Sheaf Neural Networks

Jakob Hansen and Thomas Gebhart

The Ohio State University

University of Minnesota

NeurIPS 2020 Topological Data Analysis and Beyond Workshop

Overview

Graph convolutional networks (GCN) are characterized by (spatial/spectral) graph diffusion operations [1,2].

Cellular sheaves generalize graph diffusion and **characterize** relationships.

Sheaf Neural Networks generalize GCNs via sheaf diffusion.

Sheaf Neural Networks allow for GCN-like computations over graphs with asymmetric/non-constant relations or varying node features.

1. Zhou, Jie, et al. "Graph neural networks: A review of methods and applications." *arXiv preprint arXiv:1812.08434* (2018).
2. Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In 5th International Conference on Learning Representations, ICLR 2017

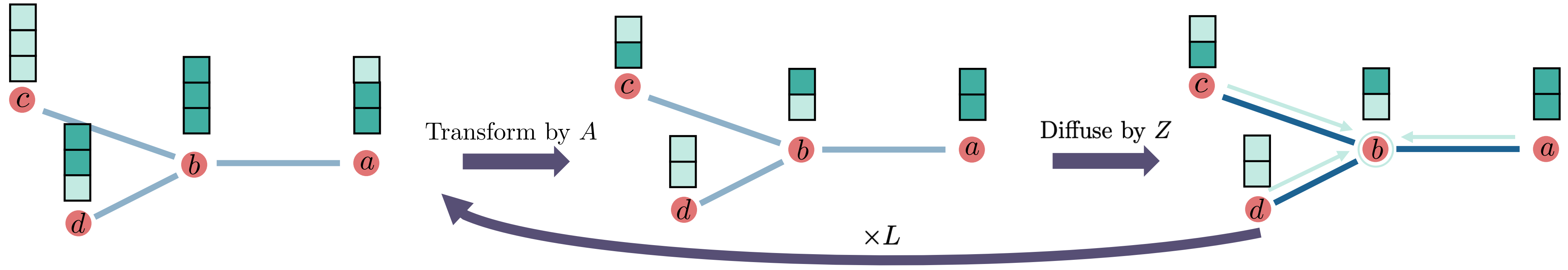
Graph Convolutional Networks

Given input feature matrix X of size $N_v \times N_{\text{feat}}^{\text{in}}$ and adjacency matrix Z

$$\text{GraphConv}(A)(X) = \rho(ZXA)$$

where A of size $N_{\text{feat}}^{\text{in}} \times N_{\text{feat}}^{\text{out}}$ is a parameter matrix.

Each layer is a one-hop diffusion step according to local connectivity.

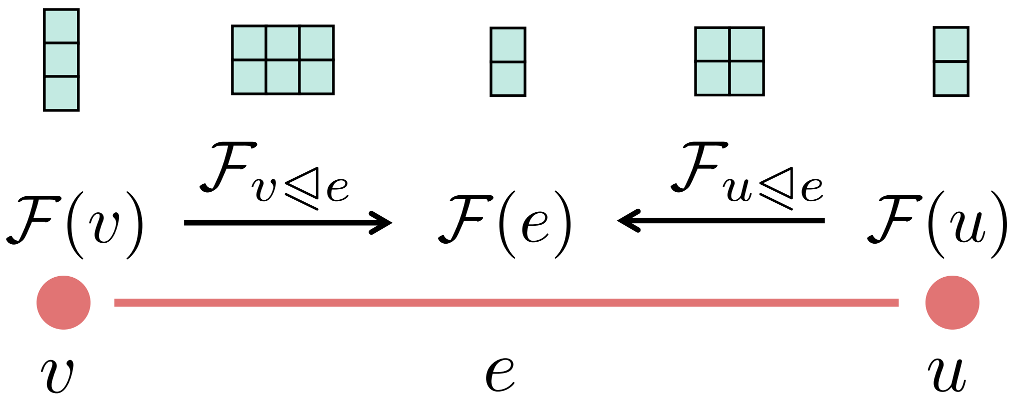


Cellular Sheaves Characterize Relations

Cellular sheaves describe relationships, not just connections.

Given undirected graph G, a cellular sheaf $\mathcal{F}$ is defined by:

- a vector space $\mathcal{F}(v)$ for each vertex v of G,
- a vector space $\mathcal{F}(e)$ for each edge e of G, and
- a linear map $\mathcal{F}_{v \trianglelefteq e} : \mathcal{F}(v) \to \mathcal{F}(e)$ for each incident vertex-edge pair $v \trianglelefteq e$



$$\delta : C^0(G;\mathcal{F}) \to C^1(G;\mathcal{F})$$

$$L_{\mathcal{F}} = \delta^T \delta = \begin{pmatrix} \mathcal{F}^T_{v \trianglelefteq e}\mathcal{F}_{v \trianglelefteq e} & -\mathcal{F}^T_{v \trianglelefteq e}\mathcal{F}_{u \trianglelefteq e} \\ -\mathcal{F}^T_{u \trianglelefteq e}\mathcal{F}_{v \trianglelefteq e} & \mathcal{F}^T_{u \trianglelefteq e}\mathcal{F}_{u \trianglelefteq e} \end{pmatrix}$$

Sheaf Neural Networks

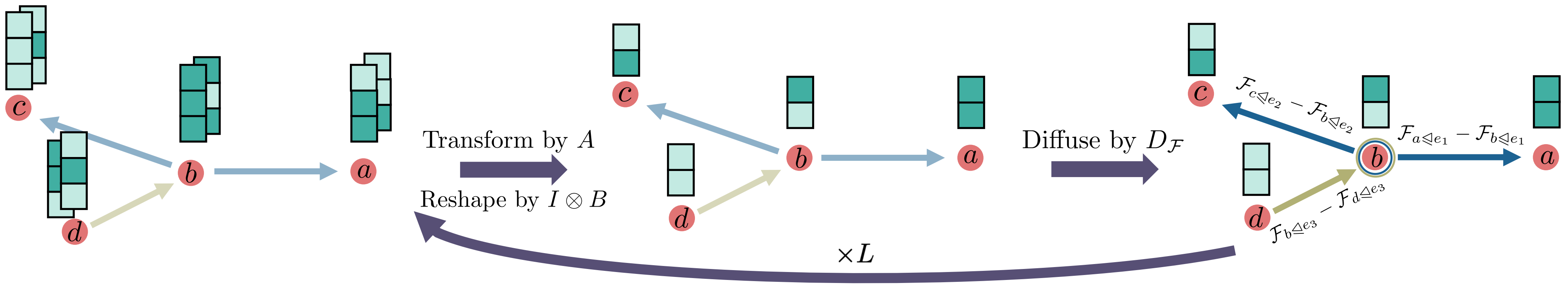
Assume N_v nodes in the graph each with N_{feat} k-dimensional features.

Concatenate node features into input matrix X of size $N_v k \times N_{\text{feat}}$.

$$\text{SheafConv}(A, B)(X) = \rho\left(D_{\mathcal{F}}(I \otimes B) X A\right)$$

A ($N_{\text{feat}}^{\text{in}} \times N_{\text{feat}}^{\text{out}}$) and B ($k \times k$) are learnable parameters.

$D_{\mathcal{F}} = I - \frac{1}{d_{\max}} L_{\mathcal{F}}.$



Increased Expressivity

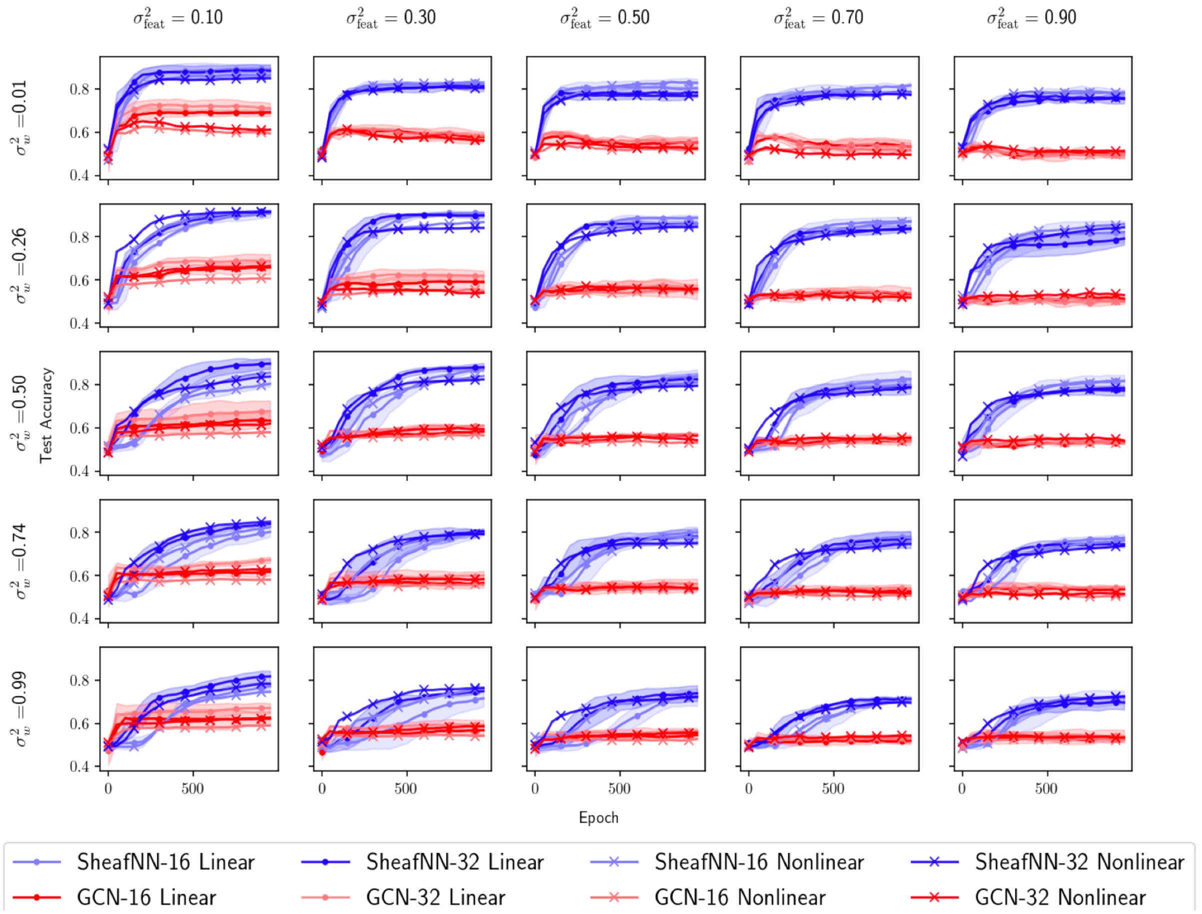
Sheaf convolution respects non-constant and asymmetric relational data.

Unfortunately, few semi-supervised graph datasets admit non-trivial sheaf structure.

Can also learn the sheaf structure during training—an exciting direction for future work.

More ideas from cellular sheaf theory may be exploited for relational learning.

Sheaf neural networks outperform GCNs on signed graphs (see paper for details).



Thank You!

Special thanks to the Topological Data Analysis and Beyond organizers: https://tda-in-ml.github.io/organisers

The Sheaf Neural Networks paper can be found at: https://openreview.net/forum?id=GgcgIJsT8HD

Jakob Hansen: https://www.math.upenn.edu/~jhansen

Thomas Gebhart: https://www.gebhartom.com