



Article ID 1007-1202(2024)02-0134-11 DOI <https://doi.org/10.1051/wujns/2024292134>

Cite this article: ZHENG Sirui, HUANG Bo, LIU Jin, *et al.* Fu-Rec: Multi-Task Learning Recommendation Model Fusing Neighbor-Discrimination and Self-Discrimination[J]. *Wuhan Univ J of Nat Sci*, 2024, 29(2): 134-144.

Fu-Rec: Multi-Task Learning Recommendation Model Fusing Neighbor-Discrimination and Self-Discrimination

□ ZHENG Sirui¹, HUANG Bo^{1†}, LIU Jin², ZENG Guohui¹, YIN Ling¹, LI Zhi³, SUN Tie⁴

1. School of Electronic and Electrical Engineering, Shanghai University of Engineering Science, Shanghai 201600, China;

2. School of Computer, Wuhan University, Wuhan 430072, Hubei, China;

3. School of Computer Science and Engineering, Guangxi Normal University, Guilin 541004, Guangxi, China;

4. AIoT Manufacturing Solutions Technology Co., Ltd., Hefei 230000, Anhui, China

© Wuhan University 2024

Abstract: In recent years, self-supervised learning has achieved great success in areas such as computer vision and natural language processing because it can mine supervised signals from unlabeled data and reduce the reliance on manual labels. However, the currently generated self-supervised signals are either neighbor discrimination or self-discrimination, and there is no model to integrate neighbor discrimination and self-discrimination. Based on this, this paper proposes Fu-Rec that integrates neighbor-discrimination contrastive learning and self-discrimination contrastive learning, which consists of three modules: (1) neighbor-discrimination contrastive learning, (2) self-discrimination contrastive learning, and (3) recommendation module. The neighbor-discrimination contrastive learning and self-discrimination contrastive learning tasks are used as auxiliary tasks to assist the recommendation task. The Fu-Rec model effectively utilizes the respective advantages of neighbor-discrimination and self-discrimination to consider the information of the user's neighbors as well as the user and the item itself for the recommendation, which results in better performance of the recommendation module. Experimental results on several public datasets demonstrate the effectiveness of the Fu-Rec proposed in this paper.

Key words: self-supervised learning; recommendation system; contrastive learning; multi-task learning

CLC number: TP391

0 Introduction

As a tool that can tap the potential interests of users, recommendation systems^[1] are widely used in

e-commerce platforms to optimize the user experience. In recent years, neural networks have been applied to recommender systems^[2,3] to improve the effectiveness of recommendations. However, the deep recommendation

Received date: 2023-12-20

Foundation item: Supported by the Scientific and Technological Innovation 2030-Major Project of New Generation Artificial Intelligence (2020AAA0109300); Science and Technology Commission of Shanghai Municipality (21DZ2203100); 2023 Anhui Province Key Research and Development Plan Project - Special Project of Science and Technology Cooperation (2023i11020002)

Biography: ZHENG Sirui, female, Master candidate, research direction: recommendation system. E-mail: zhengsirui0322@163.com

† Corresponding author. E-mail: huangbosues@sues.edu.cn

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<https://creativecommons.org/licenses/by/4.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

model requires the support of a large amount of data. At the same time, the data applied to the recommendation system is the personalized data of each user, which is more challenging to collect. Hence, the sparsity of data makes neural networks have limitations in applying recommendation systems^[4].

Existing methods to solve the data sparsity problem are generally fixed-value padding method^[5], matrix dimensionality reduction method^[6], and content-based recommendation^[7]. However, the fixed-value filling method just fills in a fixed value where there is no interaction value and does not consider the characteristics of the user or the item, and the accuracy of the recommendation will be affected; for the inverse matrix dimensionality reduction method, the process of dimensionality reduction inevitably results in the loss of valid data, which will bring bias to the recommendation effect; for the recommendation method based on the content, it cannot tap into the user's potential needs: the system can only recommend similar goods that the user has used or liked and other behaviors, it is not clear what kind of goods the user will need in the future, which leads to not recommending new goods to the user. Recently, self-supervised learning^[8] (SSL) has been used to alleviate the data sparsity problem in recommender systems due to its characteristics.

In recent years, self-supervised learning has become a popular research area that aims to mine the data's representational properties as supervisory signals for unlabeled data through auxiliary tasks that can reduce the reliance on manual labels. Self-supervised learning has been widely used in significant fields, including computer vision representation learning^[9], language modeling and training^[10], node/graph classification^[11], etc., due to its ability to overcome the labeling insufficiency, and self-supervised learning has demonstrated strong performance in all these fields. Since the basic idea of SSL is to learn automatically generated supervised signals from raw data, which can solve the problem of data sparsity in recommender systems, SSL has excellent potential to improve recommendation performance, and more and more researchers are applying self-supervised learning to recommender systems.

Self-supervised recommendation (SSR) can be traced back to the earliest unsupervised methods, such as autoencoder-based recommendation models^[12,13], which rely on different loss data to reconstruct the original inputs to avoid overfitting. Later, network embedding-

based recommendation models^[14,15] emerged, which use random wandering similarity as a self-supervised signal to capture the similarity between users and items. At the same time, some researchers began to apply generative adversarial networks^[16] (GANs) to recommender systems^[17] to enhance the interaction between users and items, which can also be regarded as a self-supervised recommendation. In 2018, the pre-trained language model BERT^[18] made a huge breakthrough, and self-supervised learning also attracted the attention of researchers as an independent concept. The performance of the latest methods based on self-supervised learning in many computer vision, as well as natural language processing, is almost comparable to that of supervised learning^[19], and the rise of contrastive learning^[20] (CL) has dramatically boosted the research on self-supervised learning. The core idea of contrastive learning is the comparison of positive and negative samples in the feature space, and how to construct positive and negative samples has become the focus of research.

With the popularity of online social communities, users can share their opinions with others more conveniently. This has led many researchers to work on learning by combining online social network information with user-item interactions to alleviate the data sparsity problem and improve recommendation performance^[21,22]. A diffusion neural network^[23] (DiffNet) simulates recursive dynamic social diffusion in the space of users and items based on social networks; however, this approach fails to account for the variability in influence among users, which is common in reality. At the same time, it does not fully leverage the potential improvement of item feature representations through interactions with users. A multi-channel hypergraph convolutional network^[24] (MHCN) is based on hypergraph convolution and proposes a social recommendation method using hyperedges to form complex user correlations.

Most current self-supervised learning involves creating complementary views by node/edge discarding or random augmentation by adding masks to the original graph and then maximizing the consistency of the same nodes learned from different views. These approaches lack consideration of constructing self-supervised signals that are more meaningful for the recommendation task and are prone to the emergence of false negative pairs, leading to degradation of recommendation performance. To address this problem, SEPT^[25] optimizes recommendation performance by mining self-supervised

signals from other users; however, this approach does not use self-supervised signals from items. On the other hand, most of the current recommendation algorithms for self-supervised learning are based on either neighbor discrimination or self-discrimination. Still, no literature investigates the effect of hybrid discrimination of these two on recommendation performance.

Acknowledging the existing challenges in self-supervised learning, particularly the propensity for generating false negative pairs and the demand for more meaningful signals, this study proposes a novel approach. It leverages self-supervised signals from both items and users while delving into hybrid discrimination's nuances. This paper explicitly introduces multi-task learning^[26] as a strategy to enhance recommendation performance, employing neighbor-discrimination and self-discrimination methods as auxiliary tasks. These tasks are designed to complement the primary recommendation task by utilizing comparison learning, thereby addressing the twin challenges of improving signal relevance and reducing the likelihood of false negatives. This approach clarifies the role of hybrid discrimination in improving recommendation systems. In particular, considering the noise problem in the original user interaction view for neighbor-discrimination contrastive learning, we introduce user social networks for data augmentation to obtain more stable social relationships. This approach seeks three distinct yet complementary perspectives: a user's socializing preferences, sharing inclinations, and interaction tendencies. By computing user embeddings across these three perspectives, we calculate user neighbors within each viewpoint. The integration of these neighbors culminates in creating final positive instances for contrastive learning. For self-discrimination comparison learning, we note that LightGCN^[27] integrates higher-order information about users and items, so in this paper, we use the corresponding outputs of the different layers of LightGCN as the current positive examples of users and items for comparison learning.

The contributions of this paper are as follows:

- This paper proposes a multi-task learning recommendation model based on hybrid discrimination contrastive-assisted learning, and the recommendation performance can be significantly improved by unifying the recommendation task and the hybrid discrimination contrastive learning task under this model.
- This paper uses neighbor information from mul-

tle complementary views to get positive examples of neighbors from the user-item interaction view for neighbor-discrimination contrastive learning.

- We conduct experiments on several real datasets to demonstrate the advantages of our proposed model and verify the effectiveness of each module in the model through ablation experiments.

1 Model

On the basis of previous work^[28], this chapter presents the Fu-Rec in four parts: preliminaries, neighbor-discrimination contrastive learning, self-discrimination contrastive learning, and recommendation. Figure 1 shows the framework of Fu-Rec. In preliminaries, we perform data augmentation on the original view to obtain three complementary views describing different aspects of user preferences and an unlabeled sample set for comparison learning. In neighbor-discrimination contrastive learning, we use multi-view learning to find the current user's neighboring users on each of the three complementary views, integrate the possibilities of neighboring users on these three views, and only if a specific user is the current user's neighbor on all three views, it will be considered as a positive example, and then we use contrastive learning to compare the current user on the interaction view with the neighboring users on the unlabeled sample set. In self-discrimination contrastive learning, we use the encodings of different layers of users and items as positive examples for contrastive learning. The neighbor-discrimination contrastive learning task and the self-discrimination contrastive learning task, which are auxiliary tasks, assist the main recommendation task in obtaining more descriptive user embeddings and accurate item embeddings.

Unlike many graph models that depend on complex feature transformations and multiple nonlinear activation layers, LightGCN opts for a more straightforward path. It streamlines the model architecture by removing parts that could increase computational load, thus significantly boosting operational efficiency and reducing computational complexity. This design allows LightGCN to more effectively capture and utilize the structural features of data with explicit graph structures. Considering our dedication to developing an efficient recommendation system emphasizing graph structures and embedding techniques, LightGCN has become our preferred encoder model for its efficiency and focused attributes.

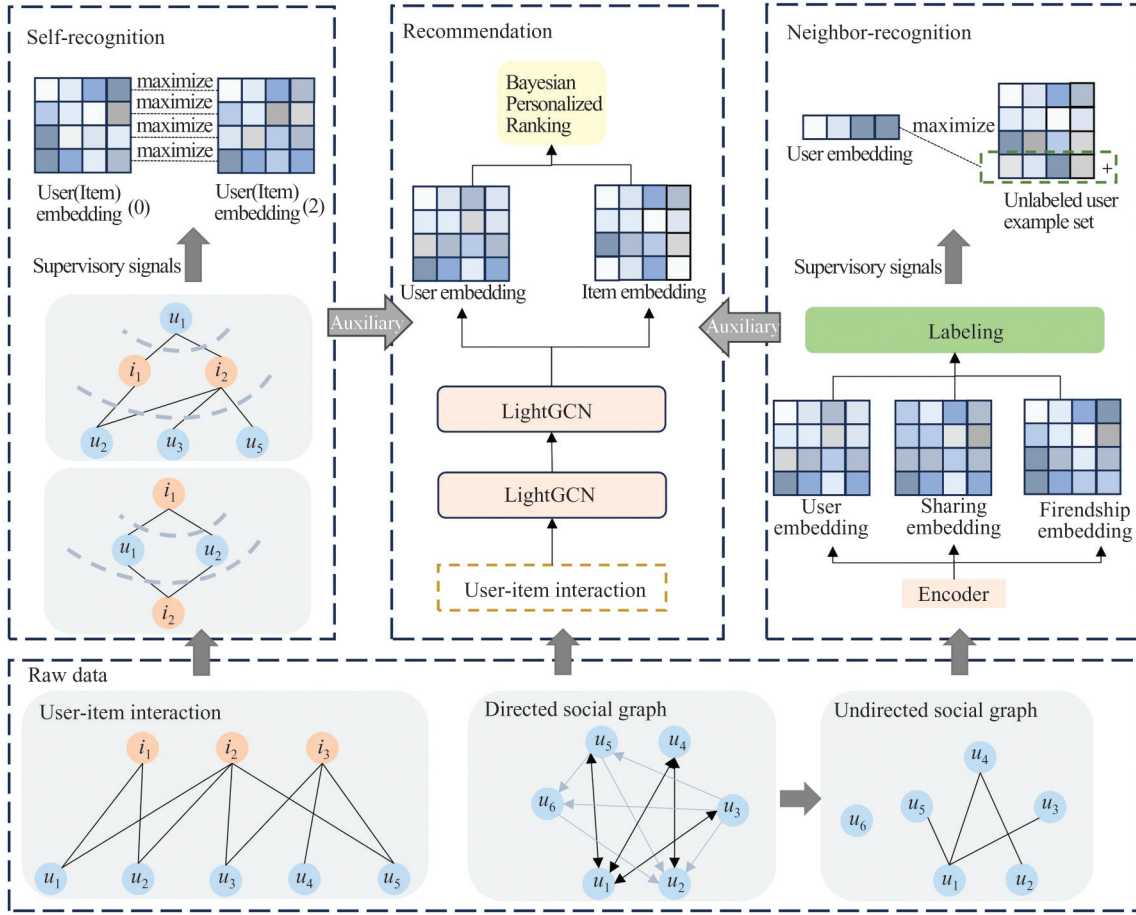


Fig. 1 The framework of Fu-Rec

1.1 Preliminaries

Performing neighbor-discrimination contrastive learning requires multiple complementary views and an unlabeled sample set from the same data source. In contrast, complementary views refer to using the same data source to construct multiple views from different perspectives. The unlabeled sample set is used for classifier prediction to generate pseudo-labels for comparison learning.

For the acquisition of multiple complementary views, this paper introduces user-item interaction views and user social network views from one data source, user behavior data, to augment the data. Since social relationships are inherently noisy, to accurately capture self-supervised information, this paper uses the same data augmentation approach as MHCN^[24], which utilizes ternary closure relationships prevalent among users as reliable social relationships and thus acquires users' semantic neighbors. Through the user-item interaction graph and the user-social network graph, we can easily identify two types of triangles: the first type, as illus-

trated in Fig. 2(a), shows an item purchased by two friends, reflecting a preference among users to share common items with their friends. The second type, depicted in Fig. 2(b), represents social connections between three users who share mutual friends, indicating a tendency among users to expand their social networks. In this way, the complementary views of the two aspects of the user's preference for expanding his social circle and recommending items to his friends can be obtained, and together with the user-item interaction view describing the user's preference for different items, three complementary views illustrating the various aspects of the user's preference are formed.

This paper uses matrix multiplication to obtain stable social relationships, $\mathbf{M}_s$ denotes the user sharing preference adjacency matrix and $\mathbf{M}_f$ denotes the user dating preference adjacency matrix, and the calculation formula is shown in the following equations:

$$\mathbf{M}_f = (\mathbf{S}\mathbf{S}^T) \odot \mathbf{S} \quad (1)$$

$$\mathbf{M}_s = (\mathbf{R}\mathbf{R}^T) \odot \mathbf{S} \quad (2)$$

where $\mathbf{S}$ denotes the social adjacency matrix, and $\mathbf{R}$ de-

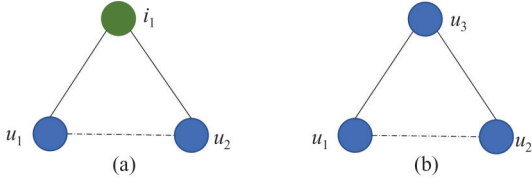


Fig. 2 Stable social relationships

notes the user-item interaction adjacency matrix. The path connecting two users through a common friend (or item) can be obtained by matrix multiplication, and the Hadamard product $\odot$ ensures that the relationships in $\mathbf{M}_s$ and $\mathbf{M}_f$ are subsets of the social neighbor matrix $\mathbf{S}$.

An unlabeled sample set is necessary to obtain pseudo-labels from multiple views for contrastive learning. In this paper, we dynamically perturb the user social network view and the user-item interaction view to create the unlabeled sample set by edge loss with probability ρ on the original graph. The specific process is shown in Eq. (3):

$$\tilde{\mathcal{G}} = (\mathcal{N}_r \cup \mathcal{N}_s, \mathbf{m} \odot (\mathcal{E}_r \cup \mathcal{E}_s)) \quad (3)$$

where, $\mathcal{N}_r$ and $\mathcal{N}_s$ are nodes, $\mathcal{E}_r$ and $\mathcal{E}_s$ are edges of the user-item interaction graph and the social network graph, $\mathcal{E}_r \cup \mathcal{E}_s$ is the set of edges, and $\mathbf{m} \in \{0, 1\}^{|\mathcal{E}_r \cup \mathcal{E}_s|}$ is a vector of masks on the set of edges, and the probability of the occurrence of zero in this vector is ρ .

1.2 Neighbor-Discrimination Contrastive Learning

In neighbor-discrimination contrastive learning, this paper proposes integrating the positive examples from different views to get the positive ones in the interaction view for contrastive learning. Precisely, in each batch, encoders from different views predict the most likely semantic neighbors in the unlabeled sample set for the user. Then, contrastive learning is introduced to optimize the user representation using the self-supervised signals generated from these predictions. During this iterative process, the encoders from all viewpoints are progressively improved, and the pseudo-labels generated by these encoders become more accurate. This recursive improvement process helps to enhance the performance of the encoders so that they can capture and represent user features and preferences from different perspectives more efficiently.

The process of obtaining neighboring nodes for user u , as illustrated in Fig. 3, is mathematically formulated in equations (4-8):

$$p_{u+}^s = \text{softmax}(\phi(\mathbf{e}^{\tilde{z}}, \mathbf{e}_u^s)) \quad (4)$$

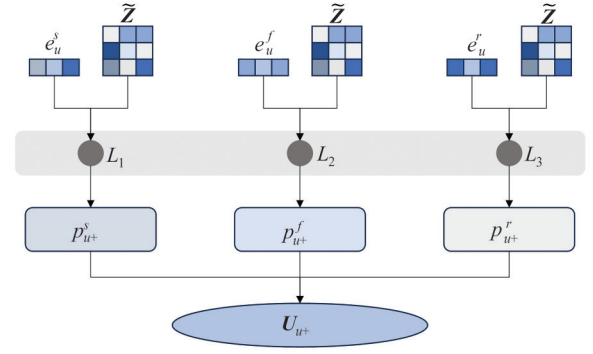


Fig. 3 The process of finding the user's neighbors

$$p_{u+}^f = \text{softmax}(\phi(\mathbf{e}^{\tilde{z}}, \mathbf{e}_u^f)) \quad (5)$$

$$p_{u+}^r = \text{softmax}(\phi(\mathbf{e}^{\tilde{z}}, \mathbf{e}_u^r)) \quad (6)$$

where p_{u+}^s , p_{u+}^f , and p_{u+}^r denote the predicted probability of each user in the unlabeled sample set to be labeled as a semantically positive sample of user u in the corresponding view, ϕ is a cosine operation, and $\mathbf{e}_u^s$, $\mathbf{e}_u^f$, and $\mathbf{e}_u^r$ are the encoder learnings defined on the user sharing preferences view, the user friending preferences view, and the user-item interaction view, respectively, to the representation of user u , and $\mathbf{e}^{\tilde{z}}$ represents user embeddings in each batch within that iteration.

To avoid noisy samples, in this paper, we use consistent predictions on the three views to obtain positive examples of the neighbors of user u . Specifically, given a user, we integrate user representations from the three views to predict the semantic neighbors of the user in the unlabeled sample set.

$$p_{u+} = p_{u+}^s + p_{u+}^f + p_{u+}^r \quad (7)$$

where p_{u+}^s , p_{u+}^f and p_{u+}^r are calculated by Eq. (4-6); we can select the N positive samples with the highest probability based on these probabilities. This process is shown in the following equation:

$$\mathbf{U}_{u+} = \{\mathbf{e}_n^{\tilde{z}} | n \in \text{Top-}N(p_{u+})\} \quad (8)$$

where $\mathbf{U}_{u+}$ is the N semantic neighbor of user u in the unlabeled sample set.

Comparison learning using these pseudo-labels is then performed, given a specific user u , to maximize the consistency between the representation of u and the user representations in $\mathbf{U}_{u+}$, and to minimize the consistency between its node representations and the other user representations. Formally, this paper adopts InfoNCE^[29], which is effective in mutual information estimation, as the learning objective to maximize the consistency between positive pairs and minimize the consistency between negative pairs.

$$L_{ND} = -\log \frac{\sum_{p \in U_{u+}} \psi(\mathbf{e}_u^r, \mathbf{e}_p^z)}{\sum_{p \in U_{u+}} \psi(\mathbf{e}_u^r, \mathbf{e}_p^z) + \sum_{j \in J(U_{u+})} \psi(\mathbf{e}_u^r, \mathbf{e}_j^z)} \quad (9)$$

where $\psi(\mathbf{e}_u^r, \mathbf{e}_p^z) = \exp(\phi(\mathbf{e}_u^r \cdot \mathbf{e}_p^z)/\tau)$, $\phi(\cdot): \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ is a discriminator function that inputs two vectors, and outputs the congruence between them. In this paper, we use the cosine operation for the discriminator, and a softmax operation is followed by giving closer negative samples more penalty, τ controls the degree of this penalty, which is set to 0.1 in this paper.

1.3 Self-Discrimination Contrastive Learning

For self-discrimination contrastive learning, inspired by NCL^[30], we take the even-hop neighbors of the user and the project as positive examples because the even-numbered layer of LightGCN will aggregate the initial embeddings of the neighboring nodes from the user or the project, so we will take the nodes corresponding to the output of the even-numbered layer of LightGCN and the encoding of the 0th layer as the positive examples for the contrastive learning, which is, nodes compare themselves with themselves, so it is called self-discrimination. Since LightGCN also has the common problem of the GNN model, the issue of over-smoothing occurs in multi-layer stacking, so we only choose the output of the second layer of LightGCN and the learnable initial embedding for comparison learning.

Specifically, the loss in self-discrimination user contrastive learning is defined as L_{SD}^U .

$$L_{SD}^U = \sum_{u \in U} -\log \frac{\exp((\mathbf{e}_u^{r(2)} \cdot \mathbf{e}_u^{r(0)}/\tau))}{\sum_{v \in U} \exp((\mathbf{e}_u^{r(2)} \cdot \mathbf{e}_v^{r(0)}/\tau))} \quad (10)$$

where $\mathbf{e}_u^{r(0)}$ is the initial embedding of user u on the user-item interaction view, $\mathbf{e}_u^{r(2)}$ is the user embedding output of the second layer of LightGCN defined on the user-item interaction view, U is the set of users, and τ is the temperature coefficient.

Similarly, the self-discrimination item contrastive learning loss function is defined as L_{SD}^I .

$$L_{SD}^I = \sum_{I \in I} -\log \frac{\exp((\mathbf{e}_I^{r(2)} \cdot \mathbf{e}_I^{r(0)}/\tau))}{\sum_{j \in I} \exp((\mathbf{e}_I^{r(2)} \cdot \mathbf{e}_j^{r(0)}/\tau))} \quad (11)$$

where, $\mathbf{e}_I^{r(0)}$ is the initial embedding of item I on the user-item interaction view, $\mathbf{e}_I^{r(2)}$ is the item embedding output of the user-item interaction view at the second layer of LightGCN, I is the set of items, and τ is the temperature coefficient.

The overall self-discrimination comparison learning loss function is:

$$L_{SD} = L_{SD}^U + L_{SD}^I \quad (12)$$

1.4 Recommendations

To directly extract information from the interactions between users and items, we utilize the Bayesian Personalization Ranking (BPR)^[31] loss, which is an objective function designed explicitly for the ranking tasks of recommendation systems. The core idea of the BPR loss is to ensure that the model's predicted scores for actual user-item interactions are higher than those for interactions that have not occurred. In short, it emphasizes the importance of actual user preferences by comparing the predicted scores for occurred interactions against those that did not occur. Specifically, the objective function of the BPR loss can be expressed as follows:

$$L_r = \sum_{i \in I(u), j \notin I(u)} -\log \sigma(r_{ui} - r_{uj}) + \lambda \|\mathbf{E}\|_2^2 \quad (13)$$

where $I(u)$ is the set of items that user u interacts with, $\hat{r}_{u_i} = \mathbf{e}_u^{rT} \mathbf{e}_i^r$, $\mathbf{e}_u^r$ and $\mathbf{e}_i^r$ are the user and item codes obtained from the encoder defined on the user-item interaction graph, and λ is the coefficient that controls the L2 regularization.

1.5 Overall Optimization Objective

This paper employs a multi-task learning framework, with neighbor-discrimination contrastive learning and self-discrimination contrastive learning as auxiliary tasks and the recommendation task as the main task. We need an overall loss function to assist the model in optimization, and the overall optimization objective is defined as:

$$L = L_r + \beta L_{ND} + \gamma L_{SD} \quad (14)$$

where, L_r is the loss of the recommendation module derived from Eq. (13), L_{ND} is the loss of neighbor-discrimination contrastive learning derived from Eq. (9), L_{SD} is the loss of self-discrimination contrastive learning derived from Eq. (12), and β and γ are the parameters that regulate the proportion of L_{ND} and L_{SD} in the total loss.

2 Experiments

2.1 Datasets and Experimental Setup

In this paper, experiments are conducted on three real datasets to evaluate the proposed model: lastfm, yelp, and FilmTrust, and the specific information of these three datasets is shown in Table 1. The proposed model aims at Top- N recommendation, and for the accuracy of the evaluation, all experiments are cross-

validated with 5 folds and average results are given. For the general settings of all the methods, the dimensionality of the representation to be learned in this paper is set to 50, the regularization parameter λ is set to 0.001, and the batch size is set to 2 000. This paper uses Adam to optimize all the models with an initial learning rate of 0.001. The processor model of the server used in this paper is Intel® Xeon(R) CPU E5-2603 V4 @ 170GHz x 6, and the graphics card model is GeForce RTX 2080.

Table 1 Datasets statistics

Dataset	User	Item	Feedback	Relation	Density/%
lastfm	1 892	17 632	92 834	25 434	0.28
yelp2018	19 539	21 266	450 884	864 157	0.11
FilmTrust	1 509	2 071	35 497	1 853	1.13

2.2 Benchmark Models and Evaluation Metrics

The following six recommended models are continued to be compared with the Fu-Rec in this paper to test the effectiveness of the Fu-Rec in this paper.

NeuMF^[32]: a recommendation algorithm based on deep learning, which combines traditional matrix decomposition and multilayer perceptron machine while extracting low and high dimensional features to improve the recommendation effect.

DiffNet^[23]: By modeling user preferences based on user social relationships and historical behaviors and applying the GraphSAGE framework to model the social propagation process, DiffNet can capture the more profound social propagation process using GNN.

NGCF^[33]: A network model explicitly models higher-order connectivity between users and items to enhance user and item representation.

LightGCN^[27]: A lightweight GCN network for recommender systems that learns user and item embeddings by linearly propagating over the user-item interaction matrix and finally uses the weighted sum of the learned representations from all layers as the final representation.

MHCN^[24]: A social recommendation model based on hypergraph convolutional networks that use hyperedges between users to model complex correlations between users to improve recommendation.

SEPT^[25]: A social network-based multi-view self-supervised learning model that uses a comparative learning approach based on neighbor discrimination to refine user representations using pseudo-labels in neighbors to optimize recommendations.

In this paper, several evaluation metrics commonly used in recommender systems are used to evaluate the model, i.e., Precision@10, Recall@10, and NDCG@10. Among them, Precision@10 measures the proportion of correct items in the top 10 recommended items. This indicator focuses on the accuracy of the recommendation system in the first few items because users are usually more concerned about the top few recommendations. Recall@10 represents the proportion of correctly recalled items in the top 10 recommended items. This indicator reflects the ability of the recommendation system to find relevant items. NDCG is an index that comprehensively considers order and relevance. It measures the quality of recommendations by assigning different weights to each correctly recommended item and assumes the position of the item in the calculation. The formulas are as follows, respectively:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (15)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (16)$$

$$\text{NDCG} = \frac{\text{DCG}_p}{\text{IDCG}_p}$$

$$\text{DCG}_p = \sum_{i=1}^p \frac{2^{\text{rel}_i} - 1}{\log(i + 1)}$$

$$\text{IDCG}_p = \sum_{i=1}^{|\text{REL}|} \frac{2^{\text{rel}_i} - 1}{\log(i + 1)} \quad (17)$$

where TP denotes correctly predicted as a positive sample, FP denotes incorrectly predicted as a positive sample, and FN denotes incorrectly predicted as a negative sample. The rel_i denotes the correlation at position i , and |REL| denotes the set of results sorted according to the optimal way.

2.3 Comparison Experiments

Table 2 gives the experimental results of the model proposed in this paper and the baseline model on three datasets. From the experimental results data in the table, it can be seen that the model has some lead over the baseline model. One of the reasons is that the SDMS model introduces LightGCN to obtain user embeddings and item embeddings, and the graph neural network-based approach can encode the higher-order information of the bipartite graph into the embeddings, which leads to better performance. Although the NeuMF model also obtains user and item embeddings through the user-item interaction graph, the matrix decomposition and the multilayer perceptual machine cannot mine the higher-order information well. Although the NGCF model can also

obtain higher-order information on the user-item interaction bipartite graph, the feature transformation and non-linear activation reduce the recommendation effect of

collaborative filtering to a certain extent. In contrast, LightGCN only uses neighbor aggregation, simplifying the graph convolution operation.

Table 2 The results of comparison experiments

Method	lastfm			yelp			FilmTrust		
	Prec@10	Rec@10	NDCG@10	Prec@10	Rec@10	NDCG@10	Prec@10	Rec@10	NDCG@10
NeuMF	0.140 5	0.142 4	0.173 1	0.020 5	0.052 6	0.036 9	0.263 0	0.497 6	0.564 6
DiffNet	0.183 6	0.186 3	0.216 5	0.023 7	0.061 5	0.045 1	0.265 6	0.504 1	0.566 3
NGCF	0.162 9	0.165 4	0.197 7	0.024 2	0.063 3	0.044 7	0.260 5	0.485 3	0.538 2
LightGCN	0.165 5	0.167 6	0.205 8	0.026 2	0.067 5	0.053 5	0.267 2	0.511 9	0.571 7
MHCN	0.200 5	0.203 7	0.243 9	0.029 6	0.076 8	0.059 7	0.276 2	0.534 5	0.596 7
SEPT	0.201 7	0.204 1	0.244 5	0.028 4	0.072 5	0.055 6	0.284 7	0.547 1	0.598 4
Fu-Rec	0.202 3	0.205 5	0.247 0	0.028 6	0.073 4	0.057 1	0.295 7	0.568 4	0.614 6

Meanwhile, neighbor-discrimination contrastive learning is introduced to obtain higher-order information on views to get the user's neighbors through LightGCN. Specifically, augmenting data from the user-item interaction view and user social network graph to obtain a more stable social view, the user-item interaction view, user sharing preference view, and user dating preference view are used as the complementary views of the three perspectives, which are encoded by LightGCN. The user embeddings on the three views are used to obtain the user's neighbors, which in turn can be used as positive examples. Comparison learning is performed, which helps to reduce the possibility of false positive samples appearing, thus obtaining more accurate user embeddings. Although DiffNet also mines deep social relationships through GNN, its redundant parameters also limit the effect of recommendation. Although it can mine higher-order information, the recommendation of the LightGCN model lacks the advantage of self-supervised learning. It lacks the potential relationships to be utilized, so the accuracy of the recommendation will be limited.

Self-supervised learning recommendation models incorporating social networks are represented by MHCN and SEPT, where MHCN considers maximizing the hierarchical mutual information between users, user-centric sub-supergraphs, and supergraph representations to facilitate recommendation, and SEPT obtains positive examples of users from the encoding level for comparative learning. Both models are based on the perspective of neighbor discrimination and do not consider the influ-

ence of users or items themselves on recommendation. However, on the Yelp dataset, Fu-Rec's performance is not as good as the MHCN model, one reason being that Fu-Rec only learns the bottom embeddings while MHCN learns many other parameters as well. At the same time, Fu-Rec runs much faster than MHCN, which makes Fu-Rec still competitive despite being slightly inferior in terms of effectiveness.

Another reason that the Fu-Rec model proposed in this paper outperforms the baseline model is that the Fu-Rec model fuses self-discrimination comparative learning and neighbor-discrimination comparative learning. This fusion strategy can comprehensively improve the depth and breadth of feature learning by combining self-discrimination (which focuses on the intrinsic characteristics of the entity itself) and neighbor-discrimination (which focuses on the relationship between the entity and its neighbors in the network) comparative learning. This enhances the generalization ability of the model when dealing with diverse and complex datasets and improves the recommender system's accuracy and personalization level. In addition, this approach can effectively alleviate the data sparsity problem, especially in scenarios with less social network and user-item interaction data. Meanwhile, fusing different types of contrastive learning also improves the robustness of the model to noise and outliers, thus reducing errors and biases. Overall, this fusion contrastive learning approach is vital in deep feature mining, model performance improvement, and accuracy and personalized recommendation of recommender systems.

2.4 Ablation Experiments

To further validate the effectiveness of the critical components of the proposed Fu-Rec model, we designed ablation experiments on the lastfm, yelp, and FilmTrust datasets. Fu-Rec/N denotes no neighbor-discrimination contrastive learning module, and Fu-Rec/S denotes no self-discrimination contrastive learning module.

Figure 4 shows the experimental results of the ablation experiments on the three datasets in the form of a

line graph. From the data in Fig. 4, it can be seen that by removing the neighbor-discrimination contrastive learning module, the model will have difficulty in capturing finer-grained neighbor information, which leads to a decrease in the recommendation performance; by removing the self-discrimination contrastive learning module, the model will not be able to capture the potential preference or information of the user or the item itself, which makes the recommendation effect unsatisfactory.

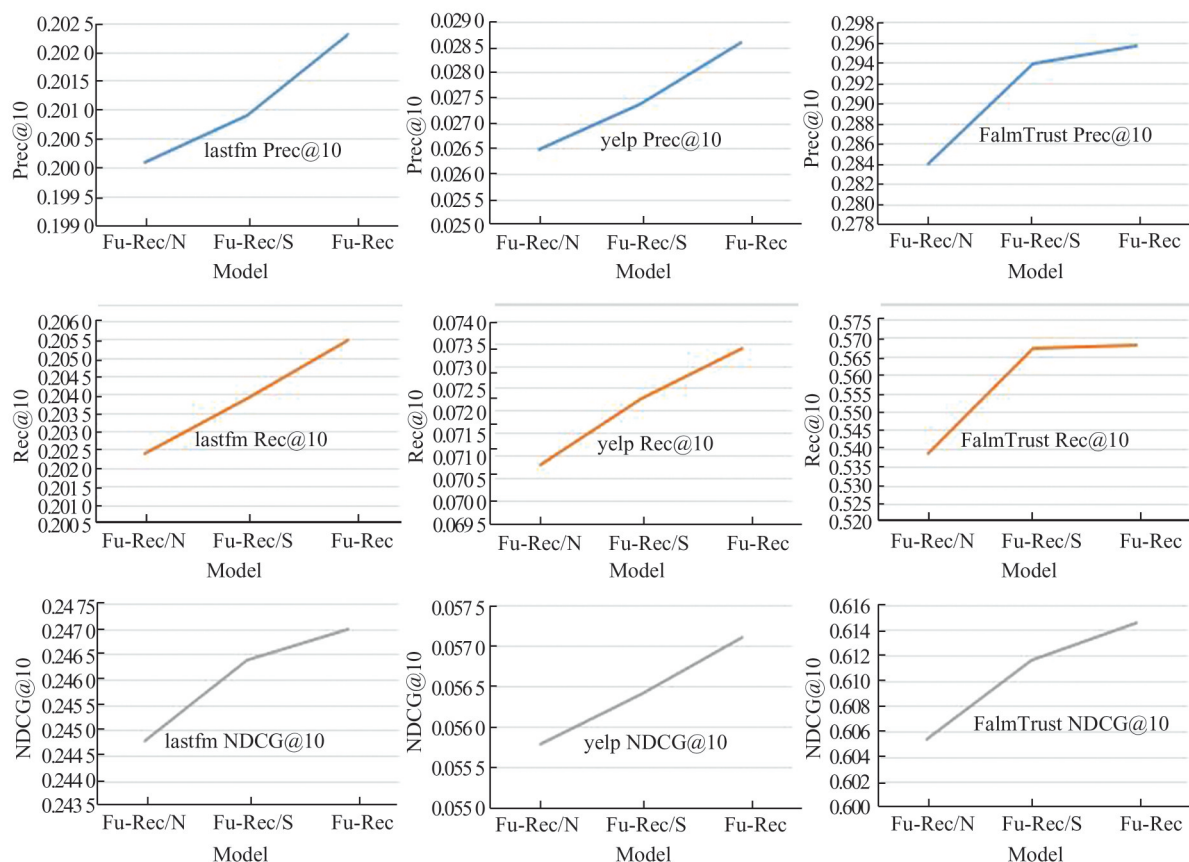


Fig. 4 The results of ablation experiments

In short, all modules of the Fu-Rec model are necessary, and removing any of them will have a negative impact on the model.

2.5 Discussion

In this subsection, the Fu-Rec model's performance surpasses that of most baseline models, demonstrating a particular advantage of the recommendation algorithm that integrates neighbor discrimination and self-discrimination in recommendation performance. By incorporating neighbor-discrimination and self-discrimination, the Fu-Rec model can capture information and behavioral characteristics from neighbors and

the users or items themselves, allowing the user and item embeddings to reflect reality accurately. The introduction of neighbor discrimination enhances the model's recognition of the relationships between users and their neighbors. At the same time, self-discrimination ensures the model's acquisition of the intrinsic features of users or items themselves. The combination of the two facilitates efficient modeling of user and item features. Ultimately, the performance of the Fu-Rec model on datasets such as last.fm, yelp, and FilmTrust prove its superiority. Moreover, ablation experiments further confirm the importance of each module within the model, demon-

strating that every part is essential for enhancing recommendation performance.

In our research, guided by theoretical studies and previous literature, we meticulously selected key parameters for our model, such as learning rate, number of iterations, and hyperparameters in the loss function. We fine-tuned these parameters through a series of experiments. We experimented with values ranging from 0 to 0.2 for the weight parameters in the overall loss, with a step size of 0.01. While we recognize the potential limitations of this approach, such as generalization issues across different datasets, we plan to explore these areas in future research further. Overall, the careful configuration of these parameters played a crucial role in enhancing the performance of our method.

3 Conclusion

In this paper, we proposed a recommendation model Fe-Rec that integrates neighbor-discrimination contrastive learning and self-discrimination contrastive learning, which consists of three modules: 1) neighbor-discrimination contrastive learning, 2) self-discrimination contrastive learning, and 3) recommendation module. Neighbor-discrimination contrastive learning and self-discrimination contrastive learning tasks are used as auxiliary tasks to assist the recommendation task. The Fu-Rec model effectively uses the respective advantages of neighbor-discrimination and self-discrimination. It considers the information of the user's neighbors, the user, and the item itself to make recommendations, thus enabling the recommendation module to perform better. Comparison experiments on the lastfm, yelp, and FilmTrust datasets verify the Fu-Rec model's advantages, and each module's effectiveness is verified through ablation experiments.

References

- [1] Bobadilla J, Ortega F, Hernando A, *et al*. Recommender systems survey[J]. *Knowledge-Based Systems*, 2013, **46**: 109-132.
- [2] Covington P, Adams J, Sargin E. Deep neural networks for YouTube recommendations[C]// *Proceedings of ACM Conference on Recommender Systems*. New York: ACM, 2016: 191-198.
- [3] Cheng H T, Koc L, Harmsen J, *et al*. Wide & deep learning for recommender systems[C]// *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*. New York: ACM, 2016:7-10.
- [4] Zhang S, Yao L N, Sun A, *et al*. Deep learning based recommender system: A survey and new perspectives[J]. *ACM Computing Surveys (CSUR)*, 2019, **52**(1): 1-38.
- [5] Dai J L. *Study on the Sparsity Problem of Collaborative Filtering Algorithm*[D]. Chongqing: Chongqing University, 2013(Ch).
- [6] Vozalis M G, Margaritis K G. Using SVD and demographic data for the enhancement of generalized collaborative filtering[J]. *Information Sciences*, 2007, **177**(15): 3017-3037.
- [7] Liu H, Guo M M, Pan W Q. Overview of personalized recommendation systems[J]. *Journal of Changzhou University(Natural Science Edition)*, 2017, **29**(3): 51-59(Ch).
- [8] Liu X, Zhang F J, Hou Z Y, *et al*. Self-supervised learning: Generative or contrastive[J]. *IEEE Transactions on Knowledge and Data Engineering*, 2021, **35**(1): 857-876.
- [9] He K M, Fan H Q, Wu Y X, *et al*. Momentum contrast for unsupervised visual representation learning[C]//*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. New York: IEEE, 2020: 9729-9738.
- [10] Lan Z Z, Chen M D, Goodman S, *et al*. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations[EB/OL].[2019-09-26].<https://arxiv.org/pdf/1909.11942.pdf>.
- [11] Qiu J Z, Chen Q B, Dong Y X, *et al*. GCC: Graph contrastive coding for graph neural network pre-training[C]// *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. New York: ACM, 2020: 1150-1160.
- [12] Wu Y, DuBois C, Zheng A X, *et al*. Collaborative denoising auto-encoders for top-N recommender systems[C]// *Proceedings of the 9th ACM International Conference on Web Search and Data Mining*. New York: ACM, 2016: 153-162.
- [13] Li S, Kawale J, Fu Y. Deep collaborative filtering via marginalized denoising auto-encoder[C]// *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. New York: ACM, 2015: 811-820.
- [14] Gao M, Chen L H, He X N, *et al*. Bine: Bipartite network embedding[C]// *Proceedings of the 41st International ACM SIGIR Conference on Research&Development in Information Retrieval*. New York: ACM, 2018: 715-724.
- [15] Zhang C X, Yu L, Wang Y, *et al*. Collaborative user network embedding for social recommender systems[C]// *Proceedings of the 2017 SIAM International Conference on Data Mining*. Beijing: Society for Industrial and Applied Mathematics, 2017: 381-389.

- [16] Goodfellow I, Pouget-Abadie J, Mirza M, *et al.* Generative adversarial nets[C]// *Proceedings of Advances in Neural Information Processing Systems*. Berlin: Springer-Verlag, 2014: 2672-2680.
- [17] Wang Q Y, Yin H Z, Wang H, *et al.* Enhancing collaborative filtering with generative augmentation[C]// *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. New York: ACM, 2019: 548-556.
- [18] Devlin J, Chang M W, Lee K, *et al.* BERT: Pre-training of deep bidirectional transformers for language understanding [EB/OL]. [2023-06-20]. <https://arXiv.org/arXiv.org/nlp.stanford.edu/tooob.com>. 2019: 4171-4186.
- [19] Grill J B, Strub F, Althé F, *et al.* Bootstrap your own latent-a new approach to self-supervised learning[J]. *Advances in Neural Information Processing Systems*, 2020, **33**: 21271-21284.
- [20] Jaiswal A, Babu A R, Zadeh M Z, *et al.* A survey on contrastive self-supervised learning[J]. *Technologies*, 2020, **9**(1): 2.
- [21] Chen C, Zhang M, Liu Y Q, *et al.* Social attentional memory network: Modeling aspect-and friend-level differences in recommendation[C]//*Proceedings of the 12th ACM International Conference on Web Search and Data Mining*. New York: ACM, 2019: 177-185.
- [22] Fan W Q, Ma Y, Li Q, *et al.* Graph neural networks for social recommendation[C]//*Proceedings of the World Wide Web Conference*. New York: ACM, 2019: 417-426.
- [23] Wu L, Sun P J, Fu Y J, *et al.* A neural influence diffusion model for social recommendation[C]//*Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York: ACM, 2019: 235-244.
- [24] Yu J L, Yin H Z, Li J D, *et al.* Self-supervised multi-channel hypergraph convolutional network for social recommendation[C]//*Proceedings of the Web Conference 2021*. New York: ACM, 2021: 413-424.
- [25] Yu J L, Yin H Z, Gao M, *et al.* Socially-aware self-supervised tri-training for recommendation[C]//*Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. New York: ACM, 2021: 2084-2092.
- [26] Zhang Y, Yang Q. A survey on multi-task learning[J]. *IEEE Transactions on Knowledge and Data Engineering*, 2022, **34** (12):5586-5609.
- [27] He X N, Deng K, Wang X, *et al.* LightGCN: Simplifying and powering graph convolution network for recommendation[C]//*Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York: ACM, 2020: 639-648.
- [28] Huang B, Zheng S R, Fujita H, *et al.* A multi-task learning model for recommendation based on fusion of dynamic and static neighbors[J]. *Engineering Applications of Artificial Intelligence*, 2024, **133**: 108190.
- [29] Oord A, Li Y Z, Vinyals O. Representation learning with contrastive predictive coding[EB/OL]. [2018-07-10]. <https://arxiv.org/pdf/1807.03748.pdf>.
- [30] Lin Z H, Tian C X, Hou Y P, *et al.* Improving graph collaborative filtering with neighborhood-enriched contrastive learning[C]// *Proceedings of the ACM Web Conference*. New York: ACM, 2022: 2320-2329.
- [31] Rendle S, Freudenthaler C, Gantner Z, *et al.* BPR: Bayesian personalized ranking from implicit feedback[C]// *Proceedings of Uncertainty in Artificial Intelligence*. New York: ACM, 2009: 452-461.
- [32] He X N, Liao L Z, Zhang H W, *et al.* Neural collaborative filtering[C]// *Proceedings of the 26th International Conference on World Wide Web*. New York: ACM, 2017: 173-182.
- [33] Wang X, He X N, Wang M, *et al.* Neural graph collaborative filtering[C]//*Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York: ACM, 2019: 165-174.

□