第 34 卷第 6 期
2017 年 6 月

控 制 理 论 与 应 用
Control Theory & Applications

Vol. 34 No. 6
Jun. 2017

DOI:10.7641/CTA.2017.60520

结合灰色预测的动态概率矩阵分解

宛袁玉, 王昌栋†, 赵知临, 赖剑煌
(中山大学 数据科学与计算机学院, 广东 广州 510006; 广东省信息安全技术重点实验室, 广东 广州 510006)

摘要: 推荐系统的目标是找出符合用户喜好的物品, 但是用户的喜好和物品的特征是动态变化的, 这种变化会影响推荐系统的准确性. 很多推荐系统只是简单的使用概率矩阵分解模型, 缺乏对这个问题的有效解决. 本文利用灰色系统理论中的灰色预测模型对用户和物品的动态性建模, 继而提出了一个基于概率矩阵分解和灰色预测模型的动态推荐系统. 首先, 利用概率矩阵分解模型生成各个连续时间窗中用户和物品的隐式向量. 接着, 利用灰色预测模型得到未来时间窗中用户和物品的隐式向量, 继而进行推荐. 实验结果说明本文的算法能够有效地对用户和商品的动态性进行建模, 且优于一些现存的最好的算法.

关键词: 推荐系统; 概率矩阵分解; 灰色预测模型
中图分类号: TP273　　**文献标识码**: A

Dynamic probabilistic matrix factorization with grey forecast

WAN Yuan-yu, WANG Chang-dong†, ZHAO Zhi-lin, LAI Jian-huang
(School of Data and Computer Science, Sun Yat-sen University, Guangzhou Guangdong, 510006, China;
Guangdong Key Laboratory of Information Security Technology, Guangzhou Guangdong, 510006, China)

Abstract: The goal of recommender system is to find out the items which meet the users' preferences. However users' preferences and items' features change over time that can affect the accuracy of recommender system. Many recommender systems simply employ probabilistic matrix factorization (PMF) model without addressing this issue. Motivated by the grey system theory, in this paper, the dynamics of both users and items are modeled by utilizing the grey forecast (GF) model. Accordingly, a new dynamic recommender system based on probabilistic matrix factorization and grey forecast model (DPMF-GF) is developed. Firstly, the probabilistic matrix factorization (PMF) model is used to produce user's and item's latent vectors between consecutive time windows. Next, the grey forecast model is used to predict user's and item's latent vectors in the following timestamp. The experimental results show that our model can effectively model users' dynamics and items' dynamics, and outperforms the existing state-of-the-art recommendation algorithms.

Key words: recommender system; probabilistic matrix factorization; grey forecast model

1 Introduction

Recommender system usually makes prediction and recommendation by analyzing historical records of users' purchasing and ratings. Collaborative filtering (CF)[1–2] is one of the most widely-used forecast technologies especially in personalized recommendation. The basic idea is that similar users would prefer similar items. But the traditional CF algorithm suffers from the sparsity problem and imbalance of rating data.

Matrix factorization technique[3] is a popular method for solving the data sparsity and imbalance problems. In general, matrix factorization technique usually learns the latent features of both users and items from the sparse user-item rating matrix by mapping both users and items to a joint low dimensional latent feature space, and then makes prediction according to user and item latent features. In fact, almost all rating matrices are imbalance and sparsity. So matrix factorization technique significantly improve the performance of recommender system. Despite great success, these methods are not able to model the change pattern of the user's preferences and item's features. However users' preferences and items' features would change over time which would seriously affect the accuracy of recommender system. Due to the uncertainty, there is still a lack of methods for modeling the evolution of users' preferences and items' features.

Grey system theory was originally proposed in 1982[4]. In many real-world practical systems, it is difficult to understand all their internal structure, param-

Received 18 July 2016; accepted 19 April 2017.
†Corresponding author. E-mail: changdongwang@hotmail.com; Tel.: +86 13450460216.
Recommended by Associate Editor YU Zhu-liang.
Supported by National Natural Science Foundation (61502543, 61573387), Guangzhou Program (201508010032), Guangdong Natural Science Funds for Distinguished Young Scholar (2016A030306014), Ph.D. Start-up Fund of Natural Science Foundation of Guangdong Province, China (2014A030310180) and Fundamental Research Funds for Central Universities (16lgzd15).

eters and features. Grey system theory mainly focuses on these systems and is widely used for conditional analysis, prediction and decision making. In this paper, the change pattern of the users' preferences and items' features is regarded as a grey system. Accordingly, we propose a dynamic recommender system based on probabilistic matrix factorization and grey forecast model (DPMF–GF). In the proposed method, the probabilistic matrix factorization (PMF) model is used to produce user's and item's latent vectors. And the grey forecast (GF) model is used to forecast the latent feature vector of both users and items. The main reasons are as follows. First, the GF model uses accumulated generation operation to build differential equations, which benefit from the correlations of the latent vectors in different time windows. Second, the GF model can achieve satisfactory results even in the case of less data so it can overcome data sparsity problem.

The contributions of this paper are summarized as follows:

1) We proposed a dynamic recommender system based on probabilistic matrix factorization and grey forecast model (DPMF–GF). Our model can model the change pattern of the users' preferences and items' features.

2) Our work is the first work that utilizes the grey system theory to overcome the weakness of traditional recommender system by modeling the change pattern of the users' preferences and items' features.

3) Extensive experiments have been conducted on several real world datasets and the results show that our model can achieve great results and outperform the existing recommender systems.

2 Related work

Non-negative matrix factorization (NMF)[5], fast maximum margin matrix factorization for collaborative prediction (MMMF)[6] and probabilistic matrix factorization (PMF)[7] are some widely used matrix factorization algorithms. The goal of these algorithms is to learn latent feature vectors from the sparse rating matrix. Many variants have been developed to improve the quality of recommendation, e.g., factored item similarity models for top-n recommender systems (FISM)[8].

However, all the above methods have not considered the change of users' preferences and items' features over time. There are only a few recommendation algorithms for addressing this issue. In [9], changes in user purchase interest are taken into consideration by computing the time weights for different items in a manner that assigns a decreasing weight to old data. But some users may have long-term interest which can't be well modeled by decreasing weight. In [10], Xiong et al. proposed a novel method called Bayesian probabilistic tensor factorization, in which the user-item-time tensor factorization is developed to model temporal effects. The downside of this method is like[11] pointed that the time dimension is a local effect and should not be compared cross all users and items. In [12], Sun et al. proposed an extension to probabilistic matrix factorization (PMF) by using Kalman filter. But their method have time-dependent and user-supplied transition parameters. It is impractical to specify such parameters for all users at all time points. In [13], Koren presented the first temporal model for the matrix factorization method by introducing time-variant biases for both users and items. But the model does not summarize the underlying pattern for the changes of biases in different windows. In [14], Zhang et al. modeled user dynamics by introducing and learning a transition matrix for each user's latent vectors between consecutive time windows. However, the transition matrix is time-invariant.

Inspired by the previous studies on probabilistic matrix factorization and grey system theory, this paper proposes a recommendation algorithm which combines probabilistic matrix factorization and grey forecast to model temporal dynamics of both user and item which can model the change pattern of the users' preferences and items' features.

3 The proposed algorithm

In this section, we will describe in detail the proposed recommendation algorithm, which integrates probabilistic matrix factorization and grey forecast to model temporal dynamics of both users and items.

3.1 Probabilistic matrix factorization

The probabilistic matrix factorization (PMF)[7] model firstly maps both users and items to a joint low dimensional latent factor space. Then ratings are computed as inner products in that space.

In our algorithm, this model is extended to the dynamic case by introducing a timestamp factor as follows. A user-item rating matrix $R_t = [r_{ijt}]_{m\times n}$ is used to represent the rating information between m users and n items at time window t. Each entry r_{ijt} in row i and column j denotes the rating of user i to item j at time window t. The PMF model factorizes the user-item rating matrix $R_t = [r_{ijt}]_{m\times n}$ into two matrices $U_t \in \mathbb{R}^{m\times l}$ and $V_t \in \mathbb{R}^{n\times l}$, with l-dimensional row vectors U_{it} and V_{jt} representing latent feature vectors of user i and item j at time window t respectively. As in [6], the conditional distribution over the observed ratings is defined as

$$p(R_t|U_t, V_t, \sigma^2) = \prod_{i=1}^{m}\prod_{j=1}^{n}[\mathcal{N}(r_{ijt}|U_{it}V_{jt}^{\mathrm{T}}, \sigma^2)]^{I_{ijt}}, \tag{1}$$

where $\mathcal{N}(x|\mu, \sigma^2)$ is the Normal distribution with mean μ and variance σ^2, and I_{ijt} is the indicator function that is equal to 1 if user i has rated item j or 0 otherwise at time window t. We also place zero-mean spherical

Gaussian priors on user and item feature vectors:

$$\begin{cases} p(U_t|\sigma_{\mathrm{U}}^2) = \prod\limits_{i=1}^{m} \mathcal{N}(U_{it}|0, \sigma_{\mathrm{U}}^2 \mathbf{I}), \\ p(V_t|\sigma_{\mathrm{V}}^2) = \prod\limits_{j=1}^{n} \mathcal{N}(V_{jt}|0, \sigma_{\mathrm{V}}^2 \mathbf{I}). \end{cases} \tag{2}$$

The log of the posterior distribution over the user and item features is given by

$$\begin{aligned} &\ln p(U_t, V_t|R_t, \sigma^2, \sigma_{\mathrm{V}}^2, \sigma_{\mathrm{U}}^2) = \\ &-\frac{1}{2\sigma^2}\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ijt}(r_{ijt}, U_{it}V_{jt}^{\mathrm{T}})^2 - \\ &\frac{1}{2\sigma_{\mathrm{U}}^2}\sum_{i=1}^{m} U_{it}U_{it}^{\mathrm{T}} - \frac{1}{2\sigma_{\mathrm{V}}^2}\sum_{j=1}^{n} V_{jt}V_{jt}^{\mathrm{T}} - \\ &\frac{1}{2}((\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ijt})\ln\sigma^2 + ml\ln\sigma_{\mathrm{U}}^2 + nl\ln\sigma_{\mathrm{V}}^2) + C, \end{aligned} \tag{3}$$

where C is a constant which does not depend on the parameters. Our goal is to find the values of U_{it}, V_{jt} that maximize the log-posterior which is equivalent to minimizing the sum-of-squared errors objective function with quadratic regularization terms:

$$\begin{aligned} E = &\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ijt}(r_{ijt} - U_{it}V_{jt}^{\mathrm{T}})^2 + \\ &\frac{\lambda_{\mathrm{U}}}{2}\sum_{i=1}^{m}\|U_{it}\|_{\mathrm{Fro}}^2 + \frac{\lambda_{\mathrm{V}}}{2}\sum_{j=1}^{n}\|V_{jt}\|_{\mathrm{Fro}}^2, \end{aligned} \tag{4}$$

where $\lambda_{\mathrm{U}} = \sigma^2/\sigma_{\mathrm{U}}^2$, $\lambda_{\mathrm{V}} = \sigma^2/\sigma_{\mathrm{V}}^2$, and $\|\cdot\|_{\mathrm{Fro}}^2$ denotes the Frobenius norm.

After learning the latent feature vectors of users and items at each time window t, we can use this information to forecast users' and items' latent feature vector in the future.

3.2 Grey forecast model

Grey forecast (GF) is an application of grey system theory which was originally proposed in 1982[4]. According to grey system theory, although the behavior of a system is hazy and its data is complex, the data is orderly in term of time and the system have overall function. The basic idea of GF is to find out law from the clutter and then make a prediction. In detail, the significance of the accumulated generating operation used in GF is to weaken the randomness and enhance the regularity. In our algorithm, the change pattern of the users preferences and items features is regarded as a grey system. And the GF model is used to forecast the latent feature vectors of both users and items. One representative GF model is called GM$(1,1)$ which indicates the model is first order and only contains one variable[4]. The GM$(1,1)$ model is widely used in dynamic forecast and has been shown to achieve great results. So we adopt the model in this paper. The general procedure for the GM$(1,1)$ model is derived as follows:

Step 1 In order to guarantee the feasibility of the method, it is necessary to do the inspection and preprocessing of the original data. Assume the original data sequence is $\boldsymbol{x_i^{(0)}}$,

$$\boldsymbol{x_i^{(0)}} = \{x_i^{(0)}(t)\},\ t = 1, 2, \cdots, w, \tag{5}$$

where $x_i^{(0)}(t)$ corresponds to the i-th dimension of latent vector at time window t and w is the length of the sequence. From Eq. (5), we can compute the level ratio of the original sequence as follows:

$$\boldsymbol{\lambda}_i = \{\lambda_i(t)\},\ t = 2, 3, \cdots, w, \tag{6}$$

where $\lambda_i(t) = x_i^{(0)}(t-1)/x_i^{(0)}(t)$. If all $\lambda_i(t) \in (\mathrm{e}^{-\frac{2}{w+1}}, \mathrm{e}^{\frac{2}{w+2}})$, the original sequence $\boldsymbol{x_i^{(0)}}$ can be directly used in the GM$(1,1)$ method. Otherwise, we need do shift transformation for the original sequence:

$$\boldsymbol{y_i^{(0)}} = \{x_i^{(0)}(t) + c\},\ t = 1, 2, \cdots, w, \tag{7}$$

such that the level ratio of sequence $\boldsymbol{y_i^{(0)}}$ must be in the range of $(\mathrm{e}^{-\frac{2}{w+1}}, \mathrm{e}^{\frac{2}{w+2}})$.

Step 2 A new sequence $\boldsymbol{x_i^{(1)}}$ is produced by the accumulated generating operation (AGO) as follows:

$$\boldsymbol{x_i^{(1)}} = \{x_i^{(1)}(t)\},\ t = 1, 2, \cdots, w, \tag{8}$$

where $x_i^{(1)}(t) = \sum\limits_{j=1}^{t} x_i^{(0)}(j),\ t = 1, 2, \cdots, w.$

Step 3 Build a first-order differential equation

$$\frac{\mathrm{d}\boldsymbol{x_i^{(1)}}}{\mathrm{d}t} + a\boldsymbol{z^{(1)}} = b, \tag{9}$$

where

$$\begin{aligned} z^{(1)}(t) = \alpha x_i^{(1)}(t) + (1-\alpha)x_i^{(1)}(t+1), \\ t = 1, 2, \cdots, w-1, \end{aligned} \tag{10}$$

with α $(0 < \alpha < 1)$ denoting a horizontal developing coefficient. In our experiments, we set $\alpha = 0.5$ according to parameter analysis.

Step 4 Solving Eq.(9), we get the forecasting model

$$\hat{x_i}^{(1)}(t+1) = (x_i^{(0)}(1) - b/a)\mathrm{e}^{-at} + b/a, \tag{11}$$

where a is the development coefficient and b is grey action, which are computed as follows

$$\begin{bmatrix} a \\ b \end{bmatrix} = (B^{\mathrm{T}}B)^{-1}B^{\mathrm{T}}Y, \tag{12}$$

where

$$B = \begin{bmatrix} -z^{(1)}(2) & 1 \\ -z^{(1)}(3) & 1 \\ \vdots & \vdots \\ -z^{(1)}(w) & 1 \end{bmatrix},\ Y = \begin{bmatrix} x_i^{(0)}(2) \\ x_i^{(0)}(3) \\ \vdots \\ x_i^{(0)}(w) \end{bmatrix}. \tag{13}$$

Step 5 Inverse accumulated generation operation (IAGO) is used to convert the prediction on data of AGO to the original data:

$$\hat{x_i}^{(0)}(t+1) = \hat{x_i}^{(1)}(t+1) - \hat{x_i}^{(1)}(t), \tag{14}$$

which can be used to get the forecasted value at the new time window.

3.3 DPMF–GF

The probabilistic matrix factorization (PMF) model[7] scales linearly with the number of observations and performs well on large, sparse and very imbalanced dataset. But in the scenario where users' preferences and items' features change over time, PMF can't capture such change which is essential for developing accurate recommender systems. In consideration of the fact that users' dynamic preferences and items' dynamic features can be regarded as grey systems, the grey forecast (GF) model can be used to forecast the latent feature vector of users and items.

So we combine PMF and GF to model both users' and items' temporal dynamics. Our algorithm has three main phases including data preprocessing, probabilistic matrix factorization and grey forecast. The detailed procedure of our algorithm is described as follows:

Phase 1 The original data are user-item rating records with time stamp. In this phase, all the original rating data are divided into w time windows according to time stamp and a user-item rating matrix $R_t = [r_{ijt}]_{m\times n}$ is generated at each time window t.

Phase 2 In this phase, PMF[7] is used to learn the latent feature vectors of users and items at each time window t by minimizing the sum-of-squared error objective function with quadratic regularization terms as follows:

$$E = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ijt}(r_{ijt} - U_{it}V_{jt}^{\mathrm{T}})^2 + \frac{\lambda_{\mathrm{U}}}{2}\sum_{i=1}^{m}\|U_{it}\|_{\mathrm{Fro}}^2 + \frac{\lambda_{\mathrm{V}}}{2}\sum_{j=1}^{n}\|V_{jt}\|_{\mathrm{Fro}}^2. \quad (15)$$

Phase 3 After the first two phases, we have learned the latent feature vector U_{jt} of user j which is a l-dimensional row vector at time window t. We get sequence $\boldsymbol{x}_{\boldsymbol{i}}^{(0)}$ from w time windows by just getting the value of the ith dimension in latent feature vector U_{jt} of user j:

$$\boldsymbol{x}_{\boldsymbol{i}}^{(0)} = \{u_{jti}\},\ t = 1,\cdots,w, \quad (16)$$

where u_{jti} represents the value of the i-th dimension in latent feature vector U_{jt} of user j at time window t. For forecasting the latent feature vector in future, we use the GM$(1,1)$ model:

$$\hat{U}_{ji} = \hat{x}_i^{(0)}(w+1) = \hat{x}_i^{(1)}(w+1) - \hat{x}_i^{(1)}(w), \quad (17)$$

where $\hat{U}_{ji}$ represents the value of the i-th dimension in the forecasted latent feature vector $\hat{U}_j$ of user j. We just need to use the same model on every user and l dimensions to get the forecasted latent feature vectors $\hat{U} \in \mathbb{R}^{m\times l}$ of all users.

Similarly, we have also learned the latent feature vector V_{jt} of item j which is a l-dimensional row vector at time window t. And we can get the forecasted latent feature vectors $\hat{V} \in \mathbb{R}^{n\times l}$ of all items. Finally, we can calculate the rating by $\hat{U}\hat{V}^{\mathrm{T}}$.

4 Experiments

In this section, we conduct extensive experiments to evaluate the prediction accuracy of our algorithm by comparing with the existing state-of-the-art algorithms. We first describe the datasets and evaluation metrics used in our experiments. And then parameter analysis experiments have been conducted to show the sensitivity of our method in different parameters. Finally, the comparison results with other state-of-the-art algorithms are reported.

To show the necessity of considering the temporal dynamics of both users' preferences and items' features, we also analyze the performances of two algorithms related with the proposed DPMF–GF.

1) DPMF–GF–U: This algorithm is similar to DPMF–GF, but it only models temporal dynamics of users' preferences. That is, when we set $\hat{V} = V_{\mathrm{w}}$ in DPMF–GF, DPMF–GF becomes DPMF–GF–U.

2) DPMF–GF–V: This algorithm is similar to DPMF–GF, but it only models temporal dynamics of items' features. That is, when we set $\hat{U} = U_{\mathrm{w}}$ in DPMF–GF, DPMF–GF becomes DPMF–GF–V.

4.1 Datasets

In our experiments, two widely tested real-world datasets are used, which are FineFoods[15] (food ratings) and Epinions[16] (product ratings) with time stamp information. We preprocess these datasets by removing the users with less than 20 ratings as in [14]. Each rating record in these datasets contains [User_ID, Item_ID, Rating, Timestamp] and all the ratings range from 1 to 5. The statistics of the preprocessed datasets are shown in Table 1.

Table 1 Datasets

Statistics	FineFoods	Epinions
# Users	7590	14077
# Items	27385	96291
# Ratings	141294	470557
Timespan	1999.10 – 2012.10	1999.3 – 2000.12

On the FineFoods dataset, $w = 10$ time windows are created by first partitioning it yearly and then merging several oldest windows into one window. On the second dataset, we partition it bimonthly to get 10 time windows. On both datasets, the last three windows for testing. After we create $w = 10$ time windows, the user-item rating matrix can be constructed according to the rating records at each time window. For each testing window, we use all the time windows prior to it as the training set. And we run the experiment five times and report the average results for reliability as in [14].

4.2 Evaluation metrics

In order to evaluate the quality of the recommendation algorithms, two widely-used evaluation metrics, namely RMSE and Recall@k, are used to measure the accuracy of the ratings predicted by our algorithm and other state-of-the-art algorithms. RMSE is the root mean square error which is defined as

$$\mathrm{RMSE} = \sqrt{\frac{1}{T}\sum_{i,j}(r_{ij} - \hat{r}_{ij})^2}, \quad (18)$$

where r_{ij} denotes the true rating user i gives to item j, $\hat{r}_{i,j}$ denotes the predicted rating user i gives to item j and T denotes the number of tested ratings. Recall@k for a user i is defined as

$$\text{Recall@}k = \frac{|N(k;i)|}{|N(i)|}, \tag{19}$$

where $|\cdot|$ denotes the number of elements in the set, $N(i)$ denotes the set of items rated by user i in the testing set and $N(k;i)$ denotes the set of items in the top-k list sorted by their predicted ratings. In order to show the global view, we report the average value of the Recall@k over all users in the testing set as in [14]. According to the definitions, the prediction quality of an algorithm is better when the algorithm has a smaller RMSE value or a higher Recall@k value.

4.3 Parameter analysis

As we have mentioned in Subsection 3.2, in Eq.(10), α $(0 < \alpha < 1)$ denotes a horizontal developing coefficient. In order to select a suitable parameter, we analyze the impact of different values of this parameter to the result in our algorithm by experiments. In the experiments of parameter analysis, we vary α from 0.1 to 0.9 while keep $k = 300$ in the top-k list and the latent dimensionality $l = 20$ without loss of generality.

Figures 1–2 plot the results in terms of RMSE as a function of α on the two datasets. And Figs.3–4 plot the results in terms of Recall@300 as a function of α on the two datasets. We observe that the performances of our algorithm and its related algorithms (DPMF–GF–U and DPMF–GF–V) are relatively stable according to the values of RMSE and Recall@300 when different α values are used. In particular, the value of RMSE becomes slightly larger when we vary α from 0.1 to 0.9. The value of Recall@300 also slightly changes but not monotonously when we vary α from 0.1 to 0.9. According to these results, our algorithm and its related algorithms (DPMF–GF–U and DPMF–GF–V) is relatively robust to the parameter α. Therefore, we use $\alpha = 0.5$ in our experiments without loss of generality.

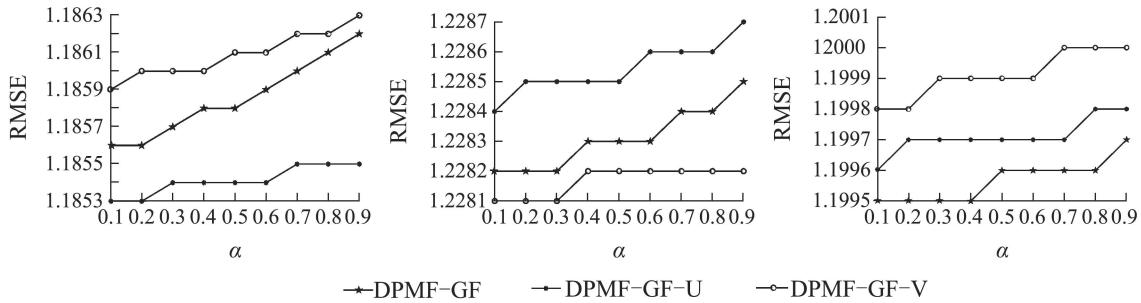

Fig. 1 RMSE on FineFoods: $l = 20$, varying α. The three figures from left to right in turn represent three testing windows

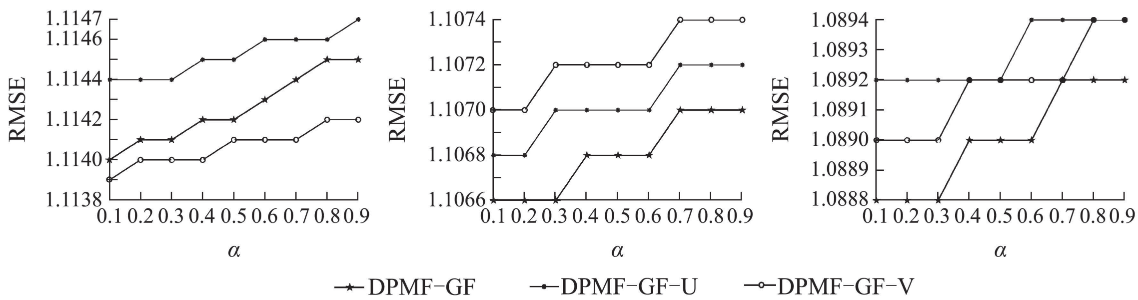

Fig. 2 RMSE on Epinions: $l = 20$, varying α. The three figures from left to right in turn represent three testing windows

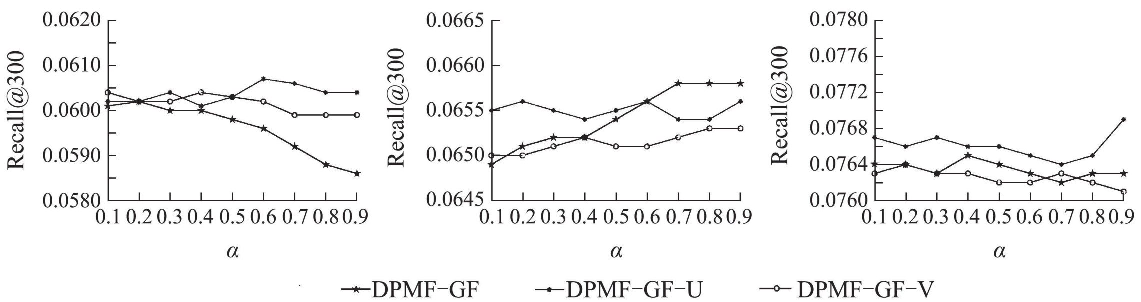

Fig. 3 Recall@300 on FineFoods: $l = 20$, varying α. The three figures from left to right in turn represent three testing windows

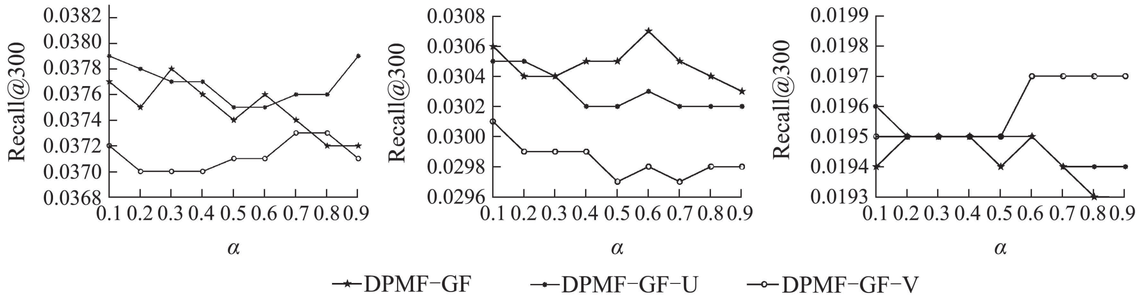

Fig. 4 Recall@300 on Epinions: $l = 20$, varying α. The three figures from left to right in turn represent three testing windows

4.4 Comparison experiments

In order to show the effectiveness of our recommendation algorithm, we compare the recommendation results with those generated by PMF[7] and TMF[14].

In our experiments, the maximum number of iterations MaxIter is set to 80 for all the matrix factorization methods. The criterion of the choice of MaxIter is these methods' objective values no longer change. We set $\mu = 0.0001$, $\lambda_{\rm U} = 0.01$, $\lambda_{\rm V} = 0.01$ in PMF, TMF, DPMF–GF, DPMF–GF–U and DPMF–GF–V, and $\lambda_B = 0.01$ in TMF. Appropriate $\lambda_{\rm U}$, $\lambda_{\rm V}$ can prevent overfitting. For the choice of μ, $\lambda_{\rm U}$, $\lambda_{\rm V}$, we pruning based on previously values in [7] and [14].

4.4.1 Comparison on RMSE

Table 2 presents the RMSE of different models on two datasets. It is easily observed that our proposed DPMF–GF algorithm is better than PMF and TMF on the two datasets. Although the improvement is not so significant, as pointed out in [13] and [14], achievable RMSE values lie in a quite compressed range and small improvements in terms of RMSE can have a significant impact on the quality of a few top presented recommendations.

Table 2 RMSE (mean ± standard error) of different models with $l = 20$. The best performer is in boldface

Algorithm	FineFoods			Epinions		
	Window 1	Window 2	Window 3	Window 1	Window 2	Window 3
PMF	1.2013±0.0008	1.2480±0.0004	1.2299±0.0003	1.1836±0.0017	1.1654±0.0013	1.1472±0.0020
TMF	1.1869±0.0006	1.2295±0.0006	1.2003±0.0004	1.1153±0.0004	1.1085±0.0002	1.0906±0.0003
DPMF–GF–U	**1.1858±0.0003**	1.2285±0.0002	1.1995±0.0005	1.1143±0.0002	1.1072±0.0001	1.0892±0.0002
DPMF–GF–V	1.1862±0.0004	1.2286±0.0005	1.1996±0.0003	**1.1142±0.0002**	1.1071±0.0003	1.0892±0.0002
DPMF–GF	**1.1858±0.0003**	**1.2283±0.0002**	**1.1995±0.0002**	**1.1142±0.0002**	**1.1071±0.0001**	**1.0890±0.0002**

4.4.2 Comparison on Recall@k

Figures 5–8 report the recall performance of different models at three testing windows on the two datasets. From left to right in turn each column is a testing window. We vary k in the top-k list from 50 to 500 and fix the latent dimensionality $l = 20$ as shown in Figs.5–6. And we vary l from 10 to 30 and fix $k = 300$ as shown in Figs.7–8. According to the results of Recall@k, we can compare our algorithm with TMF and PMF obviously. Compared with PMF which does not consider dynamics, our DPMF–GF algorithm is better on the two datasets. Compared with TMF which introduces and learns a time-invariant transition matrix for each user's latent vector, our DPMF–GF algorithm is as good as it on Epinions and performs better on FineFoods. These evidence suggest that considering the effect of dynamics helps improve the accuracy of recommender system and the grey forecast model are better than time-invariant transition matrix for modeling dynamics.

4.4.3 Comparison with both DPMF–GF–U and DPMF–GF–V

As we can see in Figs.5–8 (higher values are better), although the result of our DPMF–GF algorithm is very similar to its related algorithms (DPMF–GF–U and DPMF–GF–V), it is relatively stable in comparison with both DPMF–GF–U and DPMF–GF–V. According the result in Table 2, it is easily observed that our proposed DPMF–GF algorithm is better than its related algorithms (DPMF–GF–U and DPMF–GF–V) on the two datasets. In addition to the improvement in the mean error, the improvement in standard error shows that our DPMF–GF algorithm is better than its related algorithms DPMF–GF–U and DPMF–GF–V and more stable on the two datasets.

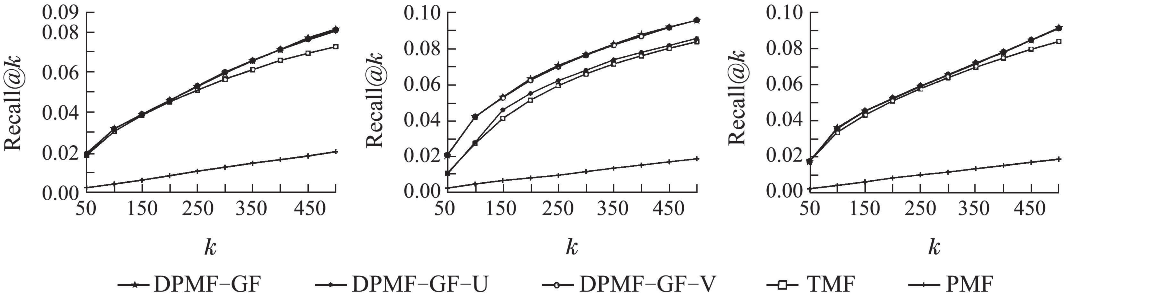

Fig. 5 Recall@k on FineFoods: $l = 20$, varying k. The three figures from left to right in turn represent three testing windows

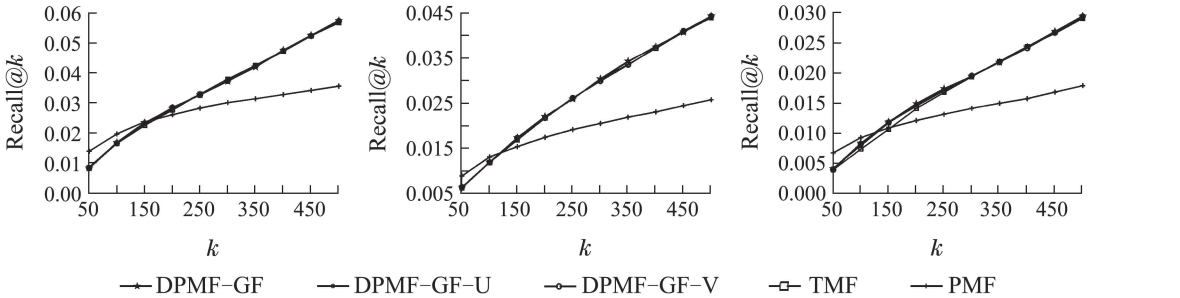

Fig. 6 Recall@k on Epinions: $l = 20$, varying k. The three figures from left to right in turn represent three testing windows

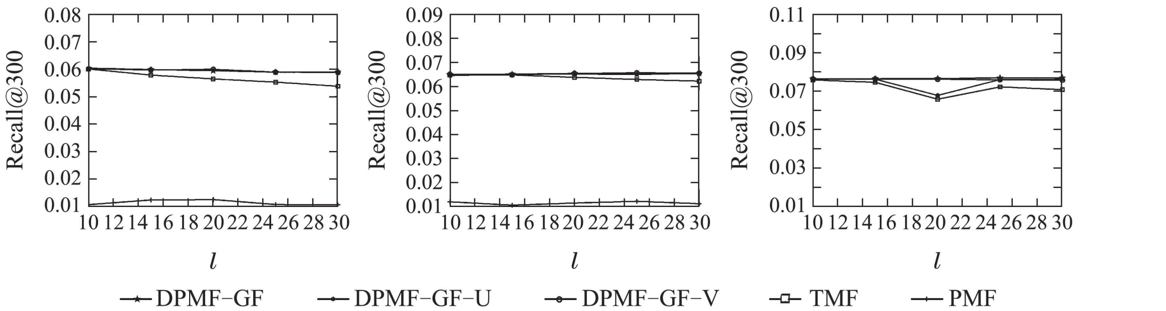

Fig. 7 Recall@300 on FineFoods: varying l. The three figures from left to right in turn represent three testing windows

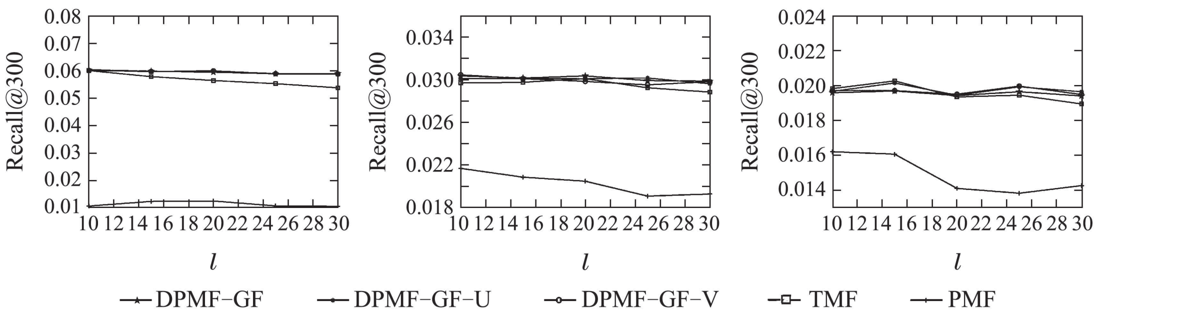

Fig. 8 Recall@300 on Epinions: varying l. The three figures from left to right in turn represent three testing windows

In summary, the proposed DPMF–GF algorithm outperforms previous models and its related algorithms in most cases. There are three reasons for this improvement. First, compared with PMF which does not consider dynamics, our algorithm introduces the grey system theory to model users' and items' dynamics and therefore gets better results. Second, although TMF introduces and learns a time-invariant transition matrix for each user's latent vector to model dynamics, yet in many cases, the patterns of the evolution for users are not time-invariant. Our proposed algorithm does not need to assume the time-invariant pattern of the evolution for users and can achieve great results even in the case of less data due to the characteristics of the grey forecast model. Third, two related algorithms of our DPMF–GF algorithm only model the change pattern of users' preferences or items' features. But both users' preferences and items' features would change over time which would seriously affect the accuracy of recommender system.

5 Conclusions and future work

Recommender system is a very popular research topic and successfully applied to plenty of real-world applications. Nevertheless, most of the existing recommender systems don't consider temporal dynamics which can affect the accuracy of recommendation.

To address this problem, in this paper, we have proposed a novel method, which combines probabilistic matrix factorization and grey forecast to model temporal dynamics of both users and items. The key idea is to model the change pattern of the users' preferences and items' features by the grey forecast model. Different from the existing dynamic models, our model introduces the grey system theory and regards the change pattern of the users' preferences and items' features as a grey system. Experiments on two real-world datasets namely FineFoods and Epinions show that our method outperforms the existing methods.

For our future work, we are interested in the cross domain recommender system with social network. This is because social network can provide auxiliary data that is beneficial to extracting features and modeling the change pattern of the features .

References:

[1] SHI Y, LARSON M, HANJALIC A. Collaborative filtering beyond the user-item matrix: a survey of the state of the art and future challenges [J]. *ACM Computing Surveys*, 2014, 47(1): 3.

[2] LIU Yueqian, YAO Hongyu. Load-forecasting model based on normalized Gaussian pLSA collaborative filtering [J]. *Control Theory & Applications*, 2008, 25(5): 929 – 932.
(刘粤钳, 姚红玉. 基于标准化高斯pLSA协同过滤的用电量预测模型 [J]. 控制理论与应用, 2008, 25(5): 929 – 932.)

[3] KOREN Y, BELL R, VOLINSKY C. Matrix factorization techniques for recommender systems [J]. *Computer*, 2009, 42(8): 30 – 37.

[4] DENG J L. Control problems of grey systems [J]. *Systems & Control Letters*, 1982, 1(5): 288 – 294.

[5] LEE D D, SEUNG H S. Algorithms for non-negative matrix factorization [C] //*Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2001: 556 – 562.

[6] RENNIE J D M, SREBRO N. Fast maximum margin matrix factorization for collaborative prediction [C] //*Proceedings of the 22nd International Conference on Machine Learning*. New York: ACM, 2005: 713 – 719.

[7] SALAKHUTDINOV R, MNIH A. Probabilistic matrix factorization [C] //*Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2011, 20: 1 – 8.

[8] KABBUR S, NING X, KARYPIS G. Fism: factored item similarity models for top-n recommender systems [C] //*Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York: ACM, 2013: 659 – 667.

[9] DING Y, LI X. Time weight collaborative filtering [C] //*Proceedings of the 14th ACM International Conference on Information and Knowledge Management*. New York: ACM, 2005: 485 – 492.

[10] RAGARAUSKAS P H, GROSS O. Temporal collaborative filtering with bayesian probabilistic tensor factorization [C] //*Proceedings of the 2010 SIAM International Conference on Data Mining*. Ohio: SIAM, 2010: 211 – 222.

[11] XIANG L, YUAN Q, ZHAO S, et al. Temporal recommendation on graphs via long-and short-term preference fusion [C] //*Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York: ACM, 2010: 723 – 732.

[12] SUN J Z, VARSHNEY K R, SUBBIAN K. Dynamic matrix factorization: a state space approach [C] //*IEEE International Conference on Acoustics, Speech and Signal Processing*. Kyoto: IEEE, 2012: 1897 – 1900.

[13] KOREN Y. Collaborative filtering with temporal dynamics [J]. *Communications of the ACM*, 2010, 53(4): 89 – 97.

[14] ZHANG C, WANG K, YU H, et al. Latent factor transition for dynamic collaborative filtering [C] //*Proceedings of the 2014 SIAM International Conference on Data Mining*. Pennsylvania: SIAM, 2014: 452 – 460.

[15] MCAULEY J J, LESKOVEC J. From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews [C] //*Proceedings of the 22nd International Conference on World Wide Web*. New York: ACM, 2013: 897 – 908.

[16] JAMALI M, ESTER M. A transitivity aware matrix factorization model for recommendation in social networks [C] //*Proceedings of the 22nd International Joint Conference on Artificial Intelligence*. Menlo Park: AAAI, 2011: 2644 – 2649.

作者简介:

宛袁玉 (1994–), 男, 硕士研究生, 主要研究方向为推荐算法与机器学习, E-mail: wanyy6@mail2.sysu.edu.cn;

王昌栋 (1984–), 男, 博士, 副教授, 硕士生导师, 主要研究方向为非线性聚类、社交网络、推荐算法, 发表学术论文50余篇, E-mail: changdongwang@hotmail.com;

赵知临 (1993–), 男, 硕士研究生, 主要研究方向为推荐算法与机器学习, E-mail: zhaozhl7@mail2.sysu.edu.cn;

赖剑煌 (1964–), 男, 博士, 教授, 博士生导师, 广东省图象图形学会理事长, 中国图象图形学会常务理事, 主要研究方向为生物特征识别、数字图像处理、模式识别和机器学习, E-mail: stsljh@mail.sysu.edu.cn.