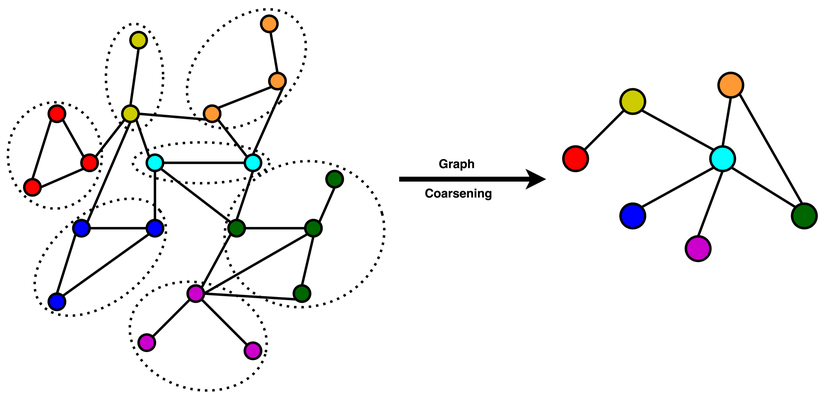
Multi-Component Coarsened Graph Learning for Scaling Graph Machine Learning

Subhanu Halder [1] Manoj Kumar [2] Sandeep Kumar [1]

[1]Indian Institute of Technology Delhi, India [2]The LNM Institute of Information Technology, Jaipur, India

Graph Coarsening



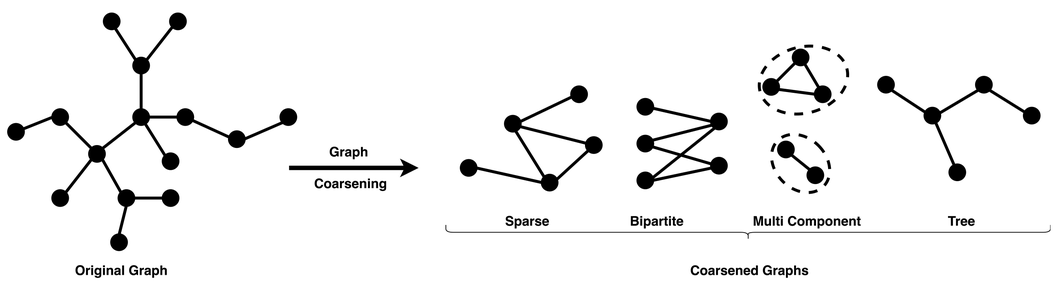
- How do we map a large graph to a coarsened graph? Using $C \in \mathbb{R}_{+}^{p \times k}$, which belongs to

$$\mathcal{C} = \Big\{ \langle C_i, C_j \rangle = 0 \; \forall \; i \neq j, \quad \langle C_i, C_i \rangle = d_i, \|C_i\|_0 \geq 1, \|[C^T]_i\|_0 = 1 \Big\}$$

- How do we find Laplcaian and feature matrix of the coarsened graph?

$$\Theta_c = C^T \Theta C, \quad X_c = PX, \quad X = P^{\dagger} X_c = C X_c$$

Structured Graph Reduction

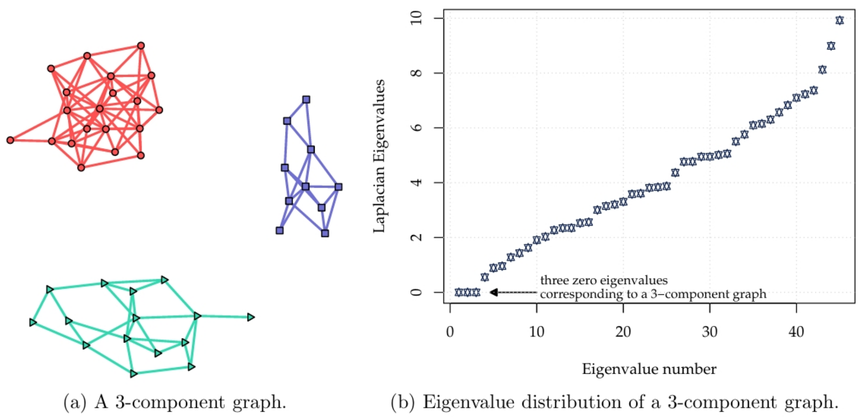


Structured Graph Learning via Laplacian Spectral Constraints

This formulation uses the Laplacian matrix spectral properties to learn the Multi-component graph. The eigenvalues of the Laplacian matrix for an n-component graph are expressed as:

$$\mathcal{S}_\lambda = \{\{\lambda_j = 0\}_{j=1}^{n}, c_1 \leq \lambda_{n+1} \leq \ldots \leq \lambda_k \leq c_2\} \tag{1}$$

where $n \geq 1$ denotes the number of connected components in the learned coarsened graph, and $c_1, c_2 > 0$ are constants that depend on the number of edges and their weights.



(a) A 3-component graph. (b) Eigenvalue distribution of a 3-component graph.

Figure 1. A 3-component graph and the eigenvalue distribution of its Laplacian matrix: three zero eigenvalues corresponding to three components.

Application of Multi-Component Graph

Recommendation	Edge Computing	Classification/Clustering
Capture diverse user-item interactions for improved recommendations.	Optimize dynamic application placement efficiency.	Accurately group nodes and classify graph entities.

Framework for Multi-Component Coarsened graph Learning

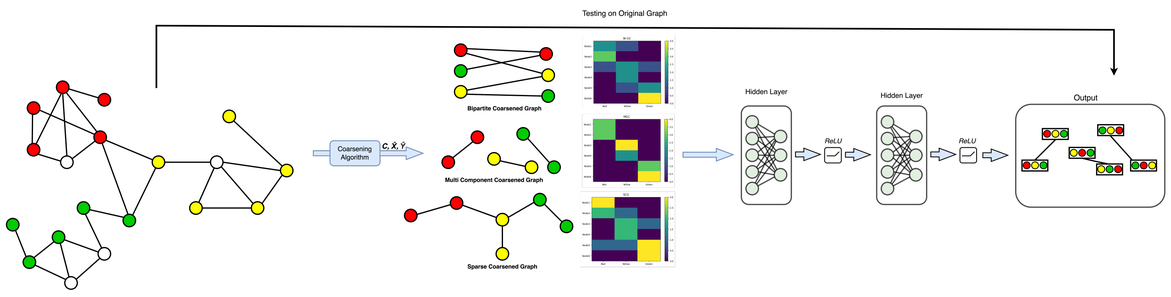


Figure 2. Sequence of steps in performing node classification task using a coarsened graph

Our approach for node classification using coarsened graph:

Input Graph: $G(A, X)$ → Learn structured-coarsened graph $G_c(A_c, X_c, C)$ using $G(A, X)$ → Label determination of coarsened graph $Y_c = \text{argmax}(C^T Y)$ → Graph Neural Network training using $G_c(A_c, X_c, Y_c)$ → Testing on Original graph

Optimization problem for MGC

$$\min_{C,\Lambda,U} \; -\gamma \log\det(\Lambda) + \frac{\lambda}{2}\|C^T\|_{1,2}^2 + \frac{\beta}{2}||C^T \Theta C - U\Lambda U^T||_F^2$$
$$\text{s.t. } \; C \in \mathcal{S}_c = \Big\{ C \geq 0 | \; \|[C^T]_i\|_2^2 \leq 1 \; \forall \; i = 1, .., p \Big\}, \quad U^T U = I, \; \Lambda \in S_\lambda$$

Algorithm

Proposed algorithm: Variables $\mathcal{X} = \left(\tilde{X}, C\right)$, solved using alternate block majorization-minimization:

$$\text{Sub-problem for } C: \quad \min_{C \in \mathcal{S}_c} \; \frac{1}{2} C^T C - C^T A$$

$$\text{Sub-problem for } U: \quad \min_{U^T U = I_q} \; \frac{\beta}{2}||C^T \Theta C - U\Lambda U^T||_F^2 = -\frac{\beta}{2}\text{tr}(U^T C^T \Theta C U \Lambda)$$

$$\text{Sub-problem for } \Lambda: \quad \min_{\Lambda \in \mathcal{S}_\lambda} \; -\gamma \log\det(\Lambda) + \frac{\beta}{2}||C^T \Theta C - U\Lambda U^T||_F^2$$

Algorithm 1: Multi-component Graph Coarsening (MGC)

Input: Graph Laplacian Θ, hyperparameters β, γ, λ
while stopping criteria not met **do**
  Update C:
    Compute gradient:
    $\nabla f(C^{(t)}) = \lambda C^{(t)} \mathbf{1} + 2\beta \Theta C^{(t)} (C^{(t)T} \Theta C^{(t)} - U\Lambda U^T)$
    Update: $C^{(t+1)} = \left(C^{(t)} - \frac{1}{L}\nabla f(C^{(t)})\right)^+$
  Update U:
    $U^{(t+1)} =$ eigenvectors of $C^T \Theta C$ corresponding to eigenvalues λ_{n+1} to λ_k
  Update Λ:
    Compute $d_i = (U^T C^T \Theta C U)_{ii}$ for $i = 1, \ldots, q$
    Update: $\lambda_i = \frac{1}{2}(d_i + \sqrt{d_i^2 + \frac{4\gamma}{\beta}})$
end while
Return: Final coarsening matrix C and coarsened Laplacian $\Theta_c = C^T \Theta C$

The worst-case computational complexity $O(p^2 k)$

Node classification Using MGC

Data set	r=k/p	Baseline		Proposed
		GCOND	SCAL	MGC
CORA	0.5	81.02 ± 0.37	82.7 ± 0.50	**87.20 ± 0.43**
	0.3	81.56 ± 0.62	79.42 ± 1.71	**84.56 ± 1.40**
	0.1	79.47 ± 0.45	71.38 ± 3.62	**76.02 ± 0.93**
CITESEER	0.5	74.28 ± 1.45	72.04 ± 0.54	**78.80 ± 1.20**
	0.3	72.43 ± 0.94	68.87 ± 1.37	**74.60 ± 2.31**
	0.1	70.46 ± 0.41	68.58 ± 2.32	**70.57 ± 1.25**
PUBMED	0.1	78.57 ± 0.24	73.59 ± 3.56	**84.81 ± 1.56**
	0.05	78.16 ± 0.30	72.82 ± 2.62	**81.89 ± 0.00**
	0.03	78.04 ± 0.47	70.24 ± 2.63	**80.70 ± 0.00**
CO PHY	0.1	92.98 ± 0.52	86.43 ± 2.40	**94.71 ± 0.22**
	0.05	93.05 ± 0.26	73.09 ± 7.41	**94.52 ± 0.19**
	0.03	92.81 ± 0.31	63.65 ± 9.65	**93.64 ± 0.25**
NELL	0.1	77.3±0.4	*OOM*	**84.62±0.85**
OGBN-arxiv	0.1	61.3±0.5	*OOM*	**64.80±1.90**

Figure 3. Node classification accuracy on real datasets comparing the proposed MGC algorithms against GCOND and SCAL. MGC outperforms state-of-the-art methods across all datasets.

Comparison with Full dataset and Runtime Efficiency

Data set	MGC	Whole datset
CORA	87.20±0.43	89.51±1.23
CITESEER	78.80±1.20	78.09±1.96
PUBMED	84.81±1.56	88.89± 0.59

Figure 4. **Node classification accuracy:** GNN trained on the original graph (whole dataset) vs. the coarsened graph using MGC. Coarsened graph achieves similar accuracy with improved efficiency. OOM = Out Of Memory.

Dataset(τ)	r = k/p	GCOND	SCAL	MGC	Whole dataset
CORA	0.05	329.86	27.76	**2.78**	2.86
CITESEER	0.05	331.33	56.21	**3.73**	5.24
PUBMED	0.05	202.04	54.09	**46.09**	58.85
CO-CS	0.05	1600.32	180.16	**49.80**	72.31

Figure 5. **Computation time (τ):** Time (in sec.) for coarsening and node classification with a ratio of 0.05. MGC is faster than SOTA baselines and even faster than classification on the full graph.

References

[SBP17] Y. Sun, P. Babu, and D. P. Palomar. Majorization-Minimization Algorithms in Signal Processing, Communications, and Machine Learning. ().

[JLJ20] Y. Jin, A. Loukas, and J. JaJa. Graph Coarsening with Preserved Spectral Properties.

[Kum+20] S. Kumar et al. A unified framework for structured graph learning via spectral constraints. ().

[Jin+21] W. Jin et al. *Graph Condensation for Graph Neural Networks*. arXiv: 2110.07580 [cs.LG].

[Kum+23] M. Kumar et al. Featured Graph Coarsening with Similarity Guarantees.