Author Collaboration Prediction for OGBL-COLLAB

Presented by Group 3

November 11, 2022

Team

18CS30018 Gajula Sai Chaitanya

18AE30001 Bale Veeresh Siva Sai

18AE30022 Eate Sohil

18AE30028 Yegurula Sumanth

Introduction

Problem Statement

- The ogbl-collab dataset is an undirected graph, representing a subset of the collaboration network between authors indexed by MAG.
- Each node represents an author and edges indicate the collaboration between authors. The task is to predict the future author collaboration relationships given the past collaborations.
- The goal is to rank true collaborations higher than false collaborations.

Importance Of the Problem Statement

- The problem statement may be interesting particularly for the researchers who may be interested to know future collaborations among various authors which may be useful analyse successful collaborations.

- For example, consider a scenario of a research institute, knowing probable collaborations between two authors helps to decide whether to fund a particular research facility.

Dataset Splitting

- The dataset is already provided with necessary train, valid and test splits. They split the data according to time, in order to simulate a realistic application in collaboration recommendation.

- Specifically, they put the collaborations until 2017 as training edges, those in 2018 as validation edges, and those in 2019 as test edges. We are using both train and valid edges while evaluating the test set (transductive setting).

Model Architecture

Model Architecture

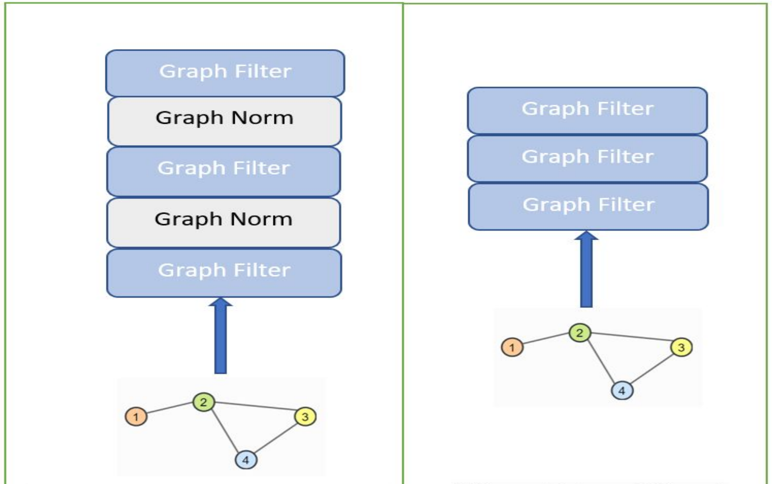
- The architecture of our model mainly contains two modules.

- Module 1 for learning node embeddings or node representations and module 2 for predicting link between two different nodes.

- These two modules are trained end-end to learn the parameters of the model.

Model Architecture

- First is a 3-layer Graph Neural Network stack. For different experiments, we considered either **SAGEConv** or **GCNConv** as graph filters to learn the node features.
- For the first module, we also considered using **GraphNorm** normalisation, which applies graph normalisation over graphs.
- We are stacking graphNorm layers similar to how we arrange batchnorm layers in CNN.
- The second module is a link prediction module which is a normal 3-layer Dense Neural Network.



a) CNN-style 3-layer GNN with GraphNorm

b) Normal 3-layer GNN stack

Model Architecture

- We have trained four different models to see whether graph normalization improves the performance of the models.
- 1) SAGEConv

  2) SAGEConv +GraphNorm

  3) GCNConv

  4) GCNConv + GraphNorm
- We are also using negative sampling while calculating the loss function. For evaluating performance on test split, we are considering both train and val split links (transductive setting).

Evaluation Metric

- **Hits@50:** as the evaluation metric, which is defined as the fraction of positives that rank in the top 50 among their negatives (higher is better, best is 1).

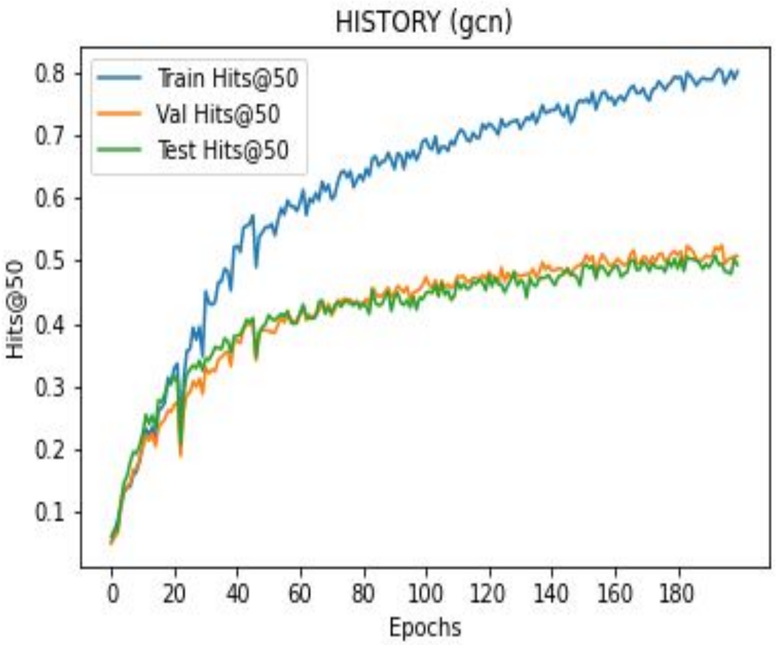
- For example, let’s assume that, in the top 50, there are 30 positives and 20 negatives, then the Hits@50 for this situation is 30/50 = 0.6

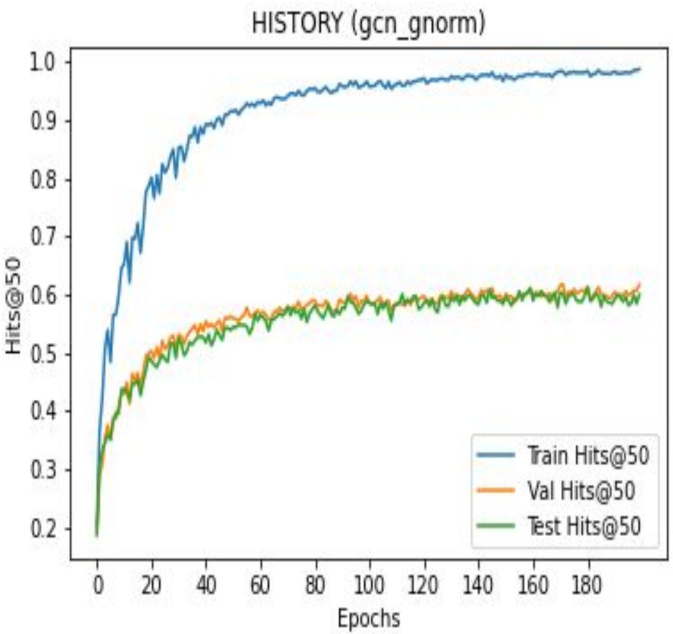
Results and Analysis

Performance

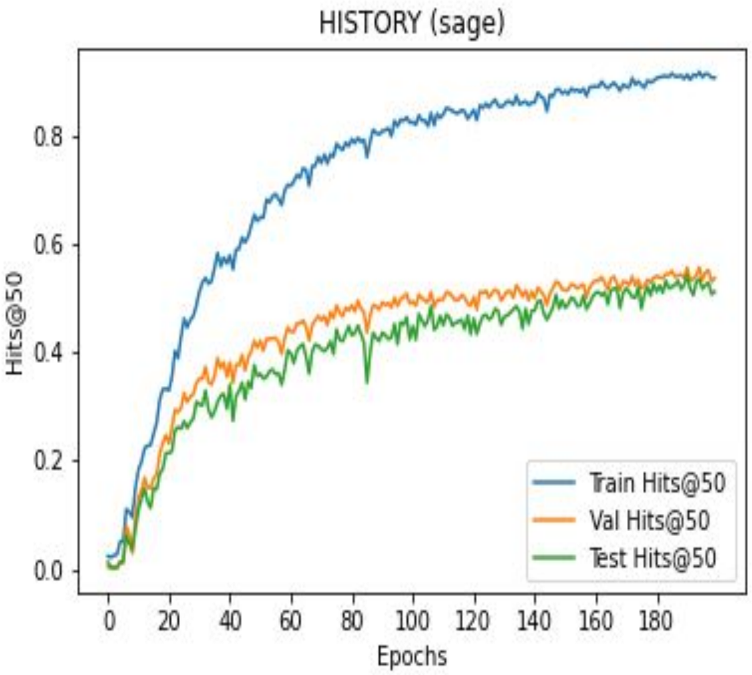
Setup	Val Hits@50	Test Hits@50
SAGEConv	0.5365	0.5113
SAGEConv + GraphNorm	0.6091	0.5935
GCNConv	0.5073	0.4935
GCNConv + GraphNorm	**0.6158**	**0.5991**

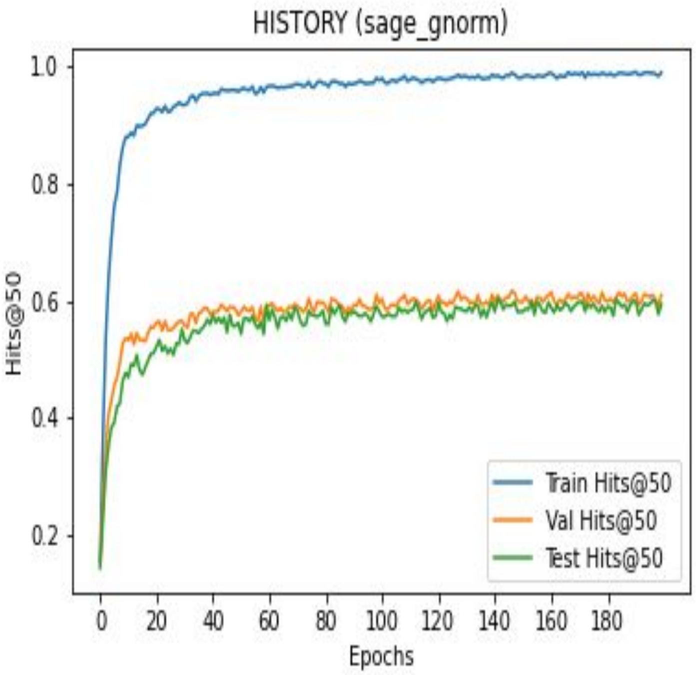
Training Graphs





Training Graphs





Conclusion

- From the above results and observations, we can conclude that for our setting, GCNConv with GraphNorm performs the best.

- On comparing our results with that in the ogbl-collab [leaderboard](), this approach stands somewhere between 10 to 12.

10	Common Neighbor	No	0.6137 ± 0.0000	0.6036 ± 0.0000
11	**SEAL-nofeat**	No	0.5471 ± 0.0049	0.6495 ± 0.0043
12	GraphSAGE (val as input)	No	0.5463 ± 0.0112	0.5688 ± 0.0077

THANK YOU