Represent Items by Items: An Enhanced Representation of the Target Item for Recommendation

Yinjiang Cai[1,2], Zeyu Cui[2,3], Shu Wu[2,3], Zhen Lei[1,2], Xibo Ma[1,2,*]

[1]CBSR&NLPR, Institute of Automation, Chinese Academy of Sciences, Beijing, China

[2]School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China

[3]Center for Research on Intelligent Perception and Computing, Institute of Automation, Chinese Academy of Sciences

{yinjiang.cai,zeyu.cui,shu.wu,zlei,xibo.ma}@nlpr.ia.ac.cn

ABSTRACT

Item-based collaborative filtering (ICF) has been widely used in industrial applications such as recommender system and online advertising. It models users' preference on target items by the items they have interacted with. Recent models use methods such as attention mechanism and deep neural network to learn the user representation and scoring function more accurately. However, despite their effectiveness, such models still overlook a problem that performance of ICF methods heavily depends on the quality of item representation especially the target item representation. In fact, due to the long-tail distribution in the recommendation, most item embeddings can not represent the semantics of items accurately and thus degrade the performance of current ICF methods. In this paper, we propose an enhanced representation of the target item which distills relevant information from the co-occurrence items. We design sampling strategies to sample fix number of co-occurrence items for the sake of noise reduction and computational cost. Considering the different importance of sampled items to the target item, we apply attention mechanism to selectively adopt the semantic information of the sampled items. Our proposed Co-occurrence based Enhanced Representation model (CER) learns the scoring function by a deep neural network with the attentive user representation and fusion of raw representation and enhanced representation of target item as input. With the enhanced representation, CER has stronger representation power for the tail items compared to the state-of-the-art ICF methods. Extensive experiments on two public benchmarks demonstrate the effectiveness of CER.

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

KEYWORDS

collaborative filtering, item-based CF, attention networks, neural recommender models

*To whom correspondence should be addressed.

1 INTRODUCTION

Personalized recommendation have been widely used in many online services such as E-commerce, advertising and social media sites. The core of a personalized recommender system is modeling the preference of users on items based on their behaviors. Various methods have been applied to boost the recommendation performance, among these methods, collaborative filtering (CF) is a milestone one.

A series of CF based methods focused on how to represent the user more accurately. As shown in figure 1, Matrix Factorization (MF) [15, 22, 23] represents users by the embedding of user ID, which is a user-based CF (UCF) method. In MF, interest between user and item is calculated by using the inner product of the user embedding and item embedding. Previous works [13, 17, 27] have suggested that item-based CF (ICF) methods which directly represent the users by their historically interacted items have advantages in accuracy, real-time personalization and interpretability [27]. Among these methods, FISM [17] predicts the interest by calculating the inner product between historical items and target item instead of between user embedding and item embedding in UCF. NAIS [13] further extends FISM by dynamically adjusting the importance of historical items according to different target items.

Although these ICF methods demonstrated their performance in the recommendation, we argue that they neglected a significant issue, the quality of item embeddings. From figure 1, for ICF methods, it is not difficult to find that if the embedding of the target item is not accurate, the final prediction will be deviated greatly. Then we will ask, is the quality of the current item embeddings good enough in fact? As we all know that the quality of one embedding depends on the training method and the frequency of the item. As the higher frequency of the item, the more fully the model learns it. Unfortunately, long-tail distributions widely exist in many real-world datasets [32]. As shown in the left part of figure 2, the long-tail distribution in the recommender system determines that a large number of items can not be trained well as they have little user feedback. Thus these tail items will finally degrade the performance of the model. To sum up, we find that the performance of the ICF methods depends heavily on the quality of the item representations, but current models can not represent most items well due to the
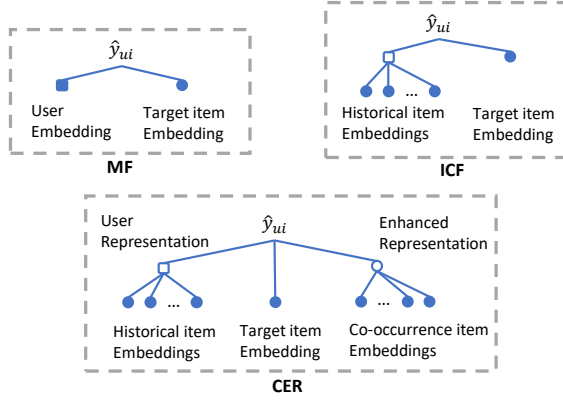
Figure 1: Illustration of different CF methods. Solid square and circle represent user and item embedding. The hollow node denotes a representation that encodes other solid nodes. From MF to our CER, ICF encodes historical items to enhance the user representation, CER encodes co-occurrence items to enhance the target item representation.

Figure 2: Illustration of long-tail distribution and enhanced representation. α represents the relevance between movie 'Agent Carter' and other movies. For the tail item 'Agent Carter', its enhanced representation are learned by the co-occurrence items such as 'Avengers 4' and 'Iron Man' basing on their relevance α_1 and α_4. Thus Marvel movie 'Agent Cater' is more likely to be recommended to fans of Marvel movies.

long-tail distribution. The experiment results in figure 5 show that the accuracy of the current ICF method for low-frequency items is much lower than that for high-frequency items, which verifies the problem mentioned above.

To alleviate the aforementioned issue of ICF methods, here we propose an enhanced representation to complement the semantics of the target items, which is shown in figure 1. We all know that items which user have interacted with are out of the same user interest. Thus these items are highly complementary in semantics. Here we proposed to learn the enhanced representation of the target item based on the co-occurrence items of it. Among these co-occurrence items, head items that have high quality of embeddings can provide accurate semantics for the target item. Meanwhile, a large number of tail items among them can still provide relevant information for the target item by adopting attention mechanism. Specifically, for each target item, we search for all items which have co-appeared with the target item and construct a co-occurrence item set. For the sake of noise reduction and computational complexity, we design sampling strategies to sample fix number of items from the co-occurrence item set. In these sampling strategies, two criteria are considered. One is to sample items with high semantic similarity to the target item. Another is to sample items whose embeddings have low semantic uncertainty. Considering the different semantic relevance between co-occurrence items and target item, co-occurrence items should also contribute differently to the enhanced representation of the target item. To distinguish the different importance of co-occurrence items, here we employ attention mechanism [19, 26] which has been widely used in neural representation learning. Figure 2 illustrates both the user representation and enhanced representation. With enhanced representation, our proposed model CER can also work well for large among of tail items. Extensive experiments on public datasets show that our method betters FISM for a 17 percent relative improvement in terms

of NDCG@10 and 26 percent relative improvement in HR@10 on the Amazon-Books dataset.

The main contributions of this paper are summarized as follows:

- We highlight that the long-tail distribution limits the current expression of items and thus the poor quality of target item embedding will degrade the performance of existing ICF methods.
- We propose an enhanced representation of the target item, a novel expression combining the idea of item-based representation with collaborative signals, which selectively learns representation from the embeddings of co-occurrence items.
- We conduct experiments on two public datasets. Extensive results show that our model significantly outperforms the state-of-the-art methods and demonstrate its effectiveness in improving the quality of target item representation.

2 RELATED WORK

Following figure 1, we can divide each CF method into three parts, namely modeling user, modeling target item and modeling interaction. In this section, we briefly review several lines of literature closely related to these three parts.

For modeling user, early UCF methods [14, 18] represent the user by the user ID which limits the expression of the interests of the user. Recent ICF methods aim to encode more signals in the user representation rather than only user ID in UCF. ItemKNN [24] utilizes statistical measures such as cosine similarity to estimate the similarities between items and sum the similarities between historical items and target item as the interest between user and target item. Based on the idea of ItemKNN, SLIM [21] propose to use matrix to fit the similarities between items. FISM [17] calculates the similarity between two items by the inner product of their latent vectors. NAIS [13] argues that historical items have different importance for the target item. Thus it proposed a more flexible representation of the user which uses attention mechanism to dynamically adjust the weight of each historical item.

For modeling interaction, early methods directly use the inner product between user latent vector and item latent vector as the interaction function. Recent literature [4, 11, 31] show that deep learning has the potential to dramatically boost the performance of recommender systems. NCF [14] proposed to learn the complex interaction function by a standard structure of MLP. DeepICF [27] argues that the strong function learning ability of MLP is expected to capture the higher-order item relations.

For modeling target item, except for the most common use of the item ID, DMF [28] represents an item by a multi-hot vector representing the users who have interacted with the item. DELF [6] employs attention mechanism to discriminate the importance of the interacted users automatically. Another kind of approaches to encode items by the collaborative signal are graph-based methods [12, 25]. Graph-based methods construct an graph to represent the interactions between users and items and each vertex in this graph can be encoded by its neighbors. In nature, they still follow the idea that representing items by users, representing users by items. One of the advantages of graph-based methods is that they make it easy to capture the high-order connectivity. This is how it differs from the approaches that are neighbor-based [1, 29] but not graph-based. In addition to the collaborative signal, some methods model items with additional side information such as visual patterns [5, 9] and knowledge [30].

3 PRELIMINARIES

Before we introduce our model, we firstly recapitulate the framework for standard item-based collaborative filtering. We then brief the FISM and NAIS methods as they form the basis of our method.

3.1 Standard Item-Based CF

The core of item-based CF is that user is no longer represented by the unique user embedding, but by the items that the user has interacted with. So the idea of item-based CF is to predict the probability that a user u is interested in a target item i based on the user behaviors. Formally, the prediction of ICF model can be abstracted as

$$\hat{y}_{ui} = \sum_{j \in R_u} s_{ij} r_{uj}, \tag{1}$$

where R_u denotes the set of items that the user u has interacted with, r_{uj} denotes the preference of user u on item j, which can either be a binary value (implicit feedback) or a real-valued rating score (explicit feedback), s_{ij} denotes the similarity between item i and j.

3.2 FISM and NAIS Methods

Recent methods like FISM [17] apply inner product between latent vectors of items to express the similarity mentioned above. The prediction of FISM can be formulated as

$$\hat{y}_{ui} = \frac{1}{|R_u|^\alpha} \sum_{j \in R_u} r_{uj} \mathbf{q}_i^\top \mathbf{q}_j \tag{2}$$

where $\mathbf{q_i} \in \mathbb{R}^k$ and $\mathbf{q}_j \in \mathbb{R}^k$ denote the latent vectors for the target item i and historical item j, all latent vectors are trainable parameters with embedding size k. Hyper-parameter α controls the normalization on users of different behavior length, $\alpha = 0$ means

the prediction is the sum over item set, $\alpha = 1$ means the prediction is an average value.

FISM assumes that each historical item contributes equally to the final prediction score, thus fails to accurately capture user interest with respect to the target item. NAIS [13] proposed a more reasonable strategy that applying dynamic weights to historical items which vary over different target items. The prediction of NAIS is formulated as

$$\hat{y}_{ui} = \sum_{j \in R_u} a_{ij} \mathbf{q}_i^\top \mathbf{q}_j \tag{3}$$

where a_{ij} denotes the attentive weight of similarity between item i and j. In NAIS, the attentive weight is calculated by a neural network whose inputs are the embeddings $\mathbf{q_i}$ and $\mathbf{q_j}$.

4 METHOD

In this section, we introduce CER in detail. First, we elaborate the design of enhanced representation of the target item. Then we introduce the sampling methods we designed. After that, we describe how to merge the raw and enhanced representations and brief the inference process of our model as shown in figure 3. Last, we discuss the time complexity of our model.

4.1 Enhanced Representation

In the previous section, we demonstrate several modeling approaches for target item using the collaborative signal. Here we follow the idea of representing items by users to introduce our methodology.

In recommendation system, like the user can be represented by the interacted items, item can also be represented by the interacted users. For the item i and the corresponding interacted user set U_i, the collaborative signal means that the users in this set must share part of interests with respect to the item i. Thus one feasible way to represent the item based on the user is to sum over the embeddings of these users. This method can be formulated as

$$\mathbf{r}_i = \frac{1}{|U_i|^\alpha} \sum_{j \in U_i} \mathbf{u}_j \tag{4}$$

where $\mathbf{u}_j$ denotes the embedding of user j, $\mathbf{r}_i$ denotes the enhanced representation of item i. However, each user may only have part of interests that are aligned with the target item. For instance, one user may be enthusiastic about the target item, thus his representation is highly related to the semantic information of the target item. On the contrary, one user may only have interacted with the item once for some reasons and his interest has nothing to do with the item. From this point, it is not appropriate that each user contributes equally to the $\mathbf{r}_i$.

To address the above problem, we adopt the attention mechanism to dynamically adjust the weight of each user. The new representation can be formulated as

$$\mathbf{r}_i = \sum_{j \in U_i} f(\mathbf{q}_i, \mathbf{u}_j) \mathbf{u}_j \tag{5}$$

where $f(\mathbf{q}_i, \mathbf{u}_j)$ denotes the function calculating the attentive weight of user embedding $\mathbf{u}_j$. The attention function can be MLP or inner product with item embedding $\mathbf{q}_i$ and user embedding $\mathbf{u}_j$ as input. In this way, user embeddings associated with the target item i will dominate in the representation $\mathbf{r}_i$ and therefore $\mathbf{r}_i$ will be more accurate than the former approach.
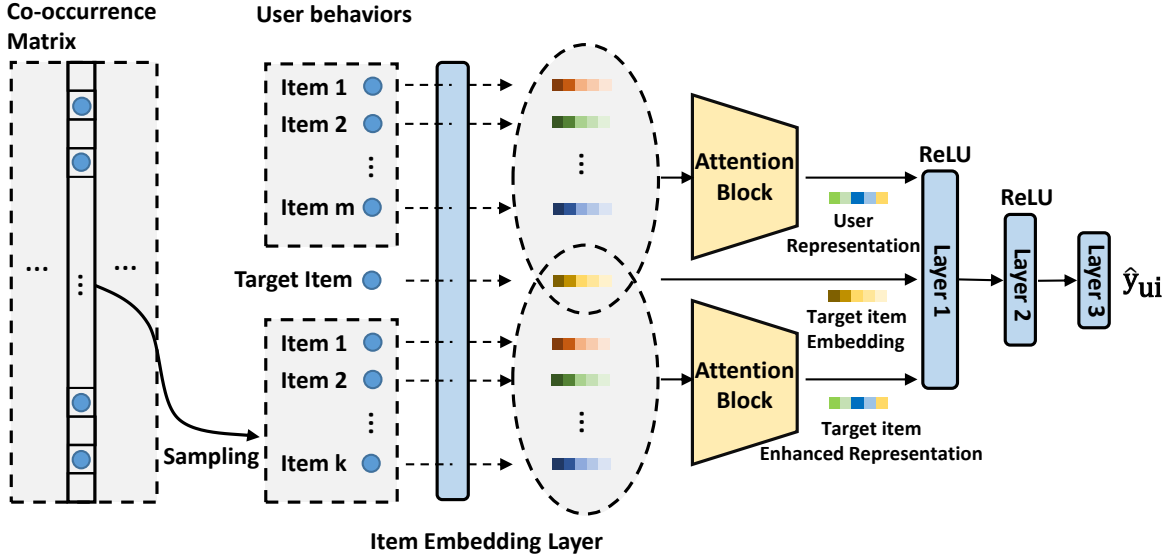
Figure 3: Overview of CER. The user representation encodes the user behaviors. The enhanced representation is learned based on the sampled co-occurrence items and the target item. Attention block uses the target item to selectively adopt information from user behaviors and co-occurrence items respectively. The neural network outputs $\hat{y}_{ui}$ as the preference of user u on target item i.

In the above section, we introduced the disadvantages of UCF. We find that equation 5 also has these limitations as it uses the user embedding $\mathbf{u}_j$. So we introduce an item-based representation of the user to tackle these problems. We firstly formulate the user representation $\mathbf{u}_j$ as

$$\mathbf{u}_j = \sum_{k \in R_j} g(\mathbf{q}_i, \mathbf{q}_k)\mathbf{q}_k \qquad (6)$$

where the function $g(\mathbf{q}_i, \mathbf{q}_k)$ outputs the similarity between target item i and historical item k. Some readers may find that this formulation is exactly the user representation used in both NAIS and DIN [33] (a widely used CTR model). This formulation focuses on the interest of the user with respect to the target item i, thus filters out some of the noise at the user level. On this basis, we further modify the formulation of enhanced representation as

$$\mathbf{r}_i = \sum_{j \in U_i} f(\mathbf{q_i}, \sum_{k \in R_j} g(\mathbf{q}_i, \mathbf{q}_k)\mathbf{q}_k) \sum_{k \in R_j} g(\mathbf{q}_i, \mathbf{q}_k)\mathbf{q}_k \qquad (7)$$

Actually, since the function f and g output real numbers, we can simplify the formulation as:

$$\mathbf{r}_i = \sum_{k \in O_i} (\sum_{n \in U_k} \sum_{m \in R_n} \alpha_n \beta_m)\mathbf{q}_k \qquad (8)$$

where O_i denotes the co-occurrence items set of target item i, α denotes the attentive weight of the user, β denotes the attentive weight of the historical item. In nature, this formulation aims to allocate an appropriate weight of each co-occurrence item based on two attentive weights. However, we argue that these attention mechanisms add additional constraints on learning the relevance between items. Since the part in parentheses essentially tries to

calculate a correlation, we can modify it into a more flexible form:

$$\mathbf{r}_i = \sum_{k \in O_i} f(\mathbf{q}_i, \mathbf{q}_k)\mathbf{q}_k \qquad (9)$$

where f denotes the attentive function. We choose the MLP as the attentive function in this paper and introduce the details of it in the latter subsection. This formulation also meets the idea of item2vec [2] that treating the co-occurrence items as items having similar semantics.

The current enhanced representation seems good as it has sufficient motivation and a concise form that demonstrates its effectiveness. However, there is still a problem hidden in the set O_i.

4.2 Sampling Strategy

Generally, the average length of the co-occurrence item set tends to be much larger than the average length of user behaviors. We analyze the datasets used later and find that these two lengths can sometimes differ by two orders of magnitude. For this kind of item set, even if adopting an efficient method, the calculation cost is still unacceptable. In addition, too large a co-occurrence item set will exacerbate the noise problem in enhanced representation.

To alleviate the above limitations, here we design sampling strategies to sample fix number of items from the corresponding co-occurrence item set. These sampling strategies follow two criteria: 1) the semantics of the picked item is as relevant as possible to the ground truth meaning of the target item. 2) the semantic inaccuracy of picked item is as small as possible. As the enhanced representation $\mathbf{r}_i$ is based on these picked items, if the picked items have these two advantages, $\mathbf{r}_i$ will also share the same advantages.

Global Sampling is a sampling method that aims to decrease semantic inaccuracy of representation $\mathbf{r}_i$. We already know that the semantic accuracy is highly related to the frequency of item. Thus we define the sampling probabilities of items as

$$s_k = \frac{f_k^g}{\sum_{j \in O_i} f_j^g} \tag{10}$$

where f_k^g denotes the times item k appears in the whole dataset. In simple terms, this method tends to pick out the items whose embeddings have high accuracy.

Local Sampling is a sampling method that aims to increase the semantic correlation between the picked items and the target item. Obviously, we need to pick out the items that may be more relevant to the target item i. However, it is hard to measure the semantic correlation accurately without any side information. Here we follow the basic idea that correlation between items is proportional to their co-occurrence frequency. Thus the sampling probability can be formulated as

$$s_k = \frac{f_k^l}{\sum_{j \in O_i} f_j^l} \tag{11}$$

where f_k^l denotes the times item k appears in the co-occurrence set. Different from the global sampling, this method focuses on relevance between item j and target item i, thus the meaning of enhanced representation is more close to the ground truth.

Weighted Sampling combines the global and local sampling by a weighted sum to strike a balance between correlation and inaccuracy. The sampling probability can be formulated as

$$s_k = \alpha * s_k^g + (1 - \alpha) s_k^l \tag{12}$$

where s_k^g denotes the global sampling probability of item k and s_k^l denotes the local sampling probability, α is a hyper-parameter to control the proportion of these two methods in the final probability.

Uniform Sampling is an unbiased sampling strategy in which each item has the same sampling probability

$$s_k = \frac{1}{|O_i|} \tag{13}$$

This method can be treated as a baseline sampling strategy compared with others.

4.3 Inference

In this subsection, we first describe how to merge the raw representation and enhanced representation of target item. At the same time, we introduce the process of inference based on user representation and merged representation. Then we brief the design of user representation and attention mechanism used in this paper.

In the inference stage, previous works adopted inner product or MLP to calculate the interest $\hat{y}_{ui}$ between user u and target item i. Now we get two representations of the target item, raw embedding $\mathbf{q}_i$ and enhanced representation $\mathbf{r}_i$. Intuitively, the final prediction can be formulated as

$$\hat{y}_{ui} = f(\mathbf{q}_i, \mathbf{e}_u) + f(\mathbf{r}_i, \mathbf{e}_u) \tag{14}$$

where $\mathbf{e}_u$ is the representation of user u calculated by equation 6, f is a function which can be either inner product or MLP. However,

this formulation sets the weights of raw and enhanced representations as the same by default, which has a limitation in dynamically adjusting the weights of different target items. For instance, the raw representation of a head item is good enough to be used in prediction, but its co-occurrence items may not reflect its semantic information as it appears too many times thus any item (especially the popular ones) can be added into the co-occurrence item set. Therefore, the final enhanced representation will be deviated from the raw meaning of the target item and exacerbates the prediction. On the contrary, for tail items, enhanced representations can alleviate the deficiency of raw embedding in expressing semantic information, thus deserve larger weight compared with the head items.

To tackle the above problem, we apply MLP to solve the fusion of two representations and the selection of prediction function. Specifically, we directly concatenate these three vectors and use the concatenated vector as the input of MLP. The formulation can be formulated as

$$\mathbf{c}_0 = concat(\mathbf{q}_i, \mathbf{r}_i, \mathbf{e}_u)$$
$$\mathbf{c}_k = ReLU(\mathbf{W}_k \mathbf{c}_{k-1} + \mathbf{b}_k) \tag{15}$$
$$\hat{y}_{ui} = \mathbf{W}_K \mathbf{c}_{K-1} + \mathbf{b}_K$$

where k denotes the number of the layers in MLP, K denotes the number of final layer in MLP, W_k and b_k are respectively the weight matrix and bias vector which are trainable parameters. In this design, the weights between $\mathbf{q}_i$ and $\mathbf{r_i}$ are learnable and vary over different target item i, which ensures that the contributions of two representations are optimal.

Finally, we brief the design of user representation $\mathbf{e}_u$ and attention mechanism used in this paper. For user representation $\mathbf{e}_u$, we follow the previous works [13, 33] and formulate the $\mathbf{e_u}$ as

$$\mathbf{e}_u = \sum_{j \in R_u} a_{ij} \mathbf{q}_j \tag{16}$$

where $\mathbf{q}_j$ denotes the embedding of interacted item j, a_{ij} denotes the weight calculated by attentive function with embedding of target item i and interacted item j as the input. In this work, all attentive weights (between interacted items or co-occurrence items) adopt the MLP as the function like

$$a_0 = concat(\mathbf{q}_i, \mathbf{q}_j)$$
$$a_k = ReLU(\mathbf{W_k} a_{k-1} + \mathbf{b}_{k-1}) \tag{17}$$
$$a_{ij} = \mathbf{W}_K a_{K-1} + \mathbf{b}_K$$

where $\mathbf{W}_k$, $\mathbf{b}_k$ and a_k denote the weight matrix, bias vector and output vector of the kth hidden layer, respectively. We will report the detailed setting of both prediction MLP and attentive MLP in the experiment section.

4.4 Time complexity

In this subsection, we discuss the time complexity of our model. In the training stage, we first show that the time complexity of FISM in evaluating a prediction is $O(d|R_u|)$, where d denotes the size of embedding. For the NAIS, as reported in reference [13], the time complexity is $O(d'd|R_u|)$, where d' denotes the attention factor. For our model, the time complexity in calculating the attentive function is $O(\sum_{k=1}^{K} d_{k-1}d_k)$, thus the final time complexity of our model

Table 1: The statistics of datasets.

Dataset	#Users	#Items	#Interactions	Density
Books	603,668	367,982	8,898,041	0.004%
Beauty	22,363	12,101	198,502	0.073%

is $O(|R_u| \sum_{k=1}^{K} d_{k-1}d_k + |O_i| \sum_{k=1}^{K} d_{k-1}d_k + \sum_{k=1}^{K} d'_{k-1}d'_k)$, where three parts denote the time complexity of calculating user representation, enhanced representation and prediction respectively. d'_k denotes the size of kth hidden layer in prediction MLP.

The time complexity of our model in the training stage seems pretty large compared with other methods. Actually, if we look back at the equation 9, we will find that the enhanced representation has nothing to do with the users and only depends on the sampled co-occurrence items. Therefore, once the training of the model is completed, the enhanced representation of items can be calculated offline. We can first calculate the enhanced embedding matrix and then store it as part of the model. In the stage of online inference, the model can directly look up the enhanced embedding matrix and skip the calculation of it. So the time complexity in testing stage actually is $O(|R_u| \sum_{k=1}^{K} d_{k-1}d_k + \sum_{k=1}^{K} d'_{k-1}d'_k)$.

5 EXPERIMENTS

In this section, we conduct experiments on two public datasets with the aim of answering the following research questions:

- **RQ1:** How does our proposed model perform compared to other state-of-the-art recommender models?
- **RQ2:** How does our proposed model perform on target items with low frequency?
- **RQ3:** How does the choice of sampling strategy impose an influence on the performance of our model?
- **RQ4:** Besides the performance on tail items, does the enhanced representations actually better than the embeddings for tail items? Could we see the differences visually?

5.1 Experimental Settings

5.1.1 Dataset Description. To evaluate the effectiveness of our model, we conduct experiments on two real-world datasets: Amazon-Beauty and Amazon-Books, which vary in terms of size, domain and sparsity. Table 1 summarizes the statistics of the two datasets.

Amazon dataset [20] is widely used as a benchmark dataset in the recommendation. Here we choose two subsets of Amazon product dataset to conduct our experiments, containing Beauty and Books. These datasets are all reduced to satisfy the 5-core property.

All datasets are truncated at length 200 (keep the last 200 behaviors of users). Given all behaviors of a user $[x_1, x_2, \ldots, x_n]$, we construct training set by taking the sequence $[x_1, \ldots, x_{n-2}]$ as user behaviors and the task is to predict the (n-1)-th item. In the testing stage, the task is to predict the n-th item based on the behaviors of user $[x_1, \ldots, x_{n-1}]$. To tune hyper-parameters, we select half of the users as the validation set and treat the remaining users as the testing set. For each interaction, we pair it with one negative item that the user did not interact with before.

5.1.2 Evaluation Metrics. For easy and fair evaluation, following references [3, 16], we randomly sample 100 negative items for each positive item and rank the test item among the 100 items for each user. We employ evaluation metrics including Hit Ratio (HR) [8] and Normalized Discounted Cumulative Gain (NDCG) [10] at position 10. Considering we have only one positive item for each user, HR@K is equivalent to Recall@K and proportional to Precision@K. All these metrics are that the higher the values are, the better the performance is. All experiments are repeated 5 times on the testing set and averaged values are reported.

5.1.3 Compared Methods. We choose several baselines to evaluate the effectiveness of our model.

- **MF.** This is a standard matrix factorization that exploits the user-item direct interactions.
- **MLP.** This method replaces the inner product in MF with a multi-layer perceptron to learn the scoring function about user and item embeddings. Here we adopt the same structure of MLP used in our prediction stage, which has 3 layers and optimizes the model with the pointwise log loss.
- **Youtube Rec [7].** YouTube employed this method in their video recommendation. This method averages the embeddings of videos the user has interacted with and feeds it into a feed-forward neural network.
- **FISM.** FISM is one of the state-of-the-art item-based CF methods. It represents users by their interacted items and adopts the inner product as the scoring function.
- **NAIS.** NAIS additionally introduces attention to adjust the representation of users dynamically for different target items. Here we choose the version which uses the element-wise product based attention function.
- **DIN.** This method is a widely used CTR prediction model. It shares a similar structure to NAIS, but it adopts the MLP as the scoring function.

5.1.4 Parameter Settings. We implement our model and all baselines in Tensorflow. For the hyper-parameters, we apply a grid search on the validation set: the embedding size is tuned among {8, 16, 32, 64}, the learning rate in {0.03,0.1,0.3,1.0}. For the hyper-parameter α in the model FISM and NAIS, the range is {0, 0.1, ..., 0.9,1}. The scaled factor β of attention in NAIS is set as 0.5 which is the default setting in their paper. For all models, we set the batch size as 32 and choose the SGD as the optimizer. Both the prediction MLP and attentive MLP used in our model adopt the same structure that the sizes of hidden layers are [80, 40, 1] with ReLU as the active function. For the hyper-parameter α we used in weighted sampling, it was set to 0.6 for the Beauty dataset and 0.2 for the Books dataset.

5.2 Performance Comparison (RQ1)

In this subsection, we compare our proposed model with other baselines. Table 2 reports the performance of HR@10 and NDCG@10 of all compared methods.

First, we can see that CER achieves the best performance on both datasets and surpasses the state-of-the-art methods NAIS and DIN by a large margin. On the Beauty dataset, our proposed method outperforms FISM with about 10.42% relative improvement in terms of HR and 10.51% relative improvement in terms of NDCG. On the

Table 2: Model comparison on public datasets. Relative Improvement (RI) is based on FISM.

Methods	Books				Beauty			
	HR	RI	NDCG	RI	HR	RI	NDCG	RI
MF	0.4910	-27.54%	0.3169	-33.74%	0.3628	-19.36%	0.2074	-25.13%
MLP	0.4917	-27.44%	0.3171	-33.70%	0.3777	-16.05%	0.2101	-24.15%
Youtube Rec	0.7162	5.70%	0.4991	4.35%	0.4399	-2.22%	0.2588	-6.57%
FISM	0.6776	0.00%	0.4783	0.00%	0.4499	0.00%	0.2770	0.00%
NAIS	0.6774	-0.03%	0.4824	0.86%	0.4617	2.62%	0.2846	2.74%
DIN	0.7193	6.15%	0.5069	5.98%	0.4482	-0.38%	0.2763	-0.25%
CER	**0.7987**	**17.87%**	**0.6062**	**26.74%**	**0.4968**	**10.42%**	**0.3061**	**10.51%**
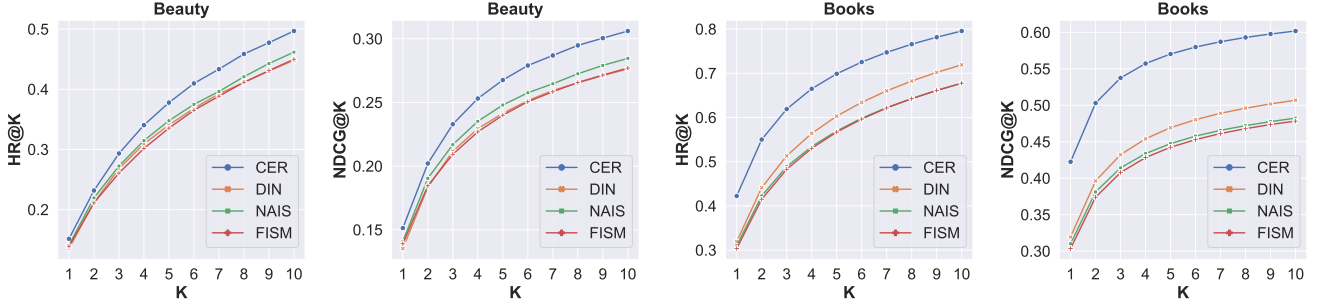
Figure 4: Performance of Top-K recommendation with ranking positions K ranging from 1 to 10.

Books dataset, the relative improvement is 17.87% HR and 26.74% NDCG respectively. Basically, we can divide the baselines into three categories. First, UCF methods including MF and MLP. Second, representing User By Items (UBI) methods including Youtube, FISM, NAIS and DIN. Last, we representing both user and item by items method. As shown in table 2, among each type of methods, different tricks like applying MLP or attention lead to a certain amount of difference in performance. However, the performance of methods from different types tends to show an obvious gap.

For UCF methods, whether the inner product or MLP is adopted as the interaction function, it always achieves the worst performance among all methods. This is because representing the interests of user by a single embedding limits the ability of the model to fit and generalize.

Compared with UCF methods, UBI methods achieve great improvements on both datasets. These improvements are attributed to that representing users with their consumed items encodes more information than UCF methods, which improves the quality of user expression. Among these methods, the performance of methods that apply no attention like Youtube Rec and FISM is slightly lower than the performance of NAIS and DIN which are attention-based methods. It demonstrates that dynamical adjusting the expression of user interest with respect to target items helps improve the accuracy of user modeling.

Based on the UBI methods, our proposed CER provides an item-based representation of the target item based on its co-occurrence items. This enhanced representation contributes to alleviating the semantic loss of tail items and thus improve the performance of ICF

methods, especially for low-frequency target items. As we can see in table 2, our method significantly outperforms all UBI methods on both datasets, which demonstrates the effectiveness of our method.

Figure 4 shows the performance of Top-K recommendations with ranking position K ranging from 1 to 10. As can be seen, CER outperforms UBI methods consistently across positions. Among all baselines, attention-based methods NAIS and DIN perform well on Beauty and Books dataset respectively.

5.3 Performance on Low Frequency Target Items (RQ2)

To figure out how does our proposed model performs on low frequency target items, we conduct experiments to demonstrate the effectiveness of our model, especially for the tail target items. Here we split the target items in the testing set by their global frequencies. For each frequency, we calculate both evaluation metrics based on the items with that frequency. Here we select the frequencies to range from 0 to 50 for purpose of focusing on low frequency target items.

Figure 5 shows the performance on different frequencies of items. Firstly, as we can see, in all UBI methods, the values of both HR and NDCG increase with the increase of target item frequency. It proves that the frequency of the target item does have a powerful influence on the performance of UBI methods. In addition, we can see that the low frequency target items (frequencies ranging from 0 to 30) perform far below the overall level of the model. Since the long-tail distribution, substantial such items can dramatically drag down the overall performance of the model.
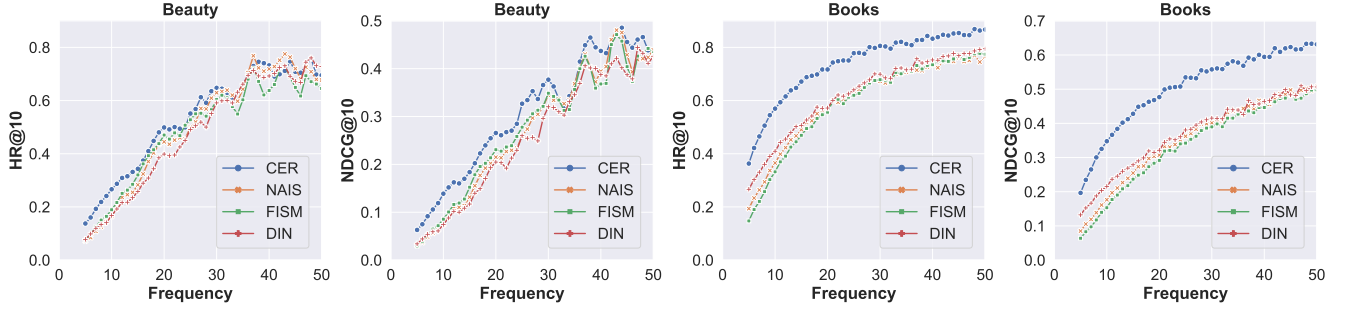
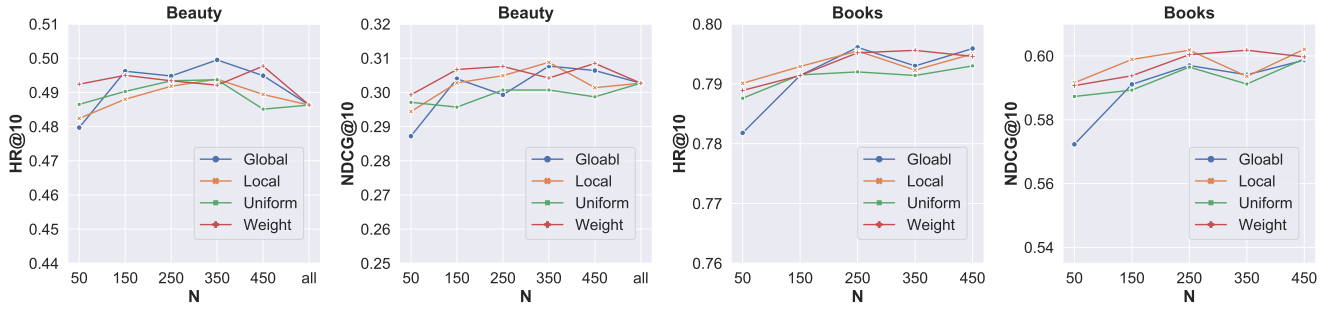**Figure 5: Performance comparison on low frequency target items**

Figure 6: Performance of different sampling strategies. 'all' means directly using all co-occurrence items without sampling.

Secondly, on the Beauty dataset, our proposed model outperforms other UBI methods, especially for frequencies ranging from 0 to 30. On the Books dataset, our method is far superior to other UBI approaches on both evaluation metrics. This demonstrates that enhanced representation is effective in boosting the performance of UBI model on the low frequency target item. To illustrate figure 5 more clearly, we take the Books dataset as an example. Considering the HR@10 value when the frequency equals 10, the best performance of UBI methods is DIN which is almost 0.41. However, our proposed model achieves almost 0.57 on the same frequency which has a nearly 39 percent relative improvement on the basis of DIN. Considering the corresponding frequency when NDCG@10 equals 0.5, the best performing UBI method DIN has a frequency of almost 44. For our proposed model, the corresponding frequency is almost 21. That is to say, to achieve the same performance on NDCG@10, our method only need the target item to appear 21 times in the dataset and the UBI method need at least 44 times. In general, the results show that our proposed model has incomparable advantages over other ICF models in low frequency target items.

## 5.4 Performance of Different Sampling Strategies (RQ3)

Considering noise reduction and computational efficiency, we design several sampling strategies to sample fix number of items from the co-occurrence item set, following different criteria. The question is does the choice of sampling strategy play an important role in our proposed model? To figure out this question, we conduct

experiments to show the influence of different sampling strategies and the influence of the number of the sampled items. We choose the four aforementioned sampling methods and the numbers of sampling include [50, 150, 250, 350, 450]. For Beauty dataset, we additionally add an experiment that directly using all co-occurrence items without sampling. Considering the computational cost, this experiment was not performed on the Books dataset.

Figure 6 shows the experiment results of different sampling strategies. As we can see, the uniform sampling method is basically inferior to other sampling methods in most datasets and evaluation metrics. It is because that uniform sampling is an unbiased sampling method that ignores some important information such as global and local frequency. For other sampling methods, the global sampling method works well on HR but has relatively bad performance on NDCG. Local sampling, in contrast, performs well on NDCG for the most parts, but not so well on HR. This may be because that different concerns of the sampling method have different effects on the evaluation metrics. For instance, local sampling tends to pick out items that have high semantic similarity, thus it performs better on NDCG which emphasizes more on accuracy. As the weighted sampling balances the global and local sampling and it works well in most cases, here we recommend weighted sampling as a default choice.

Actually, for the numbers of sampled items, as long as N is not too small say 50 or less, it does not make much influence on the model with the weighted sampling method. As shown in figure 6, the important thing is that taking all co-occurrence items into
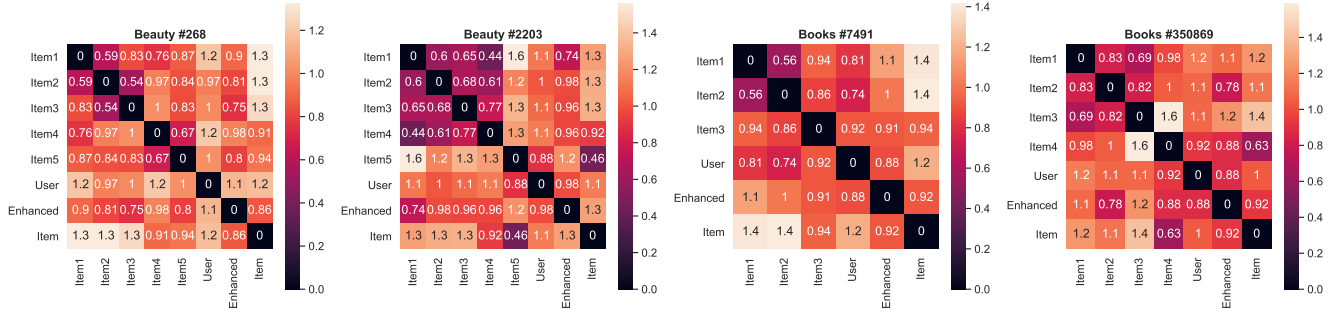
Figure 7: Visualization of cosine distances between representations including user representation, enhanced representation, target item embedding and embeddings of historical items with the same category of target item (Item1-Item5). The left two users #268 and #2203 are sampled from the Beauty dataset. Another two users #7491 and #350869 are sampled from Books dataset.

account may not contribute to the performance of the model. In fact, we can directly use the length of user truncation M (in this paper $M = 200$) as a reference, say $N = kM$ where k is a real number. In this setting, a basic guarantee can be obtained is that the computational cost is acceptable and the performance is good enough.

5.5 Visualization and Case Study (RQ4)

Finally, we conduct a case study to figure out what is this enhanced representation really learning. Although it is difficult to exactly describe what a high dimensional vector represents, its semantic information can still be inferred by comparing distances between vectors. If we already know a vector a representing information A, and if another vector b is close to a in their latent space, we could tell that b also represents information A to a great extent. To figure out the meaning of enhanced representation, we choose several representations for reference: **1)** historical items which have the same category as target item (we call them reference items for brevity), we label them from Item1 to Item5 as shown in figure 7. We choose them because they have the same category as the target item, thus can be treated as representing similar semantics. **2)** user representation. **3)** target item embedding. Here we sampled two users from each testing set. For each user, the target item must have a frequency lower than 10. Then we calculate their representations mentioned above and plot the cosine distances between them in figure 7.

First, we can focus on the 'enhanced' row and the 'item' row of each subgraph. As we can see, for most reference items their cosine distances between enhanced representation are smaller than that of target item embedding. That is to say, for low frequency items, enhanced representation can represent the semantics of the target item more accurately than its raw embedding. Second, one of our purposes of proposing enhanced representation is to alleviate the problem that target item embedding with poor quality leads to the mismatch between target item embedding and user representation. From figure 7 we can see that cosine distances between enhanced representation and user representation are smaller than that between target item embedding and user representation. In another

Table 3: Comparison of time consumption in different stages. We report the time cost of each epoch on the Beauty dataset as the training time and report the average testing time of each user on Beauty dataset as the testing time.

Methods	MF	MLP	FISM	NAIS	DIN	CER
Training (s)	41.67	58.24	47.78	83.78	111.92	173.80
Testing (ms)	0.3	0.6	0.9	2.2	3.1	3.3

word, compared with the row embedding of the target item, enhanced representation is more compatible with user representation, which can reduce the difficulty of learning the scoring function.

5.6 Time Complexity

To intuitively reflect the time complexity of each model in the training and testing stage, we report the training time cost of each epoch and the average testing time of each user on the Beauty dataset in table 3. All experiments are conducted with the same hardware environment. In the training stage, our proposed CER cost a lot compared to other methods due to the calculation of enhanced representation. However, in the testing stage, our method shows a similar inference time compared to DIN or NAIS. It benefits from that the enhanced representation can be calculated offline after the training stage and directly used in the testing stage.

6 CONCLUSION

In this paper, we propose a novel enhanced representation of the target item and an item-based CF model based on enhanced representation named CER for the recommendation. Our key argument is that the quality of target item representation plays an important role in the performance of existing ICF methods and the long-tail distribution deteriorates the quality of the target item embedding. The proposed enhanced representation selectively learns semantic information from the co-occurrence items of target item, thus alleviates the lack of semantics caused by the long-tail distribution. Extensive experiments on two public datasets demonstrate the effectiveness of our proposed model.

REFERENCES

[1] Ting Bai, Ji-Rong Wen, Jun Zhang, and Wayne Xin Zhao. 2017. A neural collaborative filtering model with interaction-based neighborhood. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 1979–1982.

[2] Oren Barkan and Noam Koenigstein. 2016. Item2vec: neural item embedding for collaborative filtering. In *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 1–6.

[3] Immanuel Bayer, Xiangnan He, Bhargav Kanagal, and Steffen Rendle. 2017. A generic coordinate descent framework for learning from implicit feedback. In *Proceedings of the 26th International Conference on World Wide Web*. 1341–1350.

[4] Alex Beutel, Paul Covington, Sagar Jain, Can Xu, Jia Li, Vince Gatto, and Ed H Chi. 2018. Latent cross: Making use of context in recurrent recommender systems. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. 46–54.

[5] Jingyuan Chen, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu, and Tat-Seng Chua. 2017. Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. 335–344.

[6] Weiyu Cheng, Yanyan Shen, Yanmin Zhu, and Linpeng Huang. 2018. DELF: A Dual-Embedding based Deep Latent Factor Model for Recommendation.. In *IJCAI*, Vol. 18. 3329–3335.

[7] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*. 191–198.

[8] Mukund Deshpande and George Karypis. 2004. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems (TOIS)* 22, 1 (2004), 143–177.

[9] Xue Geng, Hanwang Zhang, Jingwen Bian, and Tat-Seng Chua. 2015. Learning image and user features for recommendation in social networks. In *Proceedings of the IEEE International Conference on Computer Vision*. 4274–4282.

[10] Xiangnan He, Tao Chen, Min-Yen Kan, and Xiao Chen. 2015. Trirank: Review-aware explainable recommendation by modeling aspects. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. 1661–1670.

[11] Xiangnan He and Tat-Seng Chua. 2017. Neural factorization machines for sparse predictive analytics. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. 355–364.

[12] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. *arXiv preprint arXiv:2002.02126* (2020).

[13] Xiangnan He, Zhankui He, Jingkuan Song, Zhenguang Liu, Yu-Gang Jiang, and Tat-Seng Chua. 2018. Nais: Neural attentive item similarity model for recommendation. *IEEE Transactions on Knowledge and Data Engineering* 30, 12 (2018), 2354–2366.

[14] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*. 173–182.

[15] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. 2016. Fast matrix factorization for online recommendation with implicit feedback. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. 549–558.

[16] Kurt Hornik, Maxwell Stinchcombe, Halbert White, et al. 1989. Multilayer feedforward networks are universal approximators. *Neural networks* 2, 5 (1989), 359–366.

[17] Santosh Kabbur, Xia Ning, and George Karypis. 2013. Fism: factored item similarity models for top-n recommender systems. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. 659–667.

[18] Yehuda Koren. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. 426–434.

[19] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural attentive session-based recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 1419–1428.

[20] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*. 43–52.

[21] Xia Ning and George Karypis. 2011. Slim: Sparse linear methods for top-n recommender systems. In *2011 IEEE 11th International Conference on Data Mining*. IEEE, 497–506.

[22] Arkadiusz Paterek. 2007. Improving regularized singular value decomposition for collaborative filtering. In *Proceedings of KDD cup and workshop*, Vol. 2007. 5–8.

[23] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618* (2012).

[24] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*. 285–295.

[25] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*. 165–174.

[26] Jun Xiao, Hao Ye, Xiangnan He, Hanwang Zhang, Fei Wu, and Tat-Seng Chua. 2017. Attentional factorization machines: Learning the weight of feature interactions via attention networks. *arXiv preprint arXiv:1708.04617* (2017).

[27] Feng Xue, Xiangnan He, Xiang Wang, Jiandong Xu, Kai Liu, and Richang Hong. 2019. Deep item-based collaborative filtering for top-n recommendation. *ACM Transactions on Information Systems (TOIS)* 37, 3 (2019), 1–25.

[28] Hong-Jian Xue, Xinyu Dai, Jianbing Zhang, Shujian Huang, and Jiajun Chen. 2017. Deep Matrix Factorization Models for Recommender Systems.. In *IJCAI*, Vol. 17. Melbourne, Australia, 3203–3209.

[29] Hongzhi Yin, Bin Cui, Jing Li, Junjie Yao, and Chen Chen. 2012. Challenging the long tail recommendation. *arXiv preprint arXiv:1205.6700* (2012).

[30] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 353–362.

[31] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys (CSUR)* 52, 1 (2019), 1–38.

[32] Yin Zhang, Derek Zhiyuan Cheng, Tiansheng Yao, Xinyang Yi, Lichan Hong, and Ed H Chi. 2020. A Model of Two Tales: Dual Transfer Learning Framework for Improved Long-tail Item Recommendation. *arXiv preprint arXiv:2010.15982* (2020).

[33] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1059–1068.