IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 10, 2018, accepted December 30, 2018, date of publication January 11, 2019, date of current version February 6, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2892289

Movie Recommendation via Markovian Factorization of Matrix Processes

RICHONG ZHANG[1], (Member, IEEE), AND YONGYI MAO[2], (Member, IEEE)
[1]BDBC and SKLSDE, School of Computer Science and Engineering, Beihang University, Beihang 100191, China
[2]School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, ON KN56N2, Canada

Corresponding author: Richong Zhang (zhangrc@act.buaa.edu.cn)

This work was supported in part by the China 973 Program under Grant 2015CB358700, in part by the National Natural Science Foundation of China under Grant 61772059 and Grant 61421003, and in part by the Beijing Advanced Innovation Center for Big Data and Brain Computing and the State Key Lab of Software Development Environment.

ABSTRACT The success of the probabilistic matrix factorization (PMF) model has inspired the rapid development of collaborative filtering algorithms, among which timeSVD++ has demonstrated great performance advantage in solving the movie rating prediction problem. Allowing the model to evolve over time, timeSVD++ accounts for ''concept drift'' in collaborative filtering by heuristically modifying the quadratic optimization problem derived from the PMF model. As such, timeSVD++ no longer carries any probabilistic interpretation. This lack of frameworks makes the generalization of timeSVD++ to other collaborative filtering problems rather difficult. This paper presents a new model family termed Markovian factorization of matrix process (MFMP). On one hand, MFMP models, such as timeSVD++, are capable of capturing the temporal dynamics in the dataset, and on the other hand, they also have clean probabilistic formulations, allowing them to adapt to a wide spectrum of collaborative filtering problems. Two simple example models in this family are introduced for the prediction of movie ratings using time-stamped rating data. The experimental study using MovieLens dataset demonstrates that the two models, although simple and primitive, already have comparable or even better performance than timeSVD++ and a standard tensor factorization model.

INDEX TERMS Recommender system, collaborative filtering, matrix factorization.

I. INTRODUCTION

Stimulated by great commercial efforts, collaborative filtering [1], [2] has attracted intense attention in the research community. A typical collaborative filtering problem is the rating prediction problem prescribed by a dataset consisting of N users, M movies, and a collection of user ratings R_{ij}'s each given by some user i on some item j. In reality, most item have only been rated by a fraction of the users and most users have only rated a fraction of the movies. As a consequence, when organizing the observed rating R_{ij}'s in an $M \times N$ matrix R, a large fraction of the matrix entries are missing. The objective of collaborative filtering in this setting is to predict the missing entries of R based on the observed ratings.

Among various solutions to such a ''matrix completion'' problem, the model of Probabilistic Matrix Factorization (PMF) [3] is arguably on of the most impactful, as SVD++ model by Koren et al. and BRISMF model by Takacs. Briefly, PMF assumes that there is a latent space of dimension D, a number much smaller than M and N, from which each user i is associated with a user feature vector U_i and each movie is associated with a movie feature vector V_j. The rating of movie j given by user i is then modeled as the inner product of the two vectors, $U_i^T V_j$, subject to additive Gaussian noise. Imposing Gaussian priors on each U_i and each V_j, the rating prediction problem is translated to the maximum *a posteriori* (MAP) inference of U_i's and V_j's, which reduces nicely to a minimization problem with quadratic objective function.

Simple and elegant as it is, PMF appears surprisingly effective in practice and has since been applied to many application domains, which include to generate annotations for images [4], to recommend products for social network users [4], and to recommend programs for IPTV users [5]. Inspired by its great success, various extensions have been made to PMF models to date (see, e.g., [6]–[13]), among which two directions are particularly noteworthy. One direction applies a probabilistic framework similar to PMF for completing higher-dimensional data arrays, or tensors; this has led to various ''tensor factorization'' models

(see, e.g., [7]–[10]). The other direction discards the assumption that the PMF model is static across time and allows the model to evolve over time in order to capture the dynamics of ''concept drift'' [14] in collaborative filtering; this has led to the timeSVD++ algorithm for collaborative filtering [11].

In the context of predicting movie ratings, when the timestamps of the collected ratings are available, the dataset presents itself as a three-dimensional array, or a three-mode tensor [8]. In this case, both tensor factorization models and timeSVD++ can be used as solvers for the rating prediction (or ''tensor completion'') problem. In fact, depending on the dataset and its intrinsic temporal dynamics, the two classes of solvers may indeed be close competitors against each other. One advantage of tensor factorization models is their clean and well-principled probabilistic formulations, along a line very similar to that of PMF. These models are however too general: they are not specifically designed to capture concept drift over time and they assume no correlation or dependency structure specific for *temporal* dynamics. On the other hand, the timeSVD++ algorithm is specifically crafted for capturing concept drift in rating prediction problems. The downside of timeSVD++ is however that it is constructed based on a collection of heuristics and insights about the dataset. More specifically, instead of formulating the problem in the probabilistic domain, timeSVD++ directly modifies the quadratic objective function in PMF by adding various bias terms heuristically and allowing the latent feature vectors and bias terms to drift over time according to certain choices of heuristics. Although timeSVD++ is among the most celebrated algorithms in solving the movie rating prediction problem, a machine learning researcher, if he is so inquisitive, is perhaps left wondering: To what extent is timeSVD++ still effective for completing (time-stamped) matrices of different kinds of temporal dynamics? When applying timeSVD++ to other problems, is there any guiding principle in choosing the parameters for the algorithm beyond trial and error?

As timeSVD++ is more a problem-specific model than a general modeling or algorithmic framework, the answers to such questions are likely pessimistic. In the context of completing ''time-stamped'' matrices, the thrust of this research is to develop general models and algorithms for that bring together the benefit of timeSVD++ and that of probabilistic frameworks. More precisely, we would like to develop a family of models or a modeling methodology which captures concept drift in collaborative filtering and which also has a probabilistic formulation. The model family we present in this paper is termed Markovian Factorization of Matrix Process (MFMP). Using the movie rating prediction problem as a running example throughout the paper, the methodology of this work models the rating matrix R as a *matrix-valued* random process $\{R(t)\}$. We associate with each user i a user feature process $\{U_i(t)\}$ and with each movie j a movie feature process $\{V_j(t)\}$. The (i, j) entry of matrix process $\{R(t)\}$, namely random process $\{R_{ij}(t)\}$, is then modeled as the ''inner-product process'' $\{U_i(t)^T V_j(t)\}$ subject to probabilistic perturbation. Markovian structures are imposed on the latent processes (namely, all $\{U_i(t)\}$'s and all $\{V_j(t)\}$'s) in order to capture the temporal dynamics of $\{R(t)\}$. These assumptions define the family of MFMP models.

To be more concrete, we present two simple examples in this model family, which we call the first-order MFMP and the second-order MFMP. The two models assume the latent processes to be respectively first-order and second order Gaussian Markov processes and probabilistic perturbation on the inner-product processes is also assumed to be Gaussian, taking effect additively. We show that the MAP inference of the latent processes in both models reduces to the minimization of certain quadratic functions, similar to the case of PMF. Gradient descent solvers for the minimization problems then solve the MAP inference problems. Using time-stamped movie rating dataset from MovieLens, we show that the algorithms developed from both models already have comparable or even better performance than timeSVD++ and a standard tensor factorization algorithm.

Demonstrating the competitive performance of the proposed model examples is only the secondary purpose of this work, as the model examples presented in this paper are still quite simple and primitive. Our main objective is in fact advocating our methodology of modeling temporal dynamics in collaborative filtering, namely, factoring a matrix process $\{R(t)\}$ into the ''product'' of two latent matrix Markov processes $\{U(t)\}$ and $\{V(t)\}$. For that purpose, we also point to various directions along which more sophisticated MFMP models can be constructed.

Using Markov processes or Hidden Markov Models [15] to model time series is not at all a new idea. In fact, such an approach has prevailed in the area of statistical signal processing for the past decades. In the area of collaborative filtering, a Hidden Markov Model was also developed for interpreting users' blog-reading behavior and making article recommendations [16]. In that work, the proposed model contains only one latent first-order non-Gaussian Markov process and the problem studied differs from the rating prediction problem we consider here. Nevertheless, the model thereof may be considered as a special case of the MFMP model family. The success of their model and inference algorithm also manifests, to a degree, the usefulness of the MFMP model family presented in this paper.

To summarize, the main contribution of this paper is:

- This paper presents a new model family termed Markovian Factorization of Matrix Process (MFMP).This model family are capable of capturing the temporal dynamics in the dataset. In addition, it also has clean probabilistic formulations, allowing them to be easily adapted by a wide spectrum of collaborative filtering problems.
- To demonstrate the applicability of the proposed framework, we design two concrete example models in this family for the prediction of movie ratings using time-stamped rating data.
- Empirical studies on real data demonstrate the effectiveness of the two proposed model, although simple

and primitive, already have comparable or even better performance than timeSVD++ and a standard tensor factorization model.

This paper is organized as follows. In Section II, the most recent development of collaborative filtering algorithms, their extensions and the user dynamics modeling approaches are investigated. In Section III, our proposed model family, Markovian Factorization of Matrix Process is presented. Also, we propose two simple example of the MFMP to model the dynamics of user interests and to generated recommendations. In Section IV, we present experimental studies. The paper is briefly concluded in Section V.

II. RELATED WORKS

A. RECOMMENDER SYSTEM

The general goal of recommender systems is to assist potential buyers in discovering items, such as products or information. Collaborative filtering has been successfully exploited by many systems to predict the ratings by aggregating the similar experiences. Traditional recommendation models, such as item-based [1], user-based [17] and hybrid algorithms [18] all have been shown the capability of providing higher quality recommendations in various domains [19], [20]. With the increasing number of users and items emerged, the scalability [21], [22], the efficiency [23], [24] and the stability [25] have been studied to extend the traditional recommendation algorithms to meet the huge computation requirement.

However, due to the sparsity of the user-rating matrix, directly calculating the similarity cannot always generate an effective results to measure the distance between users and items. As a result, researchers propose model-based collaborative filtering approaches [3], [6], [26], [27] to incorporate advanced machine learning techniques to characterizing the user interests and item properties and generate more precise recommendations.

Matrix factorization or SVD is one of such model-base techniques and it has been successfully adopted by the application domains such as document clustering, facial recognition and collaborative filtering. For the collaborative filtering, it becomes another popular modeling approach since their success in Netflix challenge. Moreover, traditional neighbor based collaborative filtering is combined with MF model [28] for improving the recommendation accuracy. These existing models are however not specifically designed to capture concept drift over time and they assume no correlation or dependency structure specific for *temporal* dynamics. In this article, we propose a generic model by incorporating hidden Markov models into matrix factorization and provide algorithms for solving the concept drifting problem in recommender system.

Moreover, side information, such as user-generated contents, the trustworthiness between users, and social network related information, has also been used by many recommender systems to improve the effectiveness of existing models [29]. Context-aware recommendation algorithms [30] are such technique which used to characterize the correlation between users' dynamic preferences and their contexts. In [31], a context-aware recommender system for mobile application discovery is proposed that utilizes the implicit feedback from personal usage history to form a binary tensor. Tensor Factorization, as a method of contextual modeling, has earned the attention of researchers. The effectiveness of this model has been confirmed by a number of studies [32]–[34].

B. USER PREFERENCE MODELS

User preference modeling and discovering is one of the important problem in recommender system. Existing studies have presented many potential ways to increase the recommendation performance by incorporating the user preferences explicitly or implicitly. For example, Wu *et al.* [36] and Michelson and Macskassy [37] aim at extracting keywords from user generated contents to represent their explicit user interests. In [38] and [39], are modeled from item descriptions and used for improving the collaborative filter performance.

The matrix factorization [3], [26] and hierarchical matrix factorization [27], [39], [40] can also be seen as a model for discovering and making use of the user implicit preferences. The decomposed user low-rank matrix can be seen as a representation of user preferences. The capability of capturing the user preferences have been confirmed by [42] and [43].

There also exist studies that try to model user interests in semantic level, among which some algorithms are based on matrix factorization algorithm [43] and some utilize LDA topic model [44], [45].

Based on the above discussion, the matrix factorization model can discovering the implicit preferences and overcoming the sparsity problem at the same time. Also, the approaches based on this model promote the development of the recommender system.

These existing studies merely focus on the static modeling of user preferences. However, in reality, user behavior or requirements might be changing over time. This fact makes it possible to model the user preference dynamics together with the traditional collaborative filtering approach.

C. MARKOV MODEL

As the objective of this paper is to promote a general state-space model to deal with temporal dynamics in collaborative filtering, Markov processes naturally arise as they are commonly used for constructing the dependencies between adjacent temporal slots. There have been existing previous works that take both hidden Markov models and collaborative filtering into consideration when building models. However, some of these models, such as [46], study the sequences between the items purchased by users. Others, such as [16], model the latent processes as first-order Markov processes and which give rise to Kalman filters (KF). The MFMP model of this paper goes beyond first-order Markov process and in fact the latent processes in MFMP model family can be a Markov process of arbitrary order and non-Gaussian. In addition, the main contribution of this work is not advocating one or two specific models and testing them for some

specific dataset. The main thrust of this paper is to advocate the philosophy of modeling temporal dynamics in collaborative filtering using a probabilistic framework where the observed matrix process is modeled as a product of two latent Markov processes, which may have arbitrary statistics.

The previous works combining KF with MF may be viewed as specific examples of the MFMP family and may be seen as a testimony of the usefulness of the MFMP modeling framework. Noting that none of the previous works considers higher-order or more general latent Markov processes and our MFMP models are essential classical state-space models (hidden Markov models) adapted to matrix factorization settings.

III. MARKOVIAN FACTORIZATION OF MATRIX PROCESS MODELS

A. GENERAL FORMULATION

Let $\{1, 2, \ldots, M\}$ index the set of items (e.g. movies) and $\{1, 2, \ldots, N\}$ index the set of users. Let $\{0, 1, \ldots, T\}$ index the set of discrete time points. There is a latent space $\mathbb{R}^D$ of dimension D with D much smaller than M and N. At every time point $t \in \{0, 1, 2, \ldots, T\}$, associate with each user i a vector $U_i(t) \in \mathbb{R}^D$ and associate each item j a vector $V_j(t) \in \mathbb{R}^D$. Intuitively $V_j(t)$ may be regarded as the latent feature of item j at time t and $U_i(t)$ may be regarded as the "weighting" vector of user i at time t. We assume that all user feature processes $\{U_i(t)\}$'s and all item feature processes $\{V_j(t)\}$'s are mutually independent. Collectively, for any fixed t, we denote $\{U_i(t) : i = 1, 2, \ldots, N\}$ and $\{V_j(t) : i = 1, 2, \ldots, M\}$ by $D \times N$ matrix $U(t)$ and $D \times M$ matrix $V(t)$ respectively.

We model every $\{U_i(t)\}$ process and every $\{V_j(t)\}$ process both as Markov processes, and model, for each (i, j) pair, $R_{ij}(t)$ to only depend on the inner product of $U_i(t)^T V_j(t)$ and nothing else. Depending on further specification of the processes $\{U_i(t)\}$'s and $\{V_j(t)\}$'s and the dependency of $R_{ij}(t)$ on $(U_i(t)^T V_j(t))$, various models may be constructed. We call such a model Markovian Factorization of Matrix Process (MFMP) model.

To be concrete, we now introduce two examples in the MFMP model family.

B. FIRST-ORDER MFMP

Given $U_i(t)$ and $V_j(t)$, the (i, j) entry of the matrix $R(t)$ is modeled as

$$R_{i,j}(t) = U_i(t)^T V_j(t) + Z_{ij}(t), \tag{1}$$

where $Z_{ij}(0), Z_{ij}(1), \ldots Z_{ij}(T)$ are i.i.d. Gaussian random variables with zero mean and variance σ^2. Then it follows that the distribution of $R_{ij}(t)$ conditioned on $U_i(t)$ and $V_j(t)$ is

$$p(R_{ij}(t)|U_i(t), V_j(t)) = \mathcal{N}(R_{ij}(t)|U_i(t)^T V_j(t), \sigma^2) \tag{2}$$

where we have used the notation $\mathcal{N}(x|\mu, \Sigma)$ to denote the Gaussian probability density function with variable x, mean vector μ and covariance matrix Σ. Note that in (2), the Gaussian density is univariate, in which the covariance matrix reduces to a scalar value.

For simplicity, we denote matrix processes $\{U(t)\}$, $\{V(t)\}$ and $\{R(t)\}$ over time $t = 0, 1, 2, \ldots, T$ by $\mathcal{U}, \mathcal{V}$ and $\mathcal{R}$ respectively. In addition, for each time point t, denote by $\mathcal{K}(t)$ the set of all (i, j) pairs for which $R_{ij}(t)$ is observed. In the matrix process $\mathcal{R}$, we denote the set of all observed $R_{ij}(t)$ (across all (i, j) pairs and all values of t) by $\widetilde{\mathcal{R}}$. Under these notations, we have

$$p(\widetilde{\mathcal{R}}|\mathcal{U}, \mathcal{V}) = \prod_{t=0}^{T} \prod_{(i,j)\in\mathcal{K}(t)} \mathcal{N}(R_{ij}(t)|U_i(t)^T V_j(t), \sigma^2) \tag{3}$$

At the time origin $t = 0$, we model $U_i(0)$ to follow a zero-mean spherical Gaussian distribution for each user i, namely,

$$p(U_i(0)) = \mathcal{N}(U_i(0)|0, \Sigma_U^2 I) \tag{4}$$

Likewise, at $t = 0$, we model $V_j(0)$ to follow another zero-mean spherical Gaussian distribution for each item j, namely,

$$p(V_j(0)) = \mathcal{N}(V_j(0)|0, \Sigma_V^2 I) \tag{5}$$

The evolution of U_i across time for each user i is modeled as

$$U_i(t+1) = U_i(t) + X_i(t), \tag{6}$$

where $\{X_i(t)\}$ is an i.i.d. Gaussian process with each $X_i(t)$ drawn from distribution $\mathcal{N}(X_i(t)|0, \sigma_U^2 I)$. Essentially $X_i(t)$ models the drift of feature vector of user i at time t.

Similarly, the evolution of V_j across time for every item is modeled as

$$V_j(t+1) = V_j(t) + Y_j(t), \tag{7}$$

where $\{Y_j(t)\}$ is an i.i.d. Gaussian process with each $Y_j(t)$ is drawn from distribution $\mathcal{N}(Y_j(t)|0, \sigma_V^2 I)$. Apparently $Y_j(t)$ models the drift of feature vector of item j at time t. Although some may consider item feature as an intrinsic property of the considered item, there are a few good reasons to believe that it may indeed change over time [11].

The assumptions prescribed by Equations (1) through (7) completely specifies the first-order MFMP model. Under these assumptions, it is evident that both U_i process and V_j process are Gaussian Markov processes. Moreover processes U_i, V_j and R_{ij} form a Hidden Markov Model [15] with $(U_i(t), V_i(t))$ being the latent state of the underlying Markov chain at time t and $R_{ij}(t)$ being the observed variable at time t. This is best explained using the Bayesian network in Figure 1.

It is now straight-forward to express the posterior distribution $p(\mathcal{U}, \mathcal{V}|\widetilde{\mathcal{R}})$ and translate the problem of completing matrix process $\mathcal{R}$ to maximum *a posteriori* (MAP) inference of $(\mathcal{U}, \mathcal{V})$, namely, finding

$$\begin{aligned}
&(\widehat{\mathcal{U}}, \widehat{\mathcal{V}}) \\
&:= \arg\max_{\mathcal{U},\mathcal{V}} \log p(\mathcal{U}, \mathcal{V}|\widetilde{\mathcal{R}}) \\
&= \arg\max_{\mathcal{U},\mathcal{V}} \log p(\mathcal{U}, \mathcal{V}, \widetilde{\mathcal{R}}) \\
&= \arg\max_{\mathcal{U},\mathcal{V}} \log \left(\prod_{t=0}^{T} \prod_{(i,j)\in\mathcal{K}(t)} \mathcal{N}(R_{ij}(t)|U_i(t)^T V_j(t), \sigma^2)) \right)
\end{aligned}$$

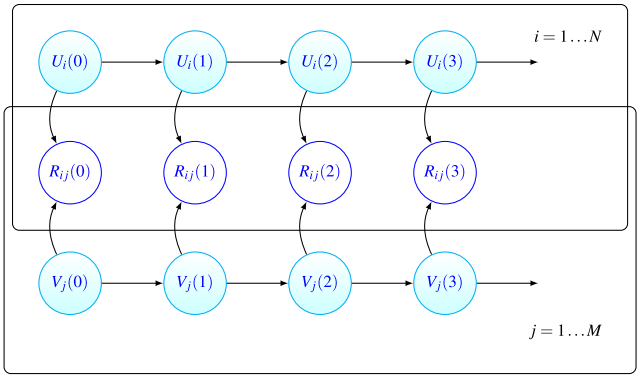


FIGURE 1. First-order MFMP model.

$$\times \left(\prod_{i=1}^{N} \mathcal{N}(U_i(0)|0, \Sigma_U^2 I) \prod_{t=1}^{T} \mathcal{N}(U_i(t)|U_i(t-1), \sigma_U^2) \right) \times \left(\prod_{j=1}^{M} \mathcal{N}(V_j(0)|0, \Sigma_V^2 I) \prod_{t=1}^{T} \mathcal{N}(V_j(t)|V_j(t-1), \sigma_V^2) \right) = \arg\min_{\mathcal{U},\mathcal{V}} L(\mathcal{U}, \mathcal{V}), \tag{8}$$

where

$$\begin{aligned} L(\mathcal{U}, \mathcal{V}) := & \sum_{t=0}^{T} \sum_{(i,j)\in\mathcal{K}(t)} \left(R_{ij}(t) - U_i(t)^T V_j(t) \right)^2 \\ & + \rho_U ||U(0)||_F^2 + \rho_V ||V(0)||_F^2 \\ & + \lambda_U \sum_{t=1}^{T} ||U(t) - U(t-1)||_F^2 \\ & + \lambda_V \sum_{t=1}^{T} ||V(t) - V(t-1)||_F^2. \end{aligned} \tag{9}$$

In the above equation, $\rho_U := \sigma^2/\Sigma_U^2$, $\rho_V := \sigma^2/\Sigma_V^2$, $\lambda_U := \lambda^2/\sigma_U^2$, $\lambda_V := \sigma^2/\sigma_V^2$ and $||\cdot||_F$ denotes the Frobenius norm.

Comparing with the objective function to optimize in PMF, one can easily identify that the function L here has the additional term $\lambda_U \sum_{t=1}^{T} ||U(t) - U(t-1)||_F^2 + \lambda_V \sum_{t=1}^{T} ||V(t) - V(t-1)||_F^2$, which precisely captures the drifts between every two consecutive U-matrices and between the consecutive V-matrices. In fact, if we make $\sigma_U^2 = \sigma_V^2 = 0$, the model forces $U(t) = U(t-1)$ and $V(t) = V(t-1)$, which makes $U(t) = U(0)$, $V(t) = V(0)$ for every t. In this case, the model essentially reduces to the PMF model. Interestingly, this reduction can also be seen from the resulting optimization problem: making $\sigma_U^2 = \sigma_V^2 = 0$ corresponds to making $\lambda_U = \lambda_V = \infty$; in practice, this can be done by choosing very large λ_U and λ_V, the solution of the minimization problem is then necessarily driven to a configuration where both $\sum_{t=1}^{T} ||U(t) - U(t-1)||_F^2$ and $\sum_{t=1}^{T} ||V(t) - V(t-1)||_F^2$ are very small or effectively zero.

Algorithm 1 First-Order MFMP

Input: invocation matrix $\mathcal{R}$, regularization parameter λ, learning rate η, number of latent factors D, max iterations *itmax*

Output: parameter set U_i and V_j

```
Initialize U_i and V_j with random value
for t = 0,t <= T,t++ do
  for (i, j) ∈ K(t) do
    if t == 0 then
      update U_i(0), V_j(0) according to Eq. (11) and Eq. (12)
    else if t == T then
      update U_i(T), V_j(T) according to Eq. (13) and Eq. (14)
    else
      update U_i(t), V_j(t) according to Eq. (15) and Eq. (16)
    end if
  end for
end for
```

The above optimization problem can be solved using gradient decent in $(\mathcal{U}, \mathcal{V})$. Denote

$$e_{ij}(t) := R_{ij}(t) - U_i(t)^T V_j(t), \tag{10}$$

the derivatives of the function L are given below.

$$\frac{\partial L}{\partial U_i(0)} = -2 \sum_{j:(i,j)\in\mathcal{K}(0)} e_{ij}(0) V_j(0) + 2(\rho_U + \lambda_U) U_i(0) - 2\lambda_U U_i(1) \tag{11}$$

$$\frac{\partial L}{\partial V_j(0)} = -2 \sum_{i:(i,j)\in\mathcal{K}(0)} e_{ij}(0) U_i(0) + 2(\rho_V + \lambda_V) V_j(0) - 2\lambda_V V_j(1) \tag{12}$$

$$\frac{\partial L}{\partial U_i(T)} = -2 \sum_{j:(i,j)\in\mathcal{K}(T)} e_{ij}(T) V_j(T) + 2\lambda_U U_i(T) - 2\lambda_U U_i(T-1) \tag{13}$$

$$\frac{\partial L}{\partial V_j(T)} = -2 \sum_{i:(i,j)\in\mathcal{K}(T)} e_{ij}(T) U_i(T) + 2\lambda_V V_j(T) - 2\lambda_V V_j(T-1) \tag{14}$$

For any $t = 1, 2, \ldots T-1$,

$$\frac{\partial L}{\partial U_i(t)} = -2 \sum_{j:(i,j)\in\mathcal{K}(t)} e_{ij}(t) V_j(t) + 4\lambda_U U_i(t) - 2\lambda_U U_i(t-1) - 2\lambda_U U_i(t+1) \tag{15}$$

$$\frac{\partial L}{\partial V_j(t)} = -2 \sum_{i:(i,j)\in\mathcal{K}(t)} e_{ij}(t) U_i(t) + 4\lambda_V V_j(t) - 2\lambda_V V_j(t-1) - 2\lambda_V V_j(t+1) \tag{16}$$

C. SECOND-ORDER MFMP

In this model, instead of modeling the drifting terms $\{X_i(t)\}$ and $\{Y_j(t)\}$ being independent across time, we model them as

(first order) Gaussian Markov processes. More precisely,

$$p(X_i(0)) = \mathcal{N}(X_i(0)|0, \sigma_U^2 I) \tag{17}$$
$$X_i(t+1) = a_U X_i(t) + \Gamma_i(t), \tag{18}$$

where for each i, $\{\Gamma_i(t)\}$ is an i.i.d. Gaussian process with zero mean and variance $b_U\sigma_U^2$, and all such processes are independent across all i's; similarly,

$$p(Y_j(0)) = \mathcal{N}(Y_j(0)|0, \sigma_V^2 I) \tag{19}$$
$$Y_j(t+1) = a_V Y_j(t) + \Delta_j(t), \tag{20}$$

where for each j, $\{\Delta_j(t)\}$ is an i.i.d. Gaussian process with zero mean and variance $b_V\sigma_V^2$, and all such processes are independent across all j's. Modifying the first-order MFMP model according to Equations (17) to (20) specifies the second-order MFMP model.

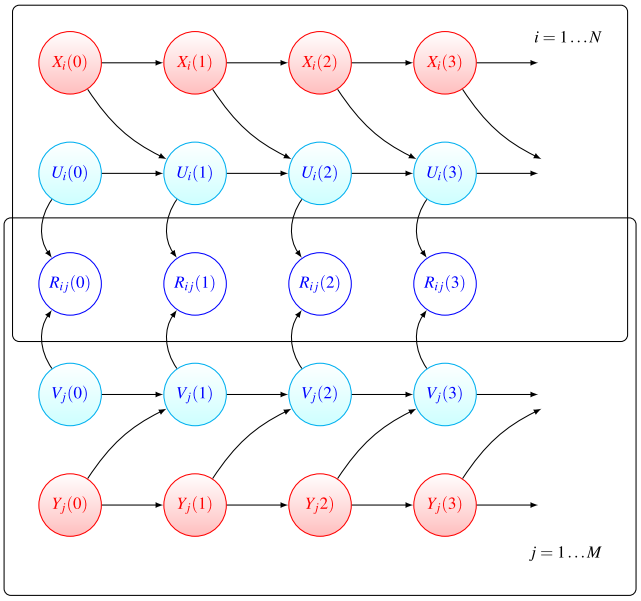


FIGURE 2. Second-order MFMP model with drifting processes included.

Collectively denoting the processes $\{X_i(t)\}$ across all i's by $\mathcal{X}$ and processes $\{Y_j(t)\}$ across all j's by $\mathcal{Y}$, the Bayesian network describing the relationship between $\mathcal{X}, \mathcal{Y}, \mathcal{U}, \mathcal{V}, \mathcal{R}$ is given in Figure 2. Since our objective is to infer $(\mathcal{U}, \mathcal{V})$, it is beneficial to marginalize out processes $\mathcal{X}$ and $\mathcal{Y}$ from the model. This gives rise to the graphical model in Figure 3. It is easy to see from the figure that the latent processes $\mathcal{U}$ and $\mathcal{V}$ in this model both are second-order Markov processes.

For any $d = 1, 2, \ldots, D$, $t = 0, 1, \ldots, T$, and $i = 1, 2, \ldots N$ (reps, $j = 1, 2, \ldots, M$), it can be shown that the d-th element of $U_i(t)$ (resp. of $V_j(t)$), denoted by $U_i(t; d)$ (resp. denoted by $V_j(t; d)$), is a zero-mean Gaussian random variable. Let

$$U_i[d] := (U_i(0; d), U_i(1; d), U_i(2; d), \ldots, U_i(T; d))^T,$$
$$V_j[d] := (V_j(0; d), V_j(1; d), V_j(2; d), \ldots, V_j(T; d))^T,$$
$$X_i[d] := (X_i(0; d), X_i(1; d), X_i(2; d), \ldots, X_i(T; d))^T,$$
$$Y_j[d] := (Y_j(0; d), Y_j(1; d), Y_j(2; d), \ldots, Y_j(T; d))^T,$$

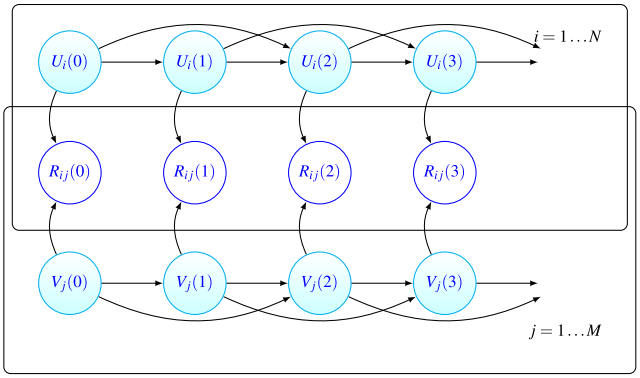


FIGURE 3. Second-order MFMP model with drifting processes marginalized out.

and let $(T+1) \times (T+1)$ matrices K_U, K_V, K_X, K_Y be respectively the covariance matrices of random vectors $U_i[d]$, $V_j[d]$, $X_i[d]$ and $Y_j[d]$ (noting that these covariance matrices are independent of i, j and d). We index the rows and columns of these covariance matrices by $\{0, 1, \ldots, T\}$ (instead of $\{1, 2, \ldots, T+1\}$) to be consistent with our notation for time indices. For any t and $t' \in \{0, 1, 2, \ldots, T\}$, it can be shown that

$$K_X(t, t') = a_U^{|t-t'|}\left(\frac{b_U - a_U^{2\tau} b_U}{1 - a_U^2} + a_U^{2\tau}\right)\sigma_U^2 \tag{21}$$
$$K_Y(t, t') = a_V^{|t-t'|}\left(\frac{b_V - a_V^{2\tau} b_V}{1 - a_V^2} + a_V^{2\tau}\right)\sigma_V^2, \tag{22}$$

where in (21) and (22), τ denotes the smaller number between t and t'. Further,

$$K_U(t, t') = \begin{cases} \Sigma_U^2 + \sum_{l'=0}^{t'-1}\sum_{l=0}^{t-1} K_X(l, l'), & \text{if } t, t' \geq 1 \\ \Sigma_U^2, & \text{otherwise} \end{cases} \tag{23}$$
$$K_V(t, t') = \begin{cases} \Sigma_V^2 + \sum_{l'=0}^{t'-1}\sum_{l=0}^{t-1} K_Y(l, l'), & \text{if } t, t' \geq 1 \\ \Sigma_V^2, & \text{otherwise} \end{cases} \tag{24}$$

Formulating the problem of completing matrix process $\mathcal{R}$ as MAP inference on $(\mathcal{U}, \mathcal{V})$ and reducing the inference problem in a way similar to (8) in the first-order MFMP model, we arrive at the following optimization problem: finding

$$(\widehat{\mathcal{U}}, \widehat{\mathcal{V}}) := \arg\min_{\mathcal{U},\mathcal{V}} L'(\mathcal{U}, \mathcal{V}), \tag{25}$$

where

$$\begin{aligned} L'(\mathcal{U}, \mathcal{V}) := &\sum_{t=0}^{T}\sum_{(i,j)\in\mathcal{K}(t)}\left(R_{ij}(t) - U_i(t)^T V_j(t)\right)^2 \\ &+ (\Sigma_U^2)^{-1}||U(0)||_F^2 + (\Sigma_V^2)^{-1}||V(0)||_F^2 \\ &+ 2\sum_{i=1}^{N}\sum_{d=1}^{D} U_i[d]^T K_U^{-1} U_i[d] \\ &+ 2\sum_{j=1}^{M}\sum_{d=1}^{D} V_j[d]^T K_V^{-1} V_j[d] \end{aligned} \tag{26}$$

Algorithm 2 Second-Order MFMP

Input: invocation matrix $\mathcal{R}$, regularization parameter λ, learning rate η, number of latent factors D, max iterations *itmax*

Output: parameter set U_i and V_j

```
Initialize U_i and V_j with random value
for t = 0,t <= T,t++ do
  for (i, j) ∈ K(t) do
    if t == 0 then
      update U_i(0), V_j(0) according to Eq. (27) and Eq. (28)
    else
      update U_i(t), V_j(t) according to Eq. (29) and Eq. (30)
    end if
  end for
end for
```

We note that in the above objective function $L'(\mathcal{U}, \mathcal{V})$, the parameters are all normalized with respect to σ^2 and as such parameter σ^2 is eliminated (i.e. set to 1). For any given parameter setting of $(\Sigma_U^2, \Sigma_V^2, \sigma_U^2, \sigma_V^2, a_U, b_U, a_V, b_V)$, the optimization problem can be solved again with gradient descent. We now present the derivatives of the objective function with respect to $\mathcal{U}$ and $\mathcal{V}$.

For notational convenience, let $\Lambda_U := 2K_U^{-1}$ and $\Lambda_V := 2K_V^{-1}$. Noting that both Λ_U and Λ_V are $(T+1) \times (T+1)$ matrices, we use $\{0, 1, \ldots, T\}$ to index their rows and columns. Identifying U_i and V_j both as $D \times (T+1)$ matrices, we use $U_i(t)$ (resp. $V_j(t)$) to denote the t-th column of matrix U_i (resp., of matrix V_j). Similarly, we use $\Lambda_U(t)$ (resp. $\Lambda_V(t)$) to denote the t-th column of matrix Λ_U (resp. of matrix Λ_V).

$$\frac{\partial L'}{\partial U_i(0)} = 2 \sum_{j:(i,j)\in\mathcal{K}(0)} e_{ij} V_j(0) + 2(\Sigma_V^2)^{-1} U_i(0) + 2U_i \Lambda_U(0), \tag{27}$$

$$\frac{\partial L'}{\partial V_j(0)} = 2 \sum_{i:(i,j)\in\mathcal{K}(0)} e_{ij} U_i(0) + 2(\Sigma_V^2)^{-1} V_j(0) + 2V_j \Lambda_V(0), \tag{28}$$

and for any $t \neq 0$,

$$\frac{\partial L'}{\partial U_i(t)} = 2 \sum_{j:(i,j)\in\mathcal{K}(t)} e_{ij} V_j(t) + 2U_i \Lambda_U(t), \tag{29}$$

$$\frac{\partial L'}{\partial V_j(t)} = 2 \sum_{i:(i,j)\in\mathcal{K}(t)} e_{ij} U_i(t) + 2V_j \Lambda_V(t). \tag{30}$$

D. DISCUSSION

The first-order and second-order MFMP models presented above are merely the simplest examples in the MFMP model family. Depending on the temporal dynamic nature of the dataset, one may impose higher-order Markovian structure on the latent processes. For data of other types, one may also consider distributions beyond the Gaussian family. The probabilistic dependency of $R_{ij}(t)$ on the inner product $U_i(t)^T V_j(t)$ may also be made arbitrary. In fact, a Hidden Markov Model has been recently presented for interpreting users' blog-reading behavior and making article recommendations [16]. The model thereof consists of only one latent non-Gaussian Markov Matrix process, which serves as the user feature process. One may regard this model as a special case of the MFMP model family, in which the process $\{V(t)\}$ is trivialized.

IV. EXPERIMENTS

Experimental study is performed to evaluate the effectiveness of the proposed first-order and second-order MFMP models in the prediction of movie ratings. The proposed models are also compared with the well-established timeSVD++ algorithm [11] and a standard tensor factorization algorithm [47].

A. DATASETS AND EVALUATION METRIC

Two datasets we use in the experiments are from MovieLens-1M and MovieLens-20M.[1] MovieLens-1M data set contains 1 million ratings from 6,000 users on 4,000 movies. MovieLens-20M data set contains and 20 million ratings from 138,000 users on over 27,000 movies respectively. These two dataset also contain the time-stamp of when each user rated the movies. Movies in MovieLens-1M data set were rated in a 1040 days period and in MovieLens-20M data set were rated in a 20 years period. The ratings are on the scale of 1 to 5. As is standard, root mean squared error (RMSE) and Precision@10 are used as the evaluation metric.

B. PARAMETER SELECTION AND OVERALL PERFORMANCES OF MFMP MODELS

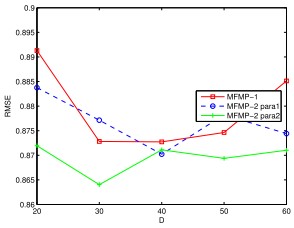
Existing matrix factorization methods merely provide the functionality of predicting missing values of a matrix. However, future value prediction is also important for many cases. As the proposed model characterizes the evolution of users and items, it naturally support the future prediction. Such, in this empirical study, the performances of future rating prediction is our focus to be examined.

In the experiments, we randomly select 80% of time-stamped ratings as the training set and take the remaining set of ratings as the testing set. Time is divided into slots of duration S days, which we let vary in our experiments.

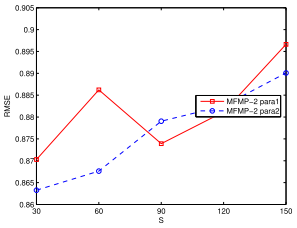
For MovieLens-1M dataset, the parameters for the first-order MFMP are chosen as $\{\rho_u = \rho_v = 0.05, \lambda_u = \lambda_v = 100\}$. For MovieLens-20M dataset, the parameters for the first-order MFMP are chosen as $\{\rho_u = \rho_v = 0.01, \lambda_u = \lambda_v = 200\}$. Noting the probabilistic meanings of the parameters, these parameter settings allow relatively large variation across the elements of any given latent (user or movie) feature vector but only allows small drifts across time.

For the second-order MFMP on MovieLens-1M dataset, two sets of parameters are chosen: Para1 $= \{a_U = 0.3,$

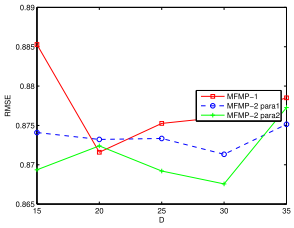
[1] http://movielens.umn.edu/



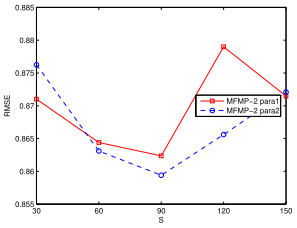
(1) Performance of MFMP models on MovieLens-1M dataset for varying values of latent space dimension D. $S = 30$ days.



(2) Performance of the second-order MFMP models on MovieLens-1M dataset for various choices of time slot size S. $D = 30$.

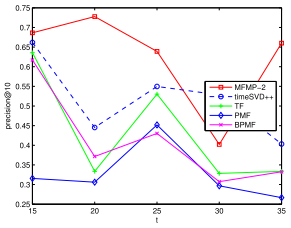


(3) Performance of MFMP models on MovieLens-20M dataset for varying values of latent space dimension D. $S = 90$ days.

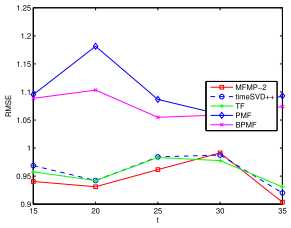


(4) Performance of the second-order MFMP models on MovieLens-20M dataset for various choices of time slot size S. $D = 30$.

FIGURE 4. **The choices of latent space dimension *D* and time slot size *S*. (1) Performance of MFMP models on MovieLens-1M dataset for varying values of latent space dimension *D*. *S* = 30 days. (2) Performance of the second-order MFMP models on MovieLens-1M dataset for various choices of time slot size *S*. *D* = 30. (3) Performance of MFMP models on MovieLens-20M dataset for varying values of latent space dimension *D*. *S* = 90 days. (4) Performance of the second-order MFMP models on MovieLens-20M dataset for various choices of time slot size *S*. *D* = 30.**

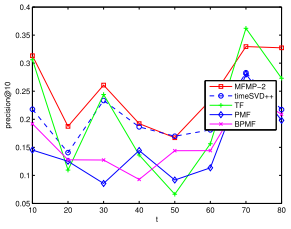


(1) Precision@10

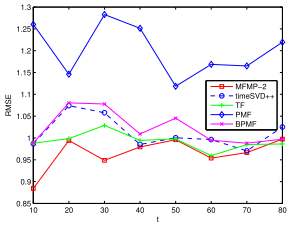


(2) RMSE

FIGURE 5. **Performance of various models on MovieLens 1M dataset. (1) Precision@10. (2) RMSE.**



(1) Precision@10



(2) RMSE

FIGURE 6. **Performance of various models on MovieLens 20M dataset. (1) Precision@10. (2) RMSE.**

$b_U = 0.5, a_V = 0.1, b_V = 0.08\}$ and Para2 $= \{a_U = 0.35, b_U = 0.1, a_V = 0.8, b_V = 0.5\}$. For the second-order MFMP on MovieLens-20M dataset, two sets of parameters are chosen: Para1 $= \{a_U = 0.5, b_U = 0.8, a_V = 0.05, b_V = 0.1\}$ and Para2 $= \{a_U = 0.7, b_U = 0.15, a_V = 1.1, b_V = 0.65\}$.

With Para1 setting, the choice a_V as a small value eliminates the correlation between consecutive drifts on movie feature vectors, which reduces the $\{V(t)\}$ process to a first-order Markov process, and the choice b_V as a small value allows only small drifts in movie feature vectors across time. With Para2, the model allows modest correlation in both user feature drifts and in movie feature drifts.

Figure 4 demonstrate performances of the first-order MFMP model (MFMP-1) and second-order MFMP model (MFMP-2) with varying settings of latent space dimension D on MovieLens-1M and MovieLens-20M respectively. In this figure, it appears that the second-order MFMP models overall perform better than the first-order model. Comparing the two second-order MFMP models, setting Para1 appears to perform better than setting Para2 when the choice of D is relatively small and the two settings perform similarly for larger values of D. According to the evaluation results, we choose $D = 30$ and $S = 30$ for MovieLens-1M dataset and $D = 30$ and $S = 90$ for MovieLens-20M dataset for further experiments.

The two parameter settings of the second-order MFMP model are further investigated for various choices of time slot size S and the results are plotted in Figure 4(2) and (4). The two settings result in similar performances. It appears that with small values of S, Para1 performs better, and with larger values of S, Para2 performs better. This makes it difficult to conclude whether in this dataset correlation indeed exists across consecutive drifts of movie features.

In the following experiments, we choose the second-order MFMP model to compare with other state-of-the-art models.

C. DYNAMICS PERFORMANCE: COMPARING MFMP, timeSVD++, AND TENSOR FACTORIZATION

In order to evaluate the temporal dynamics modeling performance of the proposed model, we choose tensor factorization (CANDECOMP/PARAFAC model) (TF) [47], probabilistic matrix factorization (PMF) [3], Bayesian probabilistic matrix factorization (BPMF) [6] and timeSVD++ [11] as the comparing models.

By choosing time slot size $S = 30$ for MovieLens-1M and $S = 90$ for MovieLens-20M, we first partition the MovieLens-1M dataset and the MovieLens-20M dataset into 35 time slots and 80 time slots respectively. For each time slot index t in the set $\{5, 10, 15, 20, 25, 30, 35\}$ for MovieLens-1M and t in the set $\{10, 20, 30, 40, 50, 60, 70, 80\}$ for

MovieLens-20M, we use the rating information from time slot 0 to time slot $t-1$ to predict the ratings in time slot t. The parameter settings for PMF, BPMF, timeSVD++ and tensor factorization are obtained by discretization the parameter space into coarse grid and selecting the best parameter setting on the grid. The parameters chosen for MPMF-2 on MovieLens-1M data set are: latent space dimension $D = 30$, time slot size $S = 30$, $a_U = 0.35$, $b_U = 0.1$, $a_V = 0.8$ and $b_V = 0.5$. The parameters for MovieLens-20M data set are: $D = 30$, $S = 90$, $a_U = 0.7$, $b_U = 0.15$, $a_V = 1.1$ and $b_V = 0.65$.

Figure 5 and 6 show the rating instances to be evaluated for each time-slot. It is clear that MFMP consistently achieves a better prediction performance for all compared time-slots in terms of RMSE. For the evaluation metric Precision@10, our MFMP performs best on most of the time slots except $t = 30$ for MovieLens-1M and $t = 90$ for MovieLens-20M. Among these two exception, MPMF achieves the second best performance and the difference between MFMP and the best models is relatively small. This confirms the temporal dynamics modeling capability of the markovian factorization matrix process model.

V. OUTLOOK AND DISCUSSION

As noted earlier, the two MFMP models presented in this paper for the prediction of movie ratings are in their most basic forms. There is plenty room to further improve the models. For example, one standard technique as has been applied in PMF [3] is using a logistic function to map the $U_i(t)^T V_j(t)$ to the range between 1 and 5. This should allow the observed ratings fit the models more closely. The prior distributions for $U_i(0)$ and $V_j(0)$ can be made to have non-zero means, in order to provide global biases to the latent user features and movie features. This should have an effect similar to the introduction of global bias in timeSVD++ [11]. Individualizing the prior means across users and across movies may effectively create user-specific bias and movie-specific bias, which has actually been implemented in timeSVD++. The models may be enhanced to a ''constrained'' version paralleling the enhancement of PMF to ''constrained PMF'' [3], in order to better account for infrequent users.

It is also worth noting that the two model examples presented in the paper have not carefully accounted for sudden changes of latent features. More general models beyond those relying on Gaussian distributions and additive drifts should be considered when there is significant presence of such phenomenon in the datasets.

Most probabilistically formulated models can be extended to a Bayesian version (see, e.g., [6], [8], [48]). A similar Bayesian approach may be taken to augment the present models with hyper-parameters for the prior distributions. Inference under such models usually exhibits an averaging effect over all model parameter settings and has the benefit of over-fitting avoidance.

At this point, we wish to articulate the message we wish to convey in this paper. MFMP models are a rich family of probabilistic models that marry the framework of PMF with the framework of Hidden Markov Models. By varying the order of the latent Markov processes, the involved distributions and the dependency of observation on the latent process, a large variety of temporal dynamical models can be constructed for collaborative filtering problems.

One possible extension of this study is to include the textual information associated with the rating to handle the ''cold-start'' or ''early-voter'' problems. We plan to include this in our future work to increase the applicability of the proposed model. Except for investigating the many possibilities to extend the presented models and to construct new MFMP models for a wide spectrum of collaborative filtering problems, we are, out of curiosity and theoretical interest, intrigued by an open research problem. To put the problem in context, it has been understood that various factorized models, including tensor factorization models, matrix factorization models and the classical PCA model [49] are inter-related. In particular, the Tucker decomposition based tensor factorization model [9] may reduce to the CANDECOMP/PARAFAC decomposition based tensor factorization model [7] when imposing a ''rank-1'' constraint on the core tensor; the CANDECOMP/PARAFAC decomposition based tensor factorization model may reduce to the PMF model when trivializing the tensor to a matrix; PMF may reduce to PCA when trivializing one of the latent matrix factors to a vector. This chain of hierarchy motivates an interest in mapping out the two MFMP models of this paper in the big picture of these factorized models. In particular, we pose the following question: May the two MFMP models of this paper be reduced from the Tucker decomposition based tensor factorization model? If the answer is yes, these MFMP models may be unified into the same mathematical framework of tensor factorization. If the answer is no, it suggests that the MFMP model family may contain unique algebraic structure beyond what has been studied in machine learning literature. Our intuition, beyond what should be stated within the scope and length constraint of this paper, makes us conjecture that the answer to this question is negative.

REFERENCES

[1] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, ''Item-based collaborative filtering recommendation algorithms,'' in *Proc. ACM 10th Int. Conf. World Wide Web (WWW)*, New York, NY, USA, 2001, pp. 285–295, doi: 10.1145/371920.372071.

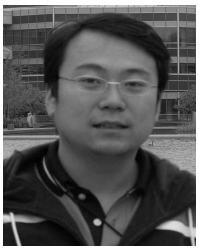
[2] G. Linden, B. Smith, and J. York, ''Amazon.com recommendations: Item-to-item collaborative filtering,'' *IEEE Internet Comput.*, vol. 7, no. 1, pp. 76–80, Jan./Feb. 2003.

[3] R. R. Salakhutdinov and A. Mnih, ''Probabilistic matrix factorization,'' in *Advances in Neural Information Processing Systems*. Cambridge, MA, USA: MIT Press, 2008, pp. 1257–1264.

[4] H. Ma, H. Yang, M. R. Lyu, and I. King, ''SoRec: Social recommendation using probabilistic matrix factorization,'' in *Proc. 17th ACM Conf. Inf. Knowl. Manage. (CIKM)*, 2008, pp. 931–940.

[5] Z. Li, J. Liu, X. Zhu, T. Liu, and H. Lu, ''Image annotation using multi-correlation probabilistic matrix factorization,'' in *Proc. ACM Int. Conf. Multimedia (MM)*, 2010, pp. 1187–1190.

[6] R. Salakhutdinov and A. Mnih, "Bayesian probabilistic matrix factorization using Markov chain Monte Carlo," in *Proc. ACM 25th Int. Conf. Mach. Learn. (ICML)*, 2008, pp. 880–887.
[7] Y. K. Yılmaz and A. T. Cemgil, "Probabilistic latent tensor factorization," in *Proc. 9th Int. Conf. Latent Variable Anal. Signal Separat. (LVA/ICA)*. New York, NY, USA: Springer-Verlag, 2010, pp. 346–353.
[8] 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*. Philadelphia, PA, USA: SIAM, 2010, pp. 211–222.
[9] W. Chu and Z. Ghahramani, "Probabilistic models for incomplete multi-dimensional arrays," *Artif. Intell. Statist.*, vol. 5, pp. 89–96, Apr. 2009.
[10] Z. Xu, F. Yan, and A. Qi, "Infinite tucker decomposition: Nonparametric Bayesian models for multiway data analysis," in *Proc. 29th Int. Conf. Mach. Learn. (ICML)*, J. Langford and J. Pineau, Eds. Edinburgh, Scotland: Omnipress, 2012, pp. 1023–1030.
[11] Y. Koren, "Collaborative filtering with temporal dynamics," *Commun. ACM*, vol. 53, no. 4, pp. 89–97, 2010.
[12] K. Y. Yılmaz, A. T. Cemgil, and U. Simsekli, "Generalised coupled tensor factorisation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 2151–2159.
[13] A. P. Singh and G. J. Gordon, "Relational learning via collective matrix factorization," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2008, pp. 650–658.
[14] J. C. Schlimmer and R. H. Granger, Jr., "Beyond incremental processing: Tracking concept drift," in *Proc. 5th Nat. Conf. Artif. Intell.* Philadelphia, PA, USA: AAAI Press, 1986, pp. 502–507.
[15] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," in *Readings in Speech Recognition*. San Mateo, CA, USA: Morgan Kaufmann, 1990, pp. 267–296.
[16] N. Sahoo, P. V. Singh, and T. Mukhopadhyay, "A hidden Markov model for collaborative filtering," *MIS Quart.*, vol. 36, no. 4, pp. 1329–1356, Dec. 2012.
[17] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: An open architecture for collaborative filtering of netnews," in *Proc. ACM Conf. Comput. Supported Cooperat. Work (CSCW)*, New York, NY, USA, 1994, pp. 175–186, doi: 10.1145/192844.192905.
[18] R. Burke, "Hybrid recommender systems: Survey and experiments," *User Model. User-Adapted Interact.*, vol. 12, no. 4, pp. 331–370, Nov. 2002, doi: 10.1023/A:1021240730564.
[19] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 6, pp. 734–749, Jun. 2005.
[20] J. Bobadilla, F. Ortega, A. Hernando, and A. GutiéRrez, "Recommender systems survey," *Know.-Based Syst.*, vol. 46, pp. 109–132, Jul. 2013, doi: 10.1016/j.knosys.2013.03.012.
[21] G. Takács, I. Pilászy, B. Németh, and D. Tikk, "Scalable collaborative filtering approaches for large recommender systems," *J. Mach. Learn. Res.*, vol. 10, pp. 623–656, Mar. 2009, doi: 10.1145/1577069.1577091.
[22] X. Luo, M. Zhou, S. Li, Z. You, Y. Xia, and Q.-S. 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.
[23] X. Luo, M. Zhou, Y. Xia, and Q. Zhu, "An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems," *IEEE Trans. Ind. Informat.*, vol. 10, no. 2, pp. 1273–1284, May 2014.
[24] X. Luo, M. C. Zhou, S. Li, Y. Xia, Z. You, Q. Zhu, and H. Leung, "An efficient second-order approach to factorize sparse matrices in recommender systems," *IEEE Trans. Ind. Informat.*, vol. 11, no. 4, pp. 946–956, Aug. 2015.
[25] G. Adomavicius and J. Zhang, "Stability of recommendation algorithms," *ACM Trans. Inf. Syst.*, vol. 30, no. 4, pp. 23-1–23-31, Nov. 2012, doi: 10.1145/2382438.2382442.
[26] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *IEEE Comput.*, vol. 42, no. 8, pp. 30–37, Aug. 2009.
[27] H. Shan, J. Kattge, P. Reich, A. Banerjee, F. Schrodt, and M. Reichstein, "Gap filling in the plant kingdom—Trait prediction using hierarchical probabilistic matrix factorization," in *Proc. 29th Int. Conf. Mach. Learn. (ICML)*, Scotland, U.K., Jun./Jul. 2012, pp. 1303–1310.
[28] Y. Koren, "Factorization meets the neighborhood: A multifaceted collaborative filtering model," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, New York, NY, USA, 2008, pp. 426–434.
[29] Y. Shi, M. Larson, and A. Hanjalic, "Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges," *ACM Comput. Surv.*, vol. 47, pp. 3-1–3-45, May 2014.
[30] G. Adomavicius and A. Tuzhilin, "Context-aware recommender systems," in *Recommender Systems Handbook*. Boston, MA, USA: Springer, 2011, pp. 217–253.
[31] A. Karatzoglou, L. Baltrunas, K. Church, and M. Böhmer, "Climbing the app wall: Enabling mobile app discovery through context-aware recommendations," in *Proc. 21st ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, New York, NY, USA, 2012, pp. 2527–2530, doi: 10.1145/2396761.2398683.
[32] B. Hidasi, "Context-aware preference modeling with factorization," in *Proc. RecSys*, 2015, pp. 371–374.
[33] H. Wermser, A. Rettinger, and V. Tresp, "Modeling and learning context-aware recommendation scenarios using tensor decomposition," in *Proc. Int. Conf. Adv. Social Netw. Anal. Mining*, 2011, pp. 137–144.
[34] Y. Shi, A. Karatzoglou, L. Baltrunas, M. A. Larson, A. Hanjalic, and N. Oliver, "TFMAP: Optimizing map for top-n context-aware recommendation," in *Proc. 35th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, Portland, OR, USA, 2012, pp. 155–164.
[35] W. Wu, B. Zhang, and M. Ostendorf, "Automatic generation of personalized annotation tags for Twitter users," in *Proc. Hum. Lang. Technol., Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics*. Los Angeles, CA, USA: Association for Computational Linguistics, 2010, pp. 689–692.
[36] M. Michelson and S. A. Macskassy, "Discovering users' topics of interest on Twitter: A first look," in *Proc. ACM 4th Workshop Anal. Noisy Unstructured Text Data*, 2010, pp. 73–80.
[37] J. Zhang, Y. Lin, M. Lin, and J. Liu, "An effective collaborative filtering algorithm based on user preference clustering," *Appl. Intell.*, vol. 45, no. 2, pp. 230–240, Sep. 2016.
[38] K.-Y. Jung, "User preference through learning user profile for ubiquitous recommendation systems," in *Knowledge-Based Intelligent Information and Engineering Systems*. Berlin, Germany: Springer, 2006, pp. 112–119.
[39] X. Wang, W. Pan, and C. Xu, "HGMF: Hierarchical group matrix factorization for collaborative recommendation," in *Proc. 23rd ACM Int. Conf. Conf. Inf. Knowl. Manage. (CIKM)*, Shanghai, China, Nov. 2014, pp. 769–778.
[40] E. Zhong, W. Fan, and Q. Yang, "Contextual collaborative filtering via hierarchical matrix factorization," in *Proc. 12th SIAM Int. Conf. Data Mining*, Anaheim, CA, USA, Apr. 2012, pp. 744–755.
[41] A. H. Nabizadeh, A. M. Jorge, S. Tang, and Y. Yu, "Predicting user preference based on matrix factorization by exploiting music attributes," in *Proc. ACM 9th Int. C* Conf. Comput. Sci. Softw. Eng. (C3S2E)*, New York, NY, USA, 2016, pp. 61–66.
[42] B. Ju, Y. Qian, M. Ye, R. Ni, and C. Zhu, "Using dynamic multi-task non-negative matrix factorization to detect the evolution of user preferences in collaborative filtering," *PLoS ONE*, vol. 10, no. 8, p. e0135090, Aug. 2015.
[43] X. Yang, Y. Guo, Y. Liu, and H. Steck, "A survey of collaborative filtering based social recommender systems," *Comput. Commun.*, vol. 41, pp. 1–10, Mar. 2014.
[44] L. He, Y. Jia, W. Han, and Z. Ding, "Mining user interest in microblogs with a user-topic model," *China Commun.*, vol. 11, no. 8, pp. 131–144, Aug. 2014.
[45] Z. Xu, L. Ru, L. Xiang, and Q. Yang, "Discovering user interest on Twitter with a modified author-topic model," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell. Intell. Agent Technol.*, vol. 1. Washington, DC, USA: IEEE Computer Society, 2011, pp. 422–429.
[46] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing personalized Markov chains for next-basket recommendation," in *Proc. ACM 19th Int. Conf. World Wide Web (WWW)*, New York, NY, USA, 2010, pp. 811–820.
[47] R. A. Harshman, "Foundations of the PARAFAC procedure: Models and conditions for an 'explanatory' multimodal factor analysis," in *Proc. UCLA Work. Papers Phonetics*, 1970, vol. 16, no. 1, p. 84.
[48] C. M. Bishop, "Bayesian PCA," in *Proc. Adv. Neural Inf. Process. Syst.*, 1999, pp. 382–388.
[49] M. E. Tipping and C. M. Bishop, "Probabilistic principal component analysis," *J. Roy. Statist. Soc. B, Stat. Methodol.*, vol. 61, no. 3, pp. 611–622, 1999.



RICHONG ZHANG received the B.Sc. and M.Sc. degrees from Jilin University, China, in 2001 and 2004, respectively, and the Ph.D. degree from the School of Information Technology and Engineering, University of Ottawa, in 2011. He is currently an Associate Professor with the School of Computer Science and Engineering, Beihang University. His research interests include machine learning, data mining and their applications in knowledge graph, NLP, and crowdsourcing.

YONGYI MAO received the B.E. degree from Southeast University, in 1992, the medical degree from Nanjing Medical University, in 1995, the M.Sc. degree from the Department of Medical Biophysics, University of Toronto, in 1998, and the Ph.D. degree in electrical engineering from the University of Toronto, in 2003. He joined the Faculty of the School of Information Technology and Engineering, University of Ottawa, as an Assistant Professor. He was promoted to Associate Professor, in 2008, and then to Full Professor, in 2012. His main research interests include communications and machine learning.