Supplementary material: Structured Graph Reduction for Efficient GNN

Anonymous Author(s)
Affiliation
Address
`email`

This document contains the proof for Lemma 1, Lemma 2, Lemma 3, Lipschitz continuity, convexity, and some additional experiments on real datasets highlighting the performance of the proposed algorithms. The "`code`" folder containing the Python code for all the experiments is attached as a separate folder.

1 Proof of Lemma, Theorem, Definition and Lipschitz continuity

Lemma 1. *(Lemma 2 in the paper) By using KKT optimality condition we can obtain the optimal solution of eq.(10 in the paper) as*

$$C^{(t+1)} = \left(C^{(t)} - \frac{1}{L}\nabla f\left(C^{(t)}\right)\right)^{+} \tag{1}$$

where $(x)^{+} = \max(x, 0)$

$Proof$. The Lagrangian function of eq.(10 in the paper) is

$$L(C, \tilde{X}, \boldsymbol{\mu}_1) = \frac{1}{2}C^\top C - C^\top A - \boldsymbol{\mu}_1^\top C \tag{2}$$

where $\boldsymbol{\mu}_1$ is the dual variable. The KKT conditions of eq.(10 in the paper) is

$$C - A - \boldsymbol{\mu}_1 = 0, \tag{3}$$
$$\boldsymbol{\mu}^\top C = 0, \tag{4}$$
$$C \geq 0, \tag{5}$$
$$\boldsymbol{\mu}_1 \geq 0 \tag{6}$$

The optimal solution of C that satisfies all KKT conditions (3-6) is

$$C^{t+1} = (A)^{+} \tag{7}$$
$$= \left(C^{(t)} - \frac{1}{L}\nabla f\left(C^{(t)}\right)\right)^{+} \tag{8}$$

This concludes the proof.

Lemma 2. *The function* $f(C) = -\gamma$*log det*$(C^\top \Theta C + J) + \frac{\lambda}{2}\|C^\top\|_{1,2}^2 + \alpha h(\Theta_C)$ *is L- Lipschitz continuous.*

Proof. The functions $-\gamma$log det$(C^\top \Theta C + J)$ $\frac{\lambda}{2}\|C^\top\|_{1,2}^2$ are L_1 and L_2 Lipschitz continuous function respectively [3].

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

Consider $h(\Theta_c) = \|C^\top \Theta C\|_F^2$

$$\left| \|C_1^\top \Theta C_1\|_F^2 - \|C_2^\top \Theta C_2\|_F^2 \right| \leq \|C_1^\top \Theta C_1 - C_2^\top \Theta C_2\|_F^2 \tag{9}$$

$$\leq \|C_1^\top \Theta C_1 - C_2^\top \Theta C_1 + C_2^\top \Theta C_1 - C_2^\top \Theta C_2\|_F^2 \tag{10}$$

$$\leq \|(C_1 - C_2)^\top \Theta C_1 + C_2^\top \Theta (C_1 - C_2)\|_F^2 \tag{11}$$

$$\leq \|(C_1 - C_2)^\top \Theta C_1\|_F^2 + \|C_2^\top \Theta (C_1 - C_2)\|_F^2 \tag{12}$$

$$\leq \|C_1 - C_2\|_F^2 (\|\Theta C_1\|_F^2 + \|C_2^\top \Theta\|_F^2) \tag{13}$$

$$\leq \|C_1 - C_2\|_F^2 \|\Theta\|_F^2 \|C_1\|_F^2 + \|C_2^\top\|_F^2 \|\Theta\|_F^2) \tag{14}$$

$$\leq 2p\|\Theta\|_F^2 \|C_1 - C_2\|_F^2 = L_3 \|C_1 - C_2\|_F^2 \tag{15}$$

Eq. (9) is obtained using the reverse triangle inequality. Applying $\|AB\|_F \leq \|A\|_F \|B\|_F$ in (12) to get (13) and same applied in (13) to get (14). Finally, using $\|C_1\|_F^2 = \|C_2\|_F^2 = p$ in (14) to get (15) which concludes that $\|C^T \Theta C\|_F^2$ is L_3 Lipschitz continuous function.

Next, The function $h(\Theta_c) = \|C^\top AC\delta \mathbf{1}_{k\times 1}\|_1 + \|C^\top DC\beta \mathbf{1}_{k\times 1}\|_1$ is also Lipschitz continuous and proof is similar to proof of $\|C^\top \Theta C\|_F^2$.

Furthermore, the addition of Lipschitz continuous functions is Lipschitz continuous, so we can say that $f(C)$ is L- Lipschitz continuous function where $L = \max\{L_1.L_2, L_3\}$.

Lemma 3. *The function $f(C) = \frac{\lambda}{2}||C^\top||_{1,2}^2 + \frac{\beta}{2}||C^\top \Theta C - U\Lambda U^\top||_F^2$ defined in eq.(12 in the paper) is a convex function.*

Proof. The function $\frac{\lambda}{2}||C^\top||_{1,2}^2$ ia convex [3] and consider the function $||C^\top \Theta C - U\Lambda U^\top||_F^2$:

$$||C^\top \Theta C - U\Lambda U^\top||_F^2 = \|C^\top \Theta C\|_F^2 - 2\text{tr}(C^\top \Theta C U \Lambda U^\top) + \text{const.} \tag{16}$$

Since $\|C^\top \Theta C\|_F^2$ is a convex function and

$$\text{tr}(C^\top \Theta C U \Lambda U^\top) = \text{tr}(\Lambda^{(1/2)\top} U^\top C^\top \Theta^{(1/2)\top} \Theta^{(1/2)} C U \Lambda^{(1/2)}) = \text{tr}(M^\top M) = \|M\|_F^2 \tag{17}$$

Since $|M\|_F^2$ is a convex function with respect to M and $M = \Theta^{(1/2)\top}\Theta^{(1/2)} C U \Lambda^{(1/2)}$ is a affine transformation of C. Hence it is also convex in C. Moreover, the sum of two convex functions is convex indicating that $f(C)$ is a convex function.

Lemma 4. *The function $f(C) = -\gamma \log\det(C^\top \Theta C + J) + \frac{\lambda}{2}||C^\top||_{1,2}^2 + \frac{\beta}{2}||C\top AC - V\Psi V^\top||_F^2$ defined in eq.(18 in the paper) is a convex function.*

Proof. The functions $-\gamma \log\det(C^\top \Theta C + J)$ and $\frac{\lambda}{2}||C^\top||_{1,2}^2$ are convex [3]. The function $\frac{\beta}{2}||C^\top AC - V\Psi V^\top||_F^2$ is also convex and proof is similar to proof provided in Lemma 3. Moreover, the sum of convex functions indicates that $f(C)$ is a convex function.

Lemma 5. *By using KKT condition we can obtain the solution of convex optimization problem (15 in the paper) is*

$$\lambda_i = \frac{1}{2}(d_i + \sqrt{d_i^2 + 4/\beta}) \qquad \forall i = 1, 2, \ldots, q \tag{18}$$

Proof: The proof is similar to the proof of Lemma 16 in the paper [4].

2 Additional Experiment

We apply the proposed structured graph coarsening algorithms in the following experiments on real benchmark datasets. These results show that the proposed methods perform outstandingly on real benchmark datasets. Next, we will illustrate the generalizability of the learning structured coarsened graph from the proposed algorithms MGC and BI_GC by using different architectures to train the GNN. Specifically, we have used GNN architectures like GCN [2], APPNP [1], and GAT [6] to train our GNN and perform the node classification task. Table 1 and 2 demonstrates that the proposed methods for learning structured coarsened graph is compatible with different widely used GNN architectures, giving almost similar Node Classification accuracy across all the datasets.

Furthermore, we have also shown the efficacy of the proposed SFCG algorithm by comparing the average degree and density of the original scale-free graphs, e.g., Amazon photos and Barabasi Albert, with the coarsened graph generated using the proposed SFCG algorithm. It is evident in Table (4) that the density of the original graph and coarsened graph are the same, while as the coarsening ratio decreases, the average degree is decreasing which implies that the original scale-free structure is preserved in the coarsened graph.

Data set	**GCN**	**GAT**	**APPNP**
Cora	78.41/89.50	73.18/83.98	78.15/82.2
Citeseer	68.59/78.09	63.79/72.5	66.45/70.0
Pubmed	80.79/88.89	75.84/79.00	78.8/79.73
Co-Phy	91.84/96.22	89.36/90.5	90.36/93.1
Co-CS	88.08/93.32	84.09/92.5	86.58/91.1
DBLP	80.79/85.35	77.95/80.27	78.92/84.17

Table 1: Node classification accuracy (%) obtained using different GNN structures like GCN, GAT, and APPNP on different datasets using the proposed BI-GC algorithm for a coarsening ratio of 0.1 and using the original graph. In the table, x/y, where x represent the node classification accuracy obtained using the proposed BI-GC algorithm and y represents the accuracy obtained using the original graph. It is evident that the proposed BI-GC method is suitable for all GNN architecture.

Data set	**GCN**	**GAT**	**APPNP**
Cora	76.02/89.50	68.79/83.98	79.24/82.2
Citeseer	70.57/78.09	63.43/72.05	69.86/70.0
Pubmed	84.81/88.89	81.96/79.00	81.39/79.73
Co-Phy	94.71/96.22	92.40/90.05	91.49/93.1
Co-CS	91.67/93.32	88.41/92.5	88.37/91.1
DBLP	81.82/85.35	77.07/80.27	79.71/84.17

Table 2: Node classification accuracy (%) obtained using different GNN structures like GCN, GAT, and APPNP on different datasets using the proposed MGC algorithm for a coarsening ratio of 0.1 and using the original graph. In the table, x/y, where x represent the node classification accuracy obtained using the proposed MGC algorithm and y represents the accuracy obtained using the original graph.. It is evident that the proposed MGC method is suitable for all GNN architecture.

Dataset	BA		Amazon Photos	
Method	SPGC	LVN	SPGC	LVN
r=1	50.0/ 0.050	50.0/ 0.050	15.5 / 0.002	15.5 / 0.002
r=0.3	**14.3/0.04**	68.6/0.09	**6.7/0.002**	95.9/0.04
r=0.5	**20.7/0.04**	217.1/0.43	**7.9/0.002**	438.0/0.11
r=0.7	**26.8/0.03**	236.1/0.33	**8.9/0.002**	312.8/0.05

Table 4: The table summarizes the structural properties e.g. average degree and density of proposed SFCG and LVN [5]. The average degree/density of the original scale-free graphs e.g. Barabasi Albert(BA) and Amazon Photos are written in the first row(r=1). It can be observed that SFCG produces scale-free graphs with a density closer to the original as compared to LVN.

Data set(ACC)	r=k/p	GCOND	SCAL	Proposed SCG	Whole data
CORA	0.5	81.02± 0.37	82.7 ± 0.50	86.96 ± 1.30	
	0.3	81.56± 0.6	79.42 ± 1.71	85.65 ± 1.16	89.50 ± 1.2
CITESEER	0.5	74.28± 1.45	72.0± 0.5	74.39 ± 5.27	
	0.3	72.43± 0.94	74.54 ± 1.34	72.67± 1.11	78.09 ± 1.9
CO-PHY	0.3	93.79 ± 0.3	92.52 ± 0.9	88.94 ± 6.31	
	0.05	93.05 ± 0.26	73.09 ± 7.41	80.09 ± 0.33	96.22 ± 0.7
	0.03	92.81 ± 0.31	63.65 ± 9.65	79.10 ± 0.44	
PUBMED	0.3	77.77 ± 0.63	75.67 ± 2.57	84.48 ± 0.97	
	0.05	78.16 ± 0.30	72.82 ± 2.62	78.67 ± 0.57	88.89 ± 0.5
	0.03	78.04 ± 0.47	70.24 ± 2.63	74.67 ± 0.67	
CO-CS	0.3	88.02 ± 0.34	78.65 ± 3.90	92.75 ± 0.25	
	0.05	86.29 ± 0.63	34.45 ± 10.07	87.25 ± 0.90	93.32 ± 0.6
	0.03	86.32 ± 0.45	26.06 ± 9.29	81.38 ± 0.11	
DBLP	0.3	80.65 ± 0.50	74.50 ± 1.90	82.53 ± 0.86	
	0.05	79.15 ± 0.20	76.52 ± 2.88	76.27 ± 1.82	85.35 ± 0.8
	0.03	78.42 ± 1.26	75.49 ± 2.84	76.13 ± 1.49	

Table 3: The table summarizes the node classification accuracy on real benchmark datasets for the proposed SCG algorithm against the GCOND and SCAL. For small datasets, we have taken coarsening ratio $r = 0.3$ and 0.5; for large datasets, we have taken $r = 0.3, 0.05$ and 0.03. It is evident that enforcing sparisty in the coarsened graph improves the node classification accuracy.

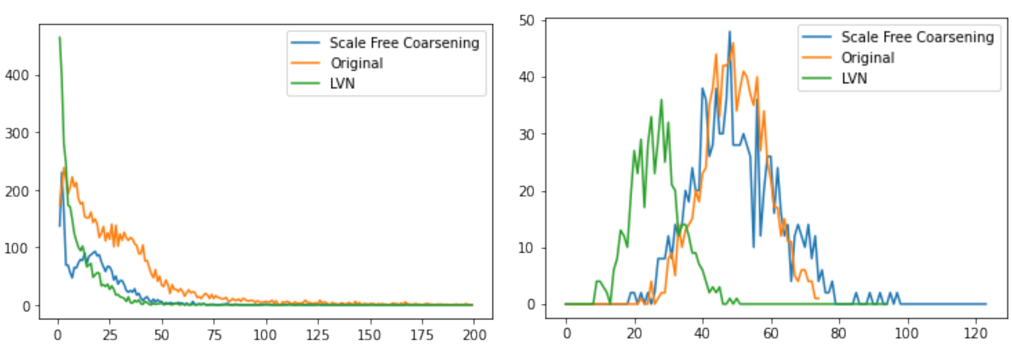


Figure 2: The figure shows the degree distribution for Amazon (left) and Erdos Renyi (right) datasets for three cases (i) original degree distribution, (ii) coarsened graph degree distribution(r=0.5) by LVN [5] and (iii) coarsened graph degree distribution by proposed SFCG. It is evident that the proposed SFCG algorithm learns coarsened graphs that preserve the structural properties and closely resemble the degree distribution of the original graph.

References

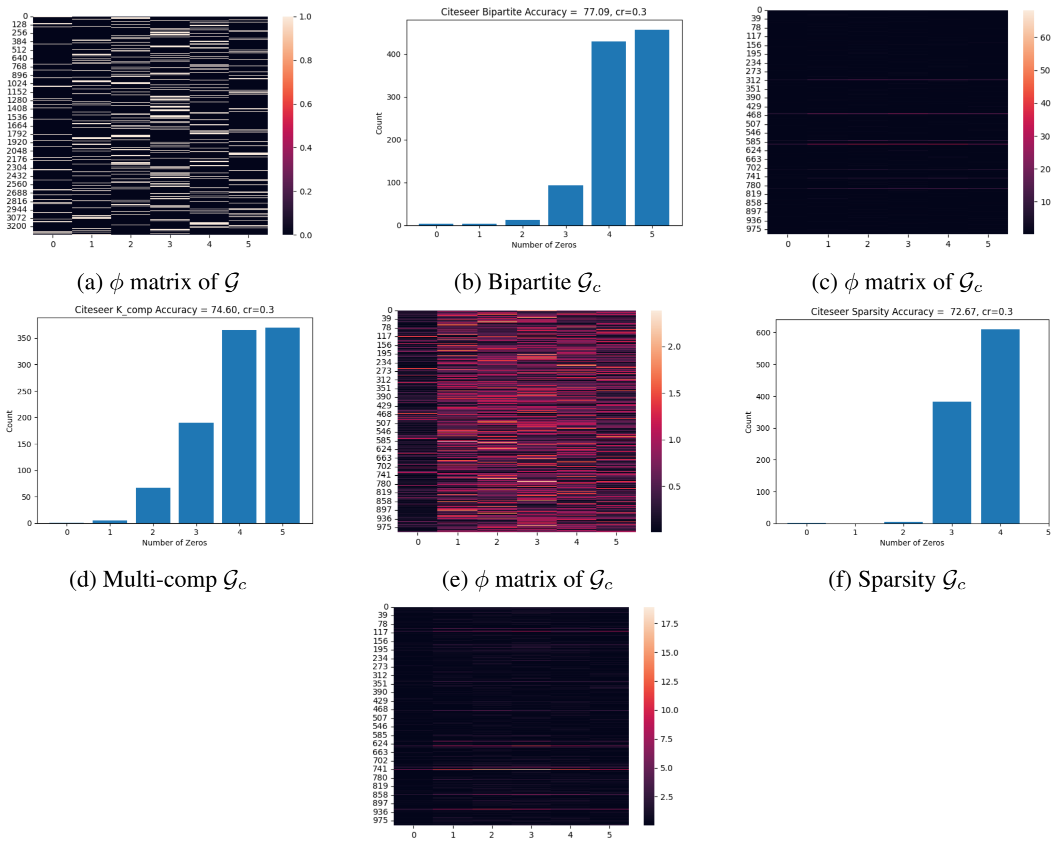
[1] Johannes Gasteiger, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. *arXiv preprint arXiv:1810.05997*, 2018.

[2] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[3] Manoj Kumar, Anurag Sharma, and Sandeep Kumar. A unified framework for optimization-based graph coarsening. *Journal of Machine Learning Research*, 24(118):1–50, 2023.

[4] Sandeep Kumar, Jiaxi Ying, José Vinícius de Miranda Cardoso, and Daniel P Palomar. A unified framework for structured graph learning via spectral constraints. *Journal of Machine Learning Research*, 21(22):1–60, 2020.

[5] Andreas Loukas. Graph reduction with spectral and cut guarantees. *Journal of Machine Learning Research*, 20(116):1–42, 2019.



(a) ϕ matrix of $\mathcal{G}$ (b) Bipartite $\mathcal{G}_c$ (c) ϕ matrix of $\mathcal{G}_c$

(d) Multi-comp $\mathcal{G}_c$ (e) ϕ matrix of $\mathcal{G}_c$ (f) Sparsity $\mathcal{G}_c$

(g) ϕ matrix of $\mathcal{G}_c$

Figure 1: The figures presented here depict the ϕ matrices of four different graph representations: the original graph, the bipartite coarsened graph, the multi-component coarsened graph, and the sparse coarsened graph. Additionally, a histogram showcases the sparsity levels within each row of the coarsened graphs, focusing on data from the Citeseer dataset. Notably, these visualizations illustrate a critical point: the sparser the heat map of the coarsened graph, the more informative it becomes. Specifically, the node classification accuracy achieved using the bipartite coarsened graph is 77.11%. Meanwhile, the multi-component and sparse coarsened graphs yield a node classification accuracy of 74.68% and 72.67% for a coarsening ratio of 0.3.

[6] Petar Velivcković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.