine the user tagging status**);
if $T_t - T_0 < \Delta t$ **then** // Calculate under the situation that $T_t - T_0 < \Delta t$
    for each $mt \in [T_0, T_{t-1}]$ **do**
        if $f_{u_k}(mt) \neq 0$ **then**
            $hastag = true$;
            BREAK;
        end if
    end for
    if $hastag = true$ **then**
        s = the Growing Status;
        Return s;
    else
        s = the Dormant Status;
        Return s;
    end if
else// Calculate under the situation that $T_t - T_0 \geq \Delta t$
    BOOL $hastag = false$;
    for each $mt \in [T_{t-\Delta t}, T_{t-1}]$ **do**
        if $f_{u_k}(mt) \neq 0$ **then**
            $hastag = true$;
            BREAK;
        end if
    end for
    if $hastag = false$ **then**
        s = the Dormant Status;
        Return s;
    else//Calculate $differT$
        $differT = \frac{g_{u_k}(T_{t-1}) - g_{u_k}(T_{t-\Delta t})}{\Delta t}$;
        //judge the user tagging status according to $differT$ and
        if $differ \geq \alpha$ **then**
            s = the Growing Status;
            Return s;
        else
            s= the Mature Status;
            Return s;
        end if
    end if
end if

5 Tag recommendation method based on user tagging status

The Fig. 1 shows the framework of tag recommending model proposed in this paper. Once the user's user tagging status is determined, we can employ different strategy to recommend tags for the user. Algorithm 2 describes the processing of the tag recommendation algorithm based on user tagging status, abbreviated as TR-UTS. First, the algorithm computes the user tagging status of the user at the moment T_t by using Algorithm 1. Then, the algorithm determines tag recommendation strategy by calculating tag probability distributions according to the user's tagging status. Finally, the top n tags, most likely to be used by the user, are recommended.

Algorithm 2 TR-UTS: tag recommendation algorithm considering user tagging status

Input: the current moment of the target user u_q, T_t ; the start moment of u_q, T_0; the threshold value: α; the length of the time period, Δt; the tags used by u_q and his/her group members; the tags labelled to the target resource r_q and its similar resources.

Output: the recommended tag set, $\widehat{Tag}^n(u_q, r_q)$.

Step1: Determine the user tagging status of user u_k at the current moment T_t, according to Algorithm 1.
Step2: Recommend tags for user u_q.
if the user tagging status is the Growing Status **then**
    Recommend tags with the strategy **TR-GS** described in the Subsection 5.1;
end if
if the user tagging status is the Mature Status **then**
    Recommend tags with the strategy **TR-MS** described in the Subsection 5.2;
end if
if the user tagging status is the Dormant Status **then**
    Recommend tags with the strategy **TR-DS** described in the Subsection 5.3;
end if
Return $\widehat{Tag}^n(u_q, r_q)$;

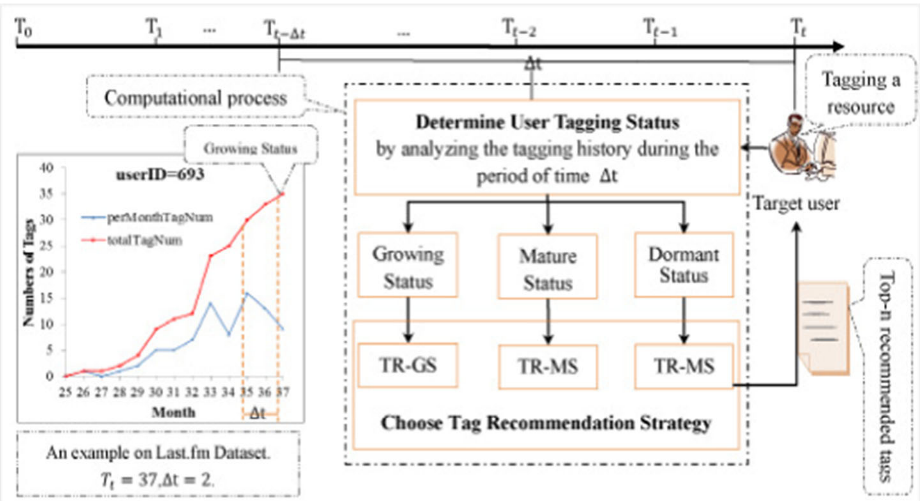

Fig. 1 The framework of tag recommendation model based on user tagging status

A few additional explanations to the proposed method. Question 1: how to obtain the user u_q's group members. It had been proved to be very helpful to improve the accuracy of recommendation by utilizing the group information. Since it is not the point of this paper, we simply think the friendship existing among the folksonomy as the user's group information. For example, each user has an average of 13.443 friends in the Last.fm; for the CiteULike, there also exists "group id | username" information. Anyway, to propose an appropriate clustering method on users should be our further work in order to further enhance the flexibility of the proposed method. Question 2: how to obtain the target resource r_q's similar resources. For this question, we will give the detail description in the Section 5.1.

5.1 The strategy for user tagging status in growing status

Considering a given user u_q to tag a given resource r_q at the current time, the user's user tagging status is the growing status, which means the number of resources tagged by the user is increasing continually during a period of time before the current time, and the total number of tags used by the user is increasing continually, too. Therefore, it is helpful to enhance the performance of recommendation by considering the following two kinds tags: (1) one kind of tags is the tags used by the target user and his/her group members; and (2) the other kind of tags is the tags to label the target resource and their similar resources. Then, we can compute those tags' probability distribution with SLM to recommending tags.This approach not only ensures recommending personalized tags, but also increases the diversity of recommended tags.

The strategy for the user tagging status in growing status, abbreviated as TR-GS, is described as follows.

Step1: according to the resource-tag matrix $\mathbf{RTag}_{L\times M}$, to calculate the similarity between the resource r_q and other resources based on the cosine similarity, and select the top S resources with highest similarity to r_q as the neighbor set of the resource r_q.

Set a row of the resource-tag matrix $\mathbf{RTag}_{L\times M}$ be the vector $\mathbf{r}$. Then, the similarity $sim(r_l, r_q)$ between r_l and r_q is computed as follows:

$$sim(\mathbf{r}_l, \mathbf{r}_q) = \frac{\mathbf{r}_l \cdot \mathbf{r}_q}{\parallel \mathbf{r}_l \parallel \parallel \mathbf{r}_q \parallel}. \tag{4}$$

Step2: considering all tags labeled for the resource r_q and its neighbors, to compute the tag probability distribution $p(tag_m \mid r_q)$ according to the following equation:

$$\begin{aligned} p(tag_m \mid r_q) = & \frac{N_{Tag_{r_q}}}{N_{Tag_{r_q}} + \lambda_{r_q}} \times \frac{TF(tag_m, Tag_{r_q})}{N_{Tag_{r_q}}} \\ & + \left(1 - \frac{N_{Tag_{r_q}}}{N_{Tag_{r_q}} + \lambda_{r_q}}\right) \times \frac{TF(tag_m, TagS)}{N_{TagS}}, \end{aligned} \tag{5}$$

where $TF(tag_m, Tag_{r_q})$ is the number of users who use the tag_m to label the r_q, namely, $TF(tag_m, Tag_{r_q}) = w_{r_q tag_m}$. $N_{Tag_{r_q}}$ is the sum of weights of tags of resource r_q. $TagS$ is the set of tags labeled to the resource r_q and its neighbors, and for $\forall tag \in TagS$, its tag weight $w'_{r_k tag_m} = w_{r_k tag_m} \times sim(\mathbf{r}_k, \mathbf{r}_q)$. N_{TagS} is the sum of weights of tags in the set $TagS$. $TF(tag_m, TagS)$ is the sum of weights of the tag tag_m labeled to the resource r_q and its neighbors. λ_{r_q} is interpreted as a Dirichlet smoothing factor, i.e. $\lambda_{r_q} = N_{Tag_{r_q}}/N_{TagS}$.

Step3: considering all the tags used by the user u_q and his/her group members, based on the user-tag matrix $\mathbf{UTag}_{K\times M}$, to compute the tag probability distribution $p(tag_m \mid u_q)$ according to the following equation:

$$p(tag_m \mid r_q) = \frac{N_{Tag_{u_q}}}{N_{Tag_{u_q}} + \lambda_{u_q}} \times \frac{TF(tag_m, Tag_{u_q})}{N_{Tag_{u_q}}} + \left(1 - \frac{N_{Tag_{u_q}}}{N_{Tag_{u_q}} + \lambda_{u_q}}\right) \times \frac{TF(tag_m, Tag_{U_q})}{N_{Tag_{U_q}}}, \tag{6}$$

where $N_{Tag_{u_q}}$ is the sum of tag weights of tags user u_q used. $TF(tag_m, Tag_{u_q})$ is the tag weight of tag_m the user have used, namely, $TF(tag_m, Tag_{u_q}) = w_{u_q tag_m}$. The set U_q consists of user u_q and users in the same groups with u_q. Tag_{U_q} is the set of tags used by user u_q and users in the same groups with u_q. $N_{Tag_{U_q}}$ is the sum of tag weights of tags in the set Tag_{U_q}. $TF(tag_m, Tag_{U_q})$ is the sum of tag weights of the tag tag_m used by users in the set U_q. λ_{u_q} is a Dirichlet smoothing factor, i.e. $\lambda_{u_q} = N_{Tag_{u_q}} / N_{Tag_{U_q}}$.

Step4: compute the possibility of the user u_q use the tag tag_m to label the resource r_q, $p(tag_m \mid r_q)$ and $p(tag_m \mid u_q)$, according the following equation:

$$p(tag_m \mid u_q, r_q) = (1-\beta) \times p(tag_m \mid u_q) + \beta \times p(tag_m \mid r_q). \tag{7}$$

where $\beta \in [0, 1]$.

Step5: sort the tags according to the probability $p(tag_m \mid u_q, r_q)$, and then select the top n elements with the highest rank values to recommend to the user u_q, that is,

$$\widehat{Tag}^{n}(u_q, r_q) = \max_{tag_m \in Tag}^{n} (p(tag_m \mid u_q, r_q)).$$

5.2 User tagging status in mature status

When the given user u_q tags the given resource r_q, if at the moment the user's user tagging status is the mature status, then during the period of time before the moment, the user's tagging behavior tends to be stable, and the amount of resources achieves a certain number; thus, the total number of the user's tags has increases slowly. We compute those tags' tag probability distribution with SLM based on the user's tags and the resource's tags. This approach not only ensures the accuracy of tag recommendation, but also reduces the computation complexity.

The strategy for the user tagging status in mature status, abbreviated as TR-MS, is described as follows.

Step1: for $\forall tag_m \in Tag_{u_q}$, the probability $p_u(tag_m \mid u_q)$ that the user u_q will use tag_m is calculated as follows:

$$p_u(tag_m \mid u_q) = \frac{w_{u_q tag_m}}{N_{Tag_{u_q}}}, \tag{8}$$

where, $N_{Tag_{u_q}}$ is the sum of tag weights of tags used by u_q, namely, $N_{Tag_{u_q}} = \sum_{tag \in Tag_{u_q}} w_{u_q tag}$.

Step2: for $\forall tag_m \in Tag_{r_q}$, the probability $p_r(tag_m \mid r_q)$ that the resource r_q will be labeled by tag_m is calculated as follows:

$$p_r(tag_m \mid r_q) = \frac{w_{r_q tag_m}}{N_{Tag_{r_q}}}, \tag{9}$$

where, $N_{Tag_{r_q}}$ is the sum of tag weights of tags labeled to r_q, namely, $N_{Tag_{r_q}} = \sum_{tag \in Tag_{r_q}} w_{r_q tag}$.

Step3: calculate the $p(tag_m \mid u_q, r_q)$, the probability that a given tag tag_m will be used by the given user u_q to label the given resource r_q, using a weighted linear combination of $p_u(tag_m \mid u_q)$ and $p_r(tag_m \mid r_q)$, as follows:

$$\begin{aligned} p(tag_m \mid u_q, r_q) = {} & (1-\gamma) \times p_u(tag_m \mid u_q) \\ & +\gamma \times p_r(tag_m \mid r_q), \end{aligned} \tag{10}$$

where, $tag_m \in (Tag_u \cup Tag_r)$, and $\gamma \in [0, 1]$.

Step4: sort the tags according to the probability $p(tag_m \mid u_q, r_q)$, and select the top n elements to recommend to the user u_q, that is:

$$\widehat{Tag}^n(u_q, r_q) = \max_{tag_m \in Tag}^{n} (p(tag_m \mid u_q, r_q)).$$

5.3 User tagging status in dormant status

When the given user u_q tags the given resource r_q, if at the moment the user's user tagging status is the dormant status, then during the period of time before the moment, this user did not tag. Thus, we compute the tag probability distribution with SLM using tags labeled to the resource r_q and its similar resources.

The strategy for the user tagging status in dormant status, abbreviated as TR-DS, is described as follows.

Step1: estimate the probability $p(tag_m \mid r_q)$ that the tag tag_m will be labeled to the resource r_q using the same method used in the Section 5.1.

Step2: sort the tags according to the probability $p(tag_m \mid r_q)$, and then select the top n elements with the highest rank values to recommend to the user u_q, that is:

$$\widehat{Tag}^n(u_q, r_q) = \max_{tag_m \in Tag}^{n} (p(tag_m \mid r_q)).$$

6 Experiments

In this session, we conducted various experiments to evaluate and analyze the effectiveness and efficiency of the proposed method on some datasets. In the first set of runs, we gave examples with purposes of assessing the effectiveness of the DUTS algorithm. In the second set of runs, we obtained experimentally the threshold values used in the TR-UTS algorithm and the most popular tags algorithm (Jäschke et al. 2008). In the third set of runs, we gave some results of TR-UTS, TR-GS, TR-MS and TR-DS. In the forth set of runs, we compared the proposed TR-UTS algorithm to other algorithm as FolkRank (Kim and El Saddik 2011), LocalRank (Kubatz et al. 2011) and the most popular tags ρ-mix (Jäschke et al. 2008). But before reporting these experimental results, we need to introduce the dataset preprocessing and the evaluation criteria that we adopt.

Table 1 Some information of datasets after preprocessing

Statistical Information	CiteULike(k=30)	Last.fm(k=10)
$\|U\|$, the number of users	1700	966
$\|R\|$, the number of resources	32208	3870
$\|Tag\|$, the number of tags	6012	1204
$\|A\|$, the number of posts	89076	48578
$\|Y\|$, the number of ternary relations	1507781	133945
Date interval	2004/11/04 $\sim$ 2012/10/16	2005/08/01 $\sim$ 2011/05/09
Average posts of per user	52	50
Average tags of per posts	17	3
Average distinct tags used by each user	113	20
Average distinct tags used for each resource	38	16

6.1 Dataset preprocessing

There are two datasets of social tagging systems used in experiments, that is, the CiteULike[1], and the Last.fm[2]. CiteULike is a web service which allows users to save and share citations to academic papers. Users can organize their libraries with freely chosen tags and this produces a folksonomy of academic interests. Last.fm is a music website, the site offers numerous social networking features and can recommend and play artists similar to the user's favourites. Though there is no palpable information for the user is belong to which group, there exists friendships between users in these two folksonomies. Thus, we can find the friends from the original data and set the user and his (her) friends into a group.

The original datasets are too sparse to be used for experiments. Therefore, the *p*-core of level k algorithm (Batagelj and Zaveršnik 2011) is applied to the datasets so that every user, every resource and every tag appear at least k times in the processed datasets. The statistical information after preprocessing are shown in Table 1. The first column denotes some statistical information of the corresponding dataset, the second column describes the statistical information on CiteULike when k=30, and the third column presents the statistical information on Last.fm when k=10.

In the dataset of CiteULike, the data between 2004/11/04 and 2012/09/30 are chosen as the train set, the data between 2012/10/01 and 2012/10/16 as the test set. In the dataset of Last.fm, we choose the data between 2005/08/01 and 2011/02/28 as the train set, and the data between 2011/03/01 and 2011/05/09 as the test set. The train sets are used to show that the statistical results are coincident with the results of the DUTS algorithm in Section 6.3.1, and used to determine parameters in Section 6.3.2; the other experiments are carried out in the test sets.

6.2 Evaluation criteria

To measure the recommendation quality, we adopt the standard evaluation criteria in the information retrieve field as the *recall* ($R@n$), the *precision* ($P@n$) and the *F-measure*

[1] CiteULike Datasets, http://www.citeulike.org/faq/data.adp2012,10,17

[2] HetRec 2011 Data Sets, http://grouplens.org/datasets/hetrec-2011/2012,10,1

($F@n$) at the Top-n (Kim and El Saddik 2011), where n is the length of recommended tags set.

Let U be the data set. For $u_q \in U$, the tag post, $a = (u_q, r_q, Tag(u_q, r_q))$, is created when the user u_q annotates a resource r_q with a set of tags $Tag(u_q, r_q)$. That is, $Tag(u_q, r_q)$ is the set of tags that user u_q tagged the resource r_q in the data. The $\widehat{Tag}^n(u_q, r_q)$ is the set of top-n recommended tags for the user-resource pair.

Recall, is a common metric for evaluating the utility of recommendation algorithms and a measure of completeness. *Precision* is another common metric for measuring the usefulness of recommendation algorithms and a measure of exactness. For a given user-resource pair (u_q, r_q), when the size of the recommended tag set is n, $R@n$ measures the percentage of tags in the tag set of the corresponding post that appear in the recommended tag set. $P@n$ measures the percentage of tags in the recommended tag set that appear in the tag set of the corresponding post. So, $R@n$ and $R@n$ are defined as follows:

$$R@n = \frac{|Tag(u_q, r_q) \cap \widehat{Tag}^n(u_q, r_q)|}{|Tag(u_q, r_q)|}, \tag{11}$$

$$P@n = \frac{|Tag(u_q, r_q) \cap \widehat{Tag}^n(u_q, r_q)|}{|\widehat{Tag}^n(u_q, r_q)|}. \tag{12}$$

$P@n$ and $R@n$ will be influenced by the n; for example, the bigger n is, the bigger is $R@n$ but the smaller is $P@n$. Therefore, we adopt $F@n$ as the measurement, which is the harmonic mean of $P@n$ and $R@n$ and defined as follows:

$$F@n = \frac{2 \times R@n \times P@n}{R@n + P@n}. \tag{13}$$

The overall *recall*, *precision* and *F-measure* for all users are computed by averaging the individual precisions and recalls, respectively.

6.3 Experimental results

The experiment contains three parts: the first part shows examples of determining user tagging status in the train sets; the second part calculates parameters in our tag recommendation algorithm as well as ρ in the most popular tags ρ-mix algorithm in the train sets; the third part compares the result of TR-UTS with TR-GS, TR-MS and TR-DS, along with the comparison between TR-UTS and FolkRank, LocalRank, and the most popular tags ρ-mix algorithm in the test sets.

6.3.1 Results of the determining user tagging status algorithm

Firstly, we calculate user tagging status using the determining user tagging status (DUTS) algorithm proposed in Section 4.2. The period of a month is chosen as the unit of time in our experiments. For the CiteULike dataset, the data between 2005/08/04 and 2012/09/30 (a total of 86 months) are considered, the time interval is $[T_0 = 1, T_t = 86]$, and the time cell is 1 month. For the Last.fm dataset, the data between 2005/10/01 and 2011/02/28 (a total of 65 months) are considered, the time interval is $[T_0 = 1, T_t = 65]$, and the time cell is 1 month.

For the space limit, Tables 2 and 3 show examples of the results of the DUTS algorithm. That is, Table 2 shows the user tagging status calculated by the DUTS algorithm for the user ID=103 in the CiteULike during this period; Table 3 shows the corresponding results of the user ID=410 in the Last.fm dataset. The thresholds used here is determined by experience,

Table 2 The results of DUTS algorithm for the user ID=103 in the CiteULike dataset

Month	1	3	5	7	9	16	26	36	46	56	66	76	86
Tagging status	DS	DS	DS	GS	GS	GS	GS	GS	GS	MS	MS	MS	MS

The DS, GS, and MS are short for the dormant status, growing status, and mature status, respectively

because the different folksonomy has the different characteristic. That is, for the CiteULike dataset, $\Delta t = 4$ and $\alpha = 3$; for the Last.fm dataset, $\Delta t = 2$ and $\alpha = 2$.

As a check, we also count the total number of tags used by users over a period of time on the datasets. Figures 2 and 3 as examples give the statistical result of the total number of tags used by the user ID=103 in CiteULike dataset and user ID =410 in Last.fm dataset respectively. The $totalTagNum$ curve shows the total numbers of tags used by the user from the 1st month to the current t-th month; that is, the $totalTagNum$ represents $g_{u_q}(T_t)$. The $perMonthTagNum$ curve shows the total numbers of tags used by the user in every time cell; that is, the $perMonthTagNum$ represents the $f_{u_q}(T_t)$.

Let us observe the results in Table 2 and Fig. 2. First, let us see Fig. 2, it is clear that the user ID=103 does not tag from the 1st month to the 5th month, whose total number of tags is zero. The statistical results show the user's tagging status is the third case (namely the dormant status) during this period. From the 6th month to the 56th month, the $totalTagNum$ (user's total number of tags) increases, and the increasing is sharply especially after the 21th month; which shows the user's tagging status is the first case (namely the growing status) during this period. From the 57th month, the user's total number of tags increases slowly, but not zero for the most part; which shows the user's tagging status is the second case (namely the mature status) during this period. Then, let's observe the results in Table 2, which give almost the same results as the statistical results. Similarly, the results in Table 3 and Fig. 3 are also coincident.

Furthermore, more statistical results show that the results of the DUTS algorithm are coincident with the facts in most cases, which means the DUTS algorithm indeed finds out the tagging status of users.

6.3.2 Calculating parameters

According to the DUTS algorithm, we divide the train set into three parts: the growing subset, the mature subset and the dormant subset. Then, we conduct the corresponding tag recommendation policy for these three subset with different parameters, and record the thresholds when $F@n$ is the best one.

Meanwhile, we also conduct the most popular tags ρ-mix algorithm in Jäschke et al. (2008) with different values of ρ in the train set to find the proper ρ which makes the best $F@n$.

Table 3 The results of DUTS algorithm for the user ID=410 in the Last.fm Dataset

Month	1	5	9	15	19	24	29	34	39	43	49	54	59
Tagging status	DS	DS	DS	GS	GS	GS	GS	MS	MS	GS	GS	MS	MS

The DS, GS, and MS are short for the dormant status, growing status, and mature status, respectively

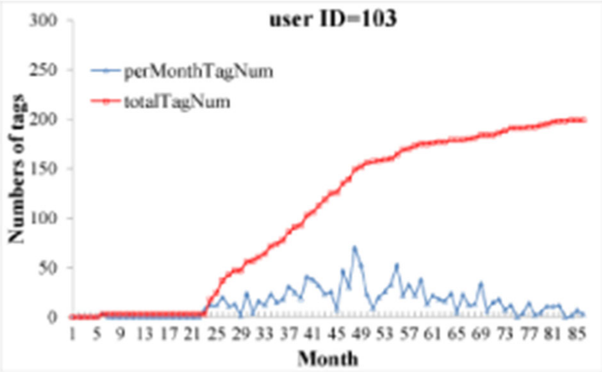

Fig. 2 The statistical results of numbers of tags used by user ID=103 in the CiteULike dataset

The detailed experimental processing is given only on the CiteULike dataset in this subsubsection; for the Last.fm dataset, we just give the final results of parameters.

(1) Results of TR-GS in the Growing Subset

We adopt the TR-GS algorithm to recommend tags for the each user-resource pair (u_q, r_q) in the growing subset. And, the number of neighbors of the resource r_q, S_1 is set to be 5, 10, 15, 20, 25, 35, 45 or 55, and the β is set to be 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, or 1, respectively. In order to save space, Table 4 just shows the performance of tag recommendation under some parameters in the growing subset.

As indicated in Table 4, the performance of tag recommendation changes with the value of S_1 or β. When $S_1 = 45$ and $\beta = 0.6$, $F@n$ of TR-GS is the best; that is, the performance of tag recommendation is best.

(2) Results of TR-MS in the Mature Subset

We adopt the TR-MS algorithm to recommend tags to the each user-resource pair (u_q, r_q) in the mature subset. In this experiment, γ is set to be 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9 or 1, respectively. To make the result more clearly, Table 5 shows the performance of tag recommendation when γ is 0.3, 0.4, 0.5, 0.6 ,0.7 and 0.8.

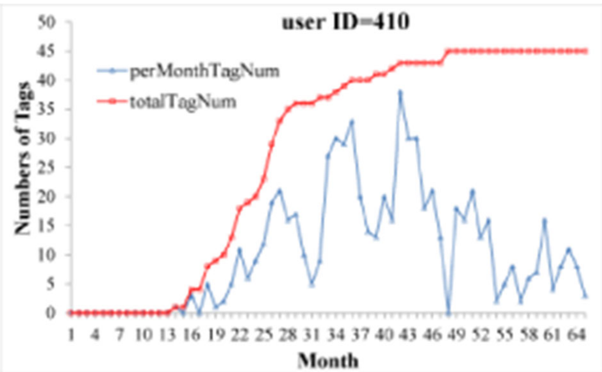

Fig. 3 The statistical result of numbers of tags used by user ID=410 in the Last.fm dataset

Table 4 Results of TR-GS in the growing subset with (S_1, β)

$F@n$ \ (S_1,β)	(15,0.4)	(25,0.5)	(45,0.6)	(25,0.7)	(25,0.8)	(25,0.9)
$F@1$	0.2921	0.2921	0.2956	0.2921	0.2921	0.2921
$F@2$	0.3690	0.4202	0.3898	0.4202	0.4202	0.4202
$F@3$	0.4013	0.4198	0.4277	0.4198	0.4198	0.4198
$F@4$	0.3729	0.4047	0.4132	0.4047	0.4047	0.4047
$F@5$	0.3675	0.3675	0.3811	0.3550	0.3550	0.3550

As indicated in Table 5, different γ would bring different influence result. When $\gamma = 0.5$, the $F@n$ of TR-MS is the best one.

(3) Results of TR-DS in the Dormant Subset

We adopt the TR-DS algorithm to recommend tags to the each user-resource pair (u_q, r_q) in the dormant subset. In this experiment, the number of neighbors S_2 of resource $r_(q)$ was set to be 5, 10, 15, 20, 25, 35, 45, or 55, respectively. Table 6 shows the performance of tag recommendation on the dormant subset.

As indicated from Table 6, different S_2 would bring different results. When $S_2 = 10$, $F@n$ of TR-DS is best and the performance of tag recommendation is best.

(4) Results of the Most Popular Tag Algorithm

We adopt the most popular tags ρ-mix algorithm to recommend tags to the each user-resource pair (u_q, r_q) in the train set. According to Jäschke et al. (2008), ρ is set to be 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, and 1 respectively. To make the results more clearly, Table 7 shows the performance of tag recommendation when ρ is 0.5, 0.6, 0.7, 0.8, 0.9 and 1.

As indicated in Table 7, different ρ leads to different result. When $\rho = 0.9$, $F@n$ is best and the performance of tag recommendation is best.

In conclusion, we can determine the appropriate parameter used in the corresponding subset of CiteULike according to the results from $F@1$ to $F@5$. For the growing subset, $S_1 = 45$ and $\beta = 0.6$; for the mature subset, $\gamma = 0.5$; for the dormant subset, $S_2 = 10$; and for the most popular tags algorithm, $\rho = 0.9$.

In the same way, the best parameters of the Last.fm dataset are determined, too. That is, $S_1 = 5$ and $\beta = 0.9$ for the growing subset; $\gamma = 0.5$ for the mature subset; $S_2 = 5$ for the dormant subset; and $\rho = 1.0$ for the most popular tags algorithm.

The parameters used in the FolkRank algorithm proposed in Kim and El Saddik (2011) are set as the same as used in the reference, i.e., $d = 0.7$, and the number of iteration is 10, and the corresponding weight of preference vector is set to be $1 + |U|$ and $1 + |R|$.

Table 5 Results of TR-MS in the mature subset with γ

$F@n$ \ γ	0.3	0.4	0.5	0.6	0.7	0.8
$F@1$	0.3078	0.3894	0.4549	0.4549	0.4262	0.3683
$F@2$	0.3994	0.4439	0.4238	0.4367	0.4164	0.3864
$F@3$	0.3772	0.4181	0.3988	0.3928	0.3918	0.3698
$F@4$	0.3704	0.3799	0.3819	0.3708	0.3714	0.3491
$F@5$	0.3489	0.3445	0.3608	0.3453	0.3413	0.3221

Table 6 Results TR-DS in the dormant subset with S_2

$F@n$ \ S_2	5	10	15	25	35	45	55
$F@1$	0.3311	0.3311	0.3311	0.3311	0.3311	0.3311	0.3311
$F@2$	0.3543	0.3415	0.3415	0.3300	0.3300	0.3300	0.3300
$F@3$	0.3201	0.3401	0.3316	0.3316	0.3222	0.3222	0.3222
$F@4$	0.3151	0.3151	0.3151	0.3134	0.3134	0.3134	0.3134
$F@5$	0.2954	0.2954	0.2937	0.2937	0.2867	0.2867	0.2867

6.3.3 Results and analysis

To make the comparison objectively, we use the best parameter calculated above for TR-UTS and the most popular tags ρ-mix algorithm in the following experiments.

(1) Validation of User Tagging Status

To validate the effectiveness of the proposed definition of user tagging status, we compare the results of TR-UTS, TR-GS, TR-MS and TR-DS in the test set of CiteULike and Last.fm datasets. Tables 8 and 9 describe the comparison results.

From Table 8, we can see that the $F@n$ of TR-UTS is better than TR-GS and TR-MS, and is much better than TR-DS. The mean value of $F@n$ of TR-UTS is heightened 1.71%, 2.58% and 21.78% than the other three algorithm respectively. This indicates that the tag recommendation strategies proposed in this paper, determining user tagging status firstly and choosing different tag recommendation strategies under different status, has a better performance than the single strategy. Meanwhile, the performance of TR-GS is slightly better than TR-MS, and much better than TR-DS, which infers that it is efficient to consider the tags of users in the same group and similar resources to recommend tags. From Table 9, we can see that the $F@n$ of TR-UTS is the best one when $n \in \{3, 4, 5\}$; when $n = 1$ or $n = 2$, the $F@n$ of TR-UTS is slightly less than the TR-GS, but much better than TR-MS and TR-DS.

In conclusion, the effectiveness of TR-UTS algorithm is much better than that of other three strategies. Therefore, we will adopt the proposed TR-UTS algorithm as the further comparison method with other existing methods in the following.

(2) Comparison Experiments

In order to observe the performance of the proposed method, we test the proposed TR-UTS algorithm with FolkRank (Kim and El Saddik 2011), LocalRank (Kubatz et al. 2011) and the most popular tags ρ-mix algorithm (Jäschke et al. 2008) in the CiteULike dataset and Last.fm dataset.

Table 7 Results of the most popular tag algorithm with ρ

$F@n$ \ ρ	0.5	0.6	0.7	0.8	0.9	1.0
$F@1$	0.2970	0.2984	0.3194	0.3242	0.3346	0.3253
$F@2$	0.3494	0.3478	0.3476	0.3540	0.3540	0.3294
$F@3$	0.3536	0.3363	0.3460	0.3526	0.3558	0.3271
$F@4$	0.3462	0.3364	0.3476	0.3476	0.3476	0.3104
$F@5$	0.3290	0.3276	0.3301	0.3301	0.3325	0.2873

Table 8 Results of TR-UTS, TR-GS, TR-MS and TR-DS on CiteULike dataset

Algorithm / $F@n$	TR-UTS	TR-GS	TR-MS	TR-DS
$F@1$	0.3972	0.3907	0.3938	0.3074
$F@2$	0.4064	0.4000	0.4004	0.3134
$F@3$	0.3854	0.3740	0.3712	0.3055
$F@4$	0.3634	0.3567	0.3474	0.2827
$F@5$	0.3370	0.3352	0.3287	0.2681

Table 9 Results of TR-UTS, TR-GS, TR-MS and TR-DS on Last.fm dataset

Algorithm / $F@n$	TR-UTS	TR-GS	TR-MS	TR-DS
$F@1$	0.2673	0.2707	0.2649	0.2652
$F@2$	0.3288	0.3346	0.3234	0.3249
$F@3$	0.3582	0.3564	0.3523	0.3448
$F@4$	0.3512	0.3499	0.3485	0.3419
$F@5$	0.3386	0.3386	0.3423	0.3290

Table 10 $P@n$ of algorithms on CiteULike dataset

Algorithm / $P@n$	TR-UTS	FolkRank	*Popular 0.9-mix*	LocalRank
$P@1$	0.6095	0.4486	0.5047	0.5794
$P@2$	0.4190	0.3738	0.3692	0.4112
$P@3$	0.3365	0.3115	0.3084	0.3271
$P@4$	0.2857	0.2593	0.2710	0.2827
$P@5$	0.2462	0.2243	0.2411	0.2486

Table 11 $R@n$ of algorithms on CiteULike dataset

Algorithm / $R@n$	TR-UTS	FolkRank	*Popular 0.9-mix*	LocalRank
$R@1$	0.2945	0.1941	0.2503	0.2864
$R@2$	0.3945	0.3293	0.3400	0.3839
$R@3$	0.4509	0.4086	0.4205	0.4449
$R@4$	0.4993	0.4557	0.4847	0.4979
$R@5$	0.5342	0.4880	0.5353	0.5524

Table 12 $F@n$ of algorithms on CiteULike dataset

Algorithm / $F@n$	TR-UTS	FolkRank	*Popular 0.9-mix*	LocalRank
$F@1$	0.3972	0.2710	0.3346	0.3834
$F@2$	0.4064	0.3502	0.3540	0.3971
$F@3$	0.3854	0.3535	0.3558	0.3770
$F@4$	0.3634	0.3306	0.3476	0.3606
$F@5$	0.3370	0.3073	0.3325	0.3429

Table 13 $P@n$ of algorithms on Last.fm dataset

Algorithm / $P@n$	TR-UTS	FolkRank	*Popular 1.0-mix*	LocalRank
$P@1$	0.3990	0.4166	0.4068	0.4088
$P@2$	0.3400	0.3532	0.3415	0.3449
$P@3$	0.3141	0.3034	0.2992	0.3067
$P@4$	0.2783	0.2722	0.2617	0.2749
$P@5$	0.2501	0.2472	0.2326	0.2500

To make the comparison objectively, the following experiments use the best parameters obtained in Section 6.3.2 for both the TR-UTS algorithm and the most popular tags ρ-mix algorithm. Thus, *Popular 0.9-mix* is short for the most popular tags ρ-mix algorithm.

(a) The comparison results of CiteULike dataset is shown in Tables 10, 11 and 12.

According to Tables 10 and 11, TR-UTS is slightly worse than LocalRank only at $P@5$ and $R@5$, but TR-UTS is much better than the other three contrastive algorithms at any other $P@n$ and $R@n$.

According to Table 12, the mean of $F@n$ of TR-UTS is 18.25%, 7.14%, 1.45% bigger than that of FolkRank, popular 0.9-mix and LocalRank respectively. Especially for $F@1$, the performance of TR-UTS is obviously better than the one of FolkRank and *Popular 0.9-mix*, with the increment is 46.56% and 20.29% respectively.

(b) The comparison results of Last.fm dataset is shown in Tables 13, 14 and 15.

According to Tables 13 and 14, the performance of TR-UTS is better than the other three contrastive algorithms at $R@n$. For $P@n$, the performance of TR-UTS is a bit worse than the comparison algorithms at $P@1$ and $P@2$, but better than the comparison algorithms at the other cases.

According to Table 15, the mean of $F@n$ of TR-UTS is 1.46%, 2.04%, 5.13% bigger than that of FolkRank, *Popular 1.0-mix* and LocalRank respectively. Especially for $F@3$, the increment of TR-UTS is 3.93%, 5.38% and 6.47% respectively.

To provide overall review of the experiments on CiteULike dataset and Last.fm dataset, Figs. 4 and 5 clearly show the performance of different tag recommendation algorithms on $F@n$.

The above experimental results indicate that TR-UTS is obviously better in tag recommendation performance than FolkRank, the most popular tags ρ-mix algorithm and LocalRank.

Table 14 $R@n$ of algorithms on Last.fm dataset

Algorithm / $R@n$	TR-UTS	FolkRank	*Popular 1.0-mix*	LocalRank
$R@1$	0.2010	0.1959	0.1944	0.1979
$R@2$	0.3184	0.3104	0.3026	0.3106
$R@3$	0.4168	0.3866	0.3844	0.3936
$R@4$	0.4759	0.4545	0.4403	0.4618
$R@5$	0.5241	0.5077	0.4800	0.5158

Table 15 $F@n$ of algorithms on Last.fm dataset

Algorithm / $P@n$	TR-UTS	FolkRank	*Popular 1.0-mix*	LocalRank
$F@1$	0.2674	0.2666	0.2665	0.2630
$F@2$	0.3288	0.3268	0.3304	0.3208
$F@3$	0.3583	0.3447	0.3400	0.3365
$F@4$	0.3512	0.3446	0.3405	0.3283
$F@5$	0.3386	0.3367	0.3325	0.3133

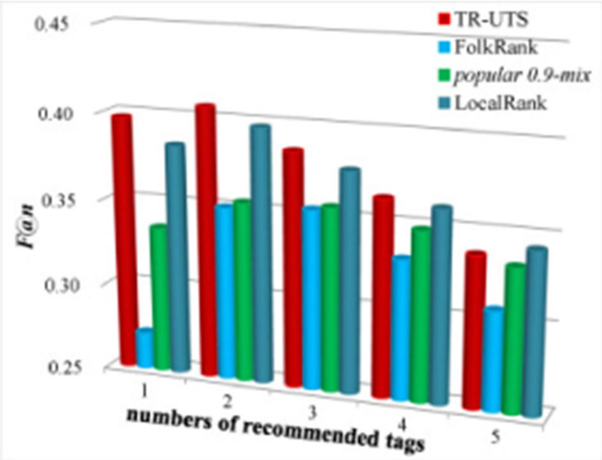

Fig. 4 $F@n$ of Each Algorithm on the CiteULike dataset

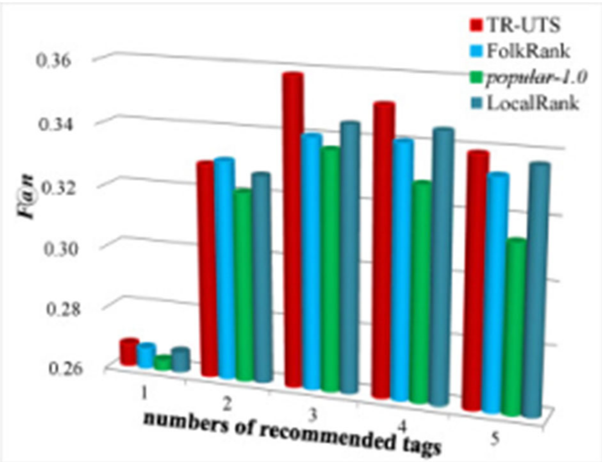

Fig. 5 $F@n$ of Each Algorithm on the Last.fm datase

7 Conclusions

This paper presented a novel method based on user tagging status to improve the quality of tag recommendation in folksonomies. The paper first introduced three types of user tagging status, after analysing the statistical results of the total number of tags used by a user during a period of time. At one moment, a user's current tagging status could be one of these three tagging status, namely the growing status, the mature status and the dormant status. Afterwards, the paper presented the determining user tagging status algorithm. Then, different strategies were developed with regard to the different user tagging status, by computing tag probability distribution in users' and resources' tag space based on the statistical language model. By contrasted with FolkRank, LocalRank and the most popular tags ρ-mix algorithms, the results of the proposed algorithm is better in accuracy, which is also to validate the effectiveness of the concepts of user tagging status introduced by this paper.

In this paper, the user tagging status is determined by analysing the historical tagging behavior of the user, but the backward time τ is fixed in advance. Developing the user's interests model is helpful to resolve the problem, which is one of our future work. This paper clusters users using the existed friendships in the folksonomy system, to develop an approach of clustering users is another direction for further research.

Acknowledgements This work was supported in part by the National Natural Science Foundation of China under grant No.61379114 and No.61533020.

References

Batagelj, V., & Zaveršnik, M. (2011). Fast algorithms for determining (generalized) core groups in social networks. *Advances in Data Analysis and Classification*, *5*(2), 129–145.

Belém, F.M., Martins, E.F., Almeida, J.M., & Goncal, M.A. (2014). Personalized and object-centered tag recommendation methods for web 2.0 applications. *Information Processing and Management*, *50*, 524C553.

Cai, X., Zhu, J., Shen, B., & Chen, Y. (2016). Greta: Graph-based tag assignment for github repositories. In *IEEE 40th annual computer software and applications conference (COMPSAC), 2016* (Vol. 1, pp. 63–72). IEEE.

Feng, W., & Wang, J. (2012). Incorporating heterogeneous information for personalized tag recommendation in social tagging systems. In *Proceedings of the 18th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1276–1284). New York, USA: ACM.

Gemmell, J., Schimoler, T., Mobasher, B., & Burke, R. (2010). Hybrid tag recommendation for social annotation systems. In *Proceedings of the 19th ACM international conference on information and knowledge management* (pp. 829–838). New York, USA: ACM.

Hmimida, M., & Kanawati, R. (2016). A graph-based meta-approach for tag recommendation. In *International workshop on complex networks and their applications* (pp. 309–320). Springer.

Hu, J., Wang, B., Liu, Y., & Li, D. (2012). Personalized tag recommendation using social influence. *Journal of Computer Science and Technolgy*, *27*(3), 527–540.

Jäschke, R., Marinho, L., Hotho, A., Schmidt-Thieme, L., & Stumme, G. (2008). Tag recommendations in social bookmarking systems. *Ai Communications*, *21*(4), 231–247.

Kim, H., & Kim, H.J. (2014). A framework for tag-aware recommender systems. *Expert Systems with Applications*, *41*, 4000–4009.

Kim, H.N., & El Saddik, A. (2011). Personalized pagerank vectors for tag recommendations: inside folkrank. In *Proceedings of the fifth ACM conference on recommender systems*. New York, USA: ACM. RecSys '11, pp 45–52.

Krestel, R., & Fankhauser, P. (2012). Personalized topic-based tag recommendation. *Neurocomputing*, *76*(1), 61–70.

Kubatz, M., Gedikli, F., & Jannach, D. (2011). Localrank - neighborhood-based, fast computation of tag recommendations. In *EC-Web, lecture notes in business information processing*, (Vol. 85, pp. 258–269). Springer.

Liu, K., Fang, B., & Zhang, W. (2011). Exploring social relations for personalized tag recommendation in social tagging systems. *IEICE Transactions on Information and Systems*, *94-D*(3), 542–551.

Liu, Z., Shi, C., & Sun, M. (2010). Folkdiffusion: A graph-based tag suggestion method for folksonomies. In Cheng, P.J., Kan, M.Y., Lam, W., & Nakov, P. (Eds.) AAIRS, *lecture notes in computer science* (Vol. 6458, pp. 231240). Springer.

Lu, C., Hu, X., Park, Jr., & Jia, H. (2011). Post-based collaborative filtering for personalized tag recommendation. In *Proceedings of the 2011 iConference* (pp. 561–568). New York, USA: ACM.

Ma, T., Zhou, J., Tang, M., Tian, Y., Al-Dhelaan, A., Al-Rodhaan, M., & Lee, S. (2015). Social network and tag sources based augmenting collaborative recommender system. *IEICE Transactions on Information and Systems*, *98*(4), 902–910.

Marinho, L.B., Nanopoulos, A., Schmidt-Thieme, L., Jäschke, R., Hotho, A., Stumme, G., & Symeonidis, P. (2011). Social tagging recommender systems. In *Recommender systems handbook* (pp. 615–644). Springer.

Ponte, J.M., & Croft, W.B. (1998). A language modeling approach to information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on research and development in information retrieval* (pp. 275–281). New York, USA: ACM.

Ramezani, M. (2011). Improving graph-based approaches for personalized tag recommendation. *Journal of Emerging Technologies in Web Intelligence*, *3*(2), 168–176.

Rawashdeh, M., Kim, H.N., Alja'am, J.M., & Saddik, A.E. (2013). Folksonomy link prediction based on a tripartite graph for tag recommendation. *Journal of Intelligent Information Systems*, *40*, 307–325.

Sood, S.C., Owsley, S.H., Hammond, K.J., & Birnbaum, L. (2007). Tagassist: Automatic tag suggestion for blog posts. In *ICWSM*.

Trant, J. (2009). Studying social tagging and folksonomy: A review and framework. *Journal of Digital Information, 10*(1).

Vander Wal, T. (2007). Folksonomy. http://vanderwal.net/folksonomy.html.

Wang, H., & Yeung, D.Y. (2016). Towards bayesian deep learning: A framework and some existing methods. *IEEE Transactions on Knowledge and Data Engineering*, *28*(12), 3395–3408.

Wang, H., Chen, B., & Li, W.J. (2013). Collaborative topic regression with social regularization for tag recommendation. In *IJCAI*.

Wang, H., Wang, N., & Yeung, D.Y. (2015). Collaborative deep learning for recommender systems. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1235–1244): ACM.

Wei, S., Zheng, X., Chen, D., & Chen, C. (2016). A hybrid approach for movie recommendation via tags and ratings. *Electronic Commerce Research and Applications*, *18*, 83–94.

Wu, Y., Yao, Y., Xu, F., Tong, H., & Lu, J. (2016). Tag2word: Using tags to generate words for content based tag recommendation. In *Proceedings of the 25th ACM international on conference on information and knowledge management* (pp. 22872292). ACM.

Xie, H., Li, X., Wang, T., Lau, R.Y., Wong, T.L., Chen, L., Wang, F.L., & Li, Q. (2016). Incorporating sentiment into tag-based user profiles and resource profiles for personalized search in folksonomy. *Information Processing and Management*, *52*(1), 61–72.

Zhang, Y., Zhang, B., Gao, K., Guo, P., & Sun, D. (2012). Combining content and relation analysis for recommendation in social tagging systems. *Physica A: Statistical Mechanics and its Applications*, *391*(22), 5759–5768.