



# A Novel Hybrid Sequential Model for Review-Based Rating Prediction

Yuanquan Lu<sup>1</sup>, Wei Zhang<sup>2</sup>(✉), Pan Lu<sup>1</sup>, and Jianyong Wang<sup>1</sup>

<sup>1</sup> Tsinghua University, Beijing, China

yuanquanlu@gmail.com, lupantech@gmail.com, jianyong@mail.tsinghua.edu.cn

<sup>2</sup> East China Normal University, Shanghai, China

zhangwei.thu2011@gmail.com

**Abstract.** Nowadays, the online interactions between users and items become diverse, and may include textual reviews as well as numerical ratings. Reviews often express various opinions and sentiments, which can alleviate the sparsity problem of recommendations to some extent. In this paper, we address the personalized review-based rating prediction problem, namely, leveraging users' historical reviews and corresponding ratings to predict their future ratings for items they have not interacted with before. While much effort has been devoted to this challenging problem mainly to investigate how to jointly model natural text and user personalization, most of them ignored sequential characteristics hidden in users' review and rating sequences. To bridge this gap, we propose a novel Hybrid Review-based Sequential Model (HRSM) to capture future trajectories of users and items. This is achieved by feeding both users' and items' review sequences to a Long Short-Term Memory (LSTM) model that captures dynamics, in addition to incorporating a more traditional low-rank factorization that captures stationary states. The experimental results on real public datasets demonstrate that our model outperforms the state-of-the-art baselines.

**Keywords:** Recommender systems · Rating prediction · Review analysis · Sequential model

## 1 Introduction

With the rapid development of the Internet, massive amounts of information spring up every day, posing both opportunities and challenges. Among many adopted techniques, recommender systems have been playing an increasingly vital role, being advantageous to alleviate information overload for ordinary users and increase sales for e-commerce companies. Particularly, in the field of recommender systems, rating prediction is a fundamental problem and has drawn much attention since the success of Netflix Prize Competition<sup>1</sup>. Given historical

---

<sup>1</sup> <https://www.netflixprize.com/>.

ratings, rating prediction is required to predict users' ratings for items they have not evaluated before.

Latent factor models [10, 13, 19] behave well and are widely applied to the rating prediction problem. The main goal of such models is to learn low dimensional vector representations for both users and items, reflecting their proximity in the corresponding latent space. Salakhutdinov et al. [19] first formulated latent factor model from a probabilistic perspective. Beyond basic latent factor models, Koren et al. [10] introduced additional user and item rating biases as new features to improve prediction. Nowadays, the online interactions between users and items become diverse, and may include textual reviews besides ratings. According to the survey [20], reviews as a kind of side information are valuable for recommender systems because of the sentiment dimension.

Review-based rating prediction problem was well formulated in the model of Hidden Factor as Topics (HFT) [15], aiming at leveraging the knowledge from ubiquitous reviews to improve rating prediction performance. As reviews can be regarded as the interactions between users and items, they contain information related to both user and item latent factors. Previous work for solving this problem could be roughly classified into two categories. One is employing topic models to generate the latent factors for users and items based on their review texts [1, 3, 14, 15, 21, 24]. Another is making use of fresh neural networks to model the semantic representation of words or sentences in the review texts [22, 25, 26]. However, most of the current review-based models mainly focus on learning semantic representations of reviews and ignore the sequential features among the reviews, which is the major focus of our work. Note that the task of review-based rating prediction is different from the task of sentiment classification. The difference is that our task focuses on leveraging users' **historical** reviews to predict their future ratings, but the sentiment classification task is to classify the **current** textual review's sentiment. Specifically, the method of learning semantic representation can be referred in the elementary component of our task.

To highlight the peculiarity of our proposed model, we first introduce the sequential models briefly, which take temporal dimension into consideration. Since preferences of users tend to vary along time and are influenced by the newly interacted items, sequential interaction history, as a kind of side information like reviews mentioned in [20], potentially serve as an important factor for predicting ratings. Apart from users, the characteristics of items might also be influenced by its recently interacted users. However, the existing methods based on matrix factorization [9] or deep neural networks [4, 5, 23], are mainly designed for mining temporal information on ratings, so that they cannot be directly employed to model the sequential features among the reviews.

From the above introduction, we can see that most of the current review-based models and sequential models only consider either review information or temporal information. To bridge this gap, we propose a novel Hybrid Review-based Sequential Model (HRSM) to capture future trajectories of users and items. The sequential information hidden in the textual reviews can help us to reveal the dynamic changes of user preferences and item characteristics. These

two kinds of side information, namely reviews and temporality, are captured simultaneously in our proposed model. Furthermore, stationary latent factors of user and item generated from latent factor model potentially keep the inherent features over a long period. We integrate these stationary states with user’s and item’s dynamic states learned from review sequences to jointly predict ratings. The key differences between our proposed model named HRSM and the representative models for comparison in the rating prediction task, including PMF [19], BMF [10], HFT [15], DeepCoNN [26], RRN [23], are summarized in Table 1.

**Table 1.** Comparison of different models.

Characteristics	PMF	BMF	HFT	DeepCoNN	RRN	HRSM
Ratings	✓	✓	✓	✓	✓	✓
Reviews			✓	✓		✓
Deep learning				✓	✓	✓
Sequences					✓	✓

In summary, the main contributions of our work are as follows.

- (1) We propose a hybrid review-based sequential model for rating prediction, which enables capturing temporal dynamics of users and items by leveraging their historical reviews.
- (2) We integrate user’s and item’s stationary latent factors with dynamic states learned from review sequences to jointly predict ratings.
- (3) Extensive experiments conducted on real public datasets demonstrate that our model outperforms the state-of-the-art baselines and obviously benefits from employing the sequential review content.

## 2 Related Work

**Document Representation.** Learning the document representation is the fundamental task of Natural Language Processing (NLP). LDA [2] as a traditional method is to learn the topic distribution from a set of documents. Based on neural networks, word2vec [17] and doc2vec [12] achieved a great success in modeling the distributed representation of words and documents, respectively. In recently years, methods employing deep learning technology outperform the previous models. Kim et al. [7] applied a convolutional layer to extract local feature among the words, and Lai et al. [11] added a recurrent structure based on it to reduce noise.

**Review-Based Model for Rating Prediction.** McAuley et al. [15] proposed the HFT model to use reviews to learn interpretable representation of users and items for review-based rating prediction problem. Many studies were inspired later, employing topic models as McAuley et al. did. TopicMF [1] as an extension of HFT, used non-negative matrix factorization for uncovering latent topics correlated with user and item factors simultaneously. Diao et al. [3] further

designed a unified framework jointly modeling aspects, ratings and sentiments of reviews. Ling et al. [14] used mixture of Gaussian instead of matrix factorization to retain the interpretability of latent topics. Tan et al. [21] proposed a rating-boosted method to integrate review features with the sentiment orientation of the user who posted it. Recently, methods under the help of neural networks perform better in review-based rating prediction. Zhang et al. [25] combined word embedding method with biased matrix factorization, and Wang et al. [22] integrated the stacked denoising autoencoders with probabilistic matrix factorization. Zheng et al. [26] designed DeepCoNN which modeled the user and item representations using review embeddings learned by Convolutional Neural Network (CNN). However, most current review-based models fail to pay attention to the sequential features among the reviews, which is the major focus of our work.

**Sequential Model for Rating Prediction.** To model the dynamics, Koren et al. [9] designed a time piecewise regression to make use of dynamic information. He et al. [5] later adopted a metric space optimization method to capture additive user-item relations in transaction sequences. Recently, Recurrent Neural Networks (RNN) based models like User-based RNN [4] and RRN [23] have been shown effective in extracting temporal features from rating sequences, leading to a further improvement in prediction. However, the existing sequential models mainly focus on rating sequences. Informative review sequences ignored by them are considered in our model.

### 3 Preliminary

#### 3.1 Problem Formulation

Assume the user set and item set are denoted as  $\mathcal{U}$  and  $\mathcal{V}$ , respectively. We further represent the rating matrix as  $\mathbf{R}$  and the collection of review text as  $\mathcal{D}$ . For  $u \in \mathcal{U}$  and  $v \in \mathcal{V}$ ,  $r_{uv} \in \mathbf{R}$  indicates the rating value which the user  $u$  assigns to the item  $v$ , while  $d_{uv} \in \mathcal{D}$  indicates the corresponding review text written by the user  $u$  to the item  $v$ . Given historical observed ratings and reviews, the problem of personalized review-based rating prediction is to predict the missed rating values in the rating matrix  $\mathbf{R}$ .

#### 3.2 Biased Matrix Factorization

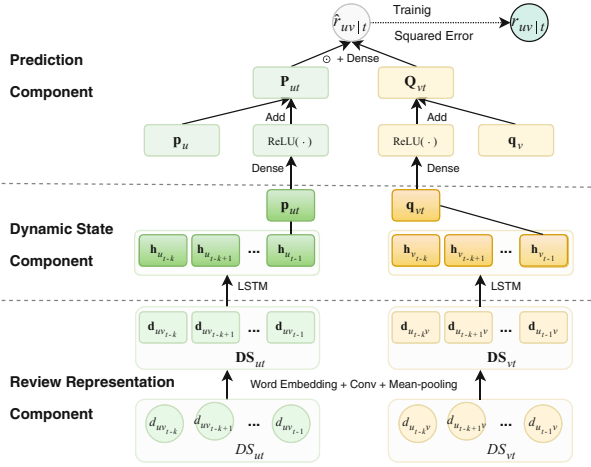
In order to verify how the temporal information and review text work, we briefly introduce a stationary model first. Biased Matrix Factorization (BMF) [10] is a collaborative filtering model for recommender systems. It is a classical and strong baseline applied in various scenes. The predicted rating  $\hat{r}_{uv}$  of the user  $u$  to the item  $v$  can be computed as:

$$\hat{r}_{uv} = \mathbf{p}_u^\top \mathbf{q}_v + b_u + b_v + g, \quad (1)$$

where  $\mathbf{p}_u$  and  $\mathbf{q}_v$  are stationary latent vectors of the user and item, respectively.  $b_u$  and  $b_v$  correspond to their rating biases, respectively, and  $g$  is the global average rating.

## 4 Proposed Methodology

In this paper, we propose a novel **Hybrid Review-based Sequential Model (HRSM)**. The overall framework of our proposed model is described in Fig. 1. Specifically, we first get each review’s representation by feeding the inside words into CNN step by step. Then LSTM [6] is employed to model the sequential property of review sequences and thus we obtain the dynamic states of users and items. We further combine the dynamic states with user and item stationary latent vectors, and train them together to make the final rating prediction.



**Fig. 1.** The architecture of the hybrid sequential model for review-based rating prediction.

### 4.1 Review Representation

As we know, reviews contain abundant information. Emotional words among reviews, such as positive or negative words, indicate the preferences shown by a user to an item. Before exploring the sequential relation among reviews, we first need to obtain the representation for each given review. Each review  $d$  ( $d = \{w_1, w_2, \dots\}$ ) consists of a certain number of words, where each word  $w \in \mathcal{W}$  comes from a vocabulary  $\mathcal{W}$ . By padding zeros in the front of review if necessary, each review could be transformed into a fixed-length matrix, with original one-hot representations for each word. After transformation by an embedding layer, each word inside the review is represented as an embedding  $\mathbf{w}$ . For each review, we adopt a convolutional layer to extract the local features and then adopt a mean-pooling layer to average the local features over its inner words. At last,

the output vector  $\mathbf{d}$  is regarded as the representation of the current input review  $d$ . The above procedures can be formulated as follows:

$$\mathbf{d} = \text{MP}(\text{CONV}(\text{EMB}(d))), \quad (2)$$

where  $\text{EMB}(\cdot)$ ,  $\text{CONV}(\cdot)$  and  $\text{MP}(\cdot)$  denote the word embedding, the convolution and the mean-pooling operations, respectively.

Note that an LSTM layer can also be applied to learn the review representation from the mention in the related work part. But in the following procedure, we employ another LSTM layer to model the review sequences. The nested structures composed of these two kinds of LSTM layer will make the whole model too complicated to perform well in the attempted trial. Therefore, we choose the CNN layer as the alternative.

## 4.2 Review-Based States of User and Item

Because the procedures of learning dynamic state representations of users and items based on their reviews are conducted in a similar fashion, we just illustrate how to model review-based state for users in more detail. In order to model the dynamic states of users, we are supposed to take the timestamps of reviews into consideration.

Assume that the current user  $u$  already has interactions with  $n$  items. After sorting the interactions by their timestamps, we get an item id sequence denoted as  $VS_u$  ( $VS_u = \{v_1, v_2, \dots, v_n\}$ ) and a review sequence denoted as  $DS_u$  ( $DS_u = \{d_{uv_1}, d_{uv_2}, \dots, d_{uv_n}\}$ ). Different from the previous studies, we model the dynamic changes of user  $u$  from its review sequence  $DS_u$  rather than item id sequence  $VS_u$ , due to the reason that reviews denote the interactions between current user  $u$  and other items, and tend to contain both user's opinion and item's characteristic simultaneously. For user  $u$  and item  $v$  at time step  $t$ , their rating is denoted as  $r_{uv|t}$ . Obviously,  $r_{uv|t}$  is only associated with the interactions before  $t$ . For a rating  $r_{uv|t}$ , the latest  $k$  interactions ( $v$  itself excluded) assigned by  $u$ , constitute a subsequence of  $DS_u$ , denoted as  $DS_{ut}$  ( $DS_{ut} = \{d_{uv_{t-k}}, d_{uv_{t-k+1}}, \dots, d_{uv_{t-1}}\}$ ). After getting review embeddings using Eq. (2), the review sequence  $DS_{ut}$  is transformed into the review embedding sequence  $\mathbf{DS}_{ut}$  ( $\mathbf{DS}_{ut} = \{\mathbf{d}_{uv_{t-k}}, \mathbf{d}_{uv_{t-k+1}}, \dots, \mathbf{d}_{uv_{t-1}}\} \in \mathbb{R}^{l \times k}$ , where  $l$  is the dimension of review embedding and  $k$  is the sequence length. Here  $k$  also means the time window size. When the time window keeps sliding over the whole review sequence  $DS_u$ , multiple  $\mathbf{DS}_{ut}$  are generated and are regarded as the input instances of the LSTM layer. To make the input sequences of LSTM have equal length, we assume that each current rating to be predicted explicitly results from its latest  $k$  interactions. The impact of parameter  $k$  will be discussed in the experiment part (see Sect. 5.5).

For the stationary model, we obtain stationary states of user and item by means of matrix factorization shown in Eq. (1). Differently, for sequential model, we apply LSTM [6] to learn the dynamic states of a user from its review sequence. At each time step  $\tau$  of the sequence  $\mathbf{DS}_{ut}$ , the hidden state  $\mathbf{h}_{u_\tau}$  of LSTM is

updated based on the current review embedding  $\mathbf{d}_{uv_\tau}$  and the previous hidden state  $\mathbf{h}_{u_{\tau-1}}$  by a transition function  $f$ . The relationship is formulated in Eq. (3). Three gates inside the function  $f$ , namely input gate, forget gate and output gate, collaboratively control how information flows through the sequence. In this way, each review among the whole review sequence is considered together, since that current review can influence all subsequent reviews when the hidden state propagates through the sequence. When we feed sequence  $\mathbf{DS}_{ut}$  into the LSTM layer, transition function  $f$  are conducted  $k$  times in total. We obtain the last hidden state as user dynamic state representation  $\mathbf{p}_{ut}$  based on its recent review sequence. This procedure is formulated in Eq. (4).

$$\mathbf{h}_{u_\tau} = f(\mathbf{d}_{uv_\tau}, \mathbf{h}_{u_{\tau-1}}), \quad (3)$$

$$\mathbf{p}_{ut} = \text{LSTM}(\mathbf{DS}_{ut}). \quad (4)$$

In a similar manner, for current item  $v$  we can obtain its review sequence  $DS_v (DS_v = \{d_{u_1v}, d_{u_2v}, \dots, d_{u_mv}\})$  consisting of reviews written by  $m$  users. For rating  $r_{uv|t}$ , item  $v$  also has a subsequence  $DS_{vt} (DS_{vt} = \{d_{u_{t-k}v}, d_{u_{t-k+1}v}, \dots, d_{u_{t-1}v}\})$  of  $DS_v$ . Note that according to the definition,  $DS_{ut}$  and  $DS_{vt}$  have not exactly the same review documents although they have the same length. After applying the LSTM layer, we can also obtain the dynamic state  $\mathbf{q}_{vt}$  of item  $v$  based on its review sequence.

### 4.3 Joint Rating Prediction

Up to now, we have obtained the dynamic states of user and item based on their review sequences. It is noted that both user and item have some inherent features that do not change with time. For example, user has fixed gender and item has stable appearance. Therefore, it is necessary to combine the dynamic and stationary states together for rating prediction. To be specific, we introduce a fully-connected layer consisting of a weight matrix  $\mathbf{W}_1$  (biases as parameter included) and a ReLU activation function [18] to map the dynamic state into the same vector space as that of the stationary state. We formulate the final states of user and item as follows

$$\mathbf{P}_{ut} = \text{ReLU}(\mathbf{W}_1^\top \mathbf{p}_{ut}) + p_u, \quad (5)$$

$$\mathbf{Q}_{vt} = \text{ReLU}(\mathbf{W}_2^\top \mathbf{q}_{vt}) + q_v, \quad (6)$$

where  $\mathbf{P}_{ut}$  denotes the joint state of user  $u$ , and  $\mathbf{Q}_{vt}$  denotes the joint state of item  $v$ .

Previous work like BMF [10], mainly illustrated in Eq. (1), simply conducts the dot product of two latent vectors to produce a scalar as predicted rating. In that case, different dimensions among the latent vectors of user and item are considered to be equally important. To improve generalization, a linear transformation using a weight matrix  $\mathbf{W}_3$  is added to distinguish significant dimensions. Finally, our model adopt the following equation to predict rating  $\hat{r}_{uv|t}$ ,

$$\hat{r}_{uv|t} = \mathbf{W}_3^\top (\mathbf{P}_{ut} \odot \mathbf{Q}_{vt}) + b_u + b_v + g, \quad (7)$$

where  $\odot$  is the hadamard product, representing the element-wise product of two vectors.

#### 4.4 Inference

We define our objective function by minimizing the regularized squared error loss between the prediction and the ground truth,

$$\min_{\theta} \sum_{(u,v,t,d) \in \mathcal{K}_{\text{train}}} (r_{uv|t} - \hat{r}_{uv|t}(\theta))^2 + \text{Reg}(\theta), \quad (8)$$

where  $\theta$  denotes all the parameters, which can be learned using backpropagation.  $(u, v, t, d)$  means each observed tuple in the training dataset  $\mathcal{K}_{\text{train}}$ , and  $\text{Reg}(\theta)$  denotes some optional regularizations.

### 5 Experiments

In this section, we describe our experimental setup and make a detailed analysis about our experimental results.

#### 5.1 Dataset

We conduct experiments based on the Amazon dataset<sup>2</sup> [16]. We generally adopt two large subsets: “CDs and Vinyl” (hereinafter called **CD**) and “Movies and TV” (hereinafter called **Movie**). The CD dataset is more related to audio term while the Movie dataset is more related to video term.

To obtain enough sequence instances, we remove users and items with less than 20 occurrences in the dataset. After filtering, the total interactions (ratings or reviews) still number over  $1 \times 10^5$  and  $4 \times 10^5$  on CD and Movie dataset, respectively. A detailed summary is shown in Table 2. As our sequential model takes the historical reviews as the input, to ensure fair comparison with other stationary models, the test set is built with the last interacted item of each user. The remaining items form the training set. Furthermore, we partition the training set with the same strategy to obtain the validation set, which is used to tune the hyper-parameters. Mean Square Error (MSE) is employed as the evaluation metric for measuring model performance.

**Table 2.** Statistics of datasets.

Datasets	#ratings/reviews	#users	#items	Density	Avg text len
CD	107,518	2,230	2,672	0.018	236
Movie	441,783	7,506	7,360	0.008	242

<sup>2</sup> <http://jmcauley.ucsd.edu/data/amazon/>.



## 5.2 Baselines

Our model HRSM is compared with three traditional and three state-of-the-art models, including GloAvg, PMF, BMF, HFT, DeepCoNN, and RRN. Specifically, the first three methods use numerical ratings, and the following two methods learn review representations with topic models or neural networks, and the last one incorporates temporal information. The differences of the comparative approaches (excluding GloAvg) are summarized in Table 1.

- **GloAvg**. GloAvg simply uses the global factor  $g$  in Eq. (1) when making predictions.
- **PMF** [19]. PMF formulates matrix factorization from a probabilistic perspective with no rating biases.
- **BMF** [10]. BMF uses matrix factorization considering additional user’s and item’s biases on the basis of PMF.
- **HFT** [15]. HFT is the classical method that combines reviews with ratings. It integrates matrix factorization with topic models, where the former learns latent factors and the later learns review parameters.
- **DeepCoNN** [26]. This is the state-of-the-art method for review-based rating prediction problem, which indistinguishably merges all reviews of each user or item into a new large document and then employs CNN to learn review representations.
- **RRN** [23]. This is the state-of-the-art sequential model for rating prediction problem, which employs LSTM to capture the dynamics by modeling user’s and item’s id sequences without considering reviews.

## 5.3 Hyper-parameter Setting

Our model is implemented in Keras<sup>3</sup>, a high-level neural network API framework. We employ Adam [8] to optimize parameters. To obtain the robust performance of our model and the compared baselines, we initialize each model with different seeds, and repeat the experiments five times, and report their average results.

Hyper-parameters are tuned in the validation sets using grid search. We apply 40-dimensional stationary latent vectors and 40-dimensional dynamic states based on reviews. Word embedding is 100-dimension and the LSTM layer contains 40 units. The batch size is set to 256 and the learning rate is set to 0.001. We use L2 regularization and its parameter is set to be  $1 \times 10^{-5}$  on CD dataset while  $1 \times 10^{-4}$  on Movie dataset. The hyper-parameters in baselines are also tuned in the similar method.

## 5.4 Results Analysis

The performances of models on two datasets are reported in Table 3. From the results, we have the following observations: (1) GloAvg is the weakest baseline,

<sup>3</sup> <https://keras.io/>.

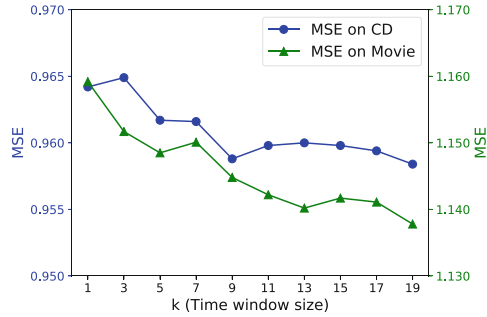
since it is a non-personalized method. Compared with PMF, BMF performs better by introducing the additional rating biases. (2) Apart from using rating matrix as PMF and BMF do, the following three methods (HFT, DeepCoNN, and RRN) consider additional information like textual review or sequential property, generally achieving better results. RRN performs poorly on Movie dataset, and the reason might be that this dataset has sparser information in user's and item's id sequences. (3) RRN and DeepCoNN are the best baselines on CD and Movie datasets, respectively. It shows approaches utilizing deep neural networks usually perform better than the other baselines. (4) Our model HRSM consistently outperforms all the baselines on two datasets. Both HRSM and RRN are deep neural networks considering the sequential information, but HRSM achieves better results, which shows that review information is complementary to ratings. Although both HRSM and DeepCoNN are deep neural networks taking textual reviews into account, our model performs better due to exploiting the sequential information in addition.

**Table 3.** Performance comparison on two datasets.

Models	CD	Movie
GloAvg	1.4956	1.6908
PMF	1.0099	1.1694
BMF	0.9905	1.1623
HFT	0.9852	1.1618
DeepCoNN	0.9812	1.1530
RRN	0.9748	1.1793
HRSM	<b>0.9584</b>	<b>1.1378</b>

### 5.5 Impact of Time Window Size

In this part, we discuss how the important parameter  $k$  influences model performance. According to the definition of  $k$ , when  $k$  increases, the input review sequence of LSTM becomes longer and the number of input instances becomes less. The shortest length of user's or item's review sequence is 20 after data preprocessing (see Sect. 5.1). To obtain the best performance of our model, we examine the impact of different time window size  $k$  from 1 to 19 on both datasets. As the results are shown in Fig. 2, we have the following observations: (1) For two datasets, MSE decreases with the increase of  $k$ . In other words, when the review sequence becomes longer, the sequential information becomes more sufficient, which leads to the better performance. (2) When  $k$  increases into the later part of the range 1–19, the performance remains stable in general. Actually, our model can obtain global sequential information to some extent because the time window keeps sliding over the whole review sequence. When  $k$  is small, the input sequence of LSTM is too short to contain enough information, resulting in the



**Fig. 2.** Impact of the varying  $k$  on two datasets.

poor performance. But when  $k$  increases to a large value, the marginal benefit for model brought by the increment of  $k$  becomes smaller. This observation can help us determine how long a sequence should be as the input instance of LSTM.

## 6 Conclusion

In this paper, we propose a novel hybrid sequential model for the personalized review-based rating prediction problem. Previous models consider either review information or temporal information. But these two kinds of side information are captured simultaneously in our proposed model. Leveraging deep neural networks, our model learns the dynamic features of users and items by exploiting the sequential property contained in their review sequences. Experimental results on real public datasets demonstrate the effectiveness of our proposed model and prove that the sequential property hidden in reviews contributes a lot in the task of rating prediction.

**Acknowledgement.** This work was supported in part by National Natural Science Foundation of China under Grant No. 61532010 and 61521002. We also thank Yifeng Zhao and Ning Liu for helpful discussions.

## References

1. Bao, Y., Fang, H., Zhang, J.: TopicMF: simultaneously exploiting ratings and reviews for recommendation. In: AAAI, pp. 2–8 (2014)
2. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. *J. Mach. Learn. Res.* **3**, 993–1022 (2003)
3. Diao, Q., Qiu, M., Wu, C.Y., Smola, A.J., Jiang, J., Wang, C.: Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS). In: SIGKDD, pp. 193–202 (2014)
4. Donkers, T., Loepp, B., Ziegler, J.: Sequential user-based recurrent neural network recommendations. In: RecSys, pp. 152–160 (2017)
5. He, R., Kang, W.C., McAuley, J.: Translation-based recommendation. In: RecSys, pp. 161–169 (2017)

6. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* **9**(8), 1735–1780 (1997)
7. Kim, Y.: Convolutional neural networks for sentence classification. In: *EMNLP*, pp. 1746–1751 (2014)
8. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. *CoRR* abs/1412.6980 (2014)
9. Koren, Y.: Collaborative filtering with temporal dynamics. In: *SIGKDD*, pp. 447–456 (2009)
10. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. *Computer* **42**(8), 30–37 (2009)
11. Lai, S., Xu, L., Liu, K., Zhao, J.: Recurrent convolutional neural networks for text classification. In: *AAAI*, pp. 2267–2273 (2015)
12. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: *ICML*, pp. II-1188–II-1196 (2014)
13. Lee, D.D., Seung, H.S.: Algorithms for non-negative matrix factorization. In: *NIPS*, pp. 535–541 (2000)
14. Ling, G., Lyu, M.R., King, I.: Ratings meet reviews, a combined approach to recommend. In: *RecSys*, pp. 105–112 (2014)
15. McAuley, J., Leskovec, J.: Hidden factors and hidden topics: understanding rating dimensions with review text. In: *RecSys*, pp. 165–172 (2013)
16. McAuley, J., Targett, C., Shi, Q., van den Hengel, A.: Image-based recommendations on styles and substitutes. In: *SIGIR*, pp. 43–52 (2015)
17. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. In: *NIPS*, pp. 3111–3119 (2013)
18. Nair, V., Hinton, G.E.: Rectified linear units improve restricted Boltzmann machines. In: *ICML*, pp. 807–814 (2010)
19. Salakhutdinov, R., Mnih, A.: Probabilistic matrix factorization. In: *NIPS*, pp. 1257–1264 (2007)
20. Shi, Y., Larson, M., Hanjalic, A.: Collaborative filtering beyond the user-item matrix: a survey of the state of the art and future challenges. *ACM Comput. Surv.* **47**(1), 3:1–3:45 (2014)
21. Tan, Y., Zhang, M., Liu, Y., Ma, S.: Rating-boosted latent topics: understanding users and items with ratings and reviews. In: *IJCAI*, pp. 2640–2646 (2016)
22. Wang, H., Wang, N., Yeung, D.Y.: Collaborative deep learning for recommender systems. In: *SIGKDD*, pp. 1235–1244 (2015)
23. Wu, C.Y., Ahmed, A., Beutel, A., Smola, A.J., Jing, H.: Recurrent recommender networks. In: *WSDM*, pp. 495–503 (2017)
24. Zhang, W., Wang, J.: Integrating topic and latent factors for scalable personalized review-based rating prediction. *IEEE Trans. Knowl. Data Eng.* **28**(11), 3013–3027 (2016)
25. Zhang, W., Yuan, Q., Han, J., Wang, J.: Collaborative multi-level embedding learning from reviews for rating prediction. In: *IJCAI*, pp. 2986–2992 (2016)
26. Zheng, L., Noroozi, V., Yu, P.S.: Joint deep modeling of users and items using reviews for recommendation. In: *WSDM*, pp. 425–434 (2017)