GATto: topological information to improve node classification via GAT

Proposal for Learning from Network's project

Francesco Biscaccia Carrara, Alessandro Viespoli, Riccardo Modolo

Master Degree in Computer Engineering
University of Padova

I. MOTIVATION

Node classification is an important topic of graph analysis for find useful informations.

Quality of classification is an hot topic, since better classification present an advantage on extract conclusion or correlation in a graph.

Our intention is try to improve node classification using some precomputed features obtained via graph embedding and clustering. To evaluate our hypothesis we use social graph with ground truth. This type of graph help us to examine any difference between simple GAT and our GATto (GAT with topological features on each node) and check which approximate better the ground truth.

Time permitting and iif the method seem to work well, we put more effort in parllelization and scaling the framework to use it in a distributed environment.

II. METHODS

We divide our workload in two main components

- **Precomputation module**: the part of code used to precompute additional features
- **GAT module**: the effective computation of node classification using a Graph Attention Network

In the precomputation module we want to test different embedding library (node2vec or GraphWave) and different clustering parameters as features.

In our expectation we use the majority work effort in build the Precomputation module and conduct tests, since we want to use and already implemented GAT library. We want also to be sure that our additional feature don't introduce any bias*** on the GAT network.

III. INTENDED EXPERIMENTS

We want to implement the methods in Python for the advantage of using deep learning library like tensorflow. The code and all implementation details will be available on **GitHub**.

The intentional experiments approximately work as follow:

- train GAT with default node features
- compute the classification error
- compute graph embedding on the graph
- compute clustering and add features to nodes
- train GAT with additional features
- compute the classification error

All tests were performed on the **CAPRI** High-Performance Computing (HPC) system, owned by the University of Padova. CAPRI is designed to provide computational power for testing innovative algorithms across various research fields. It is equipped with the following hardware:

- 16 Intel(R) Xeon(R) Gold 6130 @ 2.10GHz CPUs
- 6 TB of RAM
- 2 NVIDIA Tesla P100 16GB GPUs
- 40 TB of disk space

The objective of these tests is to observe if this precomputation provide any improvement on classification.