

Research Article

Recommending Mobile Microblog Users via a Tensor Factorization Based on User Cluster Approach

Xiangwen Liao ^{1,2,3}, Lingying Zhang,^{1,2} Jingjing Wei,⁴
Dingda Yang ^{1,2} and Guolong Chen^{1,2}

¹College of Mathematics and Computer Science, Fuzhou University, Fuzhou, China

²Fujian Provincial Key Laboratory of Network Computing and Intelligent Information Processing, Fuzhou University, Fuzhou, China

³Fujian Provincial Key Laboratory of Information Processing and Intelligent Control, Minjiang University, Fuzhou, China

⁴College of Electronics and Information Science, Fujian Jiangxia University, Fuzhou, China

Correspondence should be addressed to Xiangwen Liao; liaoqxw@fzu.edu.cn

Received 27 March 2018; Accepted 31 July 2018; Published 3 October 2018

Academic Editor: Tianyong Hao

Copyright © 2018 Xiangwen Liao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

User influence is a very important factor for microblog user recommendation in mobile social network. However, most existing user influence analysis works ignore user's temporal features and fail to filter the marketing users with low influence, which limits the performance of recommendation methods. In this paper, a Tensor Factorization based User Cluster (TFUC) model is proposed. We firstly identify latent influential users by neural network clustering. Then, we construct a features tensor according to latent influential user's opinion, activity, and network centrality information. Furthermore, user influences are predicted by the latent factors resulting from the temporal restrained CP decomposition. Finally, we recommend microblog users considering both user influence and content similarity. Our experimental results show that the proposed model significantly improves recommendation performance. Meanwhile, the mean average precision of TFUC outperforms the baselines with 3.4% at least.

1. Introduction

Microblogging services, such as Twitter or Weibo, have been one of the most popular platforms for individuals to exchange information by posting messages or comments in up to 140 characters. With the rapid growth of mobile devices, microblog has created mobile applications to provide their users instant and real-time access from anywhere they can access to the Internet. For example, as of September 2017, the number of monthly active users in Sina Weibo platform is more than 376 million, in which about 92% users are authenticated through mobile phone and/or tablet. A large amount of valuable content exists in the microblog generated data. However, as a result of the rapid increasing population on microblog platform, most users are confronted with the serious problem of information overload [1]. It is extremely difficult to find desirable information using mobile devices. In this situation, recommending relevant users for alleviating the flooding of information appears to be very significant for the users [2].

User influence can provide valuable clue about her preference and thus is indispensable for recommending microblog users in mobile social network [3]. Consequently, incorporating user influence into recommender systems has demonstrated to improve recommendation performance and receives a lot of attention. Li et al. [2] considered social influences and their indirect structural relationships and proposed a Topic-level Social Influence-based microblog recommendation model to make user prediction. Jiang et al. [4] proved that users' decisions on information adoption can be affected by individual preference and interpersonal influence and then integrated these two factors to construct a scalable algorithm for online behavior prediction. Chen et al. [5] took advantage of tweet content, user social relations, and explicit features and then proposed a collaborative ranking model for tweet recommendation task. Yan et al. [6] presented a graph-theoretic method to rank tweets and their authors simultaneously by utilizing several networks, i.e., user network, tweet network, and the network that ties the two together. Therefore, it is significant to analyze user

influence in mobile social network and integrate it into the recommendation framework.

There exist several pioneer studies on user influence analysis in mobile microblog platform. Velissarios et al. [7] proposed four different metrics for emphasizing Twitter content features and the behavior of each user's followers and then identified influential users through the comprehensive metrics considering user's affiliation and her interest rate. Mao et al. [8] introduced a learning-based method for analyzing and measuring users' social influence via predicting users' capability of propagating information. Both information extracted from social network structures and user behavior factors were combined in the method to gain a better performance. Xia et al. [9] explained the propagation mechanism of influence in terms of the diffusion of users' emotion. David et al. [10] analyzed the probabilities of one user activated by another user. Then, they combine that user's other features to obtain influence score. Cai et al. [11] proposed an OOLAM model to measure user opinion influence, they separated users interaction graph into two parts, positive graph and negative graph. They ranked users with a PageRank analogous algorithm. These methods reviewed above explored user influence from the perspective of users, which had low accuracy in specific topics.

Recently, various studies are involved in investigating topic level user influence. Those studies showed that most information was created and diffused in terms of topic. User influence can be measured more elaborately from the point of the topic. Therefore, topic level user influence analysis has received increasing attention from researchers. Weng et al. [12] proposed the TwitterRank to calculate user influence score according to the graph structure and topic similarity. Cui et al. [13] introduced item level influence using probabilistic hybrid factor matrix factorization. Chen et al. [14] proposed the MIRC algorithm which can distinguish users in different groups. Their experimental results showed that different influence roles may have stronger influence in their own role level. Wang et al. [15] calculated user influence with four features, i.e., Expert, Leader, Social, and Similar, and then applied user influence to group recommendations. Wei et al. [16] took users' opinion and topic relevance into consideration, and then predicted user influence according to the latent factors resulting from the tensor factorization.

However, most studies for user influence analysis on topic level only consider users' explicit features which can be obtained from users' profile directly [14, 15]. In particular, these existing works neglect the temporal characteristic which can be obtained from the interactions [17]. In addition, the tensor factorization algorithm of user influence analysis tends to give low propagation ability users a high ranking score, since it reduces dimensionality by retaining the critical factors. In this paper, a Tensor Factorization based on User Cluster (TFUC) model [18] is proposed for recommending users according to a specific topic. The TFUC model firstly clusters influential users into a certain groups according to their temporal characteristic. Then, we measure users' influence scores by the temporal restrained CP decomposition on the influential clusters. Finally, both user's influence and content similarity are integrated for recommending users

for a given topic. The experimental results in Sina Weibo dataset show that user influence ranking precision of TFUC is better than existing models such as TwitterRank, OOLAM, and HF_CP_ALS. Moreover, our proposed TFUC model can significantly outperform the baseline methods according to recommendation precision.

This is an extension of our previous work [18], in which we proposed a tensor factorization based user influence analysis method. In this paper, we addressed the problem of recommending users in mobile social network and claimed that user influence is a very important factor for user recommendation. We expanded the experiment dataset and added the experiments on recommendation. To summarize, the main contributes of the work are listed as follows.

(1) The latent influence users are identified by a neural network clustering model. This model can filter the marketing users with low influence before constructing tensor, which is proven to significantly enhance the recommendation effect.

(2) TFUC model is proposed by incorporating temporal features, which can improve user recommendation accuracy. Particularly, our method integrates temporal features by tensor model, predicts user influence using temporal restrained CP decomposition, and finally recommends users considering both user influence and content similarity.

(3) We conduct extensive experiments using real-world Tencent Weibo dataset to verify the effectiveness of our proposed recommendation approach. The experimental results suggest that the proposed method can considerably improve recommendation precision and outperform the baseline approaches.

The rest of the paper is organized as follows. The recommendation problem is defined in Section 2. The proposed model for user recommendation is presented in Section 3. Experiments are conducted in Section 4. We conclude the work in Section 5.

2. Problem Formulation

It is well known that people tend to trust a user with high social influence in social network. Therefore, we apply users influence analysis to recommend items for users according to different topic in social media. In this paper, the items refer to users.

We denote items in the microblog as $U_p = \{u_{p_1}, u_{p_2}, u_{p_3}, \dots, u_{p_n}\}$, where n is the number of items. In the meantime, users who have ever interacted with these items are denoted as $U_c = \{u_{c_1}, u_{c_2}, u_{c_3}, \dots, u_{c_m}\}$. There are two core elements for the recommendation system: characteristic model of items and characteristic model of users. We characterize every items in U_p as $KW(u_p) = [kw_{p_1} : w_{p_1}, kw_{p_2} : w_{p_2}, \dots, kw_{p_n} : w_{p_n}]$ and characterize every users in U_c as $KW(u_c) = [kw_{c_1} : w_{c_1}, kw_{c_2} : w_{c_2}, \dots, kw_{c_n} : w_{c_n}]$.

The goal of this work is to calculate the similarity among items and users and recommend the most similar items for users. In particular, we apply the influence scores of items while calculating the similarity to verify whether the higher influence score the user has, the more likely has

recommendation is to be accepted. To obtain these influence scores, we introduce some necessary features of users, such as the number of fans and the number of posts. Therefore, we let $F = \{f_1, f_2, \dots, f_m\}$ represent users' fan characteristic and $P = \{P_1, P_2, \dots, P_m\}$ represent users' post characteristic. Every interaction between U_p and U_c contains the time of when it takes place, so we present this data as $INTERACTION = \{(int_1, t_1), (int_2, t_2), \dots, (int_q, t_q)\}$.

Referring to Varun's theory [19], we can hardly get user's influence score from single aspect. Thus, we analyze users' influence from four aspects as follows.

(1) Users' propagation ability: getting propagation ability of users is an important purpose in social networks. This ability is usually calculated from the accumulation of time in document collection D . We denote this ability of u_{pi} as $I_{ca}(u_{pi}, U_c, D)$.

(2) Users' opinion strength [20]: one's opinion strength captures his whole tendency and effectiveness in social network. By calculating all of the users' opinion polar who has interacted with u_{pi} , we can get an opinion score of u_{pi} . We present this score as $I_o(u_{pi}, U_c, D)$ which can be analyzed from the document collection D .

(3) Users' fans activity [21]: users with higher levels of activity may contribute more influence to other users in microblog social network. In our work, we regard the number of articles which are posted by u_{cj} as his activity. We can obtain u_{pi} 's global fans activity by accumulating all u_{cj} 's activity who has ever interacted with any of u_{pi} . Formally, we define $I_l(u_{pi}, U_c, P_c)$ as u_{pi} 's global fans activity, where P_c is the collection of users' post features.

(4) Users' network centrality: according to [9, 22], users who have higher influence may have more number of fans. If a user's fans have more fans, it means that the information posted by this user may spread wider. This spreading effect is known as the network centrality and denote as $I_v(u_{pi}, U_c, F_c)$.

Overall, we formalize user influence analysis as follows: given a topic a , the goal is to find a mapping $Inf_a(U_p, U_c, F_c, P_c, D) \rightarrow (I_o, I_l, I_v)$. Users influence scores are calculated by aggregating four users' features I_{ca}, I_o, I_l, I_v . After calculating all basic users' influence scores, we can obtain a user influence ranking list sorted by influence scores.

3. Recommendation with User Influence Analysis

In this section, we propose a user influence analysis model [18] and then integrate it into recommendation. A user with high influence can receive a large number of comments in a short time. A user prefers accepting influential users (referred to items) when he is receiving recommended users by recommendation system. Therefore, the performance of recommendation system will be improved by involving the influence of items. Since the factorization based method performs poorly at low ranking users, we design a two-steps method for influence analysis. In the first step, low influence score clusters are identified by a neural network clustering method. In the first step, user influence is predicted by a tensor factorization method.

3.1. Neural Network Clustering Model. Users' global influence consists of multiple individual influence features, i.e., propagation ability, opinion strength, fans activity, and network centrality. The users with higher influence rank would have more comment and stronger opinion strength and are more central in the network. On this basis, we first partition data into clusters and filter users with low influence in U_p . We firstly describe how we obtain those four users' features.

(1) Let J denote the number of users who has interacted with u_{pi} . Within a time window t , we can get the delay between the time of u_{pi} 's first interaction happened in D and the time of u_{cj} interacted with u_{pi} according to [23] as follows: assume that the delay $(t_{u_{cj}} - t_{u_{pi}})$ has the exponential distribution form like $f(t_{u_{cj}} | t_{u_{pi}}) = \beta_{u_{pi}u_{cj}} \exp(-\beta_{u_{pi}u_{cj}}(t_{u_{cj}} - t_{u_{pi}}))$, where β_{uv} is the transmission rate parameter. The transmission rate parameter captures the capability that how wide a user can reach in the network and thus the computing process is

$$\beta_{u_{pi}u_{cj}}(D) = \frac{\rho_{u_{pi}u_{cj}}(D)}{\Delta_{u_{pi}u_{cj}}(D)} \quad (1)$$

$$\rho_{u_{pi}u_{cj}}(D) = \sum_{di \in D} \delta(pu = u_{pi}) \delta(cu = u_{cj}) \quad (2)$$

$$\begin{aligned} \Delta_{u_{pi}u_{cj}}(D) \\ = \sum_{di \in D} \delta(pu = u_{pi}) \delta(cu = u_{cj}) (t_{u_{cj}} - t_{u_{pi}}) \end{aligned} \quad (3)$$

where pu are the basic users and cu are the users who have interacted with pu , the indicator function $\delta(a = b)$ is 1 if $a = b$ is true and 0 otherwise. Equation (2) result in the fact that the total number of times u_{cj} has interacted with u_{pi} and (3) captures a time accumulation of those interactions. After calculating $\beta_{u_{pi}u_{cj}}$, we can infer time accumulation of pu by following aggregate function:

$$I_{ca}(u_{pi}) = \sum_{j=1}^J 1 - \exp(-\beta_{u_{pi}u_{cj}}(D)t) \quad (4)$$

(2) Each user would show an opinion polar to an interbehavior which can be inferred from the interaction between him and basic users. Therefore, we can get u_{pi} 's global opinion strength by accumulating all opinion polar of his interactions. We utilize (5) to obtain the opinion strength of u_{pi} :

$$I_o(u_{pi}) = \sum_{j=1}^J O(u_{cj}) \quad (5)$$

the indicator function $O(u_{cj})$ is -1 if u_{cj} has ever expressed a negative interbehavior and 1 if u_{cj} has ever expressed a nonnegative interbehavior.

(3) As we defined previously, u_{pi} 's fans activity is related to the total number of articles that are posted by all of his fans.

Based on this definition, we can obtain u_{pi} 's fans activity as follows:

$$I_i(u_{pi}) = \sum_{j=1}^J p_{u_{cj}} \quad (6)$$

(4) Recall that the number of fans of user u_{cj} is available directly from F_c ; we calculate user u_{pi} 's network centrality as follows:

$$I_c(u_{pi}) = \sum_{j=1}^J f_{u_{cj}} \quad (7)$$

We now discuss how to partition users in U_p into clusters according to those four influence features. The input samples of our method are U_p . Each sample u_{pi} involves four features which we obtain previously. We denote each sample as $\mathbf{Y} = [y_{i1}, y_{i2}, y_{i3}, y_{i4}]$, where y_{i1} - y_{i4} is $I_{ca}(u_{pi})$, $I_o(u_{pi})$, $I_l(u_{pi})$, $I_c(u_{pi})$, respectively. Let C_n denote multiple clustering centers. Each center C_i has four elements, i.e., $[c_{i1}, c_{i2}, c_{i3}, c_{i4}]$. For the clustering problem, the loss function is

$$L(\mathbf{X}; I_{ca}, I_o, I_l, I_c) = \frac{1}{2} \sum_{n,i,j} (w_{ij} y_{ni} - c_{ki})^2 \quad (8)$$

where C_k is the clustering center of X and w_{ij} is the weight of between input and interlayer.

We update each w_{ij} using stochastic gradient descent.

$$w_{ij}^{(t+1)} = w_{ij}^{(t)} - \eta \frac{\partial L}{\partial w_{ij}^{(t)}} \quad (9)$$

where

$$\frac{\partial L}{\partial w_{ij}^{(t)}} = \frac{\partial \left((w_{ij}^{(t)} y_i - c_{ki})^2 \right)}{\partial w_{ij}^{(t)}} = 2w_{ij}^{(t)} y_i^2 - 2y_i c_{ki} \quad (10)$$

Bringing (10) into (9), we have

$$w_{ij}^{(t+1)} = w_{ij}^{(t)} - \eta (w_{ij}^{(t)} y_i^2 - y_i c_{ki}) \quad (11)$$

We update clustering centers for each batch as

$$c_{ki} = \frac{\sum_{i,j} w_{ij} \delta_{C_k}(y_i)}{\text{count}_{C_k}(Y)} \quad (12)$$

where $\delta_{C_k}(x_i)$ is an indicator function for clustering center C_k and the result is 1 if y_i belongs to the cluster C_k and 0 otherwise. The denominator in (12) is a counting function which returns the number of samples in cluster C_k .

3.2. Construction of Tensor User Influence Model. We assign each cluster to a specific influence category. Specifically, the assignment with most of latent influential users in U_p is selected to construct the tensor model. Users in this cluster are denoted as U_p' , where $U_p' \subseteq U_p$. Our users influence model is represented by a 3-order tensor $\mathbf{X} \in R^{I \times J \times K}$, where I is the number of users in U_p' , J is the number of

comment users in U_c , and K is the number of influence features. Tensor decomposition is generally used to predict the distribution of data and the latent features of data. Tensor is used widely in many research area, such as weather forecast, event prediction [24], information recommendation [25], and picture processing [26–28]. Finally, we take these influence features into each tensor slice.

(1) The opinion slice of users: this slice indicates every users' interaction opinion in U_c on U_p' in detail; i.e.,

$$X_{ij1} = O(u_{cj}) \delta(cu = u_{cj}) \delta(pu = u_{pi}) \quad (13)$$

where $O(u_{cj})$ is an indicator function as same as the function in (5), $u_{pi} \in U_p'$.

(2) The fans activity slice of users: in this slice, users who have ever interacted with u_{pi} would have a activity influence upon u_{pi} . Thus every element in this slice can be represent as

$$X_{ij2} = p_{u_{cj}} \delta(cu = u_{cj}) \delta(pu = u_{pi}) \quad (14)$$

(3) The centrality slice of users: as mentioned in Section 3, we present users' network centrality by his diffusion ability which can be presented by his total number of neighbours; i.e.,

$$X_{ij3} = f_{u_{cj}} \delta(cu = u_{cj}) \delta(pu = u_{pi}) \quad (15)$$

3.3. Factorization of Tensor User Influence Model. For the tensor $\mathbf{X} \in R^{I \times J \times K}$, the loss function of rank- R CP decomposition [29, 30] is

$$L(\mathbf{X}; \mathbf{A}, \mathbf{B}, \mathbf{C}) = \sum_{ijk} \left(X_{ijk} - \sum_{r=1}^R \mathbf{A}_{ir} \mathbf{B}_{jr} \mathbf{C}_{kr} \right)^2 \quad (16)$$

The corresponding objective function for stochastic optimization problem is

$$\min_{\lambda, \mathbf{A}, \mathbf{B}, \mathbf{C}} L(\mathbf{X}; \mathbf{A}, \mathbf{B}, \mathbf{C}) \quad (17)$$

However, temporal influence feature neglects this problem. Thus, a time constraint is added to the user matrix. So the influence score of users whose propagation ability is strong will increase and the score of users who postfrequently receive few comments. The new loss function is written as

$$L_\rho = \frac{1}{2} \sum_{ijk} \left(X_{ijk} - \sum_{r=1}^R \mathbf{A}_{ir} \mathbf{B}_{jr} \mathbf{C}_{kr} \right)^2 + \frac{1}{2} \rho \sum_{i=1}^I \sum_{r=1}^R \mathbf{Q}_{ii} |\mathbf{A}_{ir}|^2 + \frac{1}{2} \rho (\|\mathbf{B}\|^2 + \|\mathbf{C}\|^2) \quad (18)$$

where \mathbf{Q} is the time constraint matrix which can be obtained from (4). \mathbf{Q} is diagonal and the main diagonal element is

$$\mathbf{Q}_{ii} = I_{ca}(u_{pi}) \quad (19)$$

where u_{pi} is the users in U_p' .

The object function is

$$\min_{\lambda, A, B, C} L_\rho(\mathbf{X}; \mathbf{A}, \mathbf{B}, \mathbf{C}) \quad (20)$$

Following the method proposed in [30], the gradient of (18) is

$$\frac{\partial L_\rho}{\partial \mathbf{A}}(\mathbf{X}; \mathbf{A}, \mathbf{B}, \mathbf{C}) = -\mathbf{Y}(\cdot, \mathbf{B}, \mathbf{C}) + \mathbf{A}\mathbf{T}(\mathbf{B}, \mathbf{C}) + \rho\mathbf{Q}\mathbf{A} \quad (21)$$

According to the theory proposed by Acar et al. [31], we can get that

$$\mathbf{T}(\mathbf{B}, \mathbf{C}) = \mathbf{B}^T \mathbf{B} \mathbf{C}^T \mathbf{C} \quad (22)$$

$$\mathbf{Y}(\cdot, \mathbf{B}, \mathbf{C}) = \mathbf{X}_{(1)} \mathbf{Z}_1 \quad (23)$$

where $\mathbf{X}_{(1)}$ is the model-1 unfolding, $\mathbf{Z}_1 = \mathbf{C} \odot \mathbf{B}$, and \odot is the Khatri-Rao product between \mathbf{C} and \mathbf{B} . In the same way, we can get $\mathbf{T}(\mathbf{A}, \mathbf{C})$, $\mathbf{T}(\mathbf{A}, \mathbf{B})$, $\mathbf{Y}(\mathbf{A}, \cdot, \mathbf{C})$, and $\mathbf{Y}(\mathbf{A}, \mathbf{B}, \cdot)$.

We can obtain a rule for updating \mathbf{A} by substituting (21) into stochastic gradient descent method as follows:

$$\begin{aligned} \mathbf{A}^{(t+1)} &= \mathbf{A}^{(t)} - \eta^{(t)} \frac{\partial L_\rho}{\partial \mathbf{A}} = \mathbf{A}^{(t)} \\ &- \eta^{(t)} \left[-\mathbf{Y}^{(t)}(\cdot, \mathbf{B}^{(t)}, \mathbf{C}^{(t)}) + \mathbf{A}^{(t)} \mathbf{T}^{(t)}(\mathbf{B}^{(t)}, \mathbf{C}^{(t)}) \right. \\ &+ \left. \rho \mathbf{Q} \mathbf{A}^{(t)} \right] = \mathbf{A}^{(t)} \left[\mathbf{I} - \eta^{(t)} \mathbf{T}^{(t)}(\mathbf{B}^{(t)}, \mathbf{C}^{(t)}) \right] \\ &+ \eta^{(t)} \mathbf{Y}^{(t)}(\cdot, \mathbf{B}^{(t)}, \mathbf{C}^{(t)}) - \eta^{(t)} \rho \mathbf{Q} \mathbf{A}^{(t)} \end{aligned} \quad (24)$$

where η is the step size. The updating rules of \mathbf{B} are similar to \mathbf{C} . We just give the updating rule of \mathbf{B} due to the space limitation.

$$\begin{aligned} \mathbf{B}^{(t+1)} &= \mathbf{B}^{(t)} - \eta^{(t)} \frac{\partial L_\rho}{\partial \mathbf{B}} \\ &= \mathbf{B}^{(t)} \\ &- \eta^{(t)} \left[-\mathbf{Y}^{(t)}(\mathbf{A}^{(t)}, \cdot, \mathbf{C}^{(t)}) + \mathbf{B}^{(t)} \mathbf{T}_\rho^{(t)}(\mathbf{A}^{(t)}, \mathbf{C}^{(t)}) \right] \\ &= \mathbf{B}^{(t)} \left[\mathbf{I} - \eta^{(t)} \mathbf{T}_\rho^{(t)}(\mathbf{A}^{(t)}, \mathbf{C}^{(t)}) \right] \\ &+ \eta^{(t)} \mathbf{Y}^{(t)}(\mathbf{A}^{(t)}, \cdot, \mathbf{C}^{(t)}) \end{aligned} \quad (25)$$

3.4. Measurement of Users Influence. We now discuss how to calculate users' influence score by utilizing the result of the tensor decomposition. Users' influence can be calculated from three different influence scores.

(1) Score of users' opinion strength:

$$I_o(u_{pi}) = \sum_{j=1}^r \widehat{\mathbf{X}}_{ij1} \quad (26)$$

(2) Score of users' fans activity:

$$I_l(u_{pi}) = \sum_{j=1}^r \widehat{\mathbf{X}}_{ij2} \quad (27)$$

(3) Score of network centrality:

$$I_v(u_{pi}) = \sum_{j=1}^r \widehat{\mathbf{X}}_{ij3} \quad (28)$$

where $\widehat{\mathbf{X}}$ is the expectation of \mathbf{X} . We unify each influence score using min-max normalized method, respectively. And then, we use final influence score by combining these three normalized scores as follows:

$$I(u_{pi}) = S_i \times (I_o(u_{pi}) + I_l(u_{pi}) + I_v(u_{pi})) \quad (29)$$

We add a user topic similarity metric to the combining function to increase users' influence score whose topic similarity is higher. This topic similarity metric will be explained in later sections.

3.5. Recommendation Model with User Social Influence.

In this section, we recommend items for users by using content-based recommendation algorithm in Tencent Weibo dataset. The items in this dataset are the person, organization, or group in the real world. Initially, we obtain the preferences and interests of the users whom should receive the recommendation. Users' preferences and interests are analyzed from the articles and comments of them. After that, we establish the users characteristic model based on these preferences and interests. The other essential processes are establishing the items characteristic model. In order to adapt to the dataset, we use the preferences and interests of the items characteristic. Based on these two models, we calculate the similarity between the users and the items. Furthermore, we combine the ranking indicators of items into the similarity and call it influence-similarity. Finally, we recommend items for users according to the influence-similarity.

There are two core parts for the above content-based recommendation process: users characteristic model and items characteristic model. Since the users and the items are all the individual users in Tencent Weibo dataset, we represent every user u to a characteristic vector by using the TF-IDF method. Formally, we denote user vector as $KW(u) = [kw_1 : w_1, kw_2 : w_2, \dots, kw_n : w_n]$, where kw_i is a word that is extracted from the articles and comments which user u has ever posted, and w_i is the corresponding weight of the word kw_i in the text collection. The weight is calculated by TF-IDF method as follows:

$$TF_{i,j} = \frac{n_{kw_i}}{\sum_k n_{kw_k}} \quad (30)$$

$$IDF = \log \frac{N}{n_i} \quad (31)$$

$$w_i = TF_{i,j} \times IDF \quad (32)$$

where n_{kw_i} is the number of times word kw_i appearing in text j , $\sum_k n_{kw_k}$ is the total number of words that text j contains, N is the total number of texts in the dataset, and n_i is the number of texts that contain word kw_i .

TABLE 1: Statistics for Sina Weibo dataset.

Topic	Seed users	Articles	Comment Users	Comments
Law	508	6992	8383	14693
Basketball	492	9463	66449	114374
Economy	509	12781	29245	51742
Health	506	9438	30884	54269

TABLE 2: Statistics Tencent Weibo dataset.

Topic	Item	Received Users	Comments
1.6.2.1	277	23627	44047
1.1.2.1	167	13065	26232
1.2.2.1	147	46550	162939
1.12.4.5	134	15253	27310

The next step is calculate the cosine similarity between the users and items. For example, when calculating the similarity between user u_{c_j} and item u_{r_i} , we have

$$S(u_{c_j}, u_{r_i}) = \frac{\sum_{i=1}^M (W_{i,u_{c_j}} \times W_{i,u_{r_i}})}{\sqrt{(\sum_{i=1}^M W_{i,u_{c_j}}^2) (\sum_{i=1}^M W_{i,u_{r_i}}^2)}} \quad (33)$$

However, when (33) was adopted to calculate the similarity between user u_{c_j} and item u_{r_i} , it does not take the influence of item u_{r_i} into consideration. Therefore, we add the influence ranking indicator of item u_{r_i} into the original cosine similarity. Thus, the item of higher influence score could have higher probability of being recommended. The cosine similarity with the influence of items is calculate as follows:

$$S(u_{c_j}, u_{r_i})' = List[I(u_{r_i})] \times \frac{\sum_{i=1}^M (W_{i,u_{c_j}} \times W_{i,u_{r_i}})}{\sqrt{(\sum_{i=1}^M W_{i,u_{c_j}}^2) (\sum_{i=1}^M W_{i,u_{r_i}}^2)}} \quad (34)$$

where $List[I(u_{r_i})]$ is the influence ranking indicator of item u_{r_i} in its topic areas.

4. Experiments

4.1. Datasets. In this section, extensive experiments are conducted on two popular microblogs' platform in China, i.e., Sina Weibo and Tencent Weibo.

For Sina Weibo dataset, we first crawled 2015 basic users in different topics, including law, basketball, economy, and health. We crawled these users' information and all articles posted by these users from October 31, 2016, to December 1, 2016. The basic statistics of Sina Weibo dataset are showed in Table 1. We annotate users' influence ranking manually according to [16]. In this dataset, the interaction between two users is present as comment. If u_j commented in u_i 's articles, there generate an interaction between them. There

exists a delay between the time u_i post article and the time u_j commented on this article. Therefore, we can obtain the temporal characteristic based on this delay. Besides, the topic similarity metric in (29) is obtained the same as [16].

For Tencent Weibo dataset, we obtain it from KDD Cup 2012, Track 1. This dataset contains about 6095 high influence users in different topics. These users are called "Item" in this dataset. There are about 73,209,277 recommendation logs in this dataset. The recommendation is send to every user corresponding to a user profile such as his gender, the number of articles he posted, and keywords extracted from all his articles. We can infer user's number of fans from the relational network in this dataset. In this dataset, the interaction between two users is present as recommendation. This interaction contains two significant information, i.e., acceptability and time stamp. Thus, we can obtain u_i 's opinion strength and temporal characteristic according to (5) and (3), respectively. For experiment convenience, we choose high influence users in four topics and the time window is from October 12, 2011, to October 13, 2011. Table 2 shows the basic statistics of this dataset. In this dataset, the topic similarity metric in (29) is obtained according to the Jaccard similarity of their characteristic set.

4.2. Baseline. We compared TFUC with the following baselines:

- (i) TwitterRank [12], which calculated user influence according to the users' interactions in a certain topic.
- (ii) TwitterRank_C, in which we apply TwitterRank to calculated user influence based on the latent influential users cluster which obtained by our cluster model.
- (iii) OOLAM [11], which is a PageRank analogous method in which interactions are divided into positive and negative parts so that users' opinion influence is calculated in positive and negative parts, respectively.
- (iv) OOLAM_c, in which we use latent influential users to construct positive and negative graph, respectively.
- (v) OOLAM_SM, in which the users' topic similarity is taken into consideration in OOLAM.

TABLE 3: Ranking precision for various methods in four topics.

Method	Law			Method	Basketball		
	$P@5$	$P@10$	$P@20$		$P@5$	$P@10$	$P@20$
TwitterRank	0.20	0.10	0.10	TwitterRank	0.00	0.00	0.00
TwitterRank_C	0.40	0.30	0.40	TwitterRank_C	0.00	0.00	0.15
OOLAM	0.40	0.40	0.60	OOLAM	0.40	0.40	0.30
OOLAM_C	0.20	0.50	0.65	OOLAM_C	0.40	0.50	0.55
OOLAM_SM	0.40	0.50	0.60	OOLAM_SM	0.40	0.50	0.40
OOLAM_SM_C	0.20	0.50	0.65	OOLAM_SM_C	0.40	0.50	0.55
HF_CP_ALS	0.40	0.40	0.65	HF_CP_ALS	0.40	0.30	0.50
HF_CP_ALS_C	0.00	0.30	0.50	HF_CP_ALS_C	0.60	0.30	0.60
CPSGD	0.40	0.50	0.55	CPSGD	0.40	0.40	0.60
Our method	0.40	0.50	0.60	Our method	0.60	0.50	0.70

Method	Economy			Method	Health		
	$P@5$	$P@10$	$P@20$		$P@5$	$P@10$	$P@20$
TwitterRank	0.00	0.00	0.05	TwitterRank	0.00	0.00	0.00
TwitterRank_C	0.00	0.10	0.20	TwitterRank_C	0.00	0.00	0.05
OOLAM	0.20	0.30	0.40	OOLAM	0.40	0.40	0.40
OOLAM_C	0.20	0.30	0.40	OOLAM_C	0.40	0.40	0.55
OOLAM_SM	0.20	0.40	0.50	OOLAM_SM	0.40	0.40	0.40
OOLAM_SM_C	0.20	0.40	0.50	OOLAM_SM_C	0.40	0.30	0.55
HF_CP_ALS	0.20	0.30	0.30	HF_CP_ALS	0.00	0.20	0.40
HF_CP_ALS_C	0.20	0.30	0.30	HF_CP_ALS_C	0.00	0.20	0.40
CPSGD	0.40	0.60	0.50	CPSGD	0.40	0.40	0.45
Our method	0.40	0.60	0.50	Our method	0.40	0.40	0.65

(vi) OOLAM_SM_C, in which the cluster model is added in OOLAM_SM.

(vii) HF_CP_ALS [16], which is a tensor model, in which users' opinion and topic relevance are taken into consideration.

(viii) HF_CP_ALS_C, in which the cluster model is added in HF_CP_ALS.

(ix) CP_SGD, in which low influence users are not filtered when we construct the user's tensor.

Besides, we need to verify whether the performance of recommendation system with influence has a better performance than the recommendation system without; we choose a simple recommendation algorithm, i.e., content-base(BC) algorithm to be the baseline.

4.3. *Evaluations.* The evaluations include 3 ranking precision evaluations and 2 recommended precision evaluations.

$$P@k = \frac{|A_k \cap B_k|}{k} \quad (35)$$

where A_k is the set of real top- k users and B_k is the predicted set of top- k users.

$$AP = \frac{\sum_{i=1}^n |A_i \cap B_i| / i}{n} \quad (36)$$

where i denotes i -th rank and n denotes the number of users.

$$MAP = \frac{\sum_a AP^a}{ca} \quad (37)$$

where a is a certain topic and ca denotes the number of topics.

The two recommended precision evaluations are as follows:

$$AP_r = \frac{1}{\sum_{i=1}^n r_i} \sum_{i=1}^n \frac{r_i \sum_{j=1}^i r_j}{i} \quad (38)$$

$$MAP_r = \frac{\sum_a AP_r^a}{ca} \quad (39)$$

where a is a certain topic, ca is the number of topics, and n represents the number of users that need to be recommended. r_i is an accepted index, the value of it is 1 if the i -th user was once accepted when he was recommended to other users and 0 otherwise.

Equation (38) is the average precision in a single topic; it reflects the performance of the model in a single topic. Equation (39) reflects the overall performance of the model in all topics. The higher the MAP_r is, the users of higher influence score are more likely to be accept by other users.

4.4. *Precision Results of User Influence Ranking.* The $P@k$ of different methods in Sina Weibo dataset is shown in Table 3. The $P@k$ of our method is optimal except for the $P@20$ in law topic. To analyze the experimental results in more detail,

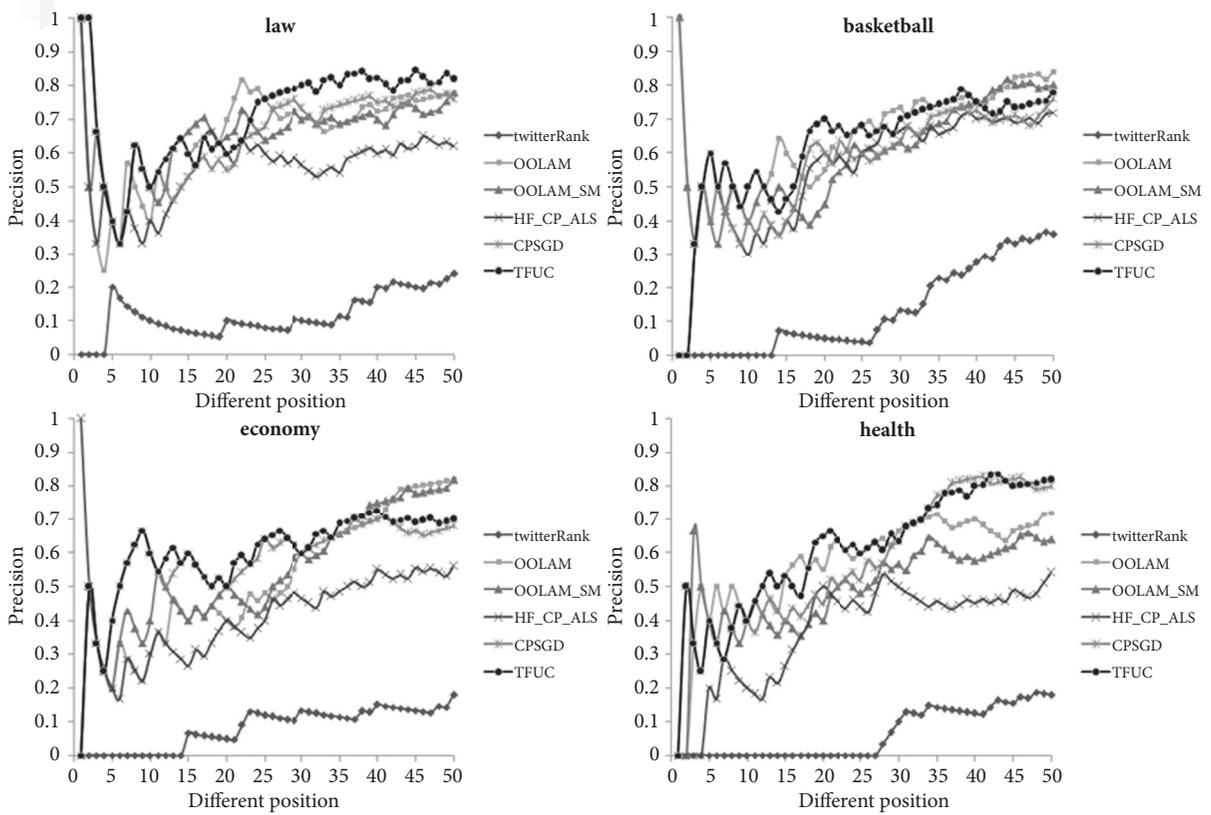


FIGURE 1: Precision in different position for various methods.

we compare our method with each baseline separately. It can be seen from Table 3 that our proposed method outperforms TwitterRank, which verifies that a user with strong opinion strength, many activity fans, and high propagation ability would be influential. The precision of our method is at least 10% higher than that of OOLAM, which demonstrates that a user with high propagation ability and high topic similarity would have a high influence score. The temporal features are neglected in OOLAM_SM, so that it performs worse than our method. HF_CP_ALS also did not take temporal features into consideration, so that the user with high propagation ability would not get a high influence score. Comparing to CPSGD, the precision of our method has improved at least 10%, which means that filtering some low influential users can improve the performance.

Furthermore, we also calculated the *AP* and *MAP* for each method. Figure 1 shows the precision in different i . The *AP* is higher when the area under the curve is larger. The detail *AP* and *MAP* of each method can be seen in Table 4. *AP* of our method is better than other baselines except for the *AP* of OOLAM in basketball topic and the *MAP* of our method is best among all methods. We can conclude that our method performs better than other baselines.

4.5. User Influence in Recommendation. In the previous section, we proved that TFUC outperforms other baselines. In this section, we apply TFUC in retrieving users' influence

scores in Tencent Weibo dataset. After that, we rank users according to these scores. For each basic users in this dataset, we obtain a recommendation result by counting all u_{pi} 's recommendation logs. If his recommendation was once accepted by other users successfully, his recommendation result could be present as 1 and 0 otherwise. By calculating the correlation coefficient between the influence ranking list and the result list, we could tell which influence analysis method we used in recommendation is closer to the practical situation.

In recommendation task, we also first recognize latent influential users by TFUC model. Now we discuss how to obtain user features to generate clustering model.

(1) As mentioned in datasets description, the interaction between two users is presented as recommendation in Tencent Weibo dataset and this interaction contains time information. Therefore, we can obtain basic users' propagation ability from this time information by (3).

(2) Due to lack of direct opinion information in Tencent Weibo dataset, we present the acceptance of the recommendations as the opinion of a user to the basic users. In this case, we can obtain basic users' opinion strength according to (5).

(3) The users' fans activity and network centrality are obtained similar to (6) and (7).

After getting these four users' features, TFUC model partitions users into different clusters and constructs tensor model based on the users whom in the latent influential

TABLE 4: AP and MAP comparisons between various methods.

Method	AP				MAP
	Law	Basketball	Economy	Health	
TwitterRank	0.1160	0.1303	0.0818	0.0633	0.0979
TwitterRank_C	0.5139	0.2013	0.1591	0.1267	0.2503
OOLAM	0.6602	0.6570	0.5211	0.5730	0.6028
OOLAM_C	0.6607	0.6609	0.5257	0.5572	0.6011
OOLAM_SM	0.6578	0.6027	0.5443	0.4960	0.5752
OOLAM_SM_C	0.6328	0.6418	0.4921	0.5490	0.5790
HF_CP_ALS	0.5577	0.5537	0.4254	0.3695	0.4766
HF_CP_ALS_C	0.4916	0.5271	0.3496	0.3838	0.4380
CPSGD	0.6668	0.5640	0.5828	0.5781	0.5979
Our method	0.7159	0.6212	0.5989	0.6110	0.6368

TABLE 5: The mean precision of the recommendations.

Method	OOLAM	OOLAM_SM	HF_CP_ALS	CPSGD	Our method
MAP_r	0.7028	0.6905	0.6734	0.6964	0.7104

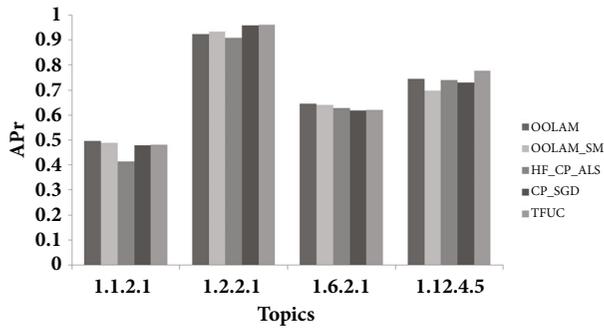


FIGURE 2: The average precision of the recommendations.

user cluster. After decomposing the tensor model, all users' influence score can be predicted according to (29).

To calculate the precision of the recommendations, we need to obtain the successful accepted list. In Tencent Weibo dataset, if a u_j accepted the recommendation of whose entity is u_i , the system will record this interaction into recommendation logs. By calculating which user is successfully accepted by other users in logs file, we can get an acceptance list.

Finally, we can get a recommendations precision by compare the influence list and the acceptance list according to (39). Figure 2 and Table 5 show the results.

It can be seen from Figure 2 and Table 5 that the recommend results which combine with the users influence ranking list obtained from our method have the similar performance to the OOLAM method in each topic. However, our method shows a better overall performance in four topics. This result illustrates that when the influence scores which add temporal characteristic and topic similarity were applied in the recommendation system, the items with a higher influence are more likely to be accept. The AP_r value of our method has promoted 2% to 8% than OOLAM_SM. The result above reflects that when the temporal characteristic is considered, the items influence ranking list is better

adapted to the actual recommendation results. Our method has a higher recommendation precision than the method HF_CP_ALS which also confirmed the above conclusion. Compared with CP_SGD methods, the AP_r value of our method has improved in every topic. This is due to the fact that we filter the impact of low influence but high activity marketing items of which the recommendations have a low probability to be accepted.

Based on the analysis above, we can conclude that the high influence items obtained from our method have a wider range of probability to be accepted by users. Therefore, we combine the recommendation system with the item influence obtained from our method. Firstly, we calculate the recommendations result list from rec_log_train dataset in topics "1.6.2.1", "1.1.2.1", "1.2.2.1", and "1.12.4.5". We next calculate the influence-similarity between the users and items in this recommendations result list. Then, we choose the top 100 most similar results as our recommendations for users and calculate the average recommended precision in each topic. Finally, we obtain the MAP_r by fusing the average recommended precision of each topic. The MAP_r of content-based recommendation method is 14.5. The MAP_r improves to 15.5 when TFUC is integrated. This indicates that the performance of the recommendation system can be improved when the influence is considered.

5. Conclusion

This paper focuses a recommendation task in which users' influence analysis is involved in microblogs. We introduce a two-steps method for influence analysis. Firstly, users are partitioned into influential part and uninformal part. And then, we expect CP decomposition with stochastic gradient descent method to expedite decomposition. In addition, a time constraint matrix is also involved in the user factor matrix during the decomposition. Finally, we apply TFUC model to recommend items for users according to

the influence of items. The experimental results show that TFUC outperforms other baselines with 3.4% at least. The extensive experiments in Tencent Weibo dataset show that the precision of the recommendation system is improved when we combine the recommendation system with the influence of items.

There are many potential future directions of this work. First, the temporal characteristic of users or items in this paper is estimate rough. The reason is that we suppose that the delay of the interaction satisfies the exponential distribution. We do not conduct in-depth research of the other temporal accumulation models. Such ambiguity can be further aggravated in the microblog. Additionally, the recommendation algorithm of our work is still primitive with challenges including how to design a more realistic recommendation algorithm and combine it with user influence aspect.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research project was supported by the National Natural Science Foundation of China (no. 61772135 and no. U1605251), the Open Project of Key Laboratory of Network Data Science & Technology of Chinese Academy of Sciences (no. CASNDST201606 and no. CASNDST201708), and the Directors Project Fund of Key Laboratory of Trustworthy Distributed Computing and Service (BUPT) Ministry of Education (no. 2017KF01). The authors thank Lin Gui and Kam-Fai Wong for their cooperation in TFUC model.

References

- [1] M. G. Rodriguez, K. Gummadi, and B. Schoelkopf, "Quantifying information overload in social media and its impact on social contagions," in *Proceedings of the 8th International AAAI Conference on Weblogs and Social Media*, pp. 170–179, Ann Arbor, Michigan, USA, 2015.
- [2] D. Li, Z. Luo, Y. Ding et al., "User-level microblogging recommendation incorporating social influence," *Journal of the Association for Information Science and Technology*, vol. 68, no. 3, pp. 553–568, 2017.
- [3] X. D. Wu, Y. Li, and L. Li, "Influence analysis of online social networks," *Chinese Journal of Computers. Jisuanji Xuebao*, vol. 37, no. 4, pp. 735–752, 2014.
- [4] M. Jiang, P. Cui, F. Wang, W. Zhu, and S. Yang, "Scalable recommendation with social contextual information," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 11, pp. 2789–2802, 2014.
- [5] K. Chen, T. Chen, G. Zheng, O. Jin, E. Yao, and Y. Yu, "Collaborative personalized tweet recommendation," in *Proceedings of the 35th Annual ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 661–670, USA, August 2012.
- [6] R. Yan, M. Lapata, and X. Li, "Tweet recommendation with graph co-ranking," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL '12)*, pp. 516–525, July 2012.
- [7] V. Zamparas, A. Kanavos, and C. Makris, "Real Time Analytics for Measuring User Influence on Twitter," in *Proceedings of the 2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 591–597, Vietri sul Mare, Italy, November 2015.
- [8] J.-X. Mao, Y.-Q. Liu, M. Zhang, and S.-P. Ma, "Social influence analysis for micro-blog user based on user behavior," *Jisuanji Xuebao/Chinese Journal of Computers*, vol. 37, no. 4, pp. 791–800, 2014.
- [9] X. Hu, L. Tang, J. Tang, and H. Liu, "Exploiting social relations for sentiment analysis in microblogging," in *Proceedings of the 6th ACM International Conference on Web Search and Data Mining (WSDM '13)*, pp. 537–546, Rome, Italy, February 2013.
- [10] D. Kempe, J. Kleinberg, and E. Tardos, "Maximizing the spread of influence through a social network," in *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 137–146, Washington, DC, USA, 2003.
- [11] K. Cai, S. Bao, Z. Yang et al., "OOLAM: An opinion oriented link analysis model for influence persona discovery," in *Proceedings of the 4th ACM International Conference on Web Search and Data Mining, WSDM 2011*, pp. 645–654, China, February 2011.
- [12] J. Weng, E. Lim, J. Jiang, and Q. He, "TwitterRank: finding topic-sensitive influential twitterers," in *Proceedings of the 3rd ACM International Conference on Web Search and Data Mining (WSDM '10)*, pp. 261–270, New York, NY, USA, February 2010.
- [13] P. Cui, W. Fei, S. Yang, and L. Sun, "Shiqiang Yang, and Lifeng Sun. Item-level social influence prediction with probabilistic hybrid factor matrix factorization," in *Proceedings of the In Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI, 2011*, San Francisco, California, USA, 2011.
- [14] C. Chen, D. Gao, W. Li, and Y. Hou, "Inferring topic-dependent influence roles of Twitter users," in *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2014*, pp. 1203–1206, Australia, July 2014.
- [15] J. Wang, Z. Liu, and H. Zhao, "Topic oriented user influence analysis in social networks," in *Proceedings of the 2015 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology Workshops, WI-IAT Workshops 2015*, pp. 123–126, Singapore, December 2015.
- [16] J.-J. Wei, C. Chen, X.-W. Liao, G.-L. Chen, and X.-Q. Cheng, "User social influence analysis based on constrained nonnegative tensor factorization," *Tongxin Xuebao/Journal on Communication*, vol. 37, no. 6, pp. 154–162, 2016.
- [17] Y. Zhou and L. Liu, "Social influence based clustering of heterogeneous information networks," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 338–346, ACM, Chicago, Ill, USA, August 2013.
- [18] X. Liao, L. Zhang, L. Gui, K. Wong, and G. Chen, "A Tensor Factorization Based User Influence Analysis Method with Clustering and Temporal Constraint," in *Natural Language Processing and Chinese Computing*, vol. 10619 of *Lecture Notes in Computer Science*, pp. 877–886, Springer International Publishing, Cham, 2018.

- [19] V. R. Embar, I. Bhattacharya, V. Pandit, and R. Vaculin, "Online topic-based social influence analysis for the Wimbledon championships," in *Proceedings of the 21st ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2015*, pp. 1759–1768, Australia, August 2015.
- [20] "Social Media Processing," in *Proceedings of Third National Conference of the Social Media Processing, SMP '14*, H. Huang, T. Liu, H.-P. Zhang, and J. Tang, Eds., vol. 489, Springer, Beijing, China, 2014.
- [21] "Proceedings of 24th Annual International Conference on Computer Science and Software Engineering, CASCON, '14," in *Proceedings of 24th Annual International Conference on Computer Science and Software Engineering, CASCON, 2014*, J. Ng, J. Li, and K. Wong, Eds., IBM / ACM, Markham, Ontario, Canada, 2014.
- [22] D. Xu, Y. Liu, M. Zhang, and S. Ma, "Study on user influence in online social networks," *Journal of Chinese Information Processing*, vol. 30, no. 2, pp. 83–89, 2016.
- [23] M. Gomez-Rodriguez, J. Leskovec, and A. Krause, "Inferring networks of diffusion and influence," in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD-2010*, pp. 1019–1028, USA, July 2010.
- [24] T. M. Do, "Non-linear Time-series Analysis of Social Influence," in *Proceedings of the 2016*, pp. 12–16, San Francisco, California, USA, June 2016.
- [25] P. Symeonidis, A. Nanopoulos, and Y. Manolopoulos, "Tag recommendations based on tensor dimensionality reduction," in *Proceedings of the 2008 2nd ACM International Conference on Recommender Systems, RecSys'08*, pp. 43–50, Switzerland, October 2008.
- [26] C. Wang, X. He, J. Bu, Z. Chen, C. Chen, and Z. Guan, "Image representation using Laplacian regularized nonnegative tensor factorization," *Pattern Recognition*, vol. 44, no. 10-11, pp. 2516–2526, 2011.
- [27] A. Karami, M. Yazdi, and A. Zolghadre Asli, "Hyperspectral image compression based on tucker decomposition and discrete cosine transform," in *Proceedings of the 2nd International Conference on Image Processing Theory, Tools and Applications, IPTA '10*, pp. 122–125, France, July 2010.
- [28] A. Karami, M. Yazdi, and A. Z. Asli, "Best rank-r tensor selection using Genetic Algorithm for better noise reduction and compression of Hyperspectral images," in *Proceedings of the Fifth International Conference on Digital Information Management (ICDIM '10)*, pp. 169–173, Thunder Bay, ON, Canada, July 2010.
- [29] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.
- [30] T. Maehara, K. Hayashi, and K.-I. Kawarabayashi, "Expected tensor decomposition with stochastic gradient descent," in *Proceedings of the 30th AAAI Conference on Artificial Intelligence, AAAI 2016*, pp. 1919–1925, USA, February 2016.
- [31] 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, 2011.

