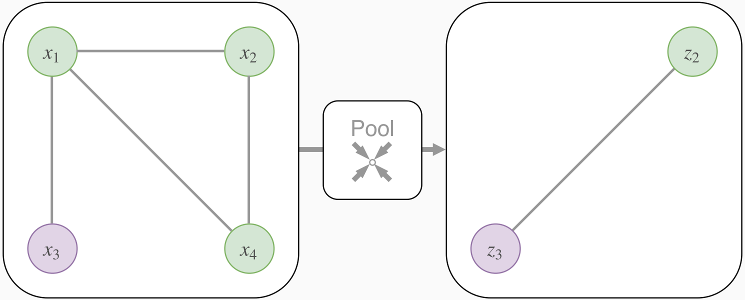
Spectral Clustering with Graph Neural Networks for Graph Pooling

F. M. Bianchi*, D. Grattarola*, C. Alippi

This talk

1. Executive summary
2. Method details
3. Experiments

Pooling in Graph Neural Networks



Reduce the number of nodes.

Pooling in Graph Neural Networks

Model-free

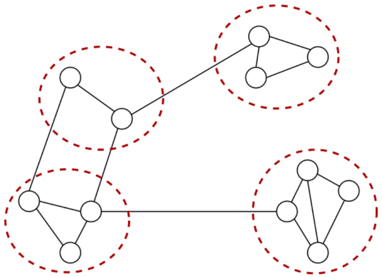
- Task-agnostic
- Pre-defined strategy
- Graph theory
- [1], [2]

Model-based

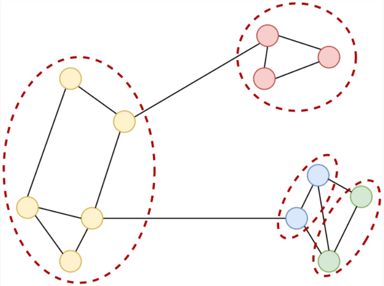
- Task-specific
- Learning to pool
- Heuristics
- [3], [4]

[1] I. S. Dhillon *et al.*, "Weighted graph cuts without eigenvectors a multilevel approach," 2007.
[2] F. M. Bianchi *et al.*, "Hierarchical Representation Learning in Graph Neural Networks with Node Decimation Pooling," 2019.
[3] Z. Ying *et al.*, "Hierarchical graph representation learning with differentiable pooling," 2018.
[4] S. J. Hongyang Gao, "Graph U-nets," 2019.

Spectral clustering



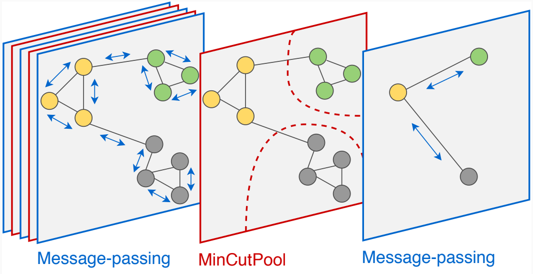
Standard spectral clustering.



Can we improve it?

MinCut pooling

- Learn to cluster with a neural network
- Find similar clusters to SC: use **minimum cut** as loss



MinCut pooling

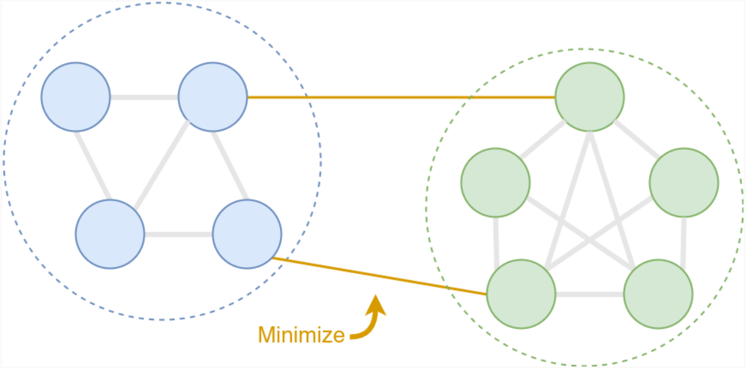
- Spectral clustering: non differentiable, expensive, edges only.
- Minimum cut objective as loss for NN
- NN can find balance between MinCut loss and task loss

Details

Spectral clustering $\leftrightarrow$ MinCut

Minimum cut: find K groups of nodes s.t.

- volume between clusters is minimized
- volume within clusters is maximized



MinCut optimization is written as

$$\text{maximize} \;\; \frac{1}{K}\sum_{k=1}^{K}\frac{\mathbf{C}_k^T\mathbf{A}\mathbf{C}_k}{\mathbf{C}_k^T\mathbf{D}\mathbf{C}_k}, \quad \text{s.t.} \quad \underbrace{\mathbf{C}\mathbf{1}_K = \mathbf{1}_N}_{\text{1 node} \longleftrightarrow \text{1 cluster}}$$

$\mathbf{C} \in \{0,1\}^{N \times K}$ is a discrete clustering matrix

Spectral clustering $\leftrightarrow$ MinCut

Relaxed formulation

$$\underset{\mathbf{Q}\in\mathbb{R}^{N\times K}}{\arg\max} \; \frac{1}{K}\sum_{k=1}^{K}\mathbf{Q}_k^T\mathbf{A}\mathbf{Q}_k,$$
$$\text{s.t. } \mathbf{Q}=\mathbf{C}(\mathbf{C}^T\mathbf{D}\mathbf{C})^{-\frac{1}{2}}, \; \underbrace{\mathbf{Q}^T\mathbf{Q}=\mathbf{I}_K,}_{\text{Orthogonal}} \; \underbrace{\mathbf{C}\mathbf{1}_K=\mathbf{1}_N}_{\substack{\text{Nodes split}\\ \text{between clusters}}}$$

$\mathbf{C}\in\mathbb{R}^{N\times K}$ is a continuous clustering matrix
$\mathbf{D}$ is the degree matrix

Spectral clustering $\leftrightarrow$ MinCut

Optimal solution:

$$\mathbf{Q}^* = \mathbf{U}_K \mathbf{O}$$

$\mathbf{U}_K$ is the eigenbasis of the top K eigenvalues
$\mathbf{O}$ is an orthogonal transformation

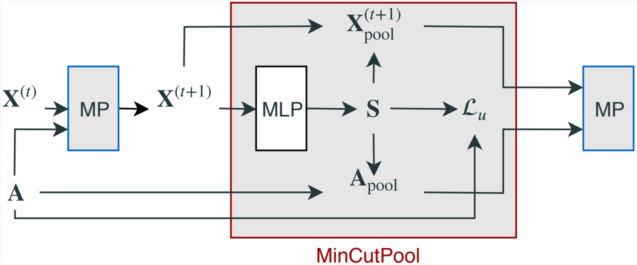
Spectral clustering

K-means on rows of $\mathbf{U}_K$ to get discrete C.

MinCut pooling

Learn to cluster:

- $\mathbf{S} = \text{MLP}(\mathbf{X})$
- Softmax $\implies \mathbf{S}\mathbf{1}_K = \mathbf{1}_N$



MinCut pooling

MinCut loss:

$$\mathcal{L}_c = -\frac{Tr(\mathbf{S}^T \mathbf{A} \mathbf{S})}{Tr(\mathbf{S}^T \mathbf{D} \mathbf{S})}$$

Optimal cluster assignments ($\mathcal{L}_c = -1$):

- $\mathbf{s}_i = [0.25, 0.25, 0.25, 0.25] \leftarrow$ This is bad
- $\mathbf{s}_i = [1.00, 0.00, 0.00, 0.00]$

MinCut pooling

Orthogonality loss (prevents bad minima og $\mathcal{L}_c$):

$$\mathcal{L}_o = \left\| \frac{\mathbf{S}^T\mathbf{S}}{\|\mathbf{S}^T\mathbf{S}\|_F} - \frac{\mathbf{I}_K}{\sqrt{K}} \right\|_F$$

MinCut pooling

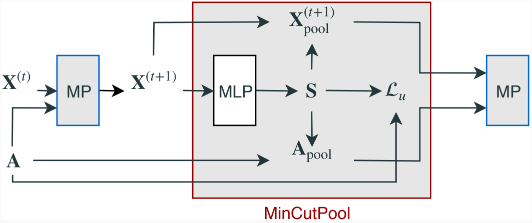
$$\mathcal{L}_u = \mathcal{L}_c + \mathcal{L}_o = \underbrace{-\frac{Tr(\mathbf{S}^T\tilde{\mathbf{A}}\mathbf{S})}{Tr(\mathbf{S}^T\tilde{\mathbf{D}}\mathbf{S})}}_{\mathcal{L}_c} + \underbrace{\left\| \frac{\mathbf{S}^T\mathbf{S}}{\|\mathbf{S}^T\mathbf{S}\|_F} - \frac{\mathbf{I}_K}{\sqrt{K}} \right\|_F}_{\mathcal{L}_o}$$

Final loss of the MinCutPool layer.

MinCut pooling

Pooling:

- $\mathbf{A}' = \mathbf{S}^\top \mathbf{A} \mathbf{S}$
- $\mathbf{X}' = \mathbf{S}^\top \mathbf{X}$
- Sum auxiliary loss $\mathcal{L}_u$ to supervised loss

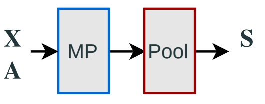


MinCut pooling

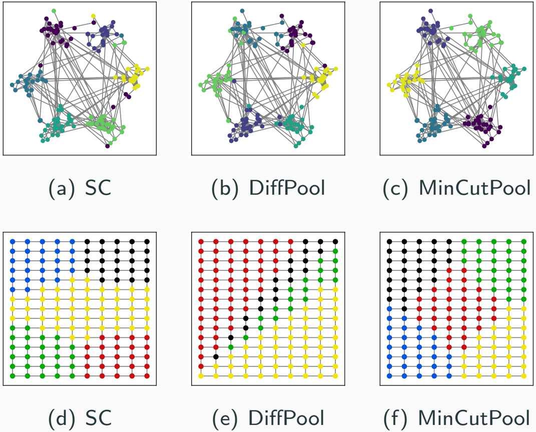
- OK for end-to-end learning
- Accounts for node features
- Cheap inference
- Balance between theoretical prior and task

Experiments

Clustering (point clouds)

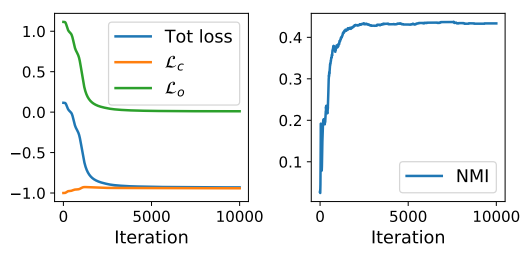


GNN architecture for clustering and segmentation.



Clustering simple graphs.

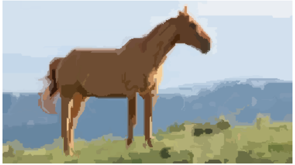
Clustering (citation networks)



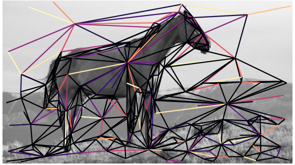
Dataset	K	Spectral clustering		DiffPool		MinCutPool	
		NMI	CS	NMI	CS	NMI	CS
Cora	7	0.025 ± 0.014	0.126 ± 0.042	0.315 ± 0.005	0.309 ± 0.005	**0.404** ± 0.018	**0.392** ± 0.018
Citeseer	6	0.014 ± 0.003	0.033 ± 0.000	0.139 ± 0.016	0.153 ± 0.020	**0.287** ± 0.047	**0.283** ± 0.046
Pubmed	3	0.182 ± 0.000	**0.261** ± 0.000	0.079 ± 0.001	0.085 ± 0.001	**0.200** ± 0.020	0.197 ± 0.019

Image segmentation

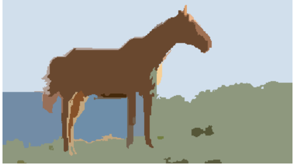
(a) Original image



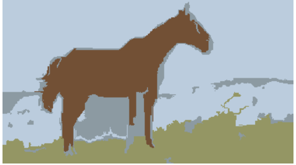
(b) Oversegmentation



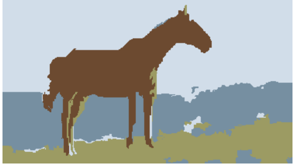
(c) Region Adjacency Graph



(d) SC



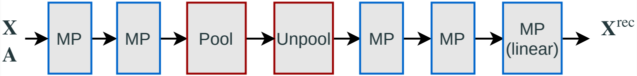
(e) DiffPool ($K = 4$)

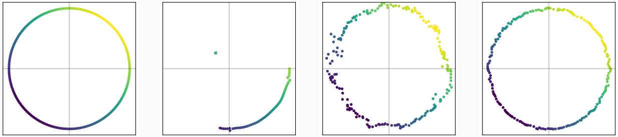


(f) MinCutPool ($K = 4$)

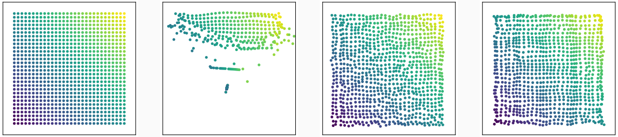
Segmentation by clustering the nodes of the Region Adjacency Graph.

Autoencoder





(g) Original (h) Top-K (i) DiffPool (j) MinCutPool



(k) Original (l) Top-K (m) DiffPool (n) MinCutPool

Graph classification

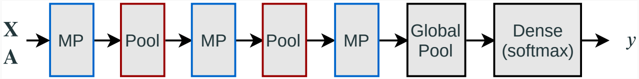


Table 1: Graph classification accuracy. Significantly better results ($p < 0.05$) are in bold.

Dataset	WL	Dense	No-pool	Graclus	NDP	DiffPool	Top-K	SAGpool	MinCutPool
Bench-easy	92.6	29.3±0.3	98.5±0.3	97.5±0.5	97.9±0.5	98.6±0.4	82.4±8.9	84.2±2.3	**99.0**±0.0
Bench-hard	60.0	29.4±0.3	67.6±2.8	69.0±1.5	**72.6**±0.9	69.9±1.9	42.7±15.2	37.7±14.5	**73.8**±1.9
Mutagenicity	**81.7**±1.1	68.4±0.3	78.0±1.3	74.4±1.8	77.8±2.3	77.6±2.7	71.9±3.7	72.4±2.4	79.9±2.1
Proteins	71.2±2.6	68.7±3.3	72.6±4.8	68.6±4.6	73.3±3.7	72.7±3.8	69.6±3.5	70.5±2.6	**76.5**±2.6
DD	78.6±2.7	70.6±5.2	76.8±1.5	70.5±4.8	72.0±3.1	**79.3**±2.4	69.4±7.8	71.5±4.5	**80.8**±2.3
COLLAB	74.8±1.3	79.3±1.6	**82.1**±1.8	77.1±2.1	79.1±1.5	81.8±1.4	79.3±1.8	79.2±2.0	**83.4**±1.7
Reddit-Binary	68.2±1.7	48.5±2.6	80.3±2.6	79.2±0.4	84.3±2.4	86.8±2.1	74.7±4.5	73.9±5.1	**91.4**±1.5

Conclusion

- Introduced MinCut pooling
- Learns how to pool graphs but is theoretically motivated
- Overcomes limitations of spectral clustering
- Works really well in practice

Presenter

Daniele Grattarola (@riceasphait)

Contacts

daniele.grattarola@usi.ch

filippombianchi@gmail.com

Code

Available on Spektral and Pytorch Geometric.

https://github.com/FilippoMB/Spectral-Clustering-with-Graph-Neural-Networks-for-Graph-Pooling