GATTO: Can Topological Information Improve Node Classification via GAT?

Proposal for Learning from Network's project

Francesco Biscaccia Carrara, Riccardo Modolo, Alessandro Viespoli

Master Degree in Computer Engineering
University of Padova

I. MOTIVATION

Node classification is an important topic in graph analysis for assigning to each node a label from a set of predefined classes.

Our intention is to see whether pre-computing features obtained via graph embedding and clustering can improve node classification via graph attention networks (GAT)[GAT].

II. DATA

The datasets we will be using are taken from Stanford Network Analysis Project (SNAP)[SNAP]:

Network	Nodes	Edges	Communities
email-EU-core	1005	25571	42
com-Amazon	334863	925872	75149
wiki-topcats	1791489	28511807	17364

III. METHODS

In order to evaluate our hypothesis we will use social graph with ground truth, in this way we will examine any difference between simple GAT classification and our GATTO (GAT with topological features on each node) structure. Our hope is to improve classification performances with simple and fast precomputation on the graph to gather useful features which will be used by the GAT.

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 GAT.

In the precomputation module we want to test an embedding library (i.e. node2vec[node2vec]) and then from the embedding calculate different clustering parameters like:

- the cluster of the node
- its nearest cluster
- the distance of a node to the center of its cluster
- the distance of a node to its nearest cluster's center

IV. INTENDED EXPERIMENTS

We want to implement the overall architecture in Python, leveraging the available algorithms' implementation:

1) *node2vec*[n2vGitHub] for node embedding task;
2) *scikit-learn*[scikitGitHub] for clustering task;
3) *spektral*[spektralGitHub] for GAT implementation.

The intentional experiments will be approximately:

- train GAT with default node features in the dataset.
- compute the classification error.
- compute additional graph features by using algorithms within the precomputation module.
- train GAT with the new additional features.
- compute the classification error.
- conduct analytical tests.

The code and all implementation details will be available on GitHub.

All tests will be performed on the **CAPRI**[CAPRI] High-Performance Computing (HPC) system, owned by the University of Padova. 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

Time permitting and if and only if the method seems to achieve some results, we will put more effort in parallelization and scaling the framework to a distributed environment.