IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received November 26, 2019, accepted December 11, 2019, date of publication December 18, 2019, date of current version January 6, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2960540

A Multi-Trans Matrix Factorization Model With Improved Time Weight in Temporal Recommender Systems

JIANYUN ZHANG[1] AND XIANLING LU[2,3]
[1]Key Laboratory of Advanced Process Control for Light Industry, Ministry of Education, School of Internet of Things Engineering, Jiangnan University, Wuxi 214122, China
[2]School of Internet of Things Engineering, Jiangnan University, Wuxi 214122, China
[3]Jiangsu Key Construction Laboratory of IoT Application Technology, Wuxi 214000, China

Corresponding author: Xianling Lu (jnluxl@jiangnan.edu.cn)

This work was supported by the 111 Project under Grant B12018.

ABSTRACT In real-world recommender systems, users' interest and products' characteristics tend to go through a distinct series of changes over time. Thus, designing a recommender system that can simultaneously track the temporal dynamics of both drifts becomes a significant task. However, most of the existing temporal recommender systems only focus on users' dynamics, ignoring changes in products' characteristics. In this study, we propose a Multi-Trans matrix factorization (MTMF) model with improved time weight to capture temporal dynamics. Firstly, we introduce a personalized time weight that combines the forgetting curve and item similarity to reduce the impact of outdated information and retain the influence of users' stable preferences. Then, we model user and item dynamics by learning the multiple transitions at the user-factor and factor-item latent space between the ongoing time period and all past time periods. Accordingly, we formulate a joint objective function and take a gradient-based alternating optimization algorithm to solve this joint problem. Experimental results on historical datasets MovieLens show that the recommendation accuracy of MTMF with improved time weight is superior to the existing temporal recommendation methods.

INDEX TERMS Recommender systems, collaborative filtering, time weight, dynamic preference.

I. INTRODUCTION

With the continuous expansion of the Internet, information is rapidly growing at an explosive rate. Excessive information appears in front of users, making it impossible for users to distinguish and obtain effective information [1]–[3]. Recommender systems are regarded as an essential measure to solve this problem by analyzing the historical data and predicting the user's interest in the items [4]. Nevertheless, most of the existing recommender systems do not consider temporal dynamics, which can affect the accuracy of recommendation [5], [6]. In real-world recommender systems, each user and product tend to go through a distinct series of changes in their characteristics [7]. For example, a user liked cartoons when he was young, then science fiction films some years later, and now he prefers romantic films. User preferences are constantly changing over time. Meanwhile, the popularity of movies is also continually changing [8]. In this situation, ignoring the changes in users' preferences and items' characteristics impact the accuracy of recommendations. Thus, simultaneously modeling both temporal dynamics of users' interests and items' characteristics is critical to successful recommender systems.

There are many efforts made by a wide range of scholars modeling users' preference and items' characteristics dynamics. Koren [9] extended singular value decomposition (SVD) and incorporated time-variant biases to model users' activity and items' popularity. Based on the assumption that the user's preferences are gradually changing over time, Zhang *et al.* [10] proposed a Temporal Matrix Factorization(TMF) model. TMF models the temporal correlation of each user by a time-invariant transition matrix between two consecutive periods. However, the users' interests do not always evolve gradually but can also change radically between two consecutive periods. Rafailidis [11] assumed the user's preference in time period t is determined by the user preferences of all preceding time periods. He tried to model these multiple correlations at the user latent space

The associate editor coordinating the review of this manuscript and approving it for publication was Xin Luo.

by formulating a joint objective function. This model has dramatically inspired our work. However, this model only focuses on users' interests evolving without considering the shift in products' characteristics.

Moreover, all of the above models assume that the rating data is static. In the real-world recommender systems, the contribution of the ratings will change over time. Ding and Li [12] firstly applied the time weight to the rating matrix. The lower the score, the more the user is not interested. This method weakens the user's past interests and highlights the current interest. Sun and Long [13] applied the short-term, long-term, and periodic effects to the time impact factor matrix, using the singular value method to decompose and predict unknown score items. Li *et al.* [14] proposed a combined recommendation algorithm based on the similarity and forgetting curve. So far, there are few studies to apply time weight to the latent transition model [15].

Motivated by above problems, we propose to add time weight into the matrix factorization model and take into account items' evolving characteristics. In this paper, we propose a Multi-Trans matrix factorization (MTMF) model with improved time weight to capture temporal dynamics, which could result in more accurate recommendations.

Our contributions in this paper are summarized as follows:

- We propose an improved time weight based on forgetting curve and item similarity and introduce it into the matrix factorization model. Previous works only focus on time decay, which highlights the importance of recent data but underestimates the impact of users' long-term preferences. Data weight based on item similarity helps to preserve the impact of user stability preferences on ratings. Therefore, we combined two weight functions by a specific scale factor.
- TMF assumes items' features are stable and less correlated. However, in the real world, items' features will change over time. Therefore, we extended the TMF algorithm by introducing the item transition matrix to capture items' drift characteristics.
- Supposing that the user's current preference and item features transform from previous time periods, we split rating data into k time periods and apply the MTMF model to compute a k-1 transition metrics between the going time period and all the past ones. Then, we formulate a joint objective function with l_1-norm regularization to generate recommendations.

The rest of the paper is organized as follows: In Section II, we will give a definition of the problem. Section III introduces the related work on time-based data weight and dynamic collaborative filtering. Section IV illustrates our method of structural parts in detail. Experimental results are shown in section V, and section VI concludes the study.

II. PROBLEM STATEMENT

The user's rating of the item is represented as a matrix of m × n, where the rows V_j represent the set of items and the columns U_i represent the set of users, as shown in Table.1.

TABLE 1. The user's rating of the item.

	V_1	$\cdots$	V_j	$\cdots$	V_n
U_1	$r_{1,1}$	$\cdots$	$r_{1,j}$	$\cdots$	$r_{1,n}$
$\vdots$	$\vdots$		$\vdots$		$\vdots$
U_i	$r_{i,1}$	$\cdots$	$r_{i,j}$	$\cdots$	$r_{i,n}$
$\vdots$	$\vdots$		$\vdots$		$\vdots$
U_m	$r_{m,1}$	$\cdots$	$r_{m,j}$	$\cdots$	$r_{m,n}$

$R_{i,j}$ represents the score of user i for item j. For example, the MovieLens dataset uses 1 to 5 points to indicate the user's preference for the item; if $r_{i,j} = 0$ means the user i did not score the item j.

We divided the whole time span of the data into k non-overlapping periods. Let t be the current time period, which contains the users' most recent preferences.

Follow the Non-Negative Matrix Factorization(NMF) [16] model, recommendations in the ongoing time period t are generated by decomposing rating matrix R_t as follows:

$$\begin{aligned} &R_t \approx U_t V_t \\ &\text{Subject to } U_t \geq 0, \quad V_t \geq 0 \end{aligned} \tag{1}$$

where $U_t \in R^{m\times d}$ represents the factors of user and $V_t \in R^{d\times n}$ stands for the factors of items. The parameter d is the number of latent factors. Especially, d $\ll$ min(m, n). And, $U_t V_t$ represents the low-rank d approximation of matrix R_t.

In our research, we have to track the temporal dynamics of users' interests and products' characteristics simultaneously. Provided that we have k − 1 past time periods in total, we calculated the time decay value and type similarity of all previous time periods and assigned weights to the rating matrix. Then, we model the multiple transitions of user preferences and item shifts between all the t − k past time periods and the ongoing time period t. The graphical representation of this process is shown in Fig.1.

III. RELATED WORK

A. TIME-BASED DATA WEIGHT

In 1998, Crabtree and Soltysialk [17] believed that the user's recent rating is a reflection of the user's current interests. The past information has no value in generating recommendations to users. In 2000, Koychev and Schwab [18] proposed that past information also has some value for current recommendations, but we need to pay more attention to recent information. Therefore, they introduced a nonlinear forgetting function as a time weight. In 2005, Ding and Li [12] applied the time weight of the information to the user's rating, allowing the score to decay over time. This approach undermines the user's past interests and highlights current interests. In 2012, Rendle [19] segmented the information by time. The scores in different time periods represented the user's interest in each time period, which had different effects on the recommendations. In 2017, Sun and Dong [13]

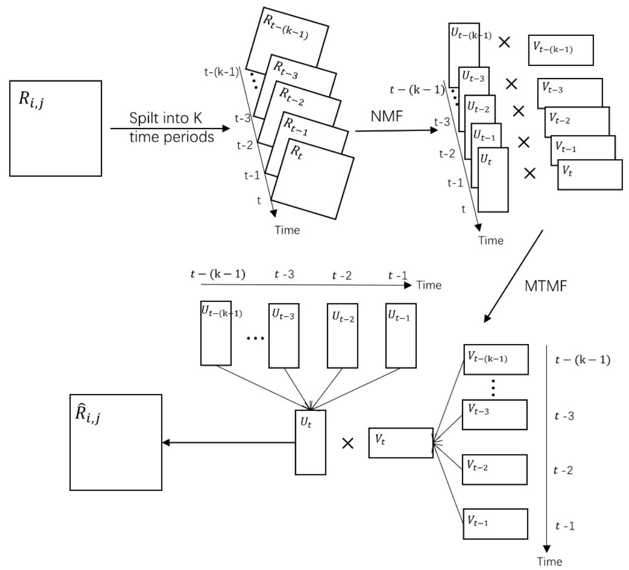


FIGURE 1. The graphical representation of MTMF.

weighted the rating matrix based on the time weight of the forgetting function and then used the singular value method to decompose and predict the unknown scoring items. In 2018, Hyunwoo Hwangbo *et al.* [20] adopted a preference decay function to reflect changes in preferences over time, and finally recommended substitute and complementary products by using product category information. In 2019, Li *et al.* [14] combines user similarity and nonlinear forgetting functions based on improved Pearson correlation coefficients.

B. DYNAMIC COLLABORATIVE FILTERING

Koren [9] extends singular value decomposition(SVD) and introduces time-variant biases for user/item to model users' activity and items' popularity throughout the entire time period. However, the model does not summarize the underlying pattern for the changes of biases in different windows. By introducing a set of additional time features to traditional factor-based collaborative filtering algorithms, Xiong *et al.* [21] proposed a Bayesian probabilistic tensor factorization (BPTF) model to deal with the global evolution of latent features. With the analogous consideration, Zhang *et al.* [10] proposed two dynamic collaborative filtering models: Temporal Matrix Factorization(TMF) and Bayesian Temporal Matrix Factorization (BTMF). TMF models the temporal dependence for each user i through a $D \times D$ transition matrix between two continuous time periods. BTMF extends the TMF method to a fully Bayesian treatment by introducing priors for the hyperparameters which capture the conditional of users, items, and a single type of interaction. Based on the same assumption that the user's current preference is determined by his preference at the previous time period, Temporal Collective Matrix Factorization (TCMF) [22] was proposed by Rafailidis in 2017. In order to capture users' preference dynamics, TCMF extracts the users' temporal pattern through a joint decomposition model and minimizing the joint objective function to generate the recommendation. In 2018, Rafailidis [11] proposed the Multi-Latent Transition (MLT) model, which tries to model these multiple correlations at the user latent space.

IV. PROPOSED MODEL

This section is divided into four parts:(I) We will propose an improved time weight to model the rating score shift rate in Section A;(II) In Section B, we introduce item Transition Matrix to capture items' shift;(III) Accordingly, we formulate a joint objective function in Section C;(IV) Finally, we detail the optimization method of the model we proposed.

A. IMPROVED TIME WEIGHT

The forgetting curve describes how humans forget information over time [23]. In the study of the application of the forgetting curve, the nonlinear exponential function and the information half-life are usually used to describe the forgetting function [24]. The information half-life refers to the time elapsed from the moment of occurrence of the information to the half of being forgotten and defines the attenuation factor $\omega = \ln(0.5)\big/T_0$. The amount of retention of the final message at the current $t-k$ interval is:

$$W_{t-k} = e^{-\omega*[t-(t-k)]} = e^{-\omega*k} \tag{2}$$

The range of values is (0, 1), and the value decreases as the time interval k increases, indicating that the longer the scoring time, the smaller contribution of the score to the current recommendation.

However, the time weight with a forgetting curve will cause the system to underestimate the impact of the user's long-term preference on user preferences. Some of the user's preferences are stable and do not fluctuate over time. For example, the user "Tom" expresses a preference for a suspense movie in all k time periods, which means the suspense feature does not decay or is forgotten. If this feature is attenuated over time, the contribution of the long-term preference to the current recommendation is reduced. Therefore, we will introduce a time weight based on the similarity of the item type:

$$I_{t-k} = \frac{\sum_{a=1}^{\left|L_i^t\right|}\sum_{b=1}^{\left|L_i^{t-k}\right|} s(a,b)}{num(L_i^t) \times num(L_i^{t-k})} \tag{3}$$

where $num(L_i^{t-k})$ represents the number of items that the user i has expressed his preferences in the time period $t-k$, with $i = 1, 2, \ldots n$, and $k = 0, 1, 2 \ldots k\text{-}1$. $s(a, b)$ denotes the type similarity of two items. The value of I_{t-k} corresponds to the speed of change of a user's interest in the recommendation system, which is a positive proportional correlation.

Finally, we apply a weight reduction factor that combines time decay and type similarity:

$$S_{t-k} = \alpha * W_{t-k} + (1-\alpha) * I_{t-k} \tag{4}$$

The improved time weight S_{t-k} takes into account that the contribution of the score value will decrease over time and simultaneously add type similarity that does not decay over time. By choosing an appropriate scale factor α value, we can combine the advantages of the two weighting methods.

B. INTRODUCING ITEM TRANSITION MATRIX

In the Temporal Probabilistic Matrix Factorization(TMF) model, Chengyi Zhang assumes that there is a temporal dependence between the latent user-factor vectors U_{it} and $U_{i,t-1}$. So he introduces a transition hypermatrix $B\epsilon\mathbb{R}^{N\times d\times d}$ to model this dependency.

$$U_{it} = B_i U_{i,t-1} \tag{5}$$

B_i is used to capture the time-invariant aspect of the user i's interest from the previous time period to the next. For example, take the d = 2 latent space. $B_i = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ represents a stable user preference that does not change over time. $B_i = \begin{bmatrix} 1.1 & 0 \\ 0 & 1 \end{bmatrix}$ represents a gradual shift to the first factor. TMF uses transition hypermatrix B_i to model user temporal dynamics.

However, TMF does not model this dependency for the item since Chengyi Zhang assumes that the item's features are stable and less correlated. In a real-world recommendation system, features of the item, such as popularity and quality, are strongly correlated with time.

So, Extending the idea of TMF, We propose a new item transition hypermatrix $C\epsilon\mathbb{R}^{M\times d\times d}$ to capture the evolution of item as follows:

$$V_{jt} = C_j V_{j,t-1} \tag{6}$$

With both transition matrix B_i and C_j, we can predict the rating of user i on item j at the next time period by the rule $R^*_{i,j} = U_{it}V_{jt} = (U_{i,t-1}B_i)(C_j V_{j,t-1})$. This formula means that we learn the user-factor matrix $U_{i,t-1}$ and item-factor matrix $V_{j,t-1}$ at the time period $t-1$, and transition matrix B_i and C_j model the transition from the previous time period $t-1$ to the current period t. The graphical representation of MTMF with item Transition Matrix showed in Fig.2

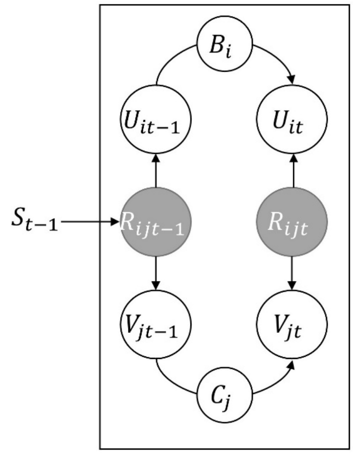


FIGURE 2. The graphical representation of MTMF with item transition matrix.

C. OBJECTIVE FUNCTION

When the transition matrix is not introduced, the scoring matrix R_t is decomposed into user-factor and item-factor vectors, which describe the user's interests and item's features by non-negative matrix factorization(NMF) model as follows:

$$R_t \approx U_t V_t \tag{7}$$

To simplify the expression, we set k = 3. The data set is divided into three time periods, which are R_t, R_{t-1} and R_{t-2}. Firstly, we apply the time weight in a personalized manner as follows:

$$\forall j = 1, \ldots\ldots n$$

$$(R_{t-1})_{*,j} \leftarrow S_{t-1,j} \cdot (R_{t-1})_{*,j} \tag{8}$$

$$(R_{t-2})_{*,j} \leftarrow S_{t-2,j} \cdot (R_{t-2})_{*,j} \tag{9}$$

where $(R_{t-1})_{*,j}$ denotes the jth column of the rating matrix R_{t-1}, and $(R_{t-2})_{*,j}$ denotes the jth column of the rating matrix R_{t-2}. Based on the minimization problem of NMF in Eq.(7), the NMFs of matrices R_t, R_{t-1} and R_{t-2} correspond to the following three minimization problems:

$$\min_{U_t, V_t} \parallel R_t - U_t V_t \parallel_F^2 \tag{10}$$

$$\min_{U_{t-1}, V_{t-1}} \parallel R_{t-1} - U_{t-1} V_{t-1} \parallel_F^2 \tag{11}$$

$$\min_{U_{t-2}, V_{t-2}} \parallel R_{t-2} - U_{t-2} V_{t-2} \parallel_F^2 \tag{12}$$

$$\text{subject to } U_t, V_t,\ U_{t-1},\ V_{t-1},\ U_{t-2},\ V_{t-2} \geq 0,$$

where $\|\cdot\|_F$ denotes the Frobenius norm. Since users' interests vary over time, we assume that there is a temporal dependence between users' current interests and users' previous interests. We model this dependency by a transition B_t. Using the user's preference transitions matrices B_{t-1} and B_{t-2}, Eq.(10) can be rewritten as follows:

$$\min_{U_{t-1}, V_{t-1}, B_{t-1}} \parallel R_t - U_{t-1} B_{t-1} V_{t-1} \parallel_F^2 \tag{13}$$

$$\min_{U_{t-2}, B_{t-2}, V_{t-2}} \parallel R_t - U_{t-2} B_{t-2} V_{t-2} \parallel_F^2 \tag{14}$$

Similarly, items' characteristics are also continually changing. Thus, we define an item transition matrix C_t which denotes the evolution of item features between the item latent matrices V_t and V_{t-1}. Using the users' preference transitions matrices B_{t-1} and B_{t-2}, Eq.(13),(14) can be rewritten as the following minimization problems:

$$\min_{U_{t-1}, V_{t-1}} \parallel R_t - U_{t-1} B_{t-1} C_{t-1} V_{t-1} \parallel_F^2 \tag{15}$$

$$\min_{U_{t-2}, V_{t-2},} \parallel R_t - U_{t-2} B_{t-2} C_{t-2} V_{t-2} \parallel_F^2 \tag{16}$$

To consider both the user preference dynamics and item characteristics changing, we combine Eq. (7), (15) and (16), which results in the following joint objective function:

$$\begin{aligned}\min_{U_t,B_{t-1},B_{t-2},C_{t-1},C_{t-2}} \quad & \mathcal{L} = \| R_t - U_t V_t \|_F^2 \\ & + \| R_t - U_{t-1}B_{t-1}C_{t-1}V_{t-1} \|_F^2 \\ & + \| R_t - U_{t-2}B_{t-2}C_{t-2}V_{t-2} \|_F^2 \\ & + \lambda(\| \Big(B_{t-1} - I \|_F^2 + \| B_{t-2} - I \|_F^2 \\ & + \| C_{t-1} - I \|_F^2 + \| C_{t-2} - I \|_F^2\Big) \\ & + \beta(\| U_{t-1} \|_1 + \| V_{t-1} \|_1 \\ & + \| U_{t-2} \|_1 + \| V_{t-2} \|_1 + \| U_t \|_1 \\ & + \| V_t \|_1) \\ \text{Subject to } & U_t,\ V_t,\ U_{t-1},\ V_{t-1},\ U_{t-2},\ V_{t-2} \geq 0\end{aligned} \tag{17}$$

where $I \in \mathbb{R}^{d\times d}$ is the identity matrix, and $\| \cdot \|_1$ stands for the l_1-norm. In Eq. (17), the first three terms denote the low-rank d approximation errors based on Eq. (15) and Eq.(16). The fourth term is the temporal regularization between the user/item transitions and identity matrices. The λ parameter controls how much we want to make the model biased to the past interest/ characteristics. The fifth term is l_1-norm regularization, which forces the factor matrices $U_t, V_t, U_{t-1}, V_{t-1}, U_{t-2}, V_{t-2}$ to be sparse. The parameter β controls the impact of the l_1-norm regularization. To summarize, we formulate a joint objective function to calculte the k-1 transitions of users' interests and items' characteristics.

D. MODEL LEARNING

Our goal is to minimize the joint objective function $\mathcal{L}$ in Eq.(17), but it is a non-convex function with respect to six variables/matrices. Learning from [11], [25], we applied an alternating optimization algorithm based on the strategy of multiplicative update rules, where we update a variable while keeping the rest of the variables fixed.

First, considering the Karush-Kuhn-Tucker(KTT) [26] conditions, we have:

$$\begin{aligned} U_t \geq 0, \quad V_t \geq 0, \quad B_{t-1} \geq 0, \\ B_{t-2} \geq 0, \quad C_{t-1} \geq 0, \quad C_{t-2} \geq 0 \end{aligned} \tag{18}$$

$$\begin{cases} \nabla_{U_t}\mathcal{L} \geq 0, \quad \nabla_{V_t}\mathcal{L} \geq 0, \quad \nabla_{B_{t-1}}\mathcal{L} \geq 0 \\ \nabla_{B_{t-2}}\mathcal{L} \geq 0, \quad \nabla_{C_{t-1}}\mathcal{L} \geq 0, \quad \nabla_{C_{t-2}}\mathcal{L} \geq 0 \end{cases} \tag{19}$$

$$\begin{cases} U_t \odot \nabla_{U_t}\mathcal{L} = 0, \quad V_t \odot \nabla_{V_t}\mathcal{L} = 0, \\ B_{t-1} \odot \nabla_{B_{t-1}}\mathcal{L} = 0 \\ B_{t-2} \odot \nabla_{B_{t-2}}\mathcal{L} = 0, \quad C_{t-1} \odot \nabla_{C_{t-1}}\mathcal{L} = 0, \\ C_{t-2} \odot \nabla_{C_{t-2}}\mathcal{L} = 0 \end{cases} \tag{20}$$

where $\odot$ denotes the element-wise product.

According to the objective function in Eq. (17), the gradients for each parameter are derived respectively:

$$\nabla_{U_t}\mathcal{L} = U_t \mathrm{V}_t \mathrm{V}_t^T - (\mathrm{R}_t \mathrm{V}_t^T - \beta) \tag{21}$$

$$\nabla_{V_t}\mathcal{L} = \mathrm{U}_t^T \mathrm{U}_t \mathrm{V}_t - (\mathrm{U}_t^T \mathrm{R}_t - \beta) \tag{22}$$

$$\begin{aligned}\nabla_{B_{t-1}}\mathcal{L} = {} & \mathrm{U}_{t-1}^T \mathrm{U}_{t-1} B_{t-1} C_{t-1} \mathrm{V}_{t-1} \mathrm{V}_{t-1}^T \mathrm{C}_{t-1}^T \\ & - \mathrm{U}_{t-1}^T \mathrm{R}_t \mathrm{V}_{t-1}^T \mathrm{C}_{t-1}^T + \lambda(B_{t-1} - I)\end{aligned} \tag{23}$$

$$\begin{aligned}\nabla_{B_{t-2}}\mathcal{L} = {} & \mathrm{U}_{t-2}^T \mathrm{U}_{t-2} B_{t-2} C_{t-2} \mathrm{V}_{t-2} \mathrm{V}_{t-2}^T \mathrm{C}_{t-2}^T \\ & - \mathrm{U}_{t-2}^T \mathrm{R}_t \mathrm{V}_{t-2}^T \mathrm{C}_{t-2}^T + \lambda(B_{t-2} - I)\end{aligned} \tag{24}$$

$$\begin{aligned}\nabla_{C_{t-1}}\mathcal{L} = {} & \mathrm{B}_{t-1}^T \mathrm{U}_{t-1}^T \mathrm{U}_{t-1} B_{t-1} C_{t-1} \mathrm{V}_{t-1} \mathrm{V}_{t-1}^T \\ & - \mathrm{B}_{t-1}^T \mathrm{U}_{t-1}^T \mathrm{R}_t \mathrm{V}_{t-1}^T + \lambda(C_{t-1} - I)\end{aligned} \tag{25}$$

$$\begin{aligned}\nabla_{C_{t-2}}\mathcal{L} = {} & \mathrm{B}_{t-2}^T \mathrm{U}_{t-2}^T \mathrm{U}_{t-2} B_{t-2} C_{t-2} \mathrm{V}_{t-2} \mathrm{V}_{t-2}^T \\ & - \mathrm{B}_{t-2}^T \mathrm{U}_{t-2}^T \mathrm{R}_t \mathrm{V}_{t-2}^T + \lambda(C_{t-2} - I)\end{aligned} \tag{26}$$

Based on the gradients in Eqs. (18) -(23), by substituting the corresponding gradients in Eq. (20), the following updating rules are derived:

$$U_t \leftarrow U_t \odot \frac{\mathrm{R}_t \mathrm{V}_t^T - \beta}{U_t \mathrm{V}_t \mathrm{V}_t^T} \tag{27}$$

$$V_t \leftarrow V_t \odot \frac{\mathrm{U}_t^T \mathrm{R}_t - \beta}{\mathrm{U}_t^T \mathrm{U}_t \mathrm{V}_t} \tag{28}$$

$$B_{t-1} \leftarrow B_{t-1} \odot \frac{\lambda(B_{t-1} - I) - \mathrm{U}_{t-1}^T \mathrm{R}_t \mathrm{V}_{t-1}^T \mathrm{C}_{t-1}^T}{\mathrm{U}_{t-1}^T \mathrm{U}_{t-1} B_{t-1} C_{t-1} \mathrm{V}_{t-1} \mathrm{V}_{t-1}^T \mathrm{C}_{t-1}^T} \tag{29}$$

$$B_{t-2} \leftarrow B_{t-2} \odot \frac{\lambda(B_{t-2} - I) - \mathrm{U}_{t-2}^T \mathrm{R}_t \mathrm{V}_{t-2}^T \mathrm{C}_{t-2}^T}{\mathrm{U}_{t-2}^T \mathrm{U}_{t-2} B_{t-2} C_{t-2} \mathrm{V}_{t-2} \mathrm{V}_{t-2}^T \mathrm{C}_{t-2}^T} \tag{30}$$

$$C_{t-1} \leftarrow C_{t-1} \odot \frac{\lambda(C_{t-1} - I) - \mathrm{B}_{t-1}^T \mathrm{U}_{t-1}^T \mathrm{R}_t \mathrm{V}_{t-1}^T}{\mathrm{B}_{t-1}^T \mathrm{U}_{t-1}^T \mathrm{U}_{t-1} B_{t-1} C_{t-1} \mathrm{V}_{t-1} \mathrm{V}_{t-1}^T} \tag{31}$$

$$C_{t-2} \leftarrow C_{t-2} \odot \frac{\lambda(C_{t-2} - I) - \mathrm{B}_{t-2}^T \mathrm{U}_{t-2}^T \mathrm{R}_t \mathrm{V}_{t-2}^T}{\mathrm{B}_{t-2}^T \mathrm{U}_{t-2}^T \mathrm{U}_{t-2} B_{t-2} C_{t-2} \mathrm{V}_{t-2} \mathrm{V}_{t-2}^T} \tag{32}$$

So far, we have analyzed the case where there are two past time periods and an ongoing time period. Accordingly, we will give the joint objective function when p $= k$-1 as follows:

$$\begin{aligned}\min_{U_t,V_t,B_{t-1},B_{t-2},C_{t-1},C_{t-2}} \mathcal{L} & \\ = {} & \| \mathrm{R}_t - \mathrm{U}_t \mathrm{V}_t \|_F^2 \\ & + \sum_{p=1}^{k-1} \| \mathrm{R}_t - \mathrm{U}_{t-\mathrm{p}} B_{t-\mathrm{p}} C_{t-\mathrm{p}} \mathrm{V}_{t-\mathrm{p}} \|_F^2 \\ & + \lambda \sum_{p=1}^{k-1} (\| B_{t-\mathrm{p}} - I \|_F^2 + \| C_{t-\mathrm{p}} - I \|_F^2) \\ & + \beta \sum_{p=1}^{k-1} (\| U_{t-\mathrm{p}} \|_1 + \| V_{t-\mathrm{p}} \|_1)\end{aligned} \tag{33}$$

The respective updating rule for transition matrices $B_{t-\mathrm{p}}$ and $C_{t-\mathrm{p}}$ is as follows:

$$B_{t-\mathrm{p}} \leftarrow B_{t-\mathrm{p}} \odot \frac{\lambda(B_{t-\mathrm{p}} - I) - \mathrm{U}_{t-p}^T \mathrm{R}_t \mathrm{V}_{t-p}^T \mathrm{C}_{t-p}^T}{\mathrm{U}_{t-p}^T \mathrm{U}_{t-p} B_{t-\mathrm{p}} C_{t-\mathrm{p}} \mathrm{V}_{t-\mathrm{p}} \mathrm{V}_{t-p}^T \mathrm{C}_{t-p}^T} \tag{34}$$

$$C_{t-\mathrm{p}} \leftarrow C_{t-\mathrm{p}} \odot \frac{\lambda(C_{t-\mathrm{p}} - I) - \mathrm{B}_{t-\mathrm{p}}^T \mathrm{U}_{t-\mathrm{p}}^T \mathrm{R}_t \mathrm{V}_{t-\mathrm{p}}^T}{\mathrm{B}_{t-\mathrm{p}}^T \mathrm{U}_{t-\mathrm{p}}^T \mathrm{U}_{t-\mathrm{p}} B_{t-\mathrm{p}} C_{t-\mathrm{p}} \mathrm{V}_{t-\mathrm{p}} \mathrm{V}_{t-\mathrm{p}}^T} \tag{35}$$

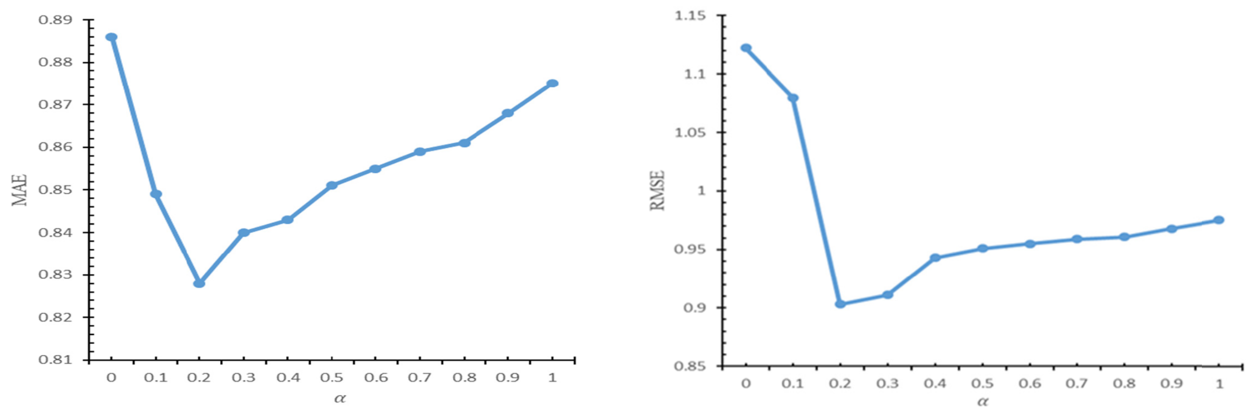


FIGURE 3. The impact of the scale factor α on MAE and RMSE.

Algorithm 1 presents our proposed Multi-trans Matrix Factorization (MTMF) method. In the second line, we initialize the factor matrices $U_t$and V_t by random matrices, with$V_t \geq 0, U_t \geq 0$. In line 3, the k-1 latent transition matrices are initialized by the rule $B_{t-\mathrm{p}} \leftarrow I, C_{t-\mathrm{p}} \leftarrow I$, with $p = 1, 2, \ldots .k - 1$. Then, we compute V_{t-p} and U_{t-p} by applying NMF on $\mathrm{R}_{t-\mathrm{p}}$. Our iterative optimization algorithm first update variables U_t and V_t based on Eq.(34) and Eq.(35). Similarly, the k-p transition matrices $B_{t-\mathrm{p}}$ and $C_{t-\mathrm{p}}$ are updated on the updating rule in Eq.(34) and (35). At the end of each iteration, we will Compute $\mathcal{L}$ based on the updated $U_t, V_t, B_{t-\mathrm{p}}, C_{t-\mathrm{p}}(Eq.(33))$. After the optimization algorithm converges, we will compute the factorized matrix $\hat{R}_t$, which contains the final recommendations.

V. EXPERIMENT

A. DATASETS AND EVALUATION METRICS

In our experiment, we evaluated our method and compared it on the MovieLens-1M. Ratings in this dataset are collected

Algorithm 1 Learning Algorithm for MNMF

Input: $\mathrm{R}_t, \ldots\ldots \mathrm{R}_{t-1}, \ldots\ldots \mathrm{R}_1, \{\alpha, \lambda, \mathrm{d}, \beta\}$ *maxIter*
Output: $\hat{R}_t$
1. $\delta' \leftarrow maxInt, \delta \leftarrow \frac{\delta'}{2}$
2. Initialize U_t and V_t
3. $B_{t-\mathrm{p}} \leftarrow I, C_{t-\mathrm{p}} \leftarrow I, \forall p = 1, 2, \ldots .k - 1$
4. Compute V_{t-p} and U_{t-p} by applying NMF on $\mathrm{R}_{t-\mathrm{p}}$
   while abs($\delta' - \delta) \geq \varepsilon \bigvee$ iter < maxIter **do**
5. Update $U_t(Eq.\ (27))$
6. Update $V_t(Eq.\ (28))$
7. Update $B_{t-\mathrm{p}} \forall p = 1, 2, \ldots .k - 1(Eq.(34))$
8 Update $C_{t-\mathrm{p}} \forall p = 1, 2, \ldots .k - 1(Eq.(35))$
9. Compute $\mathcal{L}$ based on the updated $U_t, V_t, B_{t-\mathrm{p}}, C_{t-\mathrm{p}}(Eq.\ (33))$
10. $\delta' \leftarrow \delta$
11. $\delta \leftarrow L$
12. iter ← iter + 1
13. **End while**
14. $\hat{R}_t = U_t V_t$

by the University of Minnesota. MovieLens-1M contains 1,000,209 ratings of 3,952 movies by 6,040 MovieLens' users in 2000.Each rating information consists of user ID, movie ID, rating, and score timestamp over 36 months. The value of movie rating ranges from 1 to 5. Especially, each movie's information in movie.dat consists of the movie ID, movie name, release time, and the type of movie. There are 18 movie types, such as "Romantic," "Tragedy," "Science Fiction" and so on. Moreover, a movie can belong to two or more types at the same time. We use the genre of the movie to calculate the type similarity $s(a, b)$ of the rated movie. If two films belong to the same genre $s(a, b) = 1$, otherwise $s(a, b) = 0$.

In our experiment, we slice this dataset into six time periods. Thus, we have six different time slices, where each slice corresponds to six months. We used the first five months as a training set and the sixth month as a test set. Similarly, in the second time slice, the first eleven months are used as input to the model, and the twelfth month is used as the test set. We have a total of 36 months. Therefore, we have six different test sets at 6, 12, 18, 24, 30, and 36 months.

We evaluate the performance in terms of the Mean Absolute Error(MAE) and Root-Mean-square Error (RMSE) metric compared with other baselines temporal recommendation models. MAE is defined as follows:

$$MAE = \frac{\sum_{i,j} |r_{ij} - \hat{r}_{ij}|}{N} \tag{36}$$

where r_{ij} denotes the actual rating user i gave to item j, $\hat{r}_{ij}$ denotes the predicted rating, and N denotes the total number of predictions. A higher value of MAE expresses a high accuracy of the prediction model.

$$RMSE = \sqrt{\frac{1}{N} \sum_{i,j} (r_{ij} - \hat{r}_{ij})^2} \tag{37}$$

RMSE is the root mean square of the error between the predicted and actual values. Compared with MAE, RMSE increase the punishment of significant error.

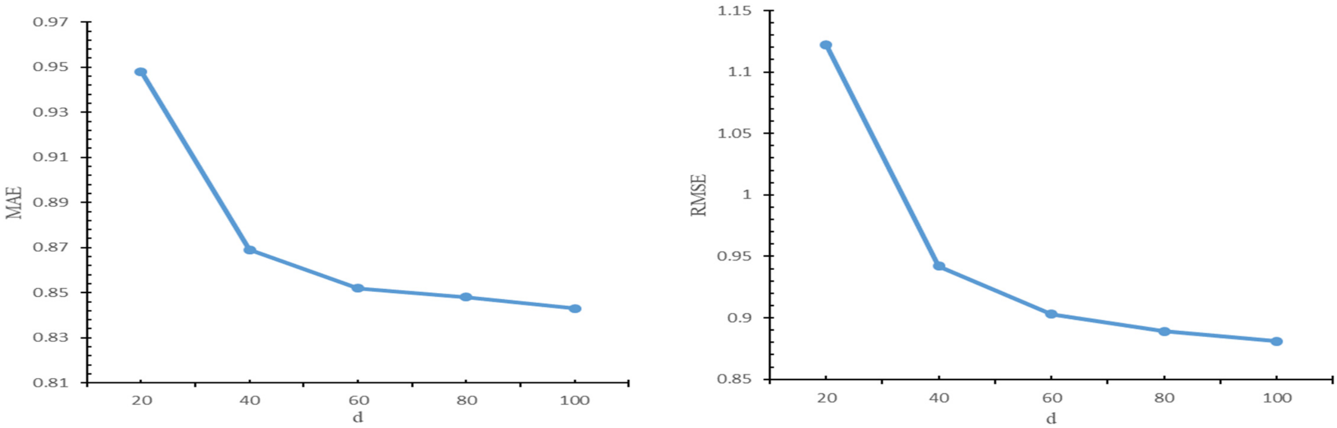


FIGURE 4. The impact of the number of latent dimensions d on MAE and RMSE.

B. COMPARED METHODS

In order to evaluate the prediction accuracy of our proposed model, we compare it with the other four temporal recommendation algorithms as follows:

- TimeSVD++: a method of extending singular value decomposition (SVD) and introducing time-dependent user and item bias to simulate user activity and items popularity
- Temporal Matrix Factorization(TMF): assumes user preferences evolve gradually and models the temporal dependence of user i through a D × D transition matrix between two consecutive time periods.
- Bayesian Temporal Matrix Factorization(BTMF): extends the TMF method to a fully Bayesian treatment by introducing priors for the hyper-parameters which capture the conditional of users, items, and a single type of interactions.
- Multi-Latent Transition (MLT): assumes the user's preference in time period t is determined by the user preferences of all preceding time periods and tries to model these multiple correlations at the user latent space by formulating a joint objective function.

C. PARAMETER ANALYSIS

There are two critical parameters in our proposed MTMF model with improved time weight: the scale factor α between type similarity and time decay and the number of latent dimensions d.

1) ADJUSTMENT WORK OF α

In Fig.3, we present the impact of the scale factor α on MAE and RMSE. α is a factor used to balance the user's score time decay and the type similarity. In this set of experiments, we vary α from 0 to 1 with an interval of 0.1. As we can see from Fig.4, the values of MAE and RMSE are the minimum when $\alpha = 0.2$, which indicates that the user's recent preference has a considerable impact.

2) ADJUSTMENT WORK OF D

In the choice of latent dimensions d, we tested it from 20 to 100 with an interval of 20. In Fig.5, we present the effect of d on MAE and RMSE. When $d \geq 60$, there is a slight improvement of MAE and RMSE. Thus, we fix d = 60.

3) PERFORMANCE EVALUATION

We compare the model we propose with some temporal recommendation methods: TimeSVD++, TMF, BTMF, and MLT. Moreover, we compare with the MTMF without time weight to evaluate the impact of the improved time weight on recommendation accuracy. We report the results of experiments in Fig.5. Clearly, all the examined models track the temporal dynamics over the time span. Also, with the expansion of the training set, the augmented training sets improve the MAE and RMSE metric throughout the data time periods. In all comparison methods, the baseline methods of TimeSVD ++ and TMF perform poorly over the entire time range. They do not capture user dynamics very well. By introducing priors for hyperparameters of TMF, BTMF significantly improves recommendation accuracy.

Meanwhile, MLT achieves higher accuracy than other matrix factorization models. This happens because MLT exploits the bimodal user-item interactions and captures the users' preference dynamics over the data sets' evolution. However, TCMF does not consider the items' drift characteristics, which limits the performance of MLT.

As we can see, the Multi-Trans matrix factorization (MTMF) model with improved time weight excels parallel models in the evaluation of MAE and RMSE. The improved time weight takes into account both the time decay of the user's preferences and the effect of the user's stable preferences on the rating. Without improved time weight, the accuracy of the MTMF model is slightly reduced.

What should be mentioned is the time consumed by MTMF with improved time weight is highest among all the compared models. This computational efficiency gap may be caused

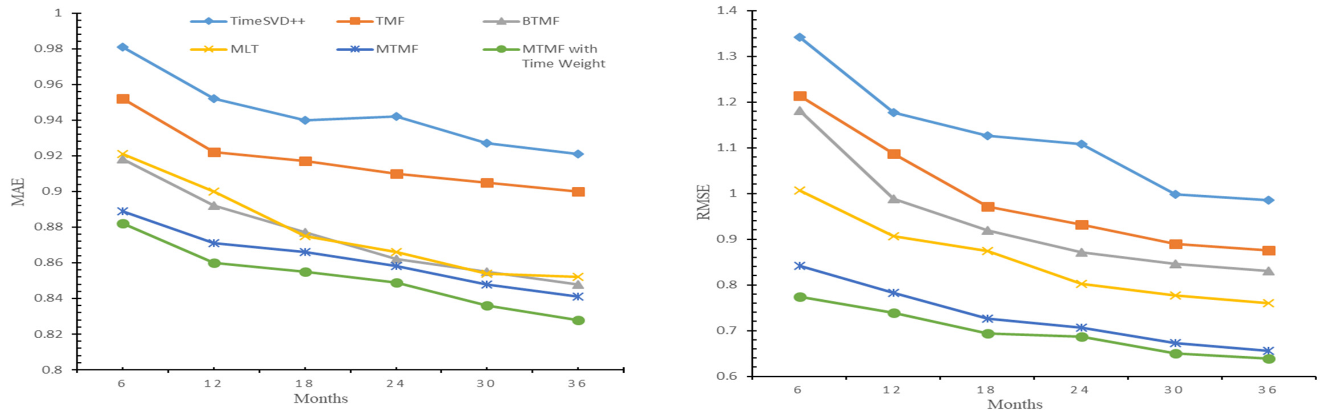


FIGURE 5. Performance throughout the data time span.

by the multi-trans between users and items latent factors matrices. This issue is worthy of further investigation.

VI. CONCLUSION AND FUTURE WORK

In this paper, we propose a multi-trans matrix factorization (MTMF) model with improved time weight to capture items' changing and users' shifting preferences to make a better recommendation. The improved time weight, which combines the forgetting curve and similarity of the rating item, is introduced into the matrix factorization model. To track the temporal dynamics of both drifts simultaneously, we extend the TMF algorithm by introducing the item transition matrix. Accordingly, we formulate a joint objective function to compute the temporal correlation of each user and item between the current period and all the previous periods. By applying on the datasets MovieLens, this model achieves high recommendation accuracy over several competitive recommender algorithms.

There are still several avenues for future work. One exciting direction is the exploitation of explicit and implicit feedbacks by the users [27]. Implicit feedback of users can provide auxiliary data that helps to reflect the user's preference for the product. In addition, referring to [28] and [29], it remains challenging to reduce the calculation and storage efficiency of MTMF while improving recommendation accuracy.

REFERENCES

[1] J. Chen, W. Zeng, J. Shao, and G. Fan, "Preference modeling by exploiting latent components of ratings," *Knowl. Inf. Syst.*, vol. 60, no. 1, pp. 495–521, Jul. 2019.

[2] X. Luo, M. Zhou, S. Li, Y. Xia, Z.-H. You, Q. Zhu, and H. Leung, "Incorporation of efficient second-order solvers into latent factor models for accurate prediction of missing QoS data," *IEEE Trans. Cybern.*, vol. 48, no. 4, pp. 1216–1228, Apr. 2018.

[3] D. Véras, R. Prudêncio, and C. Ferraz, "CD–CARS: Cross-domain context-aware recommender systems," *Expert Syst. Appl.*, vol. 135, pp. 388–409, Nov. 2019.

[4] I. Al-Hadi, N. M. Sharef, M. N. Sulaiman, and N. Mustapha, "Review of the temporal recommendation system with matrix factorization," *Int. J. Innov. Comput. Inf. Control*, vol. 13, no. 5, pp. 1579–1594, 2017.

[5] C. Xiao, C. Xie, S. Cao, Y. Zhang, W. Fan, and H. Heng, "A better understanding of the interaction between users and items by knowledge graph learning for temporal recommendation," in *Proc. 16th Pacific Rim Int. Conf. Artif. Intell.*, Cuvu, Fiji, Aug. 2019, pp. 135–147.

[6] H. Hwangbo and Y. Kim, "An empirical study on the effect of data sparsity and data overlap on cross domain collaborative filtering performance," *Expert Syst. Appl.*, vol. 89, pp. 254–265, Dec. 2017.

[7] Q. Zheng, "An improved collaborative filtering algorithm based on expert trust and time decay," in *Proc. 11th Int. Symp. Comput. Intell. Design (ISCID)*, Dec. 2018, pp. 12–15.

[8] F. Zafari, I. Moser, and T. Baarslag, "Modelling and analysis of temporal preference drifts using a component-based factorised latent approach," *Expert Syst. Appl.*, vol. 116, pp. 186–208, Feb. 2019.

[9] Y. Koren, "Collaborative filtering with temporal dynamics," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2009, pp. 447–456.

[10] C. Zhang, K. Wang, H. Yu, J. Sun, and E.-P. Lim, "Latent factor transition for dynamic collaborative filtering," in *Proc. SIAM Int. Conf. Data Mining*, Apr. 2014, pp. 452–460.

[11] D. Rafailidis, "A Multi–Latent Transition model for evolving preferences in recommender systems," *Expert Syst. Appl.*, vol. 104, pp. 97–106, Aug. 2018.

[12] Y. Ding and X. Li, "Time weight collaborative filtering," in *Proc. 14th ACM Int. Conf. Inf. Knowl. Manage.*, 2005, pp. 485–492.

[13] B. Sun and L. Dong, "Dynamic model adaptive to user interest drift based on cluster and nearest neighbors," *IEEE Access*, vol. 5, pp. 1682–1691, 2017.

[14] T. Li, L. Jin, Z. Wu, and Y. Chen, "Combined recommendation algorithm based on improved similarity and forgetting curve," *Information*, vol. 10, no. 4, p. 130, Apr. 2019.

[15] X. Luo, J. Sun, Z. Wang, S. Li, and M. Shang, "Symmetric and nonnegative latent factor models for undirected, high–dimensional, and sparse networks in industrial applications," *IEEE Trans. Ind. Informat.*, vol. 13, no. 6, pp. 3098–3107, Dec. 2017.

[16] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2001, pp. 556–562.

[17] I. Barry Crabtree and S. J. Soltysiak, "Identifying and tracking changing interests," *Int. J. Digit. Libraries*, vol. 2, no. 1, pp. 38–53, Oct. 1998.

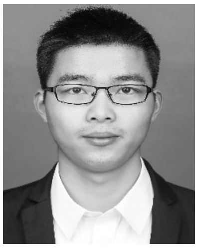
[18] I. Koychev and I. Schwab, "Adaptation to drifting user's interests," in *Proc. Proc. ECML Workshop, Mach. Learn. New Inf. Age*, 2000, pp. 39–46.

[19] S. Rendle, "Social network and click-through prediction with factorization machines," in *Proc. KDD-Cup Workshop*, 2012, p. 113.

[20] H. Hwangbo, Y. S. Kim, and K. J. Cha, "Recommendation system development for fashion retail e-commerce," *Electron. Commerce Res. Appl.*, vol. 28, pp. 94–101, Mar. 2018.

[21] L. Xiong, X. Chen, T.-K. Huang, J. Schneider, and J. G. Carbonell, "Temporal collaborative filtering with Bayesian probabilistic tensor factorization," in *Proc. SIAM Int. Conf. Data Mining*, Apr. 2010, pp. 211–222.

[22] D. Rafailidis, P. Kefalas, and Y. Manolopoulos, "Preference dynamics with multimodal user-item interactions in social media recommendation," *Expert Syst. Appl.*, vol. 74, pp. 11–18, May 2017.
[23] S. Wei, N. Ye, and Q. Zhang, "Time-aware collaborative filtering for recommender systems," in *Proc. Chin. Conf. Pattern Recognit. (CCPR)*, Beijing, China, Sep. 2012, pp. 663–670.
[24] H. Su, X. Lin, B. Yan, and H. Zheng, "The collaborative filtering algorithm with time weight based on mapreduce," in *Proc. 1st Int. Conf. Big Data Comput. Commun. (BigCom)*, Taiyuan, China, Aug. 2015, pp. 386–395.
[25] D. Song, L. Qin, M. Jiang, and L. Liao, "A temporal and topic–aware recommender model," in *Proc. IEEE Int. Conf. Big Data Smart Comput. (BigComp)*, Jan. 2018, pp. 410–417.
[26] D. P. Bertsekas, "Nonlinear programming," *J. Oper. Res. Soc.*, vol. 48, no. 3, p. 334, 1997.
[27] J. Chen, C. Wang, J. Wang, and P. S. Yu, "Recommendation for repeat consumption from user implicit feedback," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 11, pp. 3083–3097, Nov. 2016.
[28] X. Luo, M. Zhou, Y. Xia, Q. Zhu, A. C. Ammari, and A. Alabdulwahab, "Generating highly accurate predictions for missing QoS data via aggregating nonnegative latent factor models," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 3, pp. 524–537, Mar. 2016.
[29] X. Luo, M. Zhou, S. Li, Z. You, Y. Xia, and Q. Zhu, "A nonnegative latent factor model for large–scale sparse matrices in recommender systems via alternating direction method," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 3, pp. 579–592, Mar. 2016.



JIANYUN ZHANG is currently pursuing the M.Eng. degree with the Department of Computer Science and Technology, Jiangnan University, China. He is currently involved in machine learning algorithms for recommendation systems. His research interests include machine learning, data mining, and recommendation systems.

XIANLING LU received the B.S and M.S. degrees in computer science and application from the Nanjing University of Aeronautics and Astronautics (NUAA), in 1990 and 2004, respectively, and the Ph.D. degree from the Nanjing University of Science and Technology University (NUST), in 2008. He is currently a Professor with the Department of Computer Science and Technology, Jiangnan University, China. His research interests focus on deep learning, data mining, and wireless sensor network systems.