Robust Enhanced Collaborative Filtering Without Explicit Noise Filtering

Rong Fan
Zhenhai Wang
wangzhenhai@lyu.edu.cn

Yunlong Guo
Yuhao Xu
Zhiru Wang
Weimin Li

Research Article

Keywords: Recommendation, Collaborative Filtering, Graph Neural Network, Contrastive interaction noise,Maximize mutual information

Posted Date: June 19th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-3065294/v1

Additional Declarations: No competing interests reported.

Version of Record: A version of this preprint was published at The Journal of Supercomputing on April 6th, 2024. See the published version at https://doi.org/10.1007/s11227-024-06086-w.

Robust Enhanced Collaborative Filtering Without Explicit Noise Filtering

Rong Fan[1], Zhenhai Wang[1*], Yunlong Guo[1], Yuhao Xu[1], Zhiru Wang[1], Weimin Li[2]

[1]College of Information Science and Engineering, Linyi University, Linyi, 276000, Shandong, China.

[2]School of Computer Engineering and Science, Shanghai University, Shanghai, 200444, China.

*Corresponding author(s). E-mail(s): wangzhenhai@lyu.edu.cn;
Contributing authors: 210854002002@lyu.edu.cn; gyl0805123@gmail.com; xyh0811@gmail.com; wzr_9911@163.com; wmli@shu.edu.cn;

Abstract

Graph convolutional neural networks have been successfully applied to collaborative filtering to capture high-qualityuser-item representations. Despite their remarkable performance, there are still limitations that hinder further improvement of recommender systems. Most existing recommender systems utilize implicit feedback data for model training; however, this approach inevitably introduces adversarial interaction noise. The conventional graph-based collaborative filtering method fails to effectively filter out this noise and, instead amplifies its impact, resulting in degraded model performance. To address this issue, we propose a robustness-enhanced collaborative filtering graph neural network model that does not rely on explicit noise filtering. Our approach involves simulating user-item interactions that do not exist in practice as adversarial interaction noise using random noise. To mitigate the impact of this noise in hidden feedback, we replace them with randomly selected partial nodes based on the principle of mutual information maximization. This approach not only improves model performance but also enhances the robustness of the model. Through experimental demonstrations on three benchmark datasets, our model exhibits significant improvement, thereby validating the effectiveness and interpretability of our proposed approach.

Keywords: Recommendation, Collaborative Filtering, Graph Neural Network, Contrastive interaction noise,Maximize mutual information

1 Introduction

Learning vector representations of users and items is crucial in modern recommender systems[1, 2] that enable mitigating the issue of information overload. Collaborative filtering (CF) is a traditional recommendation model that relies on user feedback; however, its co-occurrence matrix often suffers from sparsity due to limited user behavior data, resulting in subpar recommendations. Matrix factorization (MF)[3] addresses this problem by decomposing the co-occurrence matrix into low-rank matrices. While deep learning has revolutionized computer vision, its application to CF in the recommendation field has been limited. Neural collaborative filtering (NCF)[4] introduces neural networks to traditional CF, utilizing a multilayer perceptron to capture user-item interactions with promising results. However, existing approaches are still deficient in learning high-quality user representations, as they only map a single ID or attribute of the user or item. Graph structures naturally capture user-item interaction behavior, and graph neural networks[5, 6] (GNNs) excel at learning representations from such structures. Consequently, GNNs have gained traction in recommendation systems. NGCF[7] proposed a CF framework based on neural graphs, leveraging a bipartite graph structure to enhance higher-order connectivity in the user-item graph and improve recommendation performance. HMLET[8] combined linear and nonlinear CF using graph convolutional networks (GCNs) to address training difficulties and excessive smoothing. ULtraGCN[9] simplified the GCN structure and introduced two auxiliary loss functions, achieving significant results. Although GCN have been successful in recommender systems, they also have noticeable limitations. Existing recommender systems typically employ implicit feedback data for training, which introduces false interactions[10–12] and a gap between implicit feedback and real user satisfaction. The limitations can be categorized into three main aspects: (1) vulnerability to adversarial noise attacks in implicit feedback data, such as simulated interactions that do not reflect genuine user preferences, (2) a significant portion of interactions driven by user mistakes and curiosity, and (3) unobserved interactions, which may not indicate user disinterest but rather lack of exposure to items. Current approaches that fit implicit feedback to recommender systems without considering inherent noise, particularly in the context of GCN, fail to effectively filter noise and inadvertently amplify its impact, leading to decreased recommendation performance and misinterpretation of user preferences[13]. To address the issue of implicit feedback noise, recent studies have explored denoising methods, including resampling[14–18] and reweighting[19, 20] While resampling methods focus on learning user preferences by designing a more efficient sampler to sample clean samples, they are heavily dependent on the sample distribution. Reweighting methods define clean data and noisy interaction data mainly based on the loss values during training, attributing lower weights to noisy data with high losses. For example, ADT[21] proposes an adaptive denoising training strategy that adaptively prunes the interactions with large losses. Furthermore, SGDL[22] based studies have proposed a novel denoising strategy, self-guided denoising learning, which uses two phases (memory phase and self-guided phase) to learn clean and rich information. However, the existing denoising approaches often rely on auxiliary information[23–25] or additional loss functions that are sensitive to hyperparameters and may result in the exclusion of difficult samples, thereby reducing model performance.

In this study, we present a research proposal for a robustness-enhanced CF GNN model to address the problem of adversarial interaction noise in implicit feedback data. Rather than explicitly filtering the noise, our approach involves randomly selecting nodes and replacing them with virtual users or items to counteract the noise and enhance the overall view. To achieve this, we simulate virtual users or items using random noise and leverage the principle of mutual information maximization. We utilize the InfoNce[26] loss optimization function to maximize the consistency between different views of the same node, thereby reducing the impact of adversarial interaction noise in hidden feedback. The key contributions of our research are as follows:

• We propose a method to simulate user-item interactions that do not occur in practice by introducing random noise and replacing selected nodes with simulated user-item interactions.

• By maximizing the consistency between augmented graphs based on mutual information maximization, we mitigate the effects of adversarial noise in hidden feedback.

• Experimental studies are conducted in three publicly available datasets and the experimental results are analyzed to demonstrate the superiority of our model.

2 PRELIMINARIES

This section provides an introduction to traditional CF methods based on matrix decomposition in recommender systems, as well as the application of GCNs in CF. Additionally, we present a model of lightweight graph convolutional neural network (LightGCN).

2.1 Traditional CF Algorithm

CF is a widely-used technique in modern recommender systems. It operates on the assumption that there are similar users or items and leverages these similarities to make recommendations. MF is a traditional CF algorithm that addresses the challenges posed by sparse matrices and high dimensionality of similarity matrices. The MF algorithm decomposes the user-item interaction scoring matrix into two matrices: a matrix of user hidden vectors and a matrix of item hidden vectors. This decomposition maps the original matrix into a lower-dimensional hidden factor space of dimension f. The interaction between the user and item hidden vectors is modeled as an inner product of the respective vectors, as shown in the following equation:

$$\hat{r}_{ui} = q_i^T p_u \tag{1}$$

where p_u is a vector representation of each user, and q_i is the vector representation of each item, the user and item.Each dimension of the vector represents the strength of a specific hidden factor. For example, if the first dimension represents the action movie, then the value of the first dimension of the vector represents the user's preference for the action movie.The first dimension of the vector, item represents the number of action components. The better the fitting of vector $\hat{r}_{ui}$ for a user-item pair, the higher the rating. In other words, the better the vectors of the user and item match, the higher the predicted rating for that item by the user.

Common matrix decomposition methods, such as singular value decomposition (SVD), have been widely used. However, applying SVD to user-item matrices requires dense matrices, and since these matrices are typically sparse, early systems used estimation techniques to fill in missing values. However, these estimations are often not accurate. As a result, most recommender systems only consider observed interaction data for modeling and incorporate regularization techniques to prevent overfitting. The optimization problem for CF can be expressed as follows:

$$min \sum_{(u,i)\epsilon k} (r_{ui} - q_i^T p_u)^2 + \lambda(\parallel q_i \parallel^2 + \parallel p_u \parallel^2) \tag{2}$$

To solve the optimization problem, stochastic gradient descent algorithms are commonly employed. In addition to traditional CF algorithms, there have been efforts to enhance their performance by incorporating additional information. One such example is SVD++[27], which combines the strengths of MF with additional factors to improve recommendation accuracy. However, traditional CF algorithms still suffer from limitations due to their reliance on limited information. As a result, these algorithms may not effectively capture the unique characteristics of individual users.

2.2 GCN

With the advent of deep learning, there has been a surge in the application of deep learning-based methods in recommender systems. GCNs have gained significant attention due to their ability to extract meaningful representations from graph structures. Given that user-item interactions can be represented as graph structures, GCNs have been employed in CF to capture higher-order information.

In the graph convolution-based CF model, user-item interactions are represented by a bipartite graph. As shown in Figure 1,Each user and item is considered as a node in the bipartite graph, while the interactions between users and items are represented as edges connecting the corresponding nodes. This bipartite graph provides a comprehensive representation of the user-item relationships.

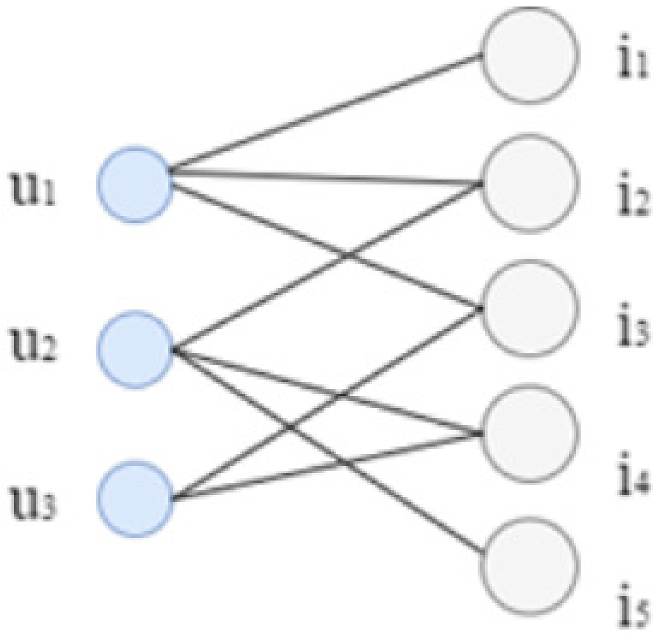

Fig. 1 user-item interaction diagram

Given a bipartite graph of user-item interactions, let the set of users be U and the set of items be I;the interaction between users and items is denoted as the set O_+,and $O_+ = \{yui \mid u\epsilon U, i\epsilon I\}$, where yui represents the user u, and project i is generated. Furthermore,

$$G = (V, \varepsilon) \tag{3}$$

where $V = U \cup I$ represents the nodes of all users and items, and $\varepsilon = O_+$ represents the generated interaction. The basic idea of GCN is to aggregate the features of the neighboring nodes on the graph and represent them as new nodes, whose first-order aggregation operation is denoted as following:

$$Z^{(1)} = H(Z^0, G) \tag{4}$$

where H represents the neighbor aggregation function,Z^0 represents the initial embedding representation of users and items, and Z^1 is the representation of the nodes in layer 1 after aggregation; similarly, after several iterations of convolution,

$$Z^{(l+1)} = H(Z^l, G) \tag{5}$$

where l represents the node representation of convolutional layer l.

2.3 LightGCN

LightGCN is based on the basic concept of GCN and simplifies two common operations in graph convolution to capture the higher-order information of the graph and accelerate the training by aggregating higher-order neighbor information to update the representation of self-nodes and form the final embedding. In this study, we adopt the basic architecture of LightGCN. The two unnecessary redundant GCN operations of feature matrix transformation and nonlinear activation function are analyzed, proved, and further simplified into the final iterative formulation as follows:

$$e_i^{(k+1)} = \sum_{u \epsilon N_i} \frac{1}{\sqrt{\mid N_i \mid}\sqrt{\mid N_u \mid}} e_u^{(k)} \tag{6}$$

$$e_u^{(k+1)} = \sum_{i \epsilon N_u} \frac{1}{\sqrt{\mid N_u \mid}\sqrt{\mid N_i \mid}} e_i^{(k)} \tag{7}$$

where the item embedding at layer k+1 is represented as the sum of the neighboring user embeddings at layer k. To prevent the increasing size of the embedding $\frac{1}{\sqrt{|N_u|}\sqrt{|N_i|}}$, a normalization operation is performed on the nodes. After stacking the multilayer graph convolutional neural network, the aggregation is performed using a weighted summation based on the following weighting formula:

$$e_u = \sum_{k=0}^{k} \alpha_k e_u^{(k)} \tag{8}$$

$$e_i = \sum_{k=0}^{k} \alpha_k e_i^{(k)} \tag{9}$$

Ultimately, the final codes of the user and item representations are multiplied to obtain the score as follows:

$$\hat{y}_{u,i} = e_u^T e_i \tag{10}$$

3 METHODOLOGY

In this section, we introduce our proposed GNN model for robustness-enhanced CF without explicit noise filtering. Our aim is to mitigate the impact of adversarial interaction noise present in implicit feedback data. To achieve this, we adopt a strategy where we randomly select certain nodes and introduce virtual users or items with random noise to simulate interactions that do not exist in real situations. These simulated interactions are considered as adversarial interaction noise in the implicit feedback. We then replace the selected nodes with these simulated user or item nodes. To optimize the model, we utilize the principle of mutual information maximization. Specifically, we employ InfoNce as the loss optimization function for the joint training of our model.

The overall framework of our proposed model is depicted in Figure 2, providing a visual representation of the model architecture.

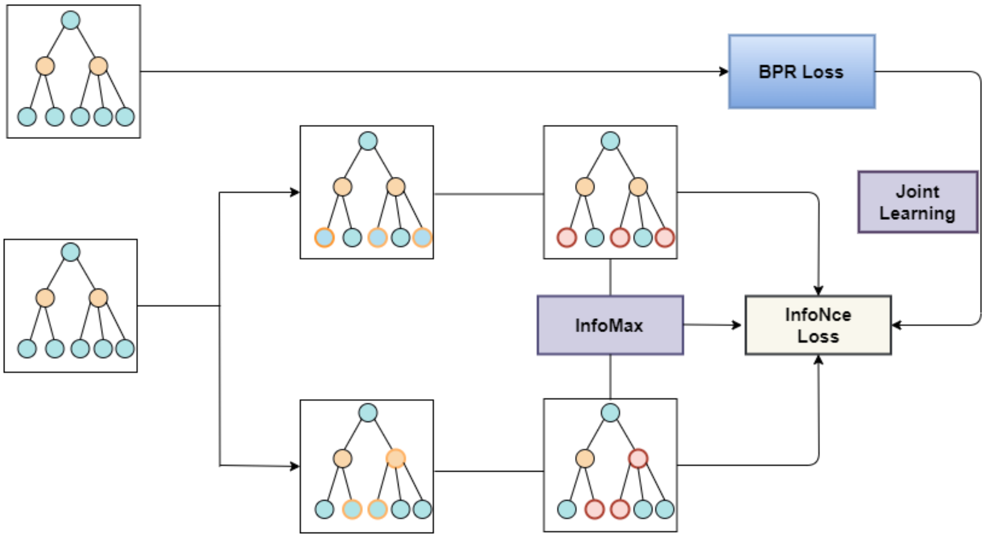

Fig. 2 Overall framework of the model

3.1 Contrastive Interaction Noise

In current recommendation systems, implicit feedback is widely utilized for learning recommendation models. The abundance of implicit feedback helps overcome the data sparsity issue in recommendation systems. However, a common challenge arises in the form of adversarial interaction noise. This noise refers to the inclusion of illegitimate samples that are mistakenly treated as legitimate by the system. Examples of adversarial interaction noise include users' accidental clicks, simulated interactions performed by merchants, and interactions that users have not been exposed to. In practical scenarios, this noise is often considered as negative examples, but it is typically treated as positive examples in recommendation models. To address this issue, we propose a strategy to mitigate the impact of interaction noise in implicit feedback.

Initially, we simulate users or items by introducing random noise. This approach helps reduce the distortion of the original graph information. Rather than simply removing or adding interactions to simulate adversarial noise, which may compromise the modeling of users' true preferences, we simulate user or item nodes. Accordingly, we generate interactions that do not exist in real-world situations, serving as adversarial interaction noise. To implement this strategy, we randomly select a subset of nodes, as expressed by the following equation:

$$V_R = G_{randint}(V, p) \tag{11}$$

where $G_{randint}$ is a function for random sampling of all nodes, and V represents all user and project nodes; furthermore, for the perturbation of the original graph minimization, we set a hyperparameter p that represents the proportion of random sampling, and the nodes extracted are denoted by V_R. After obtaining the randomly extracted nodes, we proceed to simulate user-item interactions that do not exist in real-world scenarios using random noise. These simulated entities are subsequently replaced with the randomly extracted nodes, serving as adversarial interaction noise. Mathematically, this process can be expressed as follows:

$$g^{'} = G(V_R, (\triangle^{'} * p_{noise})) \tag{12}$$

where $\triangle^{'}$ represents the random noise. Furthermore, we set a hyperparameter p_{noise} for control in order to maintain a consistent order of magnitude between the extracted nodes and the interactions simulated by random noise. This hyperparameter allows us to control the scaling of the extracted nodes. G represents the substitution function.

3.2 Mutual Information Maximization

Next, we leverage the principle of mutual information maximization to mitigate the impact of adversarial interaction noise. This is achieved by maximizing the consistency between two augmented graphs. Mutual information quantifies the degree of dependence between two variables, specifically measuring how much information one variable provides about the other variable. Mathematically, mutual information can be expressed using the following formula:

$$I(X,Y) = \sum_{y \epsilon Y} \sum_{x \epsilon X} p(x,y) log(\frac{p(x,y)}{p(x)p(y)}) \tag{13}$$

where $p(x,y)$ denotes the variables x and y joint distribution, the $p(x)$, and $p(y)$ denote the respective marginal distributions, respectively. Unlike general similarity measures, mutual information can capture nonlinear statistical correlations between variables, making it a suitable measure for reflecting true dependence. However, directly optimizing mutual information is challenging; therefore, it is often approximated using lower bounds. Accordingly, we use the general optimization function InfoNce that is formulated as follows:

$$\mathcal{L}_{\text{InfoNCE}} = \sum_{i \in \mathbb{B}} - \log \frac{\exp\left(\frac{Z_i'^T Z_i''}{\tau}\right)}{\sum_{j \in \mathbb{B}} \exp\left(\frac{Z_i'^T Z_j''}{\tau}\right)} \tag{14}$$

where $Z_i^{'}$ and $Z_i^{''}$ denote different augmented representations of the same node that we treat as positive samples. While the consistency of $Z_i^{'}$ and $Z_i^{''}$ should be maximized, the consistency of $Z_i^{'}$ and $Z_j^{'}$, which are negative samples, should be minimized.

3.3 Joint Training

In order to improve the training efficiency, we used a multi-task learning mechanism for the BPR loss task and the InfoNce loss task. The overall loss function is calculated as follows:

$$\mathcal{L} = \mathcal{L}_{BPR} + \lambda\mathcal{L}_{\text{InfoNce}} \tag{15}$$

4 EXPERIMENT

This section presents the extensive experiments conducted to verify and analyze the performance of our model in order to establish the validity of our model.

4.1 Experimental Setup

4.1.1 Dataset

We perform experimental evaluation on three publicly available benchmark datasets (MovieLens-1M, Yelp, and Ta-Feng), and the number of interactions and sparsity of each dataset are shown in the following Table 1.

Table 1 Basic information of the data set

Dataset	User	Item	Interactions	Density
MovieLens-1M	6,040	3,629	836,478	96.1838%
Yelp	16,194	13,062	916,587	99.5666%
Ta-Feng	26,040	15,484	709,356	99.8240%

4.1.2 Hyper-parameter Settings

All the models in our study were developed using the Recbole library[28]. We initialized the model parameters using the Xavier[29] method. The ADAM optimizer[30] was employed for model optimization. The embedding size for users and items was set uniformly to 64, and the batch size was set to 4096. Regarding the dataset scale, we followed the common practice of splitting the interactions into training, validation, and test sets in an 8:1:1 ratio. For evaluating the performance of our models, we used two widely-used metrics: Recall and normalized discounted cumulative gain (NDCG). The top-k recommendations were considered, with k set to both 10 and 20.

4.1.3 Contrast Model

We have compared the following models:

BPR[31]:A matrix decomposition based BPR loss function is used.

NGCF: It is a GNN-based collaborative recommendation framework that aims to capture higher-order connectivity in user?item graphs. It explicitly incorporates collaborative signals from user-item interactions into the embedding process, enabling more effective modeling of user preferences and item characteristics.

LightGCN[32]: It is a graph convolution framework that simplifies the design of GCNs by eliminating the need for a feature transformation matrix and nonlinear activation functions. This simplification reduces training complexity and facilitates efficient representation learning in CF tasks.

SGL[33]: A self-supervised auxiliary task is added to enhance the representation of learning of nodes by self-identification.

HMLET: This is a hybrid model of linear and nonlinear CF based on GCN, which analyzes the linear and nonlinear determinants of embedding propagation.

RGCF[34]: Based on CF of GNNs, the two main technical modules of graph denoising and diversity preservation are designed to achieve noise reduction for unreliable interactions.

4.2 Contrast Analysis

Table 2 shows the overall performance of all the compared models on three different datasets. Bold represents the best performance.

Table 2 Comparison of effects with different models

Dataset	Metric	BPR	NGCF	LightGCN	SGL	HMLET	RGCF	RECF	Improv.
MovieLens-1M	Recall@10	0.1746	0.1772	0.1866	0.1899	0.1876	0.1972	**0.2076**	5.27%
	NDCG@10	0.2388	0.2434	0.2497	0.2531	0.2514	0.2556	**0.2706**	5.87%
	Recall@20	0.2655	0.2654	0.2797	0.2839	0.2795	0.2903	**0.3067**	5.65%
	NDCG@20	0.2505	0.2427	0.2620	0.2657	0.2628	0.2690	**0.2833**	5.32%
Yelp	Recall@10	0.0757	0.0724	0.0842	0.0959	0.0898	0.0937	**0.1055**	10.01%
	NDCG@10	0.0652	0.0622	0.0745	0.0865	0.0785	0.0835	**0.0940**	8.67%
	Recall@20	0.1223	0.1191	0.1334	0.1455	0.1414	0.1460	**0.1572**	7.67%
	NDCG@20	0.0812	0.0783	0.0913	0.1033	0.0962	0.1012	**0.1117**	8.13%
Ta-Feng	Recall@10	0.0498	0.0510	0.0613	0.0724	0.0621	0.0553	**0.0827**	14.2%
	NDCG@10	0.0347	0.0356	0.0417	0.0481	0.0418	0.0383	**0.0552**	14.8%
	Recall@20	0.7170	0.0747	0.0894	0.1015	0.0896	0.0798	**0.1156**	13.9%
	NDCG@20	0.0411	0.0425	0.0499	0.0566	0.0498	0.0455	**0.0649**	14.7%

Through the comparative analysis of the effects of different models, we are able to draw the following conclusions:

• Compared with the performance of BPR, the performance of the recommendation system is substantially improved after the introduction of GNN, indicating that more higher-order information can be mined through graphs.

• Compared with NGCF, LightGCN improves the model significantly by removing two redundant operations?the nonlinear activation function and the feature transformation matrix.

• SGL exhibits superior performance, illustrating the effectiveness of introducing self-supervised auxiliary tasks and contrast learning, where contrast loss leads to improved uniformity of embedding distribution, resulting in improved performance.

• GCN-based HMLET combines linear and nonlinear activation for CF and exhibits better results than LightGCN, illustrating that using nonlinear activation in embedding propagation can lead to enhanced results.

• RGCF further improves the robustness and effectiveness of the model by reducing the effects of noise interactions using the graph denoising module and enriching the denoised graphs using the diversity maintenance module.

• Our models exhibit better results than all models, demonstrating that simulating users or items with noise as adversarial interaction noise in line with the principle of maximizing mutual information can effectively mitigate the effects of adversarial noise in implicit feedback and consequently improve the performance of the models.

4.3 Analysis of Key Parameters

In this section, we investigate the effects of three important hyperparameters in the model. Here we use the experimental setup used in Section 4.1.2.

4.3.1 Effect of the Replacement Ratio p

We use the replacement ratio control parameter p to control the number of replaced nodes. We set p as [0.00, 0.35]. The impact on the model performance is shown in Figure 3, where it is observed that the performance improves as p increases to 0.02 in MovieLens-1M, 0.04 in Ta-Feng, and 0.04 in Yelp, and subsequently, the performance degrades. Therefore, it can be concluded that an excessively large p will increase the corruption of the original image information, thereby reducing the model effect, whereas an exceedingly small p will be ineffective for training.

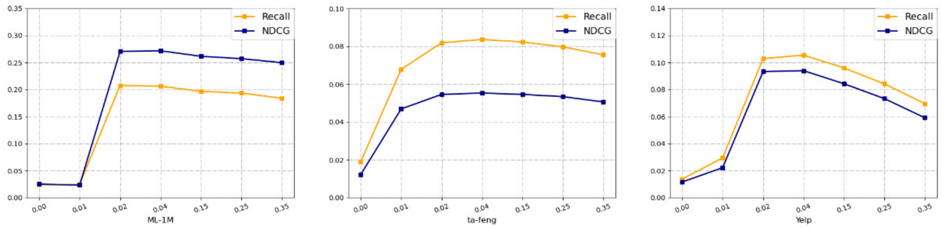

Fig. 3 Substitution of scaling control parameters p effect on model performance

4.3.2 Effect of Temperature τ

The temperature parameters play an important role in the differentiation of difficult and easy samples.Figure 4 shows the different τ parameters on the model performance curves; we set τ to [0.05, 0.40] and observe the change trend. It can be seen that the performance of the model deteriorates as τ increases, which is insufficient in distinguishing difficult samples, and similarly too small τ also makes the model's performance worse.

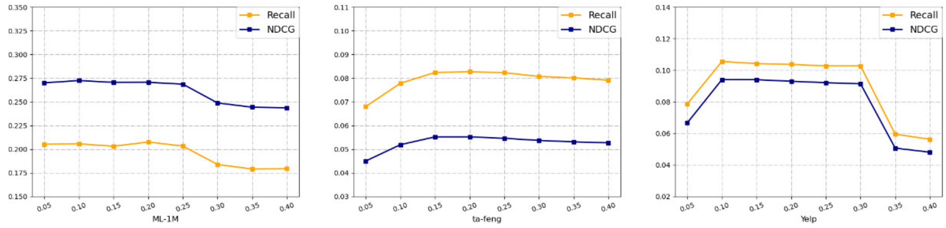

Fig. 4 Temperature parameters τ effect on model performance

4.3.3 Effect of λ

Based on the λ parameter experiments conducted, we obtained the λ influence curve for the model performance, as shown in Figure 5. Suitable performance is demonstrated by setting λ in the range [1e-7, 5e-3]. Furthermore, within a certain range of λ adjustment, the impact on the model performance is relatively small and the robustness of the model is appropriate.

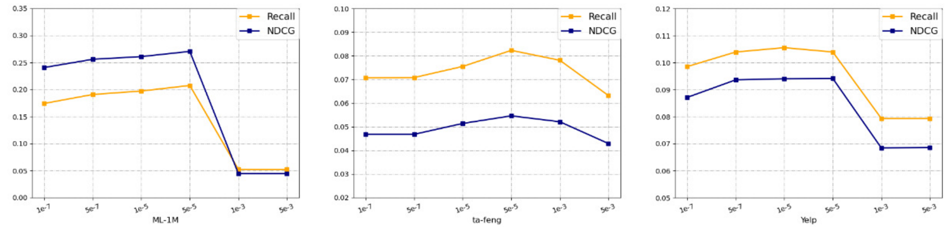

Fig. 5 λ Impact on model performance

4.4 Robustness to Noise Interactions

In this section, the focus is on evaluating the robustness of the model to noisy interactions. To achieve this, a certain percentage (15%) of adversarial examples, which represent negative user-item interactions, are introduced into the training and validation sets of the three datasets. This contamination helps simulate the presence of noise in real-world scenarios. At the same time, the test set remains unaffected to ensure the reliability and accuracy of the evaluation results. The specific approach involves randomly discarding 15% of the existing observed user-item interaction records from the training and validation sets. Additionally, 15% of false interactions are artificially generated and injected as noise into the dataset.

4.4.1 Contrast Model

We compared the models listed in Section 4.1.3. Table 3 shows the performance of all the compared models on each of the three injected-noise datasets. From the analysis,

Table 3 Comparison of the effects of different models on the dataset injected with noise

Dataset	Metric	BPR	NGCF	LightGCN	SGL	HMLET	RGCF	RECF	Improv
MovieLens-1M	Recall@10	0.1502	0.1571	0.1567	0.1582	0.1566	0.1776	**0.1796**	1.12%
	NDCG@10	0.2181	0.2264	0.2263	0.2293	0.2244	0.2315	**0.2462**	6.35%
	Recall@20	0.2287	0.2348	0.2383	0.2413	0.2363	0.2653	**0.2691**	1.43%
	NDCG@20	0.2246	0.2317	0.2329	0.2366	0.2307	0.2453	**0.2560**	4.36%
Yelp	Recall@10	0.0542	0.0559	0.0608	0.0793	0.0677	0.0815	**0.0920**	12.9%
	NDCG@10	0.0475	0.0473	0.0542	0.0704	0.0603	0.0731	**0.0830**	13.5%
	Recall@20	0.090	0.0932	0.0989	0.1235	0.1072	0.1285	**0.1390**	8.17%
	NDCG@20	0.0598	0.0605	0.0671	0.0855	0.0736	0.0892	**0.0990**	11.0%
Ta-feng	Recall@10	0.0412	0.0457	0.0422	0.0564	0.0420	0.0493	**0.0662**	17.3%
	NDCG@10	0.0307	0.0329	0.0312	0.0391	0.0311	0.0340	**0.0448**	14.6%
	Recall@20	0.0601	0.0665	0.0608	0.0809	0.0608	0.0707	**0.0934**	15.5%
	NDCG@20	0.0363	0.0391	0.0367	0.0464	0.0368	0.0404	**0.0529**	14.0%

we can confirm the following:

• In the experiments where 15% noise was injected into the training data, our proposed models demonstrated a remarkable improvement of approximately 10% in performance compared to the best baseline model. This result indicates that our models are robust and effective in handling noisy interactions and can still provide high-quality recommendations even in the presence of such noise.

• However, in the models based on the GNN approach, the performance of all models experienced a significant degradation when evaluated on the contaminated dataset. This outcome can be attributed to the repeated execution of graph convolution operations in GNN, which amplifies the effect of the injected noise. Consequently, the models based on GNN were more susceptible to the adverse impact of noise compared with other models.

• When comparing our proposed model with existing approaches such as LightGCN, SGL, and RGCF, the decline rate in performance for our model was significantly lower than that of LightGCN and SGL. This implies that our model exhibits higher resilience to noise and maintains more stable performance in the presence of noisy interactions. In comparison with RGCF, although our model achieved slightly better overall performance, its decline rate in the contaminated dataset was higher than that of RGCF.

5 RELATED WORK

In this section, we introduce two main tasks related to this study: graph CF and denoising recommendations.

5.1 Graph CF

A graph-based CF method is employed in this study that specifically focuses on modeling user-item interactions as a bipartite graph and utilizes the graph structure for recommendation purposes. GNNs have demonstrated exceptional capability in extracting information from graph-structured data, making them widely adopted in successful recommendation systems. In particular, the LightGCN model enhances the GCN architecture by removing two redundant operations. This simplification not only streamlines the model but also leads to notable performance improvements. Furthermore, self-supervised learning techniques, as seen in SGL, have been applied to graph CF to enhance model generalization. SGL achieves better recommendation performance by introducing auxiliary tasks that improve the learning of node representations. However, despite these advancements, the iterative nature of graph convolutional operations amplifies the impact of adversarial noise, limiting further performance improvement. To address this issue, our proposed model introduces virtual interactions and employs mutual information maximization to mitigate the adverse effects of noise arising from graph convolution operations. This approach allows for additional performance gains in the CF task.

5.2 Denoising Recommendations

Existing recommendation methods commonly utilize implicit feedback data for training, which inevitably contains noise. In recent research, efforts have been made to address the issue of noise in implicit feedback. Two strategies have emerged: resampling and reweighting. These methods, such as ADT dynamic loss and large loss interaction, often require auxiliary information or the introduction of additional functions. In this paper, we propose a denoising scheme that does not rely on explicit filtering techniques. Our approach effectively mitigates the impact of noise without the need for auxiliary information or additional functions.

6 CONCLUSION AND FUTUREWORK

In this study, we address the issue of adversarial interaction noise commonly found in implicit feedback. To tackle this problem, we propose a robustly enhanced CF GNN model that does not rely on explicit noise filtering techniques. Our approach involves simulating users or items that are not present in real situations by introducing random noise as adversarial interaction noise in the hidden feedback. Additionally, we replace these simulated entities with randomly selected nodes, allowing us to obtain different enhanced views. To ensure consistency between the original and enhanced views, we employ the principle of mutual information maximization and utilize InfoNCE as the optimized loss function. Furthermore, we enhance the model's effectiveness by incorporating a multi-task learning strategy.

In future work, we plan to further explore the challenges associated with noise in implicit feedback and investigate various approaches to improve the model's resilience to such disturbances.

Declarations

Ethical Approval This article does not contain any studies with human participants or animals performed by any of the authors. All texts, pictures and tables in the article belong to the original author and follow ethical guidelines, no academic misconduct.

Availability of supporting data The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Competing interests The authors have declared that no competing interests exist

Funding This research was funded by The Zhejiang University-Shandong (Linyi) Modern Agricultural Research Institute Service Local Economic Development Project (Open Project) (Grant No.ZDNY-2021-FWLY02016) and The Natural Science Foundation of Shandong Province (Grant No. ZR2019MA030)

Authors' contributions All authors contributed to the conception and design of the study. Fan Rong completed the experimental analysis and first draft of the manuscript. Review and supervision of the paper was done by Zhenhai Wang, Yuhao Xu, and Yunlong Guo. Data organization, validation work, and visualization were done by Zhiru Wang and by Weimin Li. All authors read and approved the final manuscript.

Acknowledgments I have received a great deal of support and assistance in the writing of this thesis.

I would first like to thank my supervisor, Professor Zhenhai Wang, whose expertise was invaluable in formulating the research questions and methodology.In addition particular, I would like to thank my team members for their patient support and assistance.

References

[1] Wang, Z., Wang, Z., Xu, Y., Wang, X., Tian, H.: Online course recommendation algorithm based on multilevel fusion of user features and item features. Computer Applications in Engineering Education, 469–479 (2022)

[2] Xu, Y.-H., Wang, Z.-H., Wang, Z.-R., Fan, R., Wang, X.: A recommendation algorithm based on a self-supervised learning pretrain transformer. Neural Processing Letters, 1–17 (2022)

[3] Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: Bayesian personalized ranking from implicit feedback. In: Proc. of Uncertainty in Artificial Intelligence, pp. 452–461 (2014)

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

[5] Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large

graphs. Advances in neural information processing systems **30** (2017)

[6] Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)

[7] Wang, X., He, X., Wang, M., Feng, F., Chua, T.-S.: Neural graph collaborative filtering. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 165–174 (2019)

[8] Kong, T., Kim, T., Jeon, J., Choi, J., Lee, Y.-C., Park, N., Kim, S.-W.: Linear, or non-linear, that is the question! In: Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining, pp. 517–525 (2022)

[9] Mao, K., Zhu, J., Xiao, X., Lu, B., Wang, Z., He, X.: Ultragcn: ultra simplification of graph convolutional networks for recommendation. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, pp. 1253–1262 (2021)

[10] Hu, Y., Koren, Y., Volinsky, C.: Collaborative filtering for implicit feedback datasets. In: 2008 Eighth IEEE International Conference on Data Mining, pp. 263–272 (2008). Ieee

[11] Lu, H., Zhang, M., Ma, S.: Between clicks and satisfaction: Study on multi-phase user preferences and satisfaction for online news reading. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, pp. 435–444 (2018)

[12] Wen, H., Yang, L., Estrin, D.: Leveraging post-click feedback for content recommendations. In: Proceedings of the 13th ACM Conference on Recommender Systems, pp. 278–286 (2019)

[13] Chen, J., Dong, H., Wang, X., Feng, F., Wang, M., He, X.: Bias and debias in recommender system: A survey and future directions. ACM Transactions on Information Systems **41**(3), 1–39 (2023)

[14] Ding, J., Feng, F., He, X., Yu, G., Li, Y., Jin, D.: An improved sampler for bayesian personalized ranking by leveraging view data. In: Companion Proceedings of the The Web Conference 2018, pp. 13–14 (2018)

[15] Ding, J., Yu, G., He, X., Feng, F., Li, Y., Jin, D.: Sampler design for bayesian personalized ranking by leveraging view data. IEEE transactions on knowledge and data engineering **33**(2), 667–681 (2019)

[16] Gantner, Z., Drumond, L., Freudenthaler, C., Schmidt-Thieme, L.: Personalized ranking for non-uniformly sampled items. In: Proceedings of KDD Cup 2011, pp. 231–247 (2012). PMLR

[17] Wang, J., Yu, L., Zhang, W., Gong, Y., Xu, Y., Wang, B., Zhang, P., Zhang, D.: Irgan: A minimax game for unifying generative and discriminative information retrieval models. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 515–524 (2017)

[18] Yu, W., Qin, Z.: Sampler design for implicit feedback data by noisy-label robust learning. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 861–870 (2020)

[19] Hu, K., Li, L., Xie, Q., Liu, J., Tao, X.: What is next when sequential prediction meets implicitly hard interaction? In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, pp. 710–719 (2021)

[20] Wang, Y., Xin, X., Meng, Z., Jose, J.M., Feng, F., He, X.: Learning robust recommenders through cross-model agreement. In: Proceedings of the ACM Web Conference 2022, pp. 2015–2025 (2022)

[21] Wang, W., Feng, F., He, X., Nie, L., Chua, T.-S.: Denoising implicit feedback for recommendation. In: Proceedings of the 14th ACM International Conference on Web Search and Data Mining, pp. 373–381 (2021)

[22] Gao, Y., Du, Y., Hu, Y., Chen, L., Zhu, X., Fang, Z., Zheng, B.: Self-guided learning to denoise for robust recommendation. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1412–1422 (2022)

[23] Lu, H., Zhang, M., Ma, W., Wang, C., Xia, F., Liu, Y., Lin, L., Ma, S.: Effects of user negative experience in mobile news streaming. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 705–714 (2019)

[24] Yi, X., Hong, L., Zhong, E., Liu, N.N., Rajan, S.: Beyond clicks: dwell time for personalization. In: Proceedings of the 8th ACM Conference on Recommender Systems, pp. 113–120 (2014)

[25] Kim, Y., Hassan, A., White, R.W., Zitouni, I.: Modeling dwell time to predict click-level satisfaction. In: Proceedings of the 7th ACM International Conference on Web Search and Data Mining, pp. 193–202 (2014)

[26] Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018)

[27] Koren, Y.: Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 426–434 (2008)

[28] Zhao, W.X., Mu, S., Hou, Y., Lin, Z., Chen, Y., Pan, X., Li, K., Lu, Y., Wang, H., Tian, C., *et al.*: Recbole: Towards a unified, comprehensive and efficient framework for recommendation algorithms. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, pp. 4653–4664 (2021)

[29] Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, pp. 249–256 (2010). JMLR Workshop and Conference Proceedings

[30] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)

[31] Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: Bpr: Bayesian personalized ranking from implicit feedback. arXiv preprint arXiv:1205.2618 (2012)

[32] He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., Wang, M.: Lightgcn: Simplifying and powering graph convolution network for recommendation. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 639–648 (2020)

[33] Wu, J., Wang, X., Feng, F., He, X., Chen, L., Lian, J., Xie, X.: Self-supervised graph learning for recommendation. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 726–735 (2021)

[34] Tian, C., Xie, Y., Li, Y., Yang, N., Zhao, W.X.: Learning to denoise unreliable interactions for graph collaborative filtering. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 122–132 (2022)