GATTO: Can Topological Information Improve Node Classification via GAT?

Final Report for Learning from Network's project

Francesco Biscaccia Carrara *(2120934)*, Riccardo Modolo *(2123750)*,
Alessandro Viespoli *(2120824)*

Master Degree in Computer Engineering
University of Padova

I. ABSTRACT

In this report we study how topological features of a node can affect its prediction on GAT (Graph Attention Network). We introduce the dataset, the type of feature we computed and we compare our result with the original GAT paper. We also use a graph with ground truth but without freatures to observe how only topological feature produce prediction.

II. INTRODUCTION

Node classification is an important reasearch and business topic. Our framework GATTO (Graph ATtention network with TOpological information) want to improve the classic GAT prediction using node feature coming from the graph or from its embedding. The main idea is to using topological features to imporve prediction of GAT, where each node already have features in it. Another possible use is when node didn't have any features, so we compute and use them for GAT train. In this paper we explore both scenarios.

III. DATA

We're going to use the same dataset of GAT[1], with features to each node, and one Graph with ground truth from SNAP [2] (without features). the Dataset of GAT are: *Cora, Citeseer and Pubmed*. For SNAP, we have *email-Eu-core*. Each node have only one label and the graph is mix directed/undirected

Network	**Nodes**	**Edges**	**labels**	**features**
email-EU-core	1005	25571	42	0
Cora	2708	5429	7	1443
Citeseer	3327	4732	6	3703
Pubmed	19717	44338	3	500

The feature we intended to compute and assign to each node for all these graphs are:

- degree centrality
- betweenness centrality
- closeness centrality
- suggested label

The *suggested label* parameter is the result of a clustering made on the embedding of the Graph. We use Node2Vec[3] to produce the embedding, and for clustering we use k-mean++ method.

IV. IMPLEMENTATION

The practical implementation[4] respect the nature of the the framework concept. We have two block:

- **Precomputation Module**: the class that compute every needed or requested features from the graph or from it's embedding, and return it as feature matrix
- **GAT Module**: the GAT implementation for train and predict node labels

V. TESTS

VI. RESULTS

WORK REPORT

In this section we want to describe the distribution of the work and detailed contribution of each member.

- *Francesco Biscaccia Carrara (2120934)*: **33.3% of the work**. Produce the code compute features and Runnig test on CAPRI
- *Alessandro Viespoli (2120824)*: **33.3% of the work**. Produce the code for the GAT and the code for plotting results
- *Riccardo Modolo (2123750)*: **33.3% of the work**. Produce the code to retrieve data, review code, write Proposal, Midterm and final Paper

REFERENCES

[1] Petar Veličković et al. *Graph Attention Networks*. 2018. arXiv: 1710.10903 [stat.ML]. URL: https://arxiv.org/abs/1710.10903.

[2] Jure Leskovec and Andrej Krevl. *SNAP Datasets: Stanford Large Network Dataset Collection*. http://snap.stanford.edu/data. June 2014.

[3] Aditya Grover and Jure Leskovec. *node2vec: Scalable Feature Learning for Networks*. 2016. arXiv: 1607.00653 [cs.SI]. URL: https://arxiv.org/abs/1607.00653.

[4] *scikit-learn: GitHub Implementation*. URL: https://github.com/scikit-learn/scikit-learn.