Personalized Tag Recommendation Based on Latent Tag Mining and Fine-Grained Preferences: Postprint

Authors: Li Hongmei, Diao Xingchun, Cao Jianjun, Zhang Lei, Feng Qin

Date: 2018-11-29T00:00:00+00:00

Abstract

To further improve the performance of personalized tag recommendation, and to address the sparsity of tag data as well as the limitation of traditional methods that ignore potential tags hidden in user and item contexts, this paper proposes a personalized tag recommendation method based on latent tag mining and fine-grained preferences. First, we propose to utilize the contextual information of users and items to mine a small number of latent tags that users may be interested in from a large number of unobserved tags, reclassify tags into three categories: positive tags, latent tags, and negative tags, and thereby construct fine-grained preference relationships between <user, item> pairs and tags, which alleviates tag sparsity while enhancing the expressive capability of tag preference relationships. Then, based on the Bayesian Personalized Ranking optimization framework, we model the fine-grained preference relationships and combine pairwise interaction tensor factorization to predict preference values, constructing a fine-grained personalized tag recommendation model and proposing an optimization algorithm. Comparative experiments demonstrate that the proposed method effectively improves the accuracy of personalized tag recommendation while ensuring a fast convergence rate.

Full Text

Preamble

Personalized Tag Recommendation Based on Potential Tag Mining and Fine-grained Preference

Li Hongmei[1], Diao Xingchun[1], Cao Jianjun[2]†, Zhang Lei[1], Feng Qin[1]
(1. Army Engineering University, Nanjing 210007, China; 2. the 63rd Research Institute, National University of Defense Technology, Nanjing 210007, China)

Abstract: To further improve the performance of personalized tag recommendation, this paper addresses the sparsity of tag data and the limitation of traditional methods that ignore potential tags hidden in the context of users and items. We propose a novel personalized tag recommendation method called BPR-PITF-P based on potential tag mining and fine-grained preference. First, BPR-PITF-P leverages the context information of both users and items to mine potential and informative tags from a large set of unobserved tags, resulting in three tag categories: positive tags, potential tags, and negative tags. Based on this classification, we transform the traditional pairwise preference into a fine-grained preference relationship among user-item pairs and tags. This approach helps alleviate the sparsity problem of tagging data while enhancing the expressive power of preference relationships. Second, combined with pairwise interaction tensor factorization for preference value prediction, BPR-PITF-P models the preference relationship based on the optimization criteria of Bayesian personalized ranking, and develops a personalized tag recommendation model with a corresponding optimization algorithm. Experimental results demonstrate that our proposed method improves tag recommendation performance while guaranteeing convergence speed.

Keywords: tag recommendation; potential tag mining; Bayesian personalized ranking; pairwise interaction tensor factorization

0 Introduction

Social tagging, as an important feature of Web 2.0, allows users to freely create tags to annotate resources (such as webpages, music, movies, images, etc., collectively referred to as items). Tags describe both the explicit semantic features of items and the explicit preference content of users, enabling more convenient information retrieval, organization, and sharing, thereby alleviating the "information overload" problem to some extent. Websites such as Delicious, CiteULike, and Douban have all implemented tag recommendation modules to provide users with personalized tags. When browsing resources, the tag recommendation system suggests tags that users may be interested in, enabling them to better annotate and manage these resources. However, since many users are reluctant to add tags to information resources, personalized tag recommendation systems are needed to automatically recommend relevant tags.

Typical personalized tag recommendation methods mainly include collaborative filtering-based tag recommendation algorithms [?], graph model-based tag recommendation algorithms [?, ?, ?], tensor factorization-based tag recommendation [?, ?, ?, ?], content-based tag recommendation algorithms [?, ?, ?], and hybrid tag recommendation algorithms [?]. These methods have their own advantages and disadvantages and are suitable for different application scenarios. Additionally, some literature has proposed utilizing additional contextual information (such as time [?, ?] and social relationships [?]) to assist in improving the capability of personalized tag recommendation.

Since data in tag recommendation systems is typically sparse, which severely affects the performance of recommendation algorithms, and tensor factorization methods can effectively handle sparse data with accurate predictability and good scalability, this paper focuses on tensor factorization-based tag recommendation. Typical tensor factorization methods for tag recommendation include Higher-Order Singular Value Decomposition (HOSVD) [?] and Ranking Tensor Factorization (RTF) [?]. These methods are based on the classical Tucker Decomposition (TD) model but have relatively high computational complexity [?]. To further reduce computational complexity, [?] proposed a method based on Pairwise Interaction Tensor Factorization (PITF) combined with the popular Bayesian Personalized Ranking (BPR) [?] method, which is based on pairwise preference assumptions. This approach models the pairwise preference relationship between <user, item> pairs and tags, improving both computational efficiency and recommendation accuracy.

These methods assume that for a given item, users' preference for added tags (i.e., observed tags) is greater than for unadded tags (i.e., unobserved tags), and use uniform sampling to extract tag pairs. However, the long-tail distribution characteristic of tag data makes uniform sampling problematic: on one hand, it samples some meaningless tag pairs, affecting recommendation accuracy; on the other hand, it fails to provide effective information, resulting in slow model convergence. To address this issue, [?] further proposed an adaptive oversampling-based pairwise sampling method on top of [?] to improve model convergence capability and tag recommendation performance. However, this method focuses too much on popular tags in the long-tail distribution, ignoring some potential and meaningful tags hidden in the long-tail data, which can lead to premature convergence. Moreover, these methods are affected by sparsity and ignore the impact of user and item context information on different tag preference relationships.

Due to the scale of unobserved tags typically being much larger than that of observed tags, the long-tail distribution of data easily leads to sampling bias. Treating different tags equally reduces the expressive power of preference relationships. If we can mine a small number of potential tags that users may be interested in from a large set of unobserved tags, we can construct more comparable and meaningful tag pairs, alleviating the impact of sparsity while improving the preference expression capability of <user, item> pairs for tags and the recommendation accuracy of the model. In fact, these potential tags can be obtained from the context of target users and items: first, tags added by other users to the item, which well characterize the item' s attributes; second, tags added by the user to other similar items, which express the user' s preference content. Compared with other unobserved tags, these potential tags are more closely associated with the target user and item and are more likely to be of interest to the user. It is necessary to differentiate them from other unobserved tags to construct more comparable tag pairs, thereby providing effective information for ranking-based recommendation models.

Based on the above analysis, we propose a personalized tag recommendation method based on potential tag mining. By leveraging user and item context information, we mine a small set of potential tags closely related to the target user and item from a large amount of unobserved tag data, forming different types of tags and constructing preference relationships between <user, item> pairs and tags. This idea is similar to [?] and [?], which mine potentially more interesting items for users using social relationships to build fine-grained preference relationships for items, thereby improving personalized item recommendation performance. These methods provide strong support for our approach. However, for the task of personalized tag recommendation, when user social relationships are lacking, a key step is how to mine potential tags using only user-item-tag interaction data. This paper proposes fully utilizing user and item context to achieve potential tag mining.

1 Potential Tag Mining and Fine-grained Preference Relationship Construction

First, we leverage the contextual relationships of users and items to mine potential and valuable tag samples, constructing three tag categories: positive tags, potential tags, and negative tags. Then, based on these samples, we reconstruct the preference relationships between (user, item) pairs and different types of tags.

1.1 Potential Tag Mining

Since the set of unobserved tags is relatively large, extracting more valuable samples from it plays an important role in improving recommendation model performance. How can we mine potentially useful tags? We believe that user and item context information can be used to mine some potential and valuable tags to reconstruct the sample set and preference relationships between <user, item> pairs and tags.

Table 1 shows some examples of <user, item, tag> triples from the dataset. The goal of personalized tag recommendation is to recommend tags that a user may be interested in for a given item, i.e., to recommend a top-n tag list for a specific <user, item> combination.

As shown in Table 1, user u_1 added tag t_1 to item i_1. Traditional tag recommendation methods consider that for the <user, item> pair (u_1, i_1), the preference for observed tag t_1 is greater than for other unobserved tags, even treating tags $t_5, t_6, t_7, t_8, t_9, t_{10}$ equally. However, these methods ignore the impact of user and item context on tag recommendation.

For example, item i_1 was tagged by user u_1 with tag t_1, but was also tagged by other users u_3 and u_4 with tag t_2, indicating that tag t_2 well expresses the characteristics of item i_1. Therefore, tag t_2 can constitute a potential tag of interest to the user. Similarly, user u_1 is interested not only in tag t_1 but also in tag t_4, making t_4 another potential tag of interest. Therefore, we can consider that for the <user, item> pair (u_1, i_1), the preference for unobserved tag t_2 may be greater than for other unobserved tags. By leveraging these potential tags, we can better express the preference relationships between <user, item> pairs and tags, providing valuable information for personalized tag recommendation and thereby improving model performance.

Formally, we define three types of tags for each <user, item> pair: positive tags T_A, potential tags T_P, and negative tags T_B.

1) Positive tags T_A: $T_A = \{t|(u, i, t) \in S\}$, where S is the training set containing a large number of (user, item, tag) triples.
2) Potential tags T_P: From the item context perspective, we can mine potential tags: $T_i = \{t|(u', i, t) \in S\}$. From the user context perspective, we can mine potential tags: $T_u = \{t|(u, i', t) \in S\}$. Thus, for <user, item> pair (u, i), we can mine potential tags $T_P = T_i \cup T_u$.
3) Negative tags T_B: $T_B = \{t|t \notin T_A \cup T_P, t \in T\}$.

Based on these three types of tag data, we construct user preference relationships, formalized as $T_A \succ T_P \succ T_B$. This fine-grained preference relationship can then be modeled and solved using the BPR-PITF model [**?**].

1.2 Fine-grained Preference Relationship

Based on the three types of tags constructed above, we propose to improve the traditional pairwise preference assumption (observed tag $\succ$ unobserved tag) to a fine-grained preference relationship assumption: observed tag $\succ$ potential tag $\succ$ other unobserved tags. This fine-grained preference relationship can be formalized as $t_A \succ t_P \succ t_B$, where t_A is an observed tag, t_P is a potential tag, and t_B is another unobserved tag. Compared with traditional pairwise preference relationships, the proposed fine-grained preference relationship can fully mine users' potential preferences and improve the expressive power of user preference relationships.

2 Personalized Tag Recommendation Model and Solution

2.1 Personalized Tag Recommendation Model

For the fine-grained user preference relationship obtained through potential tag mining, we utilize the BPR-PITF optimization framework to build a new personalized tag recommendation model. The objective function is:

$$L(\Theta) = \sum_{(u,i)\in S} \sum_{t_A \in T_A^{u,i}} \sum_{t_P \in T_P^{u,i}} \sum_{t_B \in T_B^{u,i}} -\ln \sigma(\hat{x}_{u,i,t_A} - \hat{x}_{u,i,t_P}) \\ -\ln \sigma(\hat{x}_{u,i,t_A} - \hat{x}_{u,i,t_B}) + \lambda\|\Theta\|^2$$

where $\Theta = \{U, V, U_T^A, V_T^A, U_T^P, V_T^P, U_T^B, V_T^B\}$ represents the model parameters, and λ is the regularization factor to prevent overfitting.

Since the proposed model is based on BPR-PITF and Potential tags for personalized tag recommendation, it is called BPR-PITF-P.

The goal of personalized tag recommendation is to recommend tags that users may be interested in for a given item, which is essentially a ranking task that aims to place the most likely tags at the top. Bayesian Personalized Ranking (BPR) is a personalized recommendation optimization criterion and framework based on ranking learning. Pairwise Interaction Tensor Factorization (PITF) is a classic method for calculating the preference value of a <user, item> pair for a tag. Similar to commonly used matrix factorization models, PITF has solid theoretical foundations, good extensibility, and accurate prediction capabilities, making it compatible with the BPR optimization framework.

PITF is a special form of tensor factorization that models preferences using pairwise interactions among users, items, and tags. It computes the preference value $\hat{x}_{u,i,t}$ for <user, item> pair (u, i) for tag t by decomposing three interaction relationships:

$$\hat{x}_{u,i,t} = \langle U_u, V_i \rangle + \langle U_u, U_t \rangle + \langle V_i, V_t \rangle$$

where $U \in \mathbb{R}^{M\times K}$, $V \in \mathbb{R}^{N\times K}$, $U_T \in \mathbb{R}^{P\times K}$, and $V_T \in \mathbb{R}^{P\times K}$ are the latent factor matrices for users, items, and tags, respectively. M is the number of users, N is the number of items, and P is the number of tags.

For the prior probability $p(\Theta)$, a Gaussian distribution function with mean 0 is commonly used.

2.2 Model Solution

For each sample (u, i, t_A, t_P, t_B), we use Stochastic Gradient Descent (SGD) to iteratively update the parameters $\Theta = \{U, V, U_T^A, V_T^A, U_T^P, V_T^P, U_T^B, V_T^B\}$ with learning rate η.

The gradient of L can be expressed as:

$$\frac{\partial L}{\partial \Theta} = \sum_{(u,i,t_A,t_P,t_B)\in S} \left(\frac{\partial(-\ln\sigma(\hat{x}_{u,i,t_A} - \hat{x}_{u,i,t_P}))}{\partial \Theta} + \frac{\partial(-\ln\sigma(\hat{x}_{u,i,t_A} - \hat{x}_{u,i,t_B}))}{\partial \Theta} \right) + \lambda \frac{\partial \|\Theta\|^2}{\partial \Theta}$$

Specifically, the gradients are:

1) Update for U:

$$\frac{\partial L}{\partial U_u} = \sum_{(u,i,t_A,t_P,t_B)\in S} (\delta_{u,i,t_A,t_P} + \delta_{u,i,t_A,t_B}) \cdot (V_i + U_{t_A}^A - V_i - U_{t_P}^P) + \lambda U_u$$

where $\delta_{u,i,t_A,t_P} = (1-\sigma(\hat{x}_{u,i,t_A} - \hat{x}_{u,i,t_P}))$ and $\delta_{u,i,t_A,t_B} = (1-\sigma(\hat{x}_{u,i,t_A} - \hat{x}_{u,i,t_B}))$.

2) Update for V:

$$\frac{\partial L}{\partial V_i} = \sum_{(u,i,t_A,t_P,t_B)\in S} (\delta_{u,i,t_A,t_P} + \delta_{u,i,t_A,t_B}) \cdot (U_u + V^A_{t_A} - U_u - V^P_{t_P}) + \lambda V_i$$

3) Update for U_T^A:

$$\frac{\partial L}{\partial U^A_{t_A}} = \sum_{(u,i,t_A,t_P,t_B)\in S} \delta_{u,i,t_A,t_P} \cdot U_u + \lambda U^A_{t_A}$$

4) Update for V_T^A:

$$\frac{\partial L}{\partial V^A_{t_A}} = \sum_{(u,i,t_A,t_P,t_B)\in S} \delta_{u,i,t_A,t_P} \cdot V_i + \lambda V^A_{t_A}$$

Similar update rules apply to $U_T^P, V_T^P, U_T^B, V_T^B$.

2.3 BPR-PITF Algorithm

BPR-PITF-P Algorithm

Input: Training set S, potential tag set $T_P^{u,i}$ and negative tag set $T_B^{u,i}$ for each (u,i), learning rate η, maximum iterations κ

Output: Model parameters $\Theta = \{U, V, U_T^A, V_T^A, U_T^P, V_T^P, U_T^B, V_T^B\}$

1. Initialize $\Theta = \{U, V, U_T^A, V_T^A, U_T^P, V_T^P, U_T^B, V_T^B\}$ with $N(0, 0.01)$
2. repeat
3. for each $(u, i, t_A) \in S$ do
4. `Sample potential tag $t_P$ from $T_P^{u,i}$`
5. `Sample negative tag $t_B$ from $T_B^{u,i}$`
6. `Construct sample $(u,i,t_A,t_P,t_B)$`
7. `Compute gradients $\frac{\partial L}{\partial \Theta}$ using equations (9)-(24)`
8. `Update $U_u \leftarrow U_u - \eta \frac{\partial L}{\partial U_u}$`
9. `Update $V_i \leftarrow V_i - \eta \frac{\partial L}{\partial V_i}$`
10. `Update $U_T^A, V_T^A, U_T^P, V_T^P, U_T^B, V_T^B$ accordingly`
11. end for
12. until objective function (equation 6) converges or iterations $\geq \kappa$

13. return Θ

Lines 3-6 perform sample sampling to obtain <user, item, positive tag, potential tag, negative tag> quintuples. Specifically, for each triple <user, item, positive tag> in the training set S, we randomly sample a potential tag t_P from $T_P^{u,i}$ and a negative tag t_B from $T_B^{u,i}$ to construct the sample (u, i, t_A, t_P, t_B).

Lines 7-11 compute the model gradients with respect to parameters Θ and update them using SGD. Parameters U and V are updated using equations (9)-(10), U_T^A and V_T^A using equations (11)-(12), U_T^P and V_T^P using equations (13)-(18), and U_T^B and V_T^B using equations (19)-(24).

Notably, according to [**?**], each iteration typically involves $|S|$ BPR update processes, where $|S|$ is the number of triples in the training set. Each BPR update is based on one sample (u, i, t_A, t_P, t_B), ensuring that all observed samples participate in training during each iteration.

Line 12 checks for convergence. After training, we can compute scores for all unobserved tags for (user, item) pairs in the test set using equation (4), rank tags by score, and generate top-n personalized tag recommendation lists.

Complexity Analysis: Each iteration involves $|S|$ samples. The complexity of updating each sample is mainly related to the latent feature dimension K, so the time complexity per iteration is $O(|S| \cdot K)$. Since potential tags t_P and negative tags t_B can be preprocessed before training, they do not affect the overall training time.

3 Experiments

3.1 Datasets

We conduct experiments on two standard recommendation datasets: Lastfm and MovieLens (https://grouplens.org/datasets/hetrec-2011).

Lastfm: Lastfm is a popular music website containing user listening and tagging information for singers on Last.fm. The dataset appears as multiple <user, item, tag> triples. The data is very sparse. Following [**?**], we preprocess the data using the p-core method (p=10) to ensure each user, item, and tag appears in at least 10 <user, item> pairs. The preprocessed dataset includes 614 users, 1,715 artists, 873 tags, and a total of 87,285 records.

MovieLens: MovieLens is a movie recommendation website where users can add tags to movies to facilitate search and recommendation. We preprocess it using the p-core method (p=5) to ensure each user, item, and tag appears in at least 5 <user, item> pairs. The preprocessed dataset includes 366 users, 1,185 movies, 873 tags, and a total of 20,089 records.

We adopt 5-fold cross-validation to train and test models, averaging the results. Following [**?**], we construct training and test sets: for each user appearing in the dataset, we extract one <user, item> pair and its related tags to form the test set S_{test}, with the remainder as the training set S_{train}.

3.2 Evaluation Metrics

We evaluate personalized tag recommendation results using Precision@n, Recall@n, and F1@n [?].

$$\text{Precision@n} = \frac{|\{(u,i) \in S_{test} \mid \text{Top-n}(u,i) \cap T_{u,i}^{test}\}|}{|S_{test}| \cdot n}$$

$$\text{Recall@n} = \frac{|\{(u,i) \in S_{test} \mid \text{Top-n}(u,i) \cap T_{u,i}^{test}\}|}{|\{(u,i,t) \in S_{test}\}|}$$

$$\text{F1@n} = \frac{2 \cdot \text{Precision@n} \cdot \text{Recall@n}}{\text{Precision@n} + \text{Recall@n}}$$

where Top-n(u,i) represents the top-n tags recommended for (u,i), and $T_{u,i}^{test}$ is the set of tags associated with (u,i) in the test set.

3.3 Comparison Methods and Parameter Settings

We compare our method with several popular tag recommendation methods:

a) Pop: This method recommends tags based on popularity. For each item, it counts the number of times each tag is applied and recommends the most frequently used tags. This baseline validates the effectiveness of other methods.

b) BPR-PITF-U [?]: This method builds a personalized tag recommendation objective function based on the BPR-PITF optimization criterion and uses uniform sampling for training.

c) BPR-PITF-A [?]: This method builds a personalized tag recommendation objective function based on the BPR-PITF criterion and uses adaptive sampling based on exponential distribution for training.

d) BPR-PITF-P: Our proposed method that first samples potential tags to construct training samples, forms new preference relationships, and rebuilds the personalized tag recommendation model based on the BPR-PITF optimization criterion.

Parameter Settings: We set latent feature dimensions $K \in \{20, 40\}$, learning rate $\eta = 0.025$, maximum iterations $\kappa = 1000$, and test top-n values $n \in \{1, 2, ..., 10\}$. For the regularization coefficient λ, we select optimal values for each dataset ($\lambda = 0.01$ for Lastfm, $\lambda = 0.01$ for MovieLens).

3.4 Experimental Results

3.4.1 Top-n Recommendation Performance Comparison As shown in Figure 1 [Figure 1: see original paper] and Figure 2 [Figure 2: see original paper], on both Lastfm and MovieLens datasets, our proposed BPR-PITF-P method achieves the best performance across all three evaluation metrics, demonstrating its effectiveness in improving personalized tag recommendation accuracy.

Specifically, on the Lastfm dataset with $K = 20$ (Figures 1(a), 1(c), 1(e)), Precision@n decreases as top-n increases, Recall@n increases as top-n increases, and F1@n first increases then decreases, reaching optimal values at $n = 4$. With $K = 40$ (Figures 1(b), 1(d), 1(f)), BPR-PITF-U, BPR-PITF-A, and BPR-PITF-P show similar performance, but there is not much improvement over $K = 20$. This is because for latent factor models, a small number of latent features is usually sufficient to represent the characteristics of users, items, and tags, ensuring good recommendation performance within acceptable time. Since Pop is unaffected by latent dimension K, its performance is identical for $K = 20$ and $K = 40$.

Pop performs worst because it only considers the most popular tags used to annotate items, ignoring users'personalized needs. Clearly, this non-personalized method underperforms personalized tag recommendation methods (BPR-PITF-U, BPR-PITF-A, and our BPR-PITF-P).

On the MovieLens dataset (Figure 2), Pop, BPR-PITF-U, BPR-PITF-A, and BPR-PITF-P show similar trends as on Lastfm. The difference is that MovieLens achieves optimal F1@n at $n = 2$, and Precision@n decreases more rapidly as n increases. This is because MovieLens is sparser than Lastfm—most (user, item) pairs have fewer tags—so smaller n values (e.g., $n = 2$ or $n = 4$) yield better performance.

Therefore, to compare convergence performance, we focus on top-n recommendation performance with $n = 4$ on Lastfm and $n = 2$ on MovieLens.

3.4.2 Convergence Performance Comparison Figure 3 [Figure 3: see original paper] and Figure 4 [Figure 4: see original paper] show the recommendation performance versus iteration number on Lastfm and MovieLens datasets, respectively. On both datasets, BPR-PITF-U converges slowest and has the worst accuracy. BPR-PITF-A converges relatively quickly but is less accurate than our BPR-PITF-P method. BPR-PITF-P requires more time per iteration but achieves stable results after fewer iterations.

In summary, our proposed method achieves better recommendation accuracy while maintaining reasonable convergence capability, better satisfying users' personalized needs.

References

[1] Marinho L, Hotho A, Schmidt-Thieme L, et al. Tag recommendations in

social bookmarking systems [J]. Ai Communications, 2008, 21 (4): 231-247.

[2] Kim H N, Saddik A E. Personalized PageRank vectors for recommendations: inside FolkRank [C]// Proc of ACM Conference on Recommender Systems. New York: ACM Press, 2011: 45-52.

[3] Ramezani M, Improving Graph-based Approaches for Personalized Tag Recommendation [J]. Journal of Emerging Technologies in Web Intelligence, 2011, 3 (2).

[4] Rendle S, Marinho L B, Nanopoulos A, et al. Learning optimal ranking with tensor factorization for tag recommendation [C]// Proc of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM Press, 2009: 727-736.

[5] Rendle S, Schmidt-Thieme L. Pairwise interaction tensor factorization for personalized tag recommendation [C]// Proc of ACM International Conference on Web Search and Data Mining. New York: ACM Press, 2010: 81-90.

[6] Symeonidis P, Nanopoulos A, Manolopoulos Y. A Unified framework for providing recommendations in social tagging systems based on ternary semantic analysis [J]. IEEE Trans on Knowledge & Data Engineering, 2009, 22 (2): 179-192.

[7] Rendle S, Freudenthaler C. Improving pairwise learning for item recommendation from implicit feedback [C]// Proc of the 7th ACM International Conference on Web Search and Data Mining. New York: ACM Press, 2014: 273-282.

[8] Krestel R, Fankhauser P. Personalized topic-based tag recommendation [J]. Neurocomputing, 2012, 76 (1): 61-70.

[9] Zhang Bin, Zhang Yin, Gao Kening, et al. Combining relation and content analysis for social tagging recommendation [J]. Journal of Software, 2012, 23 (3): 476-488.

[10] Feng W, Wang J. Incorporating heterogeneous information for personalized tag recommendation in social tagging systems [C]// Proc of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM Press, 2012: 1276-1284.

[11] Lops P, Gemmis M D, Semeraro G, et al. Content-based and collaborative techniques for tag recommendation: an empirical evaluation [J]. Journal of Intelligent Information Systems, 2013, 40 (1): 41-61.

[12] Li Gui, Wang Shuang, Li Zhengyu, et al. Folksonomy tag recommendation algorithm based on time-weighted tripartile graph [J]. Journal of Chinese computer system, 2016, 37 (2): 269-274.

[13] Yu Hong, Zhou Bin, Deng Mingyao, et al. Tag recommendation method in folksonomy based on user tagging status [J]. Journal of Intelligent Information Systems, 2017, (2): 1-22.

[14] Mahboob V A, Jalali M, Jahan M V, et al. Swallow: resource and tag recommender system based on heat diffusion algorithm in social annotation systems [J]. Computational Intelligence, 2017, 33 (1): 99-118.

[15] Symeonidis P, Nanopoulos A, Manolopoulos Y. Tag recommendations based on tensor dimensionality reduction [C]// Proc of International Conference on Artificial Intelligence Applications & Innovations. 2009, 296: 331-340.

[16] Rendle S, Freudenthaler C, Gantner Z, et al. BPR: Bayesian personalized ranking from implicit feedback [C]// Proc of the 25th Conference on Uncertainty in Artificial Intelligence. AUAI Press, 2009: 452-461.

[17] Li Huayu, Ge Yong, Hong Richang, et al. Point-of-interest recommendations: Learning potential check-ins from friends [C]// Proc of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. New York: ACM Press, 2016: 975-984.

[18] Zhao Tong, McAuley J, King I. Leveraging social connections to improve personalized ranking for collaborative filtering [C]// Proceedings of the 23rd ACM international conference on information and knowledge management. New York: ACM Press, 2014: 261-270.