Topological and Geometric Deep Learning

Research Project

Student: Nikolay Chechulin, DSBA-201
Academic Supervisor: Oleg Kachan, Research assistant at International Laboratory for Applied Topology and Applications

Data Science and Business Analytics
Faculty of Computer Science

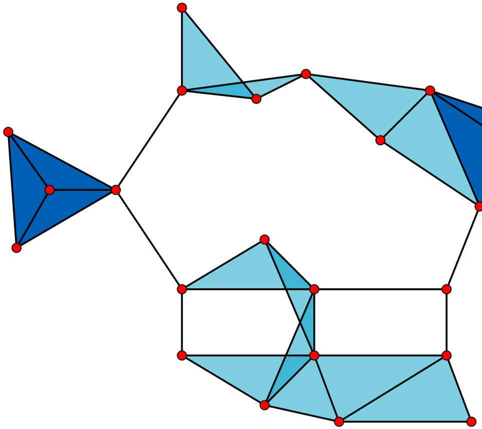
NATIONAL RESEARCH UNIVERSITY

June 5, 2022

Subject Area Overview

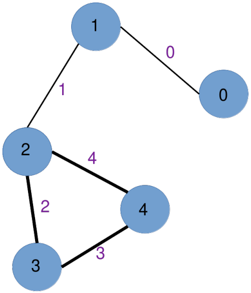
Clique

A graph clique is a subset of its nodes such that it is fully connected. We will work mainly with 3-cliques, and often will refer to them as triangles.



Incidence Matrix

If we have an undirected graph (V, E), its incidence matrix ∇ of size $|V| \times |E|$ such that $A_{i,j} = 1$ if i-th vertex is a vertex of j-th edge. It shows relations between nodes and edges.



$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

Laplacian matrix

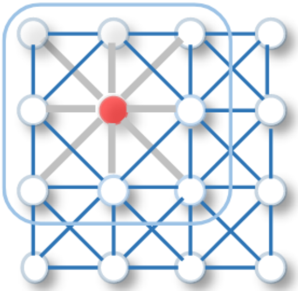
Another matrix representation of a graph. Usually is calculated using the following formula:

$$L_{i,j} = \begin{cases} \deg(v_i) & \text{if } i = j \\ -1 & \text{if } i \neq j \text{ and } v_i \text{ is adjacent to } v_j \\ 0 & \text{otherwise,} \end{cases}$$

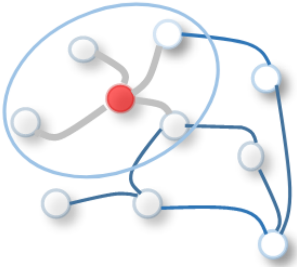
However, other definitions also take place: $L = D - A$, where D is a degree matrix and A is an adjacency matrix. Another way to calculate a Laplacian is $L = \nabla\nabla^T$, where ∇ is an incidence matrix.

Convolutional Graph Network I

A type of GNN which generalizes the convolution operation to graphs.



Convolution on image



Convolution on graph

Convolutional Graph Network II

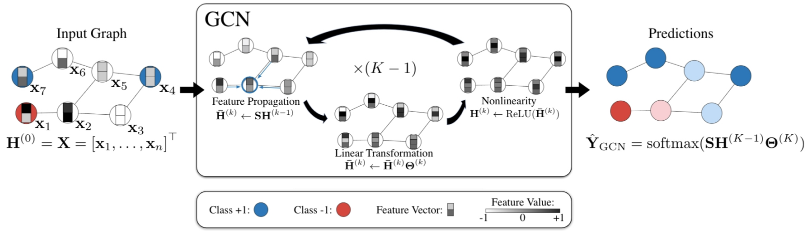
Assume we have a graph of N nodes, where each node has F features. We can construct an $N \times F$ matrix called feature matrix. The first layer takes the feature matrix, and performs the following operation: $Z = D^{-\frac{1}{2}} A D^{-\frac{1}{2}} X W$, where:

- Z is resulting $N \times C$ signal
- D is $N \times N$ degree matrix
- A is $N \times N$ adjacency matrix with self-loops
- X is $N \times F$ feature matrix (input signal)
- W is $F \times C$ learnable weight matrix

The last (output) layer usually applies *softmax* function to each row resulting in a new matrix S. Then, in order to classify a node v_i we simply take the index of maximum of S_i.

Convolutional Graph Network III

The architecture of a graph convolutional network is presented on the figure below.



Input Graph
$\mathbf{x}_7$
$\mathbf{x}_6$
$\mathbf{x}_5$
$\mathbf{x}_4$
$\mathbf{x}_1$
$\mathbf{x}_2$
$\mathbf{x}_3$
$\mathbf{H}^{(0)} = \mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^\top$
GCN
$\times(K-1)$
Feature Propagation
$\bar{\mathbf{H}}^{(k)} \leftarrow \mathbf{S}\mathbf{H}^{(k-1)}$
Linear Transformation
$\bar{\mathbf{H}}^{(k)} \leftarrow \bar{\mathbf{H}}^{(k)}\mathbf{\Theta}^{(k)}$
Nonlinearity
$\mathbf{H}^{(k)} \leftarrow \mathrm{ReLU}(\bar{\mathbf{H}}^{(k)})$
Predictions
$\hat{\mathbf{Y}}_{\mathrm{GCN}} = \mathrm{softmax}(\mathbf{S}\mathbf{H}^{(K-1)}\mathbf{\Theta}^{(K)})$
Class +1:
Class -1:
Feature Vector:
Feature Value:
-1
0
+1

Graph Attention Network I

A type of GNN which uses attention mechanism (also borrowed from ‘casual’ neural networks) which allows us to work with inputs of variable sizes and to focus on the most important features. The attention mechanism is a function $a : \mathbb{R}^C \times \mathbb{R}^C \to \mathbb{R}$ which takes two feature vectors X_i, X_j and returns a scalar representing how tight the connection between v_i and v_j is.

Graph Attention Network II

We introduce an $N \times N$ matrix e storing the attention between the nodes: $e_{i,j} = a\left(W \cdot X_i,\ W \cdot X_j\right)$.
Don't calculate all pairwise attentions! One suggested solution is to use a neighborhood $\mathcal{N}_i$ of a vertex v_i and then compute the attentions between v_i and its' neighbors. Existing models uses neighborhood of size 1, and perform great.
One might also want to normalize the coefficients. In order to do that, we can apply softmax function: $c_{i,j} = \text{softmax}_j\left(e_{i,j}\right)$.

Graph Attention Network III

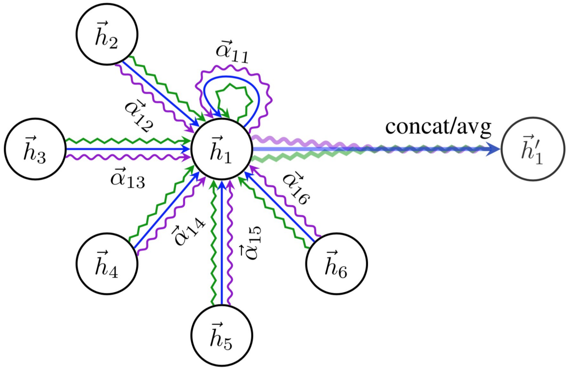


Figure: An example of multi-head attention in a neighborhood of size 1

Tasks solved by GNNs

- Node classification
- Graph classification
- Link prediction

Relevance

- The field is new, therefore, there is a lot of space for improvement
- GNNs allow us to analyze complex relations with great precision
- We can improve the performance by preprocessing the data
- We can improve models by tweaking them

Experiments and Results Overview

Method

Planetoid/Cora dataset:

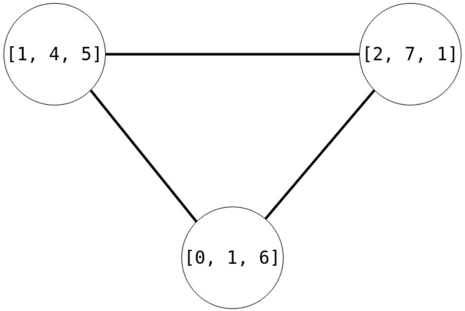
- Node classification problem
- 2708 nodes
- 10556 edges
- 1433 features per node
- 5% training node label rate
- 100 runs with 300 epochs

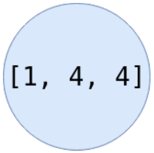
3-Clique merge with insertion Description

1. Find all 3-cliques from a graph and save them in a list
2. Sort them according to sum of pairwise distances
3. Repeat the following steps until there are any nodes in a list:
    - 3.1 Take the 'top' 3-clique
    - 3.2 Save *all* edges coming in/out of the clique
    - 3.3 Compute a 'generic feature vector' by taking average of their feature vectors
    - 3.4 Delete them from the graph
    - 3.5 Create a new 'merged' node with a generic feature vector and all saved edges
    - 3.6 Filter out the deleted nodes from list of 3-cliques

3-clique Merge Example I

The following figures show exactly how a 3-clique is merged: the resulting feature vector is the average of the feature vectors of initial nodes: $\frac{1}{3} \cdot ((1,4,5) + (2,7,1) + (0,1,6)) = \frac{1}{3} \cdot (3,12,12) = (1,4,4)$





3-clique Merge Example II

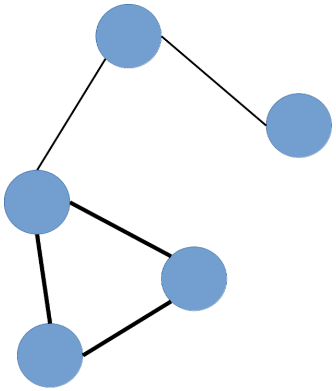


Figure: A part of some graph

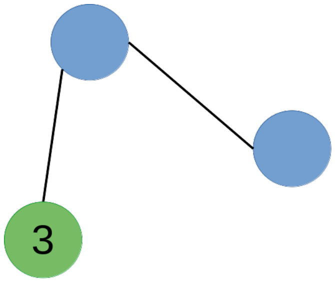


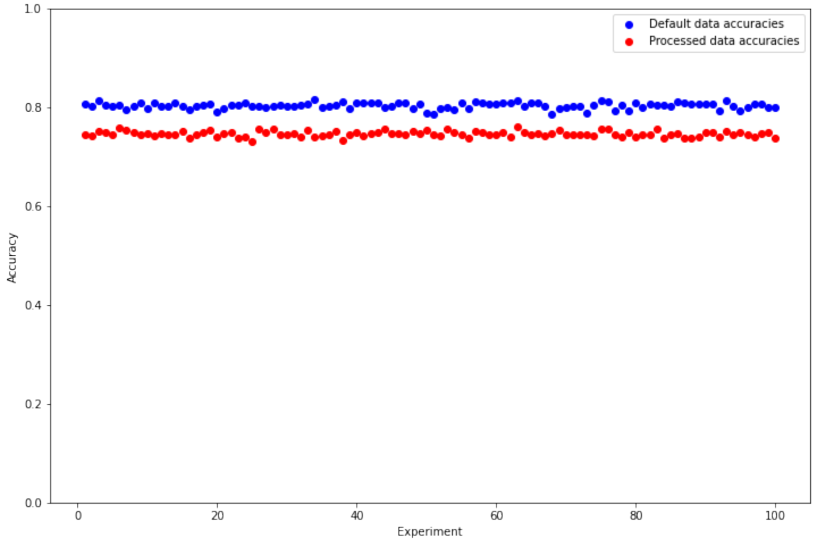
Figure: Three nodes from the left image united in 3-simplex having properties of the initial vertices

Results I

Feature	Before	After	Delta
Node num	2708	2120	21.71%
Edge num	10556	4568	56.73%
Learning time	4.171s	2.939s	29.5%
Accuracy	0.803	0.746	7%

Results II

Accuracies graph:



Infinite 3-clique merge with insertion Description

In this experiment, the methodology and the algorithm are similar to the ‘finite 3-clique merge’. The only difference is that we *allow* merged nodes to participate in merge process again.

Results I

The proposed method allowed us to remove the majority of information from a graph.

Feature	Before	After	Delta
Node num	2708	1080	60.12%
Edge num	10556	656	93.79%
Learning time	2.537s	1.059s	58%
Accuracy	0.809	0.517	36%

Results II

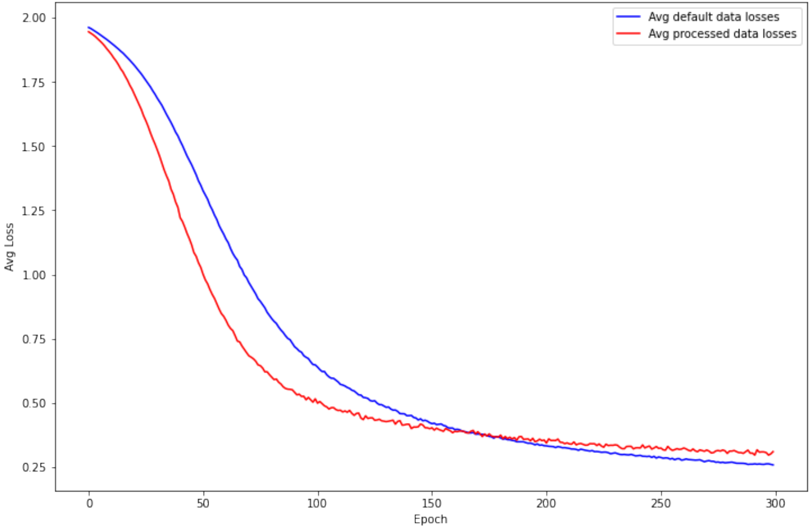


Figure: Average loss for ‘default’ and processed data

Results III

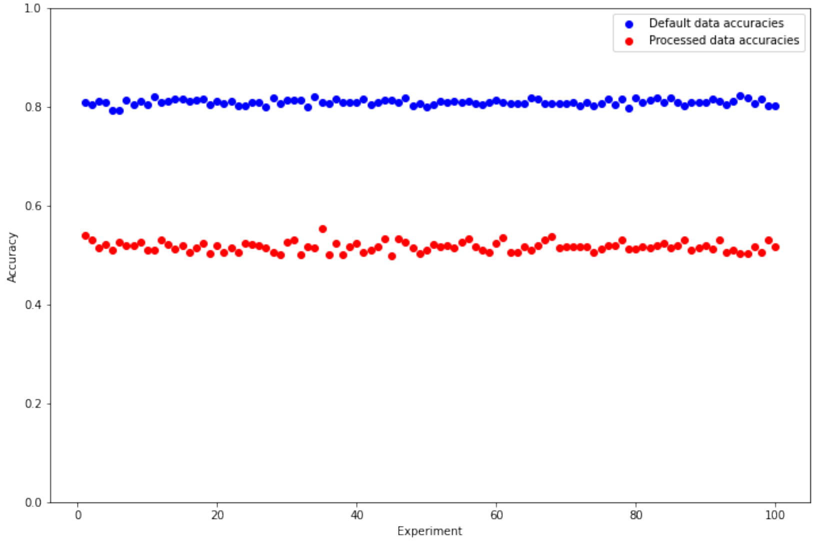


Figure: Accuracies of models with ‘default’ and processed data

Prospects

In future we want to focus on topological aspect of the transformation presented in the paper. In particular, we are interested in ‘infinite merge’, where ‘merged nodes’ can be merged again. We also want to see how the transformation presented can be used in other tasks, for example, in graph classification.

References I

[1] Vijay Prakash Dwivedi et al. "Benchmarking Graph Neural Networks". In: *CoRR* abs/2003.00982 (2020). arXiv: 2003.00982. URL: https://arxiv.org/abs/2003.00982.

[2] Thomas N. Kipf and Max Welling. "Semi-Supervised Classification with Graph Convolutional Networks". In: *arXiv:1609.02907 [cs, stat]* (Feb. 2017). arXiv: 1609.02907. URL: http://arxiv.org/abs/1609.02907.

[3] Yao Ma and Jiliang Tang. *Deep Learning on Graphs*. Cambridge University Press, 2021. DOI: 10.1017/9781108924184.

[4] Petar Veličković et al. "Graph Attention Networks". In: *arXiv:1710.10903 [cs, stat]* (Feb. 2018). arXiv: 1710.10903. URL: http://arxiv.org/abs/1710.10903.

[5] Felix Wu et al. "Simplifying Graph Convolutional Networks". In: *arXiv:1902.07153 [cs, stat]* (June 2019). arXiv: 1902.07153. URL: http://arxiv.org/abs/1902.07153.

References II

[6] Zonghan Wu et al. “A Comprehensive Survey on Graph Neural Networks”. In: *IEEE transactions on neural networks and learning systems* 32.1 (Jan. 2021), pp. 4–24. ISSN: 2162-2388. DOI: 10.1109/TNNLS.2020.2978386.

[7] Keyulu Xu et al. “How Powerful are Graph Neural Networks?” In: *CoRR* abs/1810.00826 (2018). arXiv: 1810.00826. URL: http://arxiv.org/abs/1810.00826.

Any questions?