

A Multi-Factor Collaborative Prediction for Review-based Recommendation

Junrui Liu
Beijing University of Technology
Beijing, Beijing, China
liujunrui@emails.bjut.edu.cn

Tong Li*
Beijing University of Technology
Beijing, Beijing, China
litong@bjut.edu.cn

Mingliang Yu
TravelSky Technology Limited
Beijing, Beijing, China
mlyu@travelsky.com

Shiqiu Yang
Beijing University of Technology
Beijing, Beijing, China
ysqbjut@emails.bjut.edu.cn

Zifang Tang
Beijing University of Technology
Beijing, Beijing, China
zifangtang@emails.bjut.edu.cn

Zhen Yang
Beijing University of Technology
Beijing, Beijing, China
yangzhen@bjut.edu.cn

Abstract

For items, the higher the click-through rate, the higher the rating. Thus, existing recommendation methods implicitly model click behaviors by learning user preferences and achieving accurate predictions on rating prediction tasks. However, they ignore the help of the rating behaviors for the click-through rate prediction task (CTR). Although the rating behavior occurs after the click behavior, we can still get helpful information about clicks from ratings. In this paper, we propose a multi-factor collaborative prediction method (MFC), which mines the complex relationship between click and rating behaviors, achieving accurate prediction on CTR tasks. Specifically, we factorize the complex relationship into three simple relationships, i.e., linear, sharing, and cross-correlation relationships. Thus, MFC first extracts click factors, rating factors, and their sharing factor from user click and rating behaviors with user reviews, as review-based methods have achieved great results on rating predictions. Then, a rating factor regularization method is used to learn rating factors accurately, helping to model the true relationships between click and rating behavior. Finally, MFC combines those three factors to make predictions, while click and rating factors are used to model the linear and cross-correlation relationships, and the sharing factors correspond to the sharing relation. Experiments on five real-world datasets demonstrate that MFC outperforms the best baseline by 9.19%, 9.80%, 0.69%, and 7.95%, in terms of Accuracy, Precision, Recall, and F1-score, respectively. MFC also reduces the MAE of the rating prediction task by 1.92%. The source code is available at https://github.com/dianziliu/MFC.

CCS Concepts

• **Information systems** → **Recommender systems**.

*Corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

RecSys '25, Prague, Czech Republic

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-1364-4/25/09
https://doi.org/10.1145/3705328.3748062

Keywords

multi-factor collaborative analysis, CTR prediction, user reviews

ACM Reference Format:
Junrui Liu, Tong Li, Mingliang Yu, Shiqiu Yang, Zifang Tang, and Zhen Yang. 2025. A Multi-Factor Collaborative Prediction for Review-based Recommendation. In *Proceedings of the Nineteenth ACM Conference on Recommender Systems (RecSys '25), September 22–26, 2025, Prague, Czech Republic.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3705328.3748062

1 Introduction

Recommender systems help users find items that meet their requirements and interests, and they have found wide applications in human life. To help improve the performance of recommender systems, users are encouraged to share their feelings. Take online shopping platforms as an example: users not only rate items but also comment on them after making purchases. These reviews contain significant semantic information that recommender systems can harness[36, 39]. Other users sometimes make informed shopping decisions based on these reviews [12]. Thus, many methods have been developed to improve the review-based recommendation and achieve great rating predictions [3, 32, 40].

Existing review-based recommendation methods can be divided into two categories. The first category focuses on modeling the latent features in reviews. Some methods use natural language process (NLP) techniques to extract review features [10, 28, 30, 31], such as DeepCoNN [42], CARL [37], DAML [11]. Some methods, such as RGNN [16]and MA-GNNs [41], build graphs based on reviews and then employ graph neural networks (GNNs) to learn review features. The second category mainly models the interactions between users and items with review-based representation. RGCL [25] and ReHCL [32] introduce review features into the user-item graph, enhancing the performance of GNNs. D-attn [24], TSE [40], and PBFL [12] implicitly learn click behaviors through review features. Users usually select items they would like to rate, i.e., the higher the click-through rate, the higher the rating. Thus, these methods implicitly model click behaviors, achieving accurate predictions on rating prediction tasks (RP).

However, these methods only focus on RP tasks with the help of click behaviors, but ignore the help of rating behaviors for the click-through rate (CTR) prediction task. Although the rating behavior occurs after the click behavior, we can still get helpful information about clicks from ratings. Figure 1 shows the relationships between

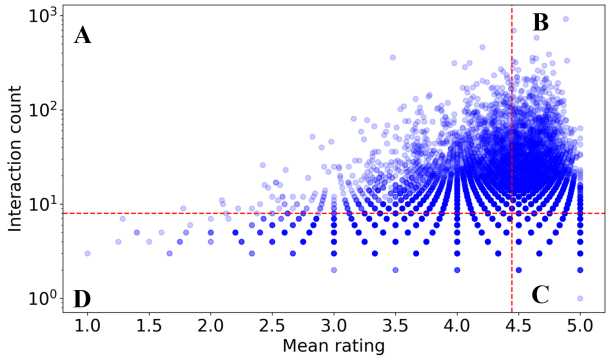


Figure 1: Average rating-click count scatter plot. We marked the median of the mean rating and click count with dotted lines, dividing the plot into 4 regions. These regions reflect the complex relationship between interaction and score features.

mean ratings and the click count of each item in an Amazon dataset. The positive correlation from click counts to ratings is easy to observe: the higher the click count on an item, the higher its average rating. In contrast, without any assistance, it is difficult to identify the effect of ratings on click behaviors. So, we marked the median of the rating and click count with dotted lines, dividing the plot into 4 regions. On the one hand, regions **B** and **D** reflect the approximately linear relationship between rating and click. In this situation, users prefer to click on items they favor, meaning that a high-rating item is more likely to be clicked than a low-rating one [23]. On the other hand, regions **A** and **C** do not comply with the above rules. Region **A** indicates that users have strong demands and will click on an item even if its quality is poor; and region **C** shows that some high-quality items have a small audience and are possibly extremely personalized. These results also show a challenge when we want to use rating behaviors to enhance model performance on the CTR task: a complex relationship exists between interaction and rating behaviors.

In this paper, we propose a multi-factor collaborative prediction method (MFC) for the review-based recommendation on the CTR task. MFC mines the complex and diverse relationships between interaction and rating behaviors based on user reviews, achieving accurate prediction on CTR tasks. Specifically, MFC factorizes the complex relationships between the two behaviors into three simple relationships, i.e., linear, sharing, and cross-correlation relationships. Thus, we first design a factor learning network that extracts click, rating, and sharing factors from user reviews and that models the direct relationship from clicks to ratings. The click and rating factors are then used to model the linear and cross-correlation relationships, while the sharing factors correspond to the sharing relation. At the same time, to ensure rating factors accurately reflect rating behaviors, we propose a rating factor regularization method, which injects supervision information of ratings into the rating factors. Finally, MFC combines those three factors to make predictions, which enhance the model performance on the CTR task by introducing rating factors on CTR predictions. In summary, the main contributions of this article are as follows.

- We design a multi-factor collaborative prediction method (MFC), which mines the complex relationships between click and rating behaviors, achieving accurate prediction on CTR tasks.
- We propose a rating factor regularization method, which applies supervision information of ratings to ensure accurate learning of rating factors.
- Experiments on five real-world datasets demonstrate that the performance of MFC significantly outperforms SOTA baselines. MFC outperform the best baseline by 9.19%, 9.80%, 0.69%, 7.95%, 1.92% on Accuracy, Precision, Recall, F1-score, and MAE, respectively.

2 Related Work

In this section, we summarize collaborative filtering and review-based recommendation methods.

2.1 Collaborative filtering methods

Early recommendation methods, mainly collaborative filtering methods, are mining user preference in ratings to train models. These methods consider the rating prediction in the recommendation problems, and are widely known to researchers by Netflix Prize competition [9]. Rating prediction requires the recommendation model to predict the ratings of users on items. Slight performance enhancements of predictions could significantly improve the recommendations [1, 8]. In these methods, matrix factorization (MF) is a successful and recognized latent factor model that attracts much attention. The variations of MF includes PMF [17], SVD [9], FM [21], etc. PMF is the probabilistic matrix factorization model, which is a classical collaborative filtering-based rating prediction method. SVD is a classic matrix factorization method that exploits both the user's explicit preferences on items and the influences of the user's historical items on the target item.

Another task in the recommender system is the click-through rate prediction (CTR) task. CTR prediction requires the recommendation model to predict the click probability of users on items. Many methods that are designed for PR tasks are also suitable for CTR tasks. In addition, some methods introduce pairwise loss to meet the problem that recommended data does not contain negative items [13, 18, 22]. Compared with previous methods, the pairwise methods no longer pursue the prediction results to be close to the ground trues [19, 20, 29]. They hypothesize that a user prefers observed items over all other non-observed items and then maximizes the relative relationships between items to achieve superior ranking outcomes. For the CTR task, the pairwise method avoids hard constraints on negative samples and can reduce the system error caused by negative sampling, thus achieving better performance.

Recent methods [5, 23, 33] develop a two-step method, that they first use a pre-training model to learn the click propensity through the CTR task, and then the interaction feature is used to influence the gradients in the training of the RP task. But these methods cut down the between CTR and RP tasks.

2.2 Review-based method

Most review-based methods predict ratings by learning user preferences and incorporating NLP techniques. These methods are typically divided into two types by considering their innovations.

The first types focus on using natural language processing (NLP) models to extract accurate text features. Exemplary works include CTR's application of LDA for topic modeling and CDL's implementation of stacked denoising autoencoders (SDAE) for comment embedding [30]. Subsequent advancements employ deep learning architectures: ConvMF [7] and DeepCoNN [42] adopt convolutional neural networks (CNN) to capture hierarchical textual patterns, whereas NARRE [2] incorporates attention mechanisms to weight influential reviews during user embedding aggregation. Recent innovations extend this paradigm through temporal modeling (TAERT [4] with temporal convolutional networks) and structural enhancement (RGNN [16] via review-based graph construction). Despite demonstrating text processing proficiency, these methodologies show limited attention to deriving interaction patterns from textual representations.

The second category emphasizes learning interaction preferences through attention mechanisms. CARL [37] leverages contextual attention to encode user-item pair relationships, while D-attn [24] employs dual attention modules for capturing localized and global interaction patterns. DAML [11] extends this paradigm by modeling cross-modal attention between users and textual reviews. HTI [34] introduces cross-entity attention to characterize textual feature propagation across entities. Building on graph architectures, RGCL [25] and ReHCL [32] enhance graph neural networks through review-aware attention propagation. Recent extensions incorporate multi-granularity modeling: TSE [40] combines global behavioral attention (coarse-grained homophily) with interaction-level attention (fine-grained homophily), whereas DSRNL [15] designs temporal attention layers to model evolving interest sequences. PBFL [12] establishes a tri-attention framework across review semantics, interaction patterns, and rating signals.

These methods implicitly learn interaction features from high-level review features, which lack supervision, and the final result can only learn part of the interaction features.

3 Methodology

The complex relationships between click and rating behaviors can be decomposed into three simple relationships: linear relationships, sharing relationships, and cross-correlation relationships. Thus, we propose a multi-factor collaborative prediction method (MFC) to mine the relationships, achieving accurate prediction on CTR tasks. Specifically, MFC first utilizes a factor learning network to extract click, rating, and sharing factors. The click and rating factors are then used to model the linear and cross-correlation relationships, while the sharing factor is used for the sharing relationship. Next, to ensure accurate learning of rating factors, we design a rating factor regularization method to guide the teaching of rating factors. Finally, we propose a factor collaborative prediction mechanism that combines click, rating, and sharing factors to model the three sub-relationships between click and rating behaviors. Figure 2 shows the framework of our method.

3.1 Factor learning network

Existing review-based methods [12, 40] develop a three-stage framework that sequentially learns user preference in reviews, clicks, and ratings. This framework models the effect of click behaviors on rating behaviors, achieving good performance in RP tasks. Based on this framework, our goal is to model the effect of rating behaviors on click behaviors. Thus, we employ an existing method, TSE [40], as the backbone of our factor learning network to extract the click factor $\mathbf{e}_c$ and rating factor $\mathbf{e}_r$ based on corresponding preference features.

Following the existing method [40], our factor learning network has three steps, and each step uses a statically shared embedding layer to learn the coarse-grained homophily in global behavior data, and a dynamically shared embedding layer to learn the fine-grained homophily in the process of user-item interaction. We first summarize the statically and dynamically shared embedding layers, that are widely used in learning three factors; then, we show how to extract review, click, and rating factors in sequence. The review factors are used to learn click and rating factors without directly participating in the prediction.

3.1.1 Statically shared embedding. Many methods introduce multiple embedding vectors to represent user preferences reflected in various types of behaviors [2, 11, 12]. However, the sparsity of recommendation data limits the learning effect of utilizing full embedding. TSE [40] proposes a statically shared embedding method, which generates multi-level static shared embeddings for users and items based on global behavioral data, and is a coarse-grained homogeneity modeling. Compared to full embedding, the shared embedding method reduces model complexity and improves model performance. We use $\mathbf{s}_u^e, \mathbf{s}_u^c, \mathbf{s}_u^a$ to denote review-level, click-level, and rating-level shared embedding vectors, respectively; and we use the $\mathbf{s}_u^e, \mathbf{s}_u^c$ in the learning of review and click factors, and $\mathbf{s}_u^a$ in the rating factor learning. The symbols of items are similar to those of users, i.e., $\mathbf{s}_i^e, \mathbf{s}_i^c, \mathbf{s}_i^a$.

3.1.2 Dynamically shared embedding. As the statically shared embedding is based on global behaviors, it models user homophily in a coarse-grained way, which is insufficient for precisely modeling users. Although multiple users are generally similar, they may differ greatly in some specific behaviors. Thus, TSE [40] proposes a dynamically shared embedding method that models homophily in a fine-grained way. Specifically, it considers the similarity between different user-item interactions in the prediction process and shares the dynamic interaction features between user-item pairs. Thus, we utilize three dynamically shared embedding layers to learn interaction features in review-, click- and rating levels. The output are denotes as $\mathbf{d}_u^e, \mathbf{d}_u^c, \mathbf{d}_u^a$ for review-, click- and rating-level, respectively. Likewise, the item's dynamically shared embedding are $\mathbf{d}_i^e, \mathbf{d}_i^c, \mathbf{d}_i^a$.

3.1.3 Review factor learning. This layer is designed to embed text, extract feature vectors from reviews, and aggregate these features to represent users and items. First, we embed the words in a review into a sequence of low-dimensional dense vectors. Let $d = \{w_x\}_{x=1}^p$ denote a review containing p words, and $\{\mathbf{w}_x\}_{x=1}^p$ be its corresponding word embedding sequence. Second, we employ CNNs

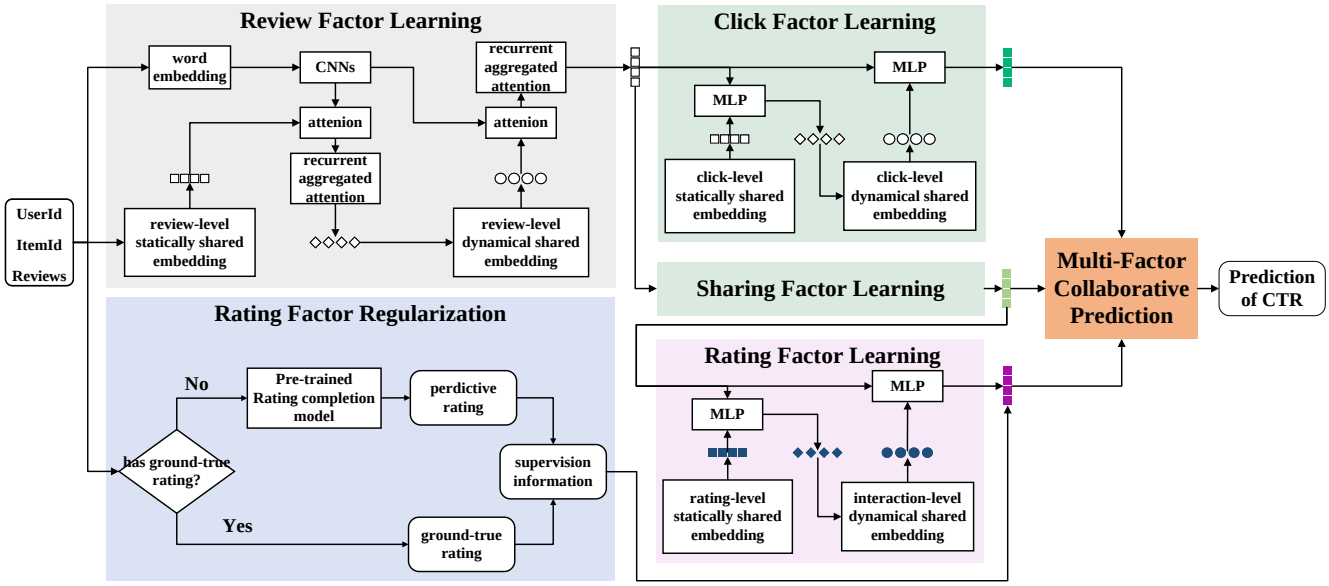


Figure 2: Framework of the proposed method, while the rating factor regularization only works in the training process.

to extract high-level word features, denoted as $\mathbf{c}_x$. Next, an attention mechanism is applied to summarize these word vectors with the statically shared embedding vectors $\mathbf{s}_u^e$ and $\mathbf{s}_i^e$ to generate a review representation. Then, we utilize a recurrent aggregated attention layer, introduced by PBFL [12], which aggregates multiple reviews to form comprehensive representations for users and items. Subsequently, a review-level dynamically shared embedding layer is employed to further encode the user and item representations, resulting in the dynamically shared embeddings $\mathbf{d}_u^e$ and $\mathbf{d}_i^e$. Afterward, $\mathbf{d}_u^e$, $\mathbf{d}_i^e$, and the CNN-extracted features are fed into the attention layer. Finally, the output is passed through the recurrent aggregated attention layer again to obtain the final review-based representations $\mathbf{e}_u^e$ and $\mathbf{e}_i^e$ for users and items.

$$\mathbf{e}_u^e = f_{re}(\mathcal{D}_u),\ \mathbf{e}_i^e = f_{re}(\mathcal{D}_i), \tag{1}$$

where f_{re} is the review factor learning netwok, $\mathcal{D}_u$ and $\mathcal{D}_i$ are the user u and item i's history reviews.

3.1.4 Click factor learning. Review features contain rich information on the interaction between users and items. Combining it with click-level preference embeddings will enhance the results of learning interaction features [12, 40]. This layer first uses MLP to aggregate review-based features $\mathbf{e}_u^e, \mathbf{e}_i^e$ and click-level statically shared embedding vectors $\mathbf{s}_u^c$ and $\mathbf{s}_i^c$,to get an interaction feature.

$$\begin{aligned}\mathbf{e}^{c'} &= MLP_1(\mathbf{e}_u^e, \mathbf{s}_u^c, \mathbf{e}_i^e, \mathbf{s}_i^c)\\ &= \phi_x^{elu}(...\phi_2^{elu}(\phi_1^{elu}([\mathbf{e}_u^e; \mathbf{s}_u^c : \mathbf{e}_i^e : \mathbf{s}_i^c]))...),\end{aligned} \tag{2}$$

where ϕ_x^{elu} denotes the mapping function of x-th neural network layer. Then, the interaction feature $\mathbf{e}^{c'}$ is used to select the interaction dynamically shared embedding $\mathbf{d}^c$. Finally, we use MLP to aggregate interaction features and $\mathbf{d}^c$ to get the final click factor.

$$\mathbf{e}^c = MLP_1(\mathbf{d}_u, \mathbf{d}_u^c, \mathbf{d}_i, \mathbf{d}_i^c) \tag{3}$$

3.1.5 Rating factor learning. Similar to the click factor learning layer, this layer utilizes click factor and user/item rating-level preference embeddings to learn the rating factor. Specifically, this layer first uses MLP to aggregate click factors and rating-level statically shared embedding vectors $\mathbf{s}_u^a$ and $\mathbf{s}_i^a$:

$$\begin{aligned}\mathbf{e}^{r'} &= MLP_1(\mathbf{e}^c, \mathbf{s}_u^a, \mathbf{s}_i^a)\\ &= \phi_x^{elu}(...\phi_2^{elu}(\phi_1^{elu}([\mathbf{e}^c : \mathbf{s}_u^a : \mathbf{s}_i^a]))...),\end{aligned} \tag{4}$$

where ϕ_x^{elu} denotes the mapping function of x-th neural network layer. The rating features $\mathbf{e}^{r'}$ are used to select the rating dynamically shared embedding $\mathbf{d}^a$. Finally, we use MLP to aggregate rating-based features and $\mathbf{d}^a$ to get the final rating factor.

$$\mathbf{e}^r = MLP_1(\mathbf{e}^c, \mathbf{d}_u^a, \mathbf{d}_i^a). \tag{5}$$

3.2 Rating Factor Regularization

It introduces supervised information on ratings to regularize the learning of rating factors only in the training process. Specifically, we input the rating factor into a single-layer MLP for rating prediction and then optimize the rating factor by constraining the difference between the predicted rating and the true rating through squared loss.

$$\mathcal{L}_r = \sum_{u,i \in \mathcal{D}} (t_{u,i}^r - MLP(\mathbf{e}_{u,i}^r))^2, \tag{6}$$

where $t_{u,i}^r$ is the rating of user u on item i.

However, in many real data, not all rating labels are always available. Recommendation models usually sample negative items

from unobserved data, which lack rating labels. Thus, we need to complete rating labels for negative samples.

Considering that the rating distribution has a significant center, in some individual rating prediction tasks, researchers try to assume the label of unobserved data as a fixed value and assign it a confidence level to make it participate in the process of the training model [6, 26, 33]. But this one-size-fits-all approach can only guarantee a lower bound on performance. The lack of independent settings will make the model's prediction more biased towards the central value and lack personalization. Here we pretrain a rating prediction model [40] and use it to complete rating labels in the training stage. The method can achieve better prediction performance than the method that regards a fixed value as the label. Thus, it gives a better lower bound of model performance than the fixed value method.

3.3 Multi-factor collaborative prediction

Users' click behavior and rating behavior have a complex relationship with each other. We use $\mathcal{F}$ to represent the relationship between rating and click behavior, as follows,

$$p^c = \mathcal{F}^r(p^r), \tag{7}$$

$$p^r = \mathcal{F}^c(p^c), \tag{8}$$

where $\mathcal{F}^c$ is the goal that we want to learn. As shown in Figure 1, it is a challenge that we cannot find simple functions to model the relationships. Fortunately, we find some potential relationships between ratings and clicks. On the one hand, regions B and D reflect a linear relationship that users prefer to click a high-rated item than a low-rated one [23]; on the other hand, region A indicates that users have strong demands and will click on an item even if its quality is poor. This situation reflects an inherent feature of click behaviors. Region C shows that some high-quality items have a small audience and are possibly extremely personalized. It shows a complex conditional probability between the two behaviors. In a word, there are three simple relations between click and rating behaviors: linear relation and high-level cross relation. Thus, our factor sharing prediction method is as:

$$p^c = \lambda_1 F^c(u, i) + \lambda_2 G^c(t^r) + \lambda_3 H^c(t^c | t^r) \tag{9}$$

where F^c is the original click behavior learning, G^c models the linear relation between rating and click behaviors, and H^c models the high-level cross relation.

Considering the true labels of t^r and t^c are unknown in the process of predictions, we use the input features instead of the true labels. For a recommendation model, the existing methods usually learn embedding vectors $\mathbf{x}_u$ and $\mathbf{x}_i$ for users and items, and then use a MLP with x hidden layers to make predictions. Given a user u and an item i, they make prediction following as

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

We denote the output of x-th layer as $\mathbf{e}$, which is the factor vector of the input. Thus, the prediction of our method is

$$p^c = \lambda_1^c F^c(\mathbf{e}^c) + \lambda_2^c G^c(\mathbf{e}^r) + \lambda_3^c H^c(\mathbf{e}^c \cdot \mathbf{e}^r) \tag{11}$$

where $\mathbf{e}^c$ and $\mathbf{e}^r$ are learned from Eq. (3) and Eq. (5), respectively. In fact, the click factor and rating factor could share some features from users' deep mind [14, 27]. Thus, we introduce a sharing factor $\mathbf{e}^s$ to model the common parts in the input factors of two tasks. Thus, our prediction method is re-written as:

$$p^c = \lambda_1^c F^c(\mathbf{e}^c, \mathbf{e}^s) + \lambda_2^c G^c(\mathbf{e}^r) + \lambda_3^c H^c(\mathbf{e}^c \cdot \mathbf{e}^r) \tag{12}$$

In practice, we use MLP to implement the functions F, G, and H, similar to Equation (10).

In a word, we factorize and model the complex relationships between click and rating behaviors into three simple relationships, i.e., linear, sharing, and cross-correlation relationships. By fusing these three relations, MFC introduces rating behaviors to enhance the model's performance on CTR tasks.

3.4 Model training

Following existing methods, we use Binary Cross-Entropy (BCE) as the loss function. The objective function of our method is defined as

$$\begin{aligned} \min \mathcal{L} = \frac{1}{|N|} \sum_{(u,i,t_{ui}^r,t_{ui}^c)\in\mathcal{D}} & l_{BCE}(t_{ui}^c, p_{ui}^c) \\ & + \lambda_r \mathcal{L}_r + \lambda_F ||\Theta||_F \end{aligned} \tag{13}$$

where $N = |\mathcal{D}|$ is the size of train data, Θ is the model parameters, λ_r is used to control the impact of rating factor regularization, $||\cdot||_F$ is the Frobenius norm, and λ_F is the regularized coefficient for the Frobenius norm. We use an Adam optimizer to optima our objective function.

4 Experimental Results

This section begins by introducing the datasets, baselines, and experimental setup. Next, we compare the proposed method against all baselines and present the results. Finally, we conduct an ablation study and a parameter sensitivity analysis to investigate the impact of different model components.

4.1 Dataset

Experiments are conducted on five publicly available Amazon 5-core datasets[1] to validate the effectiveness of the proposed model: Music Instruments (MI), Office Products (OP), Grocery and Gourmet Food (GGF), Video Games (VG), and Sports and Outdoors (SO). These standardized datasets are extensively adopted for review-driven recommendation validation [12, 37, 40]. The lexical processing pipeline involves: (1) text normalization through case folding and non-linguistic element removal via regex patterns; (2) stop word elimination using NLTK toolkit [35, 37]. Our review encoder operates on a fixed vocabulary of 20,000 high-frequency tokens derived from corpus analysis. Following established practices [2], variable-length reviews are standardized through zero-padding. Complete dataset characteristics are summarized in Table 1.

4.2 Baseline

To fully evaluate the proposed method, we compare it with two types of methods: enhancing review representation and modeling interaction patterns. Specifically, we select the following methods as baselines. The first type of method includes:

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

Table 1: Statistical results of five datasets, where *#Users*, *#Items* denotes the number of users and items, *#Ra/Re* indicate the number of ratings and reviews, and $\bar{l}$ denotes the average number of words per review.

Dataset	#Users	#Items	#Ra/Re	$\bar{l}$	Density
MI	1,429	900	10,261	59	0.798%
OP	4905	2420	53,228	56	0.449%
GGF	14,681	8,713	151,254	89	0.118%
VG	24,303	10,672	231,577	124	0.089%
SO	35,598	18,357	296,337	99	0.045%

- ERP [38] considers the collaborative signal and similarity between user historical items and the target item, to differentiate the contribution of different reviews and model user preference for the target item.
- TAERT [4] implements triple-branch attention architecture for lexical saliency weighting, review quality assessment, and rating preference alignment.
- RGNN [16] constructs heterogeneous review graphs with lexical nodes and syntactic edges encoding sequential dependencies through graph attention networks, introducing GNN into the review-based recommendation.

The second type of method includes:

- DAML [11] develops cross-attentive CNN modules that synchronously learn local review semantics and global interaction contexts through multi-head attention.
- NARRE [2] filters the useless reviews by using users' and items' embedding vectors as a part of attention scores.
- HTI [35] captures interactions based on reviews by mutually propagating textual features.
- DSRNL [15] extracts dynamic and static user preferences via dual attention layers on interaction and review data, respectively, and fuses them adaptively. Item representations are obtained via element-wise product of temporal and inherent attributes.
- TSE [40] proposes a two-tier shared embedding framework that models multi-scale homophily. It captures macro homophily by clustering global behaviors and micro homophily by focusing on each user-item interaction.
- PBFL [12] designs tripartite preference distillation framework integrating review semantics, interaction possibility, and rating preference.

4.3 Setup

The review factor learning layer first employs GloVe-100 pre-trained embeddings for lexical initialization [2, 40]. It then cascades two convolutional layers with filter sizes 3,5 followed by max-pooling operations [12]. Factor learning layers utilize MLP (300-200-100 dimensionality progression) to fuse four modality-specific embeddings through concatenation and nonlinear transformations [2, 30].

Each experiment is repeated five times, and the reported results represent the average of these runs. For each training process, each dataset was uniformly randomly shuffled and split into three parts: a training set (80%), a validation set (10%), and a test set(10%). Only reviews in the training set were used for model training. The validation set is used to tune hyperparameters. For the rating prediction task, the model's performance on the test set was evaluated using the mean absolute error (MAE). For the CTR task, we assessed performance using Accuracy (ACC), Precision (P), Recall (R), and F1-score (F1). For the CTR prediction, recommender systems typically generate negative samples from unobserved data, while observed data are treated as positive samples. In this work, we randomly select negative samples at a predefined ratio λ_{ng}, i.e., for each positive sample, we sample λ_{ng} negative samples. Here, we set λ_{ng} as 1 to balance the positive and negative samples.

4.4 Comparison results

Table 2 shows performance results for all baselines. Compared to the baseline models, which represent the state-of-the-art for single-task performance, our model achieves significant improvements in both the RP and CTR tasks. It identifies that our method successfully captures the complex relationships between click and rating behaviors. Specifically, MFC outperform the best baseline by 9.19%, 9.80%, 0.69%, 7.95%, 1.92% on ACC, P, R, F1, and MAE. Through the multi-factor sharing method to model the three relationships, the review-based recommendation achieves improved performance compared to previous methods.

4.5 Ablation study

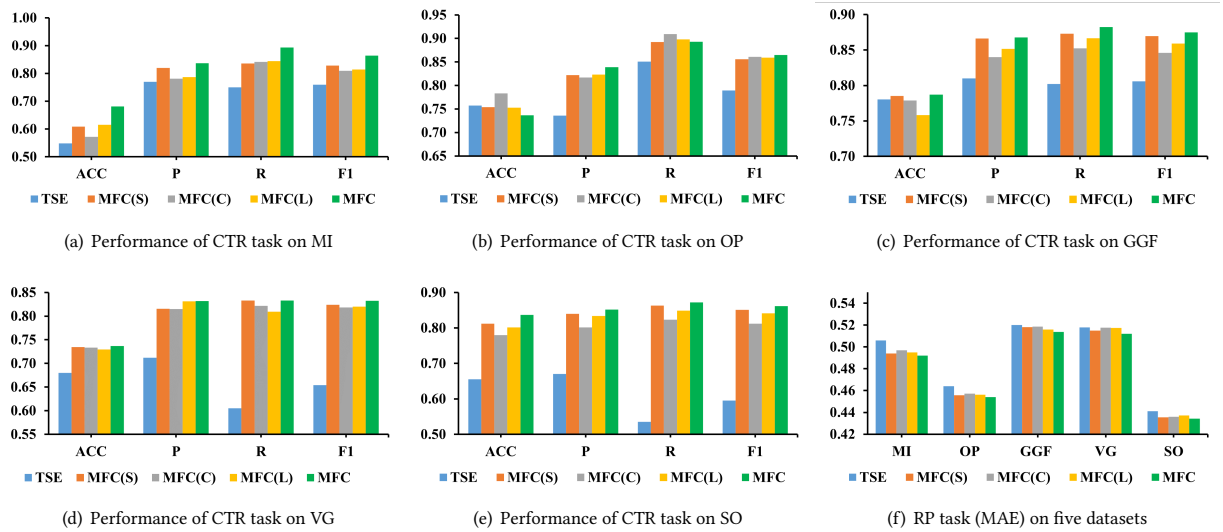
Eq. (12) captures three types of relationships between the two behaviors: sharing, linear, and cross-correlation relationships. To evaluate the impact of each relationship, we conducted an ablation study. We created three model variants that utilize only one relationship and remove the others, denoted as MFC(S), MFC(L), and MFC(C), respectively. Figure 3 presents the model's performance on the two tasks. Compared to TSE, which is the best baseline and also the backbone of our method, our variants demonstrate significant improvements in the CTR prediction task. The three relationship types effectively capture the relationships between the click and rating behaviors. Among the variants, MFC(S) performs best in most cases, indicating a strong connection between two behaviors and the effectiveness of the sharing factor in modeling this connection. Meanwhile, the other two relationships provide complementary benefits, enabling MFC to outperform all variants in most scenarios.

4.6 The impact of rating factor regularization

This experiment aims to evaluate the impact of the proposed method on rating label completion. We compare MFC with a variant, MFC(F), which assigns a fixed value (e.g., 4) to negative samples. The results are presented in Table 3. Compared to TSE, MFC(F) shows significant improvement in the CTR task but experiences a performance drop in the RP task. This one-size-fits-all approach only ensures a lower performance bound, as the lack of task-specific settings causes the model's predictions to become biased toward central values, reducing personalization. More importantly, it may mislead the model into treating the fixed value always as a negative sample. In contrast, the rating labels generated using the proposed completion method help MFC achieve better performance on both tasks. Additionally, we introduce MFC(P), a pretraining-based variant

Table 2: The performances of different recommendation methods. The best results are in bold faces and the second-best results are underlined.

		ERP	TAERT	RGNN	CARL	DAML	NARRE	HTI	DSRLN	PBFL	TSE	MFC(mean±std)
MI	ACC	0.5972	0.6045	0.6159	0.4719	0.5452	0.5694	0.5960	0.3364	0.5735	0.5480	**0.6770**±0.02
	P	0.6646	0.5651	0.6764	0.4319	0.5222	0.5268	0.5690	0.6439	0.5511	0.7700	**0.8400**±0.009
	R	0.7094	0.4239	0.7780	0.4295	0.8138	0.8058	0.7980	0.8591	0.7104	0.7500	**0.9015**±0.009
	F1	0.6863	0.4844	0.7237	0.4307	0.6362	0.6371	0.6643	0.7361	0.6207	0.7599	**0.8697**±0.009
	MAE	0.6430	0.6010	0.6050	0.6770	0.6510	0.6820	0.6110	0.6200	0.5950	0.5060	**0.4920**±0.005
OP	ACC	0.5388	0.6915	0.5871	0.4779	0.5676	0.5757	0.6852	0.5801	0.6165	**0.7570**	0.7364±0.02
	P	0.6629	0.7050	0.6043	0.4271	0.5459	0.6639	0.6474	0.6871	0.6469	0.7360	**0.8359**±0.02
	R	0.7871	0.5920	0.7715	0.4325	0.8417	0.8216	0.8006	0.8330	0.7271	0.8510	**0.8991**±0.01
	F1	0.7197	0.6436	0.6777	0.4298	0.6623	0.7344	0.7159	0.7530	0.6847	0.7893	**0.8663**±0.01
	MAE	0.7250	0.6040	0.6230	0.6470	0.6120	0.6800	0.5520	0.6210	0.5250	0.4640	**0.4517**±0.003
GGF	ACC	0.6014	0.7014	0.6037	0.4240	0.5702	0.5627	0.6811	0.5319	0.6823	0.7802	**0.7840**±0.009
	P	0.6791	0.6919	0.6274	0.3940	0.5011	0.5548	0.6799	0.7175	0.7811	0.8100	**0.8570**±0.005
	R	0.8136	0.5157	0.7569	0.4467	0.7467	0.5710	0.7266	**0.8994**	0.5479	0.8020	0.8350±0.007
	F1	0.7403	0.5909	0.6861	0.4187	0.5997	0.5628	0.7025	0.7982	0.6440	0.8060	**0.8459**±0.007
	MAE	0.8350	0.6920	0.7100	0.7530	0.7350	0.7470	0.6720	0.7210	0.6310	0.5200	**0.5120**±0.004
VG	ACC	0.5331	0.7019	0.5976	0.3780	0.5615	0.4997	0.6904	0.5091	0.7156	0.6800	**0.7497**±0.007
	P	0.6611	0.7354	0.5978	0.3780	0.5516	0.4842	0.7013	0.7671	0.7452	0.7120	**0.8322**±0.01
	R	0.7098	0.5726	0.7195	0.5000	0.7597	0.5241	0.6962	0.8314	0.7921	0.6050	**0.8818**±0.04
	F1	0.6846	0.6439	0.6530	0.4305	0.6391	0.5034	0.6987	0.7980	0.7679	0.6542	**0.8563**±0.02
	MAE	0.9380	0.6870	0.7660	0.7980	0.7880	0.7990	0.7310	0.7540	0.6460	0.5180	**0.5164**±0.005
SO	ACC	0.4859	0.6217	0.6034	0.4817	0.5223	0.4749	0.5562	0.4823	0.6215	0.6550	**0.8139**±0.02
	P	0.6617	0.6493	0.6265	0.4266	0.5350	0.4628	0.5379	0.7334	0.6058	0.6700	**0.8219**±0.03
	R	0.7130	0.6941	0.7211	0.4547	0.8217	0.8171	0.8024	**0.9267**	0.7647	0.5350	0.8707±0.05
	F1	0.6864	0.6710	0.6705	0.4402	0.6481	0.5909	0.6441	0.8188	0.6760	0.5949	**0.8456**±0.04
	MAE	0.6800	0.5940	0.6460	0.6860	0.6670	0.6900	0.6290	0.6510	0.5630	0.4410	**0.4306**±0.003

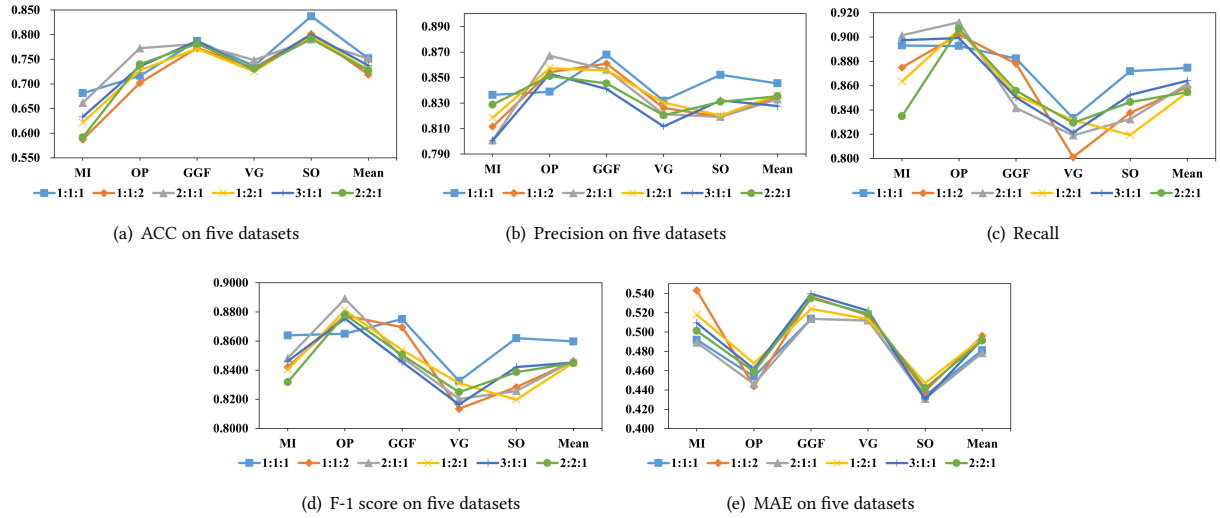


(a) Performance of CTR task on MI (b) Performance of CTR task on OP (c) Performance of CTR task on GGF

(d) Performance of CTR task on VG (e) Performance of CTR task on SO (f) RP task (MAE) on five datasets

Figure 3: Performance of various variants

of MFC. It initializes the parameters of the rating-factor learning network with the rating completion model. However, this approach



(a) ACC on five datasets (b) Precision on five datasets (c) Recall

(d) F-1 score on five datasets (e) MAE on five datasets

Figure 4: Preference of different ratios of the hyperparameters $\lambda_1 : \lambda_2 : \lambda_3$

Table 3: Performance of variants.

		TSE	MFC(F)	MFC(P)	MFC
MI	ACC	0.5480	0.6260	0.6173	0.6770
	P	0.7700	0.8360	0.8319	0.8400
	R	0.7500	0.8290	0.8850	0.9015
	F1	0.7599	0.8325	0.8576	0.8697
	MAE	0.5060	0.5530	0.5070	0.4920
OP	ACC	0.7570	0.7000	0.7338	0.6979
	P	0.7360	0.8750	0.8693	0.8359
	R	0.8510	0.9130	0.9077	0.8991
	F1	0.7893	0.8936	0.8881	0.8663
	MAE	0.4640	0.4980	0.4680	0.4517
GGF	ACC	0.7802	0.8050	0.7991	0.7840
	P	0.8100	0.8841	0.8411	0.8570
	R	0.8020	0.8278	0.8489	0.8350
	F1	0.8060	0.8550	0.8450	0.8459
	MAE	0.5200	0.5491	0.5190	0.5120
VG	ACC	0.6800	0.7550	0.7397	0.7497
	P	0.7120	0.8020	0.8439	0.8322
	R	0.6050	0.7520	0.8907	0.8818
	F1	0.6542	0.7762	0.8667	0.8563
	MAE	0.5180	0.5812	0.5155	0.5164
SO	ACC	0.6550	0.8220	0.8115	0.8139
	P	0.6700	0.8200	0.8229	0.8219
	R	0.5350	0.7990	0.8657	0.8707
	F1	0.5949	0.8094	0.8438	0.8456
	MAE	0.4410	0.5660	0.4547	0.4306

tends to overfit the RP task because the expert network has already undergone training once, limiting its generalization ability.

4.7 Hyperparameter sensitivity study

To further investigate the role of each factor, we evaluate the model's performance using different hyperparameter settings. We select six distinct configurations for these hyperparameters. The ratio settings of the three factors are $\{(1:1:1), (1:1:2), (1:1:2), (2:1:1), (3:1:1), (2:2:1)\}$. Figure 4 presents the model preference on five datasets and the average preferences. We can find that the model performance with the ratio $(1:1:1)$ outperforms other settings on the CTR task and that with the ratio $(2:1:1)$ is the best setting on the rating prediction task. Considering the results of the sharing factor shown in the ablation study, we further identify that the sharing factor has an important effect on the rating prediction task. It shows that rating features learned from user reviews can improve model performance on CTR tasks. The difference in the optimal settings between the two tasks also shows that there is a complex relationship between the two tasks.

5 Conclusion

Recommender systems have become widely used in everyday life, aiming to satisfy user requirements and interests. User reviews, as a significant form of feedback on item features, can help these systems better understand user preferences and item attributes. Existing methods implicitly model click behaviors to achieve accurate predictions on rating prediction tasks (RP), but ignore the help of the rating behaviors for the click-through rate prediction task (CTR). This paper proposes a multi-factor collaborative prediction method (MFC) for review-based recommendations, which mines the complex relationship between click and rating behaviors, achieving accurate prediction on CTR tasks. Specifically, we first factorize the complex relationships between the two behaviors into three simple relationships, i.e., linear, sharing, and cross-correlation relationships. Then, we use three individual but with similar structure factor-learning networks to extract click, rating, and sharing

factors from user reviews. At the same time, to ensure rating factors accurately reflect rating behaviors, we propose a rating factor regularization method, which injects supervision information of ratings into the rating factors. Finally, we combine these three factors to make predictions, while click and rating factors are used to model the linear and cross-correlation relationships, and the sharing factors correspond to the sharing relation. Experiments on five real-world datasets demonstrate that the performance of MFC significantly outperforms baselines in CTR and PR tasks.

Acknowledgments

This work is partially supported by the National Key R&D Program of China (No.2022YFB3103100), and the Beijing University of Technology Education and Teaching Research Project (No.ER2024KCA01).

References

[1] James Chambua and Zhendong Niu. 2021. Review text based rating prediction approaches: preference knowledge learning, representation and utilization. *Artif. Intell. Rev.* 54, 2 (2021), 1171–1200.

[2] Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2018. Neural attentional rating regression with review-level explanations. In *Proceedings of the 2018 WWW*. 1583–1592.

[3] Xin Dong, Jingchao Ni, Wei Cheng, Zhengzhang Chen, Bo Zong, Dongjin Song, Yanchi Liu, Haifeng Chen, and Gerard de Melo. 2020. Asymmetrical Hierarchical Networks with Attentive Interactions for Interpretable Review-Based Recommendation. In *Proceedings of the 34th AAAI*. AAAI Press, 7667–7674.

[4] Siyuan Guo, Ying Wang, Hao Yuan, Zeyu Huang, Jianwei Chen, and Xin Wang. 2021. TAERT: Triple-Attentional Explainable Recommendation with Temporal Convolutional Network. *Inf. Sci.* 567 (2021), 185–200.

[5] Siyuan Guo, Lixin Zou, Yiding Liu, Wenwen Ye, Suqi Cheng, Shuaiqiang Wang, Hechang Chen, Dawei Yin, and Yi Chang. 2021. Enhanced Doubly Robust Learning for Debiasing Post-Click Conversion Rate Estimation. In *Proceedings of the 44th SIGIR*. ACM, 275–284.

[6] José Miguel Hernández-Lobato, Neil Houlsby, and Zoubin Ghahramani. 2014. Probabilistic Matrix Factorization with Non-random Missing Data. In *Proceedings of the 31th ICML*, Vol. 32. JMLR.org, 1512–1520.

[7] Donghyun Kim, Chanyoung Park, Jinoh Oh, Sungyoung Lee, and Hwanjo Yu. 2016. Convolutional matrix factorization for document context-aware recommendation. In *Proceedings of the 10th RecSys*. 233–240.

[8] Yehuda Koren. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD*. ACM, 426–434.

[9] Yehuda Koren, Robert M. Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* 42, 8 (2009), 30–37.

[10] Duantengchuan Li, Ceyu Deng, Xiaoguang Wang, Zhifei Li, Chao Zheng, Jing Wang, and Bing Li. 2024. Joint inter-word and inter-sentence multi-relation modeling for summary-based recommender system. *Inf. Process. Manag.* 61, 2 (2024), 103631.

[11] Donghua Liu, Jing Li, Bo Du, Jun Chang, and Rong Gao. 2019. Daml: Dual attention mutual learning between ratings and reviews for item recommendation. In *Proceedings of the 25th ACM SIGKDD*. 344–352.

[12] Junrui Liu, Tong Li, Zhen Yang, Di Wu, and Huan Liu. 2024. Fusion learning of preference and bias from ratings and reviews for item recommendation. *Data & Knowledge Engineering* 150 (2024), 102283.

[13] Junrui Liu, Zhen Yang, Tong Li, Di Wu, and Ruiyi Wang. 2022. SPR: Similarity pairwise ranking for personalized recommendation. *Knowl. Based Syst.* 239 (2022), 107828.

[14] Shikun Liu, Edward Johns, and Andrew J. Davison. 2019. End-To-End Multi-Task Learning With Attention. In *Proceedings of the CVPR 2019*. Computer Vision Foundation / IEEE, 1871–1880.

[15] Tongcun Liu, Siyuan Lou, Jianxin Liao, and Hailin Feng. 2024. Dynamic and Static Representation Learning Network for Recommendation. *IEEE Trans. Neural Networks Learn. Syst.* 35, 1 (2024), 831–841.

[16] Yong Liu, Susen Yang, Yinan Zhang, Chunyan Miao, Zaiqing Nie, and Juyong Zhang. 2023. Learning Hierarchical Review Graph Representations for Recommendation. *IEEE Trans. Knowl. Data Eng.* 35, 1 (2023), 658–671.

[17] Andriy Mnih and Russ R Salakhutdinov. 2007. Probabilistic matrix factorization. In *Proceedings of the NIPS 2007*, Vol. 20. 1257–1264.

[18] Wentao Ning, Reynold Cheng, Xiao Yan, Ben Kao, Nan Huo, Nur Al Hasan Haldar, and Bo Tang. 2024. Debiasing Recommendation with Personal Popularity. In *Proceedings of the 33rd WWW, Singapore, May 13-17, 2024*. ACM, 3400–3409.

[19] Weike Pan and Li Chen. 2013. Gbpr: Group preference based bayesian personalized ranking for one-class collaborative filtering. In *Procedding of the 23rd IJCAI, Beijing, China, August 3-9, 2013*.

[20] Weike Pan, Li Chen, and Zhong Ming. 2019. Personalized recommendation with implicit feedback via learning pairwise preferences over item-sets. *Knowledge and Information Systems* 58, 2 (2019), 295–318.

[21] Steffen Rendle. 2010. Factorization Machines. In *Proceedings of the 10th ICDM, Sydney, Australia, 14-17 December 2010*. IEEE Computer Society, 995–1000.

[22] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the 25th UAI, Montreal, QC, Canada, June 18-21, 2009*. AUAI Press, 452–461.

[23] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. 2016. Recommendations as Treatments: Debiasing Learning and Evaluation. In *Proceedings of the 33nd ICML*, Vol. 48. JMLR.org, 1670–1679.

[24] 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 *Proceedings of the eleventh ACM RecSys*. 297–305.

[25] Jie Shuai, Kun Zhang, Le Wu, Peijie Sun, Richang Hong, Meng Wang, and Yong Li. 2022. A Review-aware Graph Contrastive Learning Framework for Recommendation. In *Proceedings of the 45th ACM SIGIR*. ACM, 1283–1293.

[26] Harald Steck. 2010. Training and testing of recommender systems on data missing not at random. In *Proceedings of the 16th ACM SIGKDD*. ACM, 713–722.

[27] Hongyan Tang, Junning Liu, Ming Zhao, and Xudong Gong. 2020. Progressive Layered Extraction (PLE): A Novel Multi-Task Learning (MTL) Model for Personalized Recommendations. In *Proceedings of the 14th ACM RecSys*. 10.

[28] Ho Thi Hoang Vy, Cuong Pham-Nguyen, and Le Nguyen Hoai Nam. 2024. Integrating textual reviews into neighbor-based recommender systems. *Expert Syst. Appl.* 249 (2024), 123648. doi:10.1016/J.ESWA.2024.123648

[29] Qi Wan, Xiangnan He, Xiang Wang, Jiancan Wu, Wei Guo, and Ruiming Tang. 2022. Cross Pairwise Ranking for Unbiased Item Recommendation. In *Proceedings of the 31st WWW*. ACM, 2370–2378.

[30] Chong Wang and David M. Blei. 2011. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD*. ACM, 448–456.

[31] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. 2015. Collaborative deep learning for recommender systems. In *Proceedings of the 21th ACM SIGKDD*. 1235–1244.

[32] Ke Wang, Yanmin Zhu, Tianzi Zang, Chunyang Wang, and Mengyuan Jing. 2024. Review-Enhanced Hierarchical Contrastive Learning for Recommendation. In *Proceedings of the 38th AAAI*. AAAI Press, 9107–9115.

[33] Xiaojie Wang, Rui Zhang, Yu Sun, and Jianzhong Qi. 2019. Doubly robust joint learning for recommendation on data missing not at random. In *Proceedings of the 36th ICML,Long Beach, California, USA, 9-15 June 2019*. PMLR, 6638–6647.

[34] J. Wen, J. Ma, H. Tu, M. Zhong, G. Zhang, W. Yin, and J. Fang. 2020. Hierarchical text interaction for rating prediction. *Knowledge-Based Systems* 206 (2020), 106344.

[35] Jiahui Wen, Jingwei Ma, Hongkui Tu, Mingyang Zhong, Guangda Zhang, Wei Yin, and Jian Fang. 2020. Hierarchical text interaction for rating prediction. *Knowledge-Based Systems* 206 (2020), 106344.

[36] Huiqiong Wu, Guibing Guo, Enneng Yang, Yudong Luo, Yabo Chu, Linying Jiang, and Xingwei Wang. 2024. PESI: Personalized Explanation recommendation with Sentiment Inconsistency between ratings and reviews. *Knowl. Based Syst.* 283 (2024), 111133. doi:10.1016/J.KNOSYS.2023.111133

[37] Libing Wu, Cong Quan, Chenliang Li, Qian Wang, Bolong Zheng, and Xiangyang Luo. 2019. A context-aware user-item representation learning for item recommendation. *ACM TOIS* 37, 2 (2019), 1–29.

[38] Shiwen Wu, Yuanxing Zhang, Wentao Zhang, Kaigui Bian, and Bin Cui. 2021. Enhanced review-based rating prediction by exploiting aside information and user influence. *Knowledge-Based Systems* (2021), 107015.

[39] Yuanbo Xu, Yongjian Yang, Jiayu Han, En Wang, Fuzhen Zhuang, Jingyuan Yang, and Hui Xiong. 2019. NeuO: Exploiting the sentimental bias between ratings and reviews with neural networks. *Neural Networks* 111 (2019), 77–88.

[40] Zhen Yang, Junrui Liu, Tong Li, Di Wu, Shiqiu Yang, and Huan Liu. 2023. A Two-tier Shared Embedding Method for Review-based Recommender Systems. In *Proceedings of the 32nd CIKM*. ACM, 2928–2938.

[41] Chenyan Zhang, Shan Xue, Jing Li, Jia Wu, Bo Du, Donghua Liu, and Jun Chang. 2023. Multi-Aspect enhanced Graph Neural Networks for recommendation. *Neural Networks* 157 (2023), 90–102.

[42] Lei Zheng, Vahid Noroozi, and Philip S Yu. 2017. Joint deep modeling of users and items using reviews for recommendation. In *Proceedings of the 10th ACM WSDM*. 425–434.