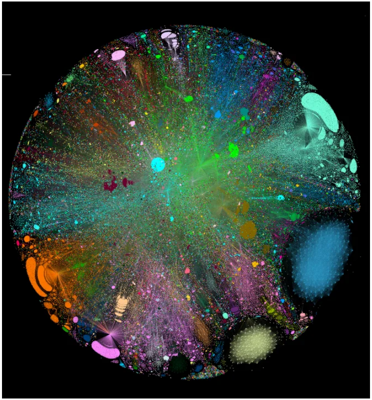
Graph Coarsening with MP guarantees

Antonin Joly, Nicolas Keriven

Background and Notations

- Graph such recommender systems (Reddit) too big to enter GPU
- Graph $G = \{V, E, A\}$
- $L = D - A$ Laplacian symmetric psd matrix
- For vector X, $\|X\|_L = \sqrt{X^T L X}$, smoothness on edges.



Motivation

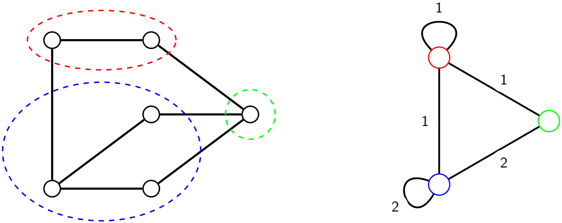


Figure: Graph Coarsening with coarsening ratio of 4/7

Motivation

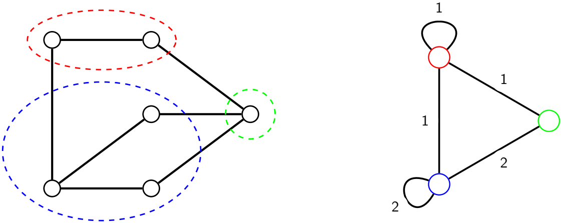
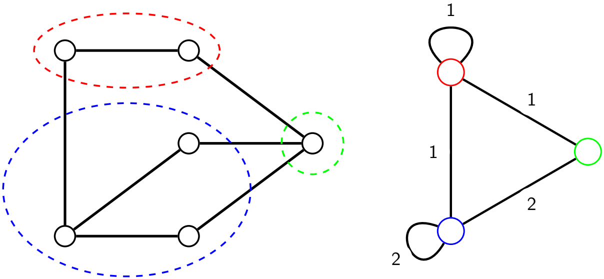


Figure: Graph Coarsening with coarsening ratio of 4/7

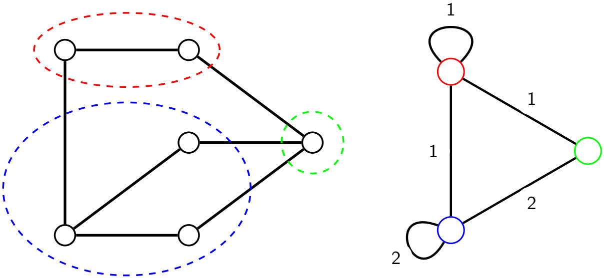
Is training a GNN on a coarsened graph probably close to training it on the original graph ?

Background Coarsening



[0] [1] Andreas Loukas, *Graph Reduction with Spectral and Cut Guarantees*, JMLR 2019

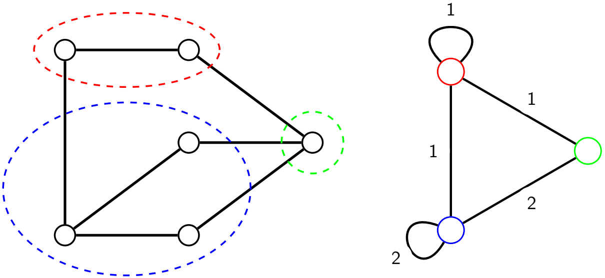
Background Coarsening



$$Q = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

[0] [1] Andreas Loukas, *Graph Reduction with Spectral and Cut Guarantees*, JMLR 2019

Background Coarsening



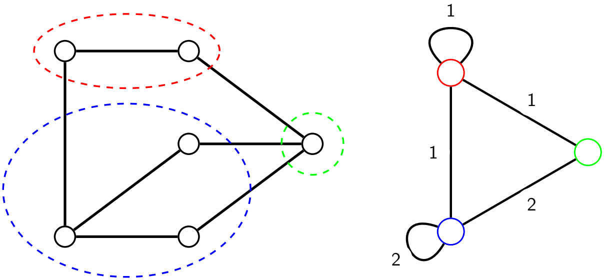
$$x_c = Qx$$

$$\tilde{x} = Q^+ x_c$$
$$= \Pi x$$

$$Q = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

[0][1] Andreas Loukas, *Graph Reduction with Spectral and Cut Guarantees*, JMLR 2019

Background Coarsening



$$x_c = Qx$$

$$\tilde{x} = Q^+ x_c$$
$$= \Pi x$$

$$Q = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad Q^+ = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \Pi = Q^+ Q = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 \\ 0 & 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 \\ 0 & 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

[0] [1] Andreas Loukas, *Graph Reduction with Spectral and Cut Guarantees*, JMLR 2019

Spectral guarantee

Definition (Restricted Spectral Approximation constant)

Consider a subspace $\mathcal{R} \subset \mathbb{R}^N$, a Laplacian L, a coarsening matrix Q and its corresponding projection operator $\Pi = Q^+Q$. The *RSA constant* $\epsilon_{L,Q,\mathcal{R}}$ is defined as

$$\epsilon_{L,Q,\mathcal{R}} = \sup_{x \in \mathcal{R}, \|x\|_L = 1} \|x - \Pi x\|_L$$

Many classical coarsening algorithms aim to minimize the RSA

Message Passing GNN

With initial node features H^0, node representation matrix at layer l H^l and the **propagation matrix** S ; the GNN Φ_θ outputs after k layers:

$$H^l = \sigma\left(SH^{l-1}\theta_l\right), \quad \Phi_\theta(H^0, S) = H^k\,,$$

Message Passing GNN

With initial node features H^0, node representation matrix at layer l H^l and the **propagation matrix** S ; the GNN Φ_θ outputs after k layers:

$$H^l = \sigma\left(SH^{l-1}\theta_l\right), \quad \Phi_\theta(H^0, S) = H^k,$$

What is the best choice of propagation matrix on a coarsened graph ?

Message Passing GNN

With initial node features H^0, node representation matrix at layer l H^l and the **propagation matrix** $\mathbf{S}$; the GNN Φ_θ outputs after k layers:

$$H^l = \sigma\left(SH^{l-1}\theta_l\right), \quad \Phi_\theta(H^0, S) = H^k,$$

What is the best choice of propagation matrix on a coarsened graph ?

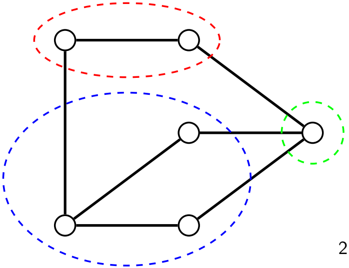
- $S_c = f_S(A)$
- S_c^{diag}, weighted self loops [2]

With $S_c = f_S(A)$ and S_c^{diag} spectral guarantees on the coarsening does not lead to message passing guarantees

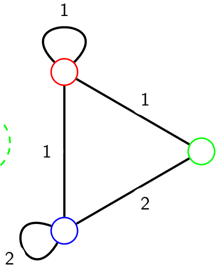
[0][2] Huang et al , *Scaling Up Graph Neural Networks Via Graph Coarsening*, KDD 2021

A new propagation matrix for G_c

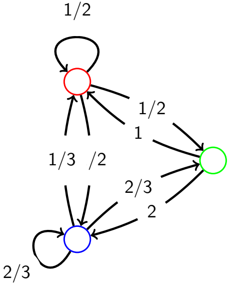
$$S_c^{\mathsf{MP}} = QSQ^{+} \in \mathbb{R}^{n \times n}$$



(a) Original graph, $S = A$



(b) $S_c = A_c$



(c) $S_c = QSQ^{+}$ (Ours)

Propagation bound theorem

Assumptions

- Π and S are both $\ker(L)$-preserving.
- S is $\mathcal{R}$-preserving (i.e $\forall x \in \mathcal{R}$, $Sx \in \mathcal{R}$).

Define S_c^{MP} as $S_c^{\mathsf{MP}} = QSQ^+$, we have

$$\boxed{\|Sx - Q^+ S_c^{\mathsf{MP}} x_c\|_L \leq \epsilon_{L,Q,\mathcal{R}} \|x\|_L \left(C_S + C_\Pi\right)}$$

Propagation bound theorem experiment

$$\|Sx - Q^+ S_c^{\mathsf{MP}} x_c\|_L \leq \epsilon_{L,Q,\mathcal{R}} \|x\|_L \left(C_S + C_\Pi\right)$$

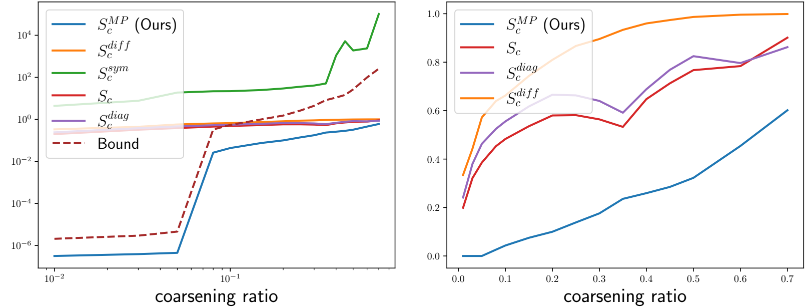


Figure: Log and linear scale of $\max_{x\in\mathcal{R}} \|S^6 x - Q^+ {S_c^{\mathsf{MP}}}^6 x_c\|_L / \|x\|_L$,the lower, the better

Train GNN on G_c (theorem)

Assumptions

- There is a constant C_J such that $|J(x) - J(x')| \leq C_J \|x - x'\|_L$, with J the loss function
- σ is $\mathcal{R}$-preserving, that is, for all $x \in \mathcal{R}$, we have $\sigma(x) \in \mathcal{R}$, $\|\sigma(x) - \sigma(x')\|_L \leq C_\sigma \|x - x'\|_L$, σ and Q^+ commute: $\sigma(Q^+ y) = Q^+ \sigma(y)$.

For all node features $X \in \mathbb{R}^{N \times d}$ such that $X_{:,i} \in \mathcal{R}$, denoting by $\theta^\star = \arg\min_{\theta \in \Theta} R(\theta)$ and $\theta_c = \arg\min_{\theta \in \Theta} R_c(\theta)$, we have

$$\boxed{R(\theta_c) - R(\theta^\star) \leq C \epsilon_{L,Q,\mathcal{R}} \|X\|_{:,L}}$$

Dataset Presentation

Dataset	# Nodes	# Edges	# Features	#classes
Reddit	232,965	114,615,892	602	41
Reddit90	23,298	8,642,864	602	41
Reddit99	2,331	10,838	602	41
Cora PCC	2,485	10,138	1,433	7
Cora70	746	3,716	1,433	7
Citeseer PCC	2,120	7,358	3,703	6
Citeseer70	636	2,122	3,703	6

Training GNN on G_c (experiments)

Relaxing activation function assumption with GCNconv

SGC	Cora		Citeseer		Reddit	
r	0.5	0.7	0.5	0.7	0.9	0.99
S_c^{sym}	16.1 $\pm$ 3.8	16.4 $\pm$ 4.7	18.6 $\pm$ 4.6	19.8 $\pm$ 5.0	37.1 $\pm$ 6.6	3.7 $\pm$ 5.5
S_c^{diff}	21.8 $\pm$ 2.2	13.6 $\pm$ 2.8	30.5 $\pm$ 0.2	23.1 $\pm$ 0.0	18.3 $\pm$ 0.0	14.9 $\pm$ 0.0
S_c	78.7 $\pm$ 0.0	74.6 $\pm$ 0.1	72.8 $\pm$ 0.1	72.5 $\pm$ 0.1	87.5 $\pm$ 0.1	37.3 $\pm$ 0.0
S_c^{diag}	78.7 $\pm$ 0.1	77.3 $\pm$ 0.0	73.4 $\pm$ 0.1	73.1 $\pm$ 0.4	87.6 $\pm$ 0.1	37.3 $\pm$ 0.0
S_c^{MP} **(ours)**	**80.3** $\pm$ 0.1	**78.5** $\pm$ 0.0	**74.6** $\pm$ 0.1	**74.2** $\pm$ 0.1	**90.2** $\pm$ 0.0	**64.1** $\pm$ 0.0
Full Graph	81.6 $\pm$ 0.1		73.6 $\pm$ 0.0		94.9	

GCNconv	Cora		Citeseer		Reddit	
r	0.5	0.7	0.5	0.7	0.9	0.99
S_c^{sym}	78.1 $\pm$ 1.3	30.8 $\pm$ 2.5	62.5 $\pm$ 11	52.7 $\pm$ 3.6	48.1 $\pm$ 8.9	34.8 $\pm$ 4.0
S_c^{diff}	74.5 $\pm$ 0.9	62.6 $\pm$ 7.1	71.2 $\pm$ 1.7	37.6 $\pm$ 0.9	71.3 $\pm$ 1.0	18.7 $\pm$ 1.7
S_c	79.9 $\pm$ 0.9	78.1 $\pm$ 1.0	70.7 $\pm$ 1.0	67.1 $\pm$ 3.1	88.0 $\pm$ 0.1	54.2 $\pm$ 2.4
S_c^{diag}	**80.4** $\pm$ 0.8	**78.6** $\pm$ 1.3	70.2 $\pm$ 0.8	69.3 $\pm$ 1.9	**88.1** $\pm$ 0.2	55.5 $\pm$ 1.8
S_c^{MP} **(ours)**	79.8 $\pm$ 1.5	78.2 $\pm$ 0.9	**72.0** $\pm$ 0.8	**70.0** $\pm$ 1.0	84.4 $\pm$ 0.3	**60.3** $\pm$ 0.9
Full Graph	81.6 $\pm$ 0.6		73.1 $\pm$ 1.5		OOM	

Appendices

Adaption of Loukas coarsening algorithm

Algorithm Loukas algorithm Adapted

```
Require: Adjacency matrix A, Laplacian L = f_L(A), propagation matrix S, a coarsening ratio r , preserved space R,
    maximum number of nodes merged at one coarsening step : n_e
n_obj ← int(N − N × r) the number of nodes wanted at the end of the algorithm.
compute cost matrix B_0 ← V V^T L^{−1/2} with V an orthonormal basis of R
Q ← I_N
while n ≥ n_obj do
    Make one coarsening STEP l
    Create candidate contraction sets.
    For each contraction C, compute cost(C, B_{l−1}, L_{l−1}) = ‖Π_C B_{l−1}(B_{l−1}^T L_{l−1} B_{l−1})^{−1/2}‖_{L_C} / (|C|−1)
    Sort the list of contraction set by the lowest score
    Select the lowest scores non overlapping contraction set while the number of nodes merged is inferior to min(n −
    n_obj, n_e)
    Compute Q_l, Q_l^+, uniform intermediary coarsening with contraction sets selected
    B_l ← Q_l B_{l−1}
    Q ← Q_l Q
    A_l ← (Q_l^+)^⊤ A_{l−1} Q_l^+ − diag((Q_l^+)^⊤ A_{l−1} Q_l^+) 1_n)
    L_{l−1} = f_L(A_{l−1})
    n ← min(n − n_obj, n_e)
end while
IF uniform coarsening THEN Q ← row-normalize(Q_l Q)
Compute S_c^MP = Q S Q^+
return Q, S_c^MP
```

Training on coarsened graph procedure

Algorithm Training Procedure

Require: Adjacency A, node features X, desired propagation matrix S, preserved space $\mathcal{R}$, Laplacian L, a coarsening ratio r

Q, $S_c^{\mathsf{MP}} \leftarrow$ Coarsening-algorithm$(A, L, S, r, \mathcal{R})$
$X_c \leftarrow QX$
Initialize model (SGC or GCNconv)
for N_{epochs} iterations **do**
    compute coarsened prediction $\Phi_\theta(S_c^{\mathsf{MP}}, X_c)$
    uplift the predictions : $Q^+\Phi_\theta(S_c^{\mathsf{MP}}, X_c)$
    compute the cross entropy loss $J(Q^+\Phi_\theta(S_c^{\mathsf{MP}}, X_c))$
    Backpropagate the gradient
    Update θ
end for

Proof Theorem propagation

Key argument : For this well-designed choice of S_c^{MP}, $Q^+ S_c^{\mathsf{MP}} x_c = \Pi S \Pi x$
Since $x \in \mathcal{R}$ and S is $\mathcal{R}$-preserving, we have

$$\|\Pi^\perp x\|_L \leq \epsilon_{L,Q,\mathcal{R}} \|x\|_L$$

where $\Pi^\perp = I_N - \Pi$, and similarly for Sx. Moreover, under Assumption, both Π and S are ker(L)-preserving, such that $\|\Pi S x\|_L \leq \|\Pi S\|_L \|x\|_L$ for all x. Then

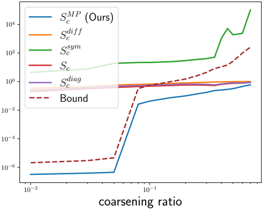
$$\begin{aligned}
\|Sx - Q^+ S_c^{\mathsf{MP}} x_c\|_L &= \|Sx - \Pi S \Pi x\|_L \\
&= \|Sx - \Pi S x + \Pi S x - \Pi S \Pi x\|_L \\
&= \|\Pi^\perp S x + \Pi S \Pi^\perp x\|_L \\
&\leq \|\Pi^\perp S x\|_L + \|\Pi S \Pi^\perp x\|_L \\
&\leq \epsilon_{L,Q,\mathcal{R}} \|Sx\|_L + \|\Pi S\|_L \|\Pi^\perp x\|_L \\
&\leq \epsilon_{L,Q,\mathcal{R}} \|Sx\|_L + \epsilon_{L,Q,\mathcal{R}} \|\Pi S\|_L \|x\|_L = \epsilon_{L,Q,\mathcal{R}} \|x\|_L \left(C_S + C_\Pi\right)
\end{aligned}$$

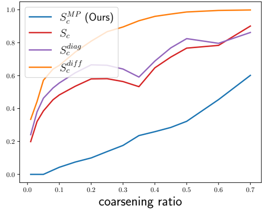
More about ker L assumption

- For uniform coarsenings with $L = D - A$ and connected graph G, $\ker(L)$ is the constant vector[1], and Π is $\ker(L)$-preserving.This is the case examined by Loukas.
- For positive definite "Laplacians", $\ker(L) = \{0\}$. This is a deceptively simple solution for which $\|\cdot\|_L$ is a true norm. This can be obtained e.g. with $L = \delta I_N + \hat{L}$ for any p.s.d. Laplacian $\hat{L}$ and small constant $\delta > 0$. This leaves its eigenvectors unchanged and add δ to its eigenvalues, and therefore does not alter the fundamental structure of the coarsening problem.

[1] Note that this would also work with several connected components, if no nodes from different components are mapped to the same super-node.

Propagation theorem other propagation matrices





Knowing that $S = f_S(A)$, we compare:

- $S_c^{\mathrm{MP}} = QSQ^+$, our proposed matrix
- $S_c = f_S(A_c)$, the naive choice
- $S_c^{diag} = \hat{D}'^{-1/2}(A_c + C)\hat{D}'^{-1/2}$, proposed in [?]
- $S_c^{diff} = QSQ^\top$, which is roughly inspired by Diffpool [?]
- $S_c^{sym} = (Q^+)^\top SQ^+$, which is the lifting employed to compute A_c

Figure: Log and linear scale of $\max_{x\in\mathcal{R}} \|S^6 x - Q^+ {S_c^{\mathrm{MP}}}^6 x_c\|_L / \|x\|_L$, the lower, the better