EDGE-HETEROGENEOUS GNNS: OGBN-ARXIV EXPERIMENTS

TECHNICAL REPORT

Khang V. Ly*
Faculty of Natural Science, Mathematics and Computer Science
University of Amsterdam
Amsterdam, North Holland, 1098 XH
ly.vy.khang@gmail.com

July 17, 2023

ABSTRACT

In this technical report, we describe a framework to enrich the `ogbn-arxiv` dataset with additional metadata. We use the provided mapping from node IDs to Microsoft Academic Graph (MAG) IDs to filter through the complete July 2020 MAG snapshot hosted on AMiner; retrieved features, e.g. author IDs, tagged fields of study, can be used to construct additional edge types, transforming the homogeneous graph into a edge-heterogeneous graph. We also use the provided raw titles and abstracts to infer SciBERT embeddings to use as node features. We experiment with several backbone models, including GCN, GraphSAGE, and simplified graph convolution (SGC).

Keywords Heterogeneous Graph Learning · Article Classification · Document Relatedness

1 Introduction

The `ogbn-arxiv` dataset consists of Computer Science papers from arXiv hand-labeled into 40 subject areas by paper authors and arXiv moderators. It is a homogeneous citation network consisting of 169,343 nodes and 1,166,243 edges. Node features are also available, constructed using textual information by averaging the embeddings of words (which are generated with the Skip-Gram model) in the articles' titles and abstracts. The dataset provides the mapping used between papers' node IDs and their original MAG IDs, as well as the raw texts and abstracts.[2]

The proposed framework focuses on applying heterogeneous graph enrichment techniques to paper node classification, which is typically a homogeneous task. Existing works on heterogeneous graphs often consider multiple node types, expanding from article to entity classification; we exclusively investigate heterogeneity of paper-to-paper relationships to remain consistent with the single-node type problem setting.

Hence, we combine edge heterogeneity, SciBERT embeddings [Beltagy et al., 2019], and the PyG implementation of the R-GCN message passing duplication technique [Schlichtkrull et al., 2018], to approach the `ogbn-arxiv` classification task from a heterogeneous perspective and improve final performance.

2 Methodology

Since MAG (and the associated API) has since been discontinued, a July 2020 snapshot of the complete MAG index (240M papers) hosted by AMiner's Open Academic Graph project was downloaded and filtered to obtain additional metadata [Zhang et al., 2022]. The data is split into 17 compressed chunks, each chunk containing 3 text files of approximately 10GB each.[3] The text files are organized as per the indicated schema, and lines can be neatly parsed as

*Use footnote for providing further information about author (webpage, alternative address)—*not* for acknowledging funding agencies.

[2]Download link for raw text here.

[3]OAG links and data schema here.

JSON records; all chunks were downloaded locally and metadata of IDs corresponding to papers in `ogbn-arxiv` were saved out.[4] Of the available features, there are 3 of interest that are potential indicators of paper relatedness: *authors*, *published venue*, and *fields of study*. Logistical metadata, e.g. DOI, volume and page numbers, are not useful for our purposes. Hence, we convert the downloaded dataset into a PyG `HeteroData` object, with the following edge types:

- Citations: as provided.
- (Co)-Authorship: Two papers are connected if they share an author, with a corresponding edge weight indicating the number of shared authors.
- Venue: Two papers are connected if they were published at the same venue. This feature was tested in the ablation study, but omitted from the final submission as it did not improve final performance. The relationship is likely too low-level, e.g. might fail to capture same-label papers published exclusively online at arXiv and presented at a physical conference, respectively, which are arbitrary differences for this task.
- Fields of Study: Two papers are connected if they share at least one field, with an edge weight based on the number of shared fields. Fields of study, e.g. "computer science," "neural networks," etc. are automatically assigned with an associated confidence score (which we do not use), and each paper can have multiple fields of study, making them functionally similar to keywords.

All edge lists are *undirected*, and all edge weights are normalized using logistic sigmoid. Since venue and fields lead to massive subgraphs, the mean number of papers associated with each feature is calculated, and edges are only created between that many sampled nodes for each unique field/venue. Some metrics for each subgraph are shown in Table 1.

	$\lvert\mathcal{E}\rvert$	Avg. Degree	Edge Homophily Ratio	# Non-Isolated Nodes
References	2,315,598	13.67	0.654	169,343
(Co-)Authorship	6,749,335	39.86	0.58	157,067
Fields of Study	8,279,687	48.89	0.319	144,714
Venue	600,930	3.55	0.077	17,848

Table 1: Summary of edge type subgraphs.

Regarding node features, the provided SkipGram features are substituted with SciBERT embeddings inferred on the raw titles and abstracts, using the base `scibert-scivocab-uncased` model without additional fine-tuning. Node2Vec was also tested, by generating a set of Node2Vec embeddings for each edge type subgraph, and concatenating them with the SciBERT embeddings to represent both the textual and structural domain [Grover and Leskovec, 2016]. This was found to negatively impact performance relative to exclusively using SciBERT, and hence was dropped. Theoretically, mixing conflicting NLP and non-NLP features could be detrimental to node separability and hence classification performance, as nodes could be structurally close, but possess distinct textual metadata, for instance.

Models are converted to support heterogeneous data using the PyG `to_hetero` method, an implementation of the modeling technique described in R-GCN wherein the message passing functions are duplicated and applied individually for each relationship type [Schlichtkrull et al., 2018]. The mean is used as the aggregation operator.

3 Experiments

	n_layers	hidden_features	weight_decay	starting_lr	dropout
GCN	[2*, 3]	[256*, 512]	[0, 0.001*, 0.003, 0.005]	[0.01*, 0.001]	[0, 0.2*, 0.5]
GCN+JK	[2*, 3]	[128, 256*, 512]	[0, 0.003*, 0.005]	[0.01*, 0.001]	[0*, 0.5]
SAGE	[2*, 3]	[256*, 512]	[0, 0.001*, 0.003, 0.005]	[0.01*, 0.001]	[0, 0.2*, 0.5]
SGC	[2*, 3]	N/A	[0*, 0.001, 0.005]	[0.1, 0.01*]	N/A

Table 2: Selected hyperparameters. Note that for SGC, `n_layers` refers to K i.e. the number of hops.

Experiments were conducted on a Databricks cluster with the `g4dn.2xlarge` EC2 instance (32GB RAM, 1 NVIDIA Tesla T4 16GB VRAM). Models are trained transductively with tuned parameters as specified in Table 2 and early stopping based on validation accuracy (upper limit of 500 epochs). The embeddings were pre-computed using multi-GPU distributed inference. We employ a scheduler that scales down the learning rate as the validation loss plateaus. Ablation results are also provided in Table 3 to examine the impact of the different edge types (averaged across three runs only) with a GCN base. Accuracy is reported using the relevant OGB evaluator.

[4]The notebook `ogbnarxiv_process_mag_data`, used to filter the MAG metadata locally, is included on the repo for reference.

Model	Subgraphs	Embeddings	Test Acc.
GCN	References	SkipGram (Default)	69.9%
	References	SciBERT	73.1%
	References, Authorship		74.7%
	References, Authorship, FoS		75.1%*
	References, Authorship, FoS, Venue		75%
	References, Authorship, FoS	SciBERT, N2V	74.8%

Table 3: GCN ablation results averaged across 3 runs.

The proposed data preparation scheme is tested with several GNN architectures. We consider two GCN setups - base and with a jumping knowledge module using concatenation as the layer aggregation mechanism - as well as GraphSAGE [Kipf and Welling, 2017, Xu et al., 2018, Hamilton et al., 2017]. We also run experiments with the simplified graph convolutional operator (SGC) [Wu et al., 2019]; the increased graph footprint can lead to scalability concerns, hence the performance of such lightweight methods is of interest. Results are listed in Table 4.

Model	Val. Acc.	Test Acc.	# Params	Baseline Acc.
GCN	0.7586 ± 0.0012	0.7461 ± 0.0006	621,944	0.7174 ± 0.0029
GCN+JK	0.7629 ± 0.0007	0.7472 ± 0.0024	809,512	0.7219 ± 0.0021
SAGE	0.7605 ± 0.0007	0.7461 ± 0.0013	1,242,488	0.7149 ± 0.0027
SGC	0.7515 ± 0.0005	0.7419 ± 0.0004	92,280	0.6581 ± 0.0007

Table 4: Results averaged across 10 runs for several GNNs. For reference, the baseline results on the unmodified dataset are also displayed. For GCN, JKNet, and GraphSAGE, they are taken from the official leaderboard; the SGC baseline was obtained by using only the reference subgraph and provided SkipGram features, with the same hyperparameters.

Considering the confidence intervals, our best-performing and most consistent result is the base 2-layer GCN.

The additional structural and semantic information provided by edge heterogeneity and BERT-based embeddings consistently improves final performance of a variety of hetero-transformed GNN frameworks, compared to their homogeneous counterparts. These improvements occur even though the added edge type subgraphs possess suboptimal graph properties, e.g. lower edge homophily ratio and isolated nodes; in particular, strong homophily is implicitly assumed by many message passing GNN frameworks [Ma et al., 2022]. Notably, SGC benefits substantially from this data preparation scheme ($\sim 8.38\%$ increase in test accuracy).

However, as we use full-batch training on a larger graph with 768-dimensional features, GPU memory usage is increased to approximately 14GB when using GCN, even with memory-efficient aggregations using sparse matrices supported by PyG.

4 Conclusion

In this report, we discussed a data preparation pipeline for `ogbn-arxiv` utilizing BERT-based node features and edge heterogeneity based on MAG metadata, to encode additional signals of document relatedness within the graph. Several graph-based classifiers were tested on this transformed dataset, including GCN, GraphSAGE, and SGC, with a notable performance boost over the original homogeneous dataset being observed in all cases. Though, the increased memory requirements may prohibit the application of more heavyweight models to this transformed dataset.

References

Iz Beltagy, Kyle Lo, and Arman Cohan. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China, November 2019. Association for Computational Linguistics. doi:10.18653/v1/D19-1371. URL `https://aclanthology.org/D19-1371`.

Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In Aldo Gangemi, Roberto Navigli, Maria-Esther Vidal, Pascal Hitzler, Raphaël Troncy, Laura Hollink, Anna Tordai, and Mehwish Alam, editors, *The Semantic Web*, pages 593–607, Cham, 2018. Springer International Publishing. ISBN 978-3-319-93417-4.

Fanjin Zhang, Xiao Liu, Jie Tang, Yuxiao Dong, Peiran Yao, Jie Zhang, Xiaotao Gu, Yan Wang, Evgeny Kharlamov, Bin Shao, Rui Li, and Kuansan Wang. Oag: Linking entities across large-scale heterogeneous knowledge graphs. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–14, 2022. doi:10.1109/TKDE.2022.3222168.

Aditya Grover and Jure Leskovec. Node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 855–864, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450342322. doi:10.1145/2939672.2939754. URL https://doi.org/10.1145/2939672.2939754.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=SJU4ayYgl.

Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. Representation learning on graphs with jumping knowledge networks. *CoRR*, abs/1806.03536, 2018. URL http://arxiv.org/abs/1806.03536.

William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 1025–1035, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Felix Wu, Tianyi Zhang, Amauri H. Souza Jr., Christopher Fifty, Tao Yu, and Kilian Q. Weinberger. Simplifying graph convolutional networks. *CoRR*, abs/1902.07153, 2019. URL http://arxiv.org/abs/1902.07153.

Yao Ma, Xiaorui Liu, Neil Shah, and Jiliang Tang. Is homophily a necessity for graph neural networks? In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=ucASPPD9GKN.