2019 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)

A Deep Temporal Collaborative Filtering Recommendation Framework via Joint Learning from Long and Short-Term Effects

Qianqian Ji[†‡], Xiaoyu Shi[†*], Mingsheng Shang[†]
[†]*Chongqing Key Laboratory of Big Data and Intelligent Computing, Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing 400714, China*
[‡]*University of Chinese Academy of Sciences, Beijing 100049, China*
email: xiaoyushi@cigit.ac.cn

***Abstract*—Recommendations based on deep learning technologies are attracting increasingly attentions from academia and industry recently, due to it has the powerful ability of data representation learning, and can capture the complex and nonlinear user-item interaction patterns. Previous studies on deep learning based recommenders heavily emphasize on learning the short-term or long-term temporal effects in separate ways. Furthermore, most of the existing recommenders studied on temporal dynamics hidden in user-item interactions by using ratings or review texts solely, without utilizing these heterogeneous side information in a comprehensive manner. As a result, they own a limited ability to exploit all of available user-item interaction data. To address the above issues, we propose a temporal collaborative filtering recommendation framework with utilizing deep learning technologies, which can capture the drift of users' preferences and items' attributes over time from available heterogeneous feedback. For studying the rich sentiment information about particular item features hidden in reviews and capturing the short-term temporal dynamics in user preference, a parallel recurrent neural network (RNN) coupled in the last layer is presented. With consideration of the life cycle of each item, we further propose an item-based time evolution model as a supplement to enrich the long-term temporal effects when recommendation. In addition, a modified cosine similarity function is incorporated to produce the personalized candidate list for each user. Finally, extensive experimental are conducted on Amazon's three datasets. They indicate that the proposed method outperforms the other state-of-the-art recommenders in terms of precision and recall when doing the Top-n recommendation task.**

***Index Terms*—Recommender System, Recurrent neural network, Time evolution model, Temporal dynamics**

I. Introduction

With explosive expansion of Internet services and contents, the information overloading is a highly challenging problem in people's daily life [1]. For instance, Amazon, as one of the most popular and famous online shopping website, can push lots of products than the amount that customer can browse at once a time. Under this background, recommender system has attracted lots of attentions in the last decades. It helps customers to discovery potentially interesting products or services from massive possible options via analyzing their historical records (e.g., ratings, reviews and clicks) [2]. Therefore, it changes the traditional communication way between business and customers imperceptibly, and strengths the interaction with customers. As reported in a recent study [3], recommender system brings more then 35% extra sales revenue to Amazon, and 75% consumption to Netflix, and 60% of YouTube's browsing behaviors comes from recommendation services.

In general, the ultimate target of a recommender system is to create a recommendation list based on the user-item past iterations including explicit and implicit feedback, user preferences, item attributes and other additional information (e.g., sequence information as timestamps). Different kinds of recommenders based on variant models or methods are proposed, they mainly divided into content-based recommenders [4], collaborative filtering (CF) based recommenders [5], and hybrid based recommenders [6]. Among them, CF based recommender is the most popular and successful in industrial, since its high prediction accuracy and ease of implementation. For the earlier CF-based methods, they focus on calculating the similarity between two entries (user/item), which believe the people with similar preferences will buy the same products in the future. After the Netflix competition, CF recommenders based on the latent factor analysis (LFA) has won great successful compared to other competitors, e.g., the singular value decomposition (SVD++) model [7], the probabilistic matrix factorization model [8], the nonnegative matrix factorization model [9] and the latent factor model (LFM) [10], [11]. Regarding of LFA, it projects items and users into a low-dimensional latent factors(LFs) space, trains the target model by given data, and then estimates unknown data based on the trained model to generate recommendation results.

However, most of these LFA-based recommenders adopt the linear model to learn the complex user-item interactions, they cannot cope with the following challenges effectively.

Firstly, the traditional CF based recommender is a single

This research is partly supported by the National Natural Science Foundation of China under Grants 61602434, 916461146, and 61902370, in part by Chongqing research program of key standard technologies innovation of key industries under grant cstc2017zdcy-zdyfX0076, cstc2018jszx-cyztzxX0025,in part by the Natural Science Foundation of Chongqing (China) under grant cstc2019jcyj-msxmX0578, in part by in part by Youth Innovation Promotion Association CAS, No.2017393. [*]Corresponding Author is Xiaoyu Shi

978-1-7281-4328-6/19/$31.00 ©2019 IEEE
DOI 10.1109/ISPA-BDCloud-SustainCom-SocialCom48970.2019.00139

and isomorphic model that can not effectively handle diverse types of data, since it solely depends on the user-item ratings as user feedback information in most scenarios. However, the interactions between user and item are diverse. For example, in e-commerce websites, user's feedback behaviors include search, browse, click, collect, purchase and etc.. Hence, how to fuse these diverse and heterogeneous feedback information in a depth way is a technical difficulty for recommender system.

Most importantly, CF based recommenders generally consider the user as a static entity whose interests are fixed in time. In fact, users' preferences are dynamic change over time, rather than in a static and unchanged way. The dynamic change can be shown in two folds. 1). The user's interesting is drift over time, which is well demonstrated by previous studies [12]. Taking music recommendation for example, a user is a fan of popular music for a long time, while he/she prefers to listen classic music in recently, due to maturity or a change in the family structure. 2). Each item has its life cycle. In movie recommendation, the popularity of several movies (e.g, popcorn movies) are short, while some movies with serious dramatic content or intellectual depth always can keep a long popularity. Thus, how to effectively explore the dynamics changes in user and item are important for improving recommender performance.

Recently, deep learning technologies have achieved enormous development in the filed of natural language processing [13], speech recognition [14], and computer version [15]. Depending on the powerful nonlinear transformation and representation learning abilities of deep neural network, several deep learning models have applied into recommender system for learning the complex and nonlinear temporal dynamics in user-item interactions. In details, the RNN-based recommender shows the superiority with respect to the effect of predicting which item will the user consume next. However, these RNN-based methods focus on representing the temporal dynamics of users' behaviors in a short-term. The short-term models mainly focus on predicting the immediate behavior of the user (i.e., what he will consume soon), ignoring the long-term effects (i.e., identifying which items the user will consume eventually), since some user-item interactions are changing slowly over time (e.g, music/movie recommendation).

To address this issue, it proposes a deep temporal CF framework via utilizing the ratings, reviews and time information as feedback in this paper. The proposed framework combines a RNN network and a time evolving model for capturing the full long- and short-term temporal dynamics in a comprehensive way. In addition, a modified cosine similarity function is designed to build the personal candidate pool for further improving the performance of Top-n recommendation. Specifically, the two Bi-directional Long Short-Term Memory network (LSTM) with Attention Mechanism (AM) is used to extract the latent features from the review texts, which can learn the temporal dynamics of users' preferences in the short-term. Then a item-based time evolution model is built as a supplement to enrich the long-term temporal effects when recommendation. Finally, a novel similarity function is used to build the candidate pool. Experiments on the industrial datasets (i.e., Amazon) demonstrate the superiority of our proposed recommender. In summary, the contributions of our work can be listed from three-fold:

- We present a deep temporal collaborative filtering framework that exploits full advantage of the available data. For enhancing the recommendation performance, the diverse and heterogeneous feedback information are deep fused in this framework, including timestamps, ratings, and review texts.
- We propose a hybrid recommender to learn the short-and long-term temporal dynamics hidden in feedback information. In details, we adopt the parallel Bi-directional LSTM with attention mechanism to extract the short-term temporal dynamics from complex and nonlinear user-item interactions, while the proposed time evolution model is utilized to capture the long-term temporal dynamics effects. Furthermore, an advanced similarity function is designed to enhance the recommendation performance.
- We conduct extensive experiments on three Industrial datasets. The experimental results show that the hybrid recommender we proposed has significant advantages in terms of precision and recall, compared to the other state-of-the-art models.

The rest of this paper is organized as follow. Section II summaries the related work on recommender systems. Section III introduces the proposed deep temporal collaborative filtering framework. In Section IV, we introduce the experimental setting, and the experimental results are shown in Section V. Finally, we conclude this paper in Section VI.

II. Related Work

The analysis of temporal dynamics plays a positive and important role in improving the performance of recommender system, since it can effectively capture the user's long- and short-term preferences. Therefore, how to construct recommender based on temporal pattern analysis is a research hotspot in recommender system. Previous studies have demonstrated that in CF model, they can learn the temporal dynamics characteristics of users and items by modeling it as a sequence prediction problem. Several studies adopt the time window or delay factor to punish the historical ratings according to timestamps. For example, Koren [16] proposed TimeSVD++ to model the temporal dynamics in users' history behaviors. Lo et al. [17] built a time latent factor analysis model to track the temporal dynamics of user preferences, which adopt an improved stochastic gradient descent algorithm to train user latent factors in each time cycle. Zeng [12] proposed the time model with aging factor to achieve the goal of suppressing the recommendation of outdated items. Furthermore, Zhang et al. [18] designed the hidden Markov model to learn the dynamics transformation of ratings. In order to explore the time-varying pattern of user preferences, Gultekin et al. [19] firstly applied Kalman filter to model the recommender system. However, the above mentioned solutions focus on learning the temporal dynamics from ratings solely, without exploiting the full advantages of the available data.

Recently, deep learning technologies based recommender is revolutionizing the recommendation architectures significantly and has attracted lots of attentions [20]. Zheng et al. [21] designed the DeepCoNN model that uses convolutional neural network(CNN) to learn item and user preference characteristics separately, then couples these two parallel neural networks to perform score prediction. [22] proposed the neural collaborative filtering to strength the traditional matrix factor model, based on the extra added multi-layer perceptron module. It introduces more nonlinear operations to improve prediction performance. Compared to the above mentioned methods, RNN is suitable to capture the temporal dynamics hidden in user behaviors, which adopts the loops and memories to remember the former results precisely. Liu et al. [23] designed the recurrent log-bilinear model based on RNN, it is used to analyze the user types and predict the user's next behavior. Lu et al. [24] designed a Bi-directional GRU model with attention mechanism, which fuses the review information and rating information for scoring prediction. However, these RNN based recommendation approaches focus on estimating the user's next behavior, respecting the temporal dynamics of item's attributes and user's predilection in the short term (i.e, session based recommendation). Compared with the aforementioned solutions, we propose a deep CF framework with fusing the available heterogeneous feedback data. It leverages the RNN modular and time evolution model for Top-N recommendation, with consideration of long- and short-term effects between user-item interactions in a fine-grained way.

III. Deep temporal collaborative filtering framework

In this part, the deep temporal CF framework is described firstly from a high level, then we introduce the technical details of each components in the proposed framework, respectively.

A. Framework Overview

Generally, rating prediction and Top-n recommendation are the two representative tasks in recommender system. The former one focuses on predicting the unknown rating of a specific item-user pair, while the latter one considers to provide a specific users with a list of potentially interesting items. In this paper, we focus on the latter one.

As analyze in the above section, we leverage deep learning based collaborative filtering framework to automatically learn the optimal recommendation, which can effectively fuse the heterogeneous feedbacks to train the recommendation model continuously, and produce the optimal recommendation list with consideration of long- and short-term temporal effects. The proposed framework consists of three key components: a Bi-directional Long Short-Term Memory (BiLSTM) network with attention mechanism (AM), a item-based time evolution model and a candidates pool. Fig.1 illustrates the overview of our proposed recommendation framework and its dataflow. First, we use the review texts and corresponding rating as the input and output of BiLSTM network, respectively. In detail, we split the review texts into two categories from the perspective of users and items respectively. Thus, two BiLSTM networks are utilized to capture the deep features hidden in the reviews, both of them work in a parallel way. After that, the estimated rating of a specific user-item pair is calculated by full connection operation. Furthermore, the rating with timestamps is used as the input for the item-based time evolution model, and the weight of long-term effects is calculated by the time evolution model. Furthermore, the final prediction rating with consideration of long- and short-effects is calculated by using the estimated rating and long-term weight. Finally, the candidate pool is created by utilizing the advanced item-item cosine similarity function.

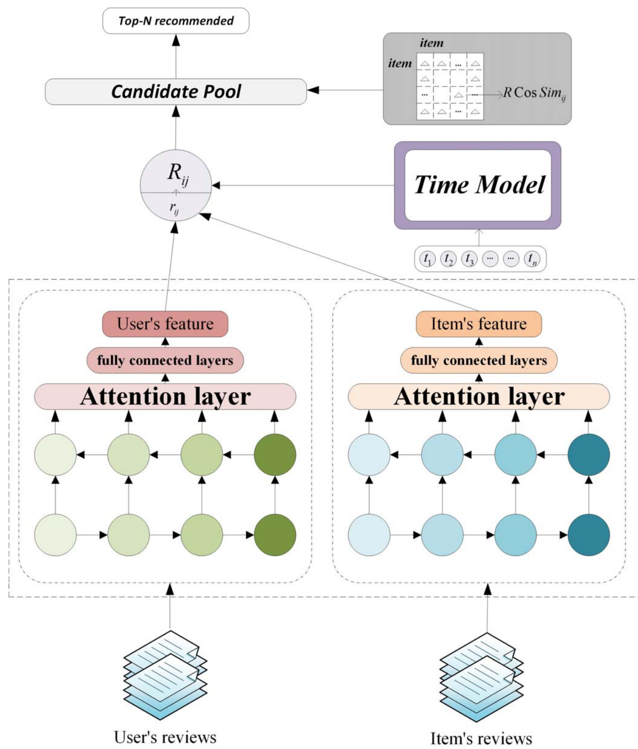

Fig. 1. Deep temporal collaborative filtering recommendation framework

B. Bi-directional LSTM with Attention Mechanism for Review Texts

The BiLSTM network with AM component focuses on capturing the short-term temporal dynamics hidden in the rich sentiment information. The main purpose of using BiLSTM is to learn hidden user and item features from review texts, and the feature fusion is performed in the last layer for estimated rating. As shown in Fig.2, it includes four layers: (i) embedding layer; (ii) sequence layer; (iii) attenation layer; (iv) feature project layer. The implementation details of each layer are describes as follow:

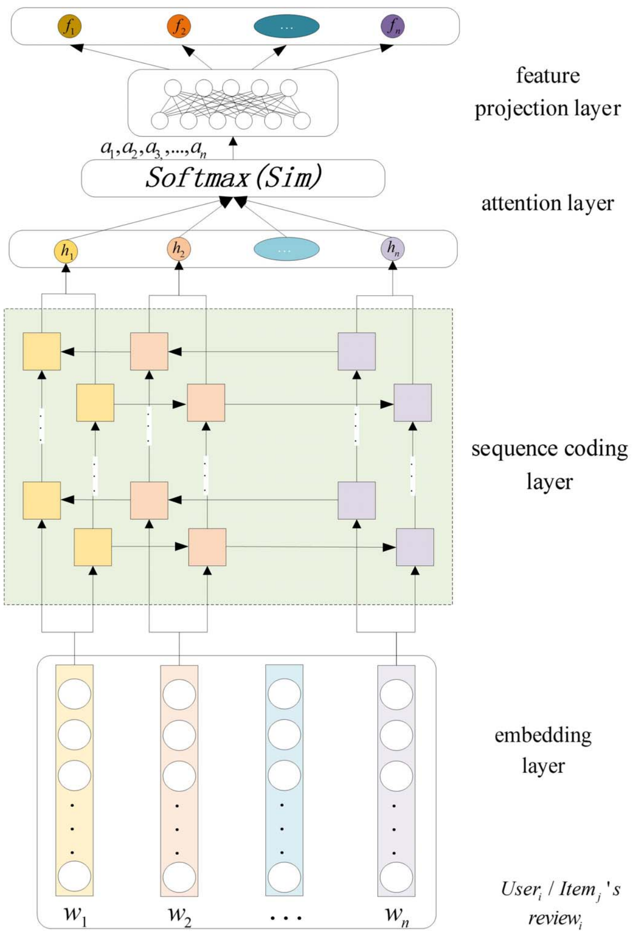

Fig. 2. Structure of Bi-directional LSTM network with Attention Mechanism.

1) Embedding layer: The function of embedding layer is to extract potential representations of users' review and items' review. D is a collection of reviews. In the word embedding layer, for any review w = $[(W_1, W_2, W_3, ..., W_n)]$, where n is the number of words in a review w. It performs the table lookup operation and retrieves from D to the m-dimensional vector x = $[(x_1, x_2, x_3, ..., x_m)]$ of the word. Hence, x represents an encoding that including word semantics and grammar. This enables the sequence encoding layer to acquire the ability to obtain contextual relevance of input information.

2) Sequence coding layer: The sequence encoding layer is designed to learn the deep hidden features of user and item. Specifically, the BiLSTM with AM is utilized in this layer.

LSTM acquires the ability to change the flow of information via a well-designed structure called a gate, enabling each loop unit to grasp the sequential dependencies of different time steps. Each LSTM unit includes cell status c_t, forgetting gate f_t, input gate i_t and output gate o_t. These gates use the sigmoid (δ) function to handle the input hidden state h_{t-1} and x_t:

$$[i_t, f_t, o_t] = \delta(W[h_{t-1}, x_t]) \tag{1}$$

Then, the cell status c_t is updated by partially forgetting the history memory $c_(t-1)$ and adding the new cell status $\widetilde{c_t}$. It can be expressed as :

$$i_t = \delta(h_{t-1}, x_t, W_i, b_i) \tag{2}$$

$$\widetilde{c_t} = tanh(h_{t-1}, x_t, W_c, b_c) \tag{3}$$

$$f_t = \delta(h_{t-1}, x_t, W_f, b_f) \tag{4}$$

$$c_t = f_t \times c_{t-1} + i_t \times \widetilde{c_t} \tag{5}$$

where W and b are the weights and bias. Finally, o_t is updated by the following equation:

$$o_t = \delta(W_o[h_{t-1}, x_t] + b_o) \tag{6}$$

After the internal update of each LSTM cell states, the hidden state h_t at time t is:

$$h_t = o_t \times tanh(c_t) \tag{7}$$

In here we leverage a bidirectional LSTM, which can divide into a forward LSTM and a backward LSTM [25]. The forward LSTM is used to get the syntax information of the sequence by reading the sequence in the normal order, while the backward LSTM loads the sequence in a reverse direction to learn the meaning of each word. In details, $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$ represents the result of forward and backward LSTM units respectively. They can express as:

$$\overrightarrow{h_t} = f(w_1 x_t + w_2 \overrightarrow{h_{t-1}}) \tag{8}$$

$$\overleftarrow{h_t} = f(w_3 x_t + w_5 \overleftarrow{h_{t+1}}) \tag{9}$$

Hence, the final output at time t is that

$$h_t = g(w_4 \overrightarrow{h_t} + w_6 \overleftarrow{h_t}) \tag{10}$$

3) Attention layer: The primary objective of introducing attention layer is to evaluate the influence on different reviews from a specific user or item. The function of this layer is to assign different weights to the extracted features from sequence coding layer.When the review sequence $H = (h_1, h_2, h_3, ...h_T)$ is obtained, the value of kth attention module att_k can be calculated by the softmax operation, it can be expressed as:

$$att_k = \sum_{t=1}^{T} a_t h_t \tag{11}$$

where a_t is the result after the softmax operation. And q is the given query parameter.

$$a_t = softmax(h \cdot q) \tag{12}$$

4) Feature projection layer: It executes a nonlinear transformation on the output feature representation att_k after the attention layer. Specifically, we adopt a single perception layer with tanh activation function. The output f_k is that:

$$f_k = tanh(W_k att_k) \tag{13}$$

where W_k is the weight.

Finally, the review can be represented by a deep latent features, i.e.,$f = [f_1, f_2, f_3, ..., f_n]$.

The predicted score of user i on item j can be calculated by:

$$\hat{r}_{i,j} = f_i^{user} \times f_j^{item} \tag{14}$$

C. Item-based Time Evolution Model

The RNN-based recommenders are good at exploring the temporal dynamics between user and item in a short-term way, which ignore the long-term effect of temporal dynamics. Hence, we leverage a time evolution model to capture the temporal dynamics from item perspective. For item j, its time effect T_j can be calculated as:

$$T_j = \frac{1}{k_j} \sum_{i \in U_j} (t_{ij} - t_j) \tag{15}$$

where k_j is the degree of item j, and U_j is the set of users who rated item j, t_{ij} represents the time when user i rated item j, t_j is the first time item j received rating.

After the LSTM network and time evolution model, the final score $\hat{r}_{i,j}$ can be obtained:

$$\hat{r}_{i,j} = r_{i,j} \times T_j \tag{16}$$

where $r_{i,j}$ is te result of LSTM network.

D. Candidate Pool

For the top-N recommendation, a candidate pool is an essential part. In CF-based recommender, we build the candidate pool by computing the similarity between each pair of items or users. Different from the existing methods (e.g.,Euclidean distance, Pearson correlation coefficient), we propose a advanced cosine function, which is mainly concerned the number of users who bought two items at the same time.

$$CosSim_{ij} = \frac{S_{ij}^2}{\sqrt{S_i^2}\sqrt{S_j^2}} \tag{17}$$

$$S_i = n_i - a(\sum_i R_i - \overline{R_i}) \tag{18}$$

Where n_i means the number of users purchasing item i, R_j is the jth rating of item i, and $\overline{R_i}$ is the mean of all evaluations of item i. a is a parameter that limits the impact of scoring information on candidate pool.

TABLE I
STATISTICS OF DATASET

Symbol	DM	B	HPC
#Us	3,524	6,759	17,525
#Is	1,517	3,446	7,182
#RWs	38,692	57,700	147,437
Avg. of reviews per each user	25.51	16.74	20.53
Avg. of reviews per each item	10.98	8.54	8.41
Sparaity	0.72%	0.24%	0.12%

IV. EXPERIMENT SETUP

A. Datasets

To evaluate the validity of our model, we conduct experiments on Amazon's public dataset, which is popular and widely used in recommender system. Each record includes the information of userId, itemId, review text, rating and timestamps. The value of rating is [1,5]. Three data sets from Amazon are selected : Digital Music (DM), Books (B), and Health and Personal Care (HPC). For each dataset, it randomly split into the subset of training, validation and test with the ratio 8:1,1. Furthermore, we choose six product categories, and exclude the users with less than 10 records. Table I shows the statistic of each dataset in detail.

For dealing with the review texts, we adopt the glove to construct the input of word embedding layer, which obtains vector representations of words by using unsupervised learning method. In this study, the Wikipedia 2014 corpus and the Gigaword 5 are adopt to train each word vector, and the varied dimension of word vector are considered. The default length of recommendation list is 10 (i.e., $n = 10$).

B. Evaluation Metric

To quantitatively evaluate our model, three performance metrics are used to evaluate the performance of tested models, i.e., Precision, Recall and F1. In general, the larger the value of the metrics is, the better the performance of the model has.

$$Recall(u) = \frac{\sum_{u \in U} R(u) \cap T(u)}{|T(u)|} \tag{19}$$

$$Precision(u) = \frac{\sum_{u \in U} R(u) \cap T(u)}{|R(u)|} \tag{20}$$

$$F1(u) = \frac{2 \times P_{ui} \times R_{ui}}{P_{ui} + R_{ui}} \tag{21}$$

Where $R(u)$ is a list of recommendations produced by recommender. $T(u)$ is the real behaviors of user u on the test set. U represents the user set.

C. Baseline Methods

In this paper, the following baseline models are adopt to evaluate the performance of the proposed recommender in a comprehensive way:

ICF [26]: It is based on the correlation analysis between the attributes of the items themselves. Its basic idea is to create a recommendation list via computing the similarity between items. It is one of the classic CF-based recommender. It only uses the ratings as the feedback information.

UCF [27]: Similar to ICF, its basic idea is to first identify and analyze users similar neighborhood based on historical data and then makes corresponding recommendations. It is also a classic CF-based recommender, which only can use ratings as the model input.

TimeSVD++ [16]: It is a singular value decomposition recommendation model that combines time series effects. The time function is used to fit the time series of user bias, project offset and user project interaction, which effectively solves the user's interest offset problem. Compared with ICF and UCF, TimeSvD++ utilize the ratings and timestamps for recommendation.

DeepCoNN [21]: The Deep Cooperative Neural Network (DeepCoNN) consists of two parallel CNN coupled in the last layer, which describes the user and item preferences separately. In DeepCoNN, it uses the reviews and ratings as feedback information.

CDAE [28]: Based on Denoising Auto-Encoders, the system filtering recommendation model CDAE (Collaborative Denoising Auto-Encoders) is proposed to complete the Top-N recommendation task based on user preference. It uses the ratings and timestamps for training the model.

V. Experimental Results

In this section, we first compare and analyze the parameters of each component of the model with varied parameters. Then the quantitatively results are given by comparing our model to the baseline methods.

A. Parameters Analysis

We first analyze the performance of our proposed framework with varied parameters under different datasets. These key parameters include the embedding dimension m and the state dimension of the sequence encoder in LSTM network. In this paper,three-layer Bi-directional LSTM are conducted in our framework. We set dropout = 0.2 for each layer of LSTM. The the vector embedding dimension and the state dimension of the sequence encoder are evaluated in this experiment, which play the important role in the performance of LSTM. Fig.3 illustrates the performance of proposed recommender with different word embedding dimensions m in LSTM. It shows that the the performance of the proposed recommender is sensitive to LSTM network with varied m. The best precision of proposed recommender is 0.105 when $m = 200$ in dataset DM. For the dataset B and HPC, the values of optimal word embedding dimension are 50 and 100, receptively. Hence, we can demonstrate that the in order to get the optimal precision, the word embedding dimension of LSTM network needs to set larger in face of dense dataset. The same case also happened in the Fig.4. Overall, the optimal result can be get in dataset DM when the word embedding dimension sets to 200 and state dimension of the sequence is 32. For dataset B, the optimal settings are $m = 50$ and state dimension of the sequence is 64. Hence, there is no universal model with fixed parameter settings for all datasets

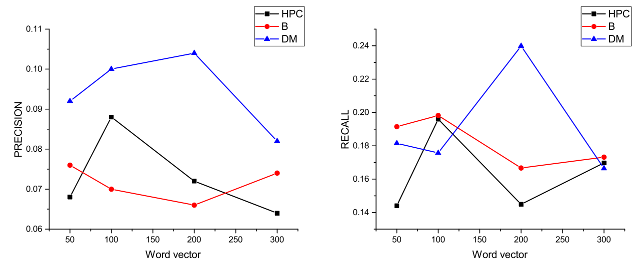

Fig. 3. Precision and Recall of proposed recommender with different word vector embedding dimension

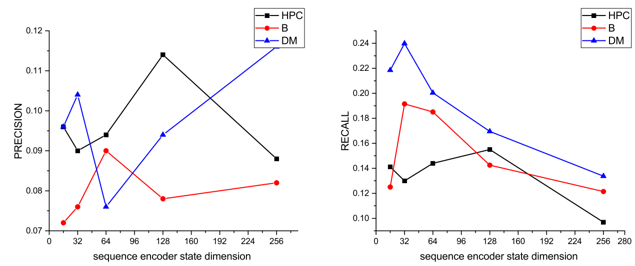

Fig. 4. Precision and Recall of proposed recommender with different the state dimension of the sequence encoder

B. Quantitative Evaluation

In this part, the performance of our hybrid recommender is analysis, compared with other state-of-the-art baselines. The summaries of the experimental results on different datasets are shown in Table II,III and IV, respectively. It demonstrates that the proposed recommender are significantly outperform other compared models in terms of precision, recall and F1. For example, in dataset DM, our proposed model has a 57.3% precision increased, compared to the optimal result of other baseline models. 49.4% and 29.8% increase for F1 and recall respectively. The same phenomenon also happened in dataset B and HPC. For dataset B and HPC, our model has a 10 times F1 increase in average, compared to the other baseline models. Furthermore, the performance of our model is the best with varying value of n. The inner reasons are that: 1). our recommender can explore the full long- and short-term temporal effects when recommendation, instead of considering the temporal effects in separate ways, e.g. the TimeSVD++ focus on capturing the long-term temporal effects, while DeepCoNN is good at learning the short-term temporal effects in current time. 2). For traditional CF models like ICF, UCF and deep learning based CDAE model, they model the complex user-item interactions by solely using the ratings. Different with those models, our framework uses the all of available heterogeneous feedback data to build the recommendation model.

TABLE II
TOP-N RECOMMENDATION RESULTS ON DATASET DM

DM		UCF	ICF	TimeSVD++	DeepCoNN	CDAE	Ours Model
n=5	R	9.58%	6.50%	1.03%	0.39%	5.75%	**9.59%**
	P	1.92%	6.61%	1.09%	0.35%	4.29%	**10.40%**
	F1	3.19%	6.55%	2.01%	0.37%	4.91%	**9.98%**
n=15	R	16.45%	12.32%	5.43%	0.58%	12.43%	**23.06%**
	P	1.33%	5.02%	1.78%	0.45%	6.08%	**10.10%**
	F1	2.42%	6.63%	2.32%	0.51%	8.15%	**13.10%**
n=15	R	16.45%	12.32%	5.43%	0.62%	16.43%	**23.06%**
	P	1.10%	4.23%	1.91%	0.56%	4.29%	**8.33%**
	F1	2.06%	6.30%	2.83%	0.5%9	6.80%	**12.24%**
n=20	R	18.48%	14.39%	6.07%	0.68%	16.79%	**23.98%**
	P	0.92%	3.68%	1.60%	0.50%	3.58%	**6.50%**
	F1	1.76%	5.86%	2.53%	0.58%	5.90%	**10.23%**

TABLE III
TOP-N RECOMMENDATION RESULTS ON DATASET B

B		UCF	ICF	TimeSVD++	DeepCoNN	CDAE	Ours Model
n=5	R	1.34%	0.56%	0.78%	0.06%	7.66%	**8.85%**
	P	0.27%	0.39%	0.55%	0.05%	5.92%	**9.00%**
	F1	0.464%	0.464%	0.65%	0.05%	6.67%	**8.92%**
n=15	R	1.89%	0.92%	1.21%	0.09%	10.72%	**16.33%**
	P	0.19%	0.32%	0.43%	0.51%	3.07%	**8.30%**
	F1	0.34%	0.47%	0.63%	0.15%	4.77%	**11.01%**
n=15	R	2.35%	1.13%	1.31%	0.17%	13.81%	**18.11%**
	P	0.16	0.26%	0.32%	0.03%	2.38%	**6.13%**
	F1	0.29%	0.43%	0.49%	0.05%	4.07%	**9.16%**
n=20	R	2.74%	1.40%	2.34%	0.42%	15.73%	**18.50%**
	P	0.14%	0.24%	0.41%	0.03%	2.87%	**4.70%**
	F1	0.26%	0.42%	0.71%	0.06%	4.85%	**7.49%**

TABLE IV
TOP-N RECOMMENDATION RESULT IN DATASET HPC

HPC		UCF	ICF	TimeSVD++	DeepCoNN	CDAE	Ours Model
n=5	R	3.05%	3.00%	0.54%	0.04%	5.80%	**7.68%**
	P	0.61%	2.47%	0.46%	0.03%	10.26%	**11.40%**
	F1	1.02%	2.70%	0.52%	0.03%	7.41%	**9.18%**
n=15	R	4.46%	4.43%	1.05%	0.72%	8.77%	**12.80%**
	P	0.45%	1.82%	0.45%	0.04%	6.13%	**9.50%**
	F1	0.81%	2.58%	0.63%	0.08%	7.21%	**10.91%**
n=15	R	5.35%	5.81%	1.54%	0.19%	10.77%	**14.28%**
	P	0.36%	1.59%	0.43%	0.05%	5.81%	**7.07%**
	F1	0.68%	2.50%	0.66%	0.08%	7.55%	**9.45%**
n=20	R	6.01%	7.09%	1.59%	0.14%	12.06%	**15.49%**
	P	0.30%	1.45%	0.34%	0.03%	4.50%	**5.75%**
	F1	0.57%	2.41%	0.56%	0.05%	6.55%	**8.38%**

VI. CONCLUSION

In this paper, the a deep temporal collaborative filtering framework based on deep learning model is proposed, which can capture the long and short term effects for Top-N recommendation. The proposed framework consists of three components, i.e., a BiLSTM network with attention mechanism, time evolution model, and candidate pool. For taking full advantage of the available and heterogeneous information, we adopt a BiLSTM network with attention mechanism to extract the temporal dynamic of user's preferences and item's attributes from review texts, while the proposed item-based time evolution model to capture the long-term temporal dynamic. Furthermore, a candidate pool is built based on the results of combining the LSTM network and time model. For enhancing the performance of personalized recommendation, a advanced cosine similarity function is proposed. Finally, we conducts extensive experiment on Amazon's three data sets, i.e, Digital Music, Books, and Health and Personal Care. We analyze the effects of proposed method with varying parameter settings. It demonstrates that our proposed model enjoys a superiority perfromance than other baseline models in terms of recommendation precision and recall.

ACKNOWLEDGMENT

This research is partly supported by the National Natural Science Foundation of China under Grants 61602434, 916461146, and 61902370, in part by Chongqing research program of key standard technologies innovation of key industries under grant cstc2017zdcy-zdyfX0076, cstc2018jszx-cyztzxX0025,in part by the Natural Science Foundation of Chongqing (China) under grant cstc2019jcyj-msxmX0578, in part by in part by Youth Innovation Promotion Association CAS, No.2017393.

REFERENCES

[1] G. Linden, B. R. Smith, and J. C. York, "Amazon.com recommendations: item-to-item collaborative filtering," *IEEE Internet Computing*, vol. 7, no. 1, pp. 76–80, 2003.

[2] A. Ahmed, C. H. Teo, S. Vishwanathan, and A. Smola, "Fair and balanced: Learning to present news stories," in *Proceedings of the fifth ACM international conference on Web search and data mining*. ACM, 2012, pp. 333–342.

[3] T. L. Keiningham, B. Cooil, L. Aksoy, T. W. Andreassen, and J. Weiner, "The value of different customer satisfaction and loyalty metrics in predicting customer retention, recommendation, and shareofwallet," *Journal of Service Theory and Practice*, vol. 17, no. 4, pp. 361–384, 2012.

[4] M. J. Pazzani and D. Billsus, "Content-based recommendation systems," in *The adaptive web*. Springer, 2007, pp. 325–341.

[5] X.-Y. Shi, X. Luo, M.-S. Shang, and X.-Y. Cai, "Empirical analysis of collaborative filtering-based recommenders in temporally evolving systems," in *2017 IEEE 14th International Conference on Networking, Sensing and Control (ICNSC)*, 2017.

[6] F. Strub, R. Gaudel, and J. Mary, "Hybrid recommender system based on autoencoders," in *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*. ACM, 2016, pp. 11–16.

[7] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2008, pp. 426–434.

[8] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," in *International Conference on Neural Information Processing Systems*, 2007.

[9] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[10] G. Takcs, I. Pilszy, B. Nmeth, and D. Tikk, "Scalable collaborative filtering approaches for large recommender systems," *Journal of Machine Learning Research*, vol. 10, no. 10, pp. 623–656, 2009.

[11] X. Luo, M. Zhou, Y. Xia, and Q. Zhu, "An efficient non-negative matrix-factorization-based approach to collaborative-filtering for recommender systems," *IEEE Transcation on Industrial Informatics*, vol. 10, no. 2, pp. 1273–1284, 2014.

[12] F. Zhang, Q. Liu, and Z. An, "Timeliness in recommender systems," *Expert Systems with Applications*, vol. 85, pp. 270–278, 2017.

[13] P. Covington, J. Adams, and E. Sargin, "Deep neural networks for youtube recommendations," in *Proceedings of the 10th ACM conference on recommender systems*. ACM, 2016, pp. 191–198.

[14] S. Nie, S. Liang, W. Liu, X. Zhang, and J. Tao, "Deep learning based speech separation via nmf-style reconstructions," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 11, pp. 2043–2055, 2018.

[15] R. Ranjan, V. M. Patel, and R. Chellappa, "Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 1, pp. 121–135, 2019.

[16] Y. Koren, "Collaborative filtering with temporal dynamics," *Communications of The ACM*, vol. 53, no. 4, pp. 89–97, 2010.

[17] Y. Lo, W. Liao, C. Chang, and Y. Lee, "Temporal matrix factorization for tracking concept drift in individual user preferences," *IEEE Transactions on Computational Social Systems*, vol. 5, no. 1, pp. 156–168, 2018.

[18] W. Zhang, H. Sun, X. Liu, and X. Guo, "Temporal qos-aware web service recommendation via non-negative tensor factorization," in *Proceedings of the 23rd International Conference on World Wide Web*, ser. WWW '14. New York, NY, USA: ACM, 2014, pp. 585–596. [Online]. Available: http://doi.acm.org/10.1145/2566486.2568001

[19] S. Gultekin and J. Paisley, "A collaborative kalman filter for time-evolving dyadic processes," in *2014 IEEE International Conference on Data Mining*. IEEE, 2014, pp. 140–149.

[20] X. Wu, X. Zhu, G. Q. Wu, and D. Wei, "Data mining with big data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 1, pp. 97–107, 2013.

[21] L. Zheng, V. Noroozi, and P. S. Yu, "Joint deep modeling of users and items using reviews for recommendation," in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. ACM, 2017, pp. 425–434.

[22] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proceedings of the 26th international conference on world wide web*. International World Wide Web Conferences Steering Committee, 2017, pp. 173–182.

[23] Q. Liu, S. Wu, and L. Wang, "Multi-behavioral sequential prediction with recurrent log-bilinear model," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1.

[24] Y. Lu, R. Dong, and B. Smyth, "Coevolutionary recommendation model: Mutual learning between ratings and reviews," in *the 2018 World Wide Web Conference*, 2018.

[25] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," *arXiv preprint arXiv:1802.05365*, 2018.

[26] B. M. Sarwar, G. Karypis, J. A. Konstan, J. Riedl *et al.*, "Item-based collaborative filtering recommendation algorithms." *Www*, vol. 1, pp. 285–295, 2001.

[27] D. Goldberg, D. A. Nichols, B. M. Oki, and D. B. Terry, "Using collaborative filtering to weave an information tapestry," *Communications of The ACM*, vol. 35, no. 12, pp. 61–70, 1992.

[28] Y. Pan, F. He, and H. Yu, "Trust-aware collaborative denoising auto-encoder for top-n recommendation," *arXiv preprint arXiv:1703.01760*, 2017.