

## PAPER

# A Novel Recommendation Algorithm Incorporating Temporal Dynamics, Reviews and Item Correlation

Ting WU<sup>†,††</sup>, Yong FENG<sup>†,††a)</sup>, JiaXing SANG<sup>†,††</sup>, BaoHua QIANG<sup>†††,††††</sup>, *Nonmembers,*  
and YaNan WANG<sup>†,††</sup>, *Student Member*

**SUMMARY** Recommender systems (RS) exploit user ratings on items and side information to make personalized recommendations. In order to recommend the right products to users, RS must accurately model the implicit preferences of each user and the properties of each product. In reality, both user preferences and item properties are changing dynamically over time, so treating the historical decisions of a user or the received comments of an item as static is inappropriate. Besides, the review text accompanied with a rating score can help us to understand why a user likes or dislikes an item, so temporal dynamics and text information in reviews are important side information for recommender systems. Moreover, compared with the large number of available items, the number of items a user can buy is very limited, which is called the sparsity problem. In order to solve this problem, utilizing item correlation provides a promising solution. Although famous methods like TimeSVD++, TopicMF and CoFactor partially take temporal dynamics, reviews and correlation into consideration, none of them combine these information together for accurate recommendation. Therefore, in this paper we propose a novel combined model called TmRevCo which is based on matrix factorization. Our model combines the dynamic user factor of TimeSVD++ with the hidden topic of each review text mined by the topic model of TopicMF through a new transformation function. Meanwhile, to support our five-scoring datasets, we use a more appropriate item correlation measure in CoFactor and associate the item factors of CoFactor with that of matrix factorization. Our model comprehensively combines the temporal dynamics, review information and item correlation simultaneously. Experimental results on three real-world datasets show that our proposed model leads to significant improvement compared with the baseline methods.

**key words:** matrix factorization, time-aware recommender systems, topic model, item correlation

## 1. Introduction

With the popularity of the Internet and the development of information technology, more and more users participate in various economic activities through internet, resulting in the flourish of many e-commerce platforms, such as eBay and Amazon. As online shopping becomes popular, the vast

amount of online products makes it an urgent and important task to develop reliable recommender systems to help customers target the information they need.

There have been several recommendation algorithms proposed, one of the most extensively investigated approaches is to model user preferences according to his/her historical choices, as well as those of the others', also known as Collaborative Filtering (CF). Recent years, the Matrix Factorization (MF) techniques become popular within recommendation algorithms, benefiting from their good scalability and prediction accuracy [7]. In its basic form, matrix factorization characterizes both items and users by vectors of factors inferred from item rating patterns. High correspondence between item and user factors leads to a recommendation. The greatest strength of matrix factorization is that it provides convenience for incorporation of additional information and we choose it as our basic model.

In reality, both user preferences and item properties are changing dynamically over time [5], [6], [16]. For example, the emergence of new items may change the focus of customers as well as the popularity of other items. The influence of those changes is global. In addition, the change can be different from person to person based on their circumstances or experience, so treating the historical decisions of a user or the received comments of an item as static, or long-term influential information sources is inappropriate [19], and it is necessary and important to model temporal dynamics at the level of each individual.

Besides, users usually evaluate a product from various aspects, while a rating score only tell us whether a user likes or dislikes an item instead why. In contrast, the review texts accompanied with rating scores can help us to uncover the reason. Most existing recommendation systems usually ignore the abundant information in review texts, and most of others treat all the reviews as a whole rather than matching the rating score and review text one-by-one to analyze users' tastes at a fine grained level.

Moreover, in comparison with the huge number of items, each user may only purchase a few items so that the user-item interaction matrix is highly sparse, making it a big challenge for basic matrix factorization. According to classical item-based collaborative filtering, the rating of a product from a user can be predicted by the ratings of the similar items. Liang et al.[8] proved that pairs of items which are often simultaneously purchased by different users are similar, they are likely to be about the same topic. But their

Manuscript received November 30, 2017.

Manuscript revised March 9, 2018.

Manuscript publicized May 18, 2018.

<sup>†</sup>The authors are with the College of Computer Science, Chongqing University, Chongqing 400030, China.

<sup>††</sup>The authors are with the Key Laboratory of Dependable Service Computing in Cyber Physical Society, Ministry of Education, Chongqing University, Chongqing 400030, China.

<sup>†††</sup>The author is with Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin 541004, China.

<sup>††††</sup>The author is with Guangxi Cooperative Innovation Center of cloud computing and Big Data, Guilin University of Electronic Technology, Guilin 541004, China.

a) E-mail: fengyong@cqu.edu.cn (Corresponding author)

DOI: 10.1587/transinf.2017EDP7387

method is mainly designed for binary rating (for example, click data).

In this paper, to tackle these issues in a practical way, we focus on fusing the temporal dynamics, reviews and item correlation on the basis of the classic recommendation model based on matrix factorization and try to precisely model user preferences and item properties. The main contributions of this paper are summarized as follows:

(i) We provide a new fusion method named TmRev to combine temporal dynamics, reviews and try to explain why a user give the score at the specific moment;

(ii) Based on TmRev, we replace the co-occurrence information of CoFactor by a more appropriate similarity information between items. Then we further associate the item factors of CoFactor with the item factors of TmRev and provide our fusion framework TmRevCo.

(iii) We evaluate the proposed model extensively on three real-world datasets and the results show that our proposed model leads to significant improvement compared with the baseline methods.

The rest of this paper is organized as follows. Section 2 introduces some related works. In Sect. 3, we first introduce classical matrix factorization model, Time-aware factor model, Topic model, Similarity measure method and then put forward our new model. Section 4 describes our experimental work with discussion of experimental results. We give some concluding reviews and directions for future research in Sect. 5.

## 2. Background and Related Work

In this section, we review several related approaches focusing on utilizing temporal data, review texts and item correlation.

Exploring the context (eg. location, time, weather, device and mood) in which users express their preferences has been proven very valuable for increasing the performance of recommendations. Among existing contextual data, time information can be considered as one of the most useful information. Chen et al.[2] presented a personalized recommendation method which discovers, stores and updates private dynamic user profiles. Chu et al.[3] proposed a feature-based bilinear regression framework for personalized recommendation on dynamic content. Hongzhi et al.[17] analysed user behaviors in social media systems and designed a temporal context-aware mixture model named TCAM, to account for the intentions and preferences behind user behaviors and capture users' changing interests. Those profiling approaches usually require specific domain knowledge which is expensive to obtain. The more general approaches are based on CF, among which the most well-known is timeSVD++[6], as demonstrated by the Netflix Prize competition, which models user preferences and item properties by bringing in different time bins and sudden drifts. Yuan et al.[18] integrated time factors into the similarity calculation of neighbors for point-of-interest recommendation and Xiang L et al.[16] used graph to explicitly model users' long

**Table 1** Important notations

Notation	Description
$N$	number of review texts
$U$	number of users
$I$	number of items
$K$	number of latent dimensions or topics
$r_{u,i}(t)$	$u$ 's rating $i$ at day $t$
$\mu$	global offset term
$b_i(t)$	bias parameter of item $i$ at day $t$
$b_u(t)$	bias parameter of user $u$ at day $t$
$q_i$	$K$ -dimensional latent factors for item $i$
$p_u(t)$	$K$ -dimensional latent factors for user $u$ at day $t$
$d_{u,i}$	review text for item $i$ by user $u$
$W$	the word-to-review matrix
$\theta_{d_{ui}}$	$K$ -dimensional topic weighting vector
$\phi_n$	$K$ -dimensional word representation vector
$W_{d_{ui}n}$	the $n$ -th word in review $d_{ui}$
$m_{i,j}$	the correlation between item $i$ and item $j$
$\omega_i, c_j$	correlation biases of item
$D$	the total number of item-item pairs
$\bar{r}_u$	the average of the $u$ -th user's rating

and short-term preference.

There also are works trying to use reviews to build models. Ganu et al.[4] used the method of manual annotation on reviews to capture aspect information to do rating prediction while more works focused on automatically identifying review dimensions [12]. Wang et al.[15] proposed supervised topic models based on Latent Dirichlet Allocation (LDA) which simultaneously considers the textual and user-item rating information. But our work takes a different approach as we attempt to correlate user factors and item factors inferred from rating matrix with the reviews. The most related works to our work are proposed by McAuley et al.[10] and Bao et al.[1], which combined latent rating user and item factors with latent review topics. However, they ignored temporal dynamic information.

In addition, there are many works attempting to combine temporal dynamic information and reviews information. Zhang et al.[19] made use of the large volume of textual reviews for the automatic extraction of domain knowledge and proposed a daily-aware personalized recommendation based on feature-level time series analysis. But it differs from ours work as they treated all reviews of a day as a whole to analyze the trend of product feature while we attempt to correlate each review with the related rating score to model changing of user tastes and item properties. McAuley et al.[11] and Subhabrata et al.[13], [14] tried to model user experience changing from reviews and led to better recommendations, which differ from ours in that they modeled experience rather than time.

Item-based CF, as one of the most classical recommendation systems, predicts the rating of an item through computing the similarity between items while ignoring the interaction of users and items. Liang et al.[8] found that pairs of items which are often simultaneously purchased by different users are similar and jointly decomposed the user-item interaction matrix and the item-item co-occurrence matrix with shared item latent factors. This model does not require

any additional information other than what is already available in the standard MF model, and it significantly improves the performance over MF models. However, their method is mainly designed for binary ratings (for example, click data).

### 3. Proposed Method

#### 3.1 Matrix Factorization: A Basic Model

The matrix factorization model, as a state-of-the-art recommender method, predicts rating score  $r_{u,i}$  for a user  $u$  and item  $i$  according to:

$$\widehat{r}_{u,i} = \mu + b_i + b_u + q_i^T p_u \quad (1)$$

Where  $\mu + b_i + b_u$  is the baseline predictor and  $\mu$  is the overall average rating,  $b_u$  and  $b_i$  indicate the average deviations of  $u$  and  $i$  from the average.  $q_i^T p_u$  indicates the interaction of  $u$  and  $i$ , and  $p_u, q_i$  are  $K$ -dimensional latent factors vector of user  $u$  and item  $i$ . Intuitively,  $p_u$  and  $q_i$  can be thought of the preferences of user and the properties of item. Matrix factorization is a classical model in recommendation systems but one disadvantage of it is that it treats all the historical behaviors as static and fails to capture the dynamic of user preferences and item properties. However, this model performs well in accuracy and scalability and it is very flexible to add side data sources for recommender. We adopt MF as a basic part of the proposed framework.

#### 3.2 TimeSVD: Integrating Ratings with Temporal Dynamics

Exploiting the temporal dynamics into recommendation systems can lead to significant improvements and the most representative method is TimeSVD++. Based on the basic MF, additional implicit feedback and temporal effects are taken into account in TimeSVD++:

$$\widehat{r}_{u,i}(t) = \mu + b_i(t) + b_u(t) + q_i^T (p_u(t) + |R(u)|^{-\frac{1}{2}} \sum_{j \in R(u)} y_j) \quad (2)$$

Since we want to highlight the impact of time, we ignore the factor  $|R(u)|^{-\frac{1}{2}} \sum_{j \in R(u)} y_j$  which indicates user's implicit information inferred from the set of items that they rated and the TimeSVD++ is simplified to TimeSVD:

$$\widehat{r}_{u,i}(t) = \mu + b_i(t) + b_u(t) + q_i^T p_u(t) \quad (3)$$

Here,  $b_i(t)$  and  $b_u(t)$  indicate the time-aware biases of item  $i$  and user  $u$  respectively,  $p_u(t)$  means that user factors are of dynamic change while item factors do not change with time as they are more static in nature. Biases  $b_i(t)$  and  $b_u(t)$  for items and users are computed as follows:

$$b_i(t) = (b_i + b_{i, \text{Bin}(t)}) * c_u(t) \quad (4)$$

$$b_u(t) = b_u + \alpha_u * dev_u(t) + b_{u,t} \quad (5)$$

It can be seen that the item bias  $b_i(t)$  is composed of a fixed part  $b_i$  and a dynamic change part  $b_{i, \text{Bin}(t)}$ , and  $c_u(t)$  is a day-specific parameter, resulting in:  $c_u(t) = c_u + c_{u,t}$ . As usual,  $c_u$

is the stable part of  $c_u(t)$ , whereas  $c_{u,t}$  represents day-specific variability. As for user bias  $b_u(t)$ ,  $b_u$  represents the fixed part,  $\alpha_u * dev_u(t)$  indicates a possible gradual drift and  $b_{u,t}$  denotes the day-specific sudden drift. Similarly, user factors are defined as:

$$p_u(t) = p_{u,k} + \alpha_{u,k} * dev_u(t) + p_{u,k,t} \quad (6)$$

Giving a training corpus of rating  $T$ , the parameters  $\Psi = \{b_u, \alpha_u, b_{u,t}, b_i, b_{i, \text{Bin}(t)}, c_u, c_{u,t}, p_{u,k}, \alpha_{u,k}, p_{u,k,t}, q_i\}$ , so that the objective function is defined as follow:

$$L_{\text{rating}} = \arg \min_{\Psi} \frac{1}{|T|} \sum_{r_{u,i}(t) \in T} (\widehat{r}_{u,i}(t) - r_{u,i}(t))^2 + \lambda \Omega(\Psi) \quad (7)$$

Where  $\Omega(\Psi)$  is the regularized part and  $\lambda$  is the regularization parameter.

#### 3.3 TopicMF: Integrating Ratings with Reviews

Review text, as a complement to a rating score, can fundamentally explain the reason why the user gives such a rating value. So, making full use of review texts can help better understand users' rating behavior. TopicMF applies topic models to uncover the hidden topic distribution in the text documents and also uses the NMF to estimate the probability distribution of each document on hidden topics and makes it possible to link each rating score and review text.

First, each review is defined as  $d_{u,i}$  (indicates the review text given to item  $i$  by user  $u$ ) and the total number of words in word dictionary, after the stop words are removed, is  $N$  ( $n \in \{1, 2, \dots, N\}$ ). Let  $W$  denotes the word-to-review matrix and  $F_{d_{u,i}n}$  as the frequency of word  $n$  in review  $d_{u,i}$ . Based on NMF, this frequency matrix can be represented by the product of matrixe  $\Theta \in \mathbb{R}^{U \times K}$  and  $\Phi \in \mathbb{R}^{N \times K}$ :

$$F \approx \Theta \Phi^T \quad (8)$$

Where  $\Theta = (\theta_{d_{u,i}k})$  denotes the distributions of topics,  $\Phi = (\phi_{nk})$  notes the distributions of word.  $\theta_{d_{u,i}k}, \phi_{nk} \geq 0$  and  $K$  is the number of factors in Matrix Factorization. The two distributions are determined by minimizing the following objective function:

$$\begin{aligned} L_{\text{review}} &= \min_{\Theta, \Phi} \|\Theta \Phi^T - W\|^2 \\ &= \min_{\Theta, \Phi} \sum_{u=1}^U \sum_{i=1}^I \sum_{n=1}^N (\theta_{d_{u,i}k} \phi_n^T - W_{d_{u,i}n})^2 \end{aligned} \quad (9)$$

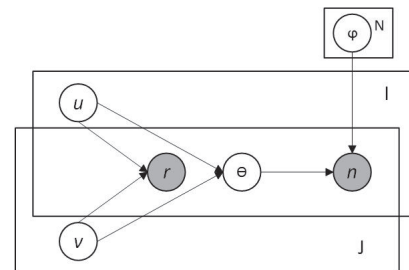


Fig. 1 TopicMF

Where  $\theta_{d_{ui}} \in \mathbb{R}^K$  is the  $ui$ -th row of  $\Theta$  and  $\phi_n \in \mathbb{R}^K$  is the  $n$ -th row of  $\Phi$ .

TopicMF, successfully links each rating score with its review text by fusing (9) and (1):

$$L = L_{rating} + L_{review} \\ = \arg \min_{\Psi, \Theta, \Phi} \frac{1}{|T|} \sum_{r_{ui} \in T} (\widehat{r}_{u,i} - r_{u,i})^2 + \|\Theta \Phi^T - W\| + \lambda \Omega(\Psi) \quad (10)$$

The goal is to model ratings accurately and obtain the most representative topics according to the review text at the same time. To fuse ratings and reviews, we use the following transformation:

$$\theta_{d_{uk}} = \frac{\exp(\omega |q_{ik} \cdot p_{uk}|)}{\sum_{k'}^K \exp(\omega |q_{ik'} \cdot p_{uk'}|)} \quad (11)$$

Where  $\omega$  is a variable indicating how likely users prefer to express their preferences or item's features. This function demonstrates the relationship of users, topics and items. It bases on the theory that users tend to talk about their preferences or items features in the reviews. We adopt TopicMF as a component of the proposed framework.

### 3.4 CoFactor: Integrating Ratings with Item Correlation

Item correlation refers to the relationship among items, and it can be inferred from the frequency of co-occurrence or the rating similarity between items. Two items are likely to be about the same topic if they have high or tight correlation. Inspired by the recent success of word embedding models, Liang et.al [8] proposed CoFactor, which jointly decomposes the user-item interaction matrix and the item-item co-occurrence matrix shared item latent factors:

$$L_{co} = \overbrace{\sum_{u,i} (r_{ui} - p_u^T q_i)^2}^{MF} + \overbrace{\sum_{m_{ij} \neq 0} (m_{ij} - q_i^T \gamma_j - \omega_i - c_j)^2}^{ItemCorrelation} \\ + \lambda_u \sum_u \|p_u\|_2^2 + \lambda_i \sum_i \|q_i\|_2^2 + \lambda_\gamma \sum_\gamma \|\gamma_j\|_2^2 \quad (12)$$

Both MF and item embedding models infer latent item representations. The difference is that the item representations inferred from MF encode users' preferences for items, while the item correlation must explain item co-occurrence patterns. Here,  $m_{ij}$  is the element in  $M \in \mathbb{R}^{I \times J}$ ,  $M$  is the co-occurrence PMI (pointwise mutual information) matrix for item consumptions.  $q_i$  is shared by both the MF and item correlation parts of the objective. The model includes  $\gamma_j$  as additional model parameter. Notice that the item embeddings  $q_i$  must account for both user-item interactions and item-item co-occurrence. PMI between item  $i$  and  $j$  can be estimated as:

$$PMI(i, j) = \log \frac{\#(i, j) \cdot D}{\#(i) \cdot \#(j)} \quad (13)$$

Where  $\#(i, j)$  is the number of users that purchased both item

$i$  and  $j$ ,  $\#(i) = \sum_j \#(i, j)$  and  $\#(j) = \sum_i \#(i, j)$ .  $D$  is the total number of item-item pairs. In CoFactor, it computes PMI based on the frequency of co-occurrence, although it can integrate ratings with item correlation for recommendation, it is unsuitable for the five-scoring rating case. So, we employ adjusted cosine similarity to deal with this problem and proposed aCoFactor(adjusted CoFactor).

There are a number of different ways to compute the similarity between items. The most basic and simple one is cosine-based similarity, in this case, two items are represented by two vectors in the  $m$  dimensional user-space. The similarity between them is measured by computing the cosine of the angle between these two vectors. Formally, similarity between items  $i$  and  $j$ , denoted by  $sim(i, j)$  is given by:

$$sim(i, j) = \cos(\vec{i}, \vec{j}) = \frac{\vec{i} \cdot \vec{j}}{\|\vec{i}\|_2 * \|\vec{j}\|_2} \quad (14)$$

Where " $\cdot$ " denotes the dot-product of the two vectors.

But computing similarity using basic cosine measure has one obvious drawback: the difference in rating preference between different users are not taken into account. So the cosine measure is more suitable for binary rating. As for our datasets, an improved method named adjusted cosine similarity can be used to overcome this drawback by subtracting the corresponding user average from each co-rated pair. Formally, the similarity between items  $i$  and  $j$  using this scheme is:

$$sim(i, j) = \frac{\sum_{u \in U} (r_{u,i} - \bar{r}_u)(r_{u,j} - \bar{r}_u)}{\sqrt{\sum_{u \in U} (r_{u,i} - \bar{r}_u)^2} \sqrt{\sum_{u \in U} (r_{u,j} - \bar{r}_u)^2}} \quad (15)$$

Here,  $\bar{r}_u$  is the average of the  $u$ -th user's rating.

We use adjusted cosine similarity method to compute the new PMI matrix.

### 3.5 TmRevCo: A Model of Ratings, Temporal Dynamics, Reviews and Item Correlation

So far, we have introduced the basic parts respectively, in view of the limitations of these basic parts, we propose our framework which jointly fuses these parts.

First, we combine the model for fusing temporal dynamics and the model for fusing reviews information. Based on TopicMF, we find that users tend to talk about the topics which they are more concerned about. However, what these topics reflect are more about users' time specific interests, and from TimeSVD, we know that users' interests is changing over time, so the topics of the review text should be corresponding with the users' time specific interests. In order to capture such correlation, we modify the transformation function in TopicMF as follows:

$$\theta_{d_{uk}} = \frac{\exp(\omega |q_{ik} \cdot p_{uk}(t)|)}{\sum_{k'}^K \exp(\omega |q_{ik'} \cdot p_{uk'}(t)|)} \quad (16)$$



Where  $p_{uk}(t)$  is the same definition as in TimeSVD. More importantly,  $\theta_{d_{ui}k}$  is actually  $\theta_{d_{ui}k}(t)$ , because  $\theta_{d_{ui}k}$  is the topic distribution of review text  $d_{ui}$  which is put at time  $t$ .

Then we can connect TimeSVD with TopicMF as our TmRev part(Time and Review) by minimizing the following objective function:

$$\begin{aligned} L &= L_{rating} + L_{review} \\ &= \arg \min_{\Psi, \Theta, \Phi} \frac{1}{|T|} \sum_{r_{ui}(t) \in T} (\widehat{r}_{u,i}(t) - r_{u,i}(t))^2 \\ &\quad + \|\Theta \Phi^T - W\|^2 + \lambda \Omega(\Psi) \end{aligned} \quad (17)$$

Secondly, on the basis of TmRev, we further integrate adjusted item correlation into the model and propose our model TmRevCo, the final objective function is as follows:

$$\begin{aligned} L &= L_{rating} + L_{review} + L_{co} \\ &= \arg \min_{\Psi, \Theta, \Phi} \frac{1}{|T|} \sum_{r_{ui}(t) \in T} (\widehat{r}_{u,i}(t) - r_{u,i}(t))^2 + \lambda_r \|\Theta \Phi^T - W\|^2 \\ &\quad + \lambda_c \sum_{m_{ij} \neq 0} (m_{ij} - q_i^T \gamma_j - \omega_i - c_j)^2 + \lambda \Omega(\Psi) + \lambda \|\gamma_j\|_2^2 \end{aligned} \quad (18)$$

Where  $\lambda_r$ ,  $\lambda_c$  are weight parameters to balance the performance of rating prediction and topic/similarity modeling,  $\lambda$  is a regularization parameter, and  $\Omega(\Psi)$  is the same definition as it in TimeSVD.

We use Gradient descent to update parameters:

$$\begin{aligned} \frac{1}{2} \frac{\partial L}{\partial p_u} &= \sum_{i=1}^I q_i (q_i^T p_u(t) + \mu + b_i(t) + b_u(t) - r_{u,i}) + \lambda p_u \\ &\quad + \sum_{i=1}^I \sum_{n=1}^N \omega (\theta_{d_{ui}} \phi_n^T - W_{d_{ui}n}) \theta_{d_{ui}} (1 - \theta_{d_{ui}}) \phi_n^T |q_i| \frac{p_u}{|p_u|} \\ \frac{1}{2} \frac{\partial L}{\partial \alpha_u} &= \sum_{u=1}^U (q_i^T p_u(t) + \mu + b_i(t) + b_u(t) - r_{u,i}) q_i dev_u(t) + \lambda \alpha_u \\ &\quad + \sum_{i=1}^I \sum_{n=1}^N \omega (\theta_{d_{ui}} \phi_n^T - W_{d_{ui}n}) \theta_{d_{ui}} (1 - \theta_{d_{ui}}) \phi_n^T |q_i| dev_u(t) \frac{\alpha_u}{|\alpha_u|} \\ \frac{1}{2} \frac{\partial L}{\partial p_{u,t}} &= \sum_{i=1}^I q_i (q_i^T p_{u,t}(t) + \mu + b_i(t) + b_u(t) - r_{u,i}) + \lambda p_{u,t} \\ &\quad + \sum_{i=1}^I \sum_{n=1}^N \omega (\theta_{d_{ui}} \phi_n^T - W_{d_{ui}n}) \theta_{d_{ui}} (1 - \theta_{d_{ui}}) \phi_n^T |q_i| \frac{p_{u,t}}{|p_{u,t}|} \\ \frac{1}{2} \frac{\partial L}{\partial q_i} &= \sum_{u=1}^U p_u (q_i^T p_u(t) + \mu + b_i(t) + b_u(t) - r_{u,i}) + \lambda q_i \\ &\quad + \sum_{i=1}^I \sum_{n=1}^N \omega (\theta_{d_{ui}} \phi_n^T - W_{d_{ui}n}) \theta_{d_{ui}} (1 - \theta_{d_{ui}}) \phi_n^T |p_u| \frac{q_i}{|q_i|} \\ &\quad + \sum_{m_{ij} \neq 0} v_j (q_i^T v_j + \omega_i + c_j - m_{ij}) \end{aligned}$$

$$\begin{aligned} \frac{1}{2} \frac{\partial L}{\partial \omega} &= \sum_{u=1}^U \sum_{i=1}^I \sum_{n=1}^N (\theta_{d_{ui}} \phi_n^T - W_{d_{ui}n}) \theta_{d_{ui}} \phi_n \\ &\quad (|q_u| |p_i| - \theta_{d_{ui}} (|q_u| |p_i|)) \end{aligned} \quad (19)$$

Due to the space limitation, the gradients of the biased terms are not listed in the paper.

The goal of our method is to simultaneously optimize the parameters associated with ratings, topics and correlations, and we solve this optimization problem by repeating the following two steps:

$$\begin{aligned} \text{update } U, V, \Theta, \omega &= \arg \min_{U, V, \Theta, \omega} L \\ \text{update } \Phi &= \arg \min_{\Phi} L_{review} \end{aligned} \quad (20)$$

## 4. Experiments

In this section, we perform experiments on three real-world datasets to investigate the efficacy of our proposed method. We first describe the datasets used in our experiments. Then we introduce five baseline methods and compare them with our method. Finally, we give an analysis of our learned model and results.

### 4.1 Datasets

Amazon is one of the largest comprehensive online shopping platforms, which owns huge transaction volume every year. McAuley et al.[10] collected a dataset that contains product reviews, time information, and metadata from Amazon, and they divided it into 26 parts based on the top-level category of each product. In particular, due to our hardware limitation, we choose three relatively small datasets of it. As we take into account the temporal dynamics, we subdivide each dataset into training and testing in chronological order, the previous 80% of each data set is used for training, and the rest is for testing. The datasets we use are shown in Table 2.

### 4.2 Preprocess

In the previous section, we describe that the review text topics are closely related to user references and item properties so that we need to build a word-to-review matrix. Before that, we do some pre-processing about the review texts. First, we adopt Stanford CoreNLP [9] to conduct word segmentation, pos tagging and stemming. After that, we remove the stop words based on the stop words list, and in order to further speed up the optimization process, we also remove the words that appear less than five times. Another preprocess we do is to calculate the correlation matrix in advance.

### 4.3 Baselines and Evaluation Metrics

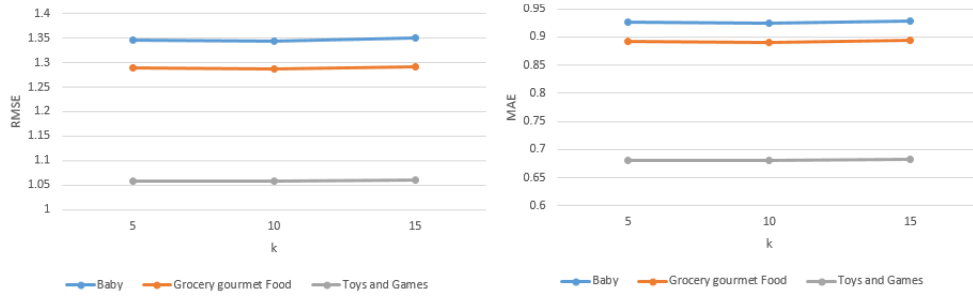
We use five baseline methods for comparison. The Probabilistic Matrix Factorization (PMF) and Latent Factor Model

**Table 2** Data description

Dataset	#users	#items	#ratings(training)	#ratings(testing)	time span
Baby	19445	7050	121555	39237	2001-02-18, 2014-07-22
Grocery, Gourmet Food	14681	8713	115682	35572	2000-08-09, 2014-07-03
Toys & Games	19412	11924	127275	40322	2000-07-28, 2014-07-23

**Table 3** Rating prediction performance

Dataset	Metric	(a) PMF	(b) LFM	(c) TimeSVD	(d) TopicMF	(e) aCoFactor	(f) TmRevCo
Baby	RMSE	1.733	1.506	1.355	1.443	1.501	<b>1.345</b>
	MAE	1.253	1.057	0.939	1.012	1.051	<b>0.927</b>
Grocery gourmet Food	RMSE	1.423	1.356	1.311	1.292	1.364	<b>1.289</b>
	MAE	1.001	0.935	0.905	0.901	0.941	<b>0.899</b>
Toys & Games	RMSE	1.421	1.179	1.073	1.128	1.172	<b>1.059</b>
	MAE	1.032	0.800	0.694	0.764	0.793	<b>0.681</b>

**Fig. 2** Performance by varying number of latent factors K

(LFM) are traditional matrix factorization models which only exploit rating information. The other three are the individual components of our model. TimeSVD takes temporal dynamics into account. TopicMF incorporates the non-negative matrix factorization with the standard matrix factorization model, correlates user latent factors and item latent factors with the review topic factors via an exponential transform function. aCoFactor brings item correlation into matrix factorization.

We evaluate these models via rating prediction performance and adopt the commonly used metrics MAE (Mean Absolute Error) and RMSE (Root Mean Square Error):

$$MAE = \frac{1}{N} \sum_{u,i} |r_{u,i}' - r_{u,i}| \quad (21)$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{u,i} (r_{u,i}' - r_{u,i})^2} \quad (22)$$

Where  $N$  is the number of instances in the testing set. The two evaluation metrics are commonly used and smaller values indicate better performance for rating prediction.

#### 4.4 Rating Prediction

Here, we show the performance comparison of our proposed TmRevCo model with all baseline methods and the results are shown in Table 3. For all methods, we choose  $K=5$  topics. For our method, the balance parameter  $\lambda_r=0.05$ ,  $\lambda_c$  is set to 0.001,  $\lambda = 0.001$  and others are fit with gradient descent.

From the table we can see that our model, which incorporates temporal dynamics, reviews information and item correlation simultaneously, achieves the best performance on the three datasets in terms of both RMSE and MAE. Though TimeSVD ignores rich information and internal connecting links between items, it fuses temporal dynamics and accurately models the short and long term change of users and items so that it gets the second best performance on Baby and Toys & Games. TimeSVD exhibits a slightly inferior performance on Grocery gourmet Food. We think this is mainly because that users' preferences in the Grocery gourmet Food dataset do not change greatly over time. TopicMF, which fully exploits the hidden information in review texts, finally achieves the third place for overall performance. The datasets we used are collected from real world and the products purchased by each user is much less than the total number of products, so it is difficult for aCoFactor to mine the correlation between products and it outputs the worst performance as a result.

#### 4.5 Parameter Sensitivity

There are three important parameters in our model: 1) the number of latent factors  $K$ ; and 2) the parameter  $\lambda_r$ , which controls the proportion of reviews; and 3) the parameter  $\lambda_c$ , which controls the proportion of item correlation. Our algorithm cannot automatically fit these parameters.

First, we set  $\lambda_r$  and  $\lambda_c$  as fixed value:  $\lambda_r = 0.05$ ,  $\lambda_c = 0.001$  and set  $K$  from  $\{5, 10, 15\}$ . As Fig. 2 shows,

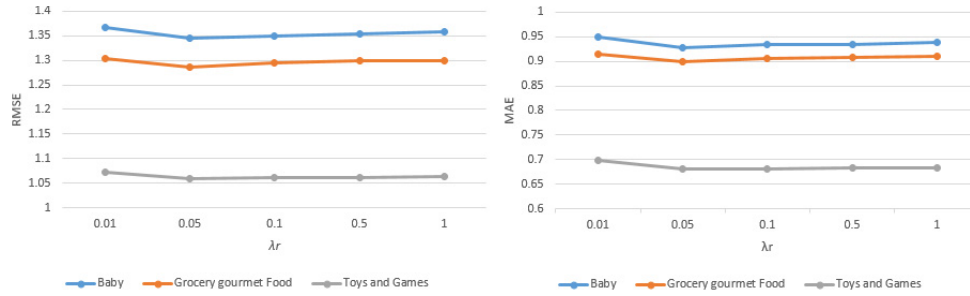


Fig. 3 Performance by varying review control parameter  $\lambda_r$

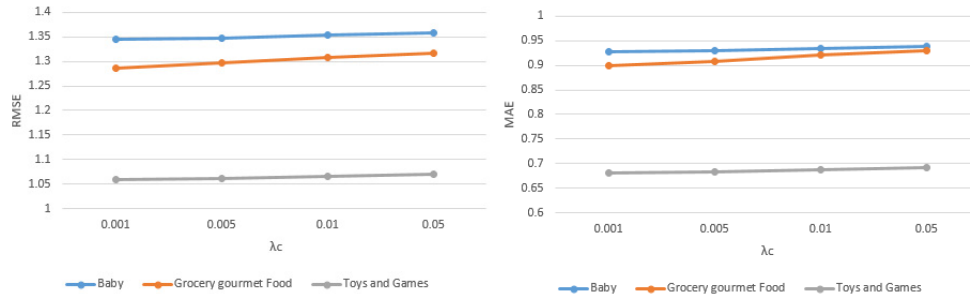


Fig. 4 Performance by varying review control parameter  $\lambda_c$

our model is stable for different values of  $K$ , which indicates our method is insensitive to different dimensions while conventional latent factor models tend to perform better as dimension increases, which is different from ours. It may be because that each review only includes a limited number of topics.

Secondly, we set  $K=5$  and  $\lambda_c = 0.001$ , set  $\lambda_r$  from  $\{0.01, 0.05, 0.1, 0.5, 1\}$  and the results are showed in Fig. 3, from which we can see that when  $\lambda_r = 0.05$ , TmRevCo gets the best performance, which may also benefit from the limited number of topics in each review text. Then we set  $K=5$  and  $\lambda_r = 0.05$ , set  $\lambda_c$  from  $\{0.001, 0.005, 0.01, 0.05\}$ , and the results are showed in Fig. 4, where we find that when  $\lambda_c = 0.001$ , we get the best performance. so we choose  $\lambda_c = 0.001$  and  $\lambda_r = 0.05$  as default.

## 5. Conclusion and Future Work

In the study of recommender systems, besides the explicit ratings, side information like temporal dynamics, reviews information and item correlation provide both opportunities and challenges. In this paper, we investigate how to fuse these three kinds information with static ratings. A unified framework named TmRevCo, which is based on matrix factorization model that factorizes user-item rating matrix into latent user and item factors for rating prediction, is proposed to fusing these three information simultaneously for precise rating prediction. We conduct experiments on three real-world datasets, each of which contains the items of a category sold on Amazon. Experimental results demonstrate that our model outperforms the state-of-the-art methods, and can lead to improved predictive performance.

The proposed model has some limitations which pro-

vides interesting directions for future work. Typically, we track the temporal dynamics of customer preferences to product by modeling the temporal dynamics along the whole time period which allows us to intelligently separate short-time factors from long-term one. But it is difficult for new-come customers. Besides, the number of hidden topics in reviews is less than the number of latent factors in ratings, therefore the assumption that these two are equal in the current model may be inappropriate under some circumstances. Moreover, there are other side information waiting to be discovered, like item brand and item descriptive information, so the issue of integrating them into our TmRevCo framework can also be included in our future works.

## Acknowledgments

Supported by National Nature Science Foundation of China (No. 61762025), Frontier and Application Foundation Research Program of CQ CSTC (No. cstc2017jcyjAX0340), The National Key Research and Development Program of China (No. 2017YFB1402400), Guangxi Key Laboratory of Trusted Software (No.kx201701), Guangxi Co-operative Innovation Center of Cloud Computing and Big Data (No.YD16E01), and Key Industries Common Key Technologies Innovation Projects of CQ CSTC (No. cstc2017zdcy-zdyxx0047), and Chongqing Postdoctoral Science Foundation (No. Xm2017125).

## References

- [1] Y. Bao, H. Fang, and J. Zhang, "Topicmf: simultaneously exploiting ratings and reviews for recommendation," Twenty-Eighth AAAI Conference on Artif. Intelli., pp.2–8, 2014.
- [2] T. Chen, W.-L. Han, H.-D. Wang, Y.-X. Zhou, B. Xu, and B.-Y.

- Zang, "Content recommendation system based on private dynamic user profile," International Conference on Machine Learning and Cybernetics, pp.2112–2118, 2007.
- [3] W. Chu and S.-T. Park, "Personalized recommendation on dynamic content using predictive bilinear models," International Conference on World Wide Web, pp.691–700, 2009.
- [4] G. Ganu, N. Elhadad, and A. Marian, "Beyond the stars: Improving rating predictions using review text content," International Workshop on the Web and Databases, WEBDB 2009, Providence, Rhode Island, USA, June, 2009.
- [5] N. Koenigstein, G. Dror, and Y. Koren, "Yahoo! music recommendations: modeling music ratings with temporal dynamics and item taxonomy," ACM Conference on Recommender Systems, pp.165–172, 2011.
- [6] Y. Koren, "Collaborative filtering with temporal dynamics," pp.447–456, 2009.
- [7] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," Computer, vol.42, no.8, pp.30–37, 2009.
- [8] D. Liang, J. Altsaar, L. Charlin, and D.M. Blei, "Factorization meets the item embedding: Regularizing matrix factorization with item co-occurrence," ACM Conference on Recommender Systems, pp.59–66, 2016.
- [9] C.D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S.J. Bethard, and D. McClosky, "The stanford corenlp natural language processing toolkit," Meeting of the Association for Computational Linguistics: System Demonstrations, pp.55–60, 2014.
- [10] J. McAuley and J. Leskovec, "Hidden factors and hidden topics: understanding rating dimensions with review text," ACM Conference on Recommender Systems, pp.165–172, 2013.
- [11] J.J. McAuley and J. Leskovec, "From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews," pp.897–908, 2013.
- [12] S. Moghaddam and M. Ester, "On the design of LDA models for aspect-based opinion mining," ACM International Conference on Information and Knowledge Management, pp.803–812, 2012.
- [13] S. Mukherjee, H. Lamba, and G. Weikum, "Experience-aware item recommendation in evolving review communities," IEEE International Conference on Data Mining, pp.925–930, 2016.
- [14] S. Mukherjee, S. Günnemann, and G. Weikum, "Continuous experience-aware language model," ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp.1075–1084, 2016.
- [15] S. Wang, F. Li, and M. Zhang, "Supervised topic model with consideration of user and item," Workshops at the Twenty-Seventh AAAI Conference on Artificial Intelligence, 2013.
- [16] L. Xiang, Q. Yuan, S. Zhao, L. Chen, X. Zhang, Q. Yang, and J. Sun, "Temporal recommendation on graphs via long- and short-term preference fusion," ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp.723–732, 2010.
- [17] H. Yin, B. Cui, L. Chen, Z. Hu, and X. Zhou, "Dynamic user modeling in social media systems," ACM Trans. Information Systems, vol.33, no.3, p.10, 2015.
- [18] Q. Yuan, G. Cong, Z. Ma, A. Sun, and N.M.-Thalmann, "Time-aware point-of-interest recommendation," International ACM SIGIR Conference on Research and Development in Information Retrieval, pp.363–372, 2013.
- [19] Y. Zhang, M. Zhang, Y. Zhang, G. Lai, Y. Liu, H. Zhang, and S. Ma, "Daily-aware personalized recommendation based on feature-level time series analysis," International Conference on World Wide Web, pp.1373–1383, 2015.



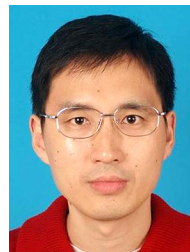
**Ting Wu** is a master student at the College of Computer Science, Chongqing University. Her research interest is Intelligent Recommendation and Neural Networks.



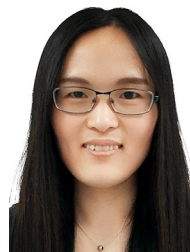
**Yong Feng** is a Professor at the College of Computer Science, Chongqing University. His research interest covers Big Data Analysis and Data Mining, Big Data Management and Intelligent Recommendation and Big Data Integration and Artificial Intelligence.



**Jia Xing Shang** is a lecturer at the College of Computer Science, Chongqing University. His research interest covers social network analysis, Data Mining, information dissemination, recommender system.



**Bao Hua Qiang** is a Professor at the Guangxi Cooperative Innovation Center of cloud computing and Big Data, Guilin University of Electronic Technology. His research interest is Big Data Processing and Information Retrieval.



**Ya Nan Wang** is a master student at the College of Computer Science, Chongqing University. Her research interest is Intelligent Recommendation and Data Mining.