PARL: Let Strangers Speak Out What You Like

Libing Wu[1], Cong Quan[1], Chenliang Li[2*], Donghong Ji[2]
1. School of Computer Science, Wuhan University, Wuhan, 430072, China
{wu, quancong}@whu.edu.cn
2. School of Cyber Science and Engineering, Wuhan University, Wuhan, 430072, China
{cllee, dhji}@whu.edu.cn

ABSTRACT

Review-based methods are one of the dominant methods to address the data sparsity problem of recommender system. However, the performance of most existing review-based methods will degrade when the review is also sparse. To this end, we propose a method to exploit user-item **p**air-dependent features from **a**uxiliary **r**eviews written by **l**ike-minded users (PARL) to address such problem. That is, both the reviews written by the user and the reviews written for the item are incorporated to highlight the useful features covered by the auxiliary reviews. PARL not only alleviates the sparsity problem of reviews but also produce extra informative features to further improve the accuracy of rating prediction. More importantly, it is designed as a plug-and-play model which can be plugged into various deep recommender systems to improve recommendations provided by them. Extensive experiments on five real-world datasets show that PARL achieves better prediction accuracy than other state-of-the-art alternatives. Also, with the exploitation of auxiliary reviews, the performance of PARL is robust on datasets with different characteristics.

CCS CONCEPTS

• **Information systems → Recommender systems**;

KEYWORDS

Recommender System, User Reviews, Deep Learning, Rating Prediction

ACM Reference Format:
Libing Wu, Cong Quan, Chenliang Li, Donghong Ji. 2018. PARL: Let Strangers Speak Out What You Like. In 2018 ACM Conference on Information and Knowledge Management (CIKM'18), October 22–26, 2018, Torino, Italy. ACM, NY, NY, USA, 10 pages. https://doi.org/10.1145/3269206.3271695

1 INTRODUCTION

In the past decade, Recommender Systems have been playing an increasingly important role in many online platforms, including E-commerce such as Amazon, video-streaming providers such as Youtube and social media such as Twitter. The key to a recommender system is to dig out user's interest and thus personalized recommendation can be provided.

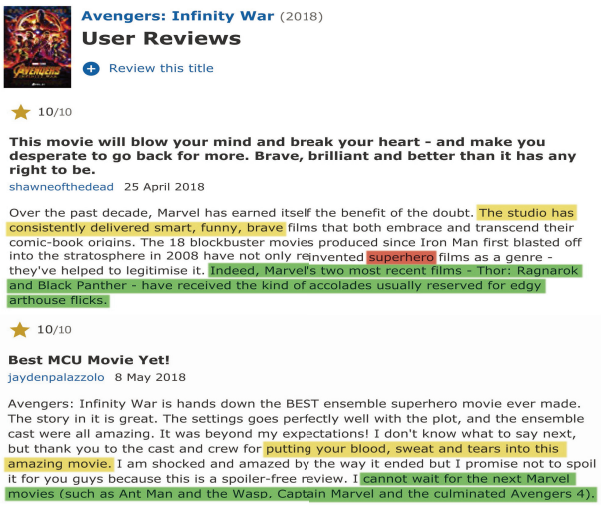


Figure 1: Examples of reviews written by two audiences for the film "Infinite War" on IMDB. Both users give the highest rate to Infinite War, and similar interests are presented by their reviews.

Many recommender systems are based on Collaborative Filtering (CF) [29], which mainly rely on the past interaction records between users and items. Although CF techniques enjoy a surge of attention and provide good performance, the sparsity problem hinders the CF-based recommender systems from providing high-quality recommendations to the users with few records. In this case, neighborhood methods [11] are employed to alleviate the data sparsity problem. Concretely, neighborhood methods utilize the relations between items or between users to improve the recommendation. For example, a neighborhood-based method [2] estimates an unkown rating on an item made by a user based on the known ratings made by the same user on other items.

Recently, some approaches also resort to reviews written by users to address the lack of data. The motivation is that ratings and review are two facets of users to depict their experience on items. Therefore, the reviews can be used to well alleviate data sparsity. Some studies [20, 37] have shown that methods considering reviews generally perform better than collaborative filtering methods which only take the interaction records into consideration. In particular, the performance of review-based methods is robust when the users

*Chenliang Li is the corresponding author.

CIKM '18, October 22–26, 2018, Torino, Italy

ACM ISBN 978-1-4503-6014-2/18/10…$15.00
https://doi.org/10.1145/3269206.3271695

only have few rating records. Although review-based methods have proven to successfully alleviate the data sparsity problem, they still have some inherent limitations: 1) few works tackle the data sparsity problem of review. There is a tendency that most users leave no comments for products purchased, and thus inevitably leads to degrade performance of review-based methods. 2) most reviews written by users are short and thus cannot fully reflect the users' interests.

To overcome above limitations, we propose to exploit the reviews written by like-minded users (denoted as auxiliary reviews) to tackle the data sparsity problem of review. Here, like-minded users refer to the users who give similar ratings toward the same item. For example, in Figure 1, the upper and lower rows show the comments given by two audiences "shawn" and "jayden" respectively. As we can see, Shawn and Jayden, who may be strangers to each other in real life, have presented similar comments and ratings on the same film. We observe that complementary and fresh informative features can be uncovered in the reviews written by like-minded users. For example, we can infer that Jayden is a superhero fan from the review written by Shawn, and Shawn may give high rate to "Ant Man" according to jayden's review. We believe that these auxiliary reviews from like-minded users could be useful, especially when review texts are scarce.

In this paper, we propose a deep learning based model to extract useful user-item **p**air-dependent features from **a**uxiliary **r**eviews written by **l**ike-minded users (PARL) for rating prediction. Specifically, PARL uses a neural network architecture to extract informative features from users' auxiliary review document which is formed by the auxiliary reviews provided by the like-minded users. The informative features are then further extracted with the help of the user review document and item review document. In this sense, the features extracted by PARL are on the basis of user-item pairs. Finally, the learnt features of user auxiliary review documents can be fed into different neural models to further improve the performance of rating prediction. Our experiments demonstrate the effectiveness and strength of PARL in rating prediction on various datasets. The main contributions of our work are summarized as follows:

- We propose a deep learning based method named PARL to extract informative features from users' auxiliary reviews to significantly improve the rating prediction upon different user-item pairs. Besides, PARL can be regarded as a plug-in for different models.
- To the best of our knowledge, we are the first to focus on the data sparsity problem of reviews and firstly propose the review-based neighborhood model to tackle this problem.
- Experimental results on five real-world datasets demonstrate that our proposed PARL consistently outperforms existing state-of-the-art methods by plugging in on top of one of the review-based method called DeepCoNN.

2 RELATED WORK

Existing approaches addressing the data sparsity problem can be roughly divided into three categories: neighborhood based methods, social information based methods, and textual information based methods. In this section, we briefly review the relevant literatures.

2.1 Neighborhood based Methods

There has been a great amount of works focusing on leveraging interaction information to tackle data sparsity problem. The most common model is called neighborhood model [11, 18] which assumes that the interaction bewteen a user and an item is affected by their interaction history. SVD++ [14] is the most famous neighborhood-based model which smoothly merges the latent factor model and the neighborhood model. It characterizes a user's rating on an item by considering not only the interaction of factors of the user-item pair, but also the user's rating behaviors on other items. It outperforms the vanilla latent factor models [15, 24] because of the incorporation of the neighborhood information. Likewise, the neighborhood information of both users and items is encoded by [7] to extend the restricted Boltzmann machines (RBM) [25] model for better rating prediction. In contrast to directly incorporate neighborhood information, a two-stage model, called DCT [1], firstly completes a rating sub-matrix excluded cold-start users/items and then transducts the knowledge from the completed sub-matrix to the cold-start users/items. DCT can prevent the errors of the completion and the transduction from propogating repetitively in an uncontrolled way. Although these models can benefit from modeling the neighborhood information of users/items, the neighboor information purely extracted from sparse user-item rating records is still limited. New source of ammenable information should be included for higher recommendation quality.

2.2 Social Information based Methods

In recent years, there has been a considerable interests in exploiting social interaction information to compensate for the lack of rating interactions. These methods can be classified as trust-based recommendation methods that use additional interactions of a trust network formed by users to deal with the cold start problem. As one of the early works, [12] uses a random walk model, named TrustWalker, to combine trust-based and latent factor model for recommendation. Apart from modeling direct relations between users, TrustSVD [8] integrates both the explicit and implicit influence of trusted users into SVD++ and achieves better accuracy. However, treating different relations homogeneously is ineffective since they reflect different types of influences from other users. Based on such assumption, [16] and [31] regard social connections as heterogeneous relations and introduce social groups as regularization into the matrix factorization model.

The above methods heavily rely on the assumption that the number of overlapped user sets beween social platform and recommendation platform is large, which may not hold in practice. Besides, most existing approaches neglect the mutual benefits between social relations and rating preferences. Towards this end, a recent effort, called CrossFire [27], is able to transfer knowledge across different platforms and simultaneously conducts cross media friend and item recommendation. Yet the effectiveness of these approaches are jeopardized by the fact that social relations are not available for most recommender systems. In this case, most social-based recommendation models become inapplicable in reality. In contrast, PARL gets rid of this dilemma by leveraging both textual information and neighborhood information, each of which is included in the recommender system.

2.3 Textual Information based Methods

Different with social information which is inavailable for most recommender system, textual information *e.g.,* users' reviews, item description and labels, is a common property in the recommender system other than the numeric ratings. Hence, exploiting textual information to address the inherent data sparsity problem of recommender system has become a hotspot in recent years.

Some researches [19, 20, 33] propose to employ topic modeling techniques to extract meaningful topics from text and then incorporate them into latent factor models. RBLT [30] uses LDA [3] to extract topic features from rating-boost review document as latent factors. The rating-boost review document is formed by simply repeating a review r times in the document if its rating score is r and so that LDA can easily extract the topic features expressed in highly rated reviews. These extracted topic features are later integrated into a MF framework to derive item's latent factors. CDL [34] generlizes the review-based method from topic model to deep learning, which tightly couples SADE over the text information and PMF [24] for the implicit rating matrix. These methods outperform the models which solely rely on sparse user-item interaction data. However, these bag-of-words (BOW) based models ignore the word order and therefore lose much semantic context information.

To overcome the shortcomings of the BOW representation, most of the recent efforts is focusing on utilizing convoluntional neural network (CNN) to obtain better latent semantic representations from textual information by considering the word order and the local context. ConvMF [13] shares the same structure with CDL, yet it uses CNN to extract item's characteristics over item description. To better model both users and items, DeepCoNN [37] uses a parallel CNN architecture to separately derive the semantic features of users and items based on their reviews. However, DeepCoNN can achieve its best performance only if the target pair-wise review is available, which is unrealistic. TransNet [4] extends DeepCoNN by adding a transform layer to mimic the latent representation of the target reviews, which is not available at test time and gains improvement in rating prediction against DeepCoNN. D-attn [26] leverages global and local attention mechanism to enable CNN network to mainly focus on selective features of review and hence improve the prediction accuracy over DeepCoNN. [5] argues that less-useful reviews substantially degrade model performance. It proposes a model which utilizes the attention mechanism to select more-useful reviews for rating prediction.

In addition to the CNN-based method, [36] proposes a framework that fuses word embedding model with standard MF model for rating prediction. ACF [6] tries to clarify user's preferences in multimedia recommendation via using two attention modules to capture informative items from user's purchased records and informative components of item. Despite these recent methods are proven to be effective on recommendation, they suffer from two problems: 1) they cannot handle the situation when the text data is also scarce. 2) review document cannot completely reflect users' preferences. Unfortunately, these issues are inevitable in practice. To alleviate such limitations, PARL finds an extra pathway to uncover user's interests from the reviews written by their like-minded users, and accordingly, recommender system gains extra information to profile users, especially when the user leaves little or incomplete information on the platform.

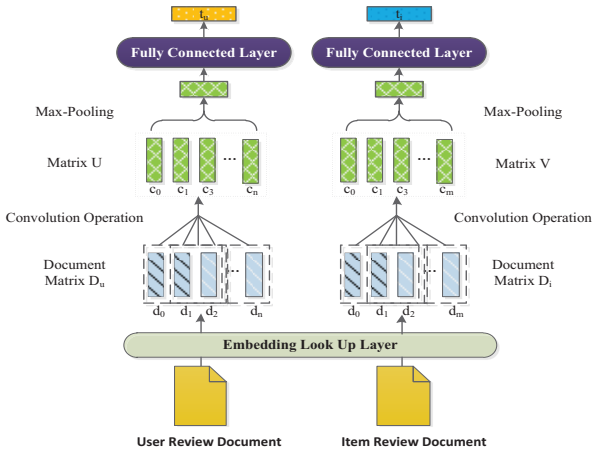


Figure 2: The network architecture of DeepCoNN.

3 THE PROPOSED MODEL

We design PARL as a plug-and-play architecture which can be plugged into different models and improve the performance of the latter ones. Here we choose *DeepCoNN*, one of the state-of-the-art review-based methods, as the basic model to test whether PARL can improve the prediction accuracy of DeepCoNN.

3.1 The DeepCoNN model

As shown in Figure 2, DeepCoNN uses two parallel CNN networks to model user perferences and item characteristics from user review documents and item review documents, respectively. A user review document is formed by concatenating all the reviews written by the same user. Similarly, an item review document is formed with all the reviews made by the users for the same item. Because parallel architectures are applied to users and items, here we only present the detail of the user network in the following.

Convolution Layer. Firstly a user review document $D_u = (d_1, d_2, ..., d_n)$ is fed into an embedding look-up layer which projects each word to its corresponding embedding $\mathbf{d} \in \mathbb{R}^{y \times 1}$. Then a word embedding matrix of document D_u, denoted as $\mathbf{D}_u \in \mathbb{R}^{n \times y}$, is constructed as follows:

$$\mathbf{D}_u = \left[\cdots \mathbf{d}_{j-1}, \mathbf{d}_j, \mathbf{d}_{j+1} \cdots\right]^{\mathrm{T}}$$

where $\mathbf{d}_j$ is the word embedding of the j-th word in document D_u. Following the look-up layer is the convolution operation. Specifically, g convolution filters with the same sliding window of size s are applied over matrix $\mathbf{D}_u$ to extract contextual features. A convolution operation regarding to each filter f_j is performed as:

$$c_h^j = \sigma(\mathbf{W}_c^j * \mathbf{D}_{h:h+s-1} + b_j) \quad (1)$$

where σ is a nonlinear activation function, $*$ is convolution operator, $\mathbf{W}_c^j \in \mathbb{R}^{s \times y}$ is the convolution weight matrix for filter j, $\mathbf{D}_{h:h+s-1}$ is the slice of matrix $\mathbf{D}$ within the sliding window starting at h-th position, b_j is the bias, c_h^j is the local contextual feature extracted by filter f_j over the sliding window starting at position h. Without

explicit specification, we opt for Rectified Linear Unit (ReLU) as the activation function, *i.e.*, $ReLU(x) = \max(x, 0)$.

Max-pooling Layer. After getting the features c^j produced by filter f_j, a max-pooling operation is applied to reserve the most valuable contextual feature of each filter:

$$o_j = max\{c_1^j, c_2^j, \cdots, c_{n-s+1}^j\} \tag{2}$$

After the max-pooling operation, all reserving features are concatenated as the output of the max-pooling layer:

$$\mathbf{o}_u = [o_1, o_2, \cdots, o_g] \tag{3}$$

Fully Connected Layer. Finally the output of the max-pooling layer is passed to a fully connected layer to get the latent representation of user u as:

$$\mathbf{t}_u = \sigma(\mathbf{W}_u \mathbf{o}_u + \mathbf{b}_u) \tag{4}$$

3.2 Limitations of DeepCoNN

Although DeepCoNN achieves satisfactory performance on rating prediction by extracting high and nonlinear latent features of user review documents and item review documents, there are two problems for it to be tackled: 1) It only achieves its best performance when the target user-item review is available [4], which is unrealistic. 2) Few useful features can be extracted by CNN when the text data is scarce. Namely, DeepCoNN will suffer from performance loss when the review document is short and imcomplete.

The first problem is handled with TransNet by forcing $\mathbf{t}_u$ and $\mathbf{t}_i$ to approximately mimic the latent representation of target user-item pair reivew. However, the second issue is still unsolved and becomes the bottleneck for many review-based methods such as DeepCoNN and TransNet.

3.3 PARL

As we have mentioned before, PARL aims to extract useful user-item pair-dependent features from user auxiliary review documents to alleviate the sparsity problem of review. For each user u, the construction procedure of the auxiliary review document is detailed in Algorithm 1.

Algorithm 1: Construction of User Auxiliary Review Document

```
1:  Input: user u
2:  Output: auxiliary review document of u
3:  record = get_record(u) # obtain u's purchased record
4:  document = None
5:  for item i in record do
6:      rate = get_rate(u, i) # get u's rating on i
7:      review = Pick_Auxiliary_Review(i, rate)
8:      if review = None then
9:          review = Pick_Auxiliary_Review(i, rate + 1)
10:     end if
11:     if review = None then
12:         review = Pick_Auxiliary_Review(i, rate − 1)
13:     end if
14:     document + = review
15: end for
16: return document
```

Here, an auxiliary review is a review written by other user with the same rating score as given by the target user. Note that, for each product commented by user u, we only randomly pick one auxiliary review and add it into the user auxiliary review document

Algorithm 2: Pick_Auxiliary_Review

```
1: Input: item i, rating r
2: Output: picked review
3: set = get_users(i, r) # get users who give rate r to i
4: u = random(set) # randomly select one like-minded user
5: return review_{u,i}
```

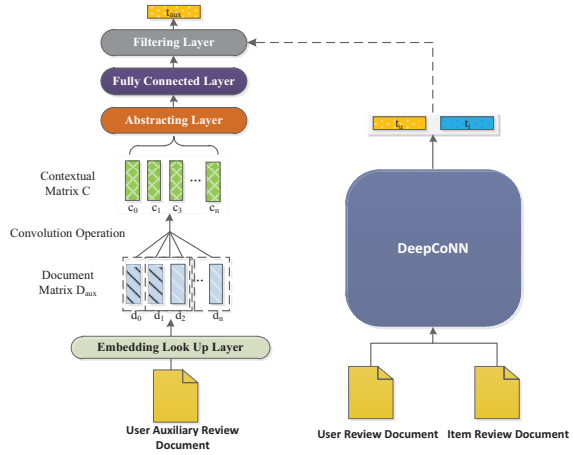


Figure 3: The network architecture of PARL.

so that the auxiliary review document will not be dominated by the reviews written for popular items. Besides, different users may have different rating behaviors and most users tend to give high rating. Hence we also regard the users who give similar ratings as like-minded users and prefer the higher rating than the lower one. Namely, if no other users give the same rating, we will consider the users who give similar rating, *e.g.*, rating +1 and rating −1, as like-minded users and prefer the user who gives higher rating (line 8 − 13 in Algorithm 1).

Since the user auxiliary review document is formed by the reviews written by different like-minded users while the user review document consists of the reviews from the same user, it is reasonable to expect that features of these two documents are heterogeneous. The primary objective of PARL is to make use of these incompatible features to infer ratings upon user-item pair. Most recent researches [17, 22] have demonstrated that utilizing user review documents to generate target user-item review can improve the quality of recommendation and make the recommendation more explainable. Here, we speculate that the features extracted from auxiliary reviews can be translated into informative features that may not be fully expressed in the user review document.

Figure 3 shows the network architecture of PARL which consists of four layers to learn the latent representation of a user auxiliary review document which will be leveraged as a new source of information to improve the performance of rating prediction.

Convolution Layer. Given a user auxiliary review document $D_{aux} = (d_1, d_2, ..., d_n)$, similar to DeepCoNN, PARL also employs a convolution layer to process it. As a result, the output of sliding window centered at poistion h is a concatenation of the contextual features c_h^f extracted by different filters, denoted as:

$$\mathbf{c}_h = [c_h^1, \cdots, c_h^g] \tag{5}$$

Since $\mathbf{c}_h$ consists of different contextual features at position h, it can be regarded as a contextual feature vector for the h-th word in D_{aux}. In this way, we can form the contextual matrix of user auxiliary review document by concatenating these contextual vectors as follows:

$$\mathbf{C} = [\mathbf{c}_1; \cdots ; \mathbf{c}_n]^{\mathrm{T}} \tag{6}$$

where $\mathbf{C} \in \mathbb{R}^{n \times g}$, $\mathbf{c}_1$ denotes the contextual vector of word d_1 in D_{aux}, and n is the length of D_{aux}, g is the number of filters.

Abstracting Layer. Recall that user auxiliary review document is formed by the concatenation of reviews written by different users who do not have explicit relation with target user. In other words, not all information in auxiliary review document is useful for the rating prediction of target user. For example, the opinions presented by an appearance-valuing user may be futile for a pragmatist though they both give a high rating to a product with both practicability and aesthetics. The goal of abstracting layer is to effectively extract important features in auxiliary reviews while avoid degrading the accuracy of rating prediction. Given the contextual matrix $\mathbf{C}$, abstracting layer is proposed to further extract more informative and higher-level semantic features from the contextual matrix $\mathbf{C}$.

According to [35], stacking a CNN network on top of the contextual matrix is effective on both rating prediction and the consistency of performance, especially when the semantics in the document are incoherent. Consequently, we apply a max-pooling based CNN network to extract higher level semantic features from $\mathbf{C}$ as follows:

$$q_h^j = \sigma(\mathbf{W}_{abs}^j * \mathbf{C}_{h:h+s-1} + b_{abs}) \tag{7}$$

$$q_j = \mathbf{max}\{q_1^j, \cdots, q_{n-s+1}^j\} \tag{8}$$

$$\mathbf{q}_{abs} = [q_1, \cdots, q_g] \tag{9}$$

where $\mathbf{W}_{abs}^j \in \mathbb{R}^{s \times g}$ is the convolution weight matrix for filter j of the abstracting layer, b_{abs} is the convolution bias, and q_j is the max-pooling ouput of filter f_j. Note that local context information has been encoded in contextual feature vector $\mathbf{c}$, a further CNN network *i.e.,* the abstracting layer, can broaden the context for latent feature extraction. In detail, if the window size of both convolution layer and abstracting layer is s, the information of $2s-1$ words will be encoded in each feature q through Equation 7.

Fully Connected Layer. Similar to DeepCoNN, the output of the abstracting layer is also passed to a fully connected layer:

$$\mathbf{q}_{aux} = \sigma(\mathbf{W}_{fc}\mathbf{q}_{abs} + \mathbf{b}_{fc}) \tag{10}$$

In general, $\mathbf{q}_{aux} \in \mathbb{R}^{l \times 1}$ can be regarded as the learnt latent vector of user auxiliary review documents and be used for other models as auxiliary information to improve understanding of the user. However, the rating prediction task is based on the user-item pair. Namely, the pair-dependent features, which are strongly related to the target user-item pair, should be carried by the learnt latent vector of user auxiliary review document for better prediction accuracy.

Filtering Layer. Because PARL serves as an extra information provider for rating prediction on different user-item pairs, it is reasonable to assume that different user-item pair is associated with user's different interests. To preserve the informative features while filtering out the other irrelevant features, filtering layer is added to adaptively extract the features carried by $\mathbf{q}_{aux}$ with respect to the target user-item pair.

In this layer, PARL applies highway network and gated-mechanism to model the dimension-level features of $\mathbf{q}_{aux}$. Firstly, a one-layer highway network is served as "transform gate" to control the information carried by $\mathbf{q}_{aux}$ as defined:

$$\mathbf{q} = \sigma(\mathbf{W}_1^H \mathbf{q}_{aux} + \mathbf{b}_1^H) \tag{11}$$

$$\eta = sigmoid(\mathbf{W}_2^H \mathbf{q} + \mathbf{b}_2^H) \tag{12}$$

$$\mathbf{q}_{high} = \eta \odot \mathbf{q} + (1 - \eta) \odot \mathbf{q}_{aux} \tag{13}$$

where $\mathbf{q}$ is a higher-level representation of $\mathbf{q}_{aux}$, η is the transform gate which controls the information flowing, $\odot$ denotes the element-wise product operation.

Given the latent vector $\mathbf{t}_u$ of user u, the latent vector $\mathbf{t}_i$ of item i and the abstracted high-level vector $\mathbf{q}_{high}$, a user-item pair-based gated mechanism is employed to highlight the relevant features carried by $\mathbf{q}_{high}$ as follows:

$$\mathbf{gt} = tanh(\mathbf{W}_u^{gate}\mathbf{t}_u + \mathbf{W}_i^{gate}\mathbf{t}_i + \mathbf{W}_{aux}^{gate}\mathbf{q}_{high} + \mathbf{b}^{gate}) \tag{14}$$

where $\mathbf{W}_u^{gate}$, $\mathbf{W}_i^{gate}$ and $\mathbf{W}_{aux}^{gate}$ are the weight matrices for user vector, item vector and user auxiliary vector, resprectively. $\mathbf{b}^{gate} \in \mathbb{R}^l$ is the bias.

Finally, with the cooperation between the highway network and the gated mechanism, the pair-dependent latent vector $\mathbf{t}_{aux}$ of the user auxiliary review document is computed as:

$$\mathbf{t}_{aux} = \mathbf{gt} \odot \mathbf{q}_{high} \tag{15}$$

3.4 Fusion of DeepCoNN and PARL

It is clear that, with the help of PARL, user u is associated with two vectors. One is $\mathbf{t}_u$ which is directly learned from her review document, the other is $\mathbf{t}_{aux}$ learned from the reviews written by her like-minded users. In order to incorporate $\mathbf{t}_{aux}$ into DeepCoNN as a new source of information, here , we combine these two vectors as a whole to better profile a user via:

$$\mathbf{t}_u^{comb} = \sigma(\mathbf{W}_{u1}[\mathbf{t}_u, \mathbf{t}_{aux}] + \mathbf{b}_{u1}) \tag{16}$$

Now, we have two vectors. One is the item latent vector $\mathbf{t}_i$, and the other one is the combined user latent vector $\mathbf{t}_u^{comb}$. Both vectors are passed through a dropout layer [28] to prevent the model from overfitting. Moreover, in order to model high-order features interaction between user vector and item vector while getting rid of the independent dimension constraint of Dot Product (DP) operation, we follow DeepCoNN by concatenating the two vectors as $\mathbf{z}_{u,i} = [\mathbf{t}_u^{comb}, \mathbf{t}_i]$ and then feed it into Factorization Machine (FM) [23] to predict the rating upon user u and item i as follows:

$$\hat{r}_{u,i} = m_0 + \mathbf{m}^{\mathrm{T}}\mathbf{z}_{u,i} + \frac{1}{2}\mathbf{z}_{u,i}^{\mathrm{T}}\mathbf{M}\mathbf{z}_{u,i} \tag{17}$$

$$\mathbf{M}_{j,k} = \mathbf{v}_j^{\mathrm{T}}\mathbf{v}_k, j \neq k \tag{18}$$

where m_0 is the global bias, $\mathbf{m}$ is the coefficent vector for latent feature vector $\mathbf{z}_{u,i}$, $\mathbf{M}$ is the weight matrix for second-order interactions and its diagonal elements are 0 (*i.e.,* $\mathbf{M}_{j,j} = 0$), $\mathbf{v}_j, \mathbf{v}_k \in \mathbb{R}^v$ are the v-dimensional latent vectors associated with dimension j and k of $\mathbf{z}_{u,i}$. Here, we restrict the dimension size of $\mathbf{t}_u$, $\mathbf{t}_i$, $\mathbf{t}_{aux}$ and $\mathbf{t}_u^{comb}$ to be the same to ease the model complexity.

Besides, in reality, there are multiple inherent tendencies behind rating behaviors. This results in various rating variation associated to different user-item pairs. Introducing user bias and item bias has been proven to be effective to deal with rating variation [15]. In this sense, we extend Equation 17 as follows:

$$\hat{r}_{u,i} = m_0 + \mathbf{m}^{\mathrm{T}}\mathbf{z}_{u,i} + \frac{1}{2}\mathbf{z}_{u,i}^{\mathrm{T}}\mathbf{M}\mathbf{z}_{u,i} + b_u + b_i \tag{19}$$

where b_u and b_i are the corresponding bias for user u and item i, respectively. In our experiment, we choose square loss function as the objective function to train parameters:

$$loss = \sum_{(u,i)\in O} (r_{u,i} - \hat{r}_{u,i})^2 + \lambda \|\mathbf{t}_u - \mathbf{t}_{aux}\|^2 \tag{20}$$

where O denotes the set of observed user-item rating pairs. The second part in Equation 20 is the constraint made between the latent vector $\mathbf{t}_u$ and the latent vector $\mathbf{t}_{aux}$. Note that, although $\mathbf{t}_{aux}$ is user-item pair dependent, the features carried by it might be still incompatible to the features carried by $\mathbf{t}_u$ because of the heterogeneity of the user auxiliary review document and the user review document. An example is illustrated in Figure 1, audience "shawn" uses smart, funny and brave to describe Infinity War while "jayden" uses blood, sweat and tears instead. Thus, we need to bridge the gap between the representation of the user auxiliary review document and the representation of the user review document to make their features compatible. Inspired by the success of TransNet, we argue that the connection between the two types of review documents can be established by forcing $\mathbf{t}_{aux}$ to moderately be similar to the latent representation of user review document. We use λ to control the degree of this constraint and the selection strategy of λ will be reported in the experimental section. The parameters of PARL and DeepCoNN are jointly learnt. For parameter update, we utilize RMSprop [32] over mini-batches.

4 EXPERIMENTS

In this section, we conduct extensive experiments on five real-world datasets with two different sources for performance evaluation. We also analyze the contribution of different components of PARL and different parameter settings for PARL. Finally, a case study is presented to demonstrate the strength of PARL[1].

4.1 Experimental Settings

Datasets. In our experiment, four Amazon-5cores datasets[2] [9] from different domains are used: *Office Products*, *Digital Music*, *Video Games*, and *Tools Improvement*. Another dataset is collected from the RateBeer website [21], called *Beer*. It consists of the ratings and comments for different beers from different people around the world. Note that, we also extract the 5 core over Beer, such that each user and item has at least 5 reviews.

Similar preprocessing steps used in [13] are performed by us to preprocess the review documents for all datasets as follows: 1) remove stop words and words that have the document frequency higher than 0.5; 2) calculate tf-idf score for each word and select the top 20,000 distinct words as vocabulary; 3) remove all words out of the vocabulary from raw documents; 4) amputate (pad) the long (short) review documents to the same length of 300 words. We further filter out the rating records which contain empty review after document preprocessing. The empty reviews are filtered out after the data preprocessing.

Table 1 summarizes the detailed statistics of the five datasets after the preprocessing steps. We can see that the five datasets hold quite different characteristics in terms of both user-item interaction sparsity and review data scarcity. Besides, it can be observed that there is a positive correlation between the length of user review document and the length of user auxiliary review document. For evaluation, we randomly select 80% of each dataset as the training set and the remaining 20% as the testing set. Moreover, we split 10% of the training set as the validation set for hyper-parameter validation. The training sets are selected such that at least one interaction for each user/item should be included. Finally, the reviews in the validation set and testing set are excluded since they are unavailable during rating prediction in practice.

Baselines. We compare the proposed PARL against the following state-of-the-art rating prediction methods:

- **PMF**: Probabilistic matrix factorization is a standard matrix factorization model that uses only rating scores [24]. We use the Alternating Least Squares (ALS) techniques for model optimization.
- **SVD++**: It is an extentsion of Singular Value Decompose which merges both the latent factor model and the neighborhood model [14].
- **CDL**: Collaborative Deep Learning [34] is the first heirarchical Bayesian model to build the connection between deep learning technique (SDAE) and matrix factorization model. Following the adaption used in [13], we set the confidence parameter to 1 if the rating is observed and 0 otherwise.
- **RBLT**: Rating-Boosted Latent Topics model integrates both matrix factorization model and topic model [30]. It proposes a rating-boosted approach which utilizes the rating-boosted reviews and rating scores together for rating prediction.
- **CMLE**: Collaborative Multi-Level Embedding Learning integrates word embedding model with matrix factorization to learn user and item embeddings [36]. Given a new user-item pair, the rating can be predicted by the dot-product of its user and item embeddings.
- **ConvMF**: Convolutional Matrix Factorization integrates CNN into PMF for rating prediction [13]. The item latent features are extracted by using CNN over the item review documents.
- **DeepCoNN**: Deep Cooperative Neural Networks uses two parallel CNN networks to extract latent feature vectors from both the user review documents and item review documents [37]. FM is then used for rating prediction.
- **DeepCoNN-Aux**: DeepCoNN-Aux is a variation of DeepCoNN. It uses the same network architecture of DeepCoNN while both the user review document and the user auxiliary review document are concatenated as the input for user latent vector extraction.
- **D-attn**: Dual local and global attention model leverages global and local attentions to enable an interpretable embedding of users and items [26]. Finally, the rating is estimated by dot-product of the user and item embeddings.

[1] Our implementation is available at https://github.com/WHUIR/PARL
[2] http://jmcauley.ucsd.edu/data/amazon/

Table 1: Statistics of the five datasets

Datasets	# users	# items	# ratings	# words per review	# words per user	# words per item	# words per auxiliary	density
Beer	7,725	21,976	66,625	17.31	34.06	103.20	71.61	0.039%
Office Products	4,905	2,420	53,228	48.15	197.93	229.52	233.95	0.448%
Digital Music	5,540	3,568	64,666	69.57	216.21	266.51	260.89	0.327%
Video Games	24,303	10,672	231,577	72.13	188.79	260.60	241.35	0.089%
Tools Improvement	16,638	10,217	134,345	38.75	162.53	212.48	185.21	0.079%

- **TransNets**: TransNets extends the DeepCoNN model by adding an additional layer to represent the target user-item review, which is unavailable at test time [4]. Then TransNets can mimic the target user-item review representation at test time and thus improve the performance of rating prediction.

Among these methods, PMF is the conventional latent model that utilizes only the user-item rating scores. SVD++ uses the neighborhood information to tackle sparsity problem. The rest methods all utilize the review documents for rating prediction.

Hyper-parameter Settings. Grid search are performed to tune the hyper-parameters for all the methods based on the setting strategies reported by their papers. We report their performances over 5 runs on the testing set. The latent dimension size is optimized from $\{25, 50, 100, 150, 200, 300\}$. The word embeddings are randomly initialized and fine-tuning strategy is employed. The dimension size of word embedding is set to 300 (*i.e.,* $y = 300$). The batch size for Beer, Office Products and Digital Music is set to 100. For the other two big datasets, the batch size is set to 200. The number of convolution filters is set to 50 (*i.e.,* $g = 50$). The statistical significance test is conducted by applying the student *t-test*.

For PARL, the dimension size for user and item latent feature vectors is $l = 50$, window size is $s = 3$, and v is set to 100 for FM to conduct rating prediction. The keep probability of dropout strategy is set to 0.8. λ is set to 0.01. And the learning rate is set to 0.002.

Evaluation Metric. The well-known Mean Square Error (MSE) is adopted to evaluate the performance of all methods:

$$MSE = \frac{1}{|O_t|} \sum_{(u,i) \in O_t} (r_{u,i} - \hat{r}_{u,i})^2 \tag{21}$$

where O_t is the set of the user-item pairs in the testing set.

4.2 Performance Evaluation

This section presents the performance of different methods over the five benchmark datasets. A summary of results is reported in Table 2. Several observations can be made:

First, for the interaction-based method: PMF performs the worst in all five datasets, especially when the dataset is very sparse, *e.g.,* Beer, Video Games and Tools Improvement. SVD++ outperforms PMF across the five datasets and gains larger improvement on the sparse datasets. It is reasonable, since the neighborhood information are incorporated into SVD++ to address data sparsity. Moreover, it is not surprising that most of the review-based methods surpass SVD++ on all datasets, especially when the dataset is much sparser. This confirms that leveraging review information as complementary information source can provide more useful semantic information for rating prediction.

Second, among review-based methods, RBLT performs relatively good on five datasets. It achieves the second best performance on Beer, Digital Music and Tools Improvement. We believe the reasons are as follows. First, its rating-boost strategy ensures that the recommendable features dicussed in higher-rating reviews can be extracted successfully. Second, leveraging user and item biases will make the model more robust on various datasets. This is also a reason why SVD++ obtains significantly improvement over PMF. On the other hand, the methods that utilize deep learning technology obtain better performance on the datasets with rich review information. For instance, D-attn and CDL are the strongest baselines on Video Games and Office Products while they lag behind RBLT on both Digital Music and Tools Improvement. Moreover, the performance of D-attn is on par with DeepCoNN on Beer while it obtains significant improvements over DeepCoNN on the other datasets. With the regularization on the mimic of the latent representation of target user-item review, TransNet outperforms DeepCoNN across all the five datasets. However, the margin is significantly shrinking on Beer which has the sparsest review data. These observations indicate that although the existing neural network based solutions are performing good on many domains, the prediction performance of them would be relatively poor when the textual information is scarce and incomplete.

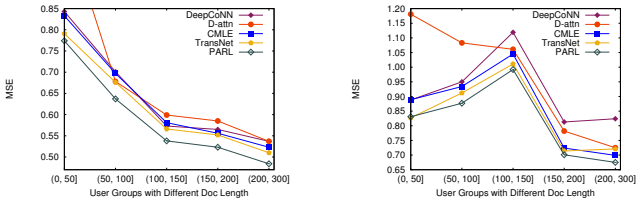
Third, as shown in Table 2, PARL consistently achieves the best performance across the five datasets. Even for the sparsest dataset—Beer, PARL still gains 2.6% improvement over the best baseline. This substantial improvement should be attributed to the exploitation of user auxiliary reviews, which is the major extension to other review-based methods. Note that, DeepCoNN-Aux also utilizes user auxiliary reviews as input of vanilla DeepCoNN. However, few improvement is gained compared to DeepCoNN which demonstrates that the user network of DeepCoNN is incapable of extracting the useful features in user auxiliary review document. This validates the effectiveness of PARL on extracting the useful features from the user auxiliary review documents for each user-item pair.

4.3 Robustness Analysis

This section is to verify the robustness of PARL over various datasets, especially for the case when the review data is scarce. We partition the users of each dataset into five groups in terms of the length of the users' review documents and conduct comparisons beween PARL and other four methods which utilize user review documents

Table 2: Overall performance comparison on five datasets. The best and second best results are highlighted in boldface and underlined respectively. † indicates that the difference to the best result is statistically significant at 0.01 level.

Method	Beer	Office Products	Digital Music	Video Games	Tools Improvement
PMF	1.636†	1.091†	1.211†	1.669†	1.564†
SVD++	0.726†	0.771†	0.950†	1.183†	1.066†
CDL	0.678†	0.754†	0.882†	1.179†	1.033†
RBLT	0.576†	0.757†	0.872†	1.147†	0.983†
CMLE	0.607†	0.761†	0.883†	1.254†	1.023†
ConvMF	0.853†	0.960†	1.084†	1.449†	1.240†
DeepCoNN	0.617†	0.860†	1.060†	1.238†	1.063†
DeepCoNN-Aux	0.615†	0.860†	1.059†	1.236†	1.058†
TransNets	0.586†	0.760†	0.910†	1.196†	1.008†
D-attn	0.616†	0.824†	0.914†	1.142†	1.046†
PARL	**0.561**	**0.731**	**0.849**	**1.117**	**0.955**



(a) Beer (b) Office Products

Figure 4: Performance of Methods over Different Groups on Beer (a) and Office Products (b).

for rating prediction. The comparisons are shown in Figure 4. To be specific, (0, 50] denotes group 1 which consists of the users whose review document is shorter than 50 words. (50, 100] denotes group 2, and so forth. We only list the comparisons of Beer and Office Products due to the space limitation. The results of the other datasets are similar to the one showed in Figure 4(b).

Figure 4(a) plots the comparison on the five user groups of Beer. The *MSE* of all methods is decreasing with respect to the increasing length of user review document. PARL and TransNet are the best and the second best models across the five groups. Note that, with the increasing of the length of user review document, the margin between PARL and TransNet is broaden. This is reasonable, since long user review document is usually accompanied with long user auxiliary review document, and thus more informative features would be extracted to improve rating prediction. This observation illustrates that user auxiliary review documents not only alleviate the sparsity problem of review data, but also produce extra features to improve the performance of rating prediction.

The observations on Office Products are a little different. The perdiction accuracy of most methods on group 1 is better than the prediction accuracy on the next two groups whose users' review document is longer. Amazon users in group 1 tend to give higher rates to products loosely without writing a detailed review, while users in group 2 and 3, who like to share their experience, are strict with their ratings on products. This is reasonable, and consequently the rating behaviors behind group 1 are easier to be modeled than the rating behaviors behind group 2 and 3. Similar observations are made from the other three Amazon datasets.

Finally, we can observe that the performance of TransNet is merely on par with other baselines when user review documents are long. We speculate that the transform layer in TransNet is ineffective when the review document is dense. Instead, information may be lost when user review documents and item review documents are forced to mimic the representation of target user-item reviews. By contrast, PARL maintains strong performance across different groups which demonstrate the robustness of the method.

4.4 Parameter Sensitivity Analysis

Here, we study the impact of different parameter settings for PARL.

Number of Dimensions. Figure 5 plots the performance of PARL by varying the latent dimension size l in $\{25, 50, 100, 150, 200, 300\}$. We can see that PARL performs consistently well in a wide range of l values (*i.e.,* $[25, 300]$) with little performance variation. As it can be seen, the best performance of PARL on most datasets is achieved when the l is set to 50. Accordingly we set $l = 50$ in the experiments. For small datasets, *i.e.,* Beer and Office Products, the *MSE* is slightly increasing when l is larger than 150. This is probably caused by the overfitting of the model.

The Impact of λ. Recall that, in Equation 20, we use λ to control the degree of the agreement between $\mathbf{t}_{aux}$ and $\mathbf{t}_u$. In Figure 6, we plot the *MSE* of PARL on Beer and Office Products when varying λ from 0.001 to 0.5. The results of other datasets are similar to that of Office Products. We can observe that PARL achieves the best performance when λ is set to relatively small value (*i.e.,* ranges from 0.005 to 0.03). When the value of λ is larger than 0.07, the *MSE* of both datasets are increasing significantly. This degradation may be incurred by the fact that few extra information would be extracted when $\mathbf{t}_{aux}$ is too similar to $\mathbf{t}_u$. On the contrary, the features extracted from user auxiliary review documents is not compatible to the features extracted from user review document when λ is too small. To sum up, smaller λ should be selected when the user review document is short, while larger λ is better for long user review document.

4.5 Model Ablation of PARL

To get a better understanding of the proposed PARL model, we further evaluate the effectiveness of the key components of PARL in this section. The ablation study is conducted by orderly stacking different layers on top of the DeepCoNN model at a time. Table 3 reports the effectiveness of different layers of PARL. Note that, DeepCoNN + bias can be viewed as DeepCoNN plus the user and item bias factors. + Aux uses three parallel CNN networks to infer latent representation from user review document, item review document and user auxiliary review document respectively. The model is trained according to Equation 20. PARL is completed until the abstracting layer is incorporated. According to Table 3, each component contributes positively for rating prediction and significant improvement is obtained when incorporating Aux and Abstracting into the model.

Besides, recall that the parameters of PARL and DeepCoNN are jointly learnt during training time. However, [10] shows that the initialization of parameters is essential for deep learning model. We first train DeepCoNN with random initiliaztion and then plug PARL into the pre-trained DeepCoNN to verify the effectiveness of pre-training on PARL. The improvement achieved by + pre-training demonstrates the usefulness of pre-trained strategy for PARL, especially when the textual data is dense.

Table 3: Impact of different components in PARL. The best result are highlighted in boldface.

Methods	Beer	Office Products	Digital Music	Video Games	Tools Improvement
DeepCoNN	0.617	0.860	1.060	1.238	1.063
+bias	0.589	0.783	0.888	1.141	0.995
+Aux	0.578	0.755	0.870	1.130	0.973
+Filtering	0.570	0.743	0.866	1.125	0.964
+Abstracting	0.561	0.731	0.849	1.117	0.955
+Pre-training	**0.559**	**0.725**	**0.842**	**1.109**	**0.951**

4.6 Study Case Visualization

In this section, we would like to investigate whether PARL can extract informative pair-dependent features from auxiliary reviews on the basis of target user-item pair.

We randomly pick up some $user_u$-$item_i$ pairs from the test set of Beer, and regard each auxiliary review contained in u's auxiliary review document as an independent auxiliary review document of u. Then the rating prediction is conducted by PARL upon the $user_u$-$item_i$ pairs with those independent auxiliary review documents respectively. Finally, the auxiliary reviews are ranked according to the prediction accuracy. More related information carried by the auxiliary reviews will make PARL perform more accurate prediction and thus win a high-ranking for the auxiliary reviews.

One example is listed in Table 4. Here, the column Target Reviews indicates the actual review that user u writes for item i, the column Auxiliary Reviews is the auxiliary reviews related to user u. We can see that, high-ranked reviews contain more informative features compared to the reviews in low-ranked position. For example, the 1st-ranked auxiliary review uses synonymous words to depict the head of the beer. The 2nd-ranked auxiliary review hits the words "white head" and "nice finish" in the original reviews. Besides, it is clear that the sentiment between the high-ranked auxiliary reviews and the target review are positive correlated very well compared to the low-ranked auxiliary reviews. These observations confirm that PARL is capable of extracting useful semantic features from the user auxiliary review document to make the rating prediction more accurate.

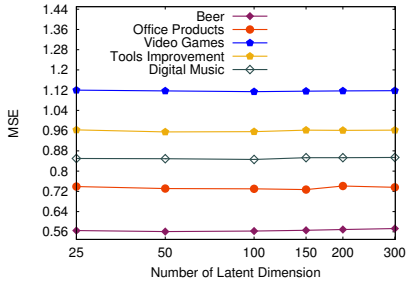
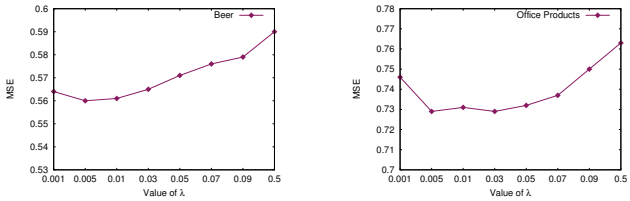


Figure 5: Impact of dimension number l across the five datasets.



(a) Beer (b) Office Products

Figure 6: Impact of λ on Beer (a) and Office Products (b).

There are also some badcases in Table 4. For example, we find that shorter review are more likely to earn a high-ranking. One possible reason is that the noisy information contained in long reviews tends to penalize the ranking of the reviews. Additionally, the 15th review should be ranked higher than the 11th review because it conveys consistent sentiment with the target reivew. We leave this case for our future work. Generally speaking, PARL achieves satisfactory prediction performance by effectively extrating informative features from users' auxiliary reviews.

5 CONCLUSION

In this paper, we propose a deep learning model called PARL which exploits informative features in the reviews written by other users to improve the performance of rating prediction. With the utilization of user auxiliary review document, PARL can alleviate the textual sparsity problem of recommender system. To the best of our knowledge, PARL is the first model to tackle the data sparsity problem of reviews data. It can be plugged into different review-based method to further improve the performance. Extensive experiments validate that the performance of PARL is robust on various datasets with different characteristics. In the future, we would like to apply attention mechanism to attend informative features and reduce the impact from the noisy features in auxiliary reviews to further improve the prediction performance.

Table 4: Target Reviews v.s. Auxiliary Reviews with different ranking ordered by the prediction accuracy. The informative features and the incompatible features are manually highlighted by orange and gray color.

Target Reviews	Rank	Auxiliary Reviews
Pours dark brown with a very thin white head that quickly recedes. Aroma is of sweet malt. Taste is also of sweet malt with a lingering sweet finish. A nice, solid beer that I will definately drink again.	1	Amber with small head. Incredibly easy to drink. A little bit of hops. Would make an excellent brew for a night of billiards with friends.
Pours dark brown with a very thin white head that quickly recedes. Aroma is of sweet malt. Taste is also of sweet malt with a lingering sweet finish. A nice, solid beer that I will definately drink again.	2	poured a cloudy amber with an off white head. Had an aroma of citrus, hops and grain. Had a nice bitter finish
⋮	⋮	⋮
Pours dark brown with a very thin white head that quickly recedes. Aroma is of sweet malt. Taste is also of sweet malt with a lingering sweet finish. A nice, solid beer that I will definately drink again.	6	On tap at Founders. Nice aroma of chocolate, coffee, vanilla, and oak. Great dark appearance with a creamy, everlasting head. Great beer, something I come to expect from Founders!!
⋮	⋮	⋮
Pours dark brown with a very thin white head that quickly recedes. Aroma is of sweet malt. Taste is also of sweet malt with a lingering sweet finish. A nice, solid beer that I will definately drink again.	11	Way overrated on this site. Porterhouses description is very accurate. Maybe this was old cause it does indeed taste like vinegar. The foamy head takes forever to simmer down. If someone knows how to find a good bottle let me know.
⋮	⋮	⋮
Pours dark brown with a very thin white head that quickly recedes. Aroma is of sweet malt. Taste is also of sweet malt with a lingering sweet finish. A nice, solid beer that I will definately drink again.	15	The foam crown does not look so good, in color it is gold-yellow. In the taste hoppy s and a little alcoholic but fits well! Ingenious Saftigkeit, such a Sffigen Bock I did not have! The aftertaste resembles the taste in the mouth.

ACKNOWLEDGEMENT

This research was supported by National Natural Science Foundation of China (No. 61472287, No. 61502344), Natural Scientific Research Program of Wuhan University (No. 2042017kf0225), Science and Technology Planning Project of Wuhan(2016060101010047), and Fund of Hubei Key Laboratory of Transportation Internet of Things(WHUTIOT-2017A0011). Chenliang Li is the corresponding author.

REFERENCES

[1] Iman Barjasteh, Rana Forsati, Farzan Masrour, Abdol-Hossein Esfahanian, and Hayder Radha. 2015. Cold-Start Item and User Recommendation with Decoupled Completion and Transduction. In *RecSys*.
[2] Robert M. Bell and Yehuda Koren. 2007. Scalable Collaborative Filtering with Jointly Derived Neighborhood Interpolation Weights. In *ICDM*.
[3] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *JMLR* (2003).
[4] Rose Catherine and William W. Cohen. 2017. TransNets: Learning to Transform for Recommendation. In *RecSys*.
[5] Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2018. Neural Attentional Rating Regression with Review-level Explanations. In *WWW*.
[6] Jingyuan Chen, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu, and Tat-Seng Chua. 2017. Attentive Collaborative Filtering: Multimedia Recommendation with Item- and Component-Level Attention. In *SIGIR*.
[7] Kostadin Georgiev and Preslav Nakov. 2013. A non-IID Framework for Collaborative Filtering with Restricted Boltzmann Machines. In *ICML*.
[8] Guibing Guo, Jie Zhang, and Neil Yorke-Smith. 2015. TrustSVD: Collaborative Filtering with Both the Explicit and Implicit Influence of User Trust and of Item Ratings. In *AAAI*.
[9] Ruining He and Julian McAuley. 2016. Ups and Downs: Modeling the Visual Evolution of Fashion Trends with One-Class Collaborative Filtering. In *WWW*.
[10] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *WWW*.
[11] Jonathan L. Herlocker, Joseph A. Konstan, Al Borchers, and John Riedl. 1999. An Algorithmic Framework for Performing Collaborative Filtering. In *SIGIR*.
[12] Mohsen Jamali and Martin Ester. 2009. *TrustWalker*: a random walk model for combining trust-based and item-based recommendation. In *SIGKDD*.
[13] Dong Hyun Kim, Chanyoung Park, Jinoh Oh, Sungyoung Lee, and Hwanjo Yu. 2016. Convolutional Matrix Factorization for Document Context-Aware Recommendation. In *RecSys*.
[14] Yehuda Koren. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *KDD*.
[15] Yehuda Koren, Robert M. Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *IEEE Computer* 8 (2009).
[16] Hui Li, Dingming Wu, Wenbin Tang, and Nikos Mamoulis. 2015. Overlapping Community Regularization for Rating Prediction in Social Recommender Systems. In *RecSys*.
[17] Piji Li, Zihao Wang, Zhaochun Ren, Lidong Bing, and Wai Lam. 2017. Neural Rating Regression with Abstractive Tips Generation for Recommendation. In *SIGIR*.
[18] Dawen Liang, Jaan Altosaar, Laurent Charlin, and David M. Blei. 2016. Factorization Meets the Item Embedding: Regularizing Matrix Factorization with Item Co-occurrence. In *RecSys*.
[19] Guang Ling, Michael R. Lyu, and Irwin King. 2014. Ratings meet reviews, a combined approach to recommend. In *RecSys*.
[20] Julian J. McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In *RecSys*.
[21] Julian J. McAuley, Jure Leskovec, and Dan Jurafsky. 2012. Learning Attitudes and Attributes from Multi-aspect Reviews. In *ICDM*.
[22] N. Ng, R. Gabriel, J. McAuley, C. Elkan, and Z. Lipton. 2017. Predicting surgery duration with neural heteroscedastic regression. In *MLHC*.
[23] Steffen Rendle. 2010. Factorization Machines. In *ICDM*.
[24] Ruslan Salakhutdinov and Andriy Mnih. 2007. Probabilistic Matrix Factorization. In *NIPS*.
[25] Ruslan Salakhutdinov, Andriy Mnih, and Geoffrey E. Hinton. 2007. Restricted Boltzmann machines for collaborative filtering. In *ICML*.
[26] Sungyong Seo, Jing Huang, Hao Yang, and Yan Liu. 2017. Interpretable Convolutional Neural Networks with Dual Local and Global Attention for Review Rating Prediction. In *RecSys*.
[27] Kai Shu, Suhang Wang, Jiliang Tang, Yilin Wang, and Huan Liu. 2018. CrossFire: Cross Media Joint Friend and Item Recommendations. In *WSDM*.
[28] Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *JMLR* 1 (2014).
[29] Xiaoyuan Su and Taghi M. Khoshgoftaar. 2009. A Survey of Collaborative Filtering Techniques. *Adv. Artificial Intellegence* (2009).
[30] Yunzhi Tan, Min Zhang, Yiqun Liu, and Shaoping Ma. 2016. Rating-Boosted Latent Topics: Understanding Users and Items with Ratings and Reviews. In *IJCAI*.
[31] Jiliang Tang, Suhang Wang, Xia Hu, Dawei Yin, Yingzhou Bi, Yi Chang, and Huan Liu. 2016. Recommendation with Social Dimensions. In *AAAI*.
[32] Tijmen Tieleman and Geoffrey Hinton. 2012. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning* 2 (2012).
[33] Chong Wang and David M. Blei. 2011. Collaborative topic modeling for recommending scientific articles. In *KDD*.
[34] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. 2015. Collaborative Deep Learning for Recommender Systems. In *KDD*.
[35] Libing Wu, Cong Quan, Chenliang Li, Qian Wang, and Bolong Zheng. 2017. A Context-Aware User-Item Representation Learning for Item Recommendation. *CoRR* abs/1712.02342 (2017).
[36] Wei Zhang, Quan Yuan, Jiawei Han, and Jianyong Wang. 2016. Collaborative Multi-Level Embedding Learning from Reviews for Rating Prediction. In *IJCAI*.
[37] Lei Zheng, Vahid Noroozi, and Philip S. Yu. 2017. Joint Deep Modeling of Users and Items Using Reviews for Recommendation. In *WSDM*.