Data & Knowledge Engineering 150 (2024) 102283

Contents lists available at ScienceDirect

Data & Knowledge Engineering

journal homepage: www.elsevier.com/locate/datak

Fusion learning of preference and bias from ratings and reviews for item recommendation

Junrui Liu [a], Tong Li [a,*], Zhen Yang [a], Di Wu [a], Huan Liu [b]

[a] Faculty of Information Technology, Beijing University of Technology, 100124, China
[b] Computer Science and Engineering, Arizona State University, Tempe, 878809, AZ, USA

ARTICLE INFO

Keywords:
Data mining
Text mining
Recommender systems
Selection bias
Interaction feature

ABSTRACT

Recommendation methods improve rating prediction performance by learning selection bias phenomenon-users tend to rate items they like. These methods model selection bias by calculating the propensities of ratings, but inaccurate propensity could introduce more noise, fail to model selection bias, and reduce prediction performance. We argue that learning interaction features can effectively model selection bias and improve model performance, as interaction features explain the reason of the trend. Reviews can be used to model interaction features because they have a strong intrinsic correlation with user interests and item interactions. In this study, we propose a preference- and bias-oriented fusion learning model (PBFL) that models the interaction features based on reviews and user preferences to make rating predictions. Our proposal both embeds traditional user preferences in reviews, interactions, and ratings and considers word distribution bias and review quoting to model interaction features. Six real-world datasets are used to demonstrate effectiveness and performance. PBFL achieves an average improvement of 4.46% in root-mean-square error (RMSE) and 3.86% in mean absolute error (MAE) over the best baseline.

1. Introduction

Recommender systems have increasing importance in human life. Selection bias exists in most recommendation data because rating data are often missing not at random (MNAR) [1]. For example, users usually rate and comment on movies that they like and rarely rate movies that they do not like [2]. Modeling selection bias can effectively improve recommendation performance [3].

To improve rating prediction performance, some methods are proposed to model selection bias. IPS [1] is a principled method that deal with selection biases. It learns unbiased propensities to determine the effect of the prediction error of observed data on the loss function. DR-JL [4] model integrates the imputed errors and propensities in a doubly robust way. MRDR [5] is a more robust doubly robust method than DR-JL. MRDR uses two post-click conversion rate models to enhance the accuracy of error imputation. These methods consider the propensity, a probability of observed rating, and require relatively accurate click-through rate estimation. However, the propensity is not easy to calculate. Inaccurate propensity introduces more noise and fails to model selection bias, ultimately leading to a reduction in prediction performance.

We consider selection bias as a part of the interaction features. Selection bias is the statistical result of interactions between users and items. It reflects the likelihood that a user is inclined to select an item. The interaction features reflect why the user interacts

* Corresponding author.
E-mail addresses: liujunrui@emails.bjut.edu.cn (J. Liu), litong@bjut.edu.cn (T. Li), yangzhen@bjut.edu.cn (Z. Yang), wuxiaodou@emails.bjut.edu.cn (D. Wu), huanliu@asu.edu (H. Liu).

https://doi.org/10.1016/j.datak.2024.102283
Received 13 September 2022; Received in revised form 24 December 2023; Accepted 1 February 2024
Available online 3 February 2024
0169-023X/

with the item through latent features. Therefore, modeling interaction features can inherently reflect and explain selection bias. Thus, we model interaction features to model selection bias and improve model performance.

We argue that reviews can be used to learn the interaction features as they have a strong intrinsic correlation with user interests and item interactions, and many studies tend to improve the performance of review-based recommendations [6–8]. The real intent can help recommender systems identify the interaction features of users and items to improve prediction performance. There are two features to help model interaction features by reviews. (1) Word distribution bias: a prior probability of word distribution that is domain dependent [9,10]. It can provide prior knowledge at the word level to judge the likelihood of interaction between users and items, while words in reviews often include some aspect that users are usually interested in [11,12]. (2) Review quoting: users will refer to the opinions of others when users comment on items [6,13]. Review quoting indicates the user's endorsement of an opinion in a review. By learning semantic similarities in historical reviews, the likelihood of interactions between users and items can be learned.

In this paper, we propose a preference- and bias-oriented fusion learning (PBFL) model to learn interaction features and three types of user preferences to make rating predictions. Different from the IPS study on interaction probability, PBFL both models traditional preference and interaction features. Specifically, PBFL has three components to model and fuse the preference and bias from reviews to ratings. First, a word-level convolutional attention network embeds each review. We use word-level user and item embeddings as part of the attention score to tackle word distribution bias. An attention mechanism in this network is used to adjust the review embedding, the aggregation result of word embedding. Since the words in reviews are usually biased, the attention mechanism can fuse the word distribution bias and word-level preference. Then, a review-level user–item interaction learning network models the selection bias and produces interaction feature vectors. We design a recurrent aggregated attention layer in this network to tackle the review quoting. By looping through similar reviews, this layer can find citation relationships between reviews. At the same time, we use interaction-level embeddings to implicitly learn the effect of the various external factors on user interaction behaviors. The interaction feature vectors fuse the review quoting features and interaction-level preference. Finally, a rating prediction network makes predictions by using the interaction feature vectors and rating-level preference embeddings.

The contributions to the literature in this paper are as follows.

- We propose a preference- and bias-oriented fusion learning model to model the interaction features and a three-level user preference to make rating predictions.
- We propose a recurrent aggregated attention layer that discovers citation relationships between reviews by calculating the similarity between user reviews and item reviews.
- We perform several experiments on six real-world datasets, the results of which demonstrate that PBFL outperforms state-of-the-art review-based and bias-based recommendation methods.

2. Related works

In this section, we discuss rating-based recommendation methods and review-based methods, and research two review features that are useful for modeling interaction features.

2.1. Recommendation with rating

Rating prediction based on bias has a long history in recommendation systems since matrix factorization (MF) was proposed [14]. Rating bias is a variation that is independent of any interactions [14,15]. The variation is due to effects associated with either users or items. Individual level bias instability influences various online decisions [14,16]. Its prediction function has four terms: global mean, item bias, user bias, and user–item interaction. Pham et al. [17] propose that ratings do not always represent user preferences. Instead, they use features to determine the consistency of ratings and user preferences. Krishnan et al. [18] identify that the displayed average rating of an item has an important influence on user decisions.

Further research showed that selection bias, which is the embodiment of user choice and is from MNAR data, affects rating bias. In MNAR data, the observed probability of each rating is not equal, and users tend to rate their favorite items. Rating bias reflects how biased a rating is, while selection bias reflects the possibility of rating bias. Thus, researchers focus on the selection bias. IPS [1] is a principled method that learns unbiased propensities to determine the effect of the prediction error of the observed data on the loss function. To model MNAR data, DR-JL [4] integrates the imputed errors and propensities in a doubly robust way. However, DR-JL could increase the variance of the IPS under inaccurate error imputation. This makes the learning process complicated and leads to sub-optimal results. MRDR [5] is a more robust doubly robust method than DR-JL. It uses two post-click conversion rate models to enhance the accuracy of error imputation. Such methods rely on the success of the propensities estimator, but it is not an easy approach by which to achieve the desired outcome since error propensities can also decrease the model performance.

2.2. Recommendation with reviews

Analyzing reviews is an important way to ascertain the logic of user behavior. There are two types of developing preference-based recommender systems that combine ratings and reviews, as follows.

The first types focus on using natural language processing (NLP) models to extract accurate text features. CTR employs latent Dirichlet allocation to learn the topics of texts [19]. CDL [20] uses a stacked denoising auto encoder to embed texts. The text features

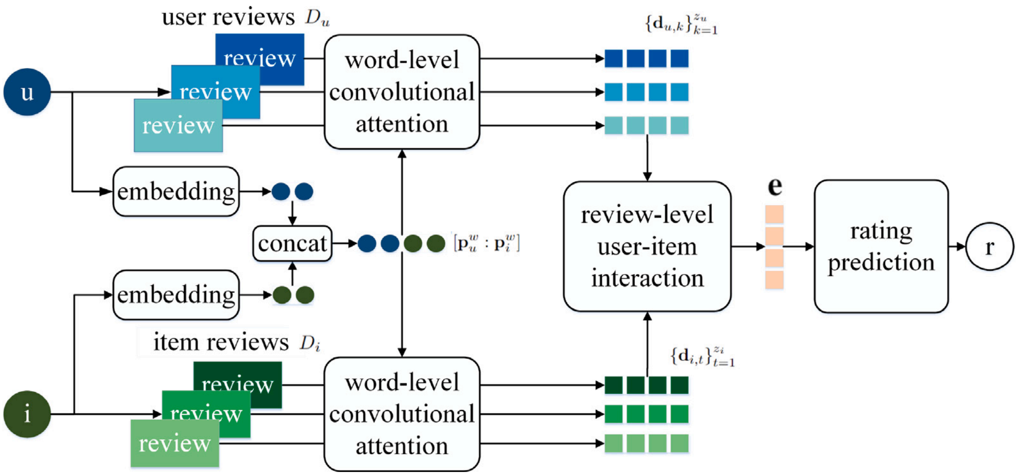

Fig. 1. Overall architecture of the proposed model.

are the item-side features used to calculate prediction ratings. However, these methods ignore the word-order information [8]. ConvMF [21] and DeepCoNN [22] use convolutional neural networks (CNN) to extract high-level text features to overcome the above limitation. NARRE [6] uses attention to determine the influence of reviews in which user embedding is the aggregation of the important reviews. EPR [13] notes that only several pieces will influence the potential users in decision-making among numerous reviews. It divides the influence of a review into three-facet influences: general user influence, personalized user influence, and review-related influence. TAERT [23] uses a temporal convolutional network to learn the sequential characteristics in reviews. RGNN [12] builds a review graph for each user that nodes are words and edges are word orders. It uses a type-aware graph attention network to summarize graph information and a personalized graph pooling to capture important words. These methods focus on extracting text features but ignore learning interaction features based on text features [24].

The second type focuses on using an attention mechanism to learn preferences in interactions. The attention mechanism captures the interaction between users and items, filters out less valuable reviews, and captures essential reviews. CARL [25] uses an attention mechanism to encode context information for each user–item pair. D-attn [26] uses two attention mechanisms to learn the local and global interaction features. DAML [7] uses mutual attention to model the relationship between users' and items' reviews. HTI [24] transfers textual features between users and items. ARPM [27] learns user preferences about review aspects and sentiment analysis. DRRNN [28] proposes that considering both rating and review for error backpropagation in the training stage can capture more review information than other review-based methods. DSRNL [29] extracts dynamic user interests by stacking attention layers that deal with sequence features. Similar to Transformers [30], DSRNL adds temporal dependencies on sequence features. However, these methods focus on preferences and ignore the effect of bias [3,14,31].

3. Problem formalization

In a recommendation problem, assume that a user set $\mathcal{U} = \{u^1, u^2, \dots, u^m\}$ and an item set $\mathcal{I} = \{i^1, i^2, \dots, i^n\}$ contains m users and n items, respectively. r_{ui} denotes the true rating of user u of item i, $\hat{r}_{ui}$ is the prediction from a model, and $R \in \mathbb{R}^{m*n}$ denotes the rating matrix. For each user–item pair (u_i, v_j), we denote $D_u = \{d_{u,1}, d_{u,2}, \dots, d_{u,z_u}\}$ as the set of user reviews, and $D_i = \{d_{i,1}, d_{i,2}, \dots, d_{j,z_i}\}$ as the set of item reviews, where z_u and z_i denote the sizes of the sets D_u and D_i respectively. The model make an prediction using (u, i) and D_u, D_i.

4. Method

PBFL has three components, i,e., a word-level convolutional attention network, a review-level user–item interaction learning network, and a rating prediction network. The architecture of PBFL is illustrated in Fig. 1. First, the word-level convolutional attention network embeds each review. We track the word distribution bias through joint user and item preferences of words as part of the attention score.

Review embedding is the fusion of the word distribution bias and word-level preference. Then, the review-level user–item interaction learning network models the selection bias and produces interaction feature vectors. The interaction feature vector is the fusion of the review quoting features and interaction-level preference. A recurrent aggregated attention layer calculates the similarity between user reviews and item reviews to model the review quoting features. The interaction-level preference implicitly learns the effect of various external factors on interaction behavior. Finally, the rating prediction network makes predictions by using the interaction feature vectors and rating-level preference.

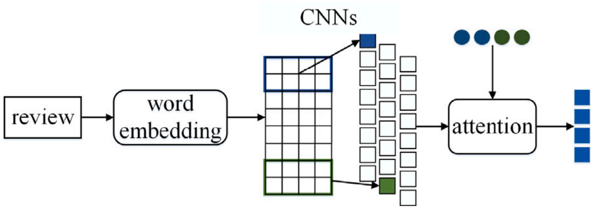

Fig. 2. Word-level convolutional attention network.

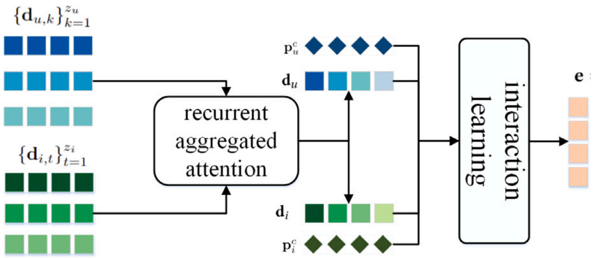

Fig. 3. Recurrent aggregated attention layer.

4.1. Word-level convolutional attention network

We first embed each word and then use CNN to extract text features. Then, an attention mechanism is used to aggregate these features. The process of this network is illustrated in Fig. 2. In the attention mechanism, we use user and item embedding as a part of the attention score. Word distribution bias is a prior probability of word distribution that is domain dependent [10]. Existing methods [7,8] use GloVe as the pre-trained word embedding model, which is trained on a large corpus. The word importance in user reviews is different from that in the corpus because of the word distribution bias. The proposed can adjust the word importance and fuse user preference and word distribution bias.

$d = \{w_x\}_{x=1}^{l}$ denotes a review of length l and $\{\mathbf{w}_x\}_{x=1}^{l}$ is the word embedding sequence in d. We then apply CNN to capture the high-level word features $\mathbf{c}_x$.

Attention. $\mathbf{p}_u^w$ and $\mathbf{p}_i^w$ denote word-level preference embeddings of user u and item i, respectively, which are used to compute the attention scores so that PBFL can fuse the word distribution bias and user preference. Word patterns in reviews do not always consistently mirror user preferences and they may have some inherent subtle complexities. Thus, we use an MLP to fuse the word-level preference embeddings and then obtain a joint preference vector at the word level. The final attention score is the inner product of the joint preference vector and the word feature vector. Thus, the score is influenced by the semantics of the word and the joint preference vector. Through the backpropagation algorithm, the MLP automatically adjusts the weights of the preference vectors to adapt to the inherent complexity in reviews. Specifically, the review representation can be aggregated as the weighted sum over the word vectors:

$$\mathbf{d} = \sum_{x=1}^{l} \alpha_x \mathbf{c}_x, \tag{1}$$

$$\alpha_x = \frac{exp(\beta^T \mathbf{c}_x)}{\sum_{x=1}^{l} exp(\beta^T \mathbf{c}_x)}, \tag{2}$$

$$\beta = f_{MLP1}^{tanh}([\mathbf{p}_u^w : \mathbf{p}_i^w]), \tag{3}$$

where f_{MLP1}^{tanh} is a one-layer multi-layer perceptron (MLP) to fuse the word-level preference.

4.2. Review-level user–item interaction network

As shown in Fig. 3, a review-level user–item interaction network aggregates the review vectors and produces interaction embedding vectors. First, a recurrent aggregated attention layer aggregates review vectors to represent users and items. Users will refer to the opinions of others when users comment on items [13], so the aggregation representation includes the latent bias. At the same time, reviews may be influenced by various external factors, which might result in discrepancies, reducing the effect of review features. Thus, review-based representations can only represent a subset of features of users and items, as users may express their opinions implicitly. If the interaction features are fully based on aggregating review vectors, they could miss important information in interaction behaviors. Considering the various external factors, we utilize an embedding layer to learn the implicit user bias in interaction behaviors and then design an interaction learning layer to fuse review features and the interaction embedding.

4.2.1. Recurrent aggregated attention layer

The recurrent aggregated attention layer models the review relationship and produces review aggregated representations for users and items, respectively. In each loop, we compute the distance between user reviews and item reviews as attention scores to adjust the representation of the reviews. Then, we aggregate the review representations into a vector by the attention scores. This vector represents the user's (item's) review information. As a mutual learning mechanism, semantic information is aggregated in similar reviews. Similar semantic information models that users will refer to the opinions of others when users comment on items [13].

Review distance. We use Euclidean distance to calculate the review distance and determine the likelihood of citations between reviews [24]:

$$\begin{aligned} e_{k,t}^{(\eta)} = |\mathbf{d}_{u,k}^{(\eta-1)} - \mathbf{d}_{i,t}^{(\eta-1)}|,& \\ k = 1, \dots, z_u;& \\ t = 1, \dots, z_i;& \\ \eta = 1, \dots, H,& \end{aligned} \tag{4}$$

where $e_{k,t}$ is the Euclidean distance between the kth user review and tth item review. The initial representations $\mathbf{d}_{u,k}^{(0)}$ and $\mathbf{d}_{i,t}^{(0)}$ are generated by the word-level convolutional attention network, as specified in the preceding Section 4.1. More discussion about review distance can be found in Section 5.5.1.

Recurrent representation. We use a_k to indicate the closest distance between the kth user review and all the item reviews. A review with small a_k is assigned a large attention score for representing the recurrent representation of user reviews:

$$\begin{aligned} a_k^{(\eta)} &= min(e_{k,1}^{(\eta)}, e_{k,2}^{(\eta)}, \dots, e_{k,n}^{(\eta)}), \\ \delta_{u,k}^{(\eta)} &= \frac{exp(-a_k^{(\eta)})}{\sum_{k=1}^{z_u} exp(-a_k^{(\eta)})}, \\ \mathbf{t}_i^{(\eta)} &= \sum_{k=1}^{z_u} \delta_{u,k}^{(\eta)} \mathbf{d}_{u,k}^{(\eta-1)}, \\ \mathbf{d}_{u,k}^{(\eta)} &= \lambda \mathbf{d}_{u,k}^{(\eta-1)} + (1-\lambda)\mathbf{t}_u^{(\eta)}, \\ \mathbf{h}_u^{(\eta)} &= \lambda \mathbf{h}_u^{(\eta-1)} + (1-\lambda)\mathbf{t}_u^{(\eta)}, \end{aligned} \tag{5}$$

where λ is a hyper-parameter.

Similar to the recurrent representation of user reviews, the recurrent representation of item reviews is expressed as:

$$\begin{aligned} b_t^{(\eta)} &= min(e_{1,t}^{(\eta)}, e_{2,t}^{(\eta)}, \dots, e_{m,t}^{(\eta)}), \\ \delta_{i,t}^{(\eta)} &= \frac{exp(-b_t^{(\eta)})}{\sum_{t=1}^{z_i} exp(-b_t^{(\eta)})}, \\ \mathbf{t}_i^{(\eta)} &= \sum_{k=1}^{z_i} \delta_{i,t}^{(\eta)} \mathbf{d}_{i,t}^{(\eta-1)}, \\ \mathbf{d}_{i,t}^{(\eta)} &= \lambda \mathbf{d}_{i,t}^{(\eta-1)} + (1-\lambda)\mathbf{t}_i^{(\eta)}, \\ \mathbf{h}_i^{(\eta)} &= \lambda \mathbf{h}_i^{(\eta-1)} + (1-\lambda)\mathbf{t}_i^{(\eta)}, \end{aligned} \tag{6}$$

where b_t indicates the closest distance between the tth item review and user reviews.

The initial representations $\mathbf{h}_u^1 = \mathbf{t}_u^{(1)}$ and $\mathbf{h}_i^1 = \mathbf{t}_i^{(1)}$. The recurrent representation of the item (i.e., i) reviews can be obtained by following (5). Finally, we use Hth representations for the output of the network, as follows:

$$\mathbf{d}_u = \mathbf{h}_u^{(H)}, \quad \mathbf{d}_i = \mathbf{h}_i^{(H)}. \tag{7}$$

4.2.2. Interaction learning layer

The previous layer learns the review features of users and items, where the review features include the review quoting features. However, user–item interactions are influenced by various factors, which are not fully reflected by reviews. If the interaction features are fully based on aggregating review vectors, they could miss important information in interaction behaviors. Thus, the interaction learning layer is designed to implicitly learn preferences that are missing from the reviews. Here, we add an interaction embedding layer for users and items and denote the interaction preference embedding of u and i as $\mathbf{p}_u^c$ and $\mathbf{p}_i^c$, respectively. We merge review features and preference features and then input them to an MLP to obtain the final interaction features. Thus, the embedding layer implicitly learns the effect of various factors on interactions beyond those contained in the reviews.

$$\begin{aligned} \mathbf{e} &= f_{MLP}(\mathbf{p}_u^c, \mathbf{d}_u, \mathbf{p}_i^c, \mathbf{d}_i) \\ &= \phi_{out}^{elu}(\ \phi_x^{elu}(\ \dots\ \phi_2^{elu}(\ \phi_1^{elu}([\mathbf{p}_u^c\ :\ \mathbf{d}_u\ :\ \mathbf{p}_i^c\ :\ \mathbf{d}_i])\ )\ \dots\ )\ ), \end{aligned} \tag{8}$$

where ϕ_{out}^{elu} and ϕ_x^{elu} denote the mapping function for the output layer and the xth neural network layer, respectively. The form of a mapping layer is $\phi(\mathbf{y}) = g(W \cdot \mathbf{y} + b)$, where W is the weights, b is the bias term, and g is an activate function. As user interaction data

Table 1
Statistical results of six datasets, where $\overline{z_u}$ and $\overline{z_i}$ indicate the average number of reviews per user and item, respectively, and $\bar{l}$ denotes the average number of words per review.

Dataset	No. of users	No. of items	No. of ratings	$\overline{z_u}$	$\overline{z_i}$	$\bar{l}$	Density
Musical instruments	1,429	900	10,261	10	17	59	0.798%
Office products	4905	2420	53,228	7	11	56	0.449%
Digital Music	5540	3568	64 666	11	18	112	0.327%
Grocery and gourmet food	14,681	8,713	151,254	10	22	89	0.118%
Video games	24,303	10,672	231,577	9	21	124	0.089%
Sports and outdoors	35,598	18,357	296,337	8	14	99	0.045%

are observational rather than experimental, bias is easily introduced into the data [32]. The bias term is used to fit the potential bias. By interactively coupling review vectors and embeddings, a more suitable interaction feature vector can be obtained.

4.3. Prediction learning network

The superiority of deep-learning methods validates the advantage of nonlinear functions. NeuMF [33] is a neural network-based extension, which is expressed as follows:

$$\hat{r}_{NeuMF} = MLP(p_u, q_i). \tag{9}$$

Based on NeuMF, the prediction function of PBFL is expressed as

$$\begin{aligned}\hat{r}_{PBFL} &= MLP(\mathbf{e}, \mathbf{p}_u^r, \mathbf{p}_i^r) \\ &= \phi_{out}^{elu}(\ \phi_x^{elu}(\ \dots\ \phi_2^{elu}(\ \phi_1^{elu}([\mathbf{e} : \mathbf{p}_u^r : \mathbf{p}_i^r])\ )\ \dots\ )\ ),\end{aligned} \tag{10}$$

where $\mathbf{p}_u^r$ and $\mathbf{p}_i^r$ denote rating-level preference embeddings of u_i and v_j, respectively. We use the *concatenate* function to fuse these factors.

4.4. Time complexity analysis

In the word-level convolutional attention network, the time complexity is $O(d * |CNN| * (z_u + z_i))$, where d is the embedding dimension, $|CNN|$ is the time complexity of the CNN, and z_u and z_i denote the number of reviews of u and i, respectively. In the review-level user–item interaction network, the time complexity is $O((z_u * z_i * d)^H + d^{|MLP|})$, where $|MLP|$ denotes the number of the MLP. In the prediction learning network, the time complexity is $O(d^{|MLP|})$. In summary, the total time complexity of PBFL is $O(d * |CNN| * (z_u + z_i) + (z_u * z_i * d)^H + d^{|MLP|})$, and it is related to the dimension and review number. Although the above time complexity is complex, many of these terms are computed in parallel in deep-learning models, such as CNN and MLP.

5. Experiments

We conducted experiments to show the effectiveness of the proposed model. First, we illustrate the details of datasets, baselines, and experimental settings. We then compare our methods with baselines, and finally, identify the effects of several designs on PBFL.

5.1. Datasets

Experiments were conducted on six publicly available Amazon 5-core datasets[1] to validate the effectiveness of the proposed model: Music Instruments (MI), Office Products (OP), Digital Music (DM), Grocery and Gourmet Food (GGF), Video Games (VG), and Sports and Outdoors (SO). The datasets are sourced from various domains, and as a result, users within each dataset exhibit distinct behavioral patterns. We first processed all words into the lower cases, filtered out non-alphabetic characters by a regularization function and stop words by NLTK tools [24,25]. The vocabulary of FBFL has 20,000 words that occur frequently in reviews. Finally, we zero-padded reviews with zero if needed [6], and reviews were removed in the test set. The statistics of all datasets are presented in Table 1.

5.2. Baselines

We compared PBFL with the following state-of-the-art models and divided the baselines into three classes. The first class only uses rating features, and includes PMF [34], NeuMF [33], IPS [1], DR-JL [4], and MRDR [5]. PMF is a probabilistic method based on MF while NeuMF is a neural network method based on MF. The last two classes use both reviews and ratings and are discussed in Section 2. The second class focuses on improving the accuracy of review features, and includes CDL [20], ConvMF [21], DeepCoNN(DCN) [22], ERP [13], TAERT [23], and RGNN [12]. The third class models the interaction between users and items, and includes D-attn [26], CARL [25], NARRE [6], DAML [7], HTI [24] and DSRNL [29].

[1] http://jmcauley.ucsd.edu/data/amazon/

Table 2
RMSE and MAE on six datasets. Best and second-best results are emboldened and underlined, respectively.

		MI		OP		DM		GGF		VG		SO	
		RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
rating	PMF	1.352	1.137	1.430	1.265	1.063	0.805	1.572	1.397	1.606	1.395	1.376	1.203
	NeuMF	0.904	0.720	0.921	0.730	0.999	0.766	1.197	0.943	1.103	0.870	0.979	0.752
	IPS	0.937	0.667	0.871	0.666	0.953	0.733	1.027	0.802	1.136	0.893	0.954	0.742
	DR-JL	0.942	0.667	0.858	0.642	0.929	0.706	1.010	0.778	1.108	0.871	0.942	0.727
	MRDR	0.964	0.682	0.903	0.664	0.966	0.698	1.061	0.779	1.097	0.832	0.923	0.673
review	CDL	1.080	0.834	1.223	1.062	1.039	0.783	1.190	0.967	1.165	0.902	1.089	0.852
	ConvMF	1.026	0.786	0.952	0.728	0.940	0.673	1.092	0.863	1.145	0.899	1.013	0.824
	DCN	1.003	0.759	0.901	0.711	0.894	0.662	1.036	0.802	1.168	0.875	0.885	0.719
	NARRE	0.922	0.695	0.867	0.681	0.889	0.673	0.963	0.747	1.039	0.799	0.882	0.690
	ERP	0.914	0.643	0.920	0.725	1.115	0.864	1.074	0.835	1.195	0.938	0.976	0.680
	TAERT	<u>0.695</u>	<u>0.601</u>	1.059	0.604	0.885	0.663	0.931	0.692	0.990	<u>0.687</u>	0.834	<u>0.594</u>
	RGNN	0.859	0.605	0.860	0.623	0.905	<u>0.655</u>	0.999	0.710	1.046	0.766	0.936	0.646
interaction	D-attn	0.956	0.742	0.923	0.716	1.277	0.810	1.070	0.824	1.062	0.842	0.997	0.784
	CARL	0.878	0.677	0.834	0.647	0.948	0.743	0.961	0.753	1.029	0.798	0.888	0.686
	DAML	0.848	0.651	0.811	0.612	0.913	0.664	0.938	0.735	1.045	0.788	0.883	0.667
	HTI	0.813	0.611	<u>0.731</u>	<u>0.552</u>	1.034	0.666	<u>0.877</u>	<u>0.672</u>	<u>0.966</u>	0.731	<u>0.826</u>	0.629
	DSRNL	1.036	0.620	0.890	0.621	<u>0.879</u>	0.672	0.952	0.721	0.983	0.754	0.900	0.651
	PBFL	**0.654**	**0.595**	**0.672**	**0.525**	**0.855**	**0.653**	**0.841**	**0.631**	**0.926**	**0.646**	**0.810**	**0.563**

5.3. Experimental setup

We used GloVe-100 [35] to initialize the word embedding. Following existing work [24], the kernel sizes of the two CNNs are 3 and 5, and the prediction network has 3 layers that combines four embeddings to make prediction. Each dimension of the prediction network reduces one embedding size and is $300, 200, 100$, respectively. The dropout rate is 0.5 and the batch size 128. We used the Adam optimizer with an initial learning rate of 0.0001.

Each experiment is repeated five times, and the results reported are the average of the five runs. In each training process, each dataset was uniformly randomly shuffled and divided into three parts, i.e., a training set (first 80%), validation set (middle 10%), and test set (last 10%). The validation set is used to tune hyperparameters. The performance on the test set was evaluated by root-mean-square error (RMSE) and mean absolute error (MAE):

$$RMSE = \sqrt{\frac{1}{|Ds|} \sum_{u,i,r_{ij} \in Ds} \left(r_{ij} - \hat{r}_{ij}\right)^2}, \tag{11}$$

$$MAE = \frac{1}{|Ds|} \sum_{u,i,r_{ij} \in Ds} |r_{ij} - \hat{r}_{ij}|. \tag{12}$$

5.4. Performance comparison

Table 2 lists the performance results for all baselines. Bold type highlights the best results and underlines the second-best results. The proposed method, i.e., PBFL, outperforms the baselines on all six datasets. On average, PBFL improves RMSE by 4.46% and MAE 3.86% compared to the best baseline. These results indicate that the proposed method is effective for rating prediction on datasets with different characteristics. Moreover, the significant performance gap between PBFL and the best baseline validates that fusing learning with preference and bias captures more user behaviors by learning the correlation between reviews and ratings.

Some methods only use ratings. PMF is a basic method and is easily affected by bias. NeuMF outperforms PMF by a margin on all the datasets, demonstrating the effectiveness of multilayer nonlinear transformations for capturing complex user–item relations. Compared to PMF and NeuMF, IPS, DR-JL, and MRDR consider the propensity, a probability of observed ratings, achieving significant performance improvement. These methods have better performance than other rating-based methods and some review-based methods. They effectively improve model preference by modeling selection bias via calculating the propensity. However, the performance of MRDR is not better than DR-JL on MI and OP datasets. MRDR aims to reduce the variance of DR loss to enhance model robustness, but it may still have poor generalization when the bias is large. IPS, DR-JL, and MRDR only consider the impact of selection bias in ratings. On the one hand, the propensity of ratings is not easy to calculate. Inaccurate propensity could introduce more noise, reducing the prediction performance; on the other hand, selection bias is one type of interaction feature, while improving model performance should consider more types. Compared to these methods, PBFL introduces an interaction feature vector, which learns user–item interaction from reviews, and reduces the impact of the rating-level preference vectors, while these methods re-weight the vectors by considering the propensity.

Some methods use reviews to enhance the performance of rating predictions. CDL is a basic review-based method that uses LDA to learn latent representations of items from reviews. However, LDA is a word-bag method, ignoring the word-order information. Compared to CDL, ConvMF uses CNN to extract high-level features for items and achieves better performance than CDL. Since CDL and ConvMF both use MF to make predictions, the performance improvement can be attributed to CNN. The difference between

Table 3
RMSE on six datasets with different review distances.

	MI	OP	DM	GGF	VG	SO
COS	**0.635**	0.848	0.894	0.970	**0.858**	<u>0.820</u>
KL	0.708	<u>0.677</u>	0.915	<u>0.941</u>	1.014	0.889
ED	<u>0.654</u>	**0.672**	0.855	**0.841**	<u>0.926</u>	**0.810**

Table 4
MAE on six datasets with different review distances.

	MI	OP	DM	GGF	VG	SO
COS	**0.580**	<u>0.618</u>	0.655	<u>0.711</u>	0.789	0.660
KL	0.596	0.622	0.651	0.721	<u>0.780</u>	<u>0.640</u>
ED	<u>0.595</u>	**0.525**	0.653	**0.631**	**0.646**	**0.563**

ConvMF and DCN is that DCN uses CNN to learn representations for users, while ConvMF uses latent vectors. CNN efficiently captures the local semantics of different levels of granularities. Most importantly, CNNs have become the usual method of processing text. Based on DCN, NARRE distinguishes the importance of reviews, further improving the performance. TAERT introduces a temporal convolutional network (TCN) to capture the sequence characteristics. It achieves the second-best performance in several datasets because TCN involves neighbor feature interactions, global feature interactions, and sequential characteristics in user embeddings.

Interaction-based methods learn interaction features based on reviews and are mostly better than other methods. We can observe that the improvements gained by CARL are consistent and stable. On average, the relative improvement of CARL against NARRE is 4.55%. In the learning process of interaction features, CARL introduces preference embeddings to justify the weights of reviews. The comparison between HTI and previous models, such as DAML, demonstrates the effectiveness of the hierarchical architecture, that formally divides the review-based recommendation process into three stages. Insufficient review and interaction data could have hindered the ability of DSRNL to learn accurate user and item representations. Compared to these methods, PBFL adopts the three-stage architecture and uses three individual embedding layers to learn latent preferences. At the same time, we design the recurrent aggregated attention layer, which models the review quoting and improves the accuracy of interaction features, thereby improving model performance.

5.5. Analysis of PBFL model

We conducted a series of experiments to verify the designs proposed in this paper. First, we verified the effect of the recurrent aggregated attention layer, which models review quoting, by determining the impact of the review distance measure and the depth of this layer. Second, we verified the several preference embeddings used in different networks. Finally, we analyzed the real-time complexity of the proposed model.

5.5.1. Impact of different measures of reviews distance

In the recurrent aggregated attention layer, we calculate the review distance, which is used to identify the importance of each review for a user. In this experiment, we want to find the optimal one from three common distance measures, i.e., cosine similarity, Kullback–Leibler divergence, and Euclidean distance. Cosine similarity (COS) measures the similarity between two vectors and maps it into $[-1, 1]$. It is measured by the cosine of the angle between two vectors. Thus, it can avoid the effect of the length of vectors. Kullback–Leibler divergence (KL) is the number of bits required to convert one distribution into another. The label of origin review is missed, so we calculate the KL for two sides of each review pair (for example, $KL(A \parallel B)$ and $KL(B \parallel A)$). It makes the distance directional and allows the user and item sides to have different distances for a review pair. In this paper, we use a word-embedding method GloVe to embed words. The geometrical properties of word embeddings are designed to capture semantic and syntactic relationships between words. Therefore, using this representation allows for interesting mathematical operations, such as measuring the similarity between two words by calculating the difference between their vectors. Euclidean distance (ED) is suitable for measuring the similarity of continuous features, so we measure the effect of ED.

Experiment result is shown in Tables 3 and 4. From the result, we can see that ED method achieves the best performance in most datasets. PBFL embeds words with GloVe model, extracts high-level features with CNNs, and fuses these features to represent reviews via vectors. The GloVe model uses vectors to represent words, which leads to the Euclidean distance between two vectors having special means, such as "King - Man = Queen - Women". Through CNNs and the fusion process, the review vectors keep the semantic features. Compared with COS and KLD, ED emphasizes the impact of varied continuous dimensions on outcomes, helping models distinguish between non-irrelevant reviews. COS distance ignores the length of vectors, which leads it to be inaccurate. Although KL is bidirectional, it cannot help recommender systems determine which review is the origin review. An original review does not always have a small KL distance than other reviews.

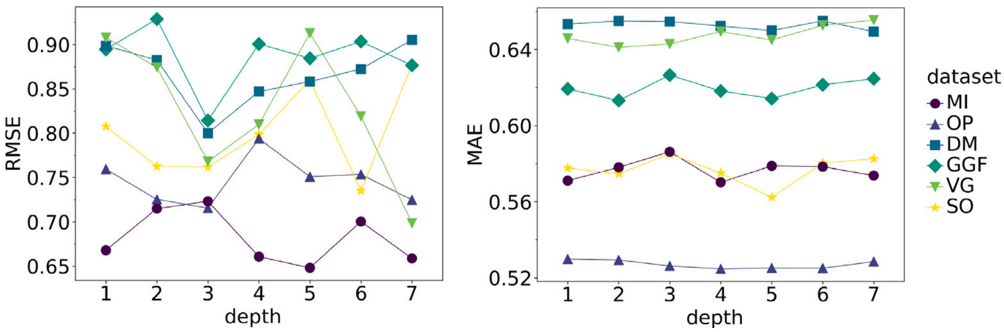

Fig. 4. Performance with respect to different depths of recurrent attention network.

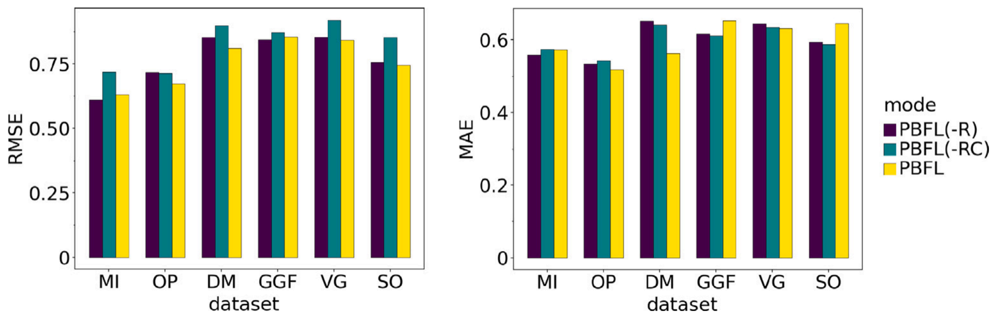

Fig. 5. Impact of preference embedding at difference levels across datasets.

5.5.2. *Impact of recurrent aggregated attention layer*

Fig. 4 shows the performance comparison with recurrent aggregated attention layer depth ranging from 1 to 7. It can be seen from the figure that the effect of recurrent aggregated attention layer depth on RMSE and MAE is different, as follows.

- RMSE is more sensitive to changes in depth, and the effect of depth varies across datasets. The model performances of the 6 datasets exhibit 2 types of distributions. In DM, GGF, and SO datasets, the PBFL achieves the best RMSE when the depth of the recurrent attention network is set to 3. Conversely, in MI, OP, and VG datasets, a higher depth value leads to better performance for PBLF. For the first type, the three datasets belong to the personal life domain and have a big average number of words per review. Users in this domain tend to express their personalized viewpoints, relying less on references from other reviews. In the second type, the three datasets belong to the special domains. MI and OP datasets have a small average number of words per review. For the VG dataset, users are usually more active than in other fields, and they are more likely to express their feelings about a game. In this type, users engage in deeper communications and carefully consider other users' opinions. For example, if a user wants to buy a musical instrument, they are likely to listen to other users' opinions and carefully decide.
- The effect of depth on MAE is small. In most cases, setting the depth of the network at 4 achieved the best performance.

As the network depth increases, the performance first becomes better and then worse. The best result is achieved for a depth of approximately 3 and, in some cases, 6. This shows that the recurrent attention network can indeed capture the situation in which users refer to other people's reviews when they themselves write a review. As such, there is a specific correlation between different reviews.

5.5.3. *Impact of preference embedding at difference levels*

We analyzed the effect of the preference embedding at different levels, for which PBFL(-R) denotes that we removed the rating preference embedding and PBFL(-RC) that we removed the rating and interaction preference embedding. Fig. 5 shows the performance results. On most datasets, PBFL exhibited the best performance, demonstrating that PBFL's fusion of different levels' preference embeddings can effectively improve the model preference. On the MI dataset, PBFL(-R) performed better than PBFL. Reviews and rating behaviors were more consistent on the musical instruments dataset. After adding the rating preference vector, the convergence ability of the model was reduced. Overall, PBFL uses multiple embeddings to achieve better results than single embedding.

5.5.4. *Ablation study*

We conducted an ablation experiment to study the contributions of different networks in PBFL more accurately. By removing the networks in PBFL, this experiment identifies their contributions. In this subsection, we use the first letter of each network to

Table 5
The experiment results of ablation study.

	MI		OP		DM		GGF		VG		SO	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
PBFL(-WR)	0.929	0.621	0.860	0.618	0.904	0.650	1.028	0.719	1.053	0.779	0.935	0.652
PBFL(-W)	0.751	0.624	0.834	0.619	0.858	0.648	0.987	0.708	1.047	0.790	0.830	0.658
PBFL(-R)	0.853	0.594	0.868	0.619	0.916	0.643	0.974	0.713	1.009	0.779	0.881	0.649
PBFL(avg)	0.888	0.596	0.836	0.621	0.916	0.649	0.940	0.714	1.027	0.781	0.905	0.644
PBFL	**0.654**	**0.595**	**0.672**	**0.525**	**0.855**	**0.653**	**0.841**	**0.631**	**0.926**	**0.646**	**0.810**	**0.563**

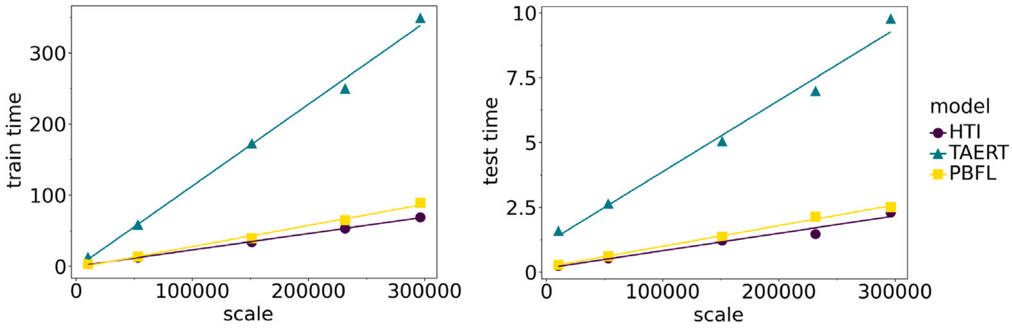

Fig. 6. Time cost on datasets of different scales.

represent that network. Thus, W denotes the word-level convolutional attention network, R denotes the review-level user–item interaction network, and P denotes the prediction learning network. PBFL(-WR) denotes that we only use the prediction learning network to make predictions; PBFL(-W) denotes that we remove the word-level convolutional attention network in PBFL; PBFL(-R) denotes that we remove the review-level user-item interaction network; and PBFL(avg) denotes that we use an average function instead of the recurrent aggregated attention layer.

The experiment results are shown in Table 5. We regard PBFL(-WR) as the basic to analyze in those variants. Compared to PBFL(-WR), PBFL(-W) has the review-level user–item interaction network to learn latent interaction features, making it significantly outperforms better than PBFL(-WR). As the network utilizes interaction-level embeddings to model user behaviors, the above results indicate that the review-level user–item interaction network implicitly learns user–item interaction features. Considering the results of PBFL(-R) and PBFL, we find that reviews offer special information, but if we fully base on reviews to model the user–item interaction features without the review-level user–item interaction network, the model performs not well. The review-level user–item interaction network accurately aggregates the review vectors and latent representations to produce interaction embedding vectors. The recurrent aggregated attention layer has a big effect on PBFL. When we remove it, the performance of PBFL significantly reduces, as shown by PBFL(avg). Users sometimes refer to other users' reviews, and are affected by these reviews. PBFL more accurately aggregates reviews to represent users and items through the recurrent aggregated attention layer, and then learns interaction features based on these aggregations. In a word, PBFL outperforms all variants better.

5.5.5. Time complexity analysis

To study the time efficiency of the proposed model, we compare its running time with two representative baselines. We compared the training cost with the HTI and TAERT models, which were trained with an NVIDIA GeForce RTX 3090 graphics card with 24 GB memory. The training time is the per epoch cost, and the prediction time is the time required to complete the prediction for the entire testing set. The experimental results are shown in Fig. 6. Compared with TAERT, PBFL achieved a significant performance improvement with little time cost. PBFL incurs slightly higher overhead than HTI, mainly due to a novel recurrent aggregated attention layer. Since this layer undergoes multiple iterative processes, PBFL exhibits a higher time complexity than HTI. As the size of the dataset increases, the time complexity exhibits a linear growth with a slight increment. The time complexity of the recurrent aggregated attention layer is $O((z_u * z_i * d)^H)$. It depends on its depth H and the number of reviews related to users and items. For scenarios with high time requirements, the value of H can be reduced, and appropriate filtering of user and item reviews can be conducted to reduce the number of reviews involved in the calculation.

5.6. Visualization

We further investigated whether PBFL can capture important features at different levels of hierarchies, and visualized the attention scores both at the word and review levels. In addition, we randomly sample one rating record.

Fig. 7 shows the visualization result of four review-level attention scores. The main features of games under review include plot, gameplay, sound, price, and multiplayer model. We highlighted sentences with such features using different colors. Based on the

Fig. 7. Review level visualization. Four reviews were analyzed in the rating record $(3475, 9267, 5)$ to obtain review-level attention scores. Several key sentences are marked in color. The main features of games include plot, gameplay, sound, multiplayer model, and price, denoted with red, yellow, cyan, purple, and green, respectively.

Fig. 8. Word level visualization. Two reviews were analyzed in the record $(3475, 9267, 5)$. The attention scores are divided into three types: high scores are marked with a large font size and red color, middle scores with a medium font size and orange color, and small scores with a small font size and black color.

visualization, we can see that the review-level recurrent aggregated attention layer tends to give high scores for informative and useful reviews. The content between these reviews is not intersecting. Specifically, review1 is awarded the highest score because it contains four game features. Review2 is mainly concerned with price. Compared with other games, low-priced and well-made games can give players a better gaming experience and psychological satisfaction. Review3 discussed the plot and the multiplayer model. Review4 is general praise of the naughty dog without any actual content, so it is awarded the lowest score. Through multiple loop computations, the review-level recurrent aggregated attention layer generally learns review quoting and generalizes specific semantic information. Attention scores also illustrate the importance of semantics.

In word-level visualization (Fig. 8), the informative words have high attention scores. For example, the words "combat", "tlou", "black Friday", "cheap" and "shooter" are assigned larger attention scores than others. These words strongly indicate users' interest. The proposed method can learn some keywords from the biased review text to characterize users.

Overall, this study shows that the proposed method can capture different review features, and those features describe why a user chose a specific item. By capturing those features, PBFL learns an interaction feature to improve the recommendation performance.

5.7. Discussion

This paper proposes a PBFL model that uses reviews to learn the interaction features. Based on reviews, PBFL models the effect of word distribution bias and review quoting in the process of review-based recommendation. In this section, we discuss some important assumptions and experimental phenomenons.

First, PBFL partly relies on the assumption that reviews have a strong and consistent intrinsic correlation with user interests and item interactions. However, it is worth mentioning that reviews can be influenced by various external factors. Thus, overfitting reviews are not always suitable. For example, Noemi et al. mention that good reviews might fail to get feedback from readers that could depend on factors such as age (recentness) and level of visibility on the websites [36]. Although extracting user preference from reviews improves model performance, recommendation methods should carefully identify the helpfulness of reviews. To avoid overfitting reviews, PBFL introduces an interaction-level embedding layer to extract latent interaction behaviors. The unconstrained embedding vector layer cannot guarantee the optimal solution. Therefore, researchers can explore more specific external factors and design corresponding constraint methods to optimize the embedding layer.

Second, PBFL models the word distribution bias by introducing user and item embeddings to justify the weights of words. The weights are used to identify the importance of words in aggregating words to represent reviews. Although users have their preferences, such as writing style, in reviews, some subtle complexities are still ignored. Except for some common aspects like price and size, a specific field may have some proper nouns. Reviews may also contain sarcasm, leading to inconsistent semantic content and ratings. These phenomena require more complex analysis and corresponding designs.

Third, PBFL improves model performance with more model complexity than before. That increases the time complexity. This phenomenon does not only exist in PBFL but also in most review-based methods. Although the model's time cost can be addressed by increasing computing power, optimizing the model's time cost is still a practical requirement for recommendation algorithms. Some works provide that every sufficiently overparameterized (dense) neural network contains a subnetwork that, even without training, achieves accuracy comparable to that of the trained large network [37,38]. PBFL adopts several dense layers and CNNs that can be pruned to reduce the time cost.

6. Conclusions

In this study, we propose a PBFL method, which uses reviews to learn the interaction features and model selection bias. To tackle word distribution bias, we use word-level embeddings of users and items as a part of the attention score in an attention mechanism. The attention mechanism adjusts the importance of words and aggregates words to represent reviews. We then design a recurrent aggregated attention layer in this network to tackle the review quoting. By looping through similar reviews, this layer can find citation relationships between reviews. Based on the citation relationships, we aggregate reviews to represent users and items. The interaction feature is the fusion of the review-based representations and latent interaction-level embeddings. PBFL uses MLP to make predictions from the interaction feature and rating-level embeddings. The experimental results demonstrate that PBFL performs significantly better than other state-of-the-art models. In addition, we validate the effectiveness of several designs and preference embeddings.

PBFL ignores some phenomena, such as the effect of external factors on reviews and interaction behaviors. In future work, we will explore more specific external factors and design corresponding constraint methods to optimize the embedding layer in PBFL. At the same time, the experiment identifies that the time complexity of PBFL has linearly increased with the increasing data scale, which affects the scalability of PBFL in larger datasets. We plan to explore a pruning method that reduces the time cost of PBFL.

CRediT authorship contribution statement

Junrui Liu: Data curation, Writing – original draft, Conceptualization, Methodology, Investigation. **Tong Li:** Methodology, Funding acquisition, Supervision, Investigation, Writing – review & editing. **Zhen Yang:** Project administration, Funding acquisition, Methodology. **Di Wu:** Visualization, Writing – review & editing. **Huan Liu:** Methodology, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The dataset is publicly available, which can be found tin the manuscript.

Acknowledgments

This work is partially supported by the National Key R&D Program of China (No. 2022YFB3103100), the Major Research Plan of the National Natural Science Foundation of China (92167102), the Importation and Development of High-Caliber Talents Project of Beijing Municipal Institutions, China (CIT&TCD20190308), the Project of Beijing Municipal Education Commission, China (No. KM202110005025), and Beijing Natural Science Foundation Project, China (No. Z200002).

References

[1] T. Schnabel, A. Swaminathan, A. Singh, N. Chandak, T. Joachims, Recommendations as treatments: Debiasing learning and evaluation, in: M. Balcan, K.Q. Weinberger (Eds.), Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016, in: JMLR Workshop and Conference Proceedings, vol. 48, JMLR.org, 2016, pp. 1670–1679, URL http://proceedings.mlr.press/v48/schnabel16.html.

[2] B. Pradel, N. Usunier, P. Gallinari, Ranking with non-random missing ratings: influence of popularity and positivity on evaluation metrics, in: P. Cunningham, N.J. Hurley, I. Guy, S.S. Anand (Eds.), Sixth ACM Conference on Recommender Systems, RecSys '12, Dublin, Ireland, September 9-13, 2012, ACM, 2012, pp. 147–154, http://dx.doi.org/10.1145/2365952.2365982.

[3] J. Chen, X. Wang, F. Feng, X. He, Bias issues and solutions in recommender system: Tutorial on the RecSys 2021, in: Proceeding 15th ACM Conference on Recommender Systems, 2021, pp. 825–827.

[4] X. Wang, R. Zhang, Y. Sun, J. Qi, Doubly robust joint learning for recommendation on data missing not at random, in: International Conference on Machine Learning, PMLR, 2019, pp. 6638–6647.

[5] S. Guo, L. Zou, Y. Liu, W. Ye, S. Cheng, S. Wang, H. Chen, D. Yin, Y. Chang, Enhanced doubly robust learning for debiasing post-click conversion rate estimation, in: F. Diaz, C. Shah, T. Suel, P. Castells, R. Jones, T. Sakai (Eds.), SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021, ACM, 2021, pp. 275–284, http://dx.doi.org/10.1145/3404835.3462917.

[6] C. Chen, M. Zhang, Y. Liu, S. Ma, Neural attentional rating regression with review-level explanations, in: Proceedings of the 2018 World Wide Web Conference, 2018, pp. 1583–1592.

[7] D. Liu, J. Li, B. Du, J. Chang, R. Gao, Daml: Dual attention mutual learning between ratings and reviews for item recommendation, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 344–352.

[8] J. Wen, Z. Zhang, Z. Zhang, L. Fei, M. Wang, Generalized incomplete multiview clustering with flexible locality structure diffusion, IEEE Trans. Cybern. 51 (1) (2020) 101–114.

[9] M. Yang, D. Zhu, K.-P. Chow, A topic model for building fine-grained domain-specific emotion lexicon, in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 2014, pp. 421–426.

[10] C. Xing, W. Wu, Y. Wu, J. Liu, Y. Huang, M. Zhou, W. Ma, Topic aware neural response generation, in: S. Singh, S. Markovitch (Eds.), Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA, AAAI Press, 2017, pp. 3351–3357, URL http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14563.

[11] X. He, T. Chen, M. Kan, X. Chen, TriRank: Review-aware explainable recommendation by modeling aspects, in: J. Bailey, A. Moffat, C.C. Aggarwal, M. de Rijke, R. Kumar, V. Murdock, T.K. Sellis, J.X. Yu (Eds.), Proceedings of the 24th ACM International Conference on Information and Knowledge Management, CIKM 2015, Melbourne, VIC, Australia, October 19 - 23, 2015, ACM, 2015, pp. 1661–1670, http://dx.doi.org/10.1145/2806416.2806504.

[12] Y. Liu, S. Yang, Y. Zhang, C. Miao, Z. Nie, J. Zhang, Learning hierarchical review graph representations for recommendation, IEEE Trans. Knowl. Data Eng. (2021).

[13] S. Wu, Y. Zhang, W. Zhang, K. Bian, B. Cui, Enhanced review-based rating prediction by exploiting aside information and user influence, Knowl.-Based Syst. (2021) 107015.

[14] Y. Koren, R. Bell, C. Volinsky, Matrix factorization techniques for recommender systems, Computer 42 (8) (2009) 30–37.

[15] M. Ochi, Y. Matsuo, M. Okabe, R. Onai, Rating prediction by correcting user rating bias, in: 2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology, vol. 1, IEEE, 2012, pp. 452–456.

[16] S. Mukhopadhyay, T.S. Chung, Preference instability, consumption and online rating behavior, Int. J. Res. Market. 33 (3) (2016) 624–638.

[17] H.X. Pham, J.J. Jung, Preference-based user rating correction process for interactive recommendation systems, Multimedia Tools Appl. 65 (1) (2013) 119–132.

[18] S. Krishnan, J. Patel, M.J. Franklin, K. Goldberg, A methodology for learning, analyzing, and mitigating social influence bias in recommender systems, in: Proceedings of the 8th ACM Conference on Recommender Systems, 2014, pp. 137–144.

[19] C. Wang, D.M. Blei, Collaborative topic modeling for recommending scientific articles, in: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2011, pp. 448–456.

[20] H. Wang, N. Wang, D.-Y. Yeung, Collaborative deep learning for recommender systems, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2015, pp. 1235–1244.

[21] D. Kim, C. Park, J. Oh, S. Lee, H. Yu, Convolutional matrix factorization for document context-aware recommendation, in: Proceedings of the 10th ACM Conference on Recommender Systems, 2016, pp. 233–240.

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

[23] S. Guo, Y. Wang, H. Yuan, Z. Huang, J. Chen, X. Wang, TAERT: Triple-attentional explainable recommendation with temporal convolutional network, Inform. Sci. 567 (2021) 185–200, http://dx.doi.org/10.1016/j.ins.2021.03.034.

[24] J. Wen, J. Ma, H. Tu, M. Zhong, G. Zhang, W. Yin, J. Fang, Hierarchical text interaction for rating prediction, Knowl.-Based Syst. 206 (2020) 106344.

[25] L. Wu, C. Quan, C. Li, Q. Wang, B. Zheng, X. Luo, A context-aware user-item representation learning for item recommendation, ACM Trans. Inf. Syst. (TOIS) 37 (2) (2019) 1–29.

[26] S. Seo, J. Huang, H. Yang, Y. Liu, Interpretable convolutional neural networks with dual local and global attention for review rating prediction, in: Proceedings of the Eleventh ACM Conference on Recommender Systems, 2017, pp. 297–305.

[27] C.-H. Lai, C.-Y. Hsu, Rating prediction based on combination of review mining and user preference analysis, Inf. Syst. 99 (2021) 101742.

[28] W.-D. Xi, L. Huang, C.-D. Wang, Y.-Y. Zheng, J.-H. Lai, Deep rating and review neural network for item recommendation, IEEE Trans. Neural Netw. Learn. Syst. (2021) 1–11, http://dx.doi.org/10.1109/TNNLS.2021.3083264.

[29] T. Liu, S. Lou, J. Liao, H. Feng, Dynamic and static representation learning network for recommendation, IEEE Trans. Neural Netw. Learn. Syst. (2022) 1–11, http://dx.doi.org/10.1109/TNNLS.2022.3177611.

[30] W. Kang, J.J. McAuley, Self-attentive sequential recommendation, in: IEEE International Conference on Data Mining, ICDM 2018, Singapore, November 17-20, 2018, IEEE Computer Society, 2018, pp. 197–206, http://dx.doi.org/10.1109/ICDM.2018.00035.

[31] T. Zheng, F. Wu, R. Law, Q. Qiu, R. Wu, Identifying unreliable online hospitality reviews with biased user-given ratings: A deep learning forecasting approach, Int. J. Hosp. Manag. 92 (2021) 102658.

[32] J. Chen, H. Dong, X. Wang, F. Feng, M. Wang, X. He, Bias and debias in recommender system: A survey and future directions, 2020, arXiv preprint arXiv:2010.03240.

[33] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, T.-S. Chua, Neural collaborative filtering, in: Proceedings of the 26th International Conference on World Wide Web, 2017, pp. 173–182.

[34] A. Mnih, R.R. Salakhutdinov, Probabilistic matrix factorization, in: Advances in Neural Information Processing Systems, 2008, pp. 1257–1264.

[35] J. Pennington, R. Socher, C.D. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP, 2014, pp. 1532–1543.

[36] N. Mauro, L. Ardissono, G. Petrone, User and item-aware estimation of review helpfulness, Inf. Process. Manag. 58 (1) (2021) 102434, http://dx.doi.org/10.1016/J.IPM.2020.102434.

[37] J. Frankle, M. Carbin, The lottery ticket hypothesis: Finding sparse, trainable neural networks, in: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, la, USA, May 6-9, 2019, OpenReview.net, 2019, URL https://openreview.net/forum?id=rJl-b3RcF7.

[38] A.C.W. da Cunha, E. Natale, L. Viennot, Proving the lottery ticket hypothesis for convolutional neural networks, in: The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022, OpenReview.net, 2022, URL https://openreview.net/forum?id=Vjki79-619-.

Junrui Liu is currently pursuing the Ph.D. degree in college of computer science and technology at Beijing University of Technology, Beijing, China. His research interests include data mining and recommender systems.

Tong Li is an Associate Professor at the Faculty of Information Technology, Beijing University of Technology, China. He has been an author or co-author of more than 80 papers in peer-reviewed journals, conferences, or workshops in the areas of requirements engineering, security engineering, and data mining. He is now hosting a National Natural Science Foundation of China, a subtask of a National Key Research and Development Program of China, and a project of Beijing Municipal Education Commission. He is a security expert of ISO/IEC JTC 1/ SC 27, and is co-editing two international standard drafts.

Zhen Yang is currently a full professor of computer science and engineering at Beijing University of Technology. He received the Ph.D. degree in signal processing from the Beijing University of Posts and Telecommunications. His research interests include data mining, machine learning, trusted computing, and content security. He has published more than 30 papers in highly ranked journals and top conference proceedings. He is a senior Member of the Chinese Institute of Electronics and a member of the IEEE.

Di Wu is currently pursuing the Ph.D. degree in college of computer science and technology at Beijing University of Technology, Beijing, China. Her research interests include many-objective algorithm and knowledge graph embedding.

Dr. Huan Liu is a professor of Computer Science and Engineering at Arizona State University. He obtained his Ph.D. in Computer Science at University of Southern California and B.Eng. in Computer Science and Electrical Engineering at Shanghai JiaoTong University. Before he joined ASU, he worked at Telecom Australia Research Labs and was on the faculty at National University of Singapore. At Arizona State University, he was recognized for excellence in teaching and research in Computer Science and Engineering and received the 2014 President's Award for Innovation. He is the recipient of the ACM SIGKDD 2022 Innovation Award. His research interests are in data mining, machine learning, feature selection, social computing, social media mining, and artificial intelligence, investigating interdisciplinary problems that arise in many real-world, data-intensive applications with high-dimensional data of disparate forms such as social media. His well-cited publications include books, book chapters, encyclopedia entries as well as conference and journal papers. He is a co-author of a text, Social Media Mining: An Introduction, Cambridge University Press. He is a founding organizer of the International Conference Series on Social Computing, Behavioral-Cultural Modeling, and Prediction, Editor in Chief of ACM TIST, and Field Chief Editor of Frontiers in Big Data and its Specialty Chief Editor of Data Mining and Management. He is a Fellow of ACM, AAAI, AAAS, and IEEE.