DL4DS Mid-Project Check-in: Suggested Academic Reading Sequence Recommendation Using Graph Neural Networks

Bowen Li

March 31, 2024

Abstract

Increasing output in the volume of academic publications over the years will inevitably lead to problems when it comes to allowing a user to be able to pick out papers that are relevant and/or useful to them, with citations being the only sense of structure one can rely on. Incidentally, graph neural networks has seen a recent surge in interest and developments, presenting new ways to apply deep learning techniques on data that can be represented in the structure of a graph. This project seeks to apply graph neural networks to create a recommendation system that, given a paper/topic, will provide users with a suggested sequence of papers to read in order to further understand the paper/topic, ultimately assisting users in navigating the ever increasingly complex landscape of academic publications. The work in progress code can be found at this repository or at the url: `https://github.com/lib250/publication-recommender`.

Introduction

As interest in the fields of STEM has grown over the past few years, and inevitably will continue to grow in the coming years, the number of academic publications will undoubtedly skyrocket. Although this certainly serves as an indicator of the accelerating advances made across these fields, it could also pose a problem for future generations, hobbyists and academics alike, as it increasingly makes the overhead of searching through and selecting relevant reading material more costly.

Typically, when an interested party happens upon a relevant paper, the references of that paper could serve as a suitable starting point for further reading. However, not only could the references of the paper be dubious due to some of the meta motivations concerning the publication and review process, it is still lacking in terms of reading about future developments. After all, a paper is not physically able to cite something that, as of its writing, has yet to exist.

This leaves the interested party to rely on search engines, scouring relevant conferences (the volume of which is only increasing), and gaming their own social media recommendations. The last option could involve one following relevant figures on social media, whom in turn follow other relevant figures that may eventually all be taken into account when recommending new papers to users. This is reminiscent of a graph structure, and is suitable for users who may wish to keep up with the latest in terms of the field. In seeking to build a recommendation system that provides more structure to the users however, we can still leverage this idea of graphs.

Graph neural networks (GNNs) have seen a recent surge in popularity as a paradigm that applies deep learning techniques to data with a graph structure. One of the main problems that it addresses is link prediction, wherein a graph is assumed to have missing edges, and the task is, for a given node, to predict which of the other nodes in the graph it would most feasibly share an edge with. With a good choice of nodes and edges, this task steadily lends itself to being applied as a recommendation system.

This project will explore the potential of link prediction to be applied to a recommendation system that will provide users with a suggested reading sequence of academic publications in order to familiarize themselves with a given topic. This way, if a user is considering a paper, they would be recommended papers to read as prerequisites and suggestions for further reading. In the same vein, if a user has a topic in mind (e.g. attention mechanism), they would be recommended readings to familiarize themselves with the fundamentals of the topic, and readings that explore the applications and further developments of that topic.

Related Work

The task of link prediction is a well-studied problem across various fields. [1] provides a survey on various classical link prediction approaches and comparisons. [2, 3] both provide classical link prediction solutions in the context of recommender systems, one based on vector similarity and the other using bipartite graphs.

In terms of GNNs, [4, 5] provide surveys and taxonomies of GNN techniques applied to recommender systems. [6] presents a particular GNN approach to the recommendation problem based on a new version of the dual-tower model for heterogeneous graphs. [7] presents another GNN approach to creating a recommender system based on a heterogenous graph with users, articles, and article topics as nodes. Most notably, the topic nodes are latent, and are thus learned from the dataset. [8] presents a GNN approach that reconstructs graph components into dense item-item graphs based on a user's history and perceived interests. [9] presents a GNN approach to creating a recommender system for Massively Open Online Courses (MOOCs), though doesn't provide a recommended sequence to take courses in.

Proposed Work

The target recommendation system is one wherein a user is able to provide either a topic or a published paper, and the system provides a sequence of recommended prior readings and recommended further readings. The length of the sequence and number of successive prerequisites can be adjusted via the user's preferences.

This project will primarily involve creating a proof-of-concept for the target recommendation system using GNNs.

The graph representation of the dataset of papers will be instrumental in deciding the approaches and limitations.

The original plan was to have a heterogeneous graph with different node embedding types for publications and topics which could aggregate publications, wherein the topic embeddings could be found through techniques such as LDA as in [7].

However, since having node embeddings for topics introduces an extra layer of complexity for what is merely a convenient redundancy, creating and experimenting on graphs with topic nodes has been pushed back in terms of priority. The reason being that it is likely more fruitful and in alignment with the core of the problem to first verify the feasibility on a homogenous graph with only publication embedding nodes. That is, to begin with, we will explore development of a model that will take an academic publication and provide suggested related publications that may serve well as prior and further readings.

The GNN architecture used will likely be a foundational architecture such as graph convolutional networks (GCN) through the PyTorch Geometric (PyG) library. Time-permitting, other architectures such as graph attention networks could be explored. As mentioned before, exploration of the incorporation of topic node embeddings could also serve as a notable stretch goal.

Datasets

Originally, the proposed datasets were citation graphs from the Stanford Network Analysis Project (SNAP)[10]. However, since these datasets were comprised of papers in the fields of high energy physics theory and high energy physics phenomenology, it would have been difficult to qualitatively assess the recommendations output by the model. As such, we will shift to a dataset that primarily has publications in the field of computer science.

The dataset we will primarily use is the Cora_ML dataset from the paper presenting Graph2Gauss [12], which is a subset of machine learning related papers extracted from the Cora dataset [13]. The graph contains node embeddings of 2995 papers, connected by 8416 directed edges. With a vocabulary size of 2879, each node embedding is a one-hot-encoding of whether or not a vocabulary word is present in the paper.

Dataset Exploration

Dataset Agreement

PyG provides built-in methods to install distributions of certain datasets, however, the issue is that it is only possible to download the graph itself via PyG, meaning that we would only have access to the node embeddings and adjacency information. However, if we want to qualitatively assess the model's performance, it would be pertinent to have a way to retrieve the original papers from the graph and predictions, and as such we wouldn't be able to use PyG's built-in loader alone.

The GitHub repository of [12], contains .npz files for the Cora_ML dataset that we can use, although it presents a slight problem when compared with the same dataset from PyG's loader. Although the adjacency information between the 2 sources of the same dataset are identical, the node embeddings are not. It is unlikely that the indexing of each publication is different, as the adjacency information is represented using the indices of the original papers, so in all likelihood the node embeddings themselves are different between the graph retrieved from the G2G repository and the one retrieved from PyG's loader. Further investigation is needed to determine why exactly the embeddings do not match, although may not be necessary as we can test the performance of a model on both representations.

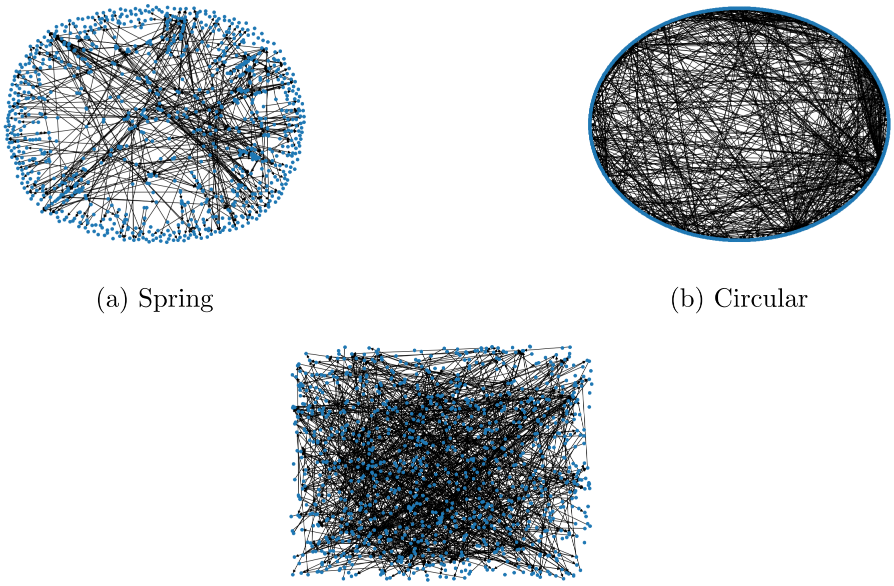
Dataset Structure

Although the node embeddings differed between the two sources, the structure of the graph was the same. Both graphs had 2995 nodes represented by 2879-dimensional embeddings, connected by 8416 directed edges. Both graphs had an average node degree of 2.81 and have no isolated nodes (all publications were either cited by another in the dataset or cites another in the dataset).

There were attempts made to visualize graphs, however, due to the high-dimensional embeddings and general volume of the graph, creating visualizations were impractical. Several attempts were made using the NetworkX library to visualize a subset of the graph with various different graph drawing techniques, but the visualizations for the most part were not very informative, as shown in Figure 1.

Evaluation

We evaluate a GCN for link prediction model based on [14]. The encoder component has 2 graph convolutional layers with output channels 128 and 64 to transform the nodes into an embedding space. During training, negative samples are also drawn to serve as negative examples for the decoder's classification task. For training, we use an Adam optimizer with learning rate 0.001 and binary cross entropy loss as the loss function. We evaluate model



(a) Spring

(b) Circular

(c) Random

Figure 1: NetworkX visualizations of a random sample of 1000 nodes from the dataset using different drawing layouts. The visualizations are not very informative. More examples are available in the notebook in the repository.

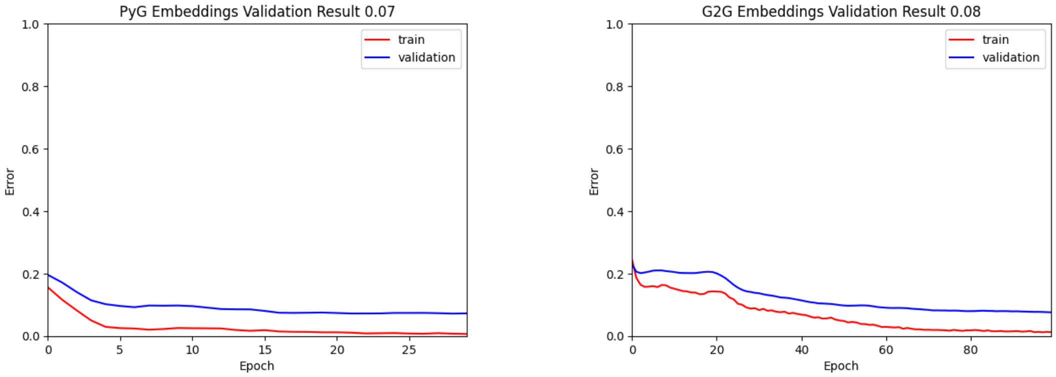
performance using ROC-AUC as is convention with link prediction tasks and the result of the model on the graphs gathered from both sources is shown in Figure 2.

Note that the scales of the x axes are different between the two graphs. It seems that the node embeddings are indeed different between the two data sources as the behaviors differ greatly during training. While the PyG graph's error rate converges at around 10 epochs, the error rate for the G2G repository's graph behaves erratically before converging much later at around 60 epochs.

Next Steps

Week 7: Come up with some qualitative evaluations of the current model. Take some of the predictions of the model and map it back to the abstracts to see if they are reasonable predictions. Perhaps test on a paper from outside the dataset as well. See if the qualitative evaluations are in line with the validation metrics.

Week 8-9: Currently, the model only predicts in one direction. Experiment with different



(a) PyG Data

(b) G2G Repo Data

Figure 2: Training loss curves for the GCN model evaluated on both (a) the graph from the PyG loader and (b) the graph from the G2G repository. Note that the x axis scales are different between the two graphs.

ways of getting predictions in other directions. One possibility is simply reversing the graph directions, but another possibility could be assigning edge weights that represent directional information (+1 for forward relationship, -1 for backwards relationship). In addition, PyG supports edge *features*. Time-permitting, more exploration with edge embeddings could be done.

Alternatively, different ways to embed nodes could also be explored which is a point where we could introduce the previously discussed topic embeddings. Could also explore different ways to embed an arbitrary user input to be usable in the link prediction task so that the user could ”look up” papers related to a certain topic using keywords or natural language.

Week 10: Wrap-up and report.

Conclusion

The project aims to apply GNNs to build a proof-of-concept for a recommendation system to assist an interested party in navigating the ever-growing expanse of available academic works.

The real-world dataset of academic publications would be represented in an embedding space as nodes along with relevant topics, and edges would be directed from citing papers to cited papers. In solving a link prediction task, one could retrieve a set of nodes that are likely to point towards a given node, and a set of nodes that the given nodes are likely to point towards.

After some refinement based on the user's preferences, the user would be presented with a recommended sequence of papers to read in order to both be onboarded to and to gain deeper understanding of a desired topic.

References

[1] Ajay Kumar, Shashank Sheshar Singh, Kuldeep Singh, Bhaskar Biswas. *Link prediction techniques, applications, and performance: A survey.* Physica A: Statistical Mechanics and its Applications, 553, 2020. `https://doi.org/10.1016/j.physa.2020.124289`

[2] Zhan Su, Xiliang Zheng, Jun Ai, Yuming Shen, Xuanxiong Zhang. *Link prediction in recommender systems based on vector similarity.* Physica A: Statistical Mechanics and its Applications, 560, 2020. `https://doi.org/10.1016/j.physa.2020.125154`

[3] T. Jaya Lakshmi, S. Durga Bhavani. *Link prediction approach to recommender systems.* Computing, 2023. `https://doi.org/10.1007/s00607-023-01227-0`

[4] Shiwen Wu, Fei Sun, Wentao Zhang, X Xie, Bin Cui. *Graph Neural Networks in Recommender Systems: A Survey.* ACM Computing Surveys, 55(5): 1-37, 2022. `https://doi.org/10.48550/arXiv.2011.02260`

[5] Zijian Liang, Hui Ding, Wenlong Fu. *A Survey on Graph Neural Networks for Recommendation.* International Conference on Culture-oriented Science & Technology (ICCST): 383-386, 2021. `https://doi.org/10.1109/ICCST53801.2021.00086`

[6] Qiang He, Xinkai Li, Biao Cai. *Graph neural network recommendation algorithm based on improved dual tower model.* Scientific Reports, 14, 3853, 2024. `https://doi.org/10.1038/s41598-024-54376-3`

[7] Linmei Hu, Chen Li, Chuan Shi, Cheng Yang, Chao Shao. *Graph neural news recommendation with long-term and short-term interest modeling.* Information Processing & Management, 57(2), 2020. `https://doi.org/10.1016/j.ipm.2019.102142`

[8] Jianxin Chang, Chen Gao, Yu Zheng, Yiqun Hui, Yanan Niu, Yang Song, Depeng Jin, Yong Li. *Sequential Recommendation with Graph Neural Networks.* Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval: 378-387, 2021. `https://doi.org/10.48550/arXiv.2106.14226`

[9] Jingjing Wang, Haoran Xie, Fu Lee Wang, Lap-Kei Lee, Oliver Tat Sheung Au. *Top-N personalized recommendation with graph neural networks in MOOCs.* Computers and Education: Artificial Intelligence, 2, 2021. `https://doi.org/10.1016/j.caeai.2021.100010`

[10] Johannes Gehrke, Paul Ginsparg, Jon Kleinberg. *Overview of the 2003 KDD Cup.* SIGKDD Explorations 5(2): 149-151, 2003. `http://www.cs.cornell.edu/home/`

kleinber/kddcup2003.pdf. Retrieved from https://www.kaggle.com/datasets/wolfram77/graphs-citation/data?select=cit-HepTh-abstracts.

[11] Mathurin Aché. *Citation Network Dataset.* March 2021. Retrieved from https://www.kaggle.com/datasets/mathurinache/citation-network-dataset/data

[12] Aleksandar Bojchevski, Stephan Günnemann. *Deep Gaussian Embedding of Graphs: Unsupervised Inductive Learning via Ranking.* February 2018. https://doi.org/10.48550/arXiv.1707.03815

[13] Andrew McCallum, Kamal Nigam, Jason Rennie, Kristie Seymore. *Automating the Construction of Internet Portals with Machine Learning.* July 2000. https://doi.org/10.1023/A:1009953814988

[14] Thomas N. Kipf, Max Welling. *Variational graph auto-encoders.* NIPS Workshop on Bayesian Deep Learning. November 2016. https://doi.org/10.48550/arXiv.1611.07308