CS3319 Project Proposal

Kaijiang Deng Kaiwen Zhu

1 Introduction

In this era of information explosion, we cannot and do not need to read all information personally. To alleviate such problems, a recommender system is needed to help people obtain the most relevant information.

In order for the recommender system to be effective, it must accurately model preferences of users and features of items according to existing information, like interactions between users and items, relationships among users, and relationships among items. To achieve the goal of learning the desired representations, deep learning is a powerful tool. Also, considering that such information among users and items could be naturally organized into a graph structure, Graph Neural Network (GNN) could be very promising in recommender systems.

In this work we focus on recommendation problems in the academic scenario. That is, given the cooperation relationships among researchers, the citation relationships between researchers and papers, and the citation relationships among papers, the task is to recommend papers to researchers.

One of the essential differences between recommendations in this scenario and other scenarios is that, there are very strong relationships among researchers and among papers, which is the foundation of researchers conducting research. Therefore, a key challenge is how to exploit the information in the communities to enhance the representations of researchers and papers.

Our work is intended to overcome such challenges and develop an effective algorithm to recommend papers to researchers by GNN. Besides, from the results, we hope to gain some insights into the nature of academic networks.

2 Related work

In the past decades, to effectively capture the non-linear and non-trivial user-item relationships and easily incorporate abundant data sources, deep learning has been a dominant methodology for recommender systems [1]. Further, graph learning provides a unified perspective to model heterogeneous data and encode the crucial signals to improve user and item representations. Guided by this motivation, there have been lots of research in GNN over recommendation scenarios.

Generally speaking, the main idea of GNN is to build a graph suitable for the given data, and then iteratively aggregate feature information from neighbors and integrate the aggregated information with the current node representation during the propagation process. As for the network architecture, GNN stacks multiple propagation layers, which consist of aggregation and update operations [1]. Most of the related works focus on different ways of these two operations in order to effectively and efficiently propagate information among entities [2]. For instance, GCN (Graph Convolutional Network) uses mean function to aggregate [3], while GAT (Graph Attention Network) leverages attention mechanism to differentiate neighbors [4].

According to the type of information used, existing works for non-sequential recommendations could be classified into three categories: user-item collaborative filtering, social recommendation, and knowledge graph based recommendation.

User-Item Collaborative Filtering Given interactions between users and items, we could use the items interacted by a user to enhance the representation of this user, and use the users having interacted with an item to enrich the representation of this item. This basic idea of user-item collaborative filtering is straightforward to the goal of predicting edges between users and items. As for the construction of the graph, one way is to directly apply GNN on the original

user-item bipartite graph [5], while another way is to enrich the original graph by introducing new edges [6] or nodes [7].

Social Recommendation According to the social influence theory that connected people would influence each other, we can safely assume that users with strong social relationships are inclined to have similar representations. GNN is especially suitable for this goal courtesy of its ability to simulate how users are influenced by recursive social diffusion process provided with data on social relationships of users. Based on this assumption, many works try to incorporate social relationships into representations. For example, DGRec dynamically infers influences of neighbors during learning to differentiate friends [8], and ESRP use the autoencoder to modify observed social relationships by filtering irrelevant relationships [9].

Knowledge Graph Based Recommendation If knowledge of items is available, we could also utilize this to enhance the item representations. Besides the benefit of improving item representations, another benefit is the results could be more interpretable. When constructing the graph, one popular direction is to incorporate users nodes into the knowledge graph [10]. When learning, many works apply attention mechanism to distinguish different relations in the knowledge graph [11].

3 Research plan

3.1 Algorithm

According to the data description, the given information mainly includes co-authors, citations between papers, and citations of papers by authors. However, it is hard to directly obtain some other important relationships, such as authors of each paper and citations between authors, from the input files, while these relationships may carry relevant information as well as that given. Therefore, as is shown in Algorithm 1, our plan is to take them into consideration by adding edges to the original graph.

Algorithm 1:

Data: $\mathcal{G}$: heterogeneous graph of authors and papers, w_r: weight for edges between each paper and its possible authors, w_c: weight for edges between authors with possible citations, n: number of epochs

Result: F_a: features of authors, F_p: features of papers

Initialize F_a and F_p;
Add weighted edges between author-paper pairs sharing one or more citations;
for every author a **do**
    Add weighted edges between a and the neighbor authors of the new neighbors of a;
for i from 1 to n **do**
    Update F_p by adjacent nodes of papers;
    Update F_a by adjacent nodes of papers;
    Update F_a by adjacent nodes of authors;
    Update F_p by adjacent nodes of authors;

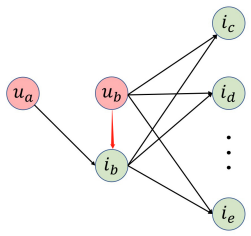

Figure 1: A Possible Author-Paper Graph

For each paper, its citations would be covered by any author of it. Figure 1 illustrates this relationship. In Figure 1, red nodes represent authors, green nodes represent papers, black

arrows represent citations, and the red arrow means that u_b is the author or i_b. Here, papers cited by i_b, including i_c, i_d, i_e, *etc.*, are all cited by u_b. In Algorithm 1, we would add weighted edges between all author-paper pairs sharing one or more citations, like u_b-i_b, and then give them a label. For each author-paper pair a-p, the weight of the edge between them equals to $\frac{|C_a \cap C_p|}{|C_p|}$, where C_a contains all papers cited by a and C_p contains all papers cited by p. It is clear that the higher the weight is, the more possible it is that the author writes the paper. After that, we could obtain possible citations between authors and papers and add edges between them. For example, since u_b is an author of i_b and u_a has cited i_b, we can add an edge between u_a and u_b. Similarly, the weight of the edges between the authors can share the same weight between the corresponding author-paper edges above, *e.g.* $w(u_a, u_b) = w(i_b, u_b)$, where w denotes weight of edges. These edges would be given a different label.

After these operations, we can apply GNN to the graph. Features of authors and papers would be updated alternately by their neighbors. Note that neighbors connected by different types of edges would have different extent of influence controlled by hyperparameters w_r and w_c. For each node, we may just sample some neighbors each time to reduce the amount of computation. Attention mechanisms or RNN might be introduced to learn the different influence degree of neighbors [1, 2]. Finally, whether to recommend a paper p to an author a would be determined by their features.

3.2 Timeline

Our timeline is listed in Table 1. In practice, we may adjust our plan flexibly.

Table 1: Timeline

Start Time	**End Time**	**Tasks**
May. 9th	May. 22nd	Further research on related work and model implementation
May. 23rd	Jun. 6th	Test and optimization
Jun. 6th	Jun. 12th	Draw conclusions and write the research paper

4 Expected outcome

Our model is expected to achieve better performance than just directly applying GNN to the original graph since the "author-write-paper" relationship and citations between authors can help aggregate more useful information of the community of each node to its features. An ablation study on the edge-adding operation would be conducted to test the effect of our algorithm.

5 Backup plan

If our algorithm does not have a good performance, we would first change the sequence of updating. We may also directly apply some related work on social network to the original graph in search of better results.

References

[1] Shiwen Wu, Fei Sun, Wentao Zhang, Xu Xie, and Bin Cui. Graph neural networks in recommender systems: A survey, 2022.

[2] Shoujin Wang, Liang Hu, Yan Wang, Xiangnan He, Quan Z. Sheng, Mehmet A. Orgun, Longbing Cao, Francesco Ricci, and Philip S. Yu. Graph learning based recommender systems: A review, 2021.

[3] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks, 2017.

[4] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks, 2018.

[5] Rianne van den Berg, Thomas N. Kipf, and Max Welling. Graph convolutional matrix completion, 2017.

[6] Jianing Sun, Yingxue Zhang, Chen Ma, Mark Coates, Huifeng Guo, Ruiming Tang, and Xiuqiang He. Multi-graph convolution collaborative filtering. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 1306–1311, 2019.

[7] Xiang Wang, Hongye Jin, An Zhang, Xiangnan He, Tong Xu, and Tat-Seng Chua. Disentangled graph collaborative filtering. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 1001–1010, New York, NY, USA, 2020. Association for Computing Machinery.

[8] Weiping Song, Zhiping Xiao, Yifan Wang, Laurent Charlin, Ming Zhang, and Jian Tang. Session-based social recommendation via dynamic graph attention networks. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, WSDM '19, page 555–563, New York, NY, USA, 2019. Association for Computing Machinery.

[9] Junliang Yu, Hongzhi Yin, Jundong Li, Min Gao, Zi Huang, and Lizhen Cui. Enhancing social recommendation with adversarial graph convolutional networks. *IEEE Transactions on Knowledge and Data Engineering*, 34(8):3727–3739, 2022.

[10] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, page 950–958, New York, NY, USA, 2019. Association for Computing Machinery.

[11] Ze Wang, Guangyan Lin, Huobin Tan, Qinghong Chen, and Xiyang Liu. Ckan: Collaborative knowledge-aware attentive network for recommender systems. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 219–228, New York, NY, USA, 2020. Association for Computing Machinery.