ICIC Express Letters
Part B: Applications
Volume **9**, Number **2**, February **2018**

ICIC International ©2018 ISSN 2185-2766
pp. 145–153

A NEW TAG RECOMMENDATION METHOD BASED ON USER CLUSTERING WITH TENSOR FACTORIZATION

Hui Zeng, Qiang Hu, Xiaohui Huang, Liyan Xiong and Chunping Tu

School of Information Engineering
East China Jiaotong University
No. 808, Shuanggang East Avenue, Nanchang 330013, P. R. China
macrohui29@sina.com; huqiang1993@foxmail.com; hxh016@hotmail.com
xly_ecjtu@163.com; tcp@ecjtu.edu.cn

Received August 2017; accepted November 2017

Abstract. *Tag recommendation is widely used in various applications, such as movies websites, and electronic business websites. In this paper, we propose a new tag recommendation method based on user clustering with tensor factorization to further improve the quality of tag recommendation. In the method, we firstly cluster some users who have important influence on items, then calculate the comprehensive weights by the relationship among users, items, tags and the items' rating. At last, we conduct tensor factorization on a tensor constructed by using user clusters, and their tags and items. In contrast to traditional tensor factorization methods like Higher Order Singular Value Decomposition (HOSVD) and 0/1 Scheme, we empirically show that our proposed method outperforms the compared tag recommendation methods.*
Keywords: Tag recommendation, Tensor factorization, Weight, Clustering

1. **Introduction.** In general, tag recommendation means that different users can employ different tags, e.g., a list of words, to annotate an item (e.g., website and movie), then recommends the appropriate items to users by predicting the user's behavior in future according to their past tagging behavior. For example, if two different users have already marked the same item, they trend to use the same tag to annotate another item in the future.

Nowadays, some tag recommendation methods rank tags effectively by means of tensor factorization techniques. Higher Order Singular Value Decomposition (HOSVD) [1], Ranking Tensor Factorization (RTF) [2] and Multiverse [3] are based on the Tucker decomposition model. RTF was shown to result in a good predictor quality. Rendle et al. [2] introduced two different tensor data interpretations: the 0/1 interpretation scheme and post-based ranking interpretation scheme. Yang [4] presented a weighting data interpretation scheme for ternary relations of users, items and tags. In this paper, we propose a new method which calculates the comprehensive weight by the relationship among users, items, tags and items' rating. Our experimental results show that our proposed model is superior to other compared methods.

The contributions of our work are summarized as follows. First, three different interpretation schemes of tensor data are generalized. Second, a weighting method is proposed to consider four important factors which are user, item, tag and rating. Third, compared to other traditional methods, the experimental results demonstrate that our method outperforms other methods in real data sets.

2. **Related Work.** There are a lot of previous works on tag recommendation. A work by Krestel et al. [5] proposed an approach based on the Latent Dirichlet Allocation (LDA). For a document, the top terms which constitute latent topics are recommended to the

users; in this way, the terms in the same topic have the opportunity to be presented in other documents. However, the system is impersonalized.

The clustering methods are also applied in the literature of recommender systems. In [6], authors incorporated social tags into two clustering methods: K-Means and a generative clustering method based on LDA. Although their work demonstrated the value of tags as an additional information source for clustering, there was a lack of user dimension in their clustering models, which led to a slightly lower F1 score.

Factorization models for tensors are studied in several fields for many years [7-9]. Kolda and Bader [10] described two different tensor factorization methods of CANDECOMP/PARAFAC and Tucker decomposition. Frolov and Oseledets [11] introduced the applications of tensors in social tagging systems and various related algorithms for tensor decomposition.

3. **Tag Recommendation.**

The task of the tag recommendation provides users with a personalized list of tags for specific items. For example, when an audience (user) wants to tag a movie (item), the movie site should recommend audience a list of keywords that he or she wants. The list of recommended tags can be learned from the annotating behavior of the past of this user for other tags and the annotating behavior of other users for both this and other tags.

3.1. **Formalization.**

Assuming that U, I and T are the collection of all users, the set of all items/resources and the set of all tags, respectively. The historical tagging information is given by $A \subseteq U \times I \times T$. As this is a ternary relation over categorical variables, it can be seen as a three-dimensional tensor where the triples in A are the positive observations in the past [8]. The ternary relation $(u, i, t) \in A$ would mean that user u employs item i and utilizes tag t to mark the item i. In this paper, combinations (u, i), (u, t), (i, t) are defined as sets of P_S, Q_S, R_S respectively: $P_S := \{(u, i)|\exists t \in T : (u, i, t) \in A\}$, $Q_S := \{(u, t)|\exists i \in I : (u, i, t) \in A\}$ and $R_S := \{(i, t)|\exists u \in U : (u, i, t) \in A\}$. Here, P_S, Q_S, R_S can be viewed as two-dimensional projections of A on the user/item, user/tag and item/tag dimensions respectively.

3.2. **Interpretation of the data.**

Tensor data can be represented in different ways, and each representation can lead to different recommendation algorithms. The following is mainly about three forms of presentation as an introduction.

3.2.1. *0/1 interpretation scheme.* The set of triples in Y can be expressed as a third-order tensor. Symeonidis [1] proposed to interpret Y as a sparse tensor in which 1 indicates positive feedback and 0 missing values (see Figure 1), the training data $Y^{0/1}$ is defined as:

$$y_{u,i,t}^{0/1} = \begin{cases} 1, & (u, i, t) \in A \\ 0, & \text{else} \end{cases}.$$

The 0/1 interpretation scheme [2] has expressed semantic inaccuracies as well as low precision defects.

3.2.2. *Post-based ranking interpretation scheme.* Rendle et al. [2], distinguish between positive/negative examples and missing values in order to learn personalized ranking of tags. The idea is that positive and negative examples are only generated from the observed tag distribution. The observed tag assignments are interpreted as positive feedback, while unmarked tag assignments of marked resources are negative evidence. All other entries are assumed to be missing values (see Figure 2).

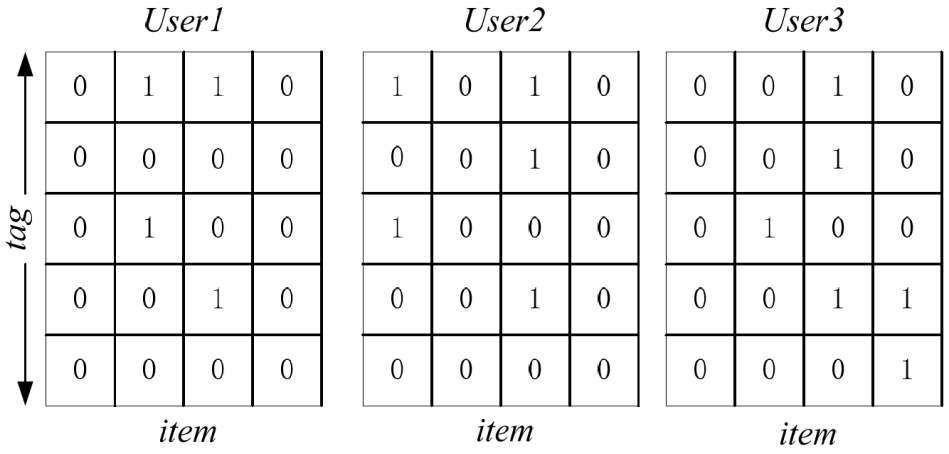


FIGURE 1. 0/1 interpretation: Positive examples are encoded as 1 and the rest as 0.

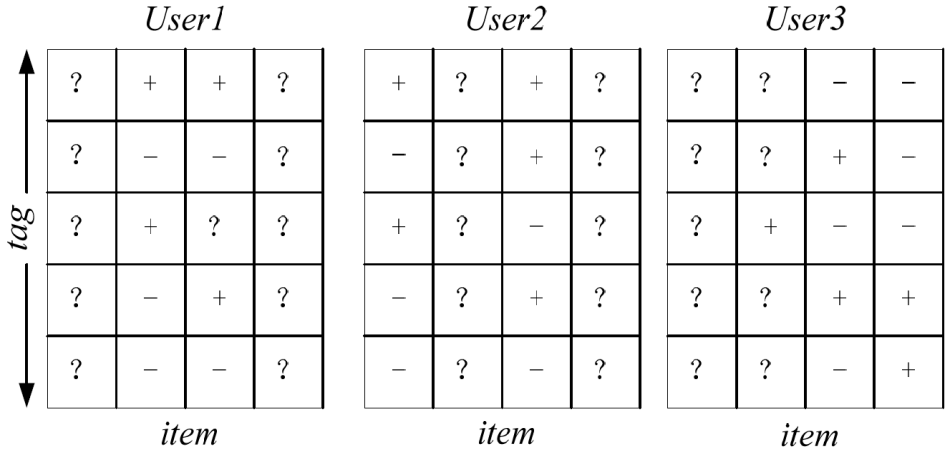


FIGURE 2. The labeled resource is positive feedback, and the unobserved mark is negative feedback.

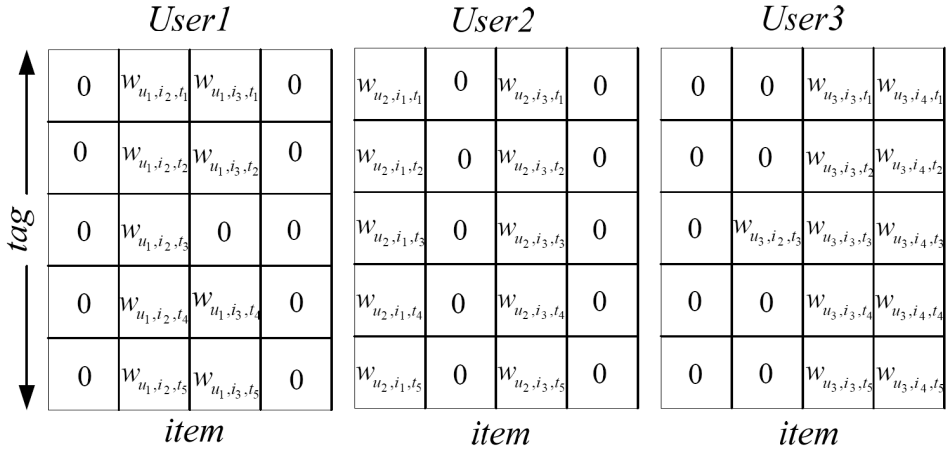


FIGURE 3. Each ternary relationship corresponds to a total weight, while unmarked items mean zero.

3.2.3. *Comprehensive weight interpretation scheme.* The relationship of the triple is expressed as a comprehensive weight which takes account of the weight relations on each dimension to obtain the total weight w_{u_j,i_k,t_l} of the ternary tuple of a "user-item-tag" (see Figure 3). The weight refers to the user j utilizing the tag l to mark the item k to represent the user's preference for the item.

4. **Methodology.** Firstly, we describe the tensor factor decomposition model. Then, we explain how to calculate the method of constructing the comprehensive weight of the tensor in detail. However, before calculating the overall weight of the user cluster, the user needs to be clustered to reconstruct the tensor. In general, the user's degree of activity to distinguish the user's category can be divided into active users, ordinary users and inactive users. This paper clusters based on the similarity between the ordinary user and the active user and then recommends to the ordinary user. The calculation method will be analyzed in the following specific way.

4.1. **Tensor factorization model.** In this paper, the method of tensor decomposition is mainly Tucker decomposition, which is a form of high-order principal component analysis. It breaks a tensor into a core tensor multiplied by a matrix on each mode. As shown in Figure 4 for a ternary relation tensor, it is decomposed into three low rank matrices and a core tensor, which represents the interaction on each dimension and retains the main information of the original tensor and has certain stability.

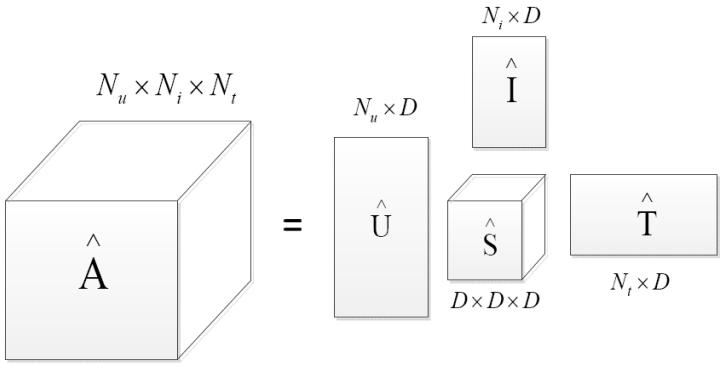


FIGURE 4. Tucker decomposition of a three-way array

The factorization of A is expressed as: $\hat{A} := \hat{S} \times_u \hat{U} \times_i \hat{I} \times_t \hat{T}$. Here, $\hat{U} \in P^{N_u \times D}$, $\hat{I} \in P^{N_i \times D}$, and $\hat{T} \in P^{N_t \times D}$ are low-rank feature matrices representing an entity [11], i.e., user, item, and tags. N_u, N_i and N_t are the dimensions of the low-rank approximation and $\hat{S} \in P^{D \times D \times D}$ is the core tensor which represents interactions between the latent factors. The symbol $\times_i$ denotes the i-mode multiplication between a tensor and a matrix. Rendle et al. [2] denote the model parameters by the quadruple $\hat{\theta} := \left(\hat{S}, \hat{U}, \hat{I}, \hat{T}\right)$. After the feature matrices and the core tensor are learned, predictions can be done as follows:

$$\hat{a}_{u,i,t} = \sum_{\tilde{u}} \sum_{\tilde{i}} \sum_{\tilde{t}} \hat{s}_{\tilde{u},\tilde{i},\tilde{t}} \cdot \hat{u}_{u,\tilde{u}} \cdot \hat{i}_{i,\tilde{i}} \cdot \hat{t}_{t,\tilde{t}} \tag{1}$$

where indices over the feature dimension of a feature matrix are marked with a tilde, and elements of a feature matrix are marked with a hat (e.g., $\hat{u}_{u,\tilde{u}}$). The predicted values of tensors can be derived from Formula (1), and the Top-N personalized recommendation list is generated [8]:

$$\text{Top}(u, i, N) = \arg\max_{t \in T}^{N} \hat{a}_{u,i,t}. \tag{2}$$

4.2. **Calculation method of comprehensive weight.**

4.2.1. *User clustering.* This article will be divided into three categories of users: active users, ordinary users, inactive users. Active users are the most representative users who employ a lot of items and evaluate the use of more labels on the item, naturally they are of great importance, so their weight will be greater. On the contrary, the weight of the inactive user will be very small. Ordinary users refer to users in addition to active users

and inactive users, accounting for the vast majority of users in the collection, but also the main recommendation of the object.

In order to improve the accuracy of recommendation, the "user-item-tag" ternary of similar users with common interest is divided into the same small data set.

First, the importance of the items is calculated using by users. Its importance w_i is proportional to the number of users employing the item. Similarly, the importance of a tag is proportional to the number of users using the tag, defined as w_t. Their importance can be calculated as follows:

$$w_{i_k} = \frac{\sum\limits_{(u_k,i_k)\in P_S} u_k}{|U|}, \quad w_{t_k} = \frac{\sum\limits_{(u_k,t_k)\in Q_S} u_k}{|U|}. \tag{3}$$

Since the number of items used by the user is too large which makes the weight get too heavy, it is necessary to set a parameter p_i to make the total weight w_{u_i} less than or equal to 1. Similarly, the weight w_{u_t} of the tag is calculated as follows:

$$w_{u_i} = p_i * \sum_{(u_k,i_k)\in P_S} w_{i_k}, \quad w_{u_t} = p_t * \sum_{(u_k,t_k)\in Q_S} w_{t_k}. \tag{4}$$

The user's weights are calculated by w_{u_i} and w_{u_t}. The weight between the user and the item is more important than the weight between the user and the tag, so the parameter α will be relatively large, but between 0 and 1:

$$w_u = \alpha * w_{u_i} + (1-\alpha) w_{u_t} \quad (\alpha \in (0,1)). \tag{5}$$

The weights w_u of all users are sorted first, and then the first N users are selected as active user sets. The users at the bottom are inactive user sets, and the last remaining users as the ordinary user collection.

This algorithm computes the similarity between ordinary users and active users to cluster. The similarity computation is similar to the user-based collaborative filtering algorithm, which uses cosine similarity method. It chooses the set of items employed by active users u_a and ordinary users u_o as vectors of the m-dimension, whose vector value is the number of times the global item is used by the user. The similarity calculation method is as follows:

$$S(u_a,u_o) = \cos(\vec{u}_a \cdot \vec{u}_o) = \frac{\vec{u}_a \cdot \vec{u}_o}{\|\vec{u}_a\| \times \|\vec{u}_o\|} = \frac{\sum\limits_{i_k\in(I_{u_a}\cap I_{u_o})} times_{u_a,i_k} \cdot times_{u_o,i_k}}{\sqrt{\sum\limits_{i_k\in(I_{u_a}\cap I_{u_o})} times^2_{u_a,i_k}} \sqrt{\sum\limits_{i_k\in(I_{u_a}\cap I_{u_o})} times^2_{u_o,i_k}}}. \tag{6}$$

The results of similarity are sorted, and a plurality of clustering sets are obtained by iterating and clustering the active users and the users with high similarity.

4.2.2. *Calculate the weights of ternary tuples.* The user weights have been computed in the previous section. In order to obtain the three-dimensional tensor of the comprehensive weight, it is necessary to find the weight of the item and the tag. The weight calculation method of the item is a little different from the calculation method of the user's weight. There are a difference between the weight calculation method of the items and the users' weight calculation method. It not only needs the user's weight of the item and the weight of the tag on the item, but also the user's score on the item.

The user's weighting $w_{i_k,u}$ of the item is calculated by averaging the weights of all the users who have used the item. The weight $w_{i_k,t}$ of the tag to the item is calculated based on the number of tags to mark the item and the proportion of the total tag that marks the item. The score s_{i_k,u_s} of the item is averaged according to the rating given by the user using the item. Similarly, the weight of the item i_k is determined by setting the different

parameters λ_i to combine the user and the tag weights in the item, and the user's rating on the item. The formula is shown in Equations (7) and (8):

$$w_{i_k,u} = \frac{\sum\limits_{(u_l,i_k)\in P_S} w_{u_l}}{|U_{i_k}|}, \quad w_{i_k,t} = \frac{\sum\limits_{(i_k,t_l)\in R_S} t_l}{|T_{i_k}|}, \quad s_{i_k,u_s} = \frac{\sum\limits_{(i_k,u_s)\in P_S} u_s}{|U_{i_k}|}, \tag{7}$$

$$w_{i_k} = \lambda_1 \cdot w_{i_k,u} + \lambda_2 \cdot w_{i_k,t} + \lambda_3 \cdot s_{i_k,u_s} \quad (\lambda_i \in (0,1)). \tag{8}$$

Since the weight of the tag is taken into account in the calculation of the weight of the user and the weight of the item in the preceding paragraph, the weighting of the tag will ignore the influence of the user and the item. The weight of the tag is calculated by the number of occurrences of the tag t_k in the ternary tuple and the maximum number of occurrences of the tag used:

$$w_{t_k} = \frac{\sum\limits_{(u_k,i_k,t_k)\in A} t_k}{\max(t)}. \tag{9}$$

According to the users' weight w_u, the items' weight w_i and the tags' weight w_t, different parameters μ_i are set up to obtain the formula of the comprehensive weight:

$$w_{u_j,i_k,t_l} = \mu_1 \cdot w_{u_j} + \mu_2 \cdot w_{i_k} + \mu_3 \cdot w_{t_l} \quad (\mu_i \in [0,1]). \tag{10}$$

5. Experimental Results and Analysis.

5.1. **Datasets.** In this paper, MovieLens data sets are used to preprocess, and finally 163295 records are obtained. The statistic description of these datasets is shown in Table 1. It contains 500 users, 9289 movies, 1128 tags, and the score range of 0.5-5, the score interval is 0.5. According to the user cluster, 500 users were divided into six categories, and two types of data are selected to compare experiment.

TABLE 1. Datasets description

Dataset	User	Item	Tag	Total
cluster1	100	5196	1123	59524
cluster2	84	4002	1108	28338

5.2. **Evaluation methodology.** This paper uses common evaluation protocols to evaluate tag recommendation. The method extracts parts of the items randomly from each user in the data set as the test set S_{test} and the rest as the training set S_{train}. This article measures the recall and accuracy of each Top-1 to Top-30 list and the F1-measures of the average recall and precision for each Top-i:

$$\text{Precision}(S_{test}, N) = \operatorname*{avg}_{(u,i)\in S_{test}} \frac{|\text{Top}(u,i,N) \cap \{t|(u,i,t) \in S_{test}\}|}{N}, \tag{11}$$

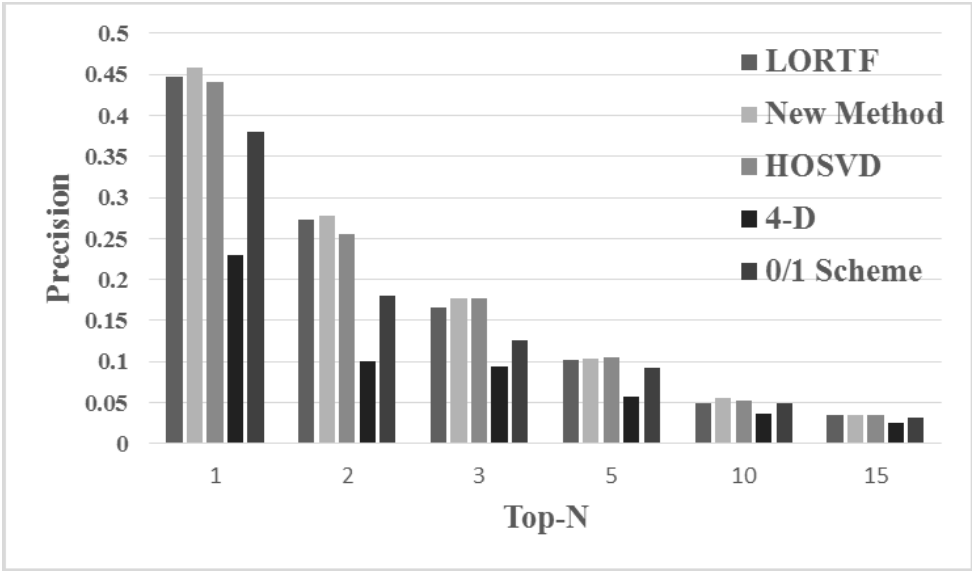
$$\text{Recall}(S_{test}, N) = \operatorname*{avg}_{(u,i)\in S_{test}} \frac{|\text{Top}(u,i,N) \cap \{t|(u,i,t) \in S_{test}\}|}{\{t|(u,i,t) \in S_{test}\}}, \tag{12}$$

$$\text{F1}(S_{test}, N) = \frac{2 \cdot \text{Precision}(S_{test}, N) \cdot \text{Recall}(S_{test}, N)}{\text{Precision}(S_{test}, N) + \text{Recall}(S_{test}, N)}. \tag{13}$$

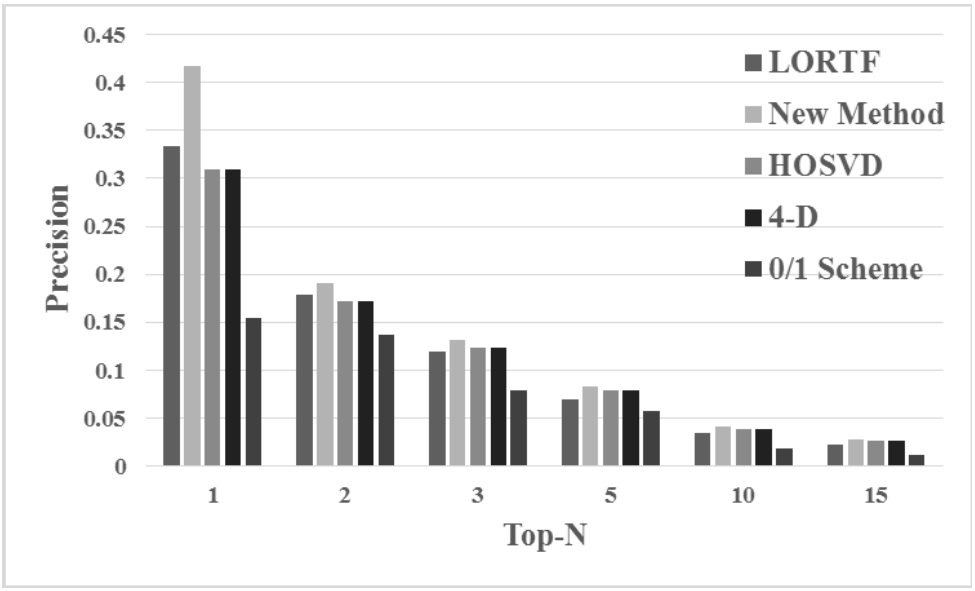
In this paper, the precision values and the F1-measures on the Top-N list are chosen as the primary quality measure in order to compare the results directly with the measure work. For two sets of clustering data, we use $(k_u, k_i, k_t) = (8, 8, 8)$ dimensions for tensor decomposition. For HOSVD, 0/1 Scheme and LORTF, we use the same data set and split method, and use the dimensions of $(8, 8, 8)$. This paper also takes the users' rating, the user, the item and the tag into a four-dimensional tensor which is named 4-D as a contrast experiment.

5.3. **Experimental results and analysis.** In the following, we used two clusters of experimental training data to compare multiple algorithms. In Figure 5, the state-of-the-art models HOSVD, 0/1 Scheme, LORTF and 4-D are compared to our model. Compared with the results of two sets of training data, it is obvious that the accuracy of the algorithm is better than other algorithms. The reason may be that the algorithm takes account of the four important factors that lead to an effective improvement in accuracy. As the predicted Top-N increases, the accuracy of prediction is smaller. In addition, since the sparseness of cluster1 is smaller than that of cluster2, the accuracy of 5(a) is slightly better than that of 5(b).

Figure 6 shows an F-Scores comparison of the state-of-the-art models HOSVD, 0/1 Scheme, LORTF and Four-dimensional tensor to our model. The F-Scores of this algorithm are significantly higher than those of other algorithms, but the overall F-Scores are small, which is due to the fact that the data set is too sparse.

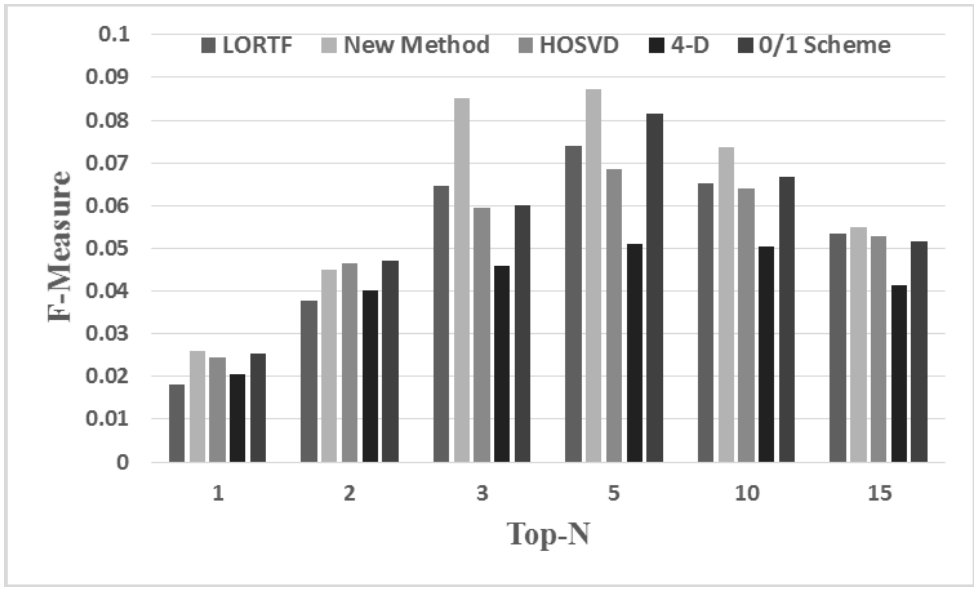


(a)

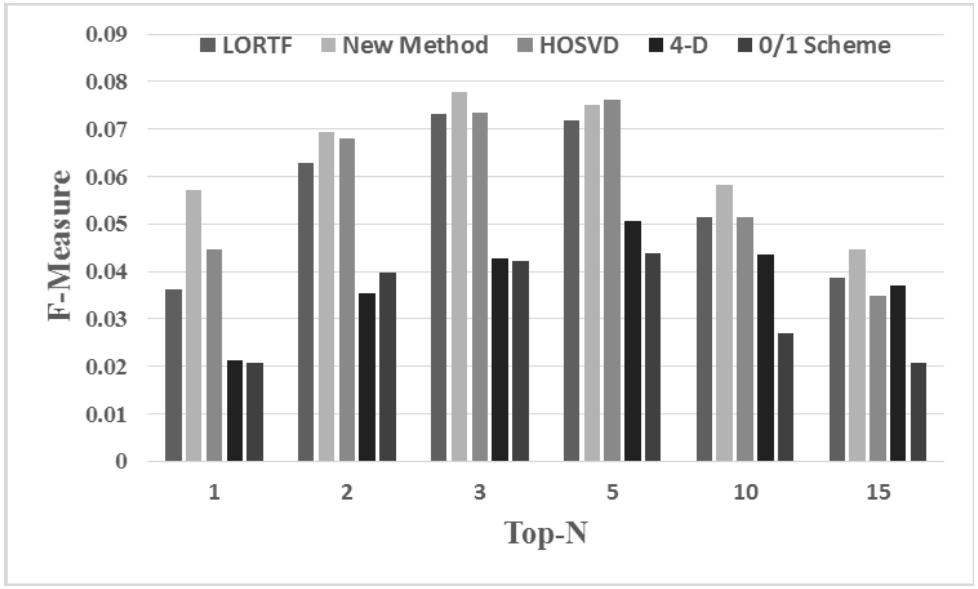


(b)

FIGURE 5. The precision of the Top-1 to Top-15 lists on two datasets in which (a) is the precision of cluster1, and (b) is the precision of cluster2



(a)



(b)

FIGURE 6. The F1-measure of the Top-1 to Top-15 lists on two datasets in which (a) is the F-measure of cluster1, and (b) is the F-measure of cluster2

6. **Conclusion and Future Works.** In the paper, three different tensor data representations are surveyed, including 0/1 interpretation scheme, post-based ranking interpretation scheme, and comprehensive weight interpretation scheme. Then, we cluster the user's weights and put forward a combination of users, items, tags, scores of the comprehensive weight of the calculation method. This paper reveals our approach evaluation result compared with the state-of-the-art modes HOSVD, LORTF and other methods, and the results demonstrate that our method is better than other methods.

For future work, we plan to study from two aspects. On the one hand, some parameters can be improved in the calculation of the weighting method or combined with some important influencing factors. On the other hand, we plan to find a better way to cope with the sparseness of data to improve the speed of operation and the accuracy of the prediction.

Acknowledgement. This research was supported by the National Natural Science Foundation of China (NSFC) under Grant No. 61363072 and No. 61562027, Science and Technology Department of Jiangxi Province under Grant No. 20161BBI90032, 20151BB990041, 20161BAB212050, Education Department of Jiangxi Province under Grant No. GJJ1605 08. The authors also gratefully acknowledge the helpful comments and suggestions provided by the reviewers.

REFERENCES

[1] P. Symeonidis, User recommendations based on tensor dimensionality reduction, *ACM Conference on Recommender Systems*, pp.43-50, 2008.

[2] S. Rendle, L. B. Marinho, A. Nanopoulos et al., Learning optimal ranking with tensor factorization for tag recommendation, *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Paris, France, pp.727-736, 2009.

[3] A. Karatzoglou, X. Amatriain, L. Baltrunas et al., Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering, *ACM Conference on Recommender Systems*, Barcelona, Spain, pp.79-86, 2010.

[4] Q. Yang, *Research of Learning Optimal Ranking with Tensor Factorization for Recommendation System*, Master Thesis, South China University of Technology, Guangzhou, 2012.

[5] R. Krestel, P. Fankhauser and W. Nejdl, Latent dirichlet allocation for tag recommendation, *ACM Conference on Recommender Systems*, New York, NY, USA, pp.61-68, 2009.

[6] D. Ramage, P. Heymann, C. D. Manning et al., Clustering the tagged web, *ACM International Conference on Web Search and Data Mining*, Barcelona, Spain, pp.54-63, 2009.

[7] E. Acar, D. M. Dunlavy and T. G. Kolda, A scalable optimization approach for fitting canonical tensor decompositions, *Journal of Chemometrics*, vol.25, no.2, pp.67-86, 2015.

[8] S. Rendle and L. Schmidt-Thieme, Pairwise interaction tensor factorization for personalized tag recommendation, *ACM International Conference on Web Search and Data Mining*, pp.81-90, 2010.

[9] G. Zhou, A. Cichocki, Q. Zhao et al., Nonnegative matrix and tensor factorizations: An algorithmic perspective, *IEEE Signal Processing Magazine*, vol.31, no.3, pp.54-65, 2014.

[10] T. G. Kolda and B. W. Bader, Tensor decompositions and applications, *Sandia Reports*, vol.51, no.3, pp.455-500, 2009.

[11] E. Frolov and I. Oseledets, Tensor methods and recommender systems, *Data Mining and Knowledge Discovery*, vol.7, no.3, 2017.