

Article

Social Recommendation Based on Multi-Auxiliary Information Constrastive Learning

Feng Jiang ¹, Yang Cao ^{2,*}, Huan Wu ³ , Xibin Wang ⁴, Yuqi Song ⁵ and Min Gao ²¹ School of Finance and Management, Chongqing Business Vocational College, Chongqing 401331, China² School of Big Data and Software Engineering, Chongqing University, Chongqing 401331, China³ College of Environmental Science and Engineering, Tongji University, Shanghai 200092, China⁴ School of Data Science, Guizhou Institute of Technology, Guiyang 550003, China⁵ Department of Computer Science and Engineering, University of South Carolina, Columbia, SC 29201, USA

* Correspondence: caoyang@cqu.edu.cn

Abstract: Social recommendation can effectively alleviate the problems of data sparseness and the cold start of recommendation systems, attracting widespread attention from researchers and industry. Current social recommendation models use social relations to alleviate the problem of data sparsity and improve recommendation performance. Although self-supervised learning based on user–item interaction can enhance the performance of such models, multi-auxiliary information is neglected in the learning process. Therefore, we propose a model based on self-supervision and multi-auxiliary information using multi-auxiliary information, such as user social relationships and item association relationships, to make recommendations. Specifically, the user social relationship and item association relationship are combined to form a multi-auxiliary information graph. The user–item interaction relationship is also integrated into the same heterogeneous graph so that multiple pieces of information can be spread in the same graph. In addition, we utilize the graph convolution method to learn user and item embeddings, whereby the user embeddings reflect both user–item interaction and user social relationships, and the item embeddings reflect user–item interaction and item association relationships. We also design multi-view self-supervising auxiliary tasks based on the constructed multi-auxiliary views. Signals generated by self-supervised auxiliary tasks can alleviate the problem of data sparsity, further improving user/item embedding quality and recommendation performance. Extensive experiments on two public datasets verify the superiority of the proposed model.



Citation: Jiang, F.; Cao, Y.; Wu, H.; Wang, X.; Song, Y.; Gao, M. Social Recommendation Based on Multi-Auxiliary Information Constrastive Learning. *Mathematics* **2022**, *10*, 4130. <https://doi.org/10.3390/math10214130>

Academic Editors: Huawen Liu, Chengyuan Zhang, Weiren Yu and Chunwei Tian

Received: 27 August 2022

Accepted: 28 October 2022

Published: 5 November 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: social relations; auxiliary information; self-supervised learning; graph convolutional neural network

MSC: 68T01

1. Introduction

The rapid development of the Internet has made life more convenient and produced a large amount of information, causing the problem of information overload. It is difficult for users to select a target product that matches their preferences among hundreds of millions of products. The recommendation system significantly alleviates information overload and improves user experience. However, it is difficult to mitigate the problems of data sparsity and the cold start problem [1–4] in recommendation systems. In recommendation scenarios, user preferences are influenced by the preferences of friends [5,6]. Based on this hypothesis, researchers have integrated users' social information into the recommendation system as auxiliary information, which can alleviate the problems of data sparseness and cold start problems, thus forming social recommendations.

Some researchers have attempted to make social recommendations based on graph embedding learning of heterogeneous networks [7–9]. Implicit friends with similar preferences do not have explicit links, but they can be indirectly linked based on what they have

interacted with. Implicit friends can be used to mine more reliable information from sparse data, specifically learning node embeddings that can accurately express user preferences. Mainstream social recommendation models use heterogeneous networks to describe user social relations and user–item interaction relations and then use the graph embedding learning method to obtain user/item node representation. Node embedding can express node attributes or relationships between nodes (such as user preferences). For example, IF-BPR [8] designed “user–item–user”, “user–user–user”, and other multi-meta paths based on domain knowledge to guide a random walk in a heterogeneous network, thus learning high-quality node embedding. MoHINRec [9] used the meta-path of a variety of motif structures (triangular structures that reflect strong connections between nodes) to guide the random walk and obtain more accurate node embedding. In recent years, graph neural networks (GNNs) have achieved considerable success in node classification and link prediction. Owing to their powerful modeling ability in graph relation, GNNs are also applied in the field of recommendation systems. However, there are three challenges in social recommendation based on GNNs: (1) information is not fully mined from existing data as auxiliary information; (2) users’ social relationships have limited ability to alleviate data sparsity; and (3) information is transmitted independently in the user–item interaction graph, and the user social network graph and node embedding are formed independently, whereas user–item interactions and user social relationships do not affect user preference simultaneously. To tackle these challenges, we design a social recommendation model based on self-supervision and multi-auxiliary information.

The main contributions of this paper are as follows:

- We mine item association relationships, user social relationships, and user–item interaction relationships as auxiliary information to alleviate the problem of data sparsity. Unlike the existing social recommendation models of graph neural networks that independently carry out interactive information and social information dissemination, we design a dissemination mode to make multiple auxiliary information affect the formation of user/item embedding simultaneously.
- We design self-supervised auxiliary tasks for the social recommendation scenario to improve the node embedding quality and alleviate data sparsity, construct several views according to different combinations of auxiliary information, and maximize the mutual information of node embedding under different perspectives based on contrastive self-supervised learning.
- We conduct extensive experiments on two public datasets to demonstrate the effectiveness of the proposed model and analyze the benefits of auxiliary information and self-supervised tasks.

2. Related Work

In this section, we introduce graph neural networks and contrastive self-supervised learning.

2.1. Graph Neural Networks (GNNs)

In recent years, GNNs have attracted increasing attention because, owing to their excellent performance on various tasks. Inspired by the success of other fields, such as node classification and link prediction, the applicability of GNNs in recommendation tasks was investigated. In particular, GCN [10] has driven a large number of graph-based neural network recommendation models, such as GCMC [11], NGCF [12], and LightGCN [13]. The basic idea of these GCN-based models is to improve the representation of the target node by aggregating the node representation of the neighbor [14] to obtain higher-order neighbor information in the user–item interaction graph. In addition to these generic models, GNNs support other recommendation methods for specific graphs, such as session and social graphs.

GNNs are often used for information transmission in social networks because information is transmitted in social networks the same as in GNNs. Researchers naturally transplanted GNN into social recommendation work. GraphRec is the first model to

introduce GNN into social recommendation [15]. It learns target node embedding by aggregating first-order neighbor information in the user–item interaction graph and the social network graph. User embedding of DiffNet [10] comes from the social network graph and the user–item interaction graph. It carries out more profound information dissemination in the social network through a multi-layer GNN. Moreover, node embedding in the user–item interaction graph only comes from the first-order neighbor. DiffNet uses a multi-layer GNN structure to realize the dynamic propagation of social influence in the social network. Wu et al. [16] proposed a dual-graph attention network for collaborative learning of node embedding influenced by two layers of society. DGRec uses two circulating neural networks to dynamically simulate user behavior and social influence [17]. Yu et al. [18] enhanced social recommendation with adversarial graph convolutional networks to process complicated high-order interactions among users. Later, they [19] improve social recommendation with a multi-channel hypergraph convolutional network to leverage high-order user relations. Huang et al. [20] proposed a knowledge-aware coupled GNN that injects knowledge across items and users into the recommendation. Yang et al. [21] proposed the ConsisRec model to calculate the consistency score between neighbors as the probability of sampling neighbors and further handle the problem of relationship inconsistency through an attention mechanism.

2.2. Contrastive Self-Supervised Learning

Self-supervised learning was first proposed in the field of robotics, whereby training data are automatically generated from data of two or more sensors. The principle of self-supervised learning can be explained a description of complete data based on observation of different aspects or different parts of data. Self-supervised learning enhances data by deforming, intercepting, and disturbing the original data and generates data as pseudo-labels of the original data to make up for the data deficiency. Self-supervised learning can be divided into two types: generative self-supervised learning and contrastive self-supervised learning. In this paper, the comparative self-supervised learning method is used to provide more auxiliary information to alleviate the problem of data sparsity and realize a recommendation algorithm.

Contrast learning is a discriminant method with the aim of making the embeddings of similar samples (positive samples) closer to each other in the representation space and the embeddings of different samples (negative samples) farther apart. This method uses a similarity measure to quantify the distance between two inserts (commonly known as cosine similarity). Studies based on contrastive self-supervision have made significant progress, such as SwAV [22], MoCo [23], and SimCLR [24], and their extensions, with performances comparable to those of related models based on supervised learning.

Pseudo-label construction is an essential strategy for embedded learning based on contrastive self-supervised learning. Positive samples and negative samples are essentially pseudo-labels of data to expand training data. The original samples are based on positive/negative samples for supervised learning. The proposed goal of self-supervised learning is to reduce the cost of manual labeling, so the generation of false labels (the selection of positive/negative samples) is an automatic process.

Based on contrastive self-supervised learning, the model needs to design auxiliary tasks to complete contrastive learning and assist with the main tasks of specific scenes to train the model. The training process is as follows:

- The auxiliary task performs data enhancement based on the original sample. The original and positive/negative samples generate corresponding low-dimensional embedding through the encoder and construct the loss function through contrast learning.
- The main task generates the corresponding low-dimensional embedding of the original sample through the encoder and constructs the principal loss function through the main task. Finally, by comparing the loss function with the main loss function, higher quality embedding can be obtained.

Contrastive self-supervised learning is also applied flexibly in graph embedding learning (graph contrast learning for short). This work mainly constructs self-supervised signals from graph structures of different perspectives to explore higher-quality graph structure embedding [25,26]. Generally, a new perspective can be obtained through random data enhancement of the same graph. Common data enhancement methods include but are not limited to the random deletion of nodes, random deletion of edges, the random transformation of features or attributes, random walk-based pattern graphs, etc. Inspired by such work, some researchers began to apply graph contrast learning to recommendation tasks [27–30]. Zhou et al. [30] designed self-supervised auxiliary tasks, specifically adding random embedding masks for item embedding and randomly skipping given items and sub-sequences to carry out pre-training for sequence recommendation. Yao et al. [29] proposed a two-tower network structure based on DNNs (deep neural networks), on which random feature mask and random feature discarding operations were carried out for self-supervised item recommendations. Ma et al. [28] reconstructed short-term future sequences by observing long-term sequences, which essentially mined more self-supervised signals using feature masks. Wu et al. [27] summarized the above random data enhancement operations (random node/edge deletion and random walk) and integrated them into a recommendation framework based on self-supervised graph learning. Long et al. [31] proposed a heterogeneous graph neural network based on meta-relation and used self-supervised learning to guide the interaction between users and items under different views, incorporating the knowledge information of the item and the high-level semantic relationship between users and items into the user representation. Liu et al. [32] designed two new information augmentation methods. Furthermore, they proposed a contrastive self-supervised learning framework, CoSeRec, for sequence recommendation, which alleviates the problems of data sparsity and noisy interaction issues. Wu et al. [33] proposed a social recommendation model that disentangles the collaborative domain and social domain to learn user representations separately and uses cross-domain contrastive learning to further improve the recommendation performance.

Unlike these models, we use item-association-aware contrastive learning in self-supervised learning. We maximize the mutual information of node embedding in two item-association-aware views in self-supervised learning; view 1 is constructed by the user–item interaction relationship and the item association relationship, and view 2 is formed by the user–item interaction relationship and the user social relationship. Although the model proposed in [33] also uses cross-domain contrastive learning, the domains are the user–item interaction domain and the social relationship domain, not taking item association into consideration. Our model is designed to utilize more auxiliary information to improve recommendation performance, and self-supervised learning can finetune the embeddings from different kinds of auxiliary information. There are two main differences between our model and SEPT [34]. First, we use item-association-aware contrastive learning in self-supervised learning, whereas SEPT does not take item association into consideration. Second, SEPT focuses more on finding positive samples for contrastive learning and uses tri-training to determine which samples are positive; however, our model does not attempt to identify additional positive samples and treat the same node in different views as a positive sample.

3. Problem Analysis

Accurately capturing user preferences is the key to improving the recommendation quality of recommendation systems. The recommendation algorithm is committed to learning more accurate user embedding to improve the recommendation performance. The current social recommendation algorithm based on a graph neural network believes that user preference is jointly determined by items that have interacted historically, as well as the influence of social friends. The process of learning user/item embedding in this kind of algorithm is summarized as follows: In the first step, by default, user–item interaction influence and social influence are spread in an independent scope, and information is

spread and embedded in the user–item interaction graph and user social network graph. In the second step, there are two different embeddings of user nodes in the two graphs. The two embeddings representing user–item interaction information and social influence are aggregated to form the final embeddings.

Social recommendations based on graph neural networks are subject to the following problems:

- They carry out user–item interaction information and social information dissemination in two graphs instead of being affected by both simultaneously (more realistic).
- As auxiliary information, social relationship alleviates the problem of data sparsity in the recommendation system, but its role is limited, and more auxiliary information is needed.

4. Methodology

To alleviate the above problems, we propose a social recommendation model based on self-supervised and multi-auxiliary information (SlightGCN). We combine user social relationships, item association relationships, and self-supervised auxiliary signals as auxiliary information to carry out social recommendations. First, the model SlightGCN mines item association relationships based on user–item interaction records and item attributes. Then, a heterogeneous network is constructed, including user social relationships, item association relationships, and user–item interaction relationships. In the heterogeneous network graph containing multi-auxiliary information, information is transmitted and output-embedded through a GCN. The user/item embedding containing rich information is used to recommend the main task. In addition, two heterogeneous networks with different perspectives are constructed according to different combinations of existing auxiliary information, and self-supervised auxiliary tasks are constructed to maximize the mutual information of nodes under different perspectives to obtain higher quality node embedding. Finally, the model is trained through the combination of primary task (recommendation task) loss and auxiliary task (contrastive self-supervised learning).

SlightGCN can be generally divided into three parts: the heterogeneous network construction, main recommendation task, and self-supervision auxiliary task (See Figure 1).

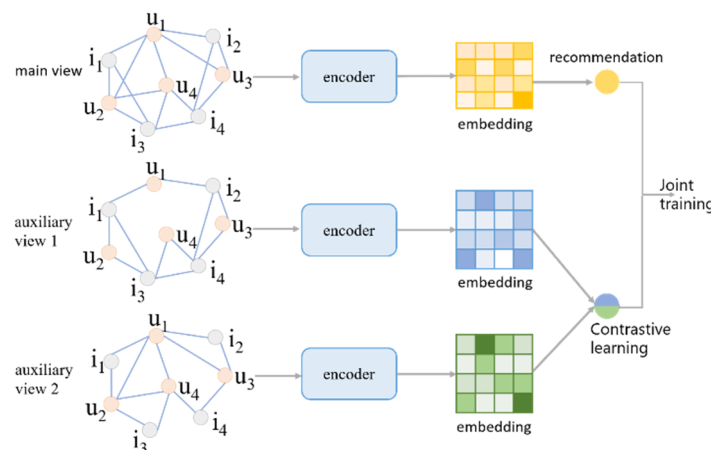


Figure 1. The overall framework of SlightGCN.

4.1. Heterogeneous Network Construction

First, the association relationship is mined from the user–item interaction relationship and item attribute through the meta-path rule to alleviate the data sparsity problem. Secondly, three types of information (user social relationship, user–item interaction relationship, and item association relationship) are integrated into the same heterogeneous network in the form of edges. The construction process of a heterogeneous network is shown in Figure 2.

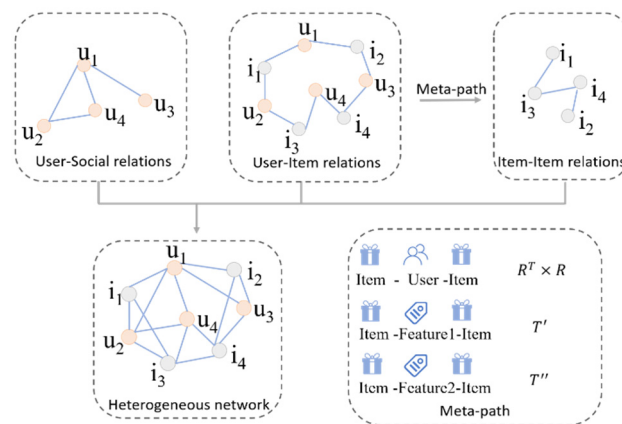


Figure 2. The heterogeneous network construction process.

In the process of information dissemination of the recommendation scenario, the user node is influenced by friends and items that have been interacted with, and the item node is influenced by items that are closely connected and users that have been interacted with to learn user/item embedding of higher quality. To realize this information transmission mode, user–item interaction relationship, user social relationship, and item association relationship need to be integrated into the same heterogeneous network in the form of an edge. The user node receives information from friends and historical interaction records through user–item interaction relationships and user social relationships, and the item node receives information from closely related items and users who have interacted with each other through user–item interaction relationships and item association relationships.

4.2. Main Tasks

A recommendation algorithm is based on the main task of supervised learning to optimize model parameters. The prediction is established on user embedded (e_u) and item-embedded (e_i). Generally, the inner product of the embedded $e_u^T e_i$ is used to predict user’s (u) preference degree (\hat{y}_{ui}) for item i , as in (1):

$$\hat{y}_{ui} = e_u^T e_i. \tag{1}$$

Specifically, it assumes the observed interactions as monitoring signals and makes the predicted preference (\hat{y}_{ui}) as close as possible to the real preference (y_{ui}). It takes the unobserved data as a negative sample. We take the generally used Bayesian personalized recommendation [35] (BPR) as the loss function for the ranking recommendation task. The core idea is that the target user (u) prefers interactive item i (observable data) to uninteractive item j (unobserved data). The specific loss function is shown in (2):

$$\mathcal{L}_{main} = \sum_{(u,i,j) \in \mathcal{D}} -\log(\hat{y}_{ui} - \hat{y}_{uj}), \tag{2}$$

where $\mathcal{D} = \{(u, i, j) | (u, i) \in \mathcal{D}^+, (u, j) \in \mathcal{D}^-\}$ is all training data, \mathcal{D}^+ is the set of observable interactions, and \mathcal{D}^- is the set of unobserved interactions.

The basis of the recommendable task is to learn user/item embedding. Heterogeneous networks can output user/item embedding through a graph encoder. Inspired by the excellent graph convolution neural network recommendation model LightGCN [13], feature transformation, nonlinear transformation, and self-connection operation are redundant for the collaborative filtering recommendation model. We integrate user social relations and item associations to realize that user nodes are affected by social friends and interactive items simultaneously, and item nodes are affected by interactive users and closely connected items simultaneously.

In graph convolution mode, simple weighted aggregation is sampled in the convolution process and feature transformation. The specific convolution operation is shown in (3):

$$e_u^{(k+1)} = \sum_{q \in N_u} \frac{1}{\sqrt{|N_u||N_p|}} e_p^{(k)}, e_i^{(k+1)} = \sum_{q \in N_i} \frac{1}{\sqrt{|N_i||N_q|}} e_q^{(k)}, \tag{3}$$

where $e_u^{(k+1)}$ and $e_i^{(k+1)}$ represent the embedment of user u and item i at the $(k + 1)$ layer, respectively; and the $(k + 1)$ layer embedment is aggregated from the k layer embedment. Taking the formation of $e_u^{(k+1)}$ as an example, user u 's neighbor, N_u , contains the user's neighbor and item neighbor, and $e_u^{(k+1)}$ is aggregated by embedding the k layer of user u 's neighbor, N_u . $e_i^{(k+1)}$ Similarly, the neighbor N_i of item i contains user neighbors and item neighbors. $e_i^{(k+1)}$ is aggregated by embedding the k layer of item i 's neighbor, N_i .

After the multi-layer convolution, the final node embedding is obtained by average aggregation of each layer embedding of the node, as shown in (4):

$$e_u = \frac{e_u^{(0)} + e_u^{(1)} + \dots + e_u^{(k)}}{k + 1}, e_i = \frac{e_i^{(0)} + e_i^{(1)} + \dots + e_i^{(k)}}{k + 1}. \tag{4}$$

In the graph matrix representation of the convolution process, the heterogeneous network adjacency matrix ($A \in \mathbb{R}^{(N+M) \times (N+M)}$) is composed of the user-item interaction matrix (R) and its transpose (R^T), the user social matrix (S), and the item association matrix (T). The specific expression of adjacency matrix A is shown in (5):

$$A = \begin{pmatrix} S & R \\ R^T & T \end{pmatrix}, \tag{5}$$

User embedding and item embedding constitute the 0-layer embedding matrix, $E^{(0)} \in \mathbb{R}^{(N+M) \times d}$, where d is the embedding dimension, and $E^{(0)}$ is randomly initialized. The matrix form of the convolution operation is shown in (6),

$$E^{(k+1)} = \left(D^{-1/2} A D^{-1/2} \right) E^{(k)}, \tag{6}$$

where $D \in \mathbb{R}^{(N+M) \times (N+M)}$ is a diagonal matrix, and the value of D_{ii} is the number of non-zero elements in the i th row of matrix A .

Finally, the embedding matrix ($E \in \mathbb{R}^{(N+M) \times d}$) is evenly aggregated by each embedding matrix, as shown in (7).

$$E = \frac{E^{(0)} + E^{(1)} + \dots + E^{(k)}}{k + 1}. \tag{7}$$

4.3. Self-Supervised Auxiliary Tasks

The main view composed of three types of relationships (user-item interaction relationship, user social relationship, and item association relationship) introduced in the previous section helps to recommend the embedding of main task learning nodes. When a designer models a character, it is necessary to observe the character information from different aspects and combine the information from multiple perspectives to build a more accurate character model. In the recommendation scenario, the recommendation algorithm needs to model the user and the item before recommending the user to build a more accurate portrait of the person/item (user/item embedding). The graph neural network plays the role of a model and can generate more accurate node embedding by combining node information from different perspectives.

Based on the main view, we remove the user's social relationship and item association relationship and generate auxiliary view 1 and auxiliary view 2, respectively. In the process of information transmission, the two auxiliary views correspond to the adjacency matrices

A_1 and A_2 , respectively, as shown in (8). The convolution of the two auxiliary views is to generate two groups of different embedded nodes (E_1 and E_2), as shown in (9).

$$A_1 = \begin{pmatrix} 0 & R \\ R^T & T \end{pmatrix}, A_2 = \begin{pmatrix} S & R \\ R^T & 0 \end{pmatrix}, \tag{8}$$

$$E_1^{(k+1)} = \left(D_1^{-1/2} A_1 D_1^{-1/2}\right) E_1^{(k)}, E_2^{(k+1)} = \left(D_2^{-1/2} A_2 D_2^{-1/2}\right) E_2^{(k)}. \tag{9}$$

Contrastive self-supervised learning is used to maximize the mutual information of node embedding in different views to learn node embedding of higher quality. Specifically, the node embedding of the same node in different views is a positive sample pair (i.e., $\{e'_u, e''_u\} | u \in U\}$), and the node embedding of different nodes in different views is a negative sample pair (i.e., $\{e'_u, e''_v\} | u, v \in U, u \neq v\}$). Auxiliary tasks make similar nodes as similar as possible, whereas different sample nodes are embedded as far away as possible. The self-supervised loss function is formed according to the comparative loss, InfoNCE, as shown in (10):

$$\mathcal{L}_{ssl}^{user} = \sum_{u \in U} -\log \frac{\exp(s(e'_u, e''_u) / \tau)}{\sum_{v \in U} \exp(s(e'_u, e''_v) / \tau)}, \mathcal{L}_{ssl}^{item} = \sum_{i \in I} -\log \frac{\exp(s(e'_i, e''_i) / \tau)}{\sum_{j \in I} \exp(s(e'_i, e''_j) / \tau)}, \tag{10}$$

$$\mathcal{L}_{ssl} = \mathcal{L}_{ssl}^{user} + \mathcal{L}_{ssl}^{item}, \tag{11}$$

where the function $s(\cdot)$ is cosine similarity, which is used to measure the distance between two embeddings, and the hyperparameter τ is the temperature index, which can reduce or amplify the effect of distance between nodes. Finally, as shown in (11), the loss function (\mathcal{L}_{ssl}) of the self-supervision auxiliary task is the sum of the loss function (\mathcal{L}_{ssl}^{user}) of the user node and the loss function (\mathcal{L}_{ssl}^{item}) of the item node.

To improve the recommendation performance, the main recommendation task and self-supervised auxiliary task are combined for training. The loss function of the combined training is shown in (12), where λ_1 and λ_2 are the hyperparameters, and θ is the model parameter.

$$\mathcal{L} = \mathcal{L}_{main} + \lambda_1 \mathcal{L}_{ssl} + \lambda_2 \|\theta\|_2^2. \tag{12}$$

5. Experiments

In this study, extensive experiments were carried out on two public datasets to verify the following points: (1) the advantages of SlightGCN in terms of recommendation performance, (2) that SlightGCN effectively alleviates data sparsity and cold start problems; (3) that multi-auxiliary information (users' social relationship, item association relationship, and self-supervised learning signal) plays an essential role in improving the recommendation performance, and (4) the influence of hyperparameters on SlightGCN.

5.1. The Datasets

We conducted extensive experiments on two public datasets: DoubanMovie and DoubanBook. Detailed information on the datasets is shown in Table 1, including the number of users, the number of items, the number of user–item interactions, the number of user social relationships, and the density of user–item interactions.

Table 1. Dataset statistics.

The Dataset	Number of Users	Number of Items	Number of User–Item Interactions	Number of Social Connections	Density of Interactions
DoubanMovie	13,367	12,677	1,067,278	4085	0.63%
DoubanBook	13,024	22,347	792,062	84,575	0.27%

DoubanMovie is a movie dataset of the Douban platform, which contains 13,367 users' 1,067,278 viewing behaviors on 12,677 movies and 4085 users' social friend relationships. We select two crucial attributes of film type and director in DoubanMovie to excavate the social relations of objects. DoubanBook is a book dataset of the Douban platform, which contains 792,062 interactions between 13,024 users and 22,347 books and 169,150 social friend relationships between users. Two important attributes of DoubanBook (publisher and author) are selected to mine the social relations. In this study, the dataset is divided into a training set, a validation set, and a test set (70%, 10%, and 20%, respectively). The model was cross-verified ten times, and the average value was taken as the result.

5.2. Baselines and Metrics

To verify the recommendation performance of SlightGCN, we select six baseline algorithms, as shown below:

- BPR: Bayesian personalized ranking [35], a classical sorting recommendation algorithm. Based on user–item interaction information, it is assumed that the target user prefers interactive items rather than non-interactive items.
- SBPR: A classic social recommendation algorithm [36] that integrates social relations to optimize the item preference priority of target users based on BPR.
- DiffNet: A social recommendation algorithm based on a graph neural network [37] that simulates the social influence of dynamic propagation in user social networks.
- LightGCN: A recommendation algorithm based on a graph convolutional neural network [13] that learns node embedding in a simple convolution mode suitable for collaborative filtering.
- SGL: A recommendation algorithm based on self-supervised learning and a graph convolution neural network [27] that creates multiple views by randomly changing the graph structure to improve the embedding quality.
- SEPT: A graph convolution neural network recommendation algorithm based on self-supervised learning and collaborative training [34] that finds more suitable positive/negative samples for self-supervised learning through collaborative training to learn more accurate embedding.

Evaluation metrics are precision, recall, $F1$, and NDCG (normalized impairment cumulative gain).

$$Prec@k = \frac{\sum u \in U |R(u) \cap T(u)|}{k}, \quad (13)$$

$$Rec@k = \frac{\sum u \in U |R(u) \cap T(u)|}{|T(u)|}, \quad (14)$$

$$F1 = \frac{2 * Prec * Rec}{Prec + Rec}, \quad (15)$$

$$NDCG@k = \frac{DCG}{iDCG}, \quad (16)$$

$$DCG = \sum_i^k \frac{2^{rel_i-1}}{\log_2(i+1)}, \quad (17)$$

where u is a user; U is the set of all users; $R(u)$ denotes recommended items for user u ; $T(u)$ is user u 's real liked items; rel_i represents item i 's relevance score, which can be predefined; and $iDCG$ is the DCG of the ideal order for the recommendation set.

In the experiments, the depth of LightGCN, SGL, and SlightGCN for information propagation is three, and that of SEPT is two because SEPT can achieve the best performance in that case [34].

5.3. Results

5.3.1. Overall Comparison

We compare the proposed model, SlightGCN, with six baselines on DoubanMovie and DoubanBook. The advantages and disadvantages of each model are shown by comparing the four metrics in top-10 and top-20 recommendations. Based on the experimental results in Tables 2 and 3 (the bold numbers mean the best performance), the following conclusions can be drawn:

1. SlightGCN's recommendation performance is significantly better than that of the other five baseline models on the two datasets. Specifically, the evaluation metrics of SlightGCN on DoubanMovie and DoubanBook improved by 1.84–4.26% and 2.08–3.30%, respectively, compared with the suboptimal model. In addition, the interactive data density of the DoubanMovie dataset is higher than that of DoubanBook. Thus, the recommendation performance of all models on the DoubanMovie dataset is significantly better than that of DoubanBook.
2. The results show that SBPR is superior to BPR in some indicators, indicating that SBPR can alleviate data sparsity to a certain extent by integrating users' social relationships based on BPR. SBPR's simple approach of integrating users' social relationships by directly optimizing the order of items does not accurately simulate the impact of social relationships on users' preferences. DiffNet enables the influence of friends' preferences on users to propagate dynamically in social networks through multi-layer graph neural network simulation. The performance of DiffNet is better than that of SBPR owing to the ability of the graph neural network to capture graph relation and the dynamic propagation of social influence. LightGCN's simple convolution mode makes it more suitable for collaborative filtering, and its recommendation performance is significantly higher than that of the previous recommendation model. Based on LightGCN as a graph encoder, collaborative training is used to find more suitable positive and negative samples for self-supervised learning, and SEPT improves the recommendation performance based on LightGCN. Using LightGCN as the graph encoder and designing the self-supervised auxiliary task based on random graph structural disturbance, the SGL recommendation performance was considerably improved.
3. SlightGCN performs best on both datasets. SlightGCN has the following advantages over other models. First, it not only uses user–item interaction relationships and user–social relationships but also mines item association relationships from existing information to form multi-auxiliary information. Second, a more appropriate convolution mode is designed so that users are affected by both user–item interaction relationships and user–social relationships, and objects are affected by both user–item interaction relationships and item association relationships. Third, multi-view self-supervised auxiliary tasks are designed to learn more accurate user/item embedding.

Table 2. Comparison of recommendation performances on DoubanMovie.

	BPR	SBPR	DiffNet	SEPT	LightGCN	SGL	SlightGCN	Ascension
Prec@10	0.1243	0.1175	0.1311	0.1617	0.1602	0.1642	0.1712	4.26%
Prec@20	0.1049	0.0994	0.1119	0.1343	0.1331	0.1352	0.1407	4.07%
Rec@10	0.0784	0.0903	0.1012	0.1218	0.1186	0.1252	0.1281	2.32%
Rec@20	0.1281	0.1426	0.1647	0.1862	0.1833	0.1890	0.1931	2.17%
F1@10	0.0962	0.1021	0.1142	0.1389	0.1363	0.1421	0.1465	3.10%
F1@20	0.1153	0.1171	0.1333	0.1560	0.1542	0.1576	0.1628	3.30%
NDCG@10	0.1493	0.1480	0.1642	0.2061	0.2034	0.2110	0.2166	2.65%
NDCG@20	0.1498	0.1517	0.1706	0.2070	0.2043	0.2122	0.2161	1.84%

Table 3. Comparison of recommendation performances on DoubanBook.

	BPR	SBPR	DiffNet	SEPT	LightGCN	SGL	SlightGCN	Ascension
Prec@10	0.0715	0.0664	0.0727	0.0942	0.0876	0.1061	0.1096	3.30%
Prec@20	0.0561	0.0539	0.0583	0.0743	0.0702	0.0826	0.0850	2.91%
Rec@10	0.0696	0.0725	0.0830	0.1008	0.0909	0.1079	0.1104	2.32%
Rec@20	0.1027	0.1093	0.1239	0.1480	0.1372	0.1587	0.1620	2.08%
F1@10	0.0705	0.0693	0.0775	0.0974	0.0892	0.1070	0.1100	2.80%
F1@20	0.0726	0.0722	0.0793	0.0989	0.0929	0.1086	0.1115	2.67%
NDCG@10	0.0956	0.0921	0.1019	0.1320	0.1195	0.1496	0.1541	3.01%
NDCG@20	0.0980	0.0980	0.1090	0.1371	0.1261	0.1532	0.1577	2.94%

5.3.2. Cold-Start User Experiment

Cold-start users have few or no interactions, and extremely sparse data make it difficult to generate recommend for such users. In this section, users with fewer than 10 interactions are selected as cold-start users; the recommendation results for these users are shown in Figures 3 and 4.

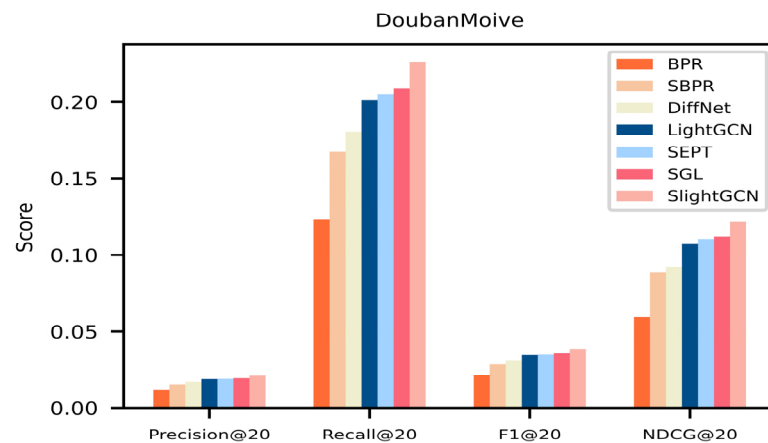


Figure 3. Comparison of the cold-start user recommendation performances on DoubanMovie.

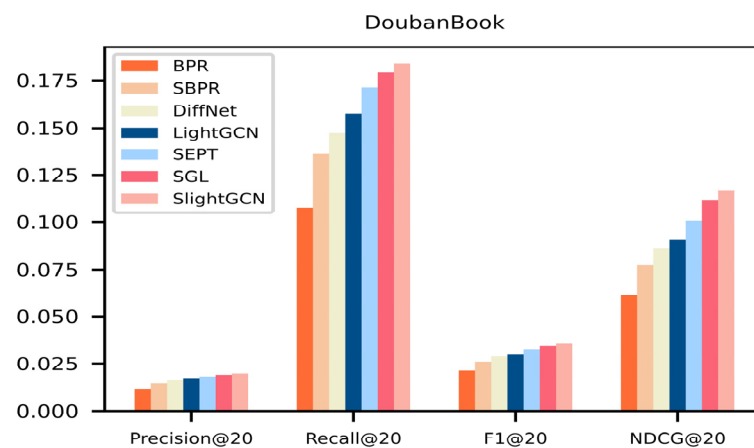


Figure 4. Comparison of the cold-start user recommendation performances on DoubanBook.

The recommendation performance of SBPR is better than that of BPR, indicating that users’ social relationships alleviate the problem of data sparsity to some extent. For the other four recommendation models based on GNN, the recommendation performance increases in the order of DiffNet, LightGCN, SEPT, SGL, and SlightGCN. LightGCN, which only uses user–item interaction relationships, achieves a better recommendation performance than DiffNet, which uses the social relationship as auxiliary information. Its simple and efficient

convolution structure is more suitable for collaborative filtering. SEPT uses collaborative training to find positive and negative samples for self-supervised learning. SGL is based on the structure of a random disturbance graph for data enhancement, and the generated self-supervised signal alleviates data sparsity and improves the embedding quality. The proposed SlightGCN model incorporates multiple pieces of auxiliary information, such as user social relationship and item association relationship, as well as various perspectives of self-supervised auxiliary task construction, all of which alleviate the problem of data sparsity to a certain extent. In addition, user/item embedding is influenced by multiple relationships simultaneously, improving the capture of user preferences.

5.3.3. The Benefit of Self-Supervised Auxiliary Tasks

To verify that user social relationship, item association relationship, and self-supervised auxiliary tasks can help improve the recommendation performance of SlightGCN, we designed an ablation experiment as follows.

To verify the effectiveness of self-supervised auxiliary tasks, the self-supervised auxiliary tasks were removed to form the UI model variant. In addition, to verify the benefits of user social relationship and item association relationship to improve the model performance, user–item interaction relationship and user social relationship are reserved to form the U model variant, and user–item interaction relationship and item association relationship are reserved to form the I model variant. The experimental results of SlightGCN and its three deformation models on the two datasets are shown in Tables 4 and 5 (the bold numbers mean the best performance).

Table 4. SlightGCN ablation experimental results on DoubanMovie.

Metrics	LightGCN	Variant-U	Variant-I	Variant-UI	SlightGCN
Prec@10	0.1602	0.1635	0.1630	0.1637	0.1712
Prec@20	0.1331	0.1348	0.1343	0.1350	0.1407
Rec@10	0.1186	0.1217	0.1223	0.1226	0.1281
Rec@20	0.1833	0.1876	0.1869	0.1881	0.1931
F1@10	0.1363	0.1395	0.1397	0.1402	0.1465
F1@20	0.1542	0.1569	0.1563	0.1572	0.1628
NDCG@10	0.2034	0.2091	0.2082	0.2095	0.2166
NDCG@20	0.2043	0.2101	0.2096	0.2104	0.2161

Table 5. SlightGCN ablation experimental results on DoubanBook.

Evaluation Index	LightGCN	Variant U	Variant I	Variant UI	SlightGCN
Prec@10	0.0876	0.0890	0.0893	0.0897	0.1096
Prec@20	0.0702	0.0712	0.0719	0.0726	0.0850
Rec@10	0.0909	0.0927	0.0934	0.0941	0.1104
Rec@20	0.1372	0.1386	0.1390	0.1389	0.1620
F1@10	0.0892	0.0908	0.0913	0.0918	0.1100
F1@20	0.0929	0.0941	0.0948	0.0954	0.1115
NDCG@10	0.1195	0.1223	0.1226	0.1238	0.1541
NDCG@20	0.1261	0.1285	0.1290	0.1296	0.1577

Compared with LightGCN, variant UI, variant U, and variant I, the three deformation models all improved to a certain extent, indicating that user social relationship and item association relationship can improve the recommendation performance by alleviating data sparsity. However, there is no significant difference in the recommendation performance between variant UI containing two auxiliary relationships and variant U and variant I, containing only one auxiliary relationship, indicating that the user social relationship and item association relationship cannot be well-integrated. SlightGCN removes the user social relationship and the item social relationship to construct different auxiliary views. Contrastive self-supervised learning maximizes the mutual information of the two views

and promotes the integration of user social relationships and item association relationships. SlightGCN’s recommendation performance is superior to that of variant UI, which does not include self-supervised assistance tasks.

5.3.4. Parameter Sensitive Analysis

In the training process of the model, it is constrained by the main recommendation task and the self-supervised auxiliary task. The self-supervised task assists the recommendation task to guide the updating direction of model parameters. In the loss function of the joint training, the hyperparameter λ_1 is the weight coefficient of the loss of the self-supervised auxiliary task, and λ_1 can control the influence of the self-supervised auxiliary task on the overall recommendation performance of the model. The training process of the parameter λ_1 ambassador model is more dependent on the information of the self-supervised auxiliary task and less dependent on the information of the self-supervised auxiliary task. The experiments of hyperparameter λ_1 on dataset DoubanMovie and dataset DoubanBook are shown in Figures 5 and 6, respectively.

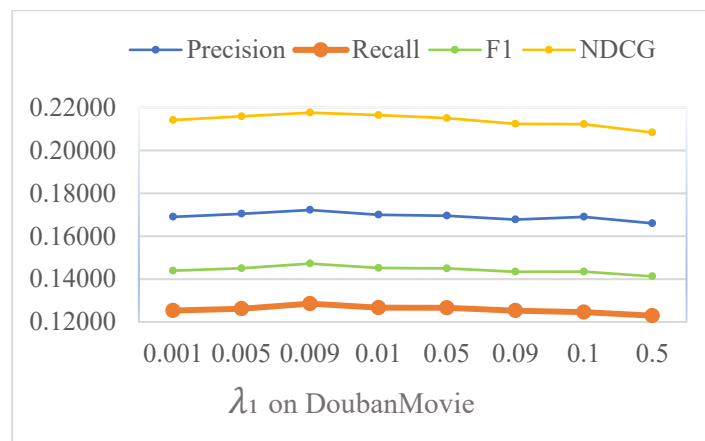


Figure 5. Parameter λ_1 sensitivity analysis on DoubanMovie.

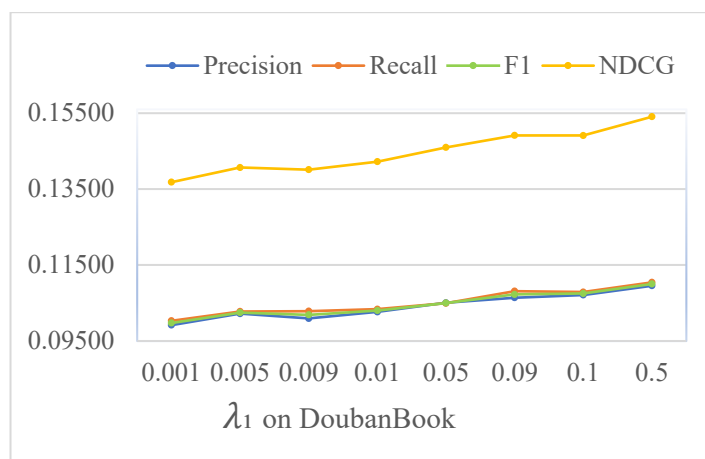


Figure 6. Parameter λ_1 sensitivity analysis on DoubanBook.

In DoubanMovie, the overall trend of the values of the four metrics is consistent, increasing first and then decreasing with increased λ_1 , with the recommendation performance reaching its peak at $\lambda_1 = 0.009$. In DoubanBook, the overall variation trend of the four metrics is consistent, increasing with increased λ_1 , with the recommendation performance reaching its peak at $\lambda_1 = 0.05$. Two conclusions can be drawn from observations:

- The optimal $\lambda_1 = 0.009$ on DoubanMovie is far less than the optimal $\lambda_1 = 0.05$ on DoubanBook, indicating that the model relies much more on self-supervised auxiliary

tasks on DoubanBook than on DoubanMovie. The reasons for the above phenomenon are as follows: the user–item interaction density on DoubanBook is lower than that on DoubanMovie. To learn higher-quality node embedding and improve recommendation performance, the model in the DoubanBook environment needs to obtain more information from self-supervised auxiliary tasks.

- The improvement of the four metrics in DoubanBook with sparse data is much more significant than that in DoubanMovie, indicating the effectiveness of self-supervised auxiliary tasks in alleviating the data sparsity problem.

6. Conclusions

In this paper, we propose a social recommendation model based on self-supervision and multi-auxiliary information (SlightGCN). The social relationships between users and the related relationships between items are mined from user–item interaction data and item attribute information. The relationships between items and users can be used as multiple auxiliary information to alleviate data sparsity. The user–item interaction relationship, user social relationship, and item associate relationship are integrated into the same heterogeneous network in the form of an edge. A heterogeneous network is used as the input of a graph convolutional neural network to carry out the main recommendation task, and an appropriate convolutional mode is designed; user embedding is simultaneously affected by user social relationship and user–item interaction relationship, and item embedding is simultaneously affected by item association relationship and user–item interaction relationship. Auxiliary views are constructed according to different combinations of information, and self-supervised auxiliary tasks of multi-auxiliary views are designed to improve the embedding quality of nodes and recommendation performance. Extensive experiments were conducted on two public datasets to verify the superiority of SlightGCN in terms of recommendation performance, the effectiveness of multi-auxiliary information, and self-supervised auxiliary tasks.

Many data argumentation methods for comparative self-supervised learning are available in the GNN-based method, such as randomly removing edges or nodes, randomly adding edges or nodes, removing/adding edges or nodes on purpose, and even rewiring the graph. Further research is required with deep analysis of the effectiveness of these data argumentation methods within the framework the proposed SlightGCN model.

Author Contributions: Conceptualization, F.J. and Y.C.; methodology, M.G. and Y.C.; software, Y.C.; validation, H.W. and X.W.; formal analysis, Y.S.; writing—original draft preparation, F.J.; writing—review and editing, Y.S.; supervision, M.G. All authors have read and agreed to the published version of the manuscript.

Funding: This study was supported by the National Natural Science Foundation (72161005), the Research Program of Chongqing Technology Innovation and Application Development (cstc2020kjjsx-phxm1304), Chongqing Technology Innovation and Application Development Project (cstc2020jcsx-lyjsAX0010), and Scientific and Technological Research Program of Chongqing Municipal Education Commission (KJZD-K202204402).

Data Availability Statement: DoubanMovie: http://shichuan.org/dataset/Semantic_Path.zip (accessed on 12 January 2022).; DoubanBook: <http://shichuan.org/dataset/Recommendation.zip> (accessed on 12 January 2022).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Aljunid, M.F.; Huchaiyah, M.D. An efficient hybrid recommendation model based on collaborative filtering recommender systems. *CAAI Trans. Intell. Technol.* **2021**, *6*, 13.
2. Meng, X.W.; Liu, S.D.; Zhang, Y.J.; Hu, X. Research on Social Recommender Systems. *J. Softw.* **2015**, *26*, 1356–1372.
3. Zhang, Y.J.; Dong, Z.; Meng, X.W. Research on personalized advertising recommendation systems and their applications. *J. Comput.* **2021**, *44*, 531–563.
4. Shang, M.; Luo, X.; Liu, Z.; Chen, J.; Yuan, Y.; Zhou, M. Randomized latent factor model for high-dimensional and sparse matrices from industrial applications. *IEEE/CAA J. Autom. Sin.* **2018**, *6*, 131–141. [[CrossRef](#)]
5. Sinha, R.R.; Swearingen, K. Comparing recommendations made by online systems and friends. *DELOS 2001*, *106*. Available online: https://www.researchgate.net/publication/2394806_Comparing_Recommendations_Made_by_Online_Systems_and_Friends (accessed on 28 April 2022).
6. Iyengar, R.; Han, S.; Gupta, S. *Do Friends Influence Purchases in a Social Network?* Harvard Business School Marketing Unit Working Paper No. 09-123; Harvard Business School: Boston, MA, USA, 2009.
7. Kang, M.; Bi, Y.; Wu, Z.; Wang, J.; Xiao, J. A heterogeneous conversational recommender system for financial products. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019; pp. 26–30.
8. Yu, J.; Gao, M.; Li, J.; Yin, H.; Liu, H. Adaptive implicit friends identification over heterogeneous network for social recommendation. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management, Torino, Italy, 22–26 October 2018; pp. 357–366.
9. Zhao, H.; Zhou, Y.; Song, Y.; Lee, D.L. Motif enhanced recommendation over heterogeneous information network. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019.
10. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. In Proceedings of the 5th International Conference on Learning Representations (ICLR 2017), Toulon, France, 24–26 April 2017.
11. Berg, R.; Kipf, T.N.; Welling, M. Graph convolutional matrix completion. In Proceedings of the 24th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2018), London, UK, 19–23 August 2018.
12. 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, Paris, France, 21–25 July 2019; pp. 165–174.
13. 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, Virtual Event, China, 25–30 July 2020; pp. 639–648.
14. Wu, S.; Sun, F.; Zhang, W.; Xie, X.; Cui, B. Graph neural networks in recommender systems: A survey. *arXiv* **2011**, arXiv:2011.02260. [[CrossRef](#)]
15. Fan, W.; Ma, Y.; Li, Q.; He, Y.; Zhao, E.; Tang, J.; Yin, D. Graph neural networks for social recommendation. In Proceedings of the World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019; pp. 417–426.
16. Wu, Q.; Zhang, H.; Gao, X.; He, P.; Weng, P.; Gao, H.; Chen, G. Dual graph attention networks for deep latent representation of multifaceted social effects in recommender systems. In Proceedings of the World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019; pp. 2091–2102.
17. Song, W.; Xiao, Z.; Wang, Y.; Charlin, L.; Zhang, M.; Tang, J. Session-based social recommendation via dynamic graph attention networks. In Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, Melbourne, VIC, Australia, 11–15 February 2019; pp. 555–563.
18. Yu, J.; Yin, H.; Li, J.; Gao, M.; Huang, Z.; Cui, L. Enhance social recommendation with adversarial graph convolutional networks. *IEEE Trans. Knowl. Data Eng.* **2020**, *34*, 3727–3739. [[CrossRef](#)]
19. Yu, J.; Yin, H.; Li, J.; Wang, Q.; Hung, N.Q.; Zhang, X. Self-supervised multi-channel hypergraph convolutional network for social recommendation. In Proceedings of the Web Conference 2021, Ljubljana, Slovenia, 19–23 April 2021; pp. 413–424.
20. Huang, C.; Xu, H.; Xu, Y.; Dai, P.; Xia, L.; Lu, M.; Bo, L.; Xing, H.; Lai, X.; Ye, Y. Knowledge-aware coupled graph neural network for social recommendation. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual Conference, 2–9 February 2021; Volume 35, pp. 4115–4122.
21. Yang, L.; Liu, Z.; Dou, Y.; Ma, J.; Yu, P.S. Consisrec: Enhancing GNN for social recommendation via consistent neighbor aggregation. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, 11–15 July 2021; pp. 2141–2145.
22. Blau, Y.; Michaeli, T. Rethinking lossy compression: The rate-distortion-perception tradeoff. In Proceedings of the International Conference on Machine Learning, PMLR, Long Beach, CA, USA, 10–15 June 2019; pp. 675–685.
23. Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T. Enriching word vectors with subword information. *Trans. Assoc. Comput. Linguist.* **2017**, *5*, 135–146. [[CrossRef](#)]
24. Brock, A.; Donahue, J.; Simonyan, K. Large Scale GAN Training for High Fidelity Natural Image Synthesis. ICLR2019. Available online: <https://openreview.net/forum?id=B1xsqj09Fm> (accessed on 28 April 2022).
25. Qiu, J.; Chen, Q.; Dong, Y.; Zhang, J.; Yang, H.; Ding, M.; Wang, K.; Tang, J. Gcc: Graph contrastive coding for graph neural network pre-training. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Virtual Event, CA, USA, 6–10 July 2020; pp. 1150–1160.

26. Velickovic, P.; Fedus, W.; Hamilton, W.L.; Liò, P.; Bengio, Y.; Hjelm, R.D. Deep Graph Infomax. *ICLR (Poster)* **2019**, *2*, 4.
27. 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, Virtual Event, Canada, 11–15 July 2021; pp. 726–735.
28. Ma, J.; Zhou, C.; Yang, H.; Cui, P.; Wang, X.; Zhu, W. Disentangled self-supervision in sequential recommenders. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Virtual Event, CA, USA, 6–10 July 2020; pp. 483–491.
29. Yao, T.; Yi, X.; Cheng, D.Z.; Felix, X.Y.; Menon, A.K.; Hong, L.; Chi, E.H.; Tjoa, S.; Kang, J.; Ettinger, E. Self-Supervised Learning for Deep Models in Recommendations. 2020. Available online: <https://openreview.net/forum?id=BCHN5z8nMRW> (accessed on 13 May 2022).
30. Zhou, K.; Wang, H.; Zhao, W.X.; Zhu, Y.; Wang, S.; Zhang, F.; Wang, Z.; Wen, J.R. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management, Galway, Ireland, 19–23 October 2020; pp. 1893–1902.
31. Long, X.; Huang, C.; Xu, Y.; Xu, H.; Dai, P.; Xia, L.; Bo, L. Social recommendation with self-supervised metagraph informax network. In Proceedings of the 30th ACM International Conference on Information & Knowledge Management, Gold Coast, QLD, Australia, 1–5 November 2021; pp. 1160–1169.
32. Liu, Z.; Chen, Y.; Li, J.; Yu, P.S.; McAuley, J.; Xiong, C. Contrastive self-supervised sequential recommendation with robust augmentation. *arXiv* **2021**, arXiv:2108.06479.
33. Wu, J.; Fan, W.; Chen, J.; Liu, S.; Li, Q.; Tang, K. Disentangled contrastive learning for social recommendation. In Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, 17–21 October 2022; pp. 4570–4574.
34. Yu, J.; Yin, H.; Gao, M.; Xia, X.; Zhang, X.; Viet Hung, N.Q. Socially-aware self-supervised tri-training for recommendation. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, Singapore, 14–18 August 2021; pp. 2084–2092.
35. Rendle, S.; Freudenthaler, C.; Gantner, Z.; Schmidt-Thieme, L. BPR: Bayesian personalized ranking from implicit feedback. In Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, 18–21 June 2009; pp. 452–461.
36. Zhao, T.; McAuley, J.; King, I. Leveraging social connections to improve personalized ranking for collaborative filtering. In Proceedings of the 23rd ACM International Conference on Information and Knowledge Management, Shanghai, China, 3–7 November 2014; pp. 261–270.
37. Wu, L.; Sun, P.; Fu, Y.; Hong, R.; Wang, X.; Wang, M. A neural influence diffusion model for social recommendation. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Paris, France, 21–25 July 2019; pp. 235–244.