Kalpa Publications in Computing
Volume 19, 2024, Pages 27–40
Proceedings of 6th International Conference on Smart Systems and Inventive Technology

Analysis of METIS Graph Partitioning Algorithms for Trust and Recommendation Systems

Bayana Mohith Siva Sai[1], Liz Maria Liyons[2], and Geetha M[3]

Department of Computer Science and Engineering, Amrita School of Computing, Amrita Vishwa Vidyapeetham,
Amritapuri, India.
mohithbayana@gmail.com[1] lizmarialiyons@gmail.com[2] geetham@am.amrita.edu[3]

Abstract.
In the era where social media and technology intersect, the vast user base of social networks presents a challenge in handling massive data. The issue intensifies when making user suggestions amidst the overwhelming data flow. This analysis addresses the complexities arising from the abundance of data in social networks and proposes a solution through advanced graph partitioning techniques, focusing on algorithms from prominent libraries like DGL and PyTorch. This analysis compares three graph partitioning algorithms for social network analysis: DGL METIS (edge-balanced and node-balanced), and PyG METIS. We analyze their performance on the Epinions social recommendation dataset, focusing on edge based and node based metrics and visualization of partitions. Our findings reveal: PYG METIS consistently exhibited suboptimal performance across various evaluation metrics, with the exception of achieving satisfactory results in node balance. Conversely, DGL Node Balanced METIS demonstrated marginally superior outcomes compared to DGL Edge Balanced METIS in terms of edge loss and average edges per partition and surpassed it in node balance.

Keywords: Graph partitioning · trust and recommendation system · METIS · gnn.

1 Introduction

In the world where social media and technology meet, there's a huge crowd using social net-works. This surge in users has created a big problem: dealing with massive amounts of data in these networks. Especially when it comes to suggesting things to users, understanding how they interact becomes really tough in this flood of information.

With so many people sharing information online, we've ended up with massive piles of data. Figuring out what's important and making useful suggestions to users has become a big puzzle. The sheer amount of data makes it hard to train the models. Training a model with large datasets takes a lot of time and the computational power required to do so is also very high.

To solve this issue, we need smarter ways to handle all this data. One promising idea is to split these huge networks into smaller, more manageable parts - Graph Partitioning. But there is a problem dealing with this technique. Graph partitioning algorithms face the formidable challenge of disentangling networks laden with social links, aiming to break down expansive networks into smaller segments while safeguarding the intricate connections that unveil individuals' interlinked nature. Preserving these social links is pivotal, holding essential insights into user connections and mutual influences. The primary goal is to divide these networks into more manageable components without losing these vital connections, ensuring a deeper understanding of relationships and preferences. Metis offers an advanced graph partitioning algorithm adept at conserving these critical social links by meticulously dissecting networks into smaller sections while retaining the intricate interweaving of user connections.

This research dives deep into the different partitioning techniques in metis - PYG- Metis, DGL metis (edge balanced partition and node balanced partition) and analyzes how each method partitions the graph over various metrics[1-4]. Based on our experimental findings, it is deduced that DGL-Node Balanced METIS exhibits superior overall performance in handling the Trust and Recommendation Epinions [5] dataset.

2 Related work

Within the field of Graph Neural Networks (GNN), extensive research, particularly on large-scale graphs, has been conducted. Addressing the social recommendation domain, [3] proposed GraphRec, an innovative graph network model for rating prediction on the Epinions dataset. Notably, [6] contributes significantly to the field by introducing a novel approach to graph partitioning, leveraging semi-supervised graph neural networks. Additionally, [7] presents a spe-cialized graph neural network tailored for influencer recommendation, emphasizing the intricate relationships between firms and influencers on YouTube[8-11]. [12-14] presents algorithmic improvements to the multithreaded graph partitioner mt-Metis. The improvements decrease the runtime by 1.5-11.7X and improve strong scaling by 82%.

Given the computational challenges posed by large-scale real-world graph datasets, graph partitioning emerges as a pivotal solution, motivating the in-depth analysis in this paper. The research focuses on METIS, an advanced graph partitioning method [2], renowned for its ability to generate high-quality partitions with minimal edge cuts. METIS's efficiency and quality position it as the preferred choice for various graph partitioning applications.

3 Methodology

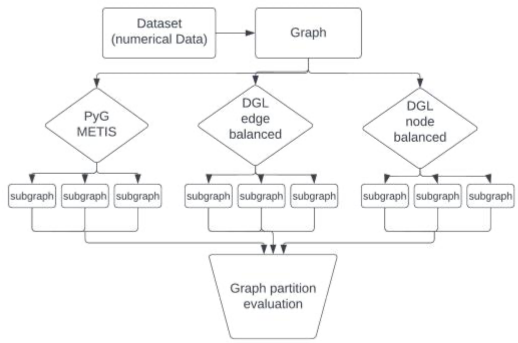

Fig. 1. Graph Partitioning Algorithm

In this section, we present our methodology (Fig 1) to analyze the graph partitioning algorithms.

3.1 Datasets

We used a social recommendation dataset: The Epinions [5] dataset stands as a cornerstone in the evaluation of trust and recommendation systems. Comprising online ratings and trust relationships between users for various product categories. The dataset has 6 columns (6 features) namely: i) User ID ii) Product ID iii) category iv) rating v) helpfulness of the rating vi) Timestamp of the rating. Each row of the dataset provides the rating given by a particular user to a particular product of a particular category, when the rating is given, and how helpful the rating is. The dataset also contains a trust network which provides information about the trust between different users.

3.2 Graph creation

The Epinions dataset, comprising numeric data in a text file, necessitates the initial construction of a graph representation. In this structural framework, users and products serve as nodes, and edges are established under specific criteria. Precisely, an edge materializes between a user and a product if the user has provided a rating for the respective product. Additionally, edges are formed between users to denote mutual trust, reflecting instances where one user places trust in another within the confines of the dataset. This structured graph representation serves as a foundational step in the analytical process of the Epinions dataset within the context of our research for the conference paper.

3.3 Graph partitioning

Subsequent to the construction of the graph from the Epinions dataset, the graph is employed as input for three distinct graph partitioning algorithms. The algorithms generate partitioned subgraphs as output, which are subsequently subjected to comprehensive evaluation using di-verse graph partitioning evaluation metrics. This systematic approach ensures a rigorous and comparative assessment of the efficacy and performance of the employed graph partitioning techniques within the context of our research presented in this conference paper.

3.4 Evaluation Metrics

We used four metrics for evaluating the quality of graph partitions. 1) Edge loss: Edge loss is used to measure the amount of edges that were lost during the process of graph partitioning. 2) Node loss: Node loss is used to measure the amount of nodes that were lost while performing graph partitioning. 3) Edge balance: Edge balance is used to measure the distribution of edges across the different partitions. 4) Node balance: Node balance is used to measure the distribution of nodes across the different partitions. Edge balance and node balance is best when it takes the value 1 and the quality decreases with the increase in value of balance.

4 Experimental Results

PyG METIS: The visual representation of the three partitions Fig 2 reveals a notable sparsity. Evidently, a substantial number of nodes within this subgraph lack connecting edges, suggesting a significant imbalance in edge distribution within the partitioned structure.

DGL METIS The observed partitions (Fig 3) exhibit a higher degree of density compared to the partitions generated by PyG METIS (Fig 2). The subgraph comprising 100 partitions, in particular, manifests a greater concentration of nodes and edges in contrast to the subgraphs with 200 and 300 partitions, indicating a discernible disparity in density across the sampled partitions.

From Fig 4 we can say that the sub-graphs associated with less number of partitions are very dense and large while the graphs with a large number of partitions are comparatively sparse.

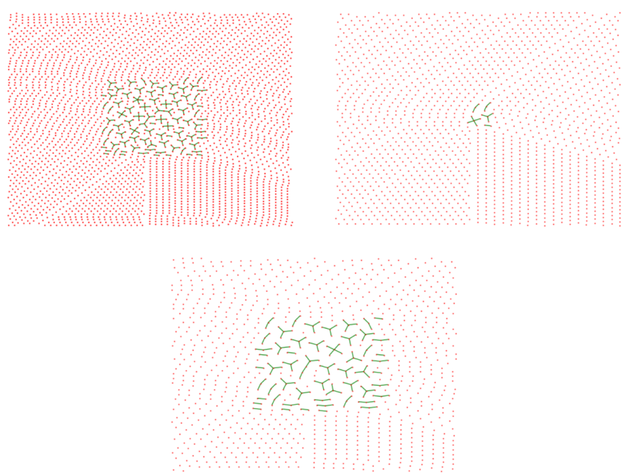

Fig. 2. visualization of sample subgraphs of PyG METIS when the number of partitions is set to a)100 b)200 c)300

Metric	DGL-Edge-Balanced 50	100	200	300	DGL-Node-Balanced 50	100	200	300	PyG 50	100	200	300
Total nodes	318441	318441	318441	318441	318441	318441	318441	318441	318438	318438	318438	318438
Avg. nodes	6368.8	3184.4	1592.20	1061.47	6368.8	3184.4	1592.20	1061.47	6368.7	3184.3	1592.1	1065.01
Min. nodes	2904	1317	472	242	6225	3105	1567	1035	6183	3091	1551	1014
Max. nodes	10557	5936	3433	2505	6560	3280	1640	1093	6560	3280	1640	1093
Total edges	397534	372334	354269	349319	429059	397244	371221	359089	51770	318438	30534	28240
Avg edges	7950.6	3723.3	1771.34	1164.396	8581.1	3972.4	1856.10	1196.963	1035.4	386.37	152.67	94.44816
Min. edges	4599	2024	810	463	6327	2466	400	61	16	1	0	0
Max. edges	11540	6033	3456	2497	18490	10205	5924	4410	5284	1822	601	382

Table 1. Edge and Node Statistics for DGL and PyG Algorithms

4.1 Evaluating the Partitions

Table 1 provides a detailed summary of node and edge data, illustrating differences that can be observed between different partitions for all three partitioning techniques. One interesting finding is that all partitions have the same number of nodes, however each partition has a different number of edges.

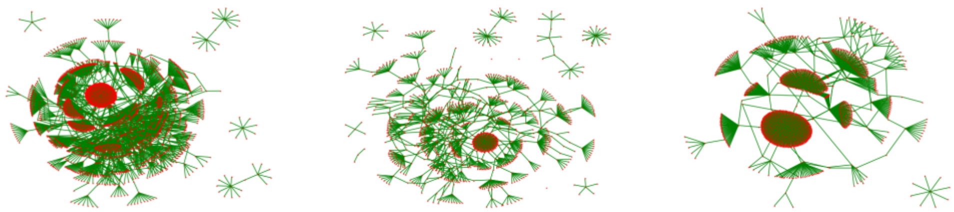

Fig. 3. visualization of sample subgraphs of DGL-edge-balanced METIS when the number of partitions is set to a)100 b)200 c)300

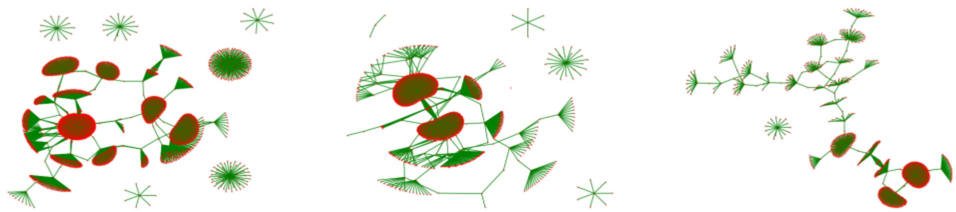

Fig. 4. visualization of sample subgraphs of DGL-node-balanced METIS when the number of partitions is set to a)100 b)200 c)300

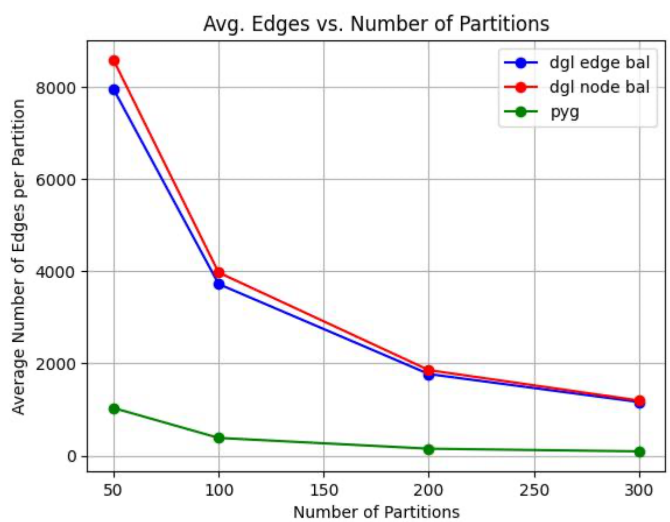

Fig. 5. Graph plotted between the average number of edges against the number of partitions for all the three partitioning techniques

Fig 5 depicts the graph that illustrates the average number of edges across a spectrum of partition sizes. The declining trend in average edges becomes apparent with an increase in the number of partitions. Notably, the plots for DGL METIS edge-balanced and node-balanced algorithms exhibit similarity, whereas the PyG METIS plot displays distinctive characteristics. Commencing with approximately 8000 edges, both DGL algorithms contrast with PyG METIS, which initiates at around 1000 edges and experiences a gradual reduction. A huge number of edge loss in PyG METIS may account for the comparatively lower numerical values. In the context of DGL, the node-balanced METIS variant exhibits slightly higher average edge counts compared to edge-balanced METIS

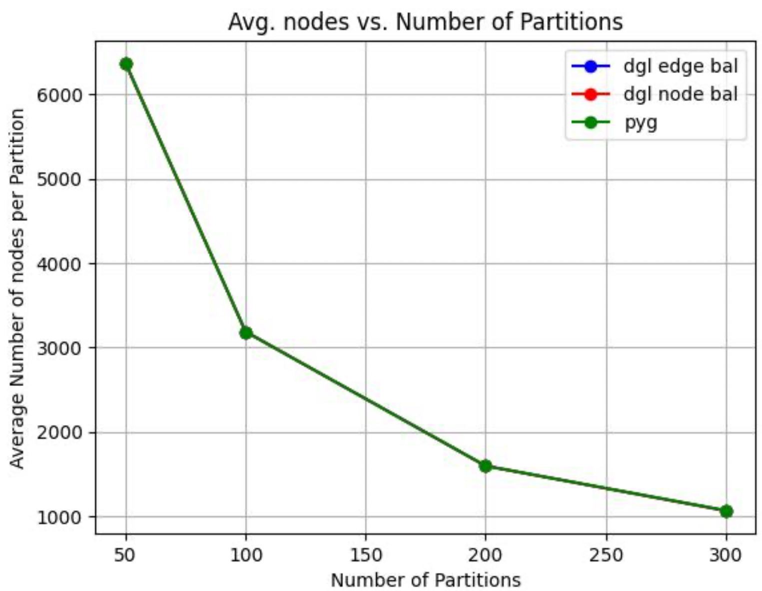

Fig. 6. Graph plotted between average number of nodes against number of partitions for all the three partitioning techniques

Fig. 6 shows a plot that depicts the relationship between average nodes and the number of partitions. A discernible trend emerges as average nodes decrease with an increase in the number of edges. A noteworthy observation from this graph is the overlapping of all three plots, corresponding to their respective algorithms, making it seem like a single line. This overlapping is indicative of nearly identical average node values across the three algorithms.

Fig 7 illustrates the variation in the number of edges per partition for each algorithm, with the number of partitions set to 300. The DGL node-balanced METIS graph shows a rather consistent amount of partitions for the major part, however, there are occasional occurrences of significant increases and decreases in the number of partitions. In comparison, the DGL edge-balanced METIS graphic shows a constant change in the number of partitions but stays within a specified range, with no significant spikes or declines. In contrast, the PyG METIS plot demonstrates exceptional stability throughout the partitions, but with a constantly low number of edges, as evidenced by its location at the bottom of the picture. This finding is due to a considerable loss of edges during the partitioning procedure.

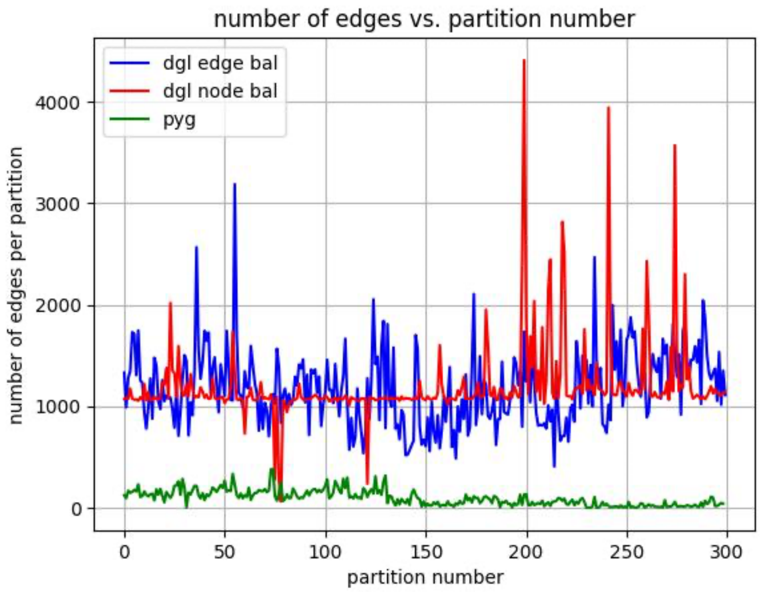

Fig. 7. Graph plotted for the distribution of edges over each partition when the number of partitions is set to 300 for all the three partitioning techniques.

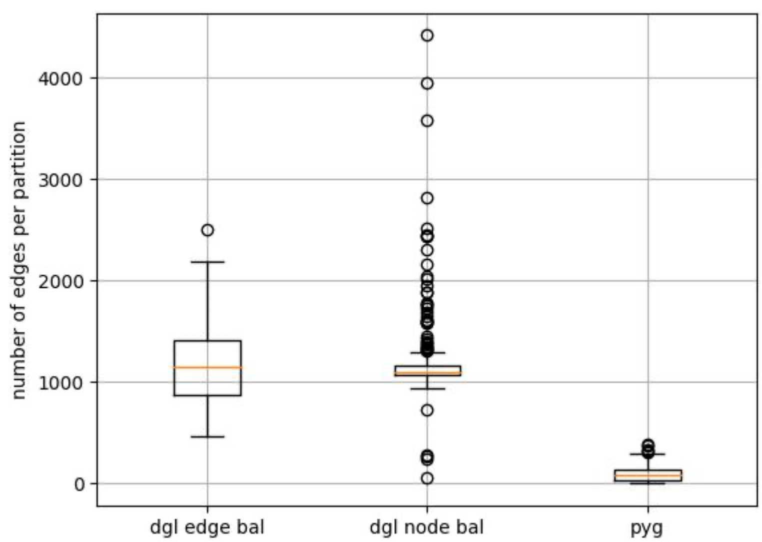

Fig. 8. Box plot of number of edges across the partition

Fig 8 comprises a box plot illustrating the distribution of edges across partitions as shown in Fig 7. This plot gives more insights into the distribution. Notably, the interquartile ranges for DGL node-balanced and PyG are comparatively narrow, signifying a more confined spread of edge distribution. In contrast, the interquartile range for DGL edge-balanced is broader, indicating a more dispersed distribution of edges, wherein a greater number of partitions exhibit varying edge counts. Furthermore, the box plot reveals the presence of numerous outliers in the DGL node-balanced distribution, underscoring substantial variations in the number of edges within certain partitions compared to others. Conversely, the PyG distribution exhibits both a notably low interquartile range and a diminished number of outliers, indicating a more uniform distribution of edges across partitions with less pronounced variations.

A salient observation lies in the consistent absence of node loss for both DGL node-balanced METIS and DGL edge-balanced METIS, denoting a node loss value of zero across varying parti-tion counts. Likewise, PyG METIS exhibits a near-zero node loss, maintaining a constant value (9.42089743468963e-06) independent of the number of partitions. This noteworthy consistency underscores the robustness of the algorithms in preserving nodes during the partitioning process.

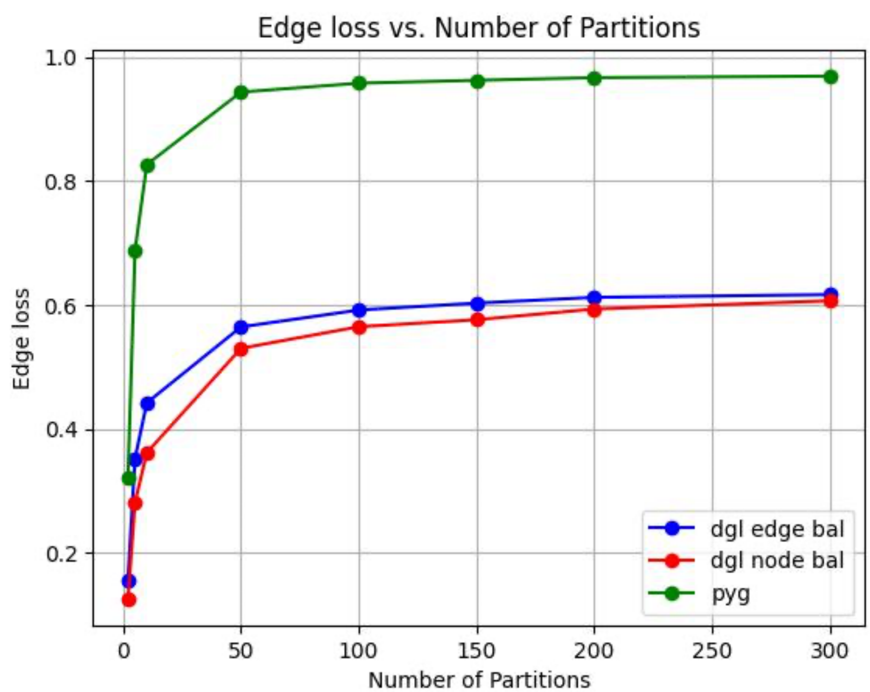

Fig. 9. Graph plotted between Edge loss against number of partitions for all the three partitioning techniques

The graph (Fig 9) demonstrates how edge loss varies with the number of partitions. A clear observation is that edge loss increases with the number of partitions. The variation in edge loss is large for the initial smaller values of partitions, but as the number of partitions increases, the variation in edge loss reduces, as seen by the extremely little difference in edge loss be-tween 200 and 300 partitions. While the overall shape of the graph remains consistent across the three algorithms, a notable distinction is the comparatively higher values of edge loss for PyG in contrast to the DGL algorithms. Specifically, the PyG METIS algorithm registers an edge loss of 0.96905 (96.905%) for 300 partitions, in contrast, the maximum value of edge loss that DGL algorithms achieved is 0.61646(61.646%) and 0.60645(60.645%). Within the DGL algorithms, the edge-balanced METIS variant exhibits slightly elevated values compared to its node-balanced counterpart.

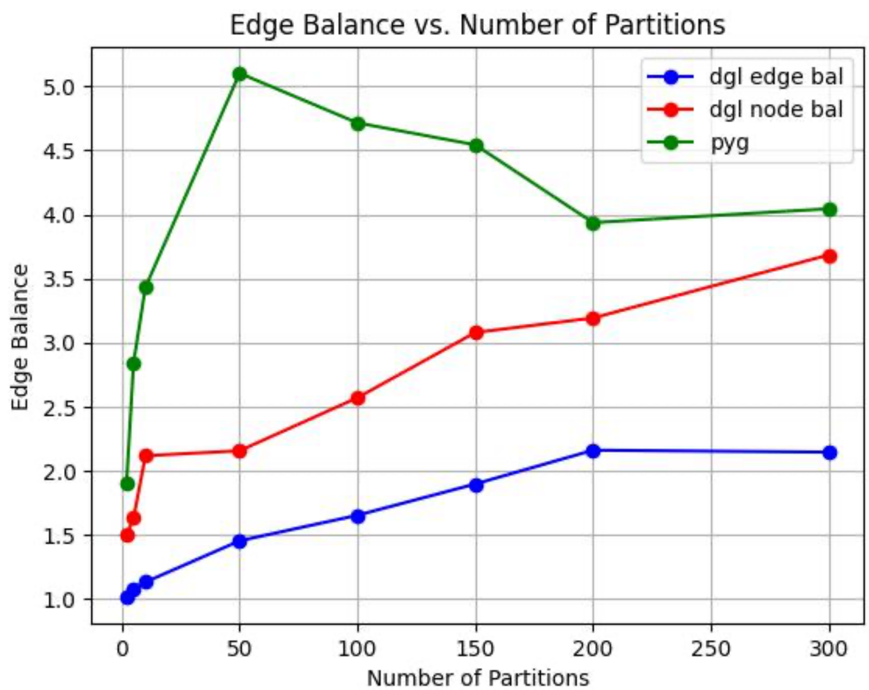

Fig. 10. Graph plotted between Edge balance against number of partitions for all the three partitioning techniques

Fig 10 shows how the edge balance varies with the change in the number of partitions. DGL edge-balanced metis as the name implies exhibits the best edge balance. The balance is nearly equal to 1 when the partitions are set to 2 and the peak value of the balance is just 2.159230541 when the partitions are set to 200. In contrast, pyg metis has the worst edge balancing of the three algorithms with a minimum at 1.89854 when the number of partitions is 2 and a maxi-mum at 5.103341704 when the partitions are set to 50 followed by a considerable decline. DGL node-balanced METIS falls between these two techniques, with values ranging from 1.49855 to 3.68432.

The presented figure (Fig 11) delineates the fluctuation in node balance concerning the number of partitions. Notably, DGL Edge Balanced METIS exhibits suboptimal node balancing com-pared to both DGL Node Balanced METIS and PyG METIS. The node balance for DGL Edge

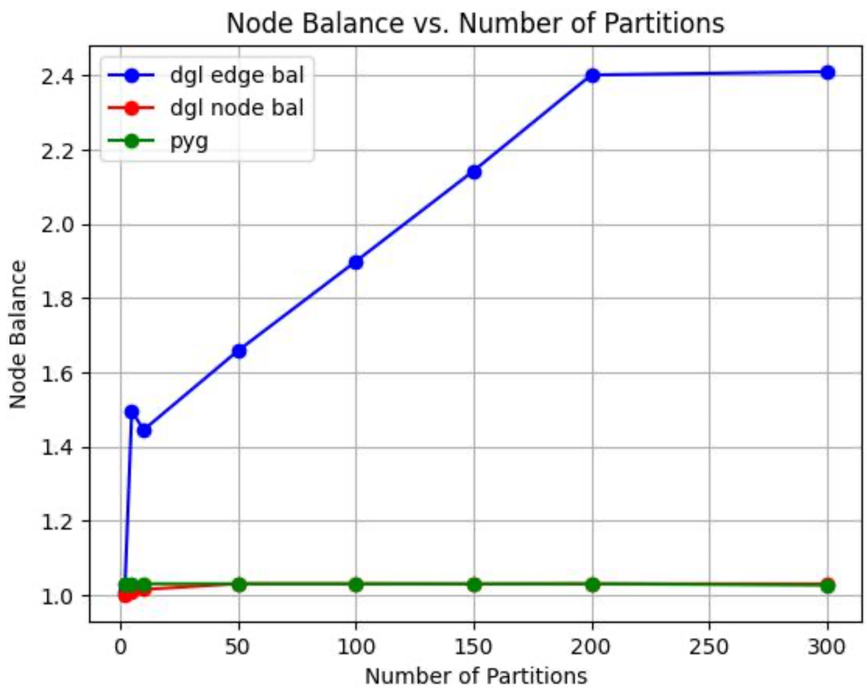

Fig. 11. Graph plotted between Node balance against number of partitions for all the three partitioning techniques

Balanced METIS demonstrates a linear increase in the range of 10 to 200. The node balance varies from 1.0073 to 2.40986 across the partitions, surpassing the maximum values of the other two algorithms, which peak at 1.0297. Given the substantial values associated with DGL Edge Balanced METIS, the scale of the graph necessitates a focused examination. Consequently, a subsequent graph is generated to exclusively display the node balance trends of DGL Node Balanced METIS and PyG METIS, allowing for a more discernible visualization of their dis-tinctions.

In Fig. 12, the node balance graph is presented, deliberately excluding DGL Edge Balanced METIS to prevent scale distortion. The observation reveals that DGL Node Balanced METIS exhibits notably low values for node balance, particularly for smaller partition counts. Specifically, when the number of partitions is set to 2, the node balance approaches 1.000003, indicative of close alignment with the ideal node balance of 1. Subsequently, across the partition range from 50 to 200, both PyG METIS and DGL Node Balanced METIS consistently maintain similar node balance values. However, with 300 partitions, DGL Node Balanced METIS records a slightly reduced node balance compared to PyG METIS.

4.2 Discussion

The comprehensive experiments and analyses presented in this section provide valuable insights into the performance and characteristics of three graph partitioning algorithms which include DGL METIS edge-

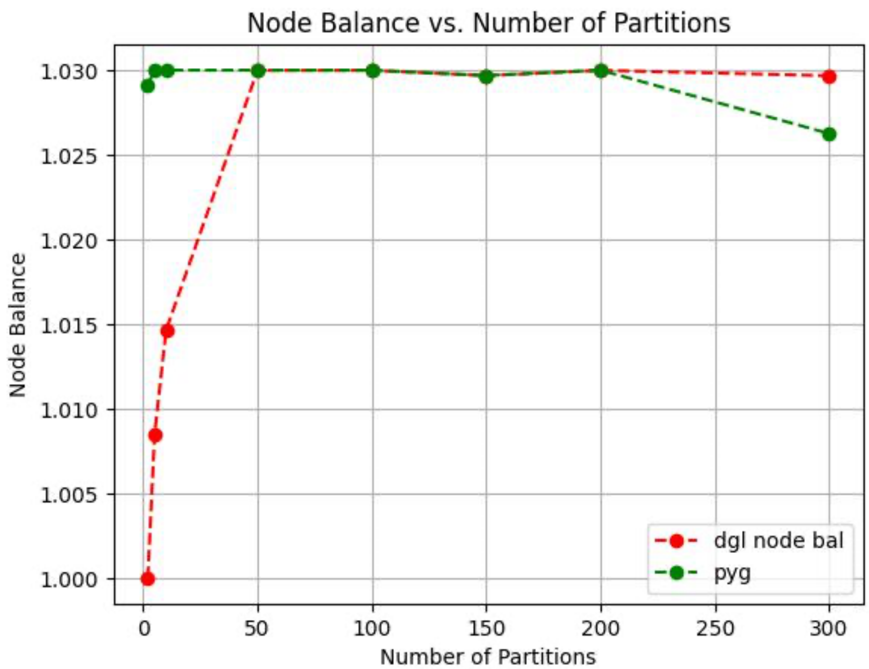

Fig. 12. Graph plotted for Node Balance excluding DGL edge balanced METIS

balanced, DGL METIS node-balanced, and PyG METIS. The visualizations of the partitions revealed PyG METIS showcasing sparser distribution. The detailed statis-tics presented in Tables 1 offer a comprehensive overview of the node and edge data across different partitions for each algorithm. A pivotal observation emerges from the analysis, high-lighting uniformity in the average number of nodes across all techniques for a specific partition count. Additionally, the node loss metric demonstrates exceptional resilience, registering zero for DGL METIS and nearly negligible values for PyG METIS, underscoring the robustness of these methodologies in preserving nodes. Notwithstanding a slight discrepancy in node balance, specifically with DGL Edge Balanced METIS exhibiting suboptimal performance, it is note-worthy that all three algorithms exhibit commendable proficiency in node preservation aspects. The analysis of edge loss highlighted a consistent increase in the number of partitions with PyG metis delivered a significantly bad performance compared to both DGL algorithms. Addition-ally, the examination of edge balance revealed that DGL edge-balanced METIS consistently outperformed its counterparts, showcasing superior balance values. Conversely, it exhibited a linear increase in node balance with partition count, surpassing the maximum values of the other two algorithms while DGL node-balanced METIS showcased better values. In summary, PYG METIS consistently exhibited suboptimal performance across various evaluation metrics, with the exception of achieving satisfactory results in node balance. Conversely, DGL Node Balanced METIS demonstrated marginally superior outcomes compared to DGL Edge Balanced METIS in terms of edge loss and average edges per partition and surpassed it in node balance. Notably, DGL Edge Balanced METIS outperformed its counterpart in edge balance and exhibited a more favorable distribution of edges across partitions, as visually depicted in the box plot presented in Fig. 8.

5 Conclusion

This research study has investigated the performance of three graph partitioning algorithms - DGL METIS (edge-balanced and node-balanced) and PyG METIS - for enhancing trust and recommendation systems. We analyzed their effectiveness on the Epinions social recommendation dataset using various evaluation metrics, including edge and node loss, edge and node balance, and visualization of partitions. Our Key findings are all algorithms exhibited excellent node preservation with nearly zero node loss, PYG METIS displayed significant edge loss compared to DGL algorithms, DGL edge-balanced METIS outperformed others in edge balance but had suboptimal node balance, DGL node-balanced METIS and PYG METIS demonstrated similar and accept-able node balance. In conclusion, the choice of the optimal graph partitioning algorithm depends on the specific priorities of the application. DGL node-balanced METIS is a well-rounded option for balancing edge loss, average edges per partition, and node balance. DGL edge-balanced METIS is preferable when edge balance and edge distribution are crucial. PYG METIS, while demonstrating satisfactory node balance, is less competitive in other metrics. Furthermore, future research could delve into the exploration of data parallelism through graph partitioning on the presented dataset, leveraging the analytical insights provided in this paper.

References

Sani, Mustapha Ahmad, Abdulmalik Mohammed, Ayaz Ahmad, Abdulkadir Haruna, Yusuf Ayaz, Khalid Mohammed, Abdulkadir Ahmad, Yusuf Overview,. (2022). An Overview of Metrics for Evaluating the Quality of Graph Partitioners. Mathematics and Computer Science. 7. 1-8. 10.11648/j.mcs.20220701.11.

Karypis, George Kumar, Vipin. (1995). METIS – Unstructured Graph Partitioning and Sparse Matrix Ordering System, Version 2.0.

Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph Neural Networks for Social Recommendation. In The World Wide Web Conference (WWW '19). Association for Computing Machinery, New York, NY, USA, 417–426. https://doi.org/10.1145/3308558.3313488

Shuangyan Yang, Minjia Zhang, Wenqian Dong, and Dong Li. 2023. Betty: Enabling Large-Scale GNN Training with Batch-Level Graph Partitioning. In Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS 2023). Association for Computing Machinery, New York, NY, USA, 103–117. https://doi.org/10.1145/3575693.3575725http://www.cse.msu.edu/tangjili/index.html

Zhengxi Yang, Ruiyang Shi, Pei Quan, Ruizhi Zhou, Lingfeng Niu, Semi-Supervised Graph Neural Networks for Graph Partitioning Problem, Procedia Computer Science, Volume 221,2023,Pages 789-796,ISSN 1877-0509,https://doi.org/10.1016/j.procs.2023.08.052.

Jinhee Park, Hyeongjin Ahn, Dongjae Kim, Eunil Park, GNN-IR: Examining graph neural networks for influencer recommendations in social media marketing, Journal of Retailing and Consumer Services, Volume 78,2024,103705,ISSN 0969-6989,https://doi.org/10.1016/j.jretconser.2024.103705.

A. B. Nair, G. Surendran, K. P. Prathyun, V. Sivaprasad and C. P. Prathibhamol, ”Comparative study of Centrality based Adversarial Attacks on Graph Convolutional Network model for Node classification,” 2022 7th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2022, pp. 731-736, doi: 10.1109/ICCES54183.2022.9835948.

M. Lolla, A. Barik and B. K R, ”GNN Based Trainable Dependency Parser,” 2023 International Conference on Recent Advances in Information Technology for Sustainable Development (ICRAIS), Manipal, India, 2023, pp. 194-200, doi: 10.1109/ICRAIS59684.2023.10367102.

S. M. Devi Thumati, K. Dhanya, H. Sathish, K. C. Sekhar Madan and S. Rani, ”A Comparative Study on the Working of GNN and CNN on Panoramic X-Rays in Prediction of Dental Diseases,” 2023 8th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2023, pp. 755-762, doi: 10.1109/ICCES57224.2023.10192836.

Gutta Akshitha sai, Komma Naga Sai Likhitha, Maddi Pavan Kalyan, Perisetla Anjani Devi, and Prathibhamol C. 2022. Combinational Features with centrality measurements on GCN+LR classification of Adversarial Attacks in homogenous Graphs. In Proceedings of the 2022 Fourteenth International Conference on Contemporary Computing (IC3-2022). Association for Computing Machinery, New York, NY, USA, 573–581. https://doi.org/10.1145/3549206.3549305

Dominique LaSalle, Md Mostofa Ali Patwary, Nadathur Satish, Narayanan Sundaram, Pradeep Dubey, and George Karypis. 2015. Improving graph partitioning for modern graphs and architectures. In Proceedings of the 5th Workshop on Irregular Applications: Architectures and Algorithms (IA¡sup¿3¡/sup¿ ’15). Association for Computing Machinery, New York, NY, USA, Article 14, 1–4. https://doi.org/10.1145/2833179.2833188

Anjan Chowdhury, Sriram Srinivasan, Animesh Mukherjee, Sanjukta Bhowmick, and Kuntal Ghosh. 2023. Improving Node Classification Accuracy of GNN through Input and Output Intervention. ACM Trans. Knowl. Discov. Data 18, 1, Article 17 (January 2024), 31 pages. https://doi.org/10.1145/3610535

Vijai, Anupa and Padmavathi, S and Venkataraman, D, A Review of Object Detection in Aerial Images by Deep Learning Approaches. Available at SSRN: https://ssrn.com/abstract=4705432 or http://dx.doi.org/10.2139/ssrn.4705432