Graph Neural Networks for Social Recommendation

Submitted in partial fulfillment of the requirements for the award of degree of

BACHELOR OF ENGINEERING
IN
COMPUTER SCIENCE & ENGINEERING

SUBMITTED TO:
Prof. Dr. Alok Kumar

Submitted By:

Chikate Sathwika (21CS002370 ,Sec A-G2)
Sangepu siddhartha(21CS002451, Sec A-G2)
Chinthakuntala Uday Charan(21CS002371, Sec A-G2)

Department of Computer Science & Engineering
Sir Padampat Singhania University
Udaipur 313601 Rajasthan India

Under the supervision of

Dr. Alok Kumar HOD (CSE),
Department of Computer Science & Engineering
Sir Padampat Singhania University
Udaipur 313601 Rajasthan India

CANDIDATES DECLARATION

I am Chikate sathwika student of B.Tech.(CSE), hereby declare that the project titled "**Graph Neural Networks for Social Recommendation** " which is submitted by me to the department of Computer Science & Engineering , School of Engineering, Sir Padampat Singhania University, Udaipur, in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology under the guidance and supervision of **Prof. Dr. Alok Kumar** (Faculty Coordinator), during January to March 2024.

Name and signature of Student: Chikate Sathwika (21CS002370 ,Sec A-G2)
Sangepu siddhartha(21CS002451, Sec A-G2)
Chinthakuntala Uday Charan(21CS002371, Sec A-G2)

Department of Computer Science & Engineering,
Sir Padampat Singhania University,
Udaipur 313601 Rajasthan India.

Date:

ACKNOWLEDGEMENT

This is to certify that the project entitled "**Graph Neural Networks for Social Recommendation**" being submitted by group members, in partial fulfillment of the requirement for the award of Bachelor of Technology, has been carried out under my supervision and guidance.

The matter embodied in this report has not been submitted, in part or in full, to any other university or institute for the award of any degree, diploma or certificate.

Chikate Sathwika (21CS002370)
Sangepu siddhartha(21CS002451, Sec A-G2)
Chinthakuntala Uday Charan(21CS002371, Sec A-G2)

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.

TABLE OF CONTENT:

ABBREVATIONS

- GNN: Graph Neural Network
- RS: Recommendation System
- GCN: Graph Convolutional Network
- GAT: Graph Attention Network
- RNN: Recurrent Neural Network
- CF: Collaborative Filtering
- MF: Matrix Factorization
- SGD: Stochastic Gradient Descent
- ReLU: Rectified Linear Unit
- RMSE: Root Mean Square Error
- MAE: Mean Absolute Error
- API: Application Programming Interface
- GPU: Graphics Processing Unit
- SGC: Simplified Graph Convolution
- MLP: Multi-Layer Perceptron

Chapter 1

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 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.

Chapter 2

Literature Survey

Here we go through the literature survey with summaries of five research papers or projects related to the topic of " **Graph Neural Networks for Social Recommendation**." These papers and projects provide insights into the recent developments in this domain:

Paper Title [1]: We have presented a Graph Network model (GraphRec) to model social recommendation for rating prediction. Particularly, we provide a principled approach to jointly capture interactions and opinions in the user-item graph. Our experiments reveal that the opinion information plays a crucial role in the improvement of our model performance. In addition, our GraphRec can differentiate the ties strengths by considering heterogeneous strengths of social relations. Experimental results on two real-world datasets show that GraphRec can outperform state-of-the-art baselines. Currently we only incorporate the social graph into recommendation, while many real-world industries are associated rich other side information on users as well as items. For example, users and items are associated with rich attributes. Therefore, exploring graph neural networks for recommendation with attributes would be an interesting future direction. Beyond that, now we consider both rating and social information static. However, rating and social information are naturally dynamic. Hence, we will consider building dynamic graph neural networks for social recommendations with dynamic.

Paper Title[2]:This paper proposes RelationalNet, a novel neural influence and diffusion algorithm for social recommendations based on Diffnet++ [13]. RelationalNet improves upon Diffnet++ with the addition of item–item and item–user interactions modeled as graphs used in the GNNs. By incorporating not only user–user relations in the form of the social graph (similar to other social recommender systems) but also item–item relations that are independent of any user interactions, RelationalNet addresses the cold-start problem of little or no user–item ratings. Furthermore, RelationalNet utilizes a multi-layer diffusion network that employs graph attention to combine graph and node-level representations, managing to capture the latent features of each graph. The experimental evaluation showed that the RelationalNet algorithm achieves better performance in generating trecommendations (with a 11.8% improvement of HR and 6.2% improvement of NDCG on average) as opposed to the current SOTA and predecessor, Diffnet++, showing the potential of such extensions that enhance the input with additional connections.There is still significant scope to explore, like the different mechanisms for forming connections between items, calculating user and item similarities, exploring higher-order connections [31,32] instead of edges, and investigating graph reasoning algorithms to learn users' preferences better. In addition, we plan to leverage our previous work on influential users [2,23,33,34] to refine the user–user similarities and create neighborhoods of influence. Perhaps one of the main shortcomings of the proposed approach is the overhead that is introduced by adding one additional graph and expanding the social network beyond direct neighbors, as the RelationalNet algorithm was 25% slower than the DiffNet++ algorithm. The biggest cost factor to consider if we want to make our algorithm scalable is to restrict the attention modules in the nodes and in the graphs. We can achieve this by reducing the dimensions of the embeddings for the user and item,respectively. One of the future ideas is to use subgraph sampling for the training. However,selecting representative subgraphs that will preserve good graph characteristics needs further investigation, especially in our case, where we have four graphs. While using subgraph sampling will increase the scalability of the model, this comes with the cost of accuracy and expressability of the model.Overall, the RelationalNet algorithm provides a significant step forward in social recommendation systems by taking advantage of social connections, item correlations, and leveraging the power of GNNs

Paper Title[3]: In this work, we combine the effects of modeling the purchase time and social relations in recommender systems, and propose a novel temporal enhanced graph model for social recommendation. Our model characterizes the real purchase time information between items as a special temporal relation in our constructed temporal graph. Experimental results demonstrate the effectiveness of our proposed temporal enhanced graph model in social recommendation **Paper .**

Chapter 3

Software Requirement Analysis

3.1 Problem Definition

Graph neural networks (GNNs) have shown great promise for social recommendation tasks by effectively leveraging graph-structured data representing social networks and user-item interactions. The key steps involve constructing a graph with nodes for users, items, and other entities, and edges capturing relationships like friendships and interactions. GNNs then propagate and aggregate information from each node's neighborhood, learning meaningful representations of users and items.

These learned representations encode the graph structure and node/edge features, enabling accurate predictions of ratings, top recommendations, or future interactions. Carefully designed GNN architectures, objective functions tailored to the recommendation task, and appropriate optimization techniques are critical for achieving high performance. Ultimately, GNNs harness the rich information in social graphs to provide personalized recommendations accounting for users' social connections and preferences.

3.2 Software Requirements:

1. Graph Data Structure Implementation: A robust and efficient data structure to represent and store the graph data, including nodes (users, items, etc.), edges (interactions, relationships), and associated features.

2. Graph Construction and Preprocessing: Tools or libraries for constructing the graph from raw data sources , handling data cleaning, feature extraction, and graph preprocessing tasks.

3. GNN Model Architecture: A flexible and modal framework for defining and implementing different GNN architectures, such as Graph Convolutional Networks (GCNs), Graph Attention Networks (GATs), or custom architectures specific to the social recommendation task.

4. Node and Edge Feature Handling: Mechanisms to incorporate node and edge featuresinto the GNN model, allowing for feature-based representation learning.

5. Neighborhood Sampling and Aggregation: Efficient algorithms and data structures for sampling and aggregating neighborhood information during the GNN's message-passing process, particularly for large-scale graphs.

6. Parallelization and Distributed Computing: Support for parallelization and distributed computing, either through multi-threading, GPU acceleration, or distributed frameworks to handle the computational demands of training GNNs on large-scale social network data.

7. Model Training and Optimization: Implementation of various optimization algorithms and loss functions suitable for different social recommendation tasks along with support for techniques like regularization and early stopping.

8. Evaluation Metrics: A suite of evaluation metrics relevant to social recommendation, such as precision, recall, hit rate, Normalized Discounted Cumulative Gain (NDCG), and tools for computing and analyzing these metrics.

9. Model Serialization and Deployment: Mechanisms for serializing trained GNN models, loading pre-trained models, and deploying models for inference and recommendation generation in production environments.

11. Visualization and Interpretation: Tools for visualizing and interpreting the learned representations, attention weights, and other model internals, aiding in model debugging, analysis, and explainability.

Chapter 4

Hardware Requirements:

- **Computer:** A desktop or laptop computer with sufficient processing power and memory for data analysis, machine learning, and deep learnig.
- Storage: Adequate storage space for datasets, project files, and model checkpoints.

- **Operating System:** The project can be developed graph Neural Networks for Social Recommendation
- **Internet Connection:** Required for data retrieval, library updates, and vscode ,Jupiter applications.
- **Development Tools**: A keyboard, mouse, and monitor for coding and development tasks.

Modules and their functionalities

1. Data Ingestion and Preprocessing:

   - Functionality: Ingest and preprocess data from various sourcesto construct the graph structure and associated node/edge features.

   - Sub-modules: Data ingestion, data cleaning, feature extraction, graph construction.

2. Graph Representation and Storage:

   - Functionality: Efficiently represent and store the graph data structure, including nodes, edges, and associated features.

   - Sub-modules: Graph data structures, indexing, caching.

3. Model Architecture:

   - Functionality: Define and implement various GNN architectures for social recommendation, such as Graph Convolutional Networks (GCNs), Graph Attention Networks (GATs), or custom architectures.

- Sub-modules: GNN layers (e.g., graph convolution, attention), message passing, neighborhood aggregation, feature propagation.

4. Training and Optimization:
   - Functionality: Train the GNN model using appropriate optimization algorithms and loss functions for social recommendation tasks.
   - Sub-modules: Optimization algorithms (Adam, SGD), loss functions, regularization techniques, early stopping.
5. Evaluation and Metrics:
   - Functionality: Evaluate the performance of the trained GNN model using relevant metrics for social recommendation, such as precision, recall, hit rate, and Normalized Discounted Cumulative Gain (NDCG).
   - Sub-modules: Evaluation metrics computation, performance analysis, model comparison.
6. Inference and Recommendation Generation:
   - Functionality: Generate personalized recommendations for users based on the trained GNN model and the graph data.
   - Sub-modules: Batch inference, real-time inference, recommendation ranking, explanation/interpretation .

7. Model Manageme:
   - Functionality: Manage the lifecycle of GNN models, including serialization, versioning, and deployment.
   - Sub-modules: Model serialization, model versioning, model registry, model deployment.
8. Visualization and Interpretation:
   - Functionality: Visualize and interpret the learned representations, attention weights, and other model internals to aid in model debugging, analysis, and explainability.
   - Sub-modules: Graph visualization, embedding visualization, attention visualization, model interpretation techniques.

_Chapter 5

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 Convo lutional 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 inte grate 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 per sonalized 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 sam pling strategies to generate negative examples for training the model on implicit feedback data

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 Cumula tive 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.

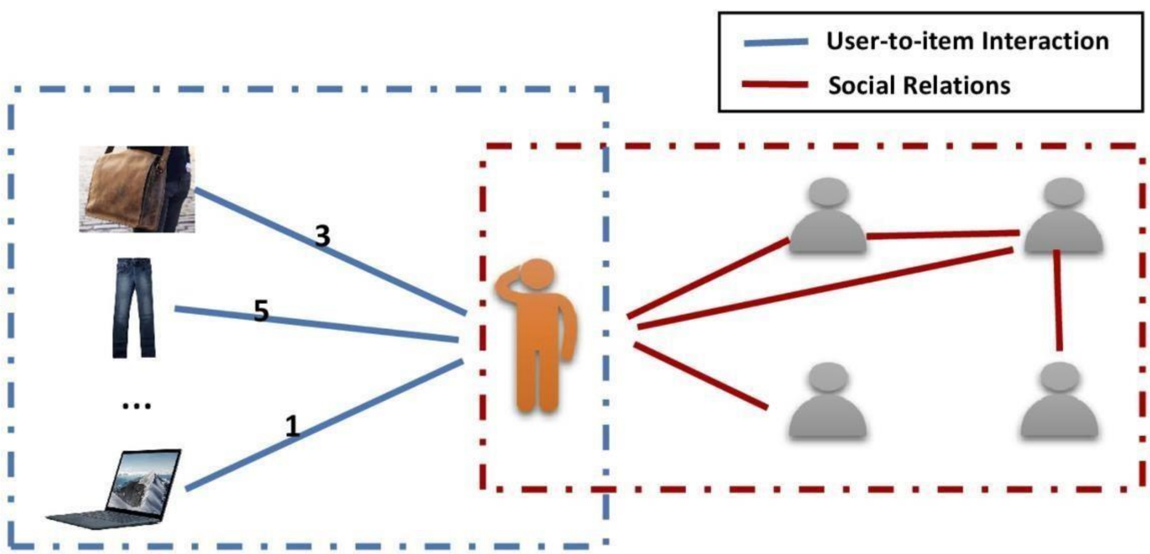
b. Online Updates: Implement mechanisms to update the model with new user-item interactions or social connections, ensuring real-time recommendations.

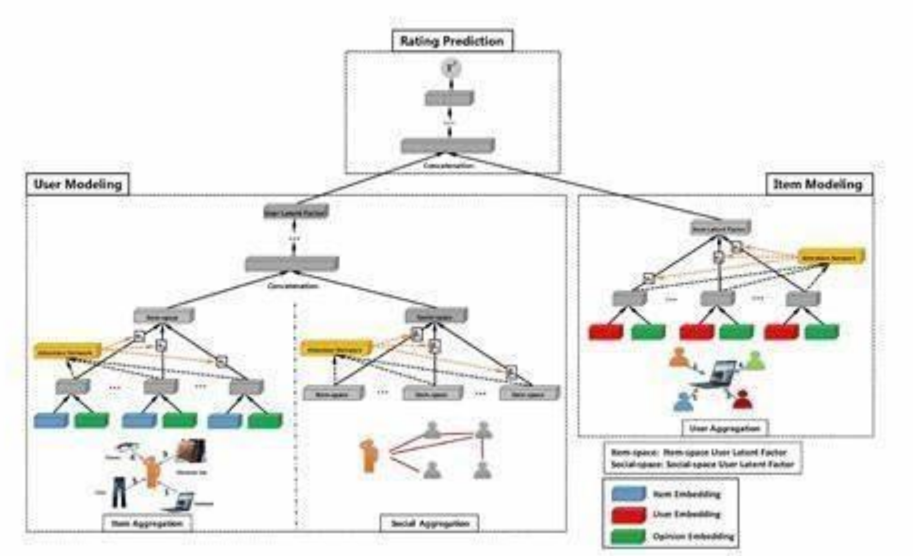
c. Explanation and Interpretability: Develop techniques to interpret and explain the recommendations generated by the GNN model, if required. Throughout the methodology, it is essential to consider the specific requirements and constraints of the social recommen dation system, such as scalability, real-time inference, and computational resources.

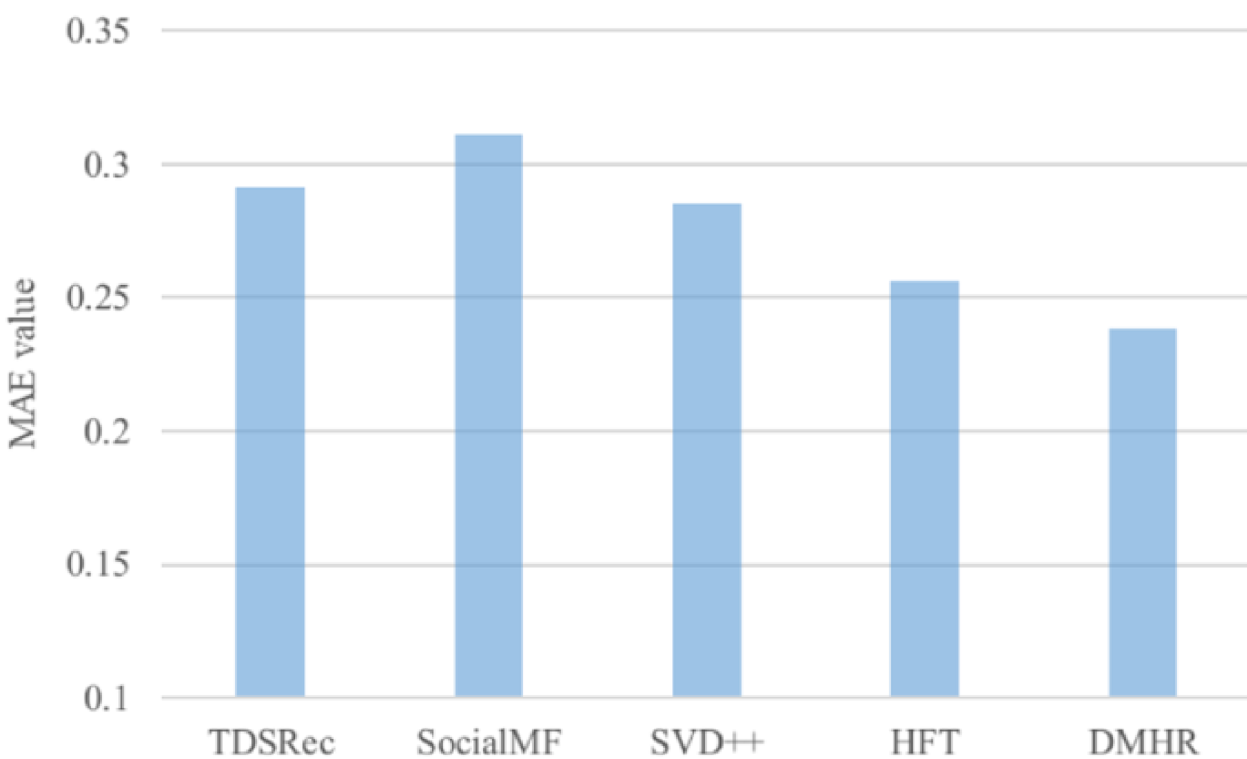
Chapter 6

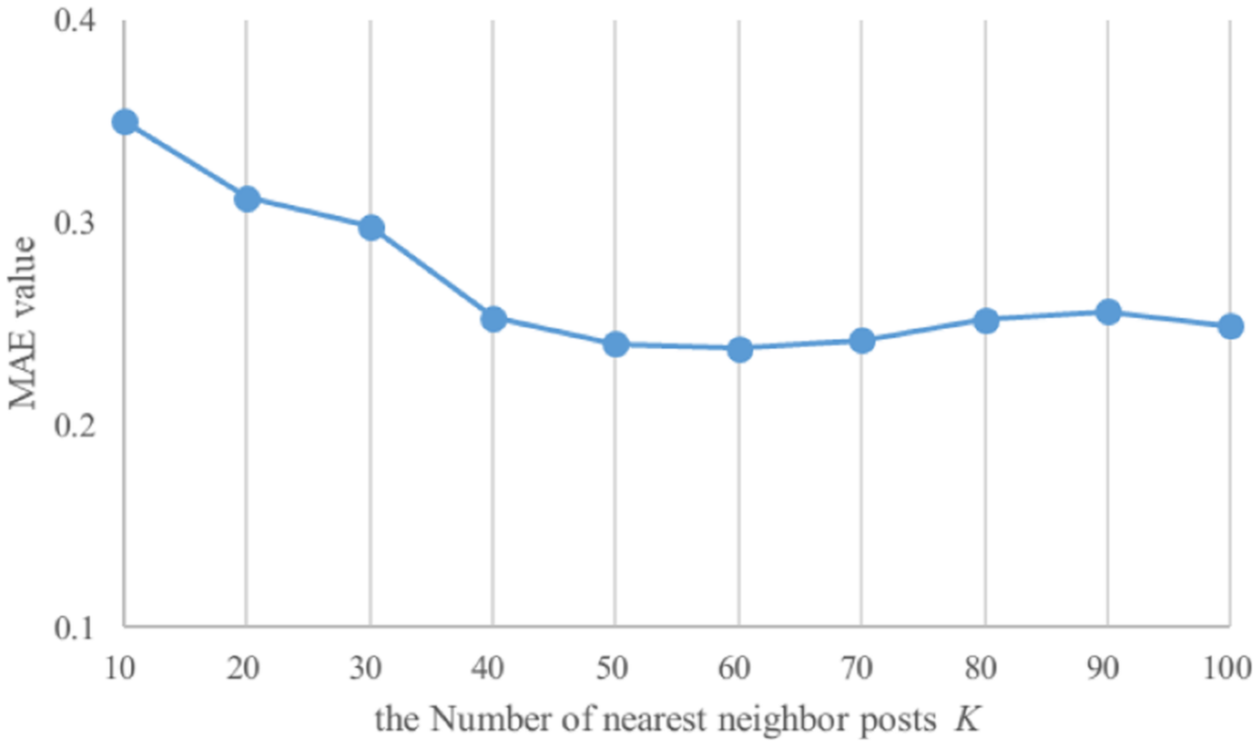
Results And Discussion

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 intro duces 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 sce narios . These approaches demonstrate superior performance in handling data sparsity, cold-start problems, and enhancing recommendation accuracy by leveraging social network infor mation effectively .




Rating Prediction
Concatenation
User Modeling
Item Modeling
Concatenation
Item Aggregation
Social Aggregation
User Aggregation
Item-space: Item-space User Latent Factor
Social-space: Social-space User Latent Factor
Item Embedding
User Embedding
Opinion Embedding


0.35
0.3
0.25
0.2
0.15
0.1
MAE value
TDSRec
SocialMF
SVD++
HFT
DMHR



Chapter 7

Conclusion

For rating prediction, we have introduced a Graph Net work 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 connec tions into the recommendation process. By jointly modeling the user-item graph and the social graph, our approach effec tively 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 net works 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 finetuning model architec tures 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 ad ditional 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. GNN-based models like STL improve recommen dation 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 infor mation 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 effective ness of GNNs in enhancing social recommendation systems by integrating social network information and user interactions to mitigate data sparsity and cold-start issues.

Chapter 8

Reference

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

[2] Michel M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. 2017. Geometric deep learn ing: 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] Micha¨ el Defferrard, Xavier Bresson, and Pierre Van dergheynst. 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, Xin dong Wu, and Zhaohui Wu. 2017. On deep learning for trust aware recommendations in social networks. IEEE transac tions 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 recommen dation 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 neigh borhood: a multifaceted collaborative filtering model. In Pro ceedings 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 con ference 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 Convo lutional Networks. In Proceedings of the 2019 SIAM Interna tional 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 Bres son. 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. Prob abilistic 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 emerg ing 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.