CAN GRAPH NEURAL NETWORKS LEARN NODE-LEVEL STRUCTURAL FEATURES?

Manuel Dileo & Matteo Zignani
Department of Computer Science
University of Milan, Italy
{manuel.dileo, matteo.zignani}@unimi.it

ABSTRACT

Graph Neural Networks (GNNs) have become one of the most widely adopted solutions for graph machine learning (GML) tasks. They perform feature learning on graphs using message passing on the network structure, avoiding the feature engineering step required for traditional tabular approaches for GML tasks. However, it is unclear which structural features GNNs can or cannot easily learn from data, especially for node- and edge-level properties. In this work, we propose a methodology to investigate which structural features GNNs can reconstruct from graph data. We conducted a first experimental analysis on one of the most used benchmarks for GML, considering some of the most well-known node-level features, such as centrality and transitivity measures. The results show that GNNs can easily reconstruct PageRank and in/out-degree centralities. But, surprisingly, GNNs can also learn centrality measures based on shortest path distances. Moreover, they reach quite good performance in learning the local clustering coefficient.

1 INTRODUCTION

Graph Neural Networks (GNNs) have emerged as a prominent choice in addressing graph machine learning (GML) tasks; in particular to identify network attacks, anonymous traffic, or spam by exploiting the relations between different connected entities (Yuan et al., 2024). By engaging in feature learning on graph structures through message passing (Gilmer et al., 2017), they circumvent the need for feature engineering inherent in traditional tabular approaches for GML tasks (Liu et al., 2016). Although Xu et al. (2019) theoretically shows that GNNs are at most expressive as the Weisfeiler-Lehman (1-WL) test for graph isomorphism, it is not clear to what extent GNNs can learn from data, i.e. predict with good approximation, structural features, which, for example, underpin graph-based approaches for network security and botnet detection (Lagraa et al., 2024). This is especially true for node- and edge-level tasks (e.g. node classification, link prediction), on which GNN embeddings are typically augmented with traditional feature engineering (You et al., 2020). In this work, we propose a methodology to investigate which structural features GNNs are able to reconstruct from the graph data. We model the problem as a regression task, initializing the node embeddings with random features, which typically strengthen GNNs. We conducted a first experimental analysis on Planetoid (Yang et al., 2016), considering some of the most well-known node-level features, such as centrality and transitivity measures. The results show that GNNs can easily reconstruct PageRank and in/out-degree. While the reconstruction capability for Page Rank and the degree should be quite expected since the analogies between the Page Rank formulation and the message passing paradigm, it is more surprising that GNNs can also learn centrality measures based on shortest-path distances, and reach quite good performance in learning the local clustering coefficient.

2 METHODOLOGY

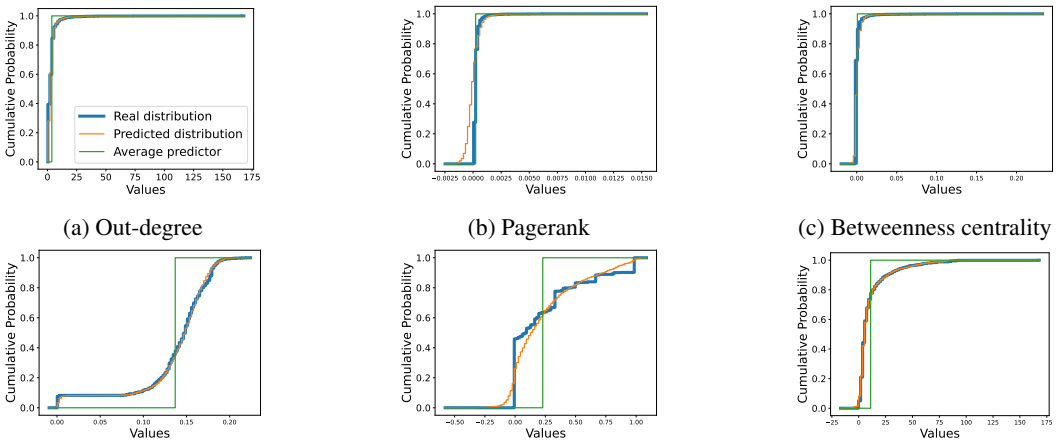
Starting from a graph dataset, we compute node-level structural features that serve as target values to predict. Specifically, we choose some well-known and studied metrics in network science (Barabási & Pósfai, 2016), which are in and out-degree, the average neighbor degree, PageRank as representative of spectral-based centrality measures, betweenness, and closeness centrality - based on shortest path distances, and the clustering coefficient, i.e. how close the neighbors of a node are to being a clique. We model the problem of predicting the distribution of a structural feature as a

Table 1: RMSE for regression on node-level features on Cora, CiteSeer, and PubMed datasets. We report the average and standard deviation results over experiments with three different random seeds.

Metric	Cora	CiteSeer	PubMed
In-degree	0.89 ± 0.20	0.69 ± 0.16	1.52 ± 0.13
Out-degree	1.01 ± 0.33	0.68 ± 0.45	1.46 ± 0.37
Pagerank	0.06 ± 0.02	0.08 ± 0.01	0.07 ± 0.01
Betweenness centrality	0.06 ± 0.01	0.05 ± 0.02	0.06 ± 0.02
Closeness centrality	0.04 ± 0.01	0.04 ± 0.01	0.05 ± 0.01
Clustering coeff.	0.24 ± 0.02	0.22 ± 0.01	0.17 ± 0.01
Avg. neigh. degree	6.72 ± 1.3	5.45 ± 0.43	10.74 ± 2.30

node regression task. We initialize the node embeddings with random features. We made this choice as recently Sato et al. (2021) shows that random features strengthen the expressive power of GNNs and allow the reconstruction of ancillary features - not related to the graph structure - for nodes (Sato, 2023). Hence, using random features allows us to better leverage and test the learning power of GNNs. Then, we leverage GNN models to solve the node regression task. We limit our analysis to 1-hop message passing GNNs (Gilmer et al., 2017) as they represent the most well-known, used, and fundamental architecture for deep learning on graphs. The methodology can be easily used to solve edge and graph regression tasks, i.e. for learning edge- and graph-level structural features such as the Adamic Adar Index (Adamic & Adar, 2003) or the network density.

3 Results



(a) Out-degree (b) Pagerank (c) Betweenness centrality

(d) Closeness centrality (e) Clustering coefficient (f) Avg. neighbor degree

Figure 1: Cumulative Distribution Function (CDF) of real (blue line) and predicted (orange line) node-level structural features distributions for Cora dataset.

Table 1 shows the root mean square error (RMSE) for regression on the different node-level structural features for the three considered datasets.[1]Figure 1 shows the cumulative distribution function of real and predicted distributions for the Cora dataset, which is representative of all three datasets. Results highlight that, intuitively, GNNs can easily reconstruct in/out-degree and PageRank by leveraging the message passing mechanism. Less expected is that GNNs reconstruct features based on shortest-path distances, which are considered hard to learn (You et al., 2019). Moreover, they reach quite good performance in learning the local clustering coefficient, which requires counting triangles, even if cycles and high-order structures are hard to learn by GNNs (You et al., 2021). Lastly, it seems that GNNs struggle to compute the average neighbor degree as its prediction can deviate up to 10 neighbors from the correct values. In short, these preliminary results on the GNN capability of learning structural features offer a few insights into which structural features are easily learnable and open up the possibility of using pre-trained GNNs to efficiently approximate structural features that are computationally expensive, as highlighted in Table 2 and 3.

[1]Code, hyperparameter-tuning, and reproducibility information are available here `https://github.com/manuel-dileo/gnn-learns-graphs`. See the Appendix for further information and future works.

URM STATEMENT

The authors acknowledge that at least one key author of this work meets the URM criteria of ICLR 2024 Tiny Papers. Author one meets the URM criteria of ICLR 2024 Tiny Papers Track.

ACKNOWLEDGE

This work was supported in part by project SERICS (PE00000014) under the NRRP MUR program funded by the EU - NGEU. Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the Italian MUR. Neither the European Union nor the Italian MUR can be held responsible for them.

REFERENCES

Lada A Adamic and Eytan Adar. Friends and neighbors on the web. *Social networks*, 25(3):211–230, 2003.

A Arciprete, V Carchiolo, D Chiavetta, M Grassia, M Malgeri, and G Mangioni. Geometric deep learning graph pruning to speed-up the run-time of maximum clique enumerarion algorithms. In *International Conference on Complex Networks and Their Applications*, pp. 415–425. Springer, 2022.

Albert-László Barabási and Márton Pósfai. *Network science*. Cambridge University Press, Cambridge, 2016. ISBN 9781107076266 1107076269. URL http://barabasi.com/networksciencebook/.

Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? In *ICLR*. OpenReview.net, 2022.

Manuel Dileo, Matteo Zignani, and Sabrina Gaito. Temporal graph learning for dynamic link prediction with text in online social networks. *Machine Learning*, 113(4):2207–2226, Apr 2024. ISSN 1573-0565. doi: 10.1007/s10994-023-06475-x. URL https://doi.org/10.1007/s10994-023-06475-x.

Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR 2019 Workshop on Representation Learning on Graphs and Manifolds*, 2019. URL https://arxiv.org/abs/1903.02428.

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In *ICML*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1263–1272. PMLR, 2017.

Francesco Di Giovanni, Lorenzo Giusti, Federico Barbero, Giulia Luise, Pietro Lio, and Michael M. Bronstein. On over-squashing in message passing neural networks: The impact of width, depth, and topology. In *ICML*, volume 202 of *Proceedings of Machine Learning Research*, pp. 7865–7885. PMLR, 2023.

Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, YongDong Zhang, and Meng 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*, SIGIR '20, pp. 639–648, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450380164. doi: 10.1145/3397271.3401063. URL https://doi.org/10.1145/3397271.3401063.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR (Poster)*. OpenReview.net, 2017.

Sofiane Lagraa, Martin Husák, Hamida Seba, Satyanarayana Vuppala, Radu State, and Moussa Ouedraogo. A review on graph-based approaches for network security monitoring and botnet detection. *International Journal of Information Security*, 23(1):119–140, 2024.

Qingyun Liu, Shiliang Tang, Xinyi Zhang, Xiaohan Zhao, Ben Y. Zhao, and Haitao Zheng. Network growth and link prediction through an empirical lens. In *Proceedings of the 2016 Internet Measurement Conference*, IMC '16, pp. 1–15, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450345262. doi: 10.1145/2987443.2987452. URL https://doi.org/10.1145/2987443.2987452.

Oleg Platonov, Denis Kuznedelev, Michael Diskin, Artem Babenko, and Liudmila Prokhorenkova. A critical look at the evaluation of GNNs under heterophily: Are we really making progress? In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=tJbbQfw-5wv.

Ryoma Sato. Graph neural networks can recover the hidden features solely from the graph structure. In *ICML*, volume 202 of *Proceedings of Machine Learning Research*, pp. 30062–30079. PMLR, 2023.

Ryoma Sato, Makoto Yamada, and Hisashi Kashima. Random features strengthen graph neural networks. In *SDM*, pp. 333–341. SIAM, 2021.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=ryGs6iA5Km.

Zhilin Yang, William W. Cohen, and Ruslan Salakhutdinov. Revisiting semi-supervised learning with graph embeddings. In *ICML*, volume 48 of *JMLR Workshop and Conference Proceedings*, pp. 40–48. JMLR.org, 2016.

Jiaxuan You, Rex Ying, and Jure Leskovec. Position-aware graph neural networks. In *ICML*, volume 97 of *Proceedings of Machine Learning Research*, pp. 7134–7143. PMLR, 2019.

Jiaxuan You, Zhitao Ying, and Jure Leskovec. Design space for graph neural networks. In *NeurIPS*, 2020.

Jiaxuan You, Jonathan Michael Gomes Selman, Rex Ying, and Jure Leskovec. Identity-aware graph neural networks. In *AAAI*, pp. 10737–10745. AAAI Press, 2021.

Ziqi Yuan, Haoyi Zhou, Tianyu Chen, and Jianxin Li. Phogad: Graph-based anomaly behavior detection with persistent homology optimization. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, WSDM '24, pp. 920–929, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400703713. doi: 10.1145/3616855.3635783.

Marinka Zitnik, Monica Agrawal, and Jure Leskovec. Modeling polypharmacy side effects with graph convolutional networks. *Bioinform.*, 34(13):i457–i466, 2018.

A Appendix

Future works. We believe our work can serve as a first basis for studying which structural features GNNs can effectively learn from graph data. In future works, the same experiments can be performed on other benchmark datasets in different real-world scenarios, ranging from several applications where GNNs are succesfully leveraged, such as online social networks (Dileo et al., 2024), biological networks (Zitnik et al., 2018), or recommender systems (He et al., 2020), up to specific datasets that exhibit structural patterns for which GNNs typically struggle, like bottlenecks (Giovanni et al., 2023) or heterophilic networks (Platonov et al., 2023). Moreover, the same methodology can be adopted to study edge- and graph-level structural features, as well as temporal or heterogeneous network properties. Lastly, this work serves as a potential basis to understand if GNNs, thanks to their inherent inductive power, can be used to approximate some computationally expensive measure on large graphs, like betweenness or closeness centralities, by learning them on a bunch of small graphs (Arciprete et al., 2022), benefiting fields such as information retrieval and web search ranking methods.

Computational, training, and inference time of shortest-path distances. Betweenness and closeness centralities require to compute the distribution of shortest-path distances between nodes. Hence, they are computationally expensive, especially on large graphs. We report in Table 2 and 3 the execution time for the computation, training, and inference of betweenness and closeness centralities for each dataset. Results show that on PubMed, the largest graph among the three, the execution time required for the GNN to compute the metrics is drastically lower than that needed for the ground-truth computation. This result incentivizes the possibility of using pre-trained GNNs to approximate structural features that are computationally expensive.

Table 2: Computational, training, and inference time for betweenness centrality on Cora, CiteSeer, and PubMed datasets. For a fair comparison, the computational time is measured for nodes in the test set only (20% of nodes). GNN Computation is the sum of training and inference times.

Dataset	Computation	Training	Inference	GNN Computation
Cora	3.36s	2.13s	27.2ms	2,16s
CiteSeer	3.38s	807ms	12.7ms	819.7ms
PubMed	6.21min	6.17s	39ms	6,21s

Table 3: Computational, training, and inference time for closeness centrality on Cora, CiteSeer, and PubMed datasets. For a fair comparison, the computational time is measured for nodes in the test set only (20% of nodes). GNN Computation is the sum of training and inference times.

Dataset	Computation	Training	Inference	GNN Computation
Cora	0.95s	2.65s	24.1ms	2,68s
CiteSeer	0.76s	808ms	5.19ms	0.81s
PubMed	59s	990ms	5.63ms	995.63ms

Training and hyperparameter details. We developed our pipeline using Pytorch Geometric (PyG) (Fey & Lenssen, 2019). We tested the three most used and well-known GNN layers, which are GCN (Kipf & Welling, 2017), GAT (version 2) (Brody et al., 2022), and GIN (Xu et al., 2019), which has the same expressive power as the 1-WL test for graph isomorphism. We ran our experiments on NVIDIA Corporation GP107GL [Quadro P400]. In all our experiments, we use the Adam (Kingma & Ba, 2015) optimizer. We split the datasets into training, validation, and test sets. Consistently, we apply identical dataset divisions and training procedures across all the experiments. Hyperparameters are tuned by optimizing the RMSE on the validation set, and the model parameters are randomly initialized. The hyperparameter search spaces are as follows: learning rate {0.1, 0.01, 0.001}, L2 weight-decay {5e-1, 5e-2, 5e-3}, number of hidden layers {1, 2, 3}, GNN layer {GCN, GAT, GIN}, hidden dimension {32, 64, 128, 256}, input size (number of random features) {64, 128, 256, 512}. Overall, the best results are achieved using GCNs with 2 or 3 layers.