[Experiment] Learning Community Embedding with Community Detection and Node Embedding on Graph (CD)

Anton Begehr
University of Passau
Passau, Germany
a.begehr@fu-berlin.de

ABSTRACT

A clear and well-documented LaTeX document is presented as an article formatted for publication by ACM in a conference proceedings or journal publication. Based on the "acmart" document class, this article presents and explains many of the common variations, as well as many of the formatting elements an author may use in the preparation of the documentation of their work.

ACM Reference Format:
Anton Begehr. 2019. [Experiment] Learning Community Embedding with Community Detection and Node Embedding on Graph (CD). In *Proceedings of Data Science Seminar*. ACM, New York, NY, USA, 1 page.

1 INTRODUCTION

In their 2017 paper *Learning Community Embedding with Community Detection and Node Embedding on Graph* the authors Cavallari et al. explore graph embeddings by utilizing a three step closed loop optimization process consisting of Community Detection, Community Embedding, and Node Embedding: ComE.[1] They then apply ComE and further graph embedding algorithms on excerpts of multiple, well known graph datasets: Karate Club, BlogCatalog, Flicker, Wikipedia, DBLP. They choose DeepWalk/SF, Line, Node2Vec, GraRep, and M-NMF to measure the quality of ComE's results using Micro-F1 and Macro-F1 for classification results and conductance and normalized mutual information (NMI) for the resulting graph embeddings. It is worthwhile to note, that ComE works with elementary graphs, meaning graphs consisting of nodes and edges, disregarding, for example, node properties, edge properties, and edge weights.

In this paper we will explore the graph embedding algorithm ComE developed by Cavallari et al., generate embeddings using ComE for the Twitter dataset [2] crawled by Rizi, and compare the results to communities gained by applying Louvain Modularity using normalized mutual information (NMI) as a comparison measure.

2 TWITTER DATA

2.1 ComE Data Requirements

The ComE algorithm requires the graph data to be supplied as a sparse adjacency matrix. The example code uses a MATLAB .mat file import. The data supplied by Rizi is in an edge list format inside a CSV file. Each row represents an edge from one node to the other.

ComE also requires the nodes to be labeled for knowing the number K of clusters it should optimize for and testing the resulting clusters against the supplied ground truth.

2.2 Data Preparation

These two requirements ComE has

REFERENCES

[1] Sandro Cavallari, Vincent W. Zheng, Hongyun Cai, Kevin Chen-Chuan Chang, and Erik Cambria. 2017. Learning Community Embedding with Community Detection and Node Embedding on Graphs. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (CIKM '17)*. ACM, New York, NY, USA, 377–386. https://doi.org/10.1145/3132847.3132925
[2] Fatemeh Salehi Rizi. [n.d.]. fatemehsrz/Twitter_Data. data retrieved from GitHub on 2019-06-17, https://github.com/fatemehsrz/Twitter_Data.

ACKNOWLEDGMENTS

To fill.

A CODE

A.1 Generate graph

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi malesuada, quam in pulvinar varius, metus nunc fermentum urna, id sollicitudin purus odio sit amet enim. Aliquam ullamcorper eu ipsum vel mollis. Curabitur quis dictum nisl. Phasellus vel semper risus, et lacinia dolor. Integer ultricies commodo sem nec semper.

A.2 Generate Labels

Etiam commodo feugiat nisl pulvinar pellentesque. Etiam auctor sodales ligula, non varius nibh pulvinar semper. Suspendisse nec lectus non ipsum convallis congue hendrerit vitae sapien. Donec at laoreet eros. Vivamus non purus placerat, scelerisque diam eu, cursus ante. Etiam aliquam tortor auctor efficitur mattis.

A.3 Read CSVs

Nam id fermentum dui. Suspendisse sagittis tortor a nulla mollis, in pulvinar ex pretium. Sed interdum orci quis metus euismod, et sagittis enim maximus. Vestibulum gravida massa ut felis suscipit congue. Quisque mattis elit a risus ultrices commodo venenatis eget dui. Etiam sagittis eleifend elementum.

Nam interdum magna at lectus dignissim, ac dignissim lorem rhoncus. Maecenas eu arcu ac neque placerat aliquam. Nunc pulvinar massa et mattis lacinia.