Recommendation of Papers to Authors Sharing Bidirectional Relations

Considering User-Item, User-User and Item-Item Relations

Shuyang Jiang

Haoxuan Xu

Yu Zhou

***Abstract*—In recent years, graph representation learning has attracted a great deal of attention. To combine information between user-item, a lot of work such as GCN which groups features within several hops or GAT, which learns distinguished features from directly connected neighbors. In this group project, given three parts of relation, i.e., user-item, user-user, and item-item relationship, we intend to use *NGCF* or *LightGCN* to model the user-item relationship, which adopting random walk followed by a bi-directional LSTM network to model the user-user or item-item relationship, namely the social relationship.**

***Key Words*—NGCF, LightGCN, Bi-directional LSTM, Collaborative Filtering**

I. Introduction

Recently, social media platform boosts the development of recommendation system. And a general approach to make a recommender system is to build a deep learning model that encodes both users and items in a shared space, thus making them directly comparable. [1] For the fundamental role of graph-structured data in recommender systems [2] [3] , GNN became the most attractive technique for its superior performance in node classification and link prediction of heterogeneous and isomorphic graphs. And in this project, we try to find a good model in citation recommendation scenarios with the given dataset.

More specifically, given a bipartite network representing the citation relationship between authors and papers, one co-authorship network and one citation network among papers, we can build an academic network as the provided document says, and adapt and apply several state-of-art models to solve a link prediction problem. Besides, we luckily have the initial semantic features for papers generated by USE(Universal Sentence Encoder), which might be useful for model training.

II. Related Work

GCN [4] is a good way to model the relationship between user-item pair for several reasons. First, it can capture information not restricted to a certain neighborhood as we can stack several GCN layers to increase the receptive field. Second, convolutional neural network is simple and proved to work in other fields, like [5], [6]. Therefore, it is natural to apply it in graph representation learning to gather features from neighborhood nodes. There are a series of variant version of vanilla GCN, like NGCF [7], which uses another way to execute convolutional operations, or LightGCN [8] which notices that when loss of semantic information, NGCF is parameter-redundant and can further deteriorate the performance of the whole recommendation system.

Attention [9] is also another kind of way to gather information from other nodes. To model the user-item interaction, we can take users as query and items as keys and values. The sequence is formed by id of each user or item. However, it is not scientific for more than 10 thousand users or items, thus prompting to just model directly connected user-item pair. With existence of rating matrix, the attention mechanism can easily distinguish each connected item for a specific user [10]. But without a proper rating matrix, attention may find it hard to distinguish different items given little semantic information.

Random walk is sufficiently powerful in graph neural network [11], [12]. It is randomly executed, meaning that each calling can bring new information and gather new feature for one query node. The length of walk can be arbitrarily long, meaning that we can manually control the receptive field. Also when targeting at different tasks, we can make random walk more like Depth-first Search to gather more global information, or more like Breadth-first Search to gather more local information. Random walk is extremely flexible and applicable in graph neural network.

III. Research Plan

A. Overview

Our model consists of four components, author-author modeling, paper-paper modeling, author-paper modeling and link prediction. We plan to adopt random walk followed by a **Bi-LSTM** to model the first two, from which we can get h_{ai} for author i and h_{pj} for paper j; then we set h_{ai} and h_{pj} as the initial embedding and run **NGCF** or **LightGCN** on author-paper network; finally we use the dot-product of final embedding in GCN to do link prediction. In our understanding, the embeddings obtained from Bi-LSTM in co-author network aggregate the research interest of co-authors, and by putting the embeddings in GCN, they'll aggregate the research interest of authors who have citated same papers. The mathematical notations used in this paper are summarized in Table I.

Symbols	Definitions and Descriptions
p_j	The embedding of paper j
q_i	The embedding of author i
d	The dimension of embedding vectors
$N(i)$	The set of co-authors of author i
$M(j)$	The set of paper that cite paper j
$X(i)$	The set of paper that author i have read
h_{ai}	Hidden state of LSTM or Transformer of author i
h_{pj}	Hidden state of LSTM or Transformer of paper j
e^l_{ai}	Layer embedding of author i in layer l
e^l_{pj}	Layer embedding of paper j in layer l
e_{ai}	Final embedding of author i in GCN
e_{pj}	Final embedding of author j in GCN
T	The length of random walk

TABEL I
NOTATION OF WHOLE MODEL

B. Author-Author Modeling and Paper-Paper Modeling

To model the item-item relationship(both author-author relationship and paper-paper relationship can be regarded as item-item relationship), due to loss of rating matrix, we cannot directly model the relationship with author-paper indirect relationship as those did [10]. Instead, we intend to model paper-paper relationship in a similar way to author-author relationship because of the fact that citation relationship can be treated as a social relationship. Those who share the same research interest are more likely to cite the same paper or cooperate with each other. The same theory can be applied to paper-paper relationship since if one paper cites another one, this paper and the cited one must focus on the same research topic, thus sharing similar features. There are various kinds of ways to model the one-end relationship, such as attention network in [10], or random walk in [13]. Considering that author-author or paper-paper relationship will not restrict within one hop region, we use bi-directional LSTM [14] network or attention-based network which fits our intuition that distant nodes will affect less than neighborhood nodes, but also integrate features of nodes not restricted within one hop neighborhood. BiLSTM is superior to simple LSTM because when we conduct a random walk from a node n to a node m with length T, simple uni-directional LSTM can only aggregate information from m to n, while the hidden state of m cannot aggregate the feature of n. BiLSTM enables one time of random walk to update T nodes embeddings instead of just several ones around the selected node. Another architecture that can also model the bi-directional relationship is the Transformer-encoder [15], which is more efficient but contains more parameters. We will in later work test both types of network to achieve efficiency-performance balance. The most simple way is to execute self-attention over all paper or authors, but considering that it is computationally unbearable, we can just choose a trajectory produced by random walks to simulate the global gathering while maintaining high efficiency and well-distinguished feature gather process.

C. Author-Paper Modeling

GCN [4], [16] is proposed to group features for an arbitrary node. However, it simply regard neighbor nodes as the same important for query node, which is not scientific because some node that directly connect to a node are more important and others distant from the node are less important. Meanwhile, when we only have id representation of authors, corporation of excessive parameters may deteriorate the performance of the whole model [8]. Therefore, we propose to conduct some experiments on **NGCF** and **LightGCN** to compare the performance, and choose the best one to model the user-item relationship, i.e., paper and authors relationship. And we plan to use the embedding from item-item modeling as the initial embedding.

D. Link Prediction

After modeling each paper and author embeddings output by GCN, we can dot-product to estimate the relationship of author-paper pair. The higher dot-product value is, the higher probability that the author favors the paper, which means that we can recommend the paper to this author. Those idle pairs serve the testing set while the other pairs with labels could serve as validation set and training set. Of course the $||\boldsymbol{e_a}||$ or $||\boldsymbol{e_p}||$ may be larger than 1, so before executing the dot-product operation, we will first make it as $\frac{\boldsymbol{e_a}}{||\boldsymbol{e_a}||}$ and $\frac{\boldsymbol{e_p}}{||\boldsymbol{e_p}||}$ to guarantee that the final dot-product result is the probability that the author favors the paper.

IV. Code Implementation Plan

In this section, we will introduce how we implement our work.

- First, we will construct random embeddings for both author and papers.
- Second, for each point in author(paper), generate a random walk for author(paper) graph. Take nodes in random work as input for author(paper) LSTM. Use the output of author(paper) LSTM as the initial embedding of the node in GCN model.
- Third, implement GCN model.
- Fourth, train LSTM and GCN as a whole model. Update parameters from LSTM and GCN and initial random embeddings at the same time.

Some back up plans.

- For the initial random embeddings, we may change it to node2vec as a better start up.
- For random walk in the second step, we may change it to some other searching methods like BFS. Our aim is to aggregate a node and its neighbors information. So we may also use Transformer encoder to replace LSTM.

V. Expected Outcome

We believe our recall on test set must be comparable with mere LightGCN or NGCF because we include more information and social interaction. However, because we have not finished the architecture of our model, we cannot state out an accuracy number for the recall value on test set clearly. We will soon conduct a great deal of experiments to check out our research plan.

References

[1] S. Wu, W. Zhang, F. Sun, and B. Cui, "Graph neural networks in recommender systems: A survey," *CoRR*, vol. abs/2011.02260, 2020. [Online]. Available: https://arxiv.org/abs/2011.02260

[2] R. v. d. Berg, T. N. Kipf, and M. Welling, "Graph convolutional matrix completion," *arXiv preprint arXiv:1706.02263*, 2017.

[3] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for web-scale recommender systems," in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 974–983.

[4] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[5] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *Advances in neural information processing systems*, vol. 27, 2014.

[6] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[7] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering," in *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*, 2019, pp. 165–174.

[8] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "Lightgcn: Simplifying and powering graph convolution network for recommendation," in *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, 2020, pp. 639–648.

[9] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[10] W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang, and D. Yin, "Graph neural networks for social recommendation," in *The world wide web conference*, 2019, pp. 417–426.

[11] L. Lovász, "Random walks on graphs," *Combinatorics, Paul erdos is eighty*, vol. 2, no. 1-46, p. 4, 1993.

[12] B. Tadić, "Exploring complex graphs by random walks," in *AIP Conference Proceedings*, vol. 661, no. 1. American Institute of Physics, 2003, pp. 24–27.

[13] W. Fan, Y. Ma, D. Yin, J. Wang, J. Tang, and Q. Li, "Deep social collaborative filtering," in *Proceedings of the 13th ACM Conference on Recommender Systems*, 2019, pp. 305–313.

[14] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "Lstm: A search space odyssey," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 10, pp. 2222–2232, 2016.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[16] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, and H. Li, "T-gcn: A temporal graph convolutional network for traffic prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 9, pp. 3848–3858, 2019.