



TSCMF: Temporal and social collective matrix factorization model for recommender systems

Hamidreza Tahmasbi¹ · Mehrdad Jalali^{2,3}  · Hassan Shakeri²

Received: 10 December 2019 / Revised: 12 June 2020 / Accepted: 27 July 2020 /
Published online: 15 August 2020
© The Author(s) 2020

Abstract

In real-world recommender systems, user preferences are dynamic and typically change over time. Capturing the temporal dynamics of user preferences is essential to design an efficient personalized recommender system and has recently attracted significant attention. In this paper, we consider user preferences change individually over time. Moreover, based on the intuition that social influence can affect the users' preferences in a recommender system, we propose a Temporal and Social Collective Matrix Factorization model called TSCMF for recommendation. We jointly factorize the users' rating information and social trust information in a collective matrix factorization framework by introducing a joint objective function. We model user dynamics into this framework by learning a transition matrix of user preferences between two successive time periods for each individual user. We present an efficient optimization algorithm based on stochastic gradient descent for solving the objective function. The experiments on a real-world dataset illustrate that the proposed model outperforms the competitive methods. Moreover, the complexity analysis demonstrates that the proposed model can be scaled up to large datasets.

Keywords Social recommender system · Preference dynamics · Temporal dynamics · Collective matrix factorization

✉ Mehrdad Jalali
mehrdad.jalali@kit.edu

Hamidreza Tahmasbi
htahma@gmail.com

Hassan Shakeri
shakeri@mshdiau.ac.ir

¹ Department of Computer Engineering, Neyshabur Branch, Islamic Azad University, Neyshabur, Iran

² Department of Computer Engineering, Mashhad Branch, Islamic Azad University, Mashhad, Iran

³ Institute of Nanotechnology, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany

1 Introduction

Recommender systems are very useful tools in overcoming the information overload problem of users. These systems provide personalized recommendations to a user that he/she might like based on past preferences or observed behavior about one or various items. An essential problem in real-world recommender systems is that users are likely to change their preferences over time. A user's preference dynamics is known in the literature as temporal dynamics (Koren 2010) that may be caused by various reasons. According to Koren (2010), Rafailidis et al. (2017), and Lo et al. (2018), the most important of these reasons are: (i) **User experiences**: The past interaction of users and items make users like some items and dislike some others. For example, if a user is satisfied with the purchase on an auction website then he/she will probably continue buying from it in future. (ii) **New items**: The appearance of new items may change the focus of users. For example, users usually like to explore new items over time instead of interacting multiple times with the same items. (iii) **Social influence**: Friends' preferences may affect a user's decision and change the user preferences over time. (iv) **Item popularity**: Popular items may affect user interactions, regardless of his/her past preferences. For example, if there is a popular action movie but the user is interested in romantic films, the user may prefer to watch this action movie instead.

Modeling temporal dynamics of user preferences is essential to design a recommender system (Koren 2010; Shokeen and Rana 2018), as it leads to significant improvements in recommendation accuracy (Zafari et al. 2019; Rana and Jain 2015; Cheng et al. 2015). The need to model the dynamics of user preferences over time in recommender systems poses several essential challenging problems. First of all, because the amount of available data dramatically is reduced in a particular time period, the issue of data sparsity (Yusefi Hafshejani et al. 2018) in this situation is more intense (Lo et al. 2018). Moreover, based on the intuition that the time change pattern for each user may differ (Rafailidis and Nanopoulos 2016; Tang et al. 2015), how can the temporal information be incorporated to capture each individual user preference dynamics? Finally, what is the efficient approach to model the dynamics of user preferences in order to generate more accurate recommendations? For this purpose, in this paper, we present a Temporal and Social Collective Matrix Factorization model called TSCMF. The model captures the user preference dynamics based on collective matrix factorization (CMF)(Singh and Gordon 2008) framework to perform temporal recommendation. CMF is an extension of the MF which takes into account the side information, leading to more effective latent features. We take into account the user preferences can change individually over time, and based on the intuition that social influence can affect the users' preferences in a recommender system, we jointly factorize the users' rating matrix and social trust matrix via introducing a joint objective function. We adopt stochastic gradient descent (SGD) method and present an efficient optimization algorithm for solving the objective function. In our model, we assume that user preferences change smoothly (Lo et al. 2018; Tang et al. 2015) and the user preferences in the current time period depend on his/her preferences in the previous time period. Therefore, we introduce and learn a transition matrix of user preferences for each individual user to model user dynamics in two successive time periods into CMF. Experimental results on a real-world dataset, Epinions, illustrate that our proposed model outperforms the competitive methods. In addition, the complexity analysis implies that our model can be scaled up to large datasets.

The remainder of this paper is structured as follows. The next section presents the related works. Section 3 defines our problem and details our proposed model. Section 4 reports the experimental results. Finally, Section 5 provides the conclusions and future research directions.

2 Related work

Some studies on capturing the dynamics of user preferences in recommender systems are based on the computing user or item neighborhoods. These approaches generally boost recent ratings and penalize older ratings that possibly have less relevance at recommendation time, by employing time windows or a decay function (Vinagre 2012). For instance, in Su et al. (2015) and Liu et al. (2010), we see that it has given more weight to recently rated items and reduced the importance of past rated items gradually in rating prediction using an exponential time decay function. They consider that the preference dynamics are homogeneous for all users, whereas the changes in user preferences may be individual. A similar method was proposed in Cheng and Wang (2020), which takes into account that different users have different degrees of sensitivity to time. However, the primary challenge in these approaches is that it is hard to estimate an appropriate weighting scheme (Rabiu et al. 2020; Zhang 2015).

The broadly used technique to implement temporal recommender systems is matrix factorization (MF) (Yin et al. 2014). The MF technique has the advantage of relatively high accuracy and scalability (Lo et al. 2018; Yang et al. 2017). In this technique, each users and items is characterized by a series of features showing latent factors of the users and items in the system. It decomposes the matrix of users' ratings on items into two low-dimensional matrices, which directly profile users and items to the latent feature space, respectively, and these latent features are later used to make user behavior predictions. TimeSVD++ (Koren 2010) is the first MF-based popular method for modeling user preference dynamics. This model adopts the singular value decomposition (SVD) that is the most basic technique to matrix factorization (Yang et al. 2014). TimeSVD++ incorporates time-varying rating biases of each item and user into the MF. It assumes that older ratings are less important in rating prediction. The parameters of this method in different aspects and time periods must be learned individually, so it needs considerable effort for parameter tuning (Lo et al. 2018). A temporal MF method to capture the temporal dynamics in each of the individual user preferences was proposed in Lo et al. (2018). This model uses both rating information within the specific time period and overall rating information to learn the latent feature vector of each user at each time period by introducing a modified SGD algorithm. The method learns a linear model to extract the transition pattern for each user's latent feature vector using Lasso regression. An approach based on multi-task non-negative MF was presented in Ju et al. (2015) that uses a transition matrix to map between latent features of users in two successive time periods in order to track the temporal dynamics of user preferences. The transition matrix used in this method needs to be fixed, while in practice, this matrix is different for each user and each time period. A temporal MF (TMF) approach was proposed in Zhang et al. (2014) that captures the temporal dynamics of user preferences by designing a transition matrix for each user latent feature vectors between two successive time periods. Next, this approach is extended to a fully Bayesian treatment called BTMF by introducing priors for the hyperparameters to control the complexity and improve the accuracy of TMF. A dynamic MF based on collaborative Kalman filtering approach was proposed in Sun et al. (2014). This method extends the Gaussian probabilistic MF to capture user preference dynamics using a transition matrix of users' features based on a dynamical state space model. For learning model parameters from historical users' preferences, it exploits an expectation-maximization (EM) algorithm that uses Kalman filter in the expectation step of the EM. Despite the comprehensiveness of this method, the transition matrix used in it is homogeneous for all users. Moreover, the method is impractical for large datasets due to the run-time performance.

The aforementioned methods exploit only a single type of user-item interaction (users' rating information) without any side information. Exploiting the side information of users or items (Sun et al. 2019) beside the users' rating information can help to alleviate the data sparsity problem and thus provides users with better-personalized recommendations (Pan 2016). In this regard, a series of studies based on MF exploit the side information in temporal recommendation systems. A method based on MF was proposed in Wu et al. (2018) that fuses ratings, review texts, and the relationship between items by considering the temporal dynamics of user preferences to improve prediction results. The authors use TimeSVD++ as part of the model to capture temporal dynamics. However, the rating prediction for new users is difficult in this method. Moreover, this method assumes that the number of latent factors in ratings is equal to the number of hidden topics in reviews, while, as the authors point out, the number of latent factors is more than the number of hidden topics. CMF is an effective method that can be employed in recommender systems to simultaneously factorize multiple related matrices such as ratings and trust matrices. A temporal CMF method to generate the recommendations was proposed in Li and Fu (2017). This work jointly factorizes the multimodal user-item interactions to extract the user temporal pattern. The method introduces a transition matrix of users' preferences between two successive user latent feature matrices. Similarly, a dynamic CMF approach to predict the behavior of users was proposed in Rafailidis et al. (2017), which introduces a transition matrix of users' behaviors. This method models the temporal dynamics between purchase activity and click response behavior of users. It exploits the side information to alleviate the sparsity problem. The transition matrix used in these two last methods is homogeneous for all users; which is a major limitation of them.

Social trust information accumulated in social networks would be a rich source of information to address the aforementioned sparsity problem (Shokeen and Rana 2018), which has recently attracted the attention of many researchers into their recommendation models (Guo et al. 2016; Wu et al. 2016). A user is more likely to be affected by users whom he/she trusts. Therefore, the trust relations between users affect users' preferences. Although trust information is also very sparse, especially in a time period, it is complementary to rating information. Taking collective preferences and social trusts between users in a social recommendation system as additional input can be helpful in making more accurate and personalized recommendations (Bao et al. 2013). An SVD-based method was presented in Tong et al. (2019) that integrates rating, trust and time information to model user preference dynamics. This method includes time-variant biases for each item and each user. However, in this method, the feature vectors of users are not optimized with temporal information. In Aravkin et al. (2016), a framework was developed that incorporates trust relations into dynamic MF model to capture user preference dynamics. The method defines a transition matrix of users' preferences, assumes that trust relations among users are a graph at each time period, and considers a regularization term for dynamics that can incorporate known trust relations via the graph Laplacian. This method assumes that the preference dynamics are homogeneous for all users. In Liu et al. (2013), an approach was proposed in which heterogeneous user feedbacks as well as time and social networks are exploited for more accurate movie recommendation. It proposes a ranking-based MF model for combining both implicit and explicit user feedback, and extends the model to a sequential MF model for enabling time awareness parameterization. An approach based on social probabilistic MF was proposed in Bao et al. (2013) which exploits both temporal and social information to predict user preferences in micro-blogging. In this method, by employing an exponential time decay function, the users' latent features and the topics associated with previous latent

features are made. The method considers the importance of all previous time periods as well as the current as the same for all users and assigns the same weight to all users. However, in practice, the importance of previous time periods varies for each user.

Some studies exploit tensor factorization (TF) (Frolov and Oseledets 2017; Oh et al. 2019) to model user preference dynamics. In these studies, TF extends MF into a three-dimensional tensor through adding the temporal effects to the model. In Xiong et al. (2010), a movie recommendation method was proposed based on the Bayesian probabilistic TF. This method introduces a set of additional time features and adds constraints in the time dimension of the tensor to model the evolution of data over time. In Dunlavy et al. (2011) and Spiegel et al. (2011) were proposed the temporal link prediction methods based on TF. In Dunlavy et al. (2011), time-evolving bipartite graphs were employed and several methods were presented based on both matrix and tensor factorizations for predicting future links. In Spiegel et al. (2011), the importance of past user preferences using a smoothing factor was reduced. This method gives all user preferences the same weight at a specific time period whereas user preference dynamics of each user may vary individually (Rafailidis and Nanopoulos 2016). A temporal recommendation model based on the coupled TF was proposed in Rafailidis and Nanopoulos (2016). In this model, the importance of user past preferences is weighted based on a proposed user preference dynamics rate. The user demographics as side information are coupled with temporal interactions of users in this model. Despite the success of temporal recommendation methods based on TF, the processing and solving the tensor decomposition is hard (Lo et al. 2018) and usually leads to very high computing cost in practice (Zou et al. 2015), especially when the tensor is large and sparse (Lo et al. 2018).

Different from the aforementioned methods, in the present study, we model the temporal dynamics of user preferences by extending the CMF formulation to jointly factorize two matrices of user-item rating and social trust. Under the assumptions that the time change pattern for each user differs and that the user preferences change smoothly, we learn a transition matrix for each individual user to capture user dynamics in two successive time periods.

3 Proposed model

In this section, first we describe the problem definition and introduce the notations used throughout the paper. Then we present our TSCMF model.

3.1 Problem definition

Table 1 presents the important notations used throughout this paper.

Suppose we have a social recommender system including m users indexed from $i=1, 2, \dots, m$ and n items indexed from $j=1, 2, \dots, n$. We consider two types of information sources with timestamps including user-item ratings and social trusts between users. Given P predefined time periods indexed from $t=1, 2, \dots, P$, we define $R^{(t)} \in R^{m \times n}$ to be the user-item rating matrix in time period t , and $R_{ij}^{(t)}$ indicates the rating given by user i on item j in time period t . The ratings are normally integer values between 0 and R_{max} (eg., 0 to 5), where 0 denotes that the user has not rated that item in time period t . The higher rating means the better satisfaction. In practice, each user rates only a few items and thus $R^{(t)}$ is usually very sparse.

Table 1 Important notations

Symbols	Description
m, n, d, P	Number of users, items, latent features and time periods
$R^{(t)}$	User-item rating matrix in time period
$T^{(t)}$	User-user trust matrix in time period t
$T_{ik}^{(t)}$	Trust value between users i and k in time period t
$\hat{R}^{(t)}$	Prediction of through a prediction algorithm
$\hat{R}_{ij}^{(t)}$	Predicted rating of user i on item j in time period t
$U^{(t)}, V^{(t)}$	Latent feature matrix of users and items in time period t
$U_i^{(t)}, V_j^{(t)}$	Latent feature vector of user i and item j in time period t
$B^{(t)}, W^{(t)}$	Latent feature matrix of trusters and trustees in time period t
$B_i^{(t)}, W_k^{(t)}$	Latent feature vector of truster i and trustee k in time period t
$M_i^{(t)}$	Transition matrix between user latent feature vectors $U_i^{(t-1)}$ and $U_i^{(t)}$
I	Identity matrix
$I_{ij}^{R^{(t)}}$	Indicator function that takes 1 if user i rated item j in time period t , and 0 otherwise
$I_{ik}^{T^{(t)}}$	Indicator function that takes 1 if user k trusted by user i in time period t , and 0 otherwise
λ_T	Controller parameter of the impact of trust on users' preference
$\lambda, \lambda_1, \lambda_2$	Regularization parameters
η	Learning rate
R_{max}	Highest rating value
$\ \cdot \ _F$	Frobenius norm
N_R, N_T	Number of nonzero entries in $R^{(t)}$ and $T^{(t)}$

In social recommender systems, a user not only can rate items, he/she can also often specify other users as trusted friends. Let $T^{(t)} \in R^{m \times m}$ be the user-user trust matrix in time period t and $T_{ik}^{(t)} \in [0, 1]$ denotes that the extent user i trusts the user k in time period t . $T_{ik}^{(t)} = 1$ indicates the user i extremely trusts user k in time period t and $T_{ik}^{(t)} = 0$ donates the user i does not trust k in this time period.

Based on intuitions that users' preferences change individually over time and social influence can affect the users' preferences in a recommendation system, our goal is to provide a model to predict $R^{(t)}$ by capturing the preference dynamics for each individual user based on integrating the ratings and trust matrices.

3.2 Temporal and social collective matrix factorization (TSCMF)

In this section, we first formulate the objective function of our TSCMF model to capture the user preference dynamics based on CMF for performing temporal recommendation. We then devise an optimization algorithm for solving the objective function. Finally, we analyze the complexity of our model. Figure 1 shows the framework of the proposed model.

3.2.1 Objective function

As mentioned before, when a user is rating, the existing ratings of users whom he/she trusts will more likely affect his/her rating. Based on this intuition as well as considering temporal dynamics of user preferences, we present an approach to fuse users' rating matrix and social

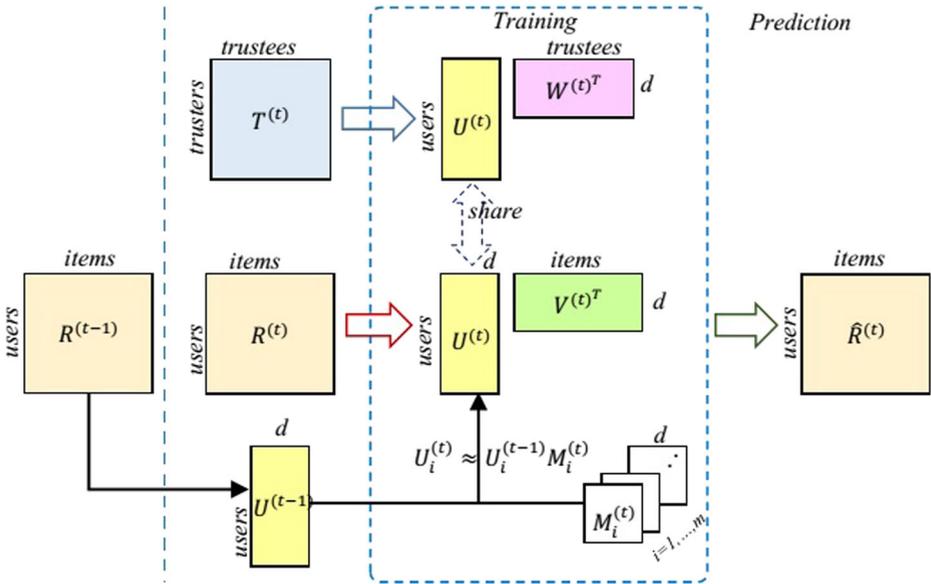


Fig. 1 The framework of the proposed TSCMF model

trust matrix under a CMF framework by considering the temporal information to model each individual user preference dynamics. The standard CMF ignores the temporal dynamics and it can only exploit all the previous data for model training. The old data may not be useful and may even have a negative impact on the making recommendations in the current time since the user preferences might change dramatically over a long period of time (Li and Fu 2017; De Pessemier et al. 2010). Therefore, unlike the standard CMF, we do not exploit the training data from all previous time periods. However, since users’ rating information at a time period is very sparse, the social trust information that we use beside the users’ rating information can alleviate the sparsity problem.

Suppose $U^{(t)} \in R^{m \times d}$ and $V^{(t)} \in R^{n \times d}$ be the latent feature matrices of users and items in time period t , respectively, with row vectors $U_i^{(t)}$ and $V_j^{(t)}$ indicating the d -dimensional latent feature vectors of user i and item j , in time period t , respectively (where $d \ll \min(m, n)$). MF learns the latent feature vectors of users and items in time period t from all known ratings in this time period and then can predict $\hat{R}_{ij}^{(t)}$ as approximation value of $R_{ij}^{(t)}$ by inner product of $U_i^{(t)}$ and $V_j^{(t)}$, i.e., $\hat{R}_{ij}^{(t)} = U_i^{(t)} V_j^{(t)T}$, where $V_j^{(t)T}$ is the transpose of $V_j^{(t)}$. Also, suppose $B^{(t)} \in R^{m \times d}$ and $W^{(t)} \in R^{m \times d}$ be the latent feature matrices of trusters and trustees in time period t , respectively, with row vectors $B_i^{(t)}$ and $W_k^{(t)}$ indicating the d -dimensional latent feature vectors of truster i and trustee k in time period t , respectively. MF learns these two vectors from existing trust relations in time period t , and then the trust value $T_{ik}^{(t)}$ can be predicted by inner product of $B_i^{(t)}$ and $W_k^{(t)}$, i.e., $B_i^{(t)} W_k^{(t)T}$, where $W_k^{(t)T}$ is the transpose of $W_k^{(t)}$.

Since the users in the rating matrix and the trusters in the trust matrix are the same Yang et al. (2017) and Guo et al. (2016), based on CMF, we jointly factorize these matrices by associating them through sharing a common user latent feature space. We consider the users

feature matrix $U^{(t)}$ as the latent space commonly shared by $R^{(t)}$ and $T^{(t)}$. Therefore, every vector $U_i^{(t)}$ simultaneously characterizes how the user i rates items and also how the same user trusts others in time period t . In addition, without loss of generality, similar to Yang et al. (2017), Yu et al. (2018), and Jamali and Ester (2010), we map the raw rating $R_{ij}^{(t)}$ to the interval $[0,1]$ by adopting the function $f(x) = x/R_{max}$. Also, we exploit the logistic function $g(x) = 1/(1 + exp(-x))$ to bound the inner product of latent feature vectors in the range of $[0,1]$. Thus, the objective function of CMF for time period t is as follows:

$$\begin{aligned} \min_{U^{(t)}, V^{(t)}, W^{(t)}} & \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^n I_{ij}^{R^{(t)}} (R_{ij}^{(t)} - g(U_i^{(t)} V_j^{(t)T}))^2 \\ & + \frac{\lambda_T}{2} \sum_{i=1}^m \sum_{k=1}^m I_{ik}^{T^{(t)}} (T_{ik}^{(t)} - g(U_i^{(t)} W_k^{(t)T}))^2 \\ & + \frac{\lambda}{2} (\|U^{(t)}\|_F^2 + \|V^{(t)}\|_F^2 + \|W^{(t)}\|_F^2) \end{aligned} \quad (1)$$

where the first two sum terms represent the approximation errors. $I_{ij}^{R^{(t)}}$ and $I_{ik}^{T^{(t)}}$ are indicator functions; $I_{ij}^{R^{(t)}}$ takes 1 if user i rated item j in time period t , and 0 otherwise. Also, $I_{ik}^{T^{(t)}}$ takes 1 if user i trusted the user k in time period t , and 0 otherwise. The parameter λ_T controls how much the user's trusters influence his/her preferences. The last three terms in (1) are regularizations to avoid overfitting. λ is the regularization parameter and $\|\cdot\|_F^2$ denotes the Frobenius norm with $\|R^{(t)}\|_F^2 = \sqrt{\sum_{i=1}^m \sum_{j=1}^n |R_{ij}^{(t)}|^2}$.

In practice, the user preferences change smoothly over time (Lo et al. 2018; Tang et al. 2015; Li and Fu 2017); therefore, the users' latent features should not significantly change in a short time period. Based on this intuition, we assume that the users' latent features in time period t ($t > 1$) have a temporal dependence to the users' latent features in time period $t-1$. We introduce a transition matrix $M_i^{(t)} \in R^{d \times d}$ between the user latent feature vectors $U_i^{(t-1)}$ and $U_i^{(t)}$ in two successive time periods $t-1$ and t for each user i . The transition matrix $M_i^{(t)}$ captures the mapping between the previous user latent feature vector $U_i^{(t-1)}$ and the current user latent feature vector $U_i^{(t)}$ for user i . We add the following temporal smoothness term in (1) to account the temporal dynamics in user preferences:

$$U_i^{(t)} \approx U_i^{(t-1)} M_i^{(t)}. \quad (2)$$

Therefore, we can rewrite the objective function in (1) as follows:

$$\begin{aligned} L = \min_{U^{(t)}, V^{(t)}, W^{(t)}} & \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^n I_{ij}^{R^{(t)}} (R_{ij}^{(t)} - g(U_i^{(t)} V_j^{(t)T}))^2 \\ & + \frac{\lambda_T}{2} \sum_{i=1}^m \sum_{k=1}^m I_{ik}^{T^{(t)}} (T_{ik}^{(t)} - g(U_i^{(t)} W_k^{(t)T}))^2 \\ & + \frac{\lambda_1}{2} \sum_{i=1}^m \|U_i^{(t)} - U_i^{(t-1)} M_i^{(t)}\|_F^2 \\ & + \frac{\lambda_2}{2} \left(\|U^{(t)}\|_F^2 + \|V^{(t)}\|_F^2 + \|W^{(t)}\|_F^2 + \sum_{i=1}^m \|M_i^{(t)}\|_F^2 \right) \end{aligned} \quad (3)$$

where the third term with respective regularization parameter λ_1 is the smoothness regularization based on the intuition that the user preferences should be smoothly changed over time. The last regularization term $\sum_{i=1}^m \|M_i^{(t)}\|_F^2$ is used to control the model complexity. λ_2 is the regularization parameter. We let $\lambda_1 = \lambda_2$ in our implementation for the sake of simplicity.

Choosing the proper length of the time period is critical to the performance of our model. We study its impact on recommendation accuracy in Section 4.7.

3.2.2 Optimization algorithm

The objective function L in (3) is not convex for all variables $U_i^{(t)}$, $V_j^{(t)}$, $W_k^{(t)}$, and $M_i^{(t)}$ simultaneously, but L is convex with respect to each variable separately. Therefore, we can obtain a local minimum of L using SGD method. The SGD has become very popular recently for using in non-convex optimization problems (Sidiropoulos et al. 2017). It usually has a very good convergence property (Li and Fu 2017). We update each variable by fixing the other variables. After selecting a pair of random entries $R_{ij}^{(t)}$ and $T_{ik}^{(t)}$, the variables $U_i^{(t)}$, $V_j^{(t)}$, $W_k^{(t)}$, and $M_i^{(t)}$ are updated as follows:

$$U_i^{(t)} = U_i^{(t)} - \eta \frac{\partial L}{\partial U_i^{(t)}} \tag{4}$$

$$V_j^{(t)} = V_j^{(t)} - \eta \frac{\partial L}{\partial V_j^{(t)}} \tag{5}$$

$$W_k^{(t)} = W_k^{(t)} - \eta \frac{\partial L}{\partial W_k^{(t)}} \tag{6}$$

$$M_i^{(t)} = M_i^{(t)} - \eta \frac{\partial L}{\partial M_i^{(t)}} \tag{7}$$

where η is the learning rate. We derive the gradients of L with respect to each variable as follows:

$$\begin{aligned} \frac{\partial L}{\partial U_i^{(t)}} &= \sum_{j=1}^n I_{ij}^{R^{(t)}} g' \left(U_i^{(t)} V_j^{(t)T} \right) \left(g \left(U_i^{(t)} V_j^{(t)T} \right) - R_{ij}^{(t)} \right) V_j^{(t)} \\ &+ \lambda_T \sum_{k=1}^m I_{ik}^{T^{(t)}} g' \left(U_i^{(t)} W_k^{(t)T} \right) \left(g \left(U_i^{(t)} W_k^{(t)T} \right) - T_{ik}^{(t)} \right) W_k^{(t)} \\ &+ \lambda_1 \left(U_i^{(t)} - U_i^{(t-1)} M_i^{(t)} \right) + \lambda_2 U_i^{(t)} \end{aligned} \tag{8}$$

$$\frac{\partial L}{\partial V_j^{(t)}} = \sum_{i=1}^m I_{ij}^{R^{(t)}} g' \left(U_i^{(t)} V_j^{(t)T} \right) \left(g \left(U_i^{(t)} V_j^{(t)T} \right) - R_{ij}^{(t)} \right) U_i^{(t)} + \lambda_2 V_j^{(t)} \tag{9}$$

$$\frac{\partial L}{\partial W_k^{(t)}} = \lambda_T \sum_{i=1}^m I_{ik}^{T^{(t)}} g' \left(U_i^{(t)} W_k^{(t)T} \right) \left(g \left(U_i^{(t)} W_k^{(t)T} \right) - T_{ik}^{(t)} \right) U_i^{(t)} + \lambda_2 W_k^{(t)} \tag{10}$$

$$\frac{\partial L}{\partial M_i^{(t)}} = \lambda_1 U_i^{(t-1)T} \left(U_i^{(t-1)} M_i^{(t)} - U_i^{(t)} \right) + \lambda_2 M_i^{(t)} \tag{11}$$

where $g'(x) = \exp(-x)/(1 + \exp(-x))^2$ is the derivative of the logistic function $g(x)$. The pseudocode of our proposed TSCMF model is presented in Algorithm 1. First, the

raw ratings $R_{ij}^{(t)}$ and $R^{(t-1)}$ are mapped to the interval $[0,1]$ in line 1. Then, in line 2, the transition matrix $M_i^{(t)}$ for each user i is initialized by setting $M_i^{(t)} = I$, where I is an $d \times d$ identity matrix. Also, in line 3 the latent feature matrices $U^{(t)}$, $V^{(t)}$ and $W^{(t)}$ are initialized with small random values. In line 4, we perform the MF on $R^{(t-1)}$ by applying LIBMF library (Chin et al. 2016) to compute the user latent feature matrix $U^{(t-1)}$. In our iterative optimization algorithm in lines 5-12, after selecting a pair of random entries $R_{ij}^{(t)}$ and $T_{ik}^{(t)}$, the variables $U_i^{(t)}$, $V_j^{(t)}$, $W_k^{(t)}$, and $M_i^{(t)}$ are updated using (4)–(7), respectively. In line 11, the objective function L in (3) is calculated based on updated variables. The algorithm repeats until L has converged or the maximum number of iterations has been reached. Convergence is achieved when the change of L between current and the previous iteration is greater than a predefined convergence threshold. In our implementation, we set the convergence threshold to 10^{-6} and the maximum number of iterations to 10^5 . Finally, in lines 13-15, the predicted rating matrix $\hat{R}^{(t)}$ as output of algorithm is computed.

Algorithm 1 The TSCMF algorithm.

Input:

matrices: $R^{(t-1)}$, $R^{(t)}$, $T^{(t)}$;
dimension size: d ;
parameters: λ_T , λ_1 , λ_2 ;
learning rate: η ;

Output:

The predicted rating matrix $\hat{R}^{(t)}$.

- 1 Map the raw ratings in $R^{(t)}$ and $R^{(t-1)}$ into $[0,1]$.
 - 2 Initialize $M_i^{(t)}$ for each user i .
 - 3 Initialize $U^{(t)}$, $V^{(t)}$ and $W^{(t)}$ with small random values.
 - 4 Compute $U^{(t-1)}$.
 - 5 **While** not reach convergence or the maximal iteration **do**
 - 6 Select a pair of random training entries $R_{ij}^{(t)} \in R^{(t)}$ and $T_{ik}^{(t)} \in T^{(t)}$.
 - 7 Update $U_i^{(t)}$ using (4).
 - 8 Update $V_j^{(t)}$ using (5).
 - 9 Update $W_k^{(t)}$ using (6).
 - 10 Update $M_i^{(t)}$ using (7).
 - 11 Compute L based on the updated $U_i^{(t)}$, $V_j^{(t)}$, $W_k^{(t)}$ and $M_i^{(t)}$ using (3).
 - 12 **end while**
 - 13 For (each user i and each item j)
 - 14 $\hat{R}_{ij}^{(t)} = g \left(U_i^{(t)} V_j^{(t)T} \right) R_{max}$.
 - 15 **end for**
-

3.2.3 Complexity analysis

The main computation cost of learning our model is to evaluate the objective function L and its gradients against variables. The computation complexity to evaluate the objective function L is $O(dN_R + dN_T)$, where N_R and N_T are the number of nonzero entries in matrices

$R^{(t)}$ and $T^{(t)}$, respectively. The number of latent features d is fixed. The computational complexities for calculating gradients $\frac{\partial L}{\partial U_i^{(t)}}$, $\frac{\partial L}{\partial V_j^{(t)}}$, $\frac{\partial L}{\partial W_k^{(t)}}$ and $\frac{\partial L}{\partial M_i^{(t)}}$ are $O(dN_R + dN_T)$, $O(dN_R)$, $O(dN_T)$ and $O(1)$, respectively. Therefore, the overall computational complexity for each iteration is $O(dN_R + dN_T)$, which is linear with respect to the number of nonzero entries in rating and trust matrices $R^{(t)}$ and $T^{(t)}$. Therefore, our model can be scaled to large datasets with millions of users and items.

4 Experiments

4.1 Dataset and evaluation methodology

Epinions¹ is a popular product review site by which the users can assign numerical ratings on a 1-5 scale and review the items. An item may be a product or service. In addition, Epinions provides a social network with trust relations where users can add other users to their trust networks. We conduct experiments on Epinions dataset (Tang 2019). This dataset contains rating information, social trust relations, and temporal information for both ratings and trust relations that make this dataset ideal for our experiments. The Epinions dataset used in our experiments contains 22166 users who have assigned ratings to at least one of a total of 296277 items. The total numbers of ratings and trust relations are 922267 and 300548 respectively. The rating data are from July 5, 1999 to May 8, 2011. The whole dataset was split into 11 time periods in chronological order. Since the temporal information about the trust relations before January 11, 2001 is not available, the first time period contains the data before January 11, 2001 and the last time periods covers data after January 11, 2010. Each of other time period contains data for one year. For example, the second time period contains data from January 12, 2001 to January 11, 2002.

We use time-dependent cross-validation based on increasing time window (Campos et al. 2014) as evaluation methodology. This method ensures that time dependencies between data are held in each train-test set pair. Based on this method, the data in each time period (except the first time period) are considered as the test set and all data prior to that time period as the training set. Therefore, we have 10 different train-test splits in total. Finally, the average results on test sets are reported. We use the threshold-based relevant item condition (Campos et al. 2014) to determine favorite items for each user. Based on this condition, the items in the user's test set rated higher than or equal to a threshold value are considered as favorite items. Accordingly, similar to Yang et al. (2017), we consider items in the user's test set with ratings higher or equal to 4 as his/her favorite items. We conduct all the experiments using MATLAB 2016a on Windows 10 PC with Intel Core i5 2.53 GHz with 8 GB memory.

4.2 Evaluation metrics

We adopt two most popular rating prediction evaluation metrics, i.e., Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) (Yang et al. 2014), to evaluate the rating accuracy of our proposed model in comparison with other methods. These metrics are defined as:

$$MAE = \frac{\sum_{(i,j) \in R_{test}} |r_{ij} - \hat{r}_{ij}|}{|R_{test}|} \quad (12)$$

¹www.epinions.com

$$RMSE = \sqrt{\frac{\sum_{(i,j) \in R_{test}} |r_{ij} - \hat{r}_{ij}|^2}{|R_{test}|}} \quad (13)$$

where R_{test} is the set of ratings in the test set, r_{ij} is the real rating of user i on item j , and \hat{r}_{ij} is the predicted rating of user i on item j . The lower the MAE and RMSE indicate better predictive accuracy.

In addition, we use the metrics Recall@K (R@K for short), Precision@K (P@K for short), and F1@K (Yang et al. 2017) to assess the quality of the top-K recommendations. These metrics are defined as:

$$R@K = \frac{1}{m} \sum_{i=1}^m \frac{|Rec_i \cap Fav_i|}{|Fav_i|} \quad (14)$$

$$P@K = \frac{1}{m} \sum_{i=1}^m \frac{|Rec_i \cap Fav_i|}{K} \quad (15)$$

$$F1@K = \frac{2 \times R@K \times P@K}{R@K + P@K} \quad (16)$$

where Fav_i is the set of favorite items of user i in the test set. Rec_i is the set of top-K recommended items for user i , which is generated by selecting the K items with the highest predicted ratings.

4.3 Comparison methods

We compare our TSCMF model with the following approaches:

- Probabilistic Matrix Factorization (PMF) (Salakhutdinov and Mnih 2008): This method is the baseline MF approach. It does not consider the temporal dynamics.
- Collective Matrix Factorization (CMF) (Singh and Gordon 2008): This method jointly factorizes two matrices that share one-side information and does not consider the temporal dynamics. We use the user-item rating and social trust matrices in this method. CMF is the basis of our proposed model.
- TimeSVD++ (Koren 2010): This method is a baseline for modeling the user preference dynamics. It incorporates the time-varying rating biases of each item and user into MF and generates the recommendations.
- Bayesian Temporal Matrix Factorization (BTMF) (Zhang et al. 2014): This is a Bayesian temporal MF approach that captures the temporal dynamics of user preferences by learning a transition matrix for each user latent feature vectors between two successive time periods.
- Dynamic Multi-Task Non-Negative Matrix Factorization (DMNMF) (Ju et al. 2015): This method models the user preference dynamics by fusing multi-task non-negative MF and a transition matrix of users' latent features.
- Temporal Matrix Factorization (TMF) (Lo et al. 2018): This method models the user preference dynamics by extracting a transition pattern for each user's latent feature vector.
- Dynamic Matrix Factorization with Social Influence (Aravkin et al. 2016): This method incorporates trust relations into dynamic MF model to capture user preference dynamics. It introduces a transition matrix of users' preferences and assumes that trust relations among users are a graph at each time period. To facilitate comparison, we refer this model as DMF.

- TimeTrustSVD (Tong et al. 2019): This method integrates rating, trust and time information. It adopts the time-variant biases for each item and each user into the model to capture temporal dynamics of user preferences.

The PMF, TimeSVD++, BTMF, DMNMF, and TMF methods exploit only the user-item rating matrix without any side information.

4.4 Parameter settings

The optimal parameters for each method are determined by cross-validation. Accordingly, we set the learning rate η to 0.001 in PMF and 0.003 in TimeSVD++, CMF, TMF, TimeTrustSVD, and TSCMF. We also set $\nu_0 = d$, $\beta_0 = 2$, $W_0 = Z_0 = I$, $\mu_0 = 0$ for BTMF, $\alpha = 0.6$ in CMF, $\lambda = 10^{-2}$ in DMF, $\lambda_T = 0.8$, and $\lambda_T = 5$ in TSCMF. For making a fair comparison, we fix the dimension of latent feature vectors to be 10 in all comparison methods. In addition, we set the regularization parameters to 0.001 in all our experiments.

4.5 Experimental results

Performance of the methods compared in terms of MAE and RMSE on the Epinions dataset is shown in Table 2. We observe that PMF performs worse than other methods. There are significant differences in terms of both MAE and RMSE between PMF and other methods. This is because PMF does not consider the temporal dynamics of user preferences and also does not exploit any side information such as trust relations.

The results show that the proposed TSCMF method has the best performance in terms of both MAE and RMSE among the compared methods. The improvements of TSCMF against competitive methods indicate that our model can significantly improve the accuracy of rating prediction.

Performance of the methods compared in terms of R@K, P@K, and F1@K (with K=5, 10) is shown in Table 3. We observe that the temporal methods achieve significantly higher R@K, P@K, and F1@K than PMF. This implies that considering temporal dynamics of user preferences is useful for improving the recommendations. From Table 3, we can see that the proposed TSCMF has the best performance in terms of R@K, P@K, and F1@K among the compared methods for both values of K. In comparison with the other temporal competitors, the results indicate that our TSCMF method can better capture the temporal dynamics of user preferences. We believe that the transition matrix introduced in our model is a key element that contributes to this improvement. Compared to the transition matrix used in DMNMF, BTMF, TMF, and DMF, this matrix is dynamic and is trained individually for each user.

Table 2 The performance of comparative methods in terms of MAE and RMSE. The boldface numbers highlight the best results in each metric, and the row ‘Improve’ presents the improvement percentage that TSCMF gains relative to respective competitors

Metric	PMF	CMF	TimeSVD++	BTMF	DMNMF	TMF	DMF	TimeTrustSVD	TSCMF
MAE	1.2982	1.0344	1.1749	0.9722	1.1493	1.1036	1.033	1.0167	0.9102
(Improve)	29.89%	12.01%	22.53%	6.38%	20.8%	17.52%	11.89%	10.48%	–
RMSE	1.4221	1.2567	1.3377	1.1941	1.3096	1.3119	1.2557	1.2542	1.126
(Improve)	20.82%	10.4%	15.83%	5.7%	14.02%	14.17%	10.33%	10.22%	–

Table 3 The performance of comparative methods in terms of R@K, P@K and F1@K (with K=5, 10). The boldface numbers highlight the best results in each metric, and the row 'Improve' presents the improvement percentage that TSCMF gains relative to respective competitors

Metric	PMF	CMF	TimeSVD++	BTMF	DMNMF	TMF	DMF	TimeTrustSVD	TSCMF
R@5	0.3782	0.4814	0.4036	0.5104	0.4391	0.4229	0.4975	0.507	0.5632
(Improve)	48.92%	16.99%	39.54%	10.34%	28.26%	33.18%	13.21%	11.08%	–
P@5	0.4416	0.5915	0.4711	0.5714	0.5097	0.4893	0.5796	0.5801	0.6413
(Improve)	45.22%	8.42%	36.13%	12.23%	25.82%	31.06%	10.65%	10.55%	–
F1@5	0.4074	0.5308	0.4347	0.5392	0.4718	0.4537	0.5354	0.5411	0.5997
(Improve)	47.2%	12.98%	37.96%	11.22%	27.11%	32.18%	12.01%	10.83%	–
R@10	0.4019	0.4987	0.4511	0.5831	0.4796	0.4733	0.5234	0.5377	0.6106
(Improve)	51.93%	22.44%	35.36%	4.72%	27.31%	29.01%	16.66%	13.56%	–
P@10	0.4349	0.5106	0.4611	0.5396	0.4995	0.4829	0.5129	0.5198	0.5937
(Improve)	36.51%	16.27%	28.76%	10.03%	18.86%	22.94%	15.75%	14.22%	–
F1@10	0.4177	0.5046	0.456	0.5605	0.4893	0.4781	0.5181	0.5286	0.602
(Improve)	44.12%	19.3%	32.02%	7.4%	23.03%	25.92%	16.19%	13.89%	–

From Tables 2 and 3, we can observe that CMF, which uses both ratings and trust information and does not consider the temporal dynamics, outperforms the temporal methods TimeSVD++, DMNMF, and TMF. This finding indicates that regardless of temporal information, incorporating the social trust relations is effective in improving recommendation accuracy. On the other hand, we can see that the temporal methods that use both ratings and trust information (i.e., DMF, TimeTrustSVD, and our TSCMF method), perform better than CMF. The better results obtained for these methods than CMF imply that temporal dynamic and trust relations could be complementary to each other in boosting the accuracy of recommendations. The superiority of TSCMF over competitive methods indicates that the latent features learned from the previous time period are helpful. Also, capturing the dynamics of user preferences in our model, regarding the fact that the user preferences change individually over time, improves the recommendation accuracy.

In order to evaluate the efficiency of our TSCMF model, we compare the running time of our model with other methods. Table 4 reports the experimental results, in seconds. We can see that PMF has the lowest running time. This is because PMF only exploits ratings to learn latent features and also does not consider the temporal dynamics. The trust-based recommendation methods CMF, DMF, and TimeTrustSVD have a higher running time than other methods, which is mainly due to the use of trust information in these methods. Among trust-based methods, our TSCMF method outperforms other methods in terms of the running time. Also, compared to methods that do not exploit any trust information, the running time of TSCMF is lower than BTMF and DMNMF. The main reason is that TSCMF only uses the data of the previous time period to learn latent features, thus the running time is reduced.

Table 4 Efficiency comparison (s)

	PMF	CMF	TimeSVD++	BTMF	DMNMF	TMF	DMF	TimeTrustSVD	TSCMF
Time	184	924	649	736	801	668	993	1015	708

We notice that from Table 2 the percentage of relative improvements of TSCMF is close to BTMF (around 6.38% in MAE). Comparing the results of Tables 2 and 3, we observe that when the performance is improved from 0.9722 to 0.9102 with respect to MAE, it achieves more than 10 percent relative improvement in precision. Since the running time of the proposed TSCMF method is lower than BTMF, this amount of improvement in the quality of the recommendations can be valuable.

4.6 Impact of parameter λ_T

The parameter λ_T plays an important role in our TSCMF model via controlling the impact of social trust on user's preferences. The larger values of λ_T indicate more influence of the social trust information on users' preference. To assess how different values of λ_T affect the final recommendation accuracy, we set λ_T to be 0.1, 0.5, 1, 2, 5, 10, and 20 in our model. We perform this assessment for each of the 10 train-test splits. Figure 2 presents the average MAE and RMSE of our model with different values of λ_T . As can be seen, λ_T affects the recommendation results dramatically, suggesting that fusing the users' rating matrix and social trust matrix can help to improve the recommendation accuracy. As λ_T increases, the average MAE and RMSE values decrease at first, indicating that the recommendation accuracy increases. In comparison, when λ_T exceeds a certain threshold, the average MAE and RMSE values increase.

These findings demonstrate that merely exploiting the user-item rating matrix or merely exploiting the social trust information cannot generate better results than appropriately fusing these two resources together in our model. As shown in Fig. 2, TSCMF has its best results for $\lambda_T = 5$.

4.7 Impact of the length of the time period

Choosing the optimal length of the time period is critical in temporal models (Li and Fu 2017), and usually depends on the application of the recommender system (Rafailidis and Nanopoulos 2016). For example, in a news recommender system, users' preferences in specific news topics may take only a few days, while in a movie recommendation, users' preferences in movies may change slowly over time. Therefore, choosing a shorter time period may be appropriate for capturing users' preferences in news than movies (Sahoo et al. 2012). In such a situation, choosing too long time period may lead to miss any change in the behavior of users within that time period.

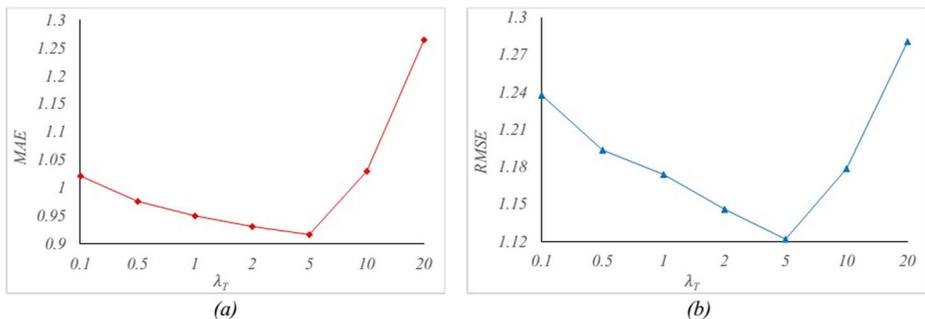


Fig. 2 Impact of different values of $\lambda_T = 5$ **a** MAE and **b** RMSE

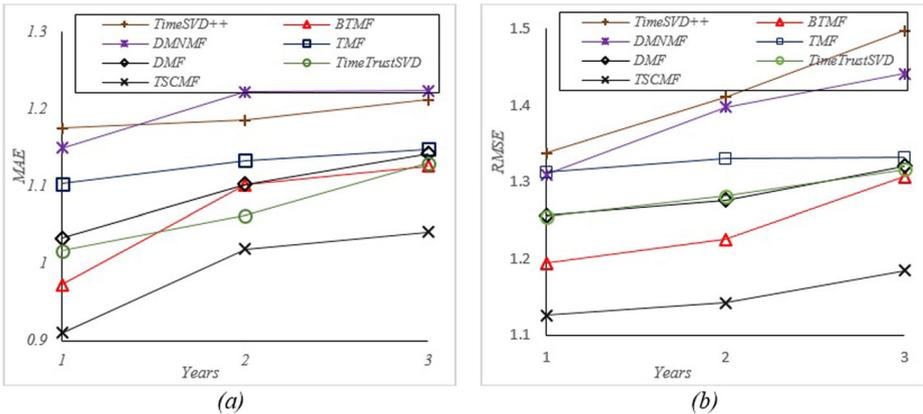


Fig. 3 Impact of different length of the time period **a** MAE and **b** RMSE

In order to study of the effect of the time period length on the methods' performance, we only consider the methods that incorporate the temporal dynamics of user preferences into models. Since the trust information in Epinions dataset used in our experiments is available as annually, the shortest time period length that we select is 1 year. Figure 3 shows the performance of three different lengths of the time period in terms of average MAE and RMSE. From this figure, we see that all methods gain their best results when the length of the time period is set to 1 year. With increasing the length of the time period, the performance of all methods decreases. Another interesting finding in this regard is that for all three examined time period lengths, our model outperforms the other compared methods.

5 Conclusion

In this paper, we proposed the Temporal and Social Collective Matrix Factorization (TSCMF) model to capture the temporal dynamics of user preferences for implementing temporal recommendation. We jointly factorized the users' rating information and social trust information in a collective matrix factorization framework by introducing a joint objective function. We assumed that the user preferences in current time period have a temporal dependence to user preferences in the previous time period and model user dynamics into the collective matrix factorization framework by learning a transition matrix of user preferences between two successive time periods for each individual user. We presented an efficient optimization algorithm by adopting stochastic gradient descent method for solving the objective function. The experiments on a real-world dataset collected from a popular product review website, i.e., Epinions, show that our proposed model outperforms the other compared methods. In addition, the proposed model can be scaled to large datasets with millions of users and items. Our findings strengthen the idea that modeling the dynamics of user preferences based on the fact that the changes in user preferences vary individually leads to improvements in recommendation accuracy and, consequently, user satisfaction. In addition, considering temporal dynamic and trust relations could be complementary to each other to the development of social recommender systems.

The proposed method can help to improve the quality of social recommender systems. However, in some social recommender systems, trust information is not explicitly available.

For future work, we plan to extract implicit trust based on users' interactions whenever explicit trust is not available and use in our model. We also want to extend the model to address the problem of cold-start users who do not have any rating and any trust relation in both previous and current time periods (named as new users). One possible approach to deal with this problem is exploiting additional side information such as users' attributes. In some social recommender systems, users can express distrust toward other users. Additionally, we want to exploit distrust relations among users with temporal information in addition to trust relations in our model for generating better-personalized recommendations.

Compliance with Ethical Standards

Conflict of interests The authors declare that they have no conflict of interest.

Funding Information Open Access funding provided by Projekt DEAL.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Aravkin, A. Y., Varshney, K. R., Yang, L. (2016). Dynamic matrix factorization with social influence. In *2016 IEEE International Workshop on Machine Learning for Signal Processing* (pp. 1–6): IEEE.
- Bao, H., Li, Q., Liao, S. S., et al. (2013). A new temporal and social PMF-based method to predict users' interests in micro-blogging. *Decision Support Systems*, *55*, 698–709. <https://doi.org/10.1016/j.dss.2013.02.007>.
- Campos, P. G., Díez, F., Cantador, I. (2014). Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols. *User Modeling and User-Adapted Interaction*, *24*, 67–119. <https://doi.org/10.1007/s11257-012-9136-x>.
- Cheng, J., Liu, Y., Zhang, H., et al. (2015). A new recommendation algorithm based on user's dynamic information in complex social network. *Mathematical Problems in Engineering* 2015. <https://doi.org/10.1155/2015/281629>.
- Cheng, S., & Wang, W. (2020). Rating prediction algorithm based on user time-sensitivity. *Information*, *11*, 4.
- Chin, W.-S., Yuan, B.-W., Yang, M.-Y., et al. (2016). LIBMF: A library for parallel matrix factorization in shared-memory systems. *The Journal of Machine Learning Research*, *17*, 2971–2975.
- De Pessemier, T., Dooms, S., Deryckere, T., Martens, L. (2010). Time dependency of data quality for collaborative filtering algorithms. In *Proceedings of the Fourth ACM Conference on Recommender systems* (pp. 281–284).
- Dunlavy, D. M., Kolda, T. G., Acar, E. (2011). Temporal link prediction using matrix and tensor factorizations. *ACM Transactions on Knowledge Discovery from Data*, *5*, 1–27. <https://doi.org/10.1145/1921632.1921636>.
- Frolov, E., & Oseledets, I. (2017). Tensor methods and recommender systems. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *7*, 1–41. <https://doi.org/10.1002/widm.1201>.
- Guo, G., Zhang, J., Yorke-Smith, N. (2016). A novel recommendation model regularized with user trust and item ratings. *IEEE Transactions on Knowledge and Data Engineering*, *28*, 1607–1620. <https://doi.org/10.1109/TKDE.2016.2528249>.
- Jamali, M., & Ester, M. (2010). A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the Fourth ACM conference on Recommender systems* (pp. 135–142): ACM.

- Ju, B., Qian, Y., Ye, M., et al. (2015). Using dynamic multi-task non-negative matrix factorization to detect the evolution of user preferences in collaborative filtering. *PloS One*, *10*, 1–20. <https://doi.org/10.1371/journal.pone.0135090>.
- Koren, Y. (2010). Collaborative filtering with temporal dynamics. *Communications of the ACM*, *53*, 89–97. <https://doi.org/10.1145/1721654.1721677>.
- Li, S., & Fu, Y. (2017). Robust representations for response prediction. In *Robust Representation for Data Analytics* (pp. 147–174): Springer.
- Liu, N.N., He, L., Zhao, M. (2013). Social temporal collaborative ranking for context aware movie recommendation. *ACM Transactions on Intelligent Systems and Technology*, *4*, 15:1–15:26. <https://doi.org/10.1145/2414425.2414440>.
- Liu, N.N., Zhao, M., Xiang, E., Yang, Q. (2010). Online evolutionary collaborative filtering. In *Proceedings of the fourth ACM conference on Recommender systems* (pp. 95–102).
- Lo, Y.-Y., Liao, W., Chang, C.-S., Lee, Y.-C. (2018). Temporal matrix factorization for tracking concept drift in individual user preferences. *IEEE Transactions on Computational Social Systems*, *5*, 156–168. <https://doi.org/10.1109/TCSS.2017.2772295>.
- Oh, B., Shin, S., Eom, S., et al. (2019). Reliable TF-based recommender system for capturing complex correlations among contexts. *Journal of Intelligent Information Systems*, *52*, 337–365. <https://doi.org/10.1007/s10844-018-0514-7>.
- Pan, W. (2016). A survey of transfer learning for collaborative recommendation with auxiliary data. *Neurocomputing*, *177*, 447–453. <https://doi.org/10.1016/j.neucom.2015.11.059>.
- Rabiu, I., Salim, N., Da'u, A., Osman, A. (2020). Recommender system based on temporal models: A systematic review, (Vol. 10).
- Rafailidis, D., Kefalas, P., Manolopoulos, Y. (2017). Preference dynamics with multimodal user-item interactions in social media recommendation. *Expert Systems with Applications*, *74*, 11–18.
- Rafailidis, D., & Nanopoulos, A. (2016). Modeling users preference dynamics and side information in recommender systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *46*, 782–792. <https://doi.org/10.1109/TSMC.2015.2460691>.
- Rana, C., & Jain, S.K. (2015). A study of the dynamic features of recommender systems. *Artificial Intelligence Review*, *43*, 141–153. <https://doi.org/10.1007/s10462-012-9359-6>.
- Sahoo, N., Tepper, D.A., Mukhopadhyay, T. (2012). A hidden markov model for collaborative filtering. *MIS Quarterly*, *36*, 1329–1356.
- Salakhutdinov, R., & Mnih, A. (2008). Probabilistic matrix factorization. In *Advances in neural information processing systems* (pp. 1257–1264).
- Shokeen, J., & Rana, C. (2018). A review on the dynamics of social recommender systems. *International Journal of Web Engineering and Technology*, *13*, 255–276. <https://doi.org/10.1504/IJWET.2018.095184>.
- Sidiropoulos, N.D., De Lathauwer, L., Fu, X., et al. (2017). Tensor decomposition for signal processing and machine learning. *IEEE Transactions on Signal Processing*, *65*, 3551–3582. <https://doi.org/10.1109/TSP.2017.2690524>.
- Singh, A.P., & Gordon, G.J. (2008). Relational learning via collective matrix factorization. *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD*, *08*, 650–658. <https://doi.org/10.1145/1401890.1401969>.
- Spiegel, S., Clausen, J., Albayrak, S., Kunegis, J. (2011). Link prediction on evolving data using tensor factorization. In *New frontiers in applied data mining* (pp. 100–110). Berlin: Springer.
- Su, H., Lin, X., Yan, B., Zheng, H. (2015). The collaborative filtering algorithm with time weight based on mapreduce, Springer, Cham.
- Sun, Z., Guo, Q., Yang, J., et al. (2019). Research commentary on recommendations with side information: a survey and research directions. *Electronic Commerce Research and Applications* 100879, <https://doi.org/10.1016/j.elerap.2019.100879>.
- Sun, J.Z., Parthasarathy, D., Varshney, K.R. (2014). Collaborative kalman filtering for dynamic matrix factorization. *IEEE Transactions on Signal Processing*, *62*, 3499–3509.
- Tang, J. (2019). Epinions Dataset. <http://www.cse.msu.edu/tangjili/trust.html>. Accessed 5 Jan 2019.
- Tang, J., Gao, H., Das, S.A., et al. (2015). Trust evolution: Modeling and its applications. *IEEE Transactions on Knowledge and Data Engineering*, *27*, 1724–1738. <https://doi.org/10.1109/TKDE.2014.2382576>.
- Tong, C., Qi, J., Lian, Y., et al. (2019). TimetrustSVD: A collaborative filtering model integrating time, trust and rating information. *Future Generation Computer Systems*, *93*, 933–941. <https://doi.org/10.1016/j.future.2017.07.037>.
- Vinagre, J. (2012). Time-aware collaborative filtering: a review. *Doctoral Symposium in Informatics Engineering*, pp 43.

- Wu, T., Feng, Y., Sang, J., et al. (2018). A novel recommendation algorithm incorporating temporal dynamics, reviews and item correlation. *IEICE Transactions on Information and Systems*, 101, 2027–2034. <https://doi.org/10.1587/transinf.2017EDP7387>.
- Wu, H., Yue, K., Pei, Y., et al. (2016). Collaborative topic regression with social trust ensemble for recommendation in social media systems. *Knowledge-Based Systems*, 97, 111–122. <https://doi.org/10.1016/j.knsys.2016.01.011>.
- Xiong, L., Chen, X., Huang, T.-K., et al. (2010). Temporal collaborative filtering with bayesian probabilistic tensor factorization. In *Proceedings of the 2010 SIAM International Conference on Data Mining* (pp. 211–222).
- Yang, X., Guo, Y., Liu, Y., Steck, H. (2014). A survey of collaborative filtering based social recommender systems. *Computer Communications*, 41, 1–10. <https://doi.org/10.1016/j.comcom.2013.06.009>.
- Yang, B., Lei, Y., Liu, J., Li, W. (2017). Social collaborative filtering by trust. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39, 1633–1647. <https://doi.org/10.1109/TPAMI.2016.2605085>.
- Yin, H., Cui, B., Chen, L., et al. (2014). A temporal context-aware model for user behavior modeling in social media systems. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data* (pp. 1543–1554).
- Yu, Y., Gao, Y., Wang, H., Wang, R. (2018). Joint user knowledge and matrix factorization for recommender systems. *World Wide Web*, 21, 1141–1163. <https://doi.org/10.1007/s11280-017-0476-7>.
- Yusefi Hafshejani, Z., Kaedi, M., Fatemi, A. (2018). Improving sparsity and new user problems in collaborative filtering by clustering the personality factors. *Electronic Commerce Research*, 18, 813–836. <https://doi.org/10.1007/s10660-018-9287-x>.
- Zafari, F., Moser, I., Baarslag, T. (2019). Modelling and analysis of temporal preference drifts using a component-based factorised latent approach. *Expert Systems with Applications*, 116, 186–208. <https://doi.org/10.1016/j.eswa.2018.09.010>.
- Zhang, C. (2015). Improving recommender systems with rich side information. Dissertation, Simon Fraser University.
- Zhang, C., Wang, K., Yu, H., et al. (2014). Latent factor transition for dynamic collaborative filtering. In *Proceedings of the 2014 SIAM International Conference on Data Mining* (pp. 452–460).
- Zou, B., Li, C., Tan, L., Chen, H. (2015). GPU-TENSOR: Efficient tensor factorization for context-aware recommendations. *Information Sciences*, 299, 159–177.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.