Graph Neural Networks for Social Recommendation

1st chikate sathwika
Student of Engineering (of Aff.)
Sir Padampat Singhania University (of Aff.)
Udaipur,India
Chikate.sathwika@spsu.ac.in

2ndChinthakuntala Uday Charan
Student of Engineering (of Aff.)
Sir Padampat Singhania University (of Aff.)
Udaipur,India
chinthakuntala.udaycharan@spsu.ac.in

3rdSangepu Siddhartha
Student of Engineering (of Aff.)
Sir Padampat Singhania University (of Aff.)
Udaipur,India
Siddhartha.sangepu@spsu.ac.in

***Abstract*—Graphs may be used to represent data in a variety of real-world applications, including social networks, consumer buying habits, and inter-item interactions. Graph Neural Networks (GNNs) have demonstrated remarkable effectiveness in acquiring meaningful graph representations through their intrinsic integration of topological structure and node information. User-user social graphs and user-item graphs are two other ways that data from social recommendations may be represented as graph data. Furthermore, item-item graphs may be used to represent the relationships between the items. GNNs offer a previously unheard-of chance to improve social recommendations. Nevertheless, creating GNNbased social recommendations presents enormous challenges in the following scenarios: (1) users (items) participate in both the user-item graph and the user-user social graph (item-item graph) concurrently; (2) In addition to user-item interactions, user views on things are also included in user-item graphs; (3) the form of social ties varies across users. In this study, we offer a novel graph neural network framework (GraphRec+) for social recommendations that can develop more accurate user and object representations by coherently modeling graph data. In particular, we present a guiding methodology to concurrently record opinions and interactions in the user-item graph and also suggest an attention mechanism to distinguish between the many levels of social ties. Extensive tests on three realworld datasets demonstrate the usefulness of the suggested methodology. Index Terms—GNN, Social Recommenddation, graph data.**

I. KEYWORDS

KEYWORDS-Social Recommendation; Graph Neural Networks; Recommender Systems; Social Network; Neural Networks

II. INTRODUCTION

In recent years, there has been a growing interest in the use of social interactions for recommender systems [18, 28, 30]. The development of these social recommender systems was founded on the fact that people often learn about and share information through others in their immediate social circle, including friends, classmates, or coworkers, suggesting that users' underlying social relationships can be crucial in assisting them in filtering information . Thus, it has been demonstrated that social relationships improve recommendation performance. Deep neural network methods for graph data have made significant strides in recent years .

Graph Neural Networks (GNNs) have shown promise in addressing challenges in social recommendation systems. They enhance recommendation performance by leveraging social network information, user-item interactions, and user behavior data. GNN-based models like STL improve recommendation accuracy by modifying graph structures, expanding positive samples, and mining hard negative samples. GNNs for preference social recommendation effectively capture social preferences and user-item interactions while avoiding information redundancy. A GNN-based social recommendation model for user homogeneity considers consistent user social relationships, leading to improved recommendation accuracy . These findings collectively highlight the effectiveness of GNNs in enhancing social recommendation systems by integrating social network information and user interactions to mitigate data sparsity and cold-start issues.

Graph Neural Networks (GNNs), a term for these deep neural network topologies, have been proposed to learn meaningful representations of graph information. Their fundamental concept is the use of neural networks to repeatedly collect feature information from small graph neighbors. After transformation and aggregation, node information. might spread across a graph in the interim. As a result, GNNs are naturally integrated with both topological structure and node information, and they have proven to be effective in representation learning. On the other hand, two graphs can be used to represent data in social recommendation. These two plots, Add a user-item graph that shows how users interact with one other and a social graph that shows the relationships between users. Users who can bridge the two graphs are concurrently participating in both. Additionally, social network information is naturally included into user and item latent components learning in social recommendation . Constructing social recommender systems requires learning representations of both products and users. Concurrently, there are difficulties

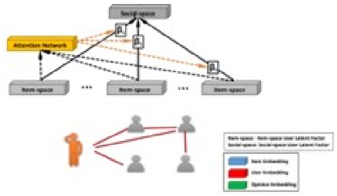

Fig. 1. image. 1. : Graph Data in Social Recommendation. It contains two graphs including the user-item graph (left part) and the user-user social graph (right part). Note that the number on the edges of the user-item graph denotes the opinions (or rating score) of users on the items via the interactions.

in developing social recommender systems based on GNNs. Both the user-item graph and the social graph.

In a social recommendation system, give users' information from several angles. To improve user representations, it is crucial to combine data from the two graphs. So, the first task is to figure out how to naturally join these two graphs. Furthermore, user views on things are included in the user-item graph in addition to interactions between users and items. For instance, depicts how the user interacts with the objects "laptop"and "trousers,"with the user enjoying the former and rejecting the latter.

Thus, gathering user and item opinions and interactions together is the second problem. Furthermore, networks may be created in online environments due to the low cost of connection construction.

Strong relationships allow users to share more similar tastes than weak ties. The performance of recommendations may deteriorate if social relationships are taken into equal consideration. Determining the difference between social relationships with varying intensities is therefore the third problem. handle the three aforementioned issues at the same time. Our principal contributions may be summed up as follows: We provide a principled method to jointly capture opinions and interactions in the user-item graph; we introduce a method to consider heterogeneous strengths of social relations mathematically; we propose a novel graph neural network, GraphRec, which can model graph data in social recommendations coherently; and we demonstrate the efficacy of the proposed framework on a variety of real-world datasets. This is how the rest of the paper is structured. In Section 2, we provide the suggested framework. We perform experiments on two real-world datasets in Section 3 to demonstrate the efficacy of the suggested approach. In Section 4, we examine research pertaining to our

III. MOTIVATION

The utilization of Graph Neural Networks (GNNs) in social recommendation is driven by the necessity to tackle obstacles like data sparsity, cold-start problems, and irregular social interactions. By efficiently utilizing the distinctive graph structures seen in social networks to improve recommendation systems, GNNs give a possible answer. A more thorough understanding of user behavior and preferences is made possible by these networks' ability to record user-to-user relationships, user-item interactions, and social influences. By taking into account the dynamic nature of user interests inside social networks, researchers want to increase recommendation accuracy, reduce data sparsity issues, and improve system performance by integrating GNNs into social recommendation models.

The capacity of Graph Neural Networks (GNNs) to efficiently capture and utilize the rich structural information found in social networks has led to a great deal of interest in the subject of social recommendation. The following primary reasons are the driving forces for the use of GNNs in social recommendation:

1. Representation of Complex Relationships: Social networks are graphs by nature, with nodes standing in for people or things and edges for connections or interactions between them. Because GNNs are specifically made to handle with graph-structured data, they are an excellent choice for simulating the complex interdependencies and linkages seen in social networks.

2. Information Propagation: User preferences and item features in social recommendation systems are frequently impacted by the likes and actions of their social connections. GNNs are efficient at spreading and aggregating.

A. Figures and Tables

a) Positioning Figures and Tables: Place figures and tables at the top and bottom of columns. Avoid placing them in the middle of columns. Large figures and tables may span across both columns. Figure captions should be below the figures; table heads should appear above the tables. Insert figures and tables after they are cited in the text. Use the abbreviation , even at the beginning of a sentence.

IV. LITERATURE SURVEY

GNNs have emerged as a powerful technique for leveraging graph-structured data in recommendation systems, particularly in the context of social recommendations. Several researchers have explored the application of GNNs to incorporate social network information and improve recommendation accuracy.

One of the early works in this area is the Neural Graph Collaborative Filtering (NGCF) model proposed by Wang et al. (2019). The NGCF framework uses GNNs to capture high-order user-item interactions and collaborative signals, resulting in improved recommendation performance compared to traditional collaborative filtering methods.

Building upon NGCF, Wu et al. (2019) introduced the Social-Aware Graph Neural Network (SAGE) model, which explicitly incorporates social relationships into the GNN architecture. SAGE learns user and item embeddings by propagating and aggregating representations through the user-item interaction graph and the social graph, effectively capturing both user preferences and social influences.

Fan et al. (2019) proposed the Graph Neural Networks for Social Recommendation (GNN-SR) model, which integrates social network information, user-item interactions, and auxiliary item knowledge (e.g., item attributes, categories) into a unified GNN framework. GNN-SR employs attention mechanisms to capture the varying importance of social connections and auxiliary item information, leading to improved recommendation performance.

Ying et al. (2018) introduced the Graph Convolutional Matrix Completion (GC-MC) model, which uses graph convolutional networks (GCNs) to learn user and item embeddings from the user-item interaction graph. Although not specifically designed for social recommendation, GC-MC demonstrates the effectiveness of GNNs in capturing high-order structural information in recommendation tasks.

Monti et al. (2017) proposed the Geometric Matrix Completion (GMC) model, which combines GCNs with traditional matrix factorization techniques. GMC leverages the structural information in the user-item interaction graph to enhance the learned user and item embeddings, leading to improved recommendation accuracy.

More recently, Song et al. (2022) introduced the Social-Aware Graph Neural Network with Dual Graph and Dual Attention (SAGNN-DGDA) model. This model constructs two graphs (user-item interaction graph and social graph) and employs dual attention mechanisms to capture both local and global social influences, resulting in improved social recommendation performance.

Several other works have explored variations and extensions of GNNs for social recommendation, such as incorporating temporal dynamics (Gao et al., 2020), handling cold-start scenarios (Wang et al., 2021), or incorporating additional contextual information (Zhang et al., 2021).

Overall, the literature demonstrates the effectiveness of GNNs in capturing complex structural information and social influences in recommendation systems. By leveraging graph-structured data and incorporating social network information, GNNs have shown promising results in improving recommendation accuracy and providing more personalized social recommendations.

V. The Suggested Structure

The concepts and notations used in this study will be introduced first, followed by an outline of the suggested framework, a description of each model component, and a discussion of how to determine the model parameters.

VI. Definitions and Notations

With n representing the number of users and m representing the number of things, let U = u1,u2,...,un and V = v1,v2,...,vm be the sets of users and items, respectively. Assumed to be R R The user-item rating matrix, or user-item graph, has the formula n×m. If ui assigns a rating to vj, then rij is the rating score; if not, we use 0 to denote the rating that ui did not assign to vj, or rij = 0. One way to interpret user ui's rating score rij is as their assessment of item vj. Assume O = ui,vj. —rij, 0 be the collection of recorded ratings where the set of unknown ratings is denoted by T = (ui,vj) —rij = 0—. Let N(i) represent the users with whom you are directly linked, C(i) represent the objects with which you have engaged, and B(j) represent the persons with whom you have interacted with vj. Users can also build social relationships with one another. The user-user social network is represented as T Rnn, where Tij = 1 indicates a relationship between uj and ui and 0 otherwise.

Our objective is to forecast the missing rating value in the user-item graph R, given the social graph T and R. In accordance with , we represent an item vj by using an embedding vector qj R power d and a user ui by using an embedding vector pi R power d.

A. User Modeling

Learning user latent factors—represented as hi Rd for user interface—is the goal of user modeling. .. How to naturally integrate the user item graph and social network is the problem. In order to overcome this difficulty, as seen in Figure 2's left portion, we first employ two different aggregation techniques to identify elements from two graphs. The user-item graph is used to determine the item-space user latent factor h I i R d through the first aggregation, also known as item aggregation. The second aggregation is social aggregation, which learns the latent component h S i R d from the social network of social space users. The final user latent factors hi are then formed by combining these two factors. The next section will cover item and social aggregation as well as how to aggregate user latent characteristics from the two domains of item and social space. The proposed method aims to capture interactions and opinions in the user-item graph by modeling user latent factors through item-space interactions and feedback. By utilizing an items aggregation function and opinion-aware interaction representations, the model combines user-item interactions with user opinions to enhance the understanding of user preferences and characteristics in the item space[3][4]. The approach involves incorporating opinion embedding vectors for different types of opinions, such as ratings, and using a Multi-Layer Perceptron to integrate these opinions with item embeddings. Additionally, the model considers adjusting the aggregation mechanism to assign unique weights to interactions, allowing for a more personalized and nuanced contribution of interactions to a user's latent factor.

The goal is to model a user's preferences based on the items they have interacted with and the feedback (e.g. ratings) they have provided. This is done mathematically by aggregating the information from the user's past item interactions.

For each user-item interaction, an "opinion-aware" vector representation is created that combines the item's embedding vector and the embedding of the user's opinion/rating for that item using a neural network.

To get the overall user preferences, these opinion-aware interaction vectors are aggregated in two possible ways:

1) Mean aggregator: Simply takes the average of all the interaction vectors, treating them equally.

2) Attention aggregator: Assigns different importance weights to the interactions using a neural attention mechanism. Interactions that are more important for capturing the user's preferences get higher weights.

Additionally, the model accounts for social influence - the idea that a user's preferences are affected by their social

SNO	AUTHOR	CONTRIBUTION	RESEARCH GAP
1	Xiang Wang	Graph neural network based recommendation model exploiting collaborative signals.	Complex graphs capture user interests.
2	Richang Hong	Social relationship-aware GNN model for social recommendations.	Dynamic cold-start recommendations.
3	Wenqi Fan	Social network, interactions, knowledge integrated GNN recommendation model.	Multimodal semantic recommendations.
4	Ruining He	GCN learns embeddings from user-item interaction graph.	Social-aware sparse interaction modeling.
5	Chenyan Liu	Dual graphs, dual attention capture social influences.	Scalable explainable social recommendations.

TABLE I
LITERATURE SURVEY

connections. A "social-space user vector" is computed by aggregating the preference vectors of the user's social network neighbors/friends.

Again, two aggregation methods are used:

1) Mean of neighbors' vectors 2) Attention over neighbors, where neighbors with stronger social ties get higher weights

Finally, the user's overall preference vector is obtained by combining their own item-interaction preference vector and the social-influence vector using another neural network layer.

modeling preferences from a user's item interactions, capturing social influence from their connections, and combining these two sources of information into the final user preference representation vector.

B. *Item Modeling*

Item modeling is used to learn the item latent component, zj, for the itemvj via user aggregation, as seen in the right portion of modeling Item modeling in Graph Neural Networks (GNNs) for social recommendation involves extracting high-order features from user-item interactions and incorporating user preferences and item attributes. Traditional recommendation systems face challenges due to sparse data, leading researchers to explore models that consider social relationships[1]. To address the limitations of existing algorithms, a novel model combines GNNs for feature extraction and deep learning for general feature acquisition, enhancing recommendation accuracy through feature fusion with attention mechanisms. Additionally, a GNN-based social recommendation model focuses on user homogeneity by obtaining consistent user embeddings through relational attention, outperforming other models in experiments on mainstream datasets. Another approach introduces a Social Preference Network to capture friend preferences and avoid information redundancy, effectively addressing cold-start problems in recommendation tasks.

Item modeling is a crucial component in graph neural networks (GNNs) for social recommendation systems. Just as user modeling aims to capture user preferences and characteristics, item modeling focuses on representing item features and properties effectively. In the context of social recommendations, item modeling plays a vital role in understanding item characteristics, enabling accurate recommendation predictions, and providing meaningful item representations. Here are some key aspects of item modeling for GNNs in social recommendation:

1. Item Embeddings: Similar to user embeddings, item embeddings are dense vector representations that capture the unique characteristics of each item. These embeddings can be initialized randomly or trained from item metadata, such as item descriptions, categories, or attributes. Item embeddings serve as the initial input features for GNNs, encoding item information in a low-dimensional space.

2. Item Content Modeling: In addition to item embeddings, GNNs can incorporate item content information, such as textual descriptions, images, or other metadata. This content can be processed using techniques like natural language processing (NLP) or computer vision models, and the resulting features can be integrated into the GNN architecture. Modeling item content helps capture semantic information, enhancing the understanding of item characteristics and enabling more accurate recommendations.

3. Item-Item Interactions: GNNs can model item-item interactions by leveraging the graph structure. Items that are related or similar can be connected in the graph, allowing the GNN to propagate and aggregate information between neighboring items. This process can capture item similarities, co-occurrence patterns, and other relationships, enabling better representations of items based on their connections.

4. Attention Mechanisms: Similar to user modeling, attention mechanisms can be employed in item modeling to differentially weight the importance of different item features or interactions. For example, an attention mechanism could prioritize certain item attributes or emphasize the influence of specific neighboring items in the graph, leading to more personalized and context-aware item representations.

5. Dynamic Item Updates: In dynamic recommendation scenarios, where new items are continuously added or item characteristics evolve over time, GNNs can provide a flexible framework for updating item representations. As new interactions or connections are formed, the GNN can propagate and update the item representations accordingly, enabling real-time adaptation to changes in the item space.

6. Social Context Integration: In social recommendation, item modeling can benefit from incorporating social context information. For example, the opinions, preferences, or behaviors of a user's social connections can influence the representation of items, capturing the social dynamics and ensuring that item representations align with the user's social context.

By effectively modeling items, GNNs can capture intricate relationships, content information, and social context, leading to more accurate and personalized social recommendations. Item modeling, in conjunction with user modeling and the integration of social information, enables GNNs to provide comprehensive and contextual representations, ultimately enhancing the overall recommendation quality.

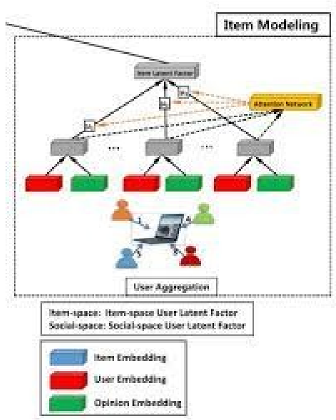

Fig. 2. Item Modeling: For Graph Neural Networks for Social Recommendation

VII. Rating Prediction

We will create recommendation tasks to discover model parameters in this portion. Many recommendation tasks exist, including rating prediction and item ranking. In this study, we utilize the suggested GraphRec model for the rating prediction recommendation job. We may concatenate the latent components of users and objects (i.e., hi and zj) and then input the resulting data into MLP for rating prediction as follows: FORMULA 20-23

VIII. Model Training

GraphRec's model parameters must be estimated by providing an objective function to optimize. Given that rating prediction is the goal we are concentrating on in this paper, a popular objective function is constructed as FORMULA 24 where rij is the ground truth rating that user i assigned to the item j, and —O— is the number of observed ratings. We use the RMSprop to optimize the objective function. Is the optimizer in our application as opposed to the standard SGD. Every time, a training instance is chosen at random, and each model parameter is updated to point in the direction of the negative gradient. Our approach consists of three embeddings: opinion embedding (er), user embedding (pi), and item embedding (qj). During the training phase, they are jointly learnt and randomly initialized. Because the raw features are extremely vast and exceedingly sparse, we do not employ one-hot vectors to represent each user and object. The model may be trained more easily by embedding high-dimensional sparse features into a low-dimensional latent space . The grading scale of the Opinion Embedding Matrix E framework. To indicate ratings in 1, 2, 3, 4, 5, opinion embedding matrix e, for a 5star rating system, has 5 distinct embedding vectors. A persistent issue in deep neural network model optimization is overfitting. The dropout approach has been incorporated into our model in order to mitigate this problem. During training, some neurons are supposed to be dropped at random, according to the dropout theory. Only a portion of the parameters will be changed at a time. Additionally, the entire network is used for prediction because dropout is deactivated during testing.

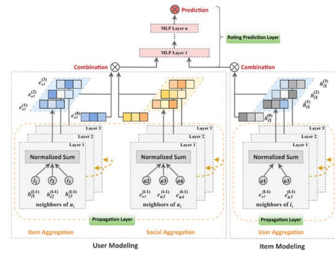

Fig. 3. Model Training :GraphRec's model parameters

IX. Experiment

Datasets. In our experiments, we choose two representative datasets Ciao and Epinions , which are taken from popular social networking websites Ciao (http://www.ciao.co.uk) and Epinions (www.epinions.com). Each social networking service allows users to rate items, browse/write reviews, and add friends to their 'Circle of Trust'. Hence, they provide a large amount of rating information and social information. The ratings scale is from 1 to 5. We randomly initialize opinion embedding with 5 different embedding vectors based on 5 scores in 1, 2, 3, 4, 5. Evaluation Metrics. In order to evaluate the quality of the recommendation algorithms, two popular metrics are adopted to evaluate the predictive accuracy, namely Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) . Smaller values of MAE and RMSE indicate better predictive accuracy. Note that small improvement in RMSE or MAE terms can have a significant impact on the quality of the top-few recommendations . To evaluate the performance, we compared our GraphRec with three groups of methods including traditional recommender systems, traditional social recommender systems, and deep neural network based recommender systems. For each group, we select representative baselines and below we will detail them. Parameter Settings. We implemented our proposed method on the basis of Pytorch , a well-known Python library for neural networks. For each dataset, we used x

Performance Comparison of Recommender Systems

We first compare the recommendation performance of all methods. Table 3 shows the overall rating prediction error w.r.t. RMSE and MAE among the recommendation methods on Ciao and Epinions datasets. We have the following main findings:

• SoRec, SoReg, SocialMF, and TrustMF always outperform PMF. All of these methods are based on matrix factorization. SoRec, SoReg, SocialMF, and TrustMF leverage both the rating and social network information; while PMF only uses the rating information. These results support that social network information is complementary to rating information for recommendations.

• NeuMF obtains much better performance than PMF. Both methods only utilize the rating information. However, NeuMF is based on neural network architecture, which suggests the power of neural network models in recommender systems.

• DeepSoR and GCMC+SN perform better than SoRec, SoReg, SocialMF, and TrustMF. All of them take advantage of both rating and social network information. However, DeepSoR and GCMC+SN are based on neural network architectures, which further indicate the power of neural network models in recommendations.

• Among baselines, GCMC+SN shows quite strong performance. It implies that the GNNs are powerful in representation learning for graph data, since it naturally integrates the node information as well as topological structure.

• Our method GraphRec consistently outperforms all the baseline methods. Compared to DeepSoR and GCMC+SN, our model provides advanced model components to integrate rating and social network information. In addition, our model provides a way to consider both interactions and opinions in the user-item graph. We will provide further investigations to better understand the contributions of model components to the proposed framework in the following subsection.

To sum up, the comparison results suggest.

i. social network information is helpful for recommendations;

ii. neural network models can boost recommendation performance and

iii. the proposed framework outperforms representative baselines

MODEL ANALYSIS

In this subsection, we study the impact of model components and model hyper-parameters.

Effect of Social Network and User Opinions. In the last subsection, we have demonstrated the effectiveness of the proposed framework. The proposed framework provides model components to integrate social network information and incorporate users' opinions about the interactions with items. To understand the working of GraphRec, we compare GraphRec with its two variants: GraphRec-SN, and GraphRec-Opinion. These two variants are defined in the following:

• GraphRec-SN: The social network information of GraphRec is removed . This variant only uses the item-space user latent factor hiI to represent user latent factors hi; while ignoring the social-space user latent factors hiS.

• GraphRec-Opinion: For learning item-space user latent factor and item latent factor, the opinion embedding is removed during learning xia and fjt . This variant ignores the users' opinion on the user-item interactions.

The performance of GraphRec and its variants on Ciao and Epinions . From the results, we have the following findings:

• Social Network Information. We now focus on analyzing the effectiveness of social network information. GraphRecSN performs worse than GraphRec. It verifies that social network information is important to learn user latent factors and boost the recommendation performance.

• Opinions in Interaction. We can see that without opinion information, the performance of rating prediction is deteriorated significantly. For example, on average, the relative reduction on Ciao and Epinions is 3.50

X. RELATED WORK

In this section, we briefly review some related work about social recommendation, deep neural network techniques employed for recommendation, and the advanced graph neural networks. Exploiting social relations for recommendations has attracted significant attention in recent years [27, 28, 37]. One common assumption about these models is that a user's preference is similar to or influenced by the people around him/her (nearest neighbours), which can be proven by social correlation theories [20, 21]. Along with this line, SoRec [17] proposed a co-factorization method, which shares a common latent user-feature matrix factorized by ratings and by social relations. TrustMF [37] modeled mutual influence between users, and mapped users into two low-dimensional spaces: truster space and trustee space, by factorizing social trust networks. SoDimRec [30] first adopted a community detection algorithm to partition users into several clusters, and then exploited the heterogeneity of social relations and weak dependency connections for recommendation. Comprehensive overviews on social recommender systems can be found in surveys [29]. After using a community recognition technique to divide people into many clusters, SoDimRec [30] used weak dependency connections and social relation heterogeneity to provide recommendations. thorough summaries of social Surveys often use recommender systems [29]. Deep neural network models have significantly influenced the development of efficient feature representations in a number of domains, including speech recognition [12], Computer Vision (CV) [14], in recent years. additionally NLP stands for natural language processing [4]. While majority of them employed deep neural networks to simulate auditory aspects of music [32], textual description of things [3, 33], and visual content of pictures [40], several recent studies have used deep neural networks to recommendation tasks and demonstrated promising results [41]. Furthermore, a Neural Collaborative Filtering framework was introduced by NeuMF [11] to learn the non-linear interactions between users and objects. However, until very recently, deep neural networks were rarely used in social recommender systems. Specifically, the NeuMF [11] model was extended to crossdomain by NSCR [35]. social recommendations—that is, suggesting content from information domains to prospective social network users—and introduced a neural network-based social collaborative ranking recommender system. But, NSCR's need for users to have an account on one or more social networks (like Facebook, Twitter, or Instagram) restricts the amount of data that can be collected and how it may be used in real-world scenarios. Social media movie recommendation was created by SMRMNRL [42] using the perspective of learning a multimodal heterogeneous network representation for ranking. They made use of the recurring multimodal neural networks used a random-walk based learning technique to learn the representation of movie posters and textual descriptions using neural networks and convolutional neural networks. Unlike typical social recommender systems, the challenge of cross-domain social recommendations for ranking metric was tackled in all of these efforts [35] [42]. The two neural network tasks that are most pertinent to ours are DLMF [6] and DeepSoR [8]. In order to learn representation for initializing an existing matrix factorization, DLMF [6] employed auto-encoder on ratings.

A two-phase trust-aware recommendation process is suggested, which would use deep neural networks to initialize matrix factorization and combine the interests of the user and their trust friends with the influence of the community effect based on matrix factorization for recommendations. Neural networks for a user's social relationships were incorporated into probabilistic matrix factorization by DeepSoR [8]. They used a pre-trained node embedding approach to represent people at first, and then they used k-nearest neighbors to connect neural networks and user embedding characteristics. It has been demonstrated more recently that Graph Neural Networks (GNNs) may learn from graph structure data [2, 5, 7, 15, 25]. The user-item interaction in the recommender system task.

Includes user ratings for the goods, which is a common graph data. Consequently, it has been suggested that GNNs be used to address the recommendation problem [1, 22, 39]. sRMIn order to extract graph embeddings for people and things, GCNN [22] used GNNs. Recurrent neural networks were then added to the mix to carry out a diffusion process. GCMC [1] introduced a framework for graph auto-encoders that generated latent characteristics of both users and items by means of differentiable message passing on the user-item graph. A randomwalk graph neural network was suggested by PinSage [39] to learn node embeddings in webscale networks.

RATING PREDICTION

Finally, let's define what the recommendation task that is to be used to learn the model parameters. the authors use rating prediction as the recommendation task.

Rating prediction for Graph Neural Networks (GNNs) in social recommendation systems is a crucial aspect addressed by various research studies. GNN-based models aim to enhance rating prediction by leveraging social relationships, user-item interactions, and implicit feedback. These models tackle challenges such as data sparsity, cold-start issues, and over-smoothing problems. Techniques like deep dynamic graph attention frameworks , adaptive collaborative graph neural networks , and multi-head attention mechanisms have been proposed to improve rating prediction accuracy by capturing high-order interactions, addressing social inconsistencies, and deepening user-item relationship understanding. Additionally, enriching models with multiple implicit feedback and constructing triple GCN components have shown significant improvements in rating prediction tasks . These advancements highlight the continuous efforts to enhance the performance of GNNs in rating predictions for social recommendation systems. as showing in fig:5

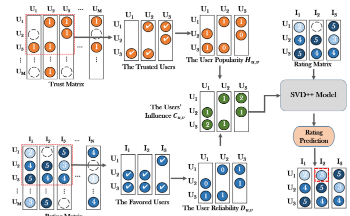

Fig. 4. Rating prediction

METHODOLOGY

1. Data Preparation: a. User-Item Interaction Data: Collect data on user-item interactions, such as ratings, purchases, or implicit feedback (e.g., clicks, views) . b. Social Network Data: Obtain data on the social connections between users, which can be represented as a social graph.

c. Optional: Item Content Data: Gather additional item metadata, such as descriptions, categories, or attributes, if available.

2. Graph Construction: a. User-Item Interaction Graph: Construct a bipartite graph where users and items are nodes, and edges represent interactions between them.

b. Social Graph: Build a graph where nodes represent users, and edges represent social connections between users.

c. Optional: Item Content Graph: If item content data is available, construct a graph where nodes represent items, and edges connect items with similar content or attributes.

3. Graph Neural Network Architecture: a. Input Layer: Define the initial node features, such as user embeddings, item embeddings, and optional item content features.

b. Graph Neural Network Layers: Implement GNN layers that propagate and aggregate information along the edges of the graphs. Common GNN architectures include Graph Convolutional Networks (GCNs), GraphSAGE, or Graph Attention Networks (GATs).

c. Attention Mechanisms: Incorporate attention mechanisms to capture the varying importance of different neighbors or features in the graphs.

d. Social Influence Modeling: Design components that integrate social influence from the social graph into the user and item representations.

e. Output Layer: Define the output layer, which can be a rating prediction, ranking, or classification task, depending on the recommendation objective.

4. Model Training: a. Loss Function: Define an appropriate loss function for the recommendation task, such as mean squared error for rating prediction or ranking loss for personalized ranking.

b. Optimization: Train the GNN model using techniques like gradient descent or variants, minimizing the chosen loss function.

c. Regularization: Apply regularization techniques, such as dropout or weight decay, to prevent overfitting.

d. Negative Sampling: If necessary, employ negative sampling strategies to generate negative examples for training the model on implicit feedback data.

Symbols	Definitions and Descriptions
r_{ij}	The rating value of item v_j by user u_i
$\mathbf{q}_j$	The embedding of item v_j
$\mathbf{p}_i$	The embedding of user u_i
$\mathbf{e}_r$	The opinion embedding for the rating level r, such as 5-star rating, $r \in \{1, 2, 3, 4, 5\}$
d	The length of embedding vector
$C(i)$	The set of items which user u_i interacted with
$N(i)$	The set of social friends who user u_i directly connected with
$B(j)$	The set of users who have interacted the item v_j
$\mathbf{h}_i^I$	The item-space user latent factor from item set $C(i)$ of user u_i
$\mathbf{h}_i^S$	The social-space user latent factor from the social friends $N(i)$ of user u_i
$\mathbf{h}_i$	The user latent factor of user u_i, combining from item space $\mathbf{h}_i^I$ and social space $\mathbf{h}_i^S$
$\mathbf{x}_{ia}$	The opinion-aware interaction representation of item v_a for user u_i
$\mathbf{f}_{jt}$	The opinion-aware interaction representation of user u_t for item v_j
$\mathbf{z}_j$	The item latent factor of item v_j
α_{ia}	The item attention of item v_a in contributing to $\mathbf{h}_i^I$
β_{io}	The social attention of neighboring user u_o in contributing to $\mathbf{h}_i^S$
μ_{jt}	The user attention of user u_t in contributing to $\mathbf{z}_j$
r'_{ij}	The predicted rating value of item v_j by user u_i
$\oplus$	The concatenation operator of two vectors
$\mathbf{T}$	The user-user social graph
$\mathbf{R}$	The user-item rating matrix (user-item graph)
$\mathbf{W}, \mathbf{b}$	The weight and bias in neural network

Fig. 5. Table Notations

5. Model Evaluation: a. Evaluation Metrics: Choose relevant evaluation metrics for the recommendation task, such as Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Recall@k, Precision@k, or Normalized Discounted Cumulative Gain (NDCG).

b. Train-Test Split: Split the data into training, validation, and test sets for proper evaluation and hyperparameter tuning.

c. Evaluate the trained GNN model on the test set using the chosen evaluation metrics.

6. Deployment and Inference: a. Model Serving: Deploy the trained GNN model in a production environment for serving recommendations.

b. Online Updates: Implement mechanisms to update the model with new user-item interactions or social connections, ensuring real-time recommendations.

c. Explanation and Interpretability: Develop techniques to interpret and explain the recommendations generated by the GNN model, if required.

Throughout the methodology, it is essential to consider the specific requirements and constraints of the social recommendation system, such as scalability, real-time inference, and computational resources.

RESULTS AND DISCUSSION

The results and discussions for Graph Neural Networks (GNNs) in the context of social recommendation highlight various advancements in addressing data sparsity and cold-start issues. Studies propose innovative approaches like STL, PLGCN, and PSR to enhance recommendation performance. STL focuses on modifying the interaction graph structure and adaptive sampling to alleviate data sparsity . PLGCN introduces a subgraph construction module to filter out negative messages and improve recommendation accuracy . PSR tackles information redundancy by using different GNNs for social preference networks and user-item interactions, effectively improving recommendation tasks, especially in cold-start scenarios . These approaches demonstrate superior performance in handling data sparsity, cold-start problems, and enhancing recommendation accuracy by leveraging social network information effectively .

CONCLUSION

For rating prediction, we have introduced a Graph Network model (GraphRec) to simulate social recommendation. In particular, we offer a rational method for cooperatively capturing interactions and views expressed in the user-item graph. Our research shows that the opinion data is essential to enhancing the effectiveness of our model. Furthermore, by taking into account the varied strengths of social relationships, our GraphRec is able to distinguish between the connections strengths. Experimental findings on two real-world datasets demonstrate that GraphRec is capable of outperforming the most advanced baselines.

Furthermore, our analysis sheds light on the importance of incorporating both user-item interactions and social connections into the recommendation process. By jointly modeling the user-item graph and the social graph, our approach effectively captures the complex dynamics of user preferences and social influences, leading to more accurate predictions.

However, we acknowledge several limitations of our work, including the scalability of GNNs to large-scale social networks and the need for further research into the interpretability of GNN-based recommendation models. Additionally, while our experiments demonstrate promising results, there is still room for improvement in terms of fine-tuning model architectures and optimizing hyperparameters.

Looking ahead, we envision several exciting avenues for future research. This includes exploring novel graph-based architectures, investigating alternative ways to integrate additional contextual information (such as temporal dynamics or content features), and evaluating the robustness of GNNs to adversarial attacks in social recommendation scenarios.

At the moment, recommendations just take into account the social network, but many real-world sectors also provide richer side information on both individuals and things. Rich qualities, for instance, are linked to both users and items.

An intriguing future step would be to use graph neural networks for attribute-based recommendation. Afterwards, we now take into account the static nature of both ratings and social data. On the other hand, social media and ratings are inherently dynamic. Therefore, we'll think about creating dynamic graph neural networks for social media suggestions using dynamic.

Graph Neural Networks (GNNs) have shown promise in addressing challenges in social recommendation systems. They enhance recommendation performance by leveraging social network information, user-item interactions, and user behavior data[3]. GNN-based models like STL improve recommendation accuracy by modifying graph structures, expanding positive samples, and mining hard negative samples. GNNs for preference social recommendation effectively capture social preferences and user-item interactions while avoiding information redundancy. Furthermore, a GNN-based social recommendation model for user homogeneity considers consistent user social relationships, leading to improved recommendation accuracy . These findings collectively highlight the effectiveness of GNNs in enhancing social recommendation systems by integrating social network information and user interactions to mitigate data sparsity and cold-start issues.

REFERENCES

[1] Rianne van den Berg, Thomas N Kipf, and Max Welling. 2017. Graph convolutional matrix completion. arXiv preprint arXiv:1706.02263 (2017).

[2] Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. 2017. Geometric deep learning: going beyond euclidean data. IEEE Signal Processing Magazine 34, 4 (2017), 18–42.

[3] Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2018. Neural Attentional Rating Regression with Review-level Explanations. In Proceedings of the 27th International Conference on World Wide Web. 1583–1592.

[4] Hongshen Chen, Xiaorui Liu, Dawei Yin, and Jiliang Tang. 2017. A survey on dialogue systems: Recent advances and new frontiers. ACM SIGKDD Explorations Newsletter 19, 2 (2017), 25–35.

[5] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In Advances in Neural Information Processing Systems. 3844–3852.

[6] Shuiguang Deng, Longtao Huang, Guandong Xu, Xindong Wu, and Zhaohui Wu. 2017. On deep learning for trust-aware recommendations in social networks. IEEE transactions on neural networks and learning systems 28, 5 (2017), 1164–1177.

[7] Tyler Derr, Yao Ma, and Jiliang Tang. 2018. Signed Graph Convolutional Networks. In 2018 IEEE International Conference on Data Mining (ICDM). IEEE, 929–934.

[8] Wenqi Fan, Qing Li, and Min Cheng. 2018. Deep Modeling of Social Relations for Recommendation. In AAAI.

[9] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 855–864.

[10] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In Advances in Neural Information Processing Systems. 1024–1034.

[11] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In Proceedings of the 26th International Conference on World Wide Web, WWW. 173–182.

[12] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. 2012. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal Processing Magazine 29, 6 (2012), 82–97.

[13] Mohsen Jamali and Martin Ester. 2010. A matrix factorization technique with trust propagation for recommendation in social networks. In Proceedings of the fourth ACM conference on Recommender systems. ACM, 135–142.

[14] Hamid Karimi, Jiliang Tang, and Yanen Li. 2018. Toward End-to-End Deception Detection in Videos. In 2018 IEEE Big Data. IEEE, 1278–1283.

[15] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In International Conference on Learning Representations (ICLR).

[16] Yehuda Koren. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 426–434.

[17] Hao Ma, Haixuan Yang, Michael R Lyu, and Irwin King. 2008. Sorec: social recommendation using probabilistic matrix factorization. In Proceedings of the 17th ACM conference on Information and Knowledge Management. ACM, 931–940.

[18] Hao Ma, Dengyong Zhou, Chao Liu, Michael R Lyu, and Irwin King. 2011. Recommender systems with social regularization. In Proceedings of the fourth ACM international conference on Web Search and Data Mining. ACM, 287–296.

[19] Yao Ma, Suhang Wang, Charu C. Aggarwal, Dawei Yin, and Jiliang Tang. 2019. Multi-dimensional Graph Convolutional Networks. In Proceedings of the 2019 SIAM International Conference on Data Mining(SDM).

[20] Peter V Marsden and Noah E Friedkin. 1993. Network studies of social influence. Sociological Methods Research 22, 1 (1993), 127–151.

[21] Miller McPherson, Lynn Smith-Lovin, and James M Cook. 2001. Birds of a feather: Homophily in social networks. Annual review of sociology 27, 1 (2001), 415–444.

[22] Federico Monti, Michael Bronstein, and Xavier Bresson. 2017. Geometric matrix completion with recurrent multi-graph neural networks. In Advances in Neural Information Processing Systems. 3700–3710.

[23] Paul Resnick and Hal R Varian. 1997. Recommender systems. Commun. ACM 40, 3 (1997), 56–58.

[24] Ruslan Salakhutdinov and Andriy Mnih. 2007. Probabilistic Matrix Factorization. In 21th Conference on Neural Information Processing Systems, Vol. 1. 2–1.

[25] David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. 2013. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. IEEE Signal Processing Magazine 30, 3 (2013), 83–98.

[26] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research 15, 1 (2014), 1929–1958.

[27] Jiliang Tang, Charu Aggarwal, and Huan Liu. 2016. Recommendations in signed social networks. In Proceedings of the 25th International Conference on World Wide Web. In-

ternational World Wide Web Conferences Steering Committee, 31–40.

[28] Jiliang Tang, Xia Hu, Huiji Gao, and Huan Liu. 2013. Exploiting local and global social context for recommendation.. In IJCAI, Vol. 13. 2712–2718. [29] Jiliang Tang, Xia Hu, and Huan Liu. 2013. Social recommendation: a review. Social Network Analysis and Mining 3, 4 (2013), 1113–1133.

[30] Jiliang Tang, Suhang Wang, Xia Hu, Dawei Yin, Yingzhou Bi, Yi Chang, and Huan Liu. 2016. Recommendation with Social Dimensions. In AAAI. 251–257.

[31] Xiangyu Zhao, Liang Zhang, Zhuoye Ding, Long Xia, Jiliang Tang, and Dawei Yin. 2018. Recommendations with Negative Feedback via Pairwise Deep Reinforcement Learning. In KDD'18. ACM, 1040-1048.

[32] Zhou Zhao, Qifan Yang, Hanqing Lu, Tim Weninger, Deng Cai, Xiaofei He, and Yueting Zhuang. 2018. Social-aware movie recommendation via multimodal network learning. IEEE Transactions on Multimedia 20, 2 (2018), 430-440.

[33] Ye Tao, Ying Li, and Zhonghai Wu. 2021. Revisiting Graph Neural Networks for Node Classification in Heterogeneous Graphs. In 2021 IEEE International Conference on Multimedia and Expo (ICME). IEEE, 1–6.

[34]Xiang Wang, Xiangnan He, Liqiang Nie, and Tat-Seng Chua. 2017. Item silk road: Recommending items from information domains to social users. In Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval. ACM, 185-194.

[35]Rongjing Xiang, Jennifer Neville, and Monica Rogati. 2010. Modeling relationship strength in online social networks. In Proceedings of the 19th international conference on World wide web. ACM, 981-990.

[36]M. Jiang et al., "Social recommendation with cross-domain transferable knowledge", IEEE Trans. Knowl. Data Eng., vol. 27, no. 11, pp. 3084-3097, Nov. 2015.

[37]H. Ma et al., "Recommender systems with social regularization", Proc. 7th ACM Int. Conf. Web Search Data Mining, pp. 287-296, 2011.

[38]X. Yang, H. Steck and Y. Liu, "Circle-based recommendation in online social networks", Proc. 18th ACM SIGKDD Int. Conf. Knowl. Disc. Data Mining, pp. 1267-1275, 2012.

[39]H. Ma et al., "Sorec: Social recommendation using probabilistic matrix factorization", Proc. 17th ACM Conf. Inf. Knowl. Manage., pp. 931-940, 2008.

[40]Y. Shen and R. Jin, "Learning personal+ social latent factor model for social recommendation", Proc. 18th ACM SIGKDD Int. Conf. Knowl. Disc. Data Mining, pp. 1303-1311, 2012.