Journal of Computers Vol. 31 No. 6, 2020, pp. 196-209
doi:10.3966/199115992020123106016

Tag-based Matrix Factorization Recommendation Based on Topic Detection Technology

Bing Fang*, Xiao-Ying Liu

Department of Information Management, School of Management, Shanghai University, Shanghai, China
{melodyfang, vickyliu0923}@shu.edu.cn

Received 19 October 2019; Revised 28 February 2020; Accepted 13 April 2020

Abstract. Recommender systems have provided a powerful tool for effectively targeting customers. As one of the most advanced recommendation methods, matrix factorization recommendation, has been used quite extensively in electronic commerce. Previous researchers have stressed the importance of factor quality on the performance of matrix factorization recommendation. However, existing studies usually used latent factors or construct explainable factors with products' categories that cannot accurately characterize items. In this paper, a novel tag-based recommendation method is proposed to improve the factor quality in matrix factorization recommendation. The proposed recommendation method is built using the following processes. First, online user reviews are collected. Second, using the latent Dirichlet allocation algorithm, tags of items are extracted to form the factor vector. Third, the rating matrix is factorized into a "user-tag" matrix and "item-tag" matrix by a gradient descent method. Finally, the missing values of the rating matrix are predicted according to the learnt user-tag and item-tag matrices. A variety of experiments on the real data are conducted to evaluate the performance of our proposed recommendation method. The results demonstrate the superior performance of our proposed method compared with traditional tag-based matrix factorization recommendation and alternative tag-based matrix factorization recommendation.

Keywords: latent Dirichlet allocation, matrix factorization, recommendation, tag extraction, topic detection

1 Introduction

With the rapid development of electronic commerce, recommender systems have become a powerful tool for business analytic toolkits. According to Hosanagar and Buja [1], 35% of Amazon's sales, including those of movies, books, and music, originated from recommender systems. This is partly because online transaction platforms can record the details of customer behavior, making this behavior available for customer targeting to both firms and researchers. It is easy to deduce that if the performance of recommender systems improves, online retailers can expect significant sales increases.

The recommendation method is the core of recommender systems [2-3], and these methods can be used to predict customers' ratings of each product on the basis of their past behavior. There are a considerable amount of research on recommendation methods. These methods can be divided three main categories: content-based recommendation, collaborative-filtering recommendation, and matrix factorization recommendation (MFR). However, there are problems in content-based recommendations, such as limit content analysis, over specialization, and the new user problem [2]. In contrast to content-based recommendation, collaborative-filtering recommendation has dominated traditional recommendation methods, but it still has the defects such as the new user problem, new item problem, and sparsity [4-6]. Since the Netflix Prize Competition [7], MFR has gradually become one of the most advanced recommendation methods because of its excellent performance when solving problems of collaborative-filtering recommendations [8].

* Corresponding Author

MFR uses factors to describe the link between the characteristics of products and preferences of customers. Researchers have pointed out that factor selection has a significant impact on the performance of MFRs [9]. In general, the factors used in MFRs are product categories [10]. However, the category of a product is insufficient for product distinction because of its limited information. For example, the movie Titanic and Sleepless in Seattle are both romantic movies, but if MFR only uses the category "romantic" as the factor, it will always recommend both of them to the same customers. In addition, product categories are usually provided by retailers, who tend to hide product defects.

Compared with category, tags are a better choice for a feature that describes products. Tags refer to phrases or words that describe the products' feature and are provided by the customers of e-commerce sites. Because tags are informative and provided by users, they can be used to construct a new kind of product classification to replace categories. Therefore, tag-based recommendations have attracted more attention in the field of recommendation research [11]. Unfortunately, the quality of tags generated from existing tag-generation mechanisms do not meet expectations because it is quite difficult for customers to figure out the proper tags for generalizing the essence of a product accurately and quickly.

In addition to tags, online reviews are another important way for customers to comment on products. Online reviews are paragraphs written by users to describe their feelings about purchased products. Nowadays, they appear on almost every e-commerce website. Generally speaking, online reviews are groups of long sentences that are deliberately written by customers. As a result, online reviews usually better indicate user descriptions of the products and their real thoughts about the products. Although the dimensions of reviews are usually too high to be directly used in MFRs, the emergence of topic detection techniques makes it possible to extract tags from long online text reviews. Topic detection techniques are usually used to detect topics and extract topic-related keywords from text documents [12]. Because the topics of online reviews are strongly related to products, the keywords (tags) extracted from online reviews are accurate representations of products.

Considering the above situation, this paper proposes a research problem: to improve the performance of the MFR method, how to construct high-quality factor vectors through mining product tags from user online review.

Based on the above analysis, a novel tag-based matrix factorization recommendation (TMFR) method is proposed in this paper. This method makes the following contributions: (1) it introduces tags into MFR to replace category-based factors and latent factors, which can significantly improve the performance of MFR because tags are high-quality factors; (2) it provides a new tag-extraction method that uses topic detection in online user reviews, which generates higher-quality tags than those used in existing tag-based recommendation research.

The main technical achievements of our work can be summarized as following: (1) through using user online review data, we propose a novel approach to improve the performance of the MFR method; (2) by utilizing natural language processing technology, we propose a novel method to mine tags of products from user online review.

This paper proposes a new way to improve the performance of the existing matrix decomposition algorithm by using user comment data. (2) it proposes a method to mine keywords from user comment data by using natural language processing technology.

The remaining parts of this paper are organized as follows: Section 2 reviews the related work in this field. Section 3 introduces the TMFR method using tags extracted from online reviews. Section 4 compares our proposed recommendation method with existing MFR methods on two real-world datasets. Section 5 presents the evaluation results, and Section 6 presents our conclusions.

2 Related Work

As the selection of factors has a significant impact on MFR performance, researchers have studied a variety of methods to select them. In contrast, although there is little research using tags in MFRs, tags have been used in recommendation tasks, such as collaborative filtering recommendations and content-based recommendations. In this section, we review methods for factor selection in MFR and tag-based recommendation.

2.1 Factor Learning Methods in Matrix Factorization Recommendations

Research on MFRs became popular after the Netflix prize competition. In this competition, the BellKor team, which originally proposed the MFR method, won the 2007 Progress Prize. The BellKor team used two kinds of categories as the factors in the competition [9]. Koren et al. pointed out that the quality of factors used in MFRs influences the performance of the recommender system.

To improve the quality of factors used in MFRs, existing researchers mainly combine other kinds of information with rating matrix to construct factor vectors in MFRs. Du et al. made use of user attribute information and user social relations to learn the extent of interest the user has in the factors [13]. Liu et al. incorporated the rating information, item content information and tag information into Document Recommendation [14]. Gu et al. combined implicit feedback with explicit contextual feature to construct factor vectors [15].

He et al. proposed a probabilistic matrix factorization method by incorporating implicit feedback into latent factor vectors construction [16]. Shen et al. proposed a matrix factorization recommendation method which combine semantic attitudes, sentiment, volume, and objectivity extracted from user-generated content to generate recommendation to generate factor vectors [17]. The above research mainly focuses on the performance improvement of matrix factorization recommendation methods through combining other kinds of information to generate factor vectors. However, these researches didn’t address the extraction of product characters, which is the key to the improvement of factor quality.

At the same time, factors are the representations of user preferences and item characteristics, it is natural to construct explainable factors with words or phrases. Thus, to improve the quality of factors, there are several studies that use natural language processing in MFRs. For example, researchers used natural language processing in MFRs to extract the factors of scientific articles from their content [18-19]. Wang and Blei provided a method for determining explainable factors [18]. Shan and Banerjee used a latent Dirichlet allocation (LDA) model to extract the topics from movie descriptions as factors for movies to improve the performance of a probabilistic MFR model [20]. Purushotham, Liu and Kuo found the item factors by means of topic modeling and found the user factors from data taken from social network sites [21]. Lin et al. utilized cross-site ratings and content features to extract explainable factors [22]. However, the factors of the items were extracted only from a brief description and the factors of users were inferred indirectly through cross-site information. All these researchers found explainable factors through natural language processing technologies. However, all of the factors used in these studies were extracted from the information provided by manufactures and distributors. As we discussed above, manufacturers and distributors tend to downplay the defects of products, so factors generated in this way may not be an accurate reflection of an item’s characteristics. The performance of MFRs based on these factors will also be impacted accordingly.

In addition to the methods that use natural language processing technologies to extract explainable factors for MFRs, there also exist other methods that enhance the factor learning process by taking various types of context information into account. Yin and Xu computed each neighbor’s weight from their similarity and integrated the weights to enhance the factor learning process [23]. They focused on the similarity of factors between neighbors, but they still used latent factors. Y. Shi et al. compared the similarity of movies by taking mood-specific tags and plot keywords into account, and further enhanced MFR using this kind of similarity [10]. The mood-specific tags were extracted from the movies’ categories and plot keywords were extracted from movie plots. Forsati, Mahdavi, Shamsfard, and Sarwat incorporated both trust and distrust relationships mined from social networks in a matrix factorization-based model to mitigate data sparsity and cold-start user issues [24]. They focused on the similarity of factors between neighbors by computing the trust relationships among users. Zhao et al. utilized the LDA model to predict the probability that a user rates an item using only positive examples, and further builds a recommendation model based on a SVD approach [25]. By taking the probability that a user rates an item into account, they focused on the replacement of information loss caused by directly optimizing some ranking targets in the factor learning process. Liu et al. [8] computed user factors by taking users’ friends factors in to account, and computed item factors by comparing the content similarity of items. They focused on both the similarity of factors among users and the similarity of factors among items. Miao quantized new trust relationships by designing a set of trust propagation rules based on the direct trust relationships of a social network, and loaded “the quantitative trust relations after the trust propagation as the trust weight into the matrix factorization-based model” [26]. They focused on the similarity of factors among users by introducing trust in the factor learning process. Shi, Larson, and Hanjalic proposed a unified recommendation model that combines rating-oriented and ranking-oriented collaborative filtering approaches [27]. They focused on building a unified recommendation by sharing the common latent features of users and items in rating-oriented and ranking-oriented models. Shi et al. computed user factors from their tagging behavior and combined user factors with image factors using the matrix factorization approach [28]. They simply treat user tags as users' interests. The above research mainly focuses on enhancing the factor learning process by computing the similarity of factors among users or items based on various types of context information. However, they did not provide an effective approach to learning factors that accurately reflect user preferences or item characteristics; the factors used in the study were still latent.

2.2 Tag-based Recommendations

Although there is little research on building TMFR systems, tag-based recommendation has become at topic of interest in the research field, there are many studies that use tags in recommendations (such as collaborative filtering recommendations and content-based recommendations). Alhamid et al. employed collaborative tags annotated by a large number of users on an item to "maximize the benefit of the extracted contextual information in the recommendation process" [29]. H. Kim and Kim proposed a hybrid framework that includes tag and item recommendations [30]. They made use of association rules and a bigram approach to help users annotate items with tags and recommend items according to them. Zhang et al. improved recommendation performance by "mining webs with information extracted from search and browser logs of users" [31]. They learnt user preferences based on tags extracted from web documents in a user's browsing history. Ignatov et al. presented a tag-aware recommender system for radio networks [32]. They utilized the tags of each radio station to match user preferences and radio networks to effectively alleviate the sparsity problem. Zheng and Li built a recommendation model based on tags and time information [11]. They used tag and time information provided by users' tagging behaviors to generate users' rating lists, calculate user similarity based on the ratings, and recommended resources according to a collaborative filtering approach. Zhao et al. learn user preferences considering tag and time information [25]. They captured the latent correlations between users, items, tags, and time information using data from social tagging systems. Movahedian and Khayyambashi proposed a recommender system using tags annotated by users [33]. They first filter noisy tags, mapped the filtered tags onto ontology concepts, build tag-based user and item profiles "based on the relationships between users, items and tags," and finally recommended items based on the similarities between user and item profiles. Gedikli improved recommendation accuracy by taking tags into considerations [34]. They automatically learn user preferences for each tag from the overall ratings and learn user preferences for items according to their preferences for tags. H. N. Kim and Saddik recommend communities to individual users according to the tags of each community [35]. They compute the similarities between user tags and community tags using latent semantic analysis to predict user preferences. Kim et al. proposed a collaborative recommendation approach "by leveraging user-generated tags as preference indicators" [36]. They mine the frequent pattern of tags, and combine rating information and tagging patterns to identify the neighbors. Movahedian and Khayyambashi proposed a recommender system "based on the similarities between user and item profiles" [37]. They generate user and item profiles by illustrating various correlations between tags and recommend "a ranked list of items relevant to user needs." Martinez et al. improved recommendations using information from tags [38]. They compute user preferences for tags according to the number of times they have been annotated and the associated ratings and recommended items to users by combining the rating and tagging information. Nanopoulos used the data from collaborative tagging systems to improve the quality of recommendations [39]. They capture the correlations between users, tags, and items, and recommend items accordingly. These studies did not directly utilize tags to improve MFRs, but their work proves that tags could be used to predict user preferences and item characteristics. and hence improve the performance of recommendation systems.

The above research highlights that the quality of factors influences the performance of MFRs. However, existing research does not provide an effective approach to learning explainable factors that can accurately reflect user preferences and item characteristics. Meanwhile, in recent years, tag-based recommendation research has shown the great potential of tags in predicting user preferences and item characteristics, but there is little research that directly utilizes tags to improve MFRs. In contrast, tags used in existing research are either annotated by users or extracted from item profile information. As we discussed above, most users cannot fully express their feelings about items by annotating items with tags, nor can tags extracted from item profiles fully express an item's characteristics.

To solve these problems, this paper proposes a tag-based matrix factorization recommendation (TMFR) method that extracts tags from long online text reviews and uses them to construct explainable factors. These factors can be used to learn user preferences and item characteristics for efficient user rating prediction.

The main difference between our work and other existing MFR methods is that the different approach of constructing factor vectors. By using natural language processing technology, our proposed TMFR method constructs factor vector using product tags extracted from user online review, while existing MFR methods constructed factor vectors using the key words extracted from the product introduction content or user profile information. Since user online review can comprehensively and truly express product features, so the tags extracted from online review can greatly improve the quality of factor vectors, and in this way the performance of MFR method can be greatly improved.

3 TMFR Method

To overcome the drawback of traditional category-based matrix factorization (CMFR) methods, a novel TMFR method is proposed. It does not treat category as a factor or use latent factors in recommendation. Instead, it extracts tags of items from online user reviews using an LDA technique, and uses tags to construct factors for MFR. In this section, our proposed method is introduced in detail.

3.1 Tag Extraction from Long User Reviews

To extract tags from user long-text reviews, LDA is utilized in this paper. One of the most popular topic detection techniques, LDA provides a topic cluster model for text documents. It is able to identify associations between topics and one document as well as associations between one topic and words based on Bayesian probability models.

Formally, we define the following terms:

A corpus is denoted as $\boldsymbol{T} = \{T_1, T_2, \ldots, T_M\}$, where T_m is the review collection of item m, and M is the number of reviews in the corpus.

A review collection of one item m is a sequence of tags denoted as $\boldsymbol{T} = \{t_{m1}, t_{m2}, \ldots, t_{m,N_m}\}$, where $t_{m,n}$ is the n-th tag in the sequence.

A topic sequence associated with tag sequence T_m is denoted as $z_{m1} = \{z_{m,1}, z_{m,2}, \ldots, z_{m,N_m}\}$.

The tag extraction process can be described as follows:

For each topic k, where $\mathrm{k} \in 1, 2, 3, \ldots, K$, choose a tag distribution $\varphi_k \sim Dir(\varphi \mid \beta)$.

For each item's review collection r_m in corpus R, where m $\in 1, 2, 3, \ldots, M$,

choose length $N_m \sim \text{Poisson}(\varepsilon)$;

choose topic distribution $\theta_m \sim Dir(\theta \mid \alpha)$. .

For each tag $t_{m,n}$ in the review collections of item r_m, where $\mathrm{n} \in 1, 2, 3, \ldots, N_m$,

choose topic $z_{m,n} \sim \text{Multinomial}(z \mid \theta_m)$;

choose tag $t_{m,n} \sim \text{Multinomial}(t \mid \varphi_{z_{m,n}})$.

The joint distribution of a topic mixture θ_m, set of N_m topics Z_m, and set of N_m tags T_m is computed as follows:

$$p(\theta_m, z_m, T_m \mid \alpha, \beta\} = p(\theta_m \mid \alpha)\prod_{n=1}^{N_m} p(z_{m,n} \mid \theta_m) p(t_{m,n} \mid \varphi_{z_{m,n}}, \beta). \tag{1}$$

The overall review-topic distribution is denoted as $\boldsymbol{\varphi} = \{\varphi_1, \varphi_2, \ldots, \varphi_K\}^T$ and the whole topic-tag distribution is denoted as $\boldsymbol{\theta} = \{\theta_1, \theta_2, \ldots, \theta_M\}^T$

Given a set of long text reviews of items, by estimating review-topic distribution $\boldsymbol{\varphi}$ and topic-tag distribution θ, the tags of each item can be extracted. In this paper, Gibbs sampling is used. Fig. 1 illustrate the tag extraction model.

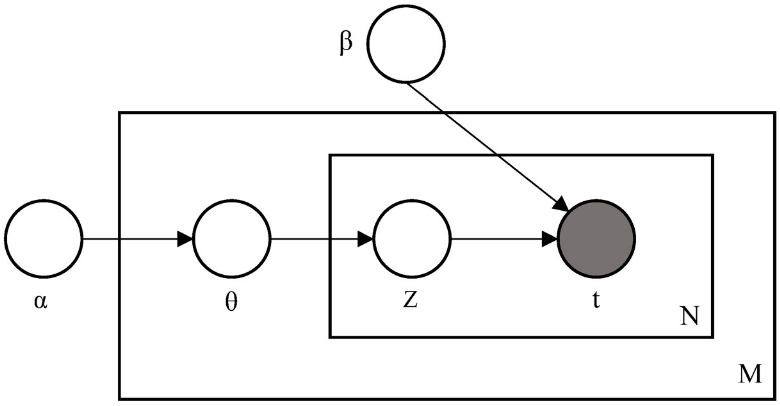

Fig. 1. Tag extraction model

3.2 TMFR

Generally speaking, given a training data set, MFR factorizes the “user-item” rating matrix into “user-factor” rating and “item-factor” rating matrices. The vectors in these two matrices can be used to measure the extent of interest the user has in the factors and the extent to which the items possess factors respectively. Through multiply these two kinds of vectors, we can predict the missing rating records in the test data set. In previous studies, researchers mainly used product category to construct factors or only used latent factors. Neither method could adequately describe user preferences or item characteristics. In TMFR, the extracted tags from long text online reviews are used to construct the factors. Fig. 2 illustrates the TMFR method.

As shown in Fig. 2, the TMFR process can be divided into the following three stages.

Stage 1: Learning the item-tag matrix.

The item collection is denoted as I = $\{I_1, I_2, I_3, \ldots, I_M\}$, and the tag collection is denoted as $T = \{t_1, t_2, \ldots, t_1\}$, where tags are learnt from the user review collection of each item. The item-tag vector is denoted as $q_i = \{q_{i1}, q_{i2}, \ldots, q_{Li}\}^T$, where $\mathrm{i} \in 1, 2, 3, \ldots, M$, to measure the extent to which the items possess the factors. Accordingly, the entire item-tag matrix is denoted as $Q = (q_1, q_2, \ldots, q_M)^T \in R^{L \times M}$.

Here,

$$q_{ki} = \begin{cases} 1, \text{if item } i \text{ has tag } k \\ 0, \qquad other \end{cases}. \tag{2}$$

Stage 2: Learning the user-tag matrix.

Here, a user collection is denoted as U = $\{u_1, u_2, u_3, \ldots, u_N\}$, and the user-tag vector is denoted as $p_u = \{p_{u1}, p_{u2}, \ldots, p_{uL}\}^T$, where $\mathrm{u} \in 1, 2, 3, \ldots, N$, to measure the extent of interest the user has in the factors. Accordingly, the overall user-tag matrix is denoted as $P = (p_1, p_2, \ldots, p_N)^T \in R^{N \times L}$.

To learn p_u given the training data, the recommendation method minimizes the error using the following objective function:

$$\min_{q^*, p^*} \sum_{(u,i) \in K} (\gamma_{ui} - q_i^T p_u)^2. \tag{3}$$

Here, γ_{ui} is user u’s rating of item i.

To avoid the overfitting problem, the regularized objective function is used in the method.

$$\min_{q^*, p^*} \sum_{(u,i) \in K} (\gamma_{ui} - q_i^T p_u)^2 + \lambda(\| q_i \|^2 + \| p_u \|^2). \tag{4}$$

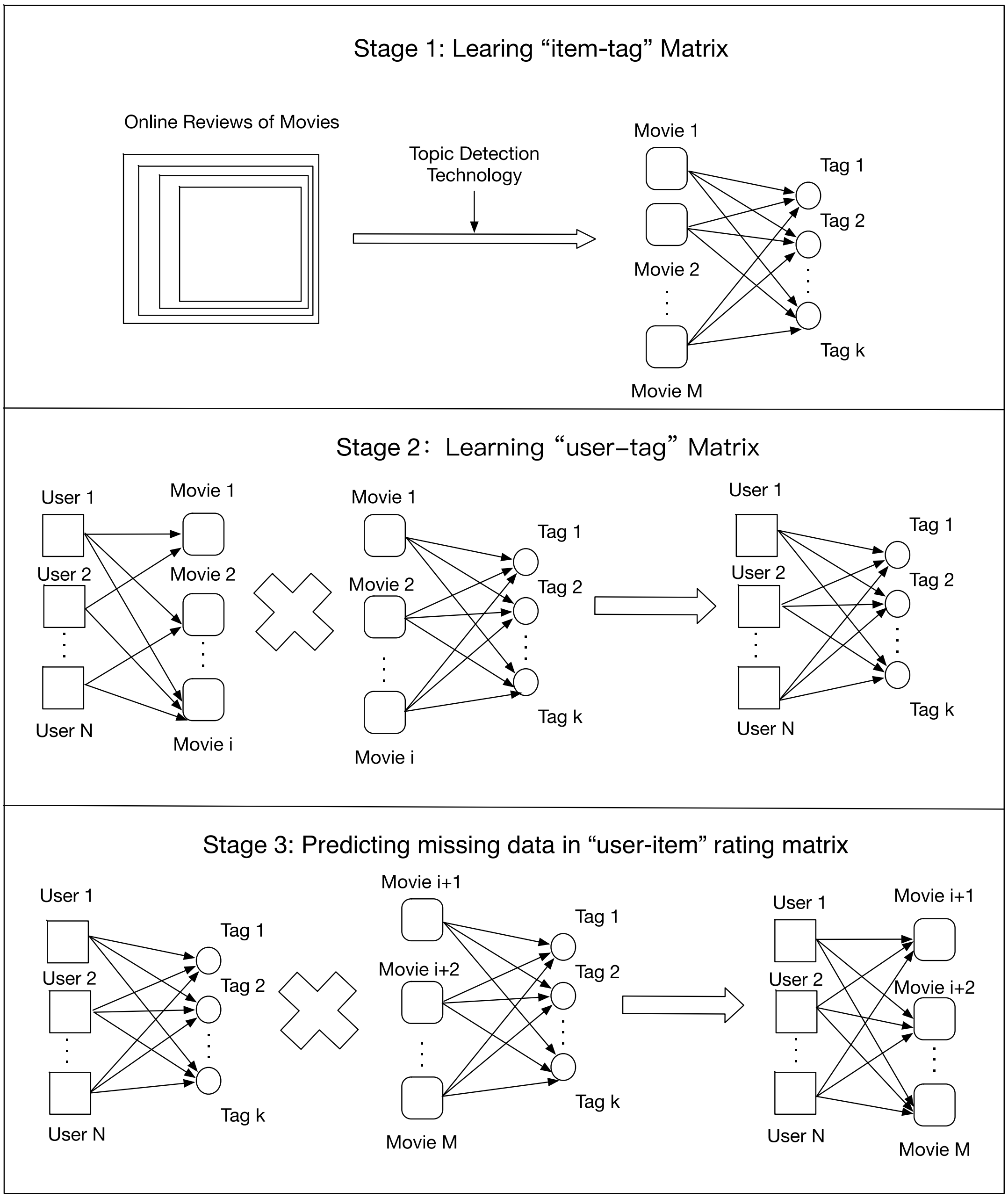

Fig. 2. Tag-based matrix factorization recommendation (TMFR) method

For each training case, the error between the actual rating and prediction rating is denoted as e_{ui}, where

$$e_{ui} = \gamma_{ui} - q_i^T p_u . \quad \textbf{(5)}$$

Vector p_u is iteratively updated by a magnitude proportional to λ in the opposite direction of the gradient:

$$p_u \leftarrow p_u + \gamma(e_{ui} * q_i - \gamma * p_u).$$

When e_{ui} equals zero or $|e_{ui}|$ is below the threshold, the iterative process stops, and p_u is obtained.

Stage 3: Predicting the missing data in the user-item rating matrix.

After the item-tag matrix $Q = (q_1, q_2, \ldots, q_M)^T$ and user-tag matrix $P = (p_1, p_2, \ldots, p_N)^T$ are learnt from the training data, the missing data of the user-item rating matrix in the test set is computed by $\tilde{R} = P^T \times Q$.

4 Experimental Evaluation

4.1 Experimental Setup

To empirically evaluate our proposed recommendation method, we compared it with the CMFR method and an alternative recommendation method based on TMFR. Accordingly, three experiments were conducted in this study.

Experiment 1: Because in previous research, the categories of items were most commonly used as factors in the recommendation methods, the performance of CMFR is tested first.

Experiment 2: Because some websites allow users annotate tags for items, the performance of the TMFR method is tested next. Here, tags of items are directly provided by users.

Experiment 3: As discussed, our proposed recommendation method makes use of tags extracted from online reviews as factors, so the performance of the TMFR method is also

The purpose of our experiments is to test how the different approaches of constructing factors can affect the performance of MFR methods. Experiment 1 is to test the performance of the most common m MFR method, where the factor vector is generated by the product category. Both experiment 2 and experiment 3 are to test whether the performance of MFR method to be improved, when the product labels are used to construct factor vectors. The different of MFR methods used in experiment 2 and experiment 3 is their different methods to construct product labels. The labels used in Experiment 2 are labeled by users, while the labels used in Experiment 3 are extracted from user online review through natural language processing technology. In this way, we can further investigate the impact of product labels generated in different ways on the performance of MFR method.

4.2 Datasets

All of the three experiments require user rating records. In addition, experiment 1 needs item category data, experiment 2 needs user tagging data, and experiment 3 needs online user text reviews.

Real data was collected for our comparative experiments using web crawling. All the data was from two movie review websites: MovieLens1 and IMDB2. MovieLens is a website run by the University of Minnesota. It provides rich data about movies for recommendation research，including movie categories, movie tags annotated by users, and individual rating records. Data used in experiments 1 and 2 are from MovieLens. Because MovieLens does not contain online user reviews, experiment 3 also uses online reviews data from IMDB in addition to the rating data collected from MovieLens. IMDB is a popular site that includes movie, TV, and celebrity content (IMDB,2015). It contains a considerable number of online reviews for each movie.

MovieLens divides movies into 20 different categories, so 3,000 movies (150 movies for each category) were randomly chosen for testing in the experiments. Note that some movies do not have enough user-annotated tags annotated to be used in experiment 2. After removing these movies, 2,272 movies were finally selected for the three experiments. Data, including 10,700 ratings of these movies by 597 users, the category information for each movie, and 50 online text reviews for each movie, were collected accordingly.

4.3 Experimental Process

The overall experimental process is divided into the following four steps.

Step 1: Factor selection and data preprocessing

In experiment 1, 20 different categories of movies form the factor collection.

In experiment 2, the tags of each movie in the dataset are collected first. This is done because users are not only allowed to annotate movies with tags, but also can approve or disapprove of the tags of other users. Hence, tags disapproved of by the majority of users are filtered out and tags with excessively low frequency (less than three) are also dropped. The final factor collection set consisted of 2,132 different tags.

[1] https://movielens.org/

[2] http://www.imdb.com/

In experiment 3, the tags of each movie were extracted from their online reviews using LDA. For each movie, the number of topics was set to three, and 45 tags (15 tags for each topic) were initially extracted from 50 online text reviews. After dropping tags with low probability (below 0.5) or low frequency (less than three), 1,365 different tags formed the final factor collection set.

Because the raw rating records collected from the MovieLens ranged between 1 and 10, the data was first normalized.

The total rating records were randomly divided into training and test sets, in a ratio of 4 to 1. The rating records in the training set were used in steps 2 and 3, and the rating records in the test set were used in step 4.

Step 2: Learning the item-factor matrix

As discussed in previous sections, to measure the extent to which items possesses factors, item-factor vector $q_i = \{q_{1i}, q_{2i}, \ldots, q_{Li}\}^T$ is introduced in this paper, where $i \in 1, 2, 3, \ldots, M$. The whole item-factor matrix is denoted as $Q = (q_1, q_2, \ldots, q_M)^T \in R^{L \times M}$. Item-factor vector and matrix of experiments are illustrated in Table 1.

Table 1. Item-factor vector and matrix of experiments

Experiments	Item-factor vector	Item-factor matrix
1	$q_i = (q_{1i}, q_{2i}, \ldots, q_{20i})^T$	$Q = (q_1, q_2, \ldots, q_{2272})^T R^{20\times 2272}$
2	$q_i = (q_{1i}, q_{2i}, \ldots, q_{2132i})^T$	$Q = (q_1, q_2, \ldots, q_{2272})^T R^{2132\times 2272}$
3	$q_i = (q_{1i}, q_{2i}, \ldots, q_{1365i})^T$	$Q = (q_1, q_2, \ldots, q_{2272})^T R^{1365\times 2272}$

Step 3: Learning the user-factor matrix

Similar to step 2, to measure the extent of interest the user has in various factors, user-factor $p_u = (p_{u1}, p_{u2}, \ldots, p_{uL})$ is introduced in the paper, where $u \in 1, 2, 3, \ldots, M, ,$. The whole user-factor matrix is denoted as $P = (p_1, p_2, \ldots, p_N)^T \in R^{N \times L}$. Given rating records and item-factor matrix, the user-factor matrix is learnt by gradient descent. User-factor vector and matrix of experiments are illustrated in table 2.

Table 2. User-factor vector and matrix of experiments

Experiments	User-factor vector	User-factor matrix
1	$p_u = (p_{u1}, p_{u2}, \ldots, p_{u20})^T$	$P = (p_1, p_2, \ldots, p_{N597})^T \in R^{597\times 20}$
2	$p_u = (p_{u1}, p_{u2}, \ldots, p_{u2132})^T$	$P = (p_1, p_2, \ldots, p_{N597})^T \in R^{597\times 2132}$
3	$p_u = (p_{u1}, p_{u2}, \ldots, p_{u1365})^T$	$P = (p_1, p_2, \ldots, p_{N597})^T \in R^{597\times 1365}$

Step 4: Predicting the user-item rating matrix

After learning the item-factor matrix and user-factor matrix from the training data set, the user-item rating matrix in the test set is predicted.

Here, in all the three experiments, item-factor matrix $\tilde{R} \in R^{597\times 2272}$

4.4 Evaluation Metrics

The most commonly used evaluation metrics are the mean absolute error (MAE) and the root mean square error (RMSE) [37].

MAE measures how close predicted values are to observed values and is given by:

$$MAE = \frac{1}{N} \times \frac{1}{M} \sum_{u=1}^{N} \sum_{i=1}^{M} | \gamma_{ui} - p_u{}^T q_i |. \tag{6}$$

RMSE is a frequently used measure of the sample standard deviation of the difference between the predictions and true outcomes and is given by:

$$RMSE = \sqrt[2]{\frac{1}{N} \times \frac{1}{M} \sum_{u=1}^{N} \sum_{i=1}^{M} (\gamma_{ui} - p_u{}^T q_i)^2} \, . \quad \textbf{(7)}$$

In our experiments, M=2,272 and N=597.

5 Results and Discussion

Fig. 3 and Fig. 4 illustrate the results of the experiments in terms of MAE and RMSE, respectively.

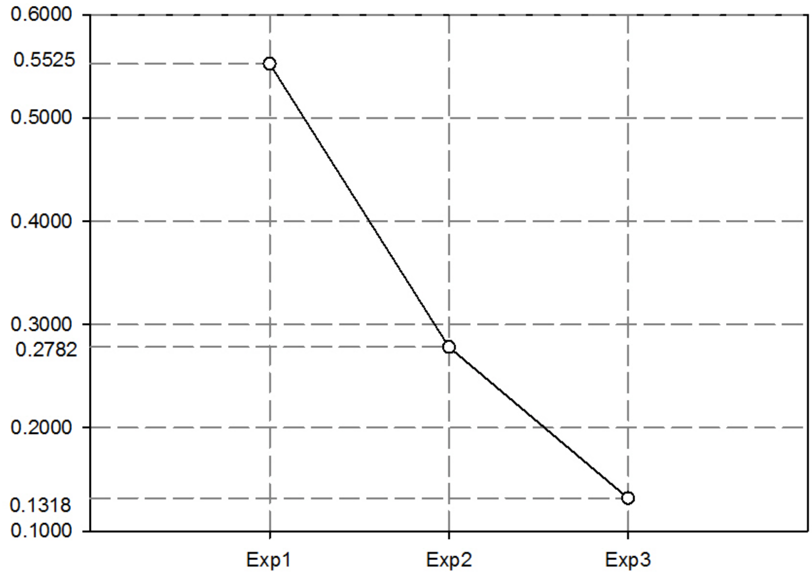

Fig. 3. MAE of the results

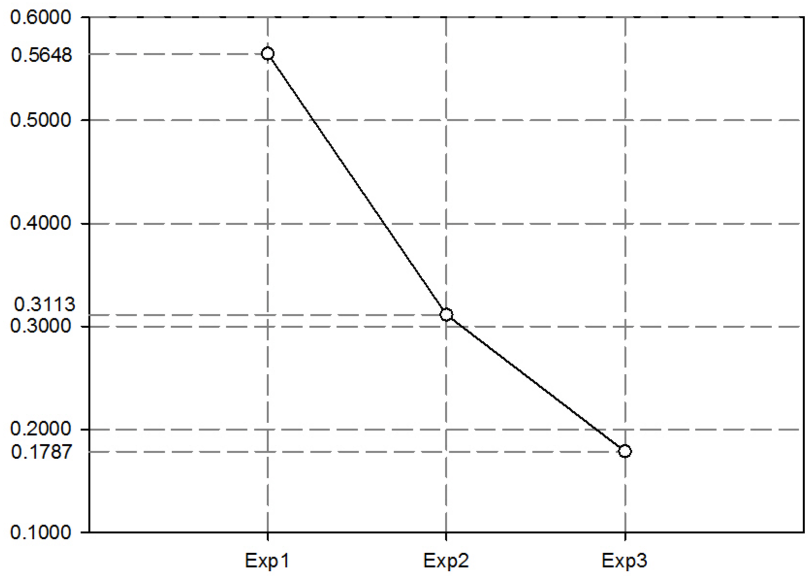

Fig. 4. RMSE of the results

First, the results show that the MAE value drops from 0.5525 (experiment 1) to 0.2782 (experiment 2), which is a reduction of 48%. Further, the RMSE value drops from 0.5648 (experiment 1) to 0.3113 (experiment 2), which is a reduction of 45%. Meanwhile, the results also show that the MAE value drops from 0.5525 (experiment 1) to 0.1318 (experiment 3), which is a reduction of 71%, and the RMSE value drops from 0.5648 (experiment 1) to 0.1787 (experiment 3), which is a reduction of 68%.

Both MAE and RMSE are the common measure of evaluating the performance of the recommendation algorithm. They compare the error between the predicted user score and the actual user score: the smaller the error, the better performance of the recommendation algorithm is. The experimental results reveals that the performance of Experiment 1 is the worst, that of Experiment 2 is in the middle, and that of Experiment 3 is the best.

These results demonstrate that compared with the traditional MFR method (i.e., the CMFR method), the performance of the TMFR method (whether the tags are directly provided by users or extracted from online user reviews) is improved significantly. The performance benefits from the increase of factor quality. Traditional MFR method usually constructed factor vectors through product content information provided by producers, who tend to hide the defects of items. On the contrary, our proposed method constructs factor vectors using product tags mining from user online review, which can express users' true feelings about the purchased items. Thus, tags from users are more appropriate for factor vector construction in MFR than category because of their completeness and genuineness with respect to item characterization.

Secondly, the results show that the MAE value drops from 0.2782 (experiment 2) to 0.1318 (experiment 3), which is a reduction of 53%, and the RMSE value drops from 0.3113 (experiment 2) to 0.1787 (experiment 3), which is a reduction of 43%.

These results demonstrate that compared with the alternative TMFR method, whose tags are directly provided by users, a higher-precision of rating prediction can be obtained by our proposed tag-based recommendation method. This is because of the increase in the quality of tags. In our recommendation method, tags are extracted from long online text reviews of items provided by users. Because the long text reviews are written down by users after careful thought, they can reflect a user's feeling about the items in a more reliable and comprehensive manner. On the contrary, tags generated from the existing generation mechanisms are a result of user decisions in a limited time. In fact, it is relatively difficult for users to represent the nature of items accurately within a very small number of words. As a result, in experiment 2, to raise the quality of the tags, the tags voted against by the majority of users were dropped, but their quality is still far below the tags used in our proposed recommendation method.

More interesting still, the number of tags used in experiment 2 is 2,132, which is nearly double those used in experiment 3 (1,365). Fewer tags lead to shorter rating prediction times. This indicates that using tags extracted from online user reviews in recommendation methods could not only improve the prediction accuracy, but also save prediction time.

In addition, as we have discussed before, these results also demonstrate that tags used in our proposed recommendation method have a higher quality than those from existing tag-generation mechanisms. This indicates that, beyond improving the recommendation performance, the tag extraction approach proposed in this paper is also suitable for the improvement of tag generation mechanisms.

The above experimental results demonstrate that our work have two contributions to recommendation methods: 1) we propose a novel approach of using product tags to construct factor vectors in MFR method. The novel approach can greatly improve the performance of MFR, which can be showed in the comparison of the results of Experiment 1 and Experiment 2. 2) we propose a novel method of mining product tags from user online review. Product tags generated from our proposed method can describe the product characteristics more accurately than tags labeled by users. Thus this proposed novel method can contribute more to the performance improvement of MFR method, which can be showed in the comparison of the results of Experiment 2 and Experiment 3.

6 Conclusions and Future Work

Factor quality largely determines the performance of the MFR method. In previous research, the categories of items are the main elements of the factor vector. However, they cannot characterize items accurately, which ultimately limits the performance of the CMFR method. Item tags, which come from users, are a better choice for constructing factor vector, because they can reflect users' true feelings about items. Unfortunately, there still exist problems with directly using tags that are annotated by users to construct factor vectors, because it is relatively difficult for most users to characterize items accurately using limited words.

A novel TMFR method is proposed in this paper. This method extracts tags from online long text user reviews through an LDA approach and constructs new factor vectors for the MFR method with the extracted tags. Because long text online reviews can fully reflect users' opinions of items, the tags extracted from them can be used to construct a factor vector of high quality. Experimental results on two large real-world datasets demonstrate that compared with the traditional CMFR method and alternative TMFR method (in which tags are from users' annotations), our proposed recommendation method can significantly improve the accuracy of rating prediction. Moreover, our proposed tag extraction approach can also be used to improve existing tag generation mechanisms.

In our proposed method, the number of online reviews used to extract tags is relatively large. However, new items usually obtain fewer reviews. As a result, how to develop the TMFR method for new items is one of our future research directions.

Furthermore, in our proposed method, we only focus on tag extraction for items. In fact, tags for users are also useful to improve recommendation. How to extract high-quality tags for users is another research direction.

Acknowledgements

This work is supported by an MOE (Ministry of Education in China) Youth Foundation Project of Humanities and Social Sciences (Project No. 18YJC860006).

References

[1] K. Hosanagar, A. Buja, Will the global village fracture into tribes? recommender systems and their effects on consumer fragmentation, Social Science Electronic Publishing 60(4)(2014) 805-823.

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

[3] D. Ben-Shimon, L. Rokach, B. Shapira, An ensemble method for top-N recommendations from the SVD, Expert Systems with Applications 64(2016) 84-92.

[4] Y. Ar, E. Bostanci, A genetic algorithm solution to the collaborative filtering problem, Expert Systems with Applications, 61(2016) 122-128.

[5] G. Guo, J. Zhang, D. Thalmann, Merging trust in collaborative filtering to alleviate data sparsity and cold start, Knowledge-Based Systems 57(2)(2013) 57-68.

[6] J. Zhang, Q. Peng, S. Sun, C. Liu, Collaborative filtering recommendation algorithm based on user preference derived from item domain features, Physica A Statistical Mechanics & Its Applications 396(2)(2014) 66-76.

[7] A. Ghoshal, S. Menon, S. Sarkar, Recommendations using information from multiple association rules: a probabilistic approach, Information Systems Research 26(3)(2014) 532-551.

[8] J. Liu, C. Wu, W. Liu, Bayesian probabilistic matrix factorization with social relations and item contents for recommendation, Decision Support Systems 55(3)(2013) 838-850.

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

[10] Y. Shi, M. Larson, A. Hanjalic, Mining contextual movie similarity with matrix factorization for context-aware recommendation, ACM Transactions on Intelligent Systems & Technology 4(1)(2010) 385-388.

[11] N. Zheng, Q. Li, A recommender system based on tag and time information for social tagging systems, Expert Systems with Applications 38(4)(2011) 4575-4587.

[12] K. Hashimoto, G. Kontonatsios, M. Miwa, S. Ananiadou, Topic detection using paragraph vectors to support active learning in systematic reviews, Journal of Biomedical Informatics 62(2016) 59-65.

[13] R.-D. Du, J.-J. Lu, H.-Y. Cai, Double regularization matrix factorization recommendation algorithm, IEEE Access 10(11)(2019) 1-1.

[14] H.-T. Liu, C. Ling, L.-Q. Yang, P. Zhao, Supervised convolutional matrix factorization for document recommendation, International Journal of Computational Intelligence and Applications 17(4)(2018) 1850018.

[15]Y.-L. Gu, J.-X. Song, W.-D. Liu, L.-X. Zou, Y. Yao, CAMF: context aware matrix factorization for social recommendation, Web Intelligence 16(10)(2018) 53-71.

[16] Y. He, C. Wang, C.-J. Jiang, Correlated matrix factorization for recommendation with implicit feedback, in: Proc. IEEE Transactions on Knowledge and Data Engineering, 2018.

[17] R.-P. Shen, H.-R. Zhang, H. Yu, F. Min, Sentiment based matrix factorization with reliability for recommendation, Expert Systems with Applications 135(10)(2019) 249-258.

[18] C. Wang, D.M. Blei, Collaborative topic modeling for recommending scientific articles, in: Proc. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego Ca Usa 8, 2011.

[19] M. Yang, L.I. Ying-Ming, Z. Zhang, Scientific articles recommendation with topic regression and relational matrix factorization, Computers & Electronics 15(11)(2014) 984-998.

[20] H. Shan, A. Banerjee, Generalized probabilistic matrix factorizations for collaborative filtering, in: Proc. 10th IEEE International Conference on Data Mining, 2010.

[21] S. Purushotham, Y. Liu, C.J. Kuo, Collaborative topic regression with social matrix factorization for recommendation systems, Computer Science 3(18)(2012) 1206.

[22] X. Xin, C.-Y. Lin, X.-C. Wei, H.-Y. Huang, When factorization meets heterogeneous latent topics: an interpretable cross-site recommendation framework, Journal of Computer Science and Technology 30(4)(2015) 917-932.

[23] J. Yin, Y. Xu, Personalised QoS-based web service recommendation with service neighbourhood-enhanced matrix factorization, International Journal of Web &Grid Services 11(1)(2015) 1206.

[24] R. Forsati, M. Mahdavi, M. Shamsfard, M. Sarwat, Matrix factorization with explicit trust and distrust side information for improved social recommendation, ACM Transactions on Information Systems 32(4)(2014) 1-2.

[25] X. Zhao, Z. Niu, W. Chen, C. Shi, K. Niu, D. Liu, A hybrid approach of topic model and matrix factorization based on two-step recommendation framework, Journal of Intelligent Information Systems 44(3)(2015) 335-353.

[26] D. Miao, Collaborative filtering recommendation algorithm based on trust propagation international, Journal of Security & Its Applications 9(7)(2015) 99-108.

[27] Y. Shi, M. Larson, A. Hanjalic, Unifying rating-oriented and ranking-oriented collaborative filtering for improved recommendation, Information Sciences 229(6)(2013) 29-39.

[28] P. Shi, Y. Qu, H. Liang, Personalized image tag recommendation algorithm for web2.0 platform utilizing tensor factorization, in: Proc. International Conference on Intelligent Systems Design & Engineering Applications, 2014.

[29] M.F. Alhamid, M. Rawashdeh, M.A. Hossain, A. Alelaiwi, A.E. Saddik, Towards context-aware media recommendation based on social tagging, Journal of Intelligent Information Systems 46(3)(2016) 499-516.

[30] H. Kim, H.J. Kim, A framework for tag-aware recommender systems, Expert Systems with Applications 41(8)(2014) 4000-4009.

[31] Z.-K. Zhang, L. Fang, K. Chuang, Website-oriented recommendation based on heat spreading and tag-aware collaborative filtering, Physica A Statistical Mechanics & Its Applications 399(4)(2014)82-88.

[32] D.I. Ignatov, S.I. Nikolenko, T. Abaev, J. Poelmans, Online recommender system for radio station hosting based on information fusion and adaptive tag-aware profiling, Expert Systems with Applications 55(2016) 546-558.

[33] H. Movahedian, M.R. Khayyambashi, A semantic recommender system based on frequent tag pattern, Intelligent Data Analysis 19(1)(2015) 109-126.

[34] F. Gedikli, Improving recommendation accuracy based on item-specific tag preferences, ACM Transactions on Intelligent Systems & Technology 4(1)(2013) 43-55.

[35] H.N. Kim, A.E. Saddik, Exploring social tagging for personalized community recommendations, User Modeling and User-Adapted Interaction 23(2-3)(2012) 249-285.

[36] H.N. Kim, A. Alkhaldi, A.E. Saddik, G.S. Jo, Collaborative user modeling with user-generated tags for social recommender systems, Expert Systems with Applications 38(7)(2011) 8488-8496.

[37] H. Movahedian, M.R. Khayyambashi, A tag-based recommender system using rule-based collaborative profile enrichment, Intelligent Data Analysis 18(5)(2014) 953-972.

[38] A.B.B. Martinez, M.R. Lopez, E.C. Montenegro, F.A.M. Fonte, J.C. Burguillo, A. Peleteiro, Exploiting Social tagging in a web 2.0 recommender system, Internet Computing IEEE 14(6)(2010) 23-30.

[39] A. Nanopoulos, Item recommendation in collaborative tagging systems, IEEE Transactions on Systems Man and Cybernetics - Part A Systems and Humans 41(4)(2011) 760-771.