

Article

A Graph Convolution Collaborative Filtering Integrating Social Relations Recommendation Method

Min Ma, Qiong Cao *  and Xiaoyang Liu 

School of Computer Science and Engineering, Chongqing University of Technology, Chongqing 400054, China
* Correspondence: caoqiong@cqut.edu.cn

Abstract: Traditional collaborative filtering recommendation algorithms only consider the interaction between users and items leading to low recommendation accuracy. Aiming to solve this problem, a graph convolution collaborative filtering recommendation method integrating social relations is proposed. Firstly, a social recommendation model based on graph convolution representation learning and general collaborative filtering (SRGCF) is constructed; then, based on this model, a social relationship recommendation algorithm (SRRA) is proposed; secondly, the algorithm learns the representations of users and items by linear propagation on the user–item bipartite graph; then the user representations are updated by learning the representations with social information through the neighbor aggregation operation in the social network to form the final user representations. Finally, the prediction scores are calculated, and the recommendation list is generated. The comparative experimental results on four real-world datasets show that: the proposed SRRA algorithm performs the best over existing baselines on Recall@10 and NDCG@10; specifically, SRRA improved by an average of 4.40% and 9.62% compared to DICER and GraphRec, respectively, which validates that the proposed SRGCF model and SRRA algorithm are reasonable and effective.

Keywords: social relations; collaborative filtering; graph convolutional network; recommendation system



Citation: Ma, M.; Cao, Q.; Liu, X. A Graph Convolution Collaborative Filtering Integrating Social Relations Recommendation Method. *Appl. Sci.* **2022**, *12*, 11653. <https://doi.org/10.3390/app122211653>

Academic Editor: Gianluca Lax

Received: 30 October 2022

Accepted: 15 November 2022

Published: 16 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/>).

1. Introduction

1.1. Background

During the age of information explosion, recommender systems have become widely used and effective method to identify the most valuable one in a massive amount of data. A recommender system (in short, RS) aims at estimating the likelihood of interactions between target users and candidates based on interactive history [1,2]. RSs first learn the users' and items' representations (also called embeddings), and then use these representations to predict how a target user will like a specific item.

The first successful algorithm to generate recommendations is Collaborative Filtering (CF) which only on user provided ratings. Traditional CF methods suffer from data sparsity. Matrix Factorization (in short MF) techniques are a viable method to alleviate data-sparsity: specifically, MF methods decompose the (high-dimensional) user–rating matrix into the product of two low-dimensional user–factor and item–factor matrices such that the inner product of the vectors associated with a user and items explains observed ratings. MF methods are also effective to cope with the cold-start problem (i.e., how to generate predictions for new members of a recommender system for which historical data are poor).

Graph Convolution Networks (GCN) have been successfully applied to improve the accuracy of an RS. However, recent studies [3] prove that two common operations in the design of GCNs (namely the task of transforming features and nonlinear activation) provide a little contribution to the performance of an RS. He et al. proposed a new GCN model for supporting CF tasks called LightGCN; the LightGCN architecture includes only the neighborhood aggregation step and, thus, it learns user and item embeddings by linear

propagation on the user–item graph. It finally computes the weighted sum of embeddings at all layers to produce final embeddings.

Although LightGCN achieves high levels of accuracy, it only takes data about historical interactions between users and items; many RS can manage a wealth of data describing user-to-user relationships (such as friendship or trust relationships) which are, de facto, ignored by LightGCN. As a consequence, the semantics of information mined by LightGCN is relatively simple and we ask if social data can be profitably integrated in the recommendation task to increase the level of accuracy and, potentially, to make the recommendation process more scalable. Motivated by theories from social science about influence and homophily, many researchers suggested to integrate user–item ratings with information describing social relationships (such as friendship or trust relationships); in fact, users' preferences as well as the decisions they take are often influenced by the behaviors from their peers and, thus, the predictive accuracy of an RS is magnified if social relationships are somewhat taken into account.

An RS which considers social information is often called social recommender system. Graph theory provide powerful tools to represent user interactions among users as well as interactions between users and items. On one hand, in fact, we could define a user–user interaction network (often called social graph), which maps user onto nodes and in which edges can represent friendship or trust relationships. On the other hand, we could use a bipartite graph (often called user–item interaction graph) to model users and items as nodes while edges are relevant to encode a variety of interactions such as purchases, clicks, and comments. Several social recommender systems which jointly leverage social graph and interactive graph have been proposed so far. The first is to extend matrix factorization methods into the interactive graph and the social graph, and the second is to apply Graph Neural Networks (in short, GNNs) to obtain meaningful representations.

1.2. Motivations

Despite it is useful to integrate social data into RS, we believe that there is still room to enhance the accuracy of an RS. We introduce a new GCN architecture which integrates social data in the recommendation process. Our approach aims at solving the following problems:

- (1) Heterogeneous data are difficult to use: the data used in social recommendation often contain both user interaction data and user social data. Heterogeneity of data implies that we are in charge of handling representations of different objects (items and users); as a consequence, we deal with nodes which are not in the same embedded space, and, thus, these nodes are hard to be fused.
- (2) High-order semantic information is hard to extract: For instance, high-order semantics describe relationships that users are indirectly connected to in the user–user social network. It is thus crucial to capture complex long-term dependencies between nodes. The more iteration layers of graph convolution architecture, the higher order semantic information will be extracted. However, excessive iteration layers will cause excessive smoothness.
- (3) Difficulties in fusing multiple semantic information: Social recommender systems manage both social network and interactive graph and it also has the task that effectively integrate the information coming from both of these graphs is still open research.

1.3. Our Contributions

The following contributions have made:

- (1) We innovatively integrate social relations into the training of graph convolution-based collaborative filtering recommendation method. Specifically, we propose a graph convolution collaborative filtering recommendation model integrating social relations (called SRGCF). The SRGCF model learns node embeddings by integrating high-order semantic information about social behaviors as well as interactions.
- (2) We propose a recommender algorithm (called SRRA) running on top of the SRGCF model. The SRRA algorithm models the high-order relations in interactive data and

social data, respectively, fuses these two types of high-order semantic information at the same layer and then forms the final embeddings. Generated embeddings are finally used in recommendation task.

- (3) We experimentally compared the SRRRA algorithm with baselines on a range of real-life and large datasets. Our experiments indicate the superiority of our model against baselines.

2. Related Work

2.1. Traditional CF Recommendation Algorithm

In e-commerce industry, collaborative filtering (in short, CF) has been widely used, and in the past two decades many CF algorithms have emerged in academia and industry. Roughly speaking, CF algorithms is classified into two categories: neighborhood-based CF algorithms [4] and model-based recommendation algorithms.

Neighborhood-based CF algorithms [5–8] can find potentially relevant items from user past behaviors without any domain knowledge, and they can be further classified into two specific types which are user-based CF and item-based CF. The key of neighborhood-based CF methods is how to calculate similarity and sum up these scores.

The primary idea behind the model-based CF algorithms [9] is to embed users as well as items into the same embedding space, and then make prediction through the inner-product of their embeddings. Using data mining and machine learning techniques, model-based approaches predict unknown scores by finding patterns in training data.

However, the accuracy of traditional CF recommendation method is limited because they make only use of interactions between users and items to predict unknown ratings.

2.2. Social Recommendation Algorithm

Users in rating platforms are often allowed to create explicit relationships between other users affiliated with the same platform. Examples of these relationships are friendship and trust. Some researchers [10,11] suggest to incorporate social relations in the recommendation process to better deal with data sparsity in the rating matrix. The resulting recommendation algorithm is often called social recommender system while the user–user social network is often called social graph.

Most traditional social recommender systems leverage CF technique. In Figure 1 we report the general structure of a social recommender systems.

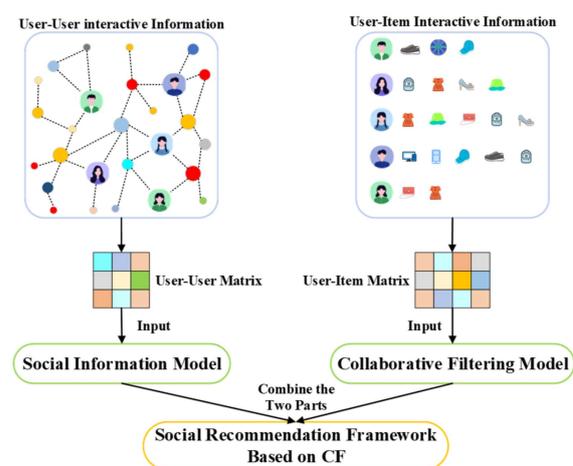


Figure 1. Social recommendation framework based on CF.

Figure 1 shows that a social recommender has two inputs, namely interactive information and social information.

According to different fusion mechanisms of these two types of data, social recommender systems can be classified into two categories: regularization-based and feature-

sharing based methods. The regularization-based social recommendation algorithm has the hypothesis that users trust friends in their social circle more than strangers and tend to conform to their preferences. The regularization-based recommendation algorithm projects social data and ratings into the same space and restricts each other so that users can consider their social influence before making decisions. Two important examples of regularization-based algorithms are SocialMF [12] and CUNE [13]. Regularized social recommendation algorithms indirectly simulate social network, while, due to the indirect modeling of social information, there is a low degree of overlap and correlation between user–item interaction information and social information, which leads to a weak integration of social information and ratings.

The basic assumption of feature sharing recommendation algorithms is that user feature vectors in interactive space and social space can be projected into the same space. TrustSVD [14] and SoRec [15] are two examples of feature-sharing social recommendation algorithms. Feature sharing based recommendation algorithm can generate accurate prediction. However, the current mainstream algorithms only use original social information, it means they cannot make full use of social data.

2.3. Graph Embedding Based Recommendation

Network embedding, also referred to as graph embedding, is a process of mapping graph data into a dense vector that is usually low-dimensional, so that the obtained vectors can have representation and reasoning ability in vector space [16,17]. Network embedding can be used as the input of machine learning model and then be applied to the recommendation task.

Graph Embedding can retain the structure information of nodes in the graph, that is, the more similar the structure is in the graph, the closer its position in the vector space will be [18,19]. The principle of graph embedding is shown in Figure 2.

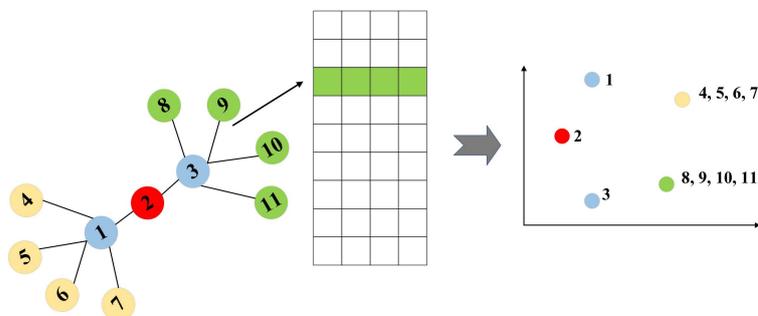


Figure 2. Illustration of graph embedding principle.

Figure 2 shows that node 1 and node 3 are similar in structure, so they maintain a symmetric position in vector space; nodes 4, 5, 6, and 7 are structurally equivalent, so they have the same position in vector space. Graph embedding based recommendation algorithms has two categories: homogenous graph embedding based and heterogeneous graph embedding based. A homogenous graph contains nodes and edges of only one type, and it only needs to aggregate neighbors of a single type to update the node representation. These algorithms are mostly based on random walk, such as Deepwalk [20], which uses truncation random walk sequence to represent the node nearest neighbor, and Node2vec [21], an improved version of Deepwalk. These algorithms only work on homogenous networks. Unfortunately, most real-world datasets can be modeled as heterogeneous graphs naturally. Thus, recommendation algorithms based on heterogeneous networks attract more attention.

In recent years, many experts and scholars [22,23] have studied the transformation of recommendation tasks into heterogeneous graph data mining tasks because real-world datasets can often be abstracted into heterogeneous graphs. Heterogeneous information

networks (HIN) include various types of nodes and edges. Figure 3 is an example diagram based on HIN recommendation system.

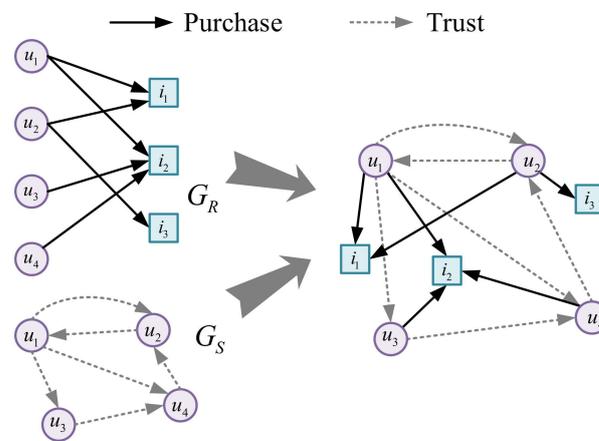


Figure 3. Illustration of a HIN recommendation system.

Figure 3 shows that a HIN contains multiple (two or more) types of entities linked by multiple (two or more) relationships.

In what follows we describe some HIN approaches which jointly consider social data and user interactions with item to generate recommendations.

In the research of Fan et al. [24], interactions and opinions are captured jointly in the interactive graph with GraphRec, which models two graphs (interactive graph and social graph) and heterogeneous edges. Wu et al. [25] proposed a DiffNet for the analysis of how the user in the social diffusion process is affected. DiffNet only modifies users' latent vectors while it does not update items' latent vectors which are independent of social influence. Yu et al. [26] present a social recommender system called MHCN, which uses hypergraphs to capture high-order social information. Each motif is encoded by a dedicated channel of a hypergraph convolutional network, and user embedding is calculated by aggregating the embeddings learned by each channel. Huang et al. [27] propose KCGN. KCGN models interdependencies between items as a triplet and it uses a coupled graph neural architecture to learn embeddings. In addition, it can automatically learn the temporal evolution of the interactive graph. Most traditional social recommenders learn embeddings in Euclidian space. Such a choice, however, is not entirely satisfactory to capture latent structural properties in graphs. In fact, both the interactive graph and the social graph display a tree-like structure that is hard to embed into a Euclidian space. To this purpose, Wang et al. [28] applied hyperbolic embeddings to represent users and items and they introduce a system, called HyperSoRec. Zhao et al. [29] describe a framework called BFHAN, which is able to improve node representations in graphs with a power-law degree distribution, and to handle various relationships of nodes associated with users. Fu et al. [30] introduce the DICER system. DICER first constructs an item-item and a user-user similarity in a weighted undirected graph way. A relation-aware graph neural network (RGNN) module is applied on the item-item graph (as well as user social network and user-user similarity graph) to obtain better users and items representations. Zhang et al. [31] describe a social recommender system called MG-HIF which constructs meta-paths and applies discrete cross-correlation to learn representations of user-item pairs; MG-HIF applies generative adversarial networks (GANs) on the social graph to learn latent friendship relationships. In addition, it uses two attentions models to fuse information from both graphs.

3. Preliminaries

3.1. Social High-Order Connectivity

Social relationships have high-order connectivity, as shown in Figure 4c.

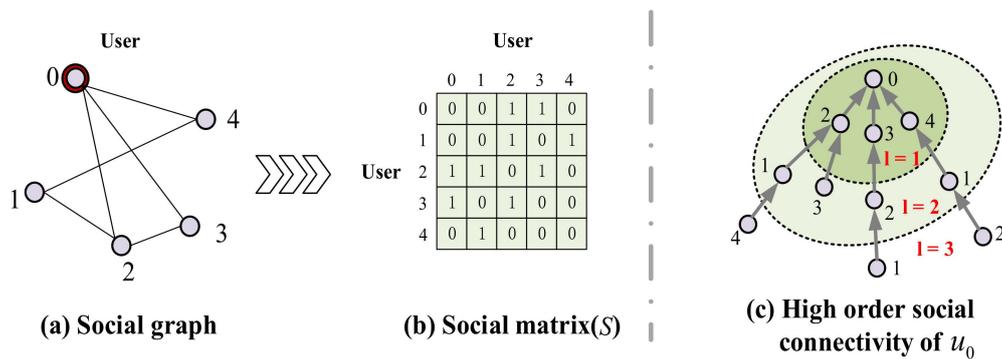


Figure 4. Social relationships.

The target node u_0 is marked with a double circle in the social graph.

Let us consider the path $u_0 \leftarrow u_2 \leftarrow u_1$, u_0 and u_1 are not directly connected, indicating that u_1 may be a potential friend of u_0 . In all the pathways that can reach u_0 , the closer a node is to u_0 , the more paths it occupies, and the greater the influence on u_0 .

3.2. Interactive High-Order Connectivity

Interaction relationships also have high-order connectivity, as shown in Figure 5c. Let us concentrate on a target user, say u_0 , marked with a double circle in the left subgraph of the user–projected interaction diagram.

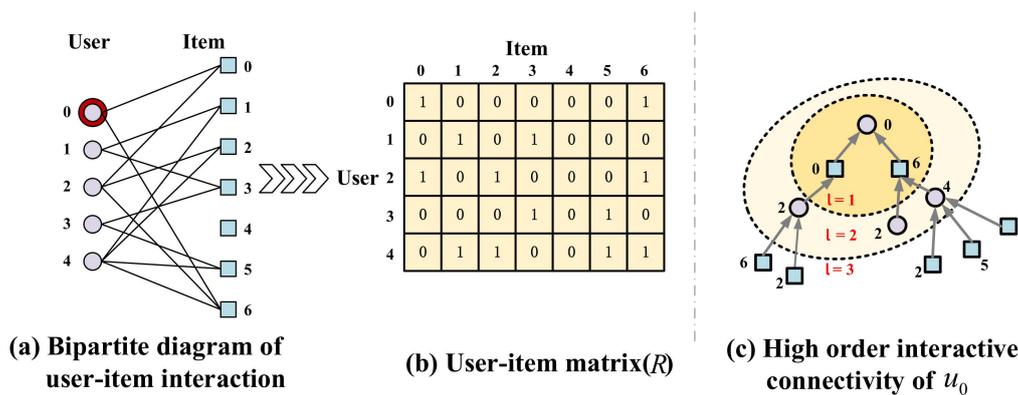


Figure 5. User–item interaction.

The subgraph on the right shows the tree structure expanded obtained by running a BFS search from u_0 . High-order connectivity indicates the existence of a path to u_0 of length l greater than 1. This high-order connectivity contains rich semantic information with collaborative signals. For example, path $u_0 \leftarrow i_6 \leftarrow u_4$ represents the behavioral similarity between u_0 and u_4 because the longer path $u_0 \leftarrow i_6 \leftarrow u_4 \leftarrow i_2$ indicates that u_0 is likely to adopt i_2 because its similar user u_4 has previously interacted with i_2 . Moreover, from the path of $l = 3$, u_0 is more likely to be interested in i_2 than i_5 , because $\langle i_2, u_0 \rangle$ has two paths connected, while $\langle i_5, u_0 \rangle$ has only one.

4. Proposed Recommendation Method

4.1. Recommendation Model Design

For the purpose of extracting the higher-order relationships in interactive data and social data and fully integrate them to learn high-quality representations, we propose the SRGCF (Social Recommendation Graph Collaborative Filtering) model. Figure 6 shows the overall architecture of SRGCF.

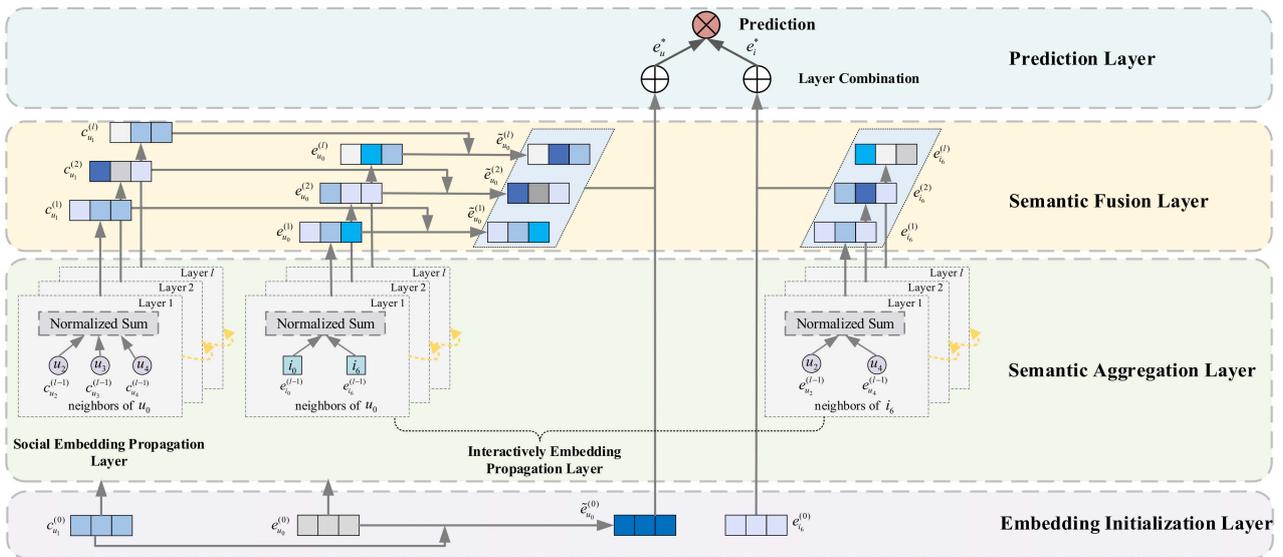


Figure 6. The overall frame structure of the proposed SRGCF model.

The SRGCF model first initializes node embeddings by using the initialization embedding layer. Second, semantic aggregation operations are carried out on the social embedding propagation layer and the interactive embedding propagation layer in the semantic aggregation layer to refine the embedding of users and items, derived from both graphs are fused in the semantic fusion layer. Then weighted sum the user and item embeddings of every layer, respectively, to form the final embeddings. Finally, the prediction layer is applied for producing recommendations.

4.1.1. Embedding Initialization Layer

Embedding matrix of the nodes are randomly initialized and the initial embeddings $e_u^{(0)} \in \mathbb{R}^d$ and $e_i^{(0)} \in \mathbb{R}^d$ of user u and item i and they can be queried, where d is the dimension of nodes' embeddings.

4.1.2. Semantic Aggregation Layer

We propose a semantic aggregation layer in order to aggregate and update the nodes' embeddings, as a result, it is a good way to keep high-level semantic information. We first introduce the concept of first-order semantic aggregation in semantic aggregation layer, and then extend it to high-level semantic aggregation to realize high-level semantic aggregation.

(1) First-order Semantic Aggregation

By iteratively aggregating the neighbor's features, GCN generates new representation of the target node. In SRGCF model, the interactive embedding propagation layer aggregates the embeddings of interacted items to refine the embedding of users. First-order semantic aggregation is reported in (1) and (2).

$$e_u = AGG_{i \in H_u}(e_i) \tag{1}$$

$$e_i = AGG_{u \in H_i}(e_u) \tag{2}$$

where $AGG(\cdot)$ denotes aggregation function; H_i represents the set of first-order neighbors of item i , that is, the set of users that have interacted with item i . Similarly, H_u represents the set of first-order neighbors of item u . Equations (1) and (2) indicate that in the interaction, e_u derived from an embedded set of its immediate neighbors, as is e_i .

The social embedding propagation layer refines user embedding by aggregating information from users' friends. The first-order semantic aggregation process is shown in (3).

$$e_u = AGG_{v \in F_u}(e_v) \tag{3}$$

where F_u represents the friends' collection of user u . It indicates that in social interaction, the embedded e_u of user u is generated through the aggregation of embedded e_v of first-order neighbor (social interaction).

(2) High-order Semantic Aggregation

The semantic aggregation layer achieves the aggregation of higher-order semantics by stacking multiple first-order semantic aggregation layers. It includes semantic aggregation for social embedding propagation layer (SEPL) and interactive embeddings propagation layer (IEPL).

(a) Semantic Aggregation for SEPL

According to social high-order connectivity, stacking l layers can fuse the information from each of l -order neighbors. Semantic aggregation for social embedding propagation layer captures higher-order friend signals by stacking multiple social embedding propagation layers to enhance user embeddings. The mathematical expression of this process is shown in (4) and (5).

$$c_u^{(l+1)} = \sum_{v \in F_u} \frac{1}{\sqrt{|F_u|}\sqrt{|F_v|}} c_v^{(l)} \tag{4}$$

$$c_v^{(l+1)} = \sum_{u \in F_v} \frac{1}{\sqrt{|F_v|}\sqrt{|F_u|}} c_u^{(l)} \tag{5}$$

where $c_u^{(l)}$ denotes the embedding of u at the l -th layer from G_S , and F_u denotes the set of friends of u .

(b) Semantic Aggregation for IEPL

It can be seen from interaction high-order connectivity that stacking even layers (i.e., from the user, the length of path is even) can capture the similarity information of user behavior, while stacking odd layers can capture the potential interaction information of users to items. Semantic aggregation for interaction embedding propagation layer captures collaborative signals of high-order connectivity in interaction data by stacking each interaction embedding propagation layer, thus enhancing embeddings. Expression of this process is shown in (6) and (7).

$$e_i^{(l+1)} = \sum_{u \in N_i} \frac{1}{\sqrt{|H_i|}\sqrt{|H_u|}} e_u^{(l)} \tag{6}$$

$$e_u^{(l+1)} = \sum_{i \in N_u} \frac{1}{\sqrt{|H_u|}\sqrt{|H_i|}} e_i^{(l)} \tag{7}$$

where $e_u^{(l)}$ and $e_i^{(l)}$ represent u 's and i 's embedding at l -th layer from G_R , respectively.

4.1.3. Semantic Fusion Layer

User embeddings can be enhanced by integrating social embedding propagation layer and interactive embedding propagation layer with certain social information.

After obtaining the social semantic aggregation embedding and interactive semantic aggregation embedding, respectively, the user embeddings of each layer are fused, and the fusion process is shown in (8).

$$\hat{e}_u^{(l)} = g(e_u^{(l)}, c_u^{(l)}) \tag{8}$$

where $\tilde{e}_u^{(l)}$ denotes the l -th layer embedding of u from G_R and G_S , and we let $e_u^{(0)} = c_u^{(0)}$. $g(\cdot)$ is a fusion function, which can be implemented in many ways, and we adopt (9) to fuse:

$$\tilde{e}_u^{(l)} = \text{norm}(\text{sum}(e_u^{(l)}, c_u^{(l)})) \tag{9}$$

where $\text{sum}(\cdot)$ denotes element-wise summation. Intuitively, it is an operation to enhance the signal representation, and also can keep the feature space unchanged; $\text{norm}(\cdot)$ denotes row-regularization operation that normalizes user vectors.

Then, the final user embedding e_u^* and item embedding e_i^* is obtained by fusing embeddings of all layers:

$$e_u^* = \sum_{l=0}^L \alpha_l \tilde{e}_u^{(l)}; e_i^* = \sum_{l=0}^L \beta_l e_i^{(l)} \tag{10}$$

where e_u^* denotes u 's final embedding, e_i^* denotes the i 's final embedding, and L denotes total number of layers. In line with LightGCN, set α and β as $1/(L + 1)$. The settings of these two parameters are flexible, and attention mechanism can be applied to learn them.

4.1.4. Prediction Layer

The last part of the model recommends products to users according to the embedding of items. We use the inner-product form for prediction:

$$\hat{y}_{ui} = e_u^{*T} e_i^* \tag{11}$$

Then BPR loss [32] was calculated and model parameters were optimized as shown in Equation (12).

$$J = \sum_{(u,i,j) \in O} -\ln \sigma(\hat{y}_{ui} - \hat{y}_{uj}) + \lambda \|\Theta\|_2^2 \tag{12}$$

where $O = \{(u, i, j) | (u, i) \in R^+, (u, j) \in R^-\}$ represents pair-wised training data, R^+ expresses interactions that exist in history, and R^- denotes interaction that does not. Θ are model parameters, where the model parameters only include the initial embedding vectors $e_u^{(0)}$ and $e_i^{(0)}$. λ is used to prevent overfitting.

4.2. The Proposed SRRA Recommendation Algorithm

In order to facilitate implementation, SRRA algorithm is proposed under the framework of SRGCF model, which is implemented in the form of matrix (see Algorithm 1 for details).

The interactive matrix is denoted as $R \in \mathbb{R}^{M \times N}$, M are the numbers of user and N are the numbers of item, R_{ui} equals to 1 if u have interaction with i , if not R_{ui} equals to 0. Then the adjacency matrix A of G_R is:

$$A = \begin{pmatrix} 0 & R \\ R^T & 0 \end{pmatrix} \tag{13}$$

Let the embedding matrix of layer 0 be $E^{(0)} \in \mathbb{R}^{(M+N) \times d}$, where d is the dimension of embedding vector, and the $(l + 1)$ -th layer matrix can be computed as:

$$E^{(l+1)} = (D^{-\frac{1}{2}} A D^{-\frac{1}{2}}) E^{(l)} \tag{14}$$

where D is the degree matrix of A , which is a diagonal matrix and its dimension is $(M + N) \times (M + N)$. Each element D_{ii} represents the number of non-zero values of the i -th row vector in A .

Algorithm 1: Social Relationship Recommendation Algorithm (SRRA).**Step1: calculate the embeddings of users and items****Input:** R, S, M, N, d, l **Initialize:** $E^{(0)} = C^{(0)}, K, \alpha = \beta = 1/(l+1)$ calculate A and B by R, S , respectivelycalculate D and P by A, B , respectively $E_U^{(0)}, E_I^{(0)} \leftarrow E^{(0)}, \dots, C_U^{(0)} \leftarrow C^{(0)}$ $\tilde{E}_U \leftarrow E_U^{(0)}, C_U^{(0)},$ set $E_I^* = E_I^{(0)}$ **For** $l \in L$:calculate $E^{(l+1)}$ and $C^{(l+1)}$ by $E^{(l)}, C^{(l)}$, respectivelyget $E_U^{(l+1)}, E_I^{(l+1)}, C_U^{(l+1)}$ by split $E^{(l+1)}, C^{(l+1)}$, respectively $\tilde{E}_U^{(l+1)} \leftarrow E_U^{(l+1)}, C_U^{(l+1)}$ $E_U^{*+} = \tilde{E}_U^{(l+1)}, E_U^{*+} = E_I^{(l+1)}$ **End For** $E_U^* = \alpha E_U^*, E_I^* = \beta E_U^*$ **Step2: calculate the loss of SRRA**set $L_{BPR} = 0$ **For** $u \in U$: $e_u^{(0)} = \text{lookup}(E_U^{(0)}, u)$ // find the initial vector of user u from $E_U^{(0)}$ $L_{BPR} + = \|e_u^{(0)}\|^2$ // add the regularization item into loss**For** $i \in R_u^+$: // iterate over the positive example item set for user u $e_i^* = \text{lookup}(E_I^*, i)$ // find the vector of item i from E_I^* $\hat{y}_{ui} = e_u^{*T} e_i^*$ // calculate the score of positive samples**For** $j \in R_u^-$: // iterate over the negative example item set for user u $e_j^* = \text{lookup}(E_I^*, j)$ $\hat{y}_{uj} = e_u^{*T} e_j^*$ $L_{BPR} + = (-\ln \sigma(\hat{y}_{ui} - \hat{y}_{uj}))$ // calculate the BPR loss**End For****End For****Step3: generate recommendations**

Train the algorithm until it converges

According to the predicted score, select Top 10 items for recommendation

Return Recall, NDCG

Similarly, the social matrix is denoted as $S \in \mathbb{R}^{M \times M}$, where S_{uv} is 0 if u and v are friends, otherwise S_{uv} is 1. The adjacency matrix B of G_S is:

$$B = \begin{pmatrix} 0 & S \\ S^T & 0 \end{pmatrix} \quad (15)$$

Let the embedding matrix of layer 0 be $C^{(0)} \in \mathbb{R}^{(M+M) \times d}$, and the user matrix of layer $l+1$ can be obtained as shown in (16).

$$C^{(l+1)} = (P^{-\frac{1}{2}} B P^{-\frac{1}{2}}) C^{(l)} \quad (16)$$

where P is the degree matrix of matrix B .

Then, due to $E^{(l)} = \text{stack}(E_U^{(l)}, E_I^{(l)})$, $E^{(l)}$ can be divided into user's and item's matrices, denoted as $E_U^{(l)}$ and $E_I^{(l)}$, respectively. Similarly, due to $C^{(l)} = \text{stack}(C_U^{(l)}, C_U^{(l)})$, $C^{(l)}$ can be divided into two parts, both of which are user embedding matrix, where $C_U^{(l)}, E_U^{(l)} \in \mathbb{R}^{M \times d}$ and $E_I^{(k)} \in \mathbb{R}^{N \times d}$.

Finally, the l -th layer user representation is calculated as:

$$\tilde{E}_U^{(l)} = \text{norm}(\text{sum}(E_U^{(l)}, C_U^{(l)})) \quad (17)$$

The final representations can be obtained through integrating the representation of each layer:

$$E_U^* = \sum_{l=0}^L \alpha_l \tilde{E}_U^{(l)}, E_I^* = \sum_{l=0}^L \beta_l E_I^{(l)} \quad (18)$$

We use inner-product to compute score:

$$\hat{Y} = E_U^{*T} E_I^* \quad (19)$$

4.3. Model Training

The loss function calculated by BPR is shown in (20).

$$L_{BPR} = - \sum_{u=1}^M \sum_{i \in H_u} \sum_{j \notin H_u} \ln \sigma(\hat{y}_{ui} - \hat{y}_{uj}) + \lambda \|E^{(0)}\|^2 \quad (20)$$

Adam [33] is used as the optimizer of loss function. Its primary characteristic is that it can self-adapt learning rates.

5. Experiment

5.1. Experiment Setup

5.1.1. Datasets

We used four datasets in this paper. The following is an introduction to these datasets, and their statistical details are summarized in Table 1.

- Brightkite. A position sharing platform with social networking platform where users share their locations through check-ins. It includes check-in data as well as social data.
- Gowalla. A position sharing platform similar to Brightkite. This dataset includes check-in data and user social data.
- Epinions. A consumer review website which allows users to clicked items and add trust users. This dataset contains users' rating data and trust network data.
- LastFM. A social music platform for music sharing. This dataset includes data about users' listening to music and users' relationships.

Table 1. Statistical details of four datasets.

Dataset	Brightkite	Gowalla	Epinions	LastFM
#User	6310	14,923	12,392	1860
#Item	317,448	756,595	112,267	17,583
#Interaction	1,392,069	2,825,857	742,682	92,601
#Connection	27,754	82,112	198,264	24,800
R-Density	6.9495×10^{-4}	2.5028×10^{-4}	5.3384×10^{-4}	2.8315×10^{-4}
S-Density	6.9705×10^{-4}	3.6872×10^{-4}	1.2911×10^{-3}	7.1685×10^{-3}

5.1.2. Baselines

For the purpose of illustrating how effective our model is, we compared SRRA with three types of approaches: one social recommendation model based on deep learning (DL), three social recommendation models based on DL and GNN, and one recommendation model based on GCN, which is shown in Table 2.

- LightGCN [3]: It is effective to extract the collaborative signal explicitly in the embedding process by modeling high-order connectivity in interactive graphs.
- DSCF [34]: It utilizes information provided by distant neighbors and explicitly captures the neighbor's different opinions towards items.
- DiffNet [25]: It is a GNN model which analyzes how users make their decisions based on recursive social diffusion.

- GraphRec [24]: It captures interactions and opinions in G_R and it also models two graphs (e.g., G_R and G_S) and the strength of heterogeneity in a coherent way.
- DICER [30]: It models user and item by introducing high-order neighbor information, and draws the most relevant interactive information based on deep context.

Table 2. Comparison of the model's characteristics.

Methods	Social Recommendation	DL	Graph-Based	
			GNN	GCN
LightGCN				✓
DSCF	✓	✓		
DiffNet	✓	✓	✓	
GraphRec	✓	✓	✓	
DICER	✓	✓	✓	
SRRA	✓			✓

5.1.3. Evaluation Metrics

We utilize two widely adopted metrics Recall@K and NDCG@K in comparisons, since we try to recommend the Top-K list items for each user. Specifically, Recall measures the percentage of the test data that users actually like from the Top-K list. In addition, NDCG is a position-aware ranking metric that measures how the hit items are placed and gives a higher score if they are at the top of the list.

5.1.4. Experiments Details

We use 80% of Brightkite, Gowalla, Epinions and LastFM for training, 10% for tuning hyper-parameters, and 10% for testing final performance. Parameters for all methods are randomly initialized with standard normal distribution. In addition, initialization and tuning of parameters for the baseline algorithms followed the procedures described in the corresponding papers. With batch size 1024, we tested each value in {8, 16, 32, 64, 128, 256} for embedding size d , and we also find the proper value for learning rate and L2 regularization factor in {0.0005, 0.001, 0.005, 0.01, 0.05, 0.1} and $\{1 \times 10^{-6}, 1 \times 10^{-5}, \dots, 1 \times 10^{-2}\}$, respectively. The aggregation factors α_l and β_l of each layer were set as $1/(L + 1)$, where L represents the total number of layers.

5.2. Overall Comparison

We compare all methods in this subsection. In Table 3, we show performance comparison between SRRA and baselines. The following conclusions can be drawn:

First, methods that incorporate social relations outperform that does not. In Table 3, for example, DSCF, DiffNet, GraphRec, DICER, and SRRA outperform LightGCN. This demonstrates that social information is effective and helpful by being incorporated into recommender systems.

Second, our method SRRA achieves the best performance on these four datasets. Specifically, in comparison to DICER, a GNN and DL-based social recommendation model, SRRA scores better by an average improvement of 2.27%, 2.85%, 5.58%, and 6.90%; and to GraphRec, a very expressive GNN-based social recommendation model, SRRA scores better by an average improvement of 5.55%, 6.44%, 13.42% and 13.08% on the four datasets, respectively. We guess a possible reason is that, for Brightkite and Gowalla, as they are social networks related to location, the activities and consumption preferences for users in this type of social platform is not easy to be affected; and for Epinions and LastFM, people strongly rely on social relations to acquire correct review of goods and lists of music they will listen to. It is possible to attribute our improved model over the baseline to two factors: (1) our model use GCN architecture to extract the social ties and interactive ties in a high-order way, which can leverage the relative information from multi-hop neighbors and high-order collaborative information propagated over the user interaction graph;

(2) we fuse all-order social and collaborative information when modeling user and item representation which generates improved user and item representation.

Table 3. Performance comparison between SRRA and baselines.

Recall@10						
Methods	LightGCN	DSCF	DiffNet	GraphRec	DICER	SRRA
Brightkite	0.1642	0.1895	0.1962	0.2172	<u>0.2235</u>	0.2293
Gowalla	0.2083	0.2253	0.2399	0.2779	<u>0.2886</u>	0.3011
Epinions	0.2269	0.2613	0.2874	0.2845	<u>0.3155</u>	0.3341
LastFM	0.2519	0.2742	0.2932	0.2876	<u>0.3059</u>	0.3272

NDCG@10						
Methods	LightGCN	DSCF	DiffNet	GraphRec	DICER	SRRA
Brightkite	0.1321	0.1393	0.1539	0.1612	<u>0.1672</u>	0.1701
Gowalla	0.1355	0.1482	0.1667	0.1724	<u>0.1744</u>	0.1782
Epinions	0.1425	0.1598	0.1642	0.1709	<u>0.1737</u>	0.1824
LastFM	0.1431	0.1563	0.1628	0.1862	<u>0.1953</u>	0.2086

5.3. Parameter Analysis

For the proposed model, there are two crucial parameters: the number of layers l and the embedding size d . In this section, we first change only one parameter and fix the others, and then observe how the performance changes.

Effect of the number of layers l . Take Gowalla and Epinions for example, we set the l from 1 to 5 to measure the impact of different layers, and then we can obtain the performance with the different number of layers that showed in Figure 7. We observed that performance increases and then decreases as the number of layers grows. When the number of layers grows from 1 to 4, performance of SRRA is improved. However, performance starts to become worse when the number of layers is 5. It demonstrates that too many layers may cause over smoothness that is a common problem existing in graph convolution methods. Thus, in order to prevent this, we need to use the proper number of layers.

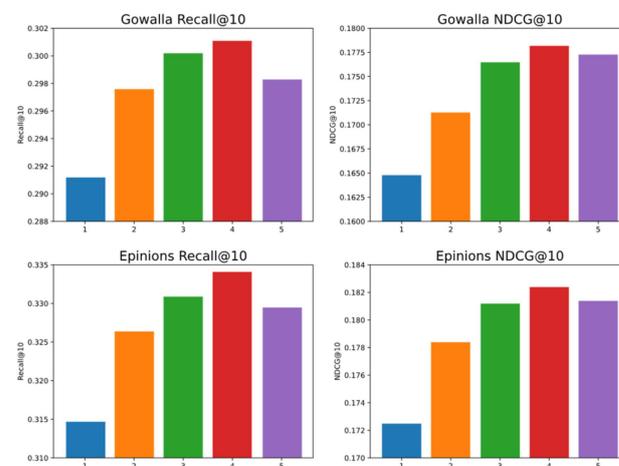


Figure 7. Effect of #layers l on Gowalla and Epinions.

Effect of embedding size d . In this subsection, take Gowalla and Epinions for example, we analyze how the embedding size of e_u and e_i affect proposed model. On these

two datasets, Figure 8 compares the performance of our proposed model when varying its embedding size d .

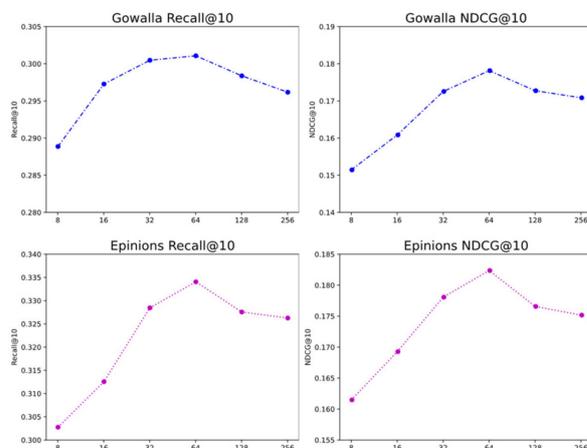


Figure 8. Effect of embedding size d on Gowalla and Epinions.

Accordingly, as embedding size increases, performance first become better, then worse. If the size grows from 8 to 64, the performance improves obviously. However, the performance of SRRA starts to deteriorate when the embedding size is 128. It demonstrates that a large embedding size is likely to produce powerful representations. Nevertheless, if the length of embeddings is too long, our model will become more complex. Thus, we must find an appropriate embedding size to make a trade-off, and we find that 64 is the optimal value.

6. Conclusions

In this work, we proposed a new social recommendation method which leverages graph convolution technique and integrates social relations. Firstly, we construct the architecture of a general collaborative filtering social recommendation model based on graph convolution (SRGCF), which consists of four parts, which are initialization embedding layer, semantic aggregation layer, semantic fusion layer and prediction layer, respectively. The semantic aggregation layer and semantic fusion layer are the core of SRGCF, which play the role of extracting high-order semantic information and integrating various semantic information, respectively. Then, we propose a feasible SRRA algorithm on top of the architecture, which can model interactions as well as social relations. It can use richer social information to mine the potential relationship, so as to improve the performance of recommendations. Comparative experiments on four datasets have proven the effectiveness of the proposed model.

Different from previous work, we try to explore how to use graph neural network method and introduce social auxiliary information to construct recommendation model in order to learn better representation. The graph-based model is superior to the traditional recommendation model because it can learn not only the representation of entities but also the relationships between them. However, limited by the shortcomings of graph neural network itself, such as excessive smoothing after several iterations, entity representation may not be fully learned, which requires some optimization in model design. In the future, we plan to optimize the model architecture by increasing the coupling between social modeling and interactive modeling, so that the representation learning is more adequate. We will also try to explore the advantages of other graphical representation learning techniques to improve the learning ability of the model.

Author Contributions: Conceptualization, M.M.; methodology, X.L.; software, Q.C.; formal analysis, Q.C.; writing—original draft preparation, Q.C.; writing—review and editing, Q.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Science and Technology Research Project of Chongqing Municipal Education Commission, grant number KJZDK202001101; Chongqing Postgraduate Research Innovation Project, grant number gzlcx20223205; General Project of Chongqing Natural Science Foundation, grant number cstc2021jcyj-msxmX0162; 2021 National Education Examination Research Project, grant number GJK2021028.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The calculated data presented in this work are available from the corresponding authors upon reasonable request.

Acknowledgments: The author would like to thank the anonymous reviewers for their valuable comments on our paper.

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

Nomenclature

U	the set of users
I	the set of items
G_R	user–item interaction graph
G_s	user–user social graph
d	the dimension of node embedding
H_i	neighbors of item i on G_R
H_u	neighbors of user u on G_s
l	#layer
L	total number of layers
F_u	the friends of user u
$e_u^{(l)}$	the embedding of user u at the l -th layer from G_R
$c_u^{(l)}$	the embedding of user u at the l -th layer from G_s
$\tilde{e}_u^{(l)}$	the l -th layer embedding of user u from G_R and G_s
e_u^*	final embedding of user u
e_i^*	final embedding of item i
M	the numbers of users
N	the numbers of items
R	user–item interaction matrix
S	social matrix
A	adjacency matrix of G_R
B	adjacency matrix of G_s
D	degree matrix of matrix A
P	degree matrix of matrix B
R^+	observable interactions
R^-	unobserved interactions
Θ	model parameters
$E^{(l)}$	the l -th layer matrix of GCN on G_R
$C^{(l)}$	the l -th layer matrix of GCN on G_s
E_U^*	final embedding matrix of users
E_I^*	final embedding matrix of items

References

1. Liu, X.; Li, X.; Fiumara, G.; De Meo, P. Link prediction approach combined graph neural network with capsule network. *Expert Syst. Appl.* **2023**, *212*, 118737. [[CrossRef](#)]
2. Liu, X.; Zhao, Z.; Zhang, Y.; Liu, C.; Yang, F. Social Network Rumor Detection Method Combining Dual-Attention Mechanism with Graph Convolutional Network. *IEEE Trans. Comput. Soc. Syst.* **2022**. [[CrossRef](#)]
3. 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 (SIGIR'20), Virtual Event, 25–30 July 2020; pp. 639–648.

4. Liu, T.; Deng, X.; He, Z.; Long, Y. TCD-CF: Triple cross-domain collaborative filtering recommendation. *Pattern Recognit. Lett.* **2021**, *149*, 185–192. [[CrossRef](#)]
5. Nassar, N.; Jafar, A.; Rahhal, Y. A novel deep multi-criteria collaborative filtering model for recommendation system. *Knowl. Based Syst.* **2020**, *187*, 32–39. [[CrossRef](#)]
6. He, X.; Liao, L.; Zhang, H.; Nie, L.; Hu, X.; Chua, T. Neural Collaborative Filtering. In Proceedings of the 26th International Conference on World Wide Web (WWW'17), Perth, Australia, 3–7 May 2017; pp. 173–182.
7. Wang, H.; Wang, N.; Yeung, D. Collaborative Deep Learning for Recommender Systems. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'15), Sydney, Australia, 10–13 August 2015; pp. 1235–1244.
8. Xue, F.; He, X.; Wang, X.; Xu, J.; Liu, K.; Hong, R. Deep Item-based Collaborative Filtering for Top-N Recommendation. *ACM Trans. Inf. Syst.* **2019**, *37*, 1–25. [[CrossRef](#)]
9. Khojamli, H.; Razmara, J. Survey of similarity functions on neighborhood-based collaborative filtering. *Expert Syst. Appl.* **2021**, *185*, 142–155. [[CrossRef](#)]
10. Sanz-Cruzado, J.; Castells, P.; Macdonald, C.; Ounis, I. Effective contact recommendation in social networks by adaptation of information retrieval models. *Inf. Process. Manag.* **2020**, *57*, 1633–1647. [[CrossRef](#)]
11. Sun, Z.; Guo, Q.; Yang, J.; Fang, H.; Guo, G.; Zhang, J.; Burke, R. Research commentary on recommendations with side information: A survey and research directions. *Electron. Commer. Res. Appl.* **2019**, *37*, 43–50. [[CrossRef](#)]
12. Jamali, M.; Ester, M. A Matrix Factorization Technique with Trust Propagation for Recommendation in Social Networks. *Assoc. Comput. Mach.* **2010**, *1*, 135–142.
13. Nitesh, C.; Wei, W. Collaborative User Network Embedding for Social Recommender Systems. In Proceedings of the 2017 SIAM International Conference on Data Mining, Sandy, UT, USA, 27–29 April 2017; pp. 381–389.
14. Guibing, G.; Jie, Z.; Yorke, S.N. TrustSVD: Collaborative Filtering with Both the Explicit and Implicit Influence of User Trust and of Item Ratings. In Proceedings of the 29th AAAI Conference on Artificial Intelligence, New York, NY, USA; 2015; pp. 123–129.
15. Ma, H.; Yang, H.; Lyu, M.R.; King, I. SoRec: Social Recommendation using Probabilistic Matrix Factorization. *Neurocomputing* **2019**, *341*, 931–940.
16. Cui, P.; Wang, X.; Pei, J.; Zhu, W. A Survey on Network Embedding. *IEEE Trans. Knowl. Data Eng.* **2019**, *31*, 833–852. [[CrossRef](#)]
17. Palash, G.; Emilio, F. Graph Embedding Techniques, Applications, and Performance: A Survey. *Knowledge-Based Syst.* **2017**, *151*, 78–84.
18. Wang, D.; Cui, P.; Zhu, W. Structural Deep Network Embedding. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 1225–1234.
19. Tang, J.; Qu, M.; Wang, M.; Zhang, M.; Yan, J.; Mei, Q. LINE: Large-scale Information Network Embedding. In Proceedings of the International Conference on World Wide Web, Florence, Italy, 18–22 May 2015; Volume 3, pp. 1067–1077.
20. Perozzi, B.; Al-Rfou, R.; Skiena, S. Deepwalk: Online learning of social representations. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24–27 August 2014; pp. 701–710.
21. Grover, A.; Leskovec, J. node2vec: Scalable Feature Learning for Networks. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'16), San Francisco, CA, USA, 13–17 August 2016; pp. 855–864.
22. De Paola, A.; Gaglio, S.; Giammanco, A.; Lo Re, G.; Morana, M. A multi-agent system for itinerary suggestion in smart environments. *CAAI Trans. Intell. Technol.* **2021**, *6*, 377–393. [[CrossRef](#)]
23. Weng, L.; Zhang, Q.; Lin, Z.; Wu, L. Harnessing heterogeneous social networks for better recommendations: A grey relational analysis approach. *Expert Syst. Appl.* **2021**, *174*, 142–154. [[CrossRef](#)]
24. Fan, W.; Ma, Y.; Li, Q.; He, Y.; Zhao, Y.E.; Tang, J.; Yin, D. Graph Neural Networks for Social Recommendation. In Proceedings of the World Wide Web Conference (WWW 2019), San Francisco, CA, USA, 13–17 May 2019; pp. 417–426.
25. 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, SIGIR 2019, Paris, France, 21–25 July 2019; pp. 235–244.
26. Yu, J.; Yin, H.; Li, J.; Wang, Q.; Hung, N.Q.V.; 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.
27. 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 35th AAAI Conference on Artificial Intelligence (AAAI), Virtual, 2–9 February 2021.
28. Wang, H.; Lian, D.; Tong, H.; Liu, Q.; Huang, Z.; Chen, E. HyperSoRec: Exploiting Hyperbolic User and Item Representations with Multiple Aspects for Social-aware Recommendation. *ACM Trans. Inf. Syst.* **2022**, *40*, 1–28. [[CrossRef](#)]
29. Zhao, M.; Deng, Q.; Wang, K.; Wu, R.; Tao, J.; Fan, C.; Chen, L.; Cui, P. Bilateral Filtering Graph Convolutional Network for Multi-relational Social Recommendation in the Power-law Networks. *ACM Trans. Inf. Syst.* **2022**, *40*, 1–24. [[CrossRef](#)]
30. Fu, B.; Zhang, W.; Hu, G.; Dai, X.; Huang, S.; Chen, J. Dual Side Deep Context-aware Modulation for Social Recommendation. In Proceedings of the Web Conference 2021, Ljubljana, Slovenia, 19–23 April 2021; pp. 2524–2534.
31. Zhang, C.; Wang, Y.; Zhu, L.; Song, J.; Yin, H. Multi-graph heterogeneous interaction fusion for social recommendation. *ACM Trans. Inf. Syst. TOIS* **2021**, *40*, 1–26. [[CrossRef](#)]

32. Rendle, S.; Freudenthaler, C.; Gantner, Z.; Schmidt-Thieme, L. BPR: Bayesian Personalized Ranking from Implicit Feedback. In Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, 18–21 June 2009; pp. 452–461.
33. Kingma, D.P.; Ba, J.L. Adam: A Method for Stochastic Optimization. In Proceedings of the International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015.
34. Fan, W.; Ma, Y.; Yin, D.; Wang, J.; Tang, J.; Li, Q. Deep Social Collaborative Filtering. In Proceedings of the 13th ACM Conference on Recommender Systems (RecSys'19), Copenhagen, Denmark, 16–20 September 2019; pp. 305–313.