Collaborative Graph Filtering Recommendation Algorithm Based on Public Interaction Ratio and Structural Neighborhood Contrastive Learning

Jianqiao Liu, Qigang Liu, Gang Sun*

***Abstract*—Graph Neural Network (GNN) directly inherits the captured collaborative information and there is no determination of whether the captured collaborative information benefits user preferences. Therefore, this paper propose the RCGCN model to solve that issue. First, we detect the level of interaction between a specific neighbor of the target node and other neighbors by the common interaction ratio (CIR), and the experimental results show that the collaboration information propagated by neighbors with higher CIR is more favorable to user preferences. Second, we introduce structural neighborhood contrastive learning to mine the association between users and items by comparing them with their structural neighbors. Finally, we use a joint training approach to optimize the learning objectives. The experimental results indicated that the RCGCN model is superior than the current benchmark models in both Recall@20 and NDCG@20 evaluation metrics.**

***Keywords:* graph neural network; public interaction ratio; contrastive learning; collaborative filtering; recommendation algorithm**

I. Introduction

Facing the huge amount of different information on the Internet, how to provide personalized recommendations to users has been a hot research topic. Collaborative Filtering (CF) [1], as a basic technique, learns the accurate representation of consumers and items and provides personalized recommendations for users based on the learned implicit feedback. The performance of CF can be improved by graph neural network (GNN) [2][3], which models interaction data as user-item bipartite graphs and learns embedded representations for recommendations. Wang et al. [2] constructed a graph neural filtering (NGCF) using graph convolutional networks (GCN). He et al. [3] designed LightGCN to simplify NGCF by retaining only the neighborhood aggregation in GCN to improve the applicability of recommendations.Wu et al. [4] designed data argumentation operators via SGL and constructed contrastive targets to enhance the performance of GCN. Nevertheless, although existing recommendation models capture the assistance information between nodes through GCN, there is no good explanation of whether the collaboration information they capture benefits user preferences.

To solve the above problem, we propose the RCGCN algorithm. Specifically, this study established a user-item bipartite graphs, and then we propagate embeddings from the l-order neighbors. To determine whether this captured collaborative information is helpful for user preferences, we use the common interaction ratio (CIR) to reflect the interaction effect of a specific neighbor of a target node and its other neighbors. On the other hand, to fully explore the potential relationships between nodes, the contrastive learning method was applied to enhance user and item representation by comparing nodes with their structural neighbors to capture potential relationships between pairs of nodes directly. The experiments was performed on four publicly datasets, MovieLens-1M, Yelp, Amazon-Books, and Gowalla [6]. And it was found that RCGCN is superior than the benchmark methods in both Recall@20 and NDCG@20 metrics [3].

The main works of present research as follows:

(1) The level of interaction in a specific neighbor of a target node and its other neighbors is measured by the public interaction ratio (CIR). It indicates that the collaborative information transmitted by neighbors with higher CIR is more favorable for user preferences.

(2) For the potential relationships in the interaction graph, we compare the target node with its structural neighbors to mine the potential relationships to leverage the collaboration information.

(3) The experiments conducted on four datasets indicate that RCGCN is superior than the benchmark models on two metrics, Recall@20 and NDCG@20.

II. Preliminary

A. Collaborative filtering effect analysis

With the development of GNN, GNN-based collaborative filtering methods [1][2] generate user-item interaction data into bipartite graphs $G(\cdot)=(V,E)$, in which the node set $V=U\cap I$ includes U and I. We only considered implicit user-item interactions [3] and expressed them as edges E, in which e_{pq} denotes the edge between nodes p and q . $A\in\{0,1\}^{|V|\times|V|}$ representing the adjacency matrix. $A_{pq}=1$ indicates an interaction between nodes p and q, otherwise $Apq=0$. N_p^l is the set of lth order neighbors of node

Manuscript received August 3, 2023; revised June 18, 2024.

The work was supported in part by the National Natural Science Foundation of China (61906044), and supported by the Natural Science Foundation of the Anhui Higher Education Institutions of China (KJ2020ZD48), and in part by Research and Innovation Team Project of Fuyang Normal University (TDJC2021008).

Jianqiao Liu is a laboratory technician of Fuyang Normal University, Fuyang 236037, China (e-mail: liujianqiaowgy@163.com).

Qigang Liu is a postgraduate student of Fuyang Normal University, Fuyang 236037, China (e-mail: liu1310852887@163.com).

Gang Sun is a professor of Fuyang Normal University, Fuyang 236037, China (e-mail: ahfysungang@163.com).

p, P_{pq}^{l} refer to the set of shortest paths of nodes p and q at path length l.

B. Base model

LightGCN [3] exhibits high property with a lightweight design, and we use it to express the GCN-based recommendation model. Firstly, $e_u^{(0)}$, $e_i^{(0)}$ refer to the ID embedding of user *u*, ID embedding of item *i*. Secondly, the graph convolution operation of LightGCN is shown in formula (1) and formula (2) as follows:

$$e_u^{(l)}=\sum_{j\in N_u}\frac{1}{\sqrt{|d_u|}\sqrt{|d_j|}}e_j^{l-1} \tag{1}$$

$$e_i^{(l)}=\sum_{v\in Ni}\frac{1}{\sqrt{|d_i|}\sqrt{|d_v|}}e_v^{l-1} \tag{2}$$

where $\forall l \in \{0,\ldots,L\}$, $e_u^{(l)}$ and $e_i^{(l)}$ represent user u and item i propagated through l convolution layers to represent, respectively, d_u represent the set of items interacting with user u. $\frac{1}{\sqrt{|d_u|}\sqrt{|d_i|}}$ is the symmetric normalization that avoids the increase in the embedding size with increasing layers of convolution results in the increase in computational complexity.

III. Methodology

A. System Architecture

There are unreliable interactions in bipartite graphs, and these interactions do not reflect the behavioral preferences of users [14][15]. Existing GCN models [2][3][4][6] use direct inheritance for the captured collaboration information and do not explain whether the captured collaboration information benefit user preferences. GCN may capture harmful collaboration information from these unreliable interactions according to the existing messaging mechanism [16]. Therefore, we propose RCGCN, as shown in Fig. 1, which uses the common interaction ratio (CIR) to assess the interaction degree between a node's specific neighbors and its other neighbors. Higher CIR values of neighbor collaboration information can better reflect user preferences. At the same time, we use structural neighborhood contrastive learning to address the data sparsity between users and items. We also form contrastive pairs of each user and item, which improves the performance of RCGCN.

B. Neighborhood Contrastive Learning

To learn the embedding representation of users and items while considering the data sparsity of them, we use contrastive learning to derive self-supervised information [17][18][27][28]. Existing graph collaborative filtering models are trained using user-item pairs [3], but do not directly capture the underlying features between the two. Therefore, we compare each user and item with its structural neighbors, where structural neighbors are nodes that are connected via higher-order paths. The information propagation over the graph using the GNN model also aggregates information. We establish the relationship between a user (item) and its structural neighbors through these representations and then use the user's embedding of the GNN at even layers as positive pairs. According to InfoNCE [18], this paper propose the structural contrastive learning. As shown in formula (3):

$$L_S^U=\sum_{u\in U}-ln\frac{exp((e_u^{(k)}\cdot e_u^{(0)}/\tau))}{\sum_{v\in U}exp((e_u^{(k)}\cdot e_v^{(0)}/\tau))} \tag{3}$$

where e^0 represents the initial features of the user, $e_u^{(k)}$ is the normalized the kth layer GNN output, τ refer to the temperature hyperparameter of softmax. The structured contrastive learning loss function for the item L_s^I is shown in formula (4):

$$L_S^I=\sum_{i\in I}-ln\frac{exp((e_i^{(k)}\cdot e_i^{(0)}/\tau))}{\sum_{j\in I}exp((e_i^{(k)}\cdot e_j^{(0)}/\tau))} \tag{4}$$

The complete structural contrastive learning function is a weighted sum of formula (3) formula (4), as shown in formula (5):

$$L_S=L_S^U+\alpha L_S^I \tag{5}$$

where α is the hyperparameter of the two-loss weights in the balanced structure learning learning.

C. Common interaction ratio (CIR)

Existing GCN models do not provide a good explanation for whether the captured collaboration information contribute to obtaining user preferences [19][20][21]. To address this problem, we detected the interaction level in a particular neighbor *j* of a target user *u* and other item neighbors of *u* via the public interaction ratio (CIR). As shown in Fig. 2: user *u* interacts directly with its first-order neighbors j_1, j_2, user *u* may have more interaction paths with other higher-order neighbors of *u* through j_1, and we believe that the information propagated by j_1 is more representative of user *u* preferences. Therefore, we calculate the interaction weight size $\varphi_u^{\hat{L}}(j)$ of item *j* with user *u*. $\varphi_u^{\hat{L}}(j)$ is defined as the common interaction ratio (CIR) of item *j* with other items of target user *u* in the set N_u^l through paths with path length less than $2l$. As shown in formula (6):

$$\varphi_u^{\hat{L}}(j)=\frac{1}{|N_u^1|}\sum_{i\in N_u^1}\sum_{l=1}^{\hat{L}}\alpha^{2l}\sum_{k\in P_{ji}^{2l}}\frac{1}{f(N_k^1)} \tag{6}$$

where $j \in N_u^1$, $\forall u \in U$, $\{N_k^1 \mid k \in P_{ji}^{2l}\}$ denotes the set of first-order neighbors of node k along the path P_{ji}^{2l} with path length $2l$ from node j to i. f is the normalization function used to distinguish the weight of paths P_{ji}^{2l}, and α^{2l} represents the importance of the path with length $2l$.

D. RCGCN model

The RCGCN model is used to enhance the collaboration information passed from neighboring nodes with high CIR to the target node, for which we calculate the weights of edges as shown in formula (7). Secondly, we use structural neighborhood contrastive learning to mine the potential relationships between nodes to mitigate the data sparsity problem.

$$\varphi_{ij}=\begin{cases}\varphi_i(j) & if \quad A_{ij}>0\\ 0 & if \quad A_{ij}=0\end{cases} \tag{7}$$

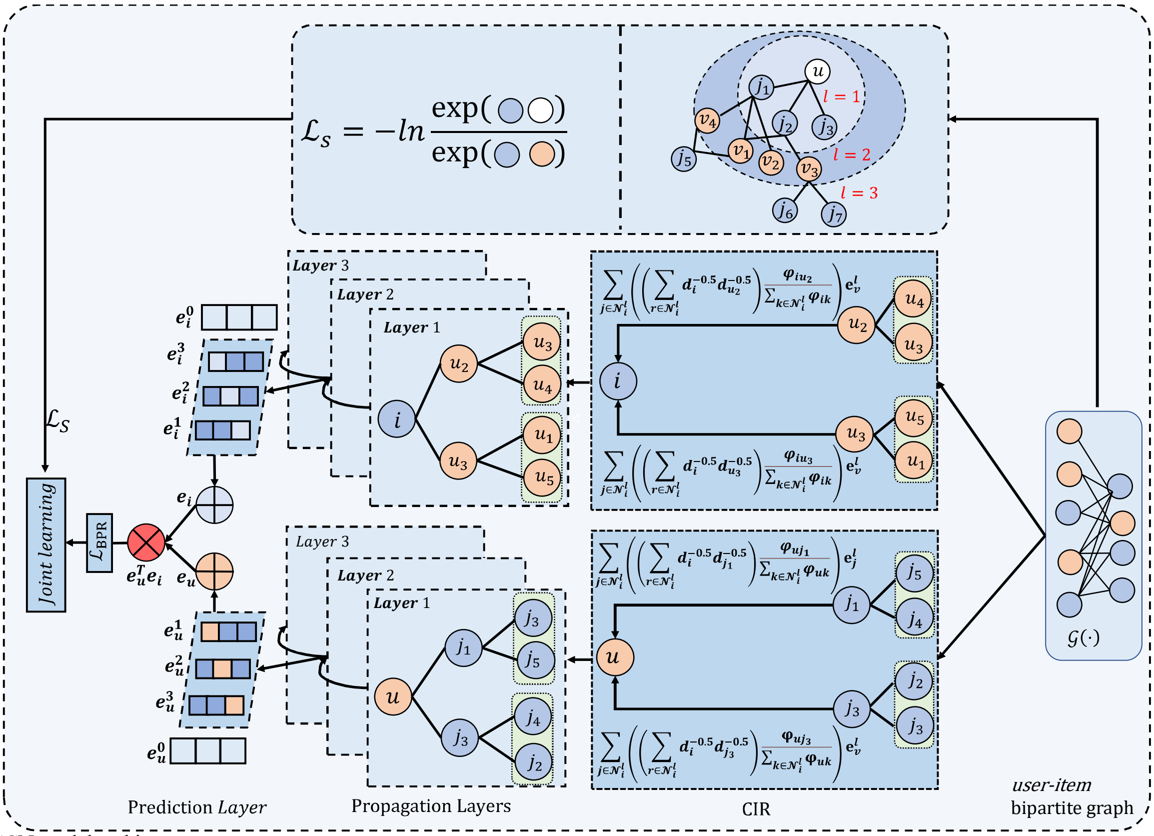

Fig. 1. RCGCN model architecture

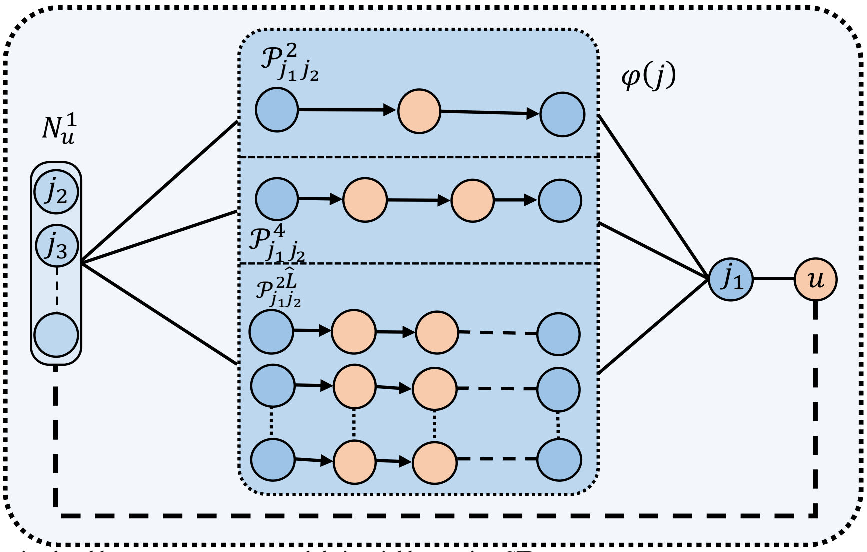

Fig. 2. Calculation of interaction level between target users and their neighbors using CIR

where $\varphi_i(j)$ is the size of the CIR value of the neighbor j of node i. We define the graph convolution operation of RCGCN as shown in formula (8) and formula (9):

$$e_i^{l+1} = \sum_{j \in N_i^1} \left(\left(\sum_{r \in N_i^1} d_i^{-0.5} d_r^{-0.5} \right) \frac{\varphi_{ij}}{\sum_{k \in N_i^1} \varphi_{ik}} \right) e_j^l \quad (8)$$

$$e_u^{l+1} = \sum_{v \in N_u^1} \left(\left(\sum_{h \in N_u^1} d_u^{-0.5} d_h^{-0.5} \right) \frac{\varphi_{uv}}{\sum_{k \in N_u^1} \varphi_{uk}} \right) e_v^l \quad (9)$$

The RCGCN all-layer propagation embedding is aggregated by mean pooling as shown in formula (10) and formula (11):

$$e_u = \frac{1}{(L+1)} \sum_{l=0}^{L} e_u^l \quad (10)$$

$$e_i = \frac{1}{(L+1)} \sum_{l=0}^{L} e_i^l \quad (11)$$

The inner product was applied to predict the likelihood of interaction in user u and item i. The formula (12) is shown below:

$$\hat{y}_{ui} = e_u^T e_i \quad (12)$$

In training phase, for every user-item pair (u,i), the RCGCN model uses a random sampling of an item i that user u has never interacted with defined as negative item i^-, and the triple (u,i,i^-) is formed. The triples form a set of training triples o[3]. We finally use the BPR loss function [8] to perform the optimization. As shown in the following formula (13):

$$L_{BPR} = \sum_{(u,i,i^-)\in O} -ln\,\sigma(y_{ui} - y_{ui^-}) \quad (13)$$

In which $\sigma(\cdot)$ refer to the Sigmoid function, and after obtaining the loss functions L_S and L_{BPR} for structural contrastive learning, we combine them for joint learning [4] as shown in the following formula (14):

$$L = L_{BPR} + \lambda_1 L_S + \lambda\|\theta\| \quad (14)$$

where λ_1 is the hyperparameter controlling the contrastive loss and λ is the L_2 regularization factor.

IV. Experimental results and analysis

A. Experimental dataset

To assess the performance of the RCGCN model, this paper performed experiments on four benchmark datasets: MovieLens1M, Yelp, Amazon Books, and Gowalla, as shown in Table 1. Moreover, we filtered users with below 10 interactions and items with below 10 occurrences. For every dataset,80% of the interaction data was selected to train the model and 10% of the interaction data was applied to verify it, and the rest 10% of the interaction data are used for performance comparison[22][23]. The statistics of the data sets is shown in Table I.

Table I
Statistics of the data sets

Datasets	Users	Items	Interaction
ML-1M	6040	3629	836478
Yelp	45478	30709	1777765
Amazon-Books	58145	58052	2517437
Gowalla	29859	40989	1027464

B. Evaluation metrics and parameter settings

To assess the RCGCN model, we use two indexes [3] Recall@K and NDCG@K, and K is set to 20. In addition, the Adam [24] optimizer was applied to improve the model and with batch size of 4096[25]. The hyperparameter λ_1 in the model is 1e-4, and the value range of τ is [0.04-0.13], and the value range of α is [0.1-2]

C. benchmark algorithms

To prove the present proposed algorithm, we compare it with other benchmark algorithms, and the results are shown in Table II.

(1) BPRMF [8]: The algorithm learns the potential representation between users and items by optimizing the BPR loss using the matrix factorization(MF) framework.

(2) NeuMF [1]: The algorithm learns user and item representations using multilayer perceptron .

(3) NGCF [2]: The algorithm develops higher-order connections in users and items by using multilayer GCNs for modeling collaborative signals in user-item interactions.

(4) GCCF [16]: The algorithm propagates higher-order information between users and items based on the bipartite graphs between users and projects.

(5) DGCF [6]: The algorithm introduces separation learning into filtering to consider users' different interests, and proposes intent-aware interaction graphs to model the multi-intent distribution of interaction.

(6) LightGCN [3]: The algorithm simplifies NGCF by removing feature transformations and nonlinear activation functions from GCN.

(7) SGL [4]: The algorithm applies random structural enhancements, like node descent, edge descent and random wandering to generate contrast views for graph enhancement in this process.

D. Performance Analysis

Table II shows the performance of the RCGCN algorithm compared with other algorithms on the four datasets. We can find that:

(1) Compared with the conventional algorithm (BPRMF), the GNN-based model encodes the higher-order information representation of the bipartite graph with superior performance.

(2) According to comparison result, the LightGCN model is superior on most datasets, and LightGCN simplifying the GCN framework and improving its robustness.

(3) The GCCF model does not perform as well as the NGCF model on the ML-1M dataset, probably because directly mappin such bipartite graph to the item graph results in overlap and indistinguishability between users and items.

(4) The DGCF model has worse recommendation performance than LightGCN on sparse datasets. When limiting dimensions, we speculate that the DGCF model cannot fully utilize the characteristic information of the dataset.

(5) The SGL model performs better than other algorithms on all four datasets, indicating that using contrastive learning is beneficial for improving recommendation performance. However, SGL did not consider potential relationships and only attempted to enhance user-item interaction modeling by contrastive learning.

(6) Our proposed RCGCN algorithm outperforms these benchmark models. We optimize the embedding of nodes and reduce the distance between neighbors by connecting neighbors through edges with higher CIR. Also, we learn more reliable node representations by mining potential neighbor relationships through contrastive learning and enhancing recommendation performance using higher-order connections of interaction graphs.

V. Further Exploration of the RCGCN Model

In present part, we further explore the relevant details about the RCGCN model to illustrate its effectiveness. We report the results through two datasets, ML-1M, and Yelp, and similar results are observed for other datasets.

A. Ablation experiments

To clarify the role of each module in the model, we designed the following models for comparison with RCGCN.

(1) R-GCN: remove the public interaction ratio module in the model that utilizes collaborative information beneficial to user ranking.

TABLE II
PERFORMANCE COMPARISON OF DIFFERENT RECOMMENDED MODELS

Dataset	Metric	BPRMF	NeuMF	NGCF	GCCF	DGCF	LightGCN	SGL	RCGCN
ML-1M	Recall@20	0.2714	0.2520	0.2741	0.2759	0.2779	0.2796	0.2848	0.3123
	NDCG@20	0.2569	0.2400	0.2607	0.2617	0.2615	0.2620	0.2649	0.2835
Yelp	Recall@20	0.1043	0.0885	0.1026	0.1053	0.1135	0.1163	0.1288	0.1357
	NDCG@20	0.0580	0.0486	0.0567	0.0575	0.0641	0.0652	0.0739	0.0836
Amazon-Books	Recall@20	0.1043	0.0823	0.0978	0.0991	0.1128	0.1206	0.1331	0.1376
	NDCG@20	0.0580	0.0447	0.0537	0.0545	0.0640	0.0689	0.0777	0.0823
Gowalla	Recall@20	0.0956	01535	01755	0.1626	01829	0.1976	0.2084	0.2125
	NDCG@20	0.0537	0.0873	0.1013	0.0940	0.1066	0.1152	0.1225	0.1256

(2) C-GCN: remove the neighborhood contrastive learning module that exploits potential user relationships and mitigates data sparsity.

The experimental results were exhibited in Fig. 3. On the datasets of ML-1M and Yelp, the removal of each module caused a degradation in the model performance. C-GCN performs slightly better than R-GCN on ML-1M, which we believe is because ML-1M has a relatively small amount of data and the neighbors are closer to each other, so it is less likely to lose information when capturing the information of nodes through CIR and propagating it to the target nodes. And R-GCN works better than C-GCN on the Yelp dataset, where the data sparsity is relatively large. This is because the strategy of mining potential relationships between nodes and solving data sparsity using structural neighborhood contrastive learning can improve the performance of the RCGCN model.

B. Adding edges based on CIR values

To illustrate that by connecting edges with higher CIR, there are more neighbors, which optimizes node embedding and reduces the distance between neighbors, which is more favorable for capturing collaborative information beneficial to user ranking. The test results were exhibited in Fig. 4. All edges from the training set were removed to create a bipartite graph without edges, then increase the proportion of edges and retrain the two embeddings according to the CIR values of the nodes. We can find that both evaluation metrics improve as the ratio of edges increases. This further indicates that neighbor nodes with more interactions are apt to have higher interactions with the predicted node.

C. Effect of order $\hat{\boldsymbol{L}}$

In RCGCN, to determine the impact of different orders on the model, we chose order $\hat{\boldsymbol{L}}$ of 1, 2, and 3 to test the effectiveness of the improved module. As shown in Fig. 5, it was found that as the order increases, the interaction between each node and its neighbors also increases. This indicates that the more useful information we capture is more helpful to user preferences. Furthermore, we can see that our model performs the best in the 3 orders. Further increasing the order will caused larger model complexity and lower efficiency. Therefore, we need to make a good balance .

D. Effect of coefficients τ *and* α

The temperature coefficients τ [29][30] defined by formula (3) and formula (4) also play an essential role in the contrastive learning. We vary the value of τ in the range of 0.04 to 0.1, and the best recommendation performance of RCGCN model is achieved for τ= 0.105 in Fig. 6. Also, it was found that too large τ values affect the recommendation results. In addition, the coefficient α defined in formula (5) is used to balance the two losses. To analyze the impact, we set α values in scope of 0.1 to 2 and report the experimental results in Fig. 7. This indicates that suitable value of α can improve the RCGCN model. The best-recommended condition of the RCGCN model is reached when the parameter α is set to 1.2.

VI. RELATED WORK

A. Collaborative filtering

Early CF approaches used matrix factorization techniques [7][8] to capture collaborative effects by optimizing the embedding on historical interactions, with the great success of deep neural networks in nonlinear representation learning. NeuCF [1] was proposed to model nonlinear interactions between users and items. Although this approach is successful, it treats the interaction as a separate instance and fails to capture suboptimal representations of users and items [2]. NGCF and LightGCN use graph convolution to aggregate messages from local neighborhoods and inject collaboration signals directly into user-item embeddings. In addition, some studies have proposed that not all captured collaboration signals improve user ranking. For example, Fan et al. [19] introduced a graph trend filtering network framework GTN. Chen et al. [20] designed structured GCN to improve the performance of GCNs by exploiting sparsity and low-rank graph structure properties. Our RCGCN model measures collaborative information from neighbors via CIR, indicating that capturing useful collaborative information better satisfies user preferences.

B. Contrastive learning

It aims to learn optimized representations by comparing local and global representations of training samples or comparing representations of the same samples in various views. This was important in computer vision and AI, etc. [9][10][11]. Moreover, contrastive learning has also recently been introduced to recommendation systems. Contrastive learning is used to learn accurate user and item representation. Zhou et al. [12] proposed maximizing the interaction information between sequences of items of different forms or granularities to enhance item feature learning and improve sequential recommendation tasks. Xie et al. [13] proposed to increase supervised learning with contrastive learning in a pre-trained manner by comparing the identical sequences of items in different views generated by other enhancement methods to extract self-supervised signals. Furthermore, Ma et al. [14] proposed self-supervision by sequence to sequence training strategy in the latent space to simultaneously extract additional supervised signals to mine the user's intention information. Then, most graph-based method ignore potential neighbor association in users and items. In present research, we model the potential neighbor relationships by contrastive learning to improve recommendation performance further.

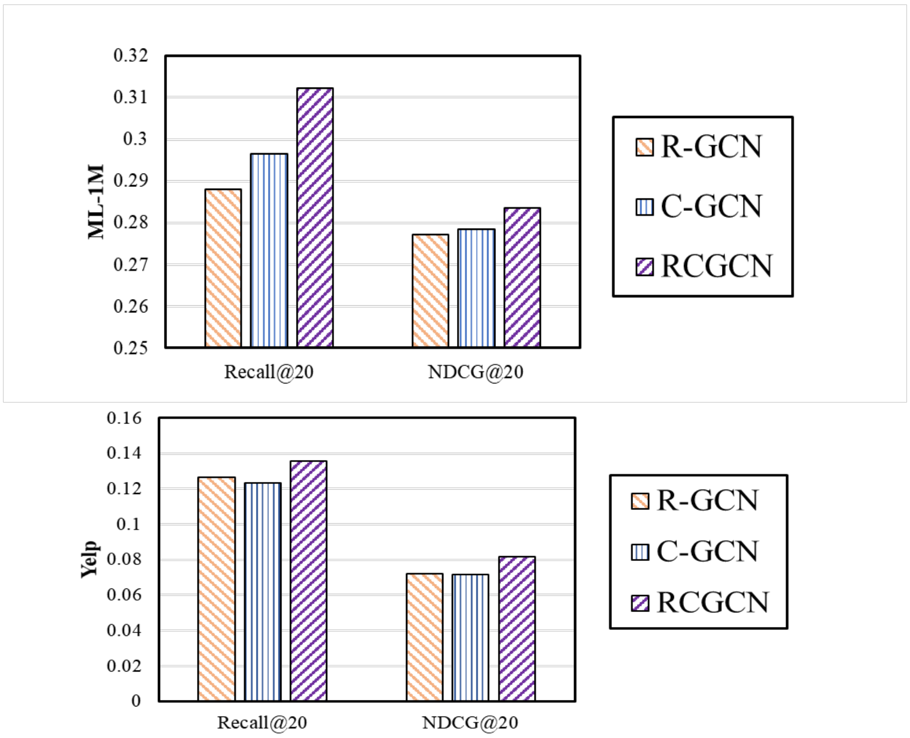

Fig. 3. RCGCN ablation experiments

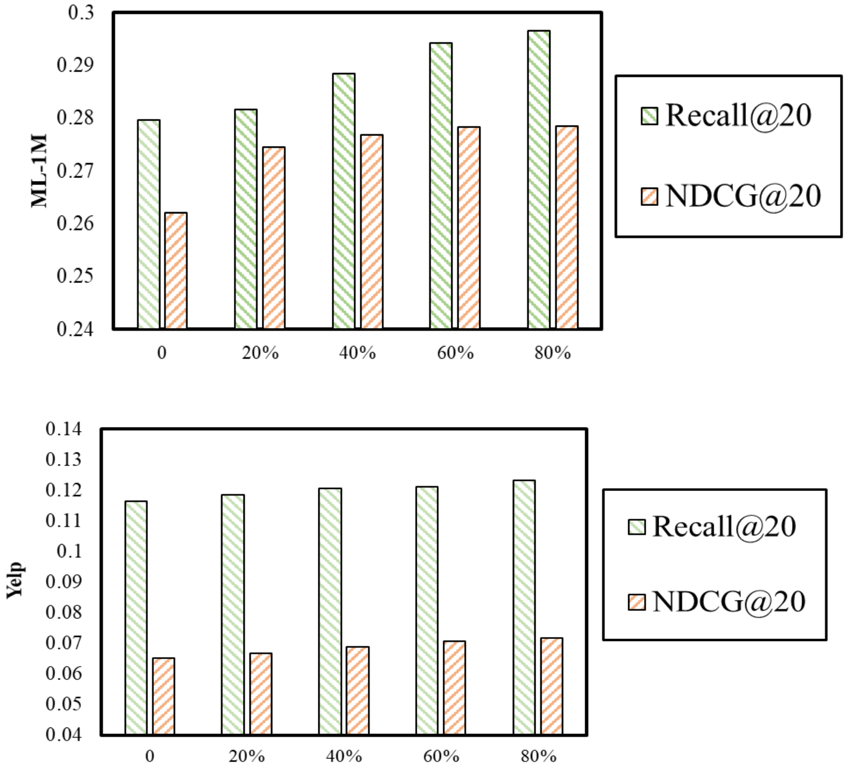

Fig. 4. Proportion of added edges according to CIR values

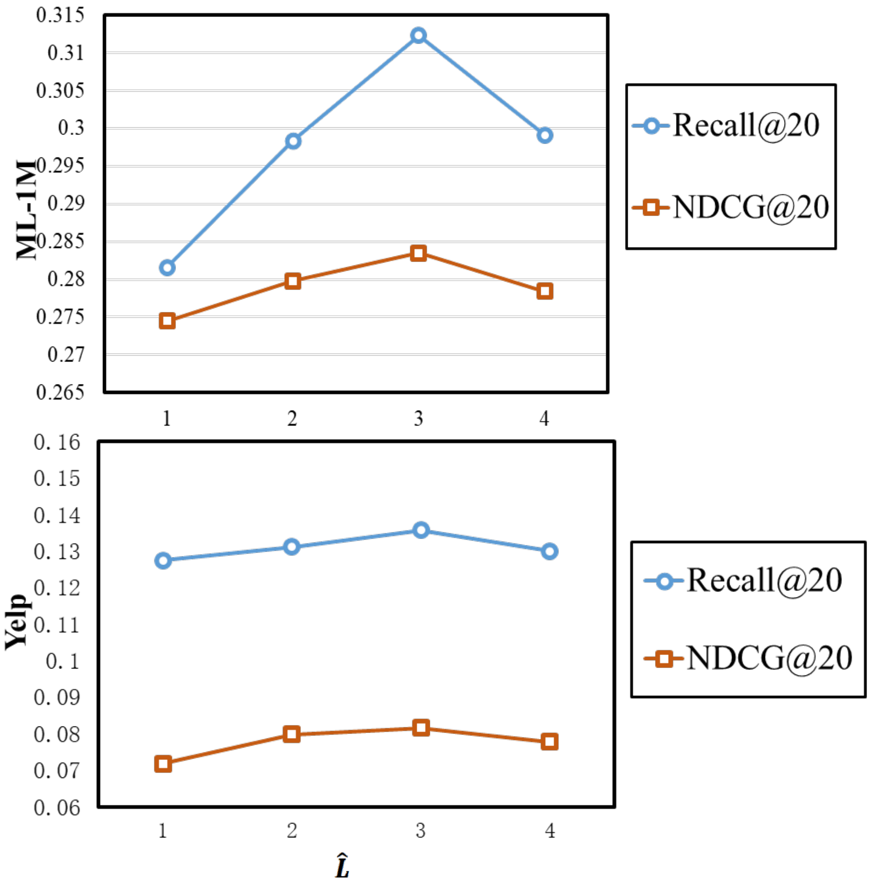

Fig. 5. Effect of parameter $\hat{L}$ in RCGCN

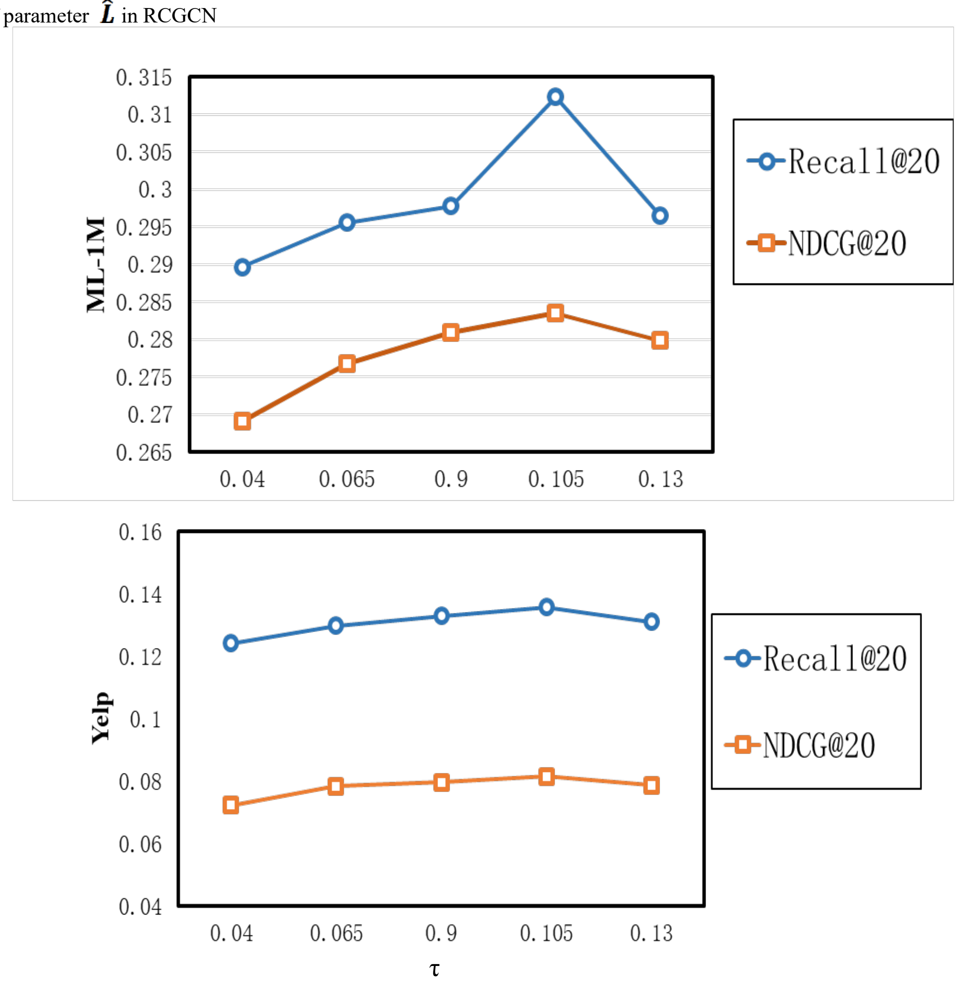

Fig. 6. Effect of parameter τ in RCGCN

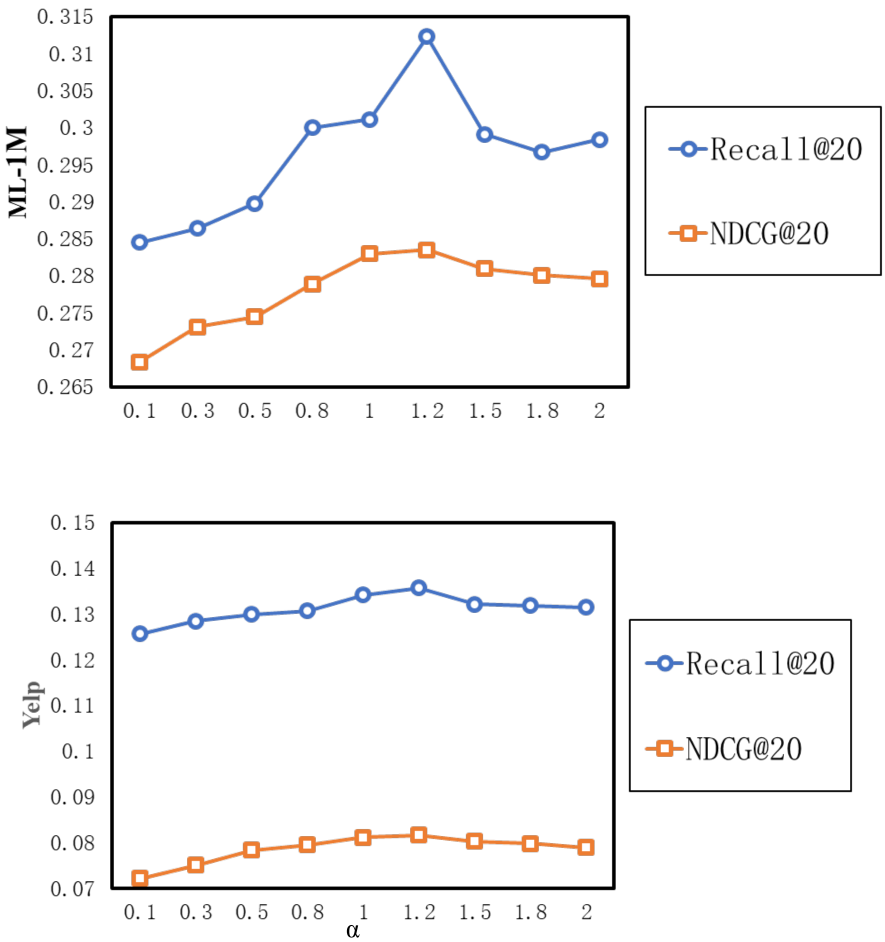

Fig. 7. Effect of parameter α in RCGCN

VII. Conclusions and Future Work

Since the existing graph collaborative filtering models do not analyze the captured collaboration signals, it is unclear whether the information obtained through convolutional operations benefits user preferences. To solve this issue, we propose the RCGCN model, which captures collaborative information that is helpful to user preferences by calculating the value of CIR to filter out neighbors with more associations with the whole neighborhood. Meanwhile, we use structural neighborhood contrastive learning to mine the association between users and items and solve the data sparsity problem. Finally, we will jointly train the two modules to better provide personalized recommendations to users. In future research, we plan to further improve recommendation accuracy to meet the personalized requirements of users.

References

[1] He X, Liao L, Zhang H, et al. "Neural Collaborative Filtering", Proceedings of the 26th International Conference on World Wide Web, 2017, pp173-182.

[2] Wang X, He X, Wang M, et al. "Neural graph Collaborative Filtering", Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2019, pp165-174.

[3] He X, Deng K, Wang X, et al. "Lightgcn: Simplifying and Powering Graph Convolution Network for Recommendation", Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020, pp639-648.

[4] Wu J, Wang X, Feng F, et al. "Self-supervised Graph Learning for Recommendation", Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2021, pp726-735.

[5] Wang Y, Zhao Y, Zhang Y, et al. "Collab-oration-Aware Graph Convolutional Net-work for Recommender Systems", Proceedings of the ACM Web Conference, 2023, pp91-101.

[6] Wang X, Jin H, Zhang A, et al. "Disentangled Graph Collaborative Filtering" Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020, pp1001-1010.

[7] Koren Y, Bell R, Volinsky C. "Matrix Factorization Techniques for Recommender Systems", Computer, 2009, vol. 42, no.8, pp30-37.

[8] Rendle S, Freudenthaler C, Gantner Z, et al. "BPR: Bayesian Personalized Ranking from Implicit Feedback", arXiv preprint arXiv:1205.2618, 2012.

[9] Wu Z, Xiong Y, Yu S X, et al. "Unsupervised Feature Learning via Non-parametric Instance Discrimination", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018, pp3733-3742.

[10] Chen T, Kornblith S, Norouzi M, et al. "A Simple Framework for Contrastive Learning of Visual Representations", Proceedings of International Conference on Machine Learning, 2020, pp1597-1607.

[11] He K, Fan H, Wu Y, et al. "Momentum Contrast for Unsupervised Visual Representation Learning", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2020, pp9729-9738.

[12] Zhou K, Wang H, Zhao W X, et al. "S3-r-ec: Self-supervised Learning for Sequential Recommendation with Mutual Information Maximization", Proceedings of the 29th ACM International Conference on Information and Knowledge Management, 2020, pp1893-1902.

[13] Xie X, Sun F, Liu Z, et al. "Contrastive Pre-training for Sequential Recommendation", arXiv preprint arXiv:2010.14395, 2020.

[14] Ma J, Zhou C, Yang H, et al. "Disentangled Self-supervision in Sequential Ecommenders", Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2020, pp483-491.

[15] Wang X, Jin H, Zhang A, et al. "Disentangled Graph Collaborative Filtering", Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020, pp1001-1010.

[16] Sun J, Zhang Y, Ma C, et al. "Multi-graph Convolution Collaborative Filtering", Proceeding of IEEE International Conference on Data Mining 2019, pp1306-1311.

[17] Oord A, Li Y, Vinyals O. “Representation Learning with Contrastive Predictive Coding”, .arXiv preprint arXiv:1807.03748, 2018.
[18] You Y, Chen T, Sui Y, et al. “Graph Contrastive Learning with Augmentations”, Advances in Neural Information Processing Systems, 2020, vol.33, pp5812-5823.
[19] Fan W, Liu X, Jin W, et al. “Graph Trend Filtering Networks for Recommendations”, Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2022, pp112-121.
[20] Chen H, Wang L, Lin Y, et al. “Structured Graph Convolutional Networks with Stochastic Masks for Recommender Systems”, Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2021, pp614-623.
[21] Tian C, Xie Y, Li Y, et al. “Learning to Denoise Unreliable Interactions for Graph Collaborative Filterin”, Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2022, pp122-132.
[22] Wu L, Lin H, Tan C, et al. “Self-supervised Learning on Graphs: Contrastive, Generative or Predictive”, IEEE Transactions on Knowledge and Data Engineering, 2021, pp1-37.
[23] Giorgi J, Nitski O, Wang B, et al. “Declutter: Deep contrastive Learning for Unsupervised Textual Representations”, arXiv preprint arXiv:2006.03659, 2020.
[24] Kingma D P, Ba J. “Adam: A Method for Stochastic Optimization”, arXiv preprint arXiv:1412.6980, 2014.
[25] Glorot X, Bengio Y. “Understanding the Difficulty of Training Deep Feed forward Neural Networks”,Proceedings of the 13th International Conference on Artificial Intelligence and Statistics, 2010, pp249-256.
[26] Zhu Y, Xu Y, Yu F, et al. “Deep Graph Contrastive Representation Learning”, arXiv preprint arXiv:2006.04131, 2020.
[27] Xia X, Yin H, Yu J, et al. “Self-supervised Hypergraph Convolutional Networks for Session-based Recommendation”, Proceedings of the AAAI Conference on Artificial Intelligence. 2021, pp4503-4511.
[28] Jing B, Xiang Y, Chen X, et al. “Graph-MVP: Multi-view Prototypical Contrastive Learning for Multiplex Graphs”. arXiv preprint arXiv:2109.03560, 2021.
[29] Chen T, Kornblith S, Norouzi M, et al. “A Simple Framework for Contrastive Learning of Visual Representations”, Proceeding of International Conference on Machine Learning, 2020, pp1597-1607.
[30] You Y, Chen T, Sui Y, et al. “Graph Contrastive Learning with Augmentations”, Advances in Neural Information Processing Systems, 2020, vol. 33, pp5812-5823.