PIETRO VOLPATO, FILIPPO D'EMILIO

Title: Evaluation of different Node Embedding algorithms

Motivation:

- Our intention is to test different node embedding algorithms in order to test their performances on different datasets. In particular the goal is to calculate different metrics for different graphs based on different features and compare them to see what algorithm gives the better performance.
- Data: for testing the algorithms we will use different datasets. Our intention is to use datasets with different sizes for testing the scalability of the algorithms to different graph sizes. We still have to decide the suitable dataset to use but this is a list of possible graphs of different types:
    1. Product co-purchasing networks:
    2. Social networks
    3. Web graphs
    4. Cora dataset
- Other dataset will be added in the mid term proposal with a detailed description of the network.

Method:

- Problem: find the performance of several embedding algorithms, two of them based on deep learning techniques and assess what are the different results of these methods.
- Algorithm: we want to present four algorithms, all based on different techniques and already present in the literature:
    1. GraphWave: this algorithm has been proposed in 2018 by some researchers at the University of Stanford. The detailed description of the algorithm can be found at this link. It is an unsupervised method for learning node embeddings based on structural similarity in networks and it uses spectral graph wavelets as a distribution probability. The code is available here.
    2. Node2vec: the algorithm learns a mapping of nodes to a low-dimensional space of features that maximizes the likelihood of preserving network neighborhoods of nodes. The code is available here.
    3. DeepWalk: it uses local information obtained from truncated random walks to learn latent representations by treating walks as the equivalent of sentences
    4. LINE (Large-scale Information Network Embedding): this method optimizes a carefully designed objective function that preserves both the local and global network structures. An edge-sampling algorithm is proposed that addresses the limitation of the classical stochastic gradient descent and improves both the effectiveness and the efficiency of the inference.Link to the code.

5. Spectral Embedding: it is a method already implemented in the Networkx library and it is based on the spectral decomposition of the Laplacian matrix associated with the graph. Link to the documentation.
6. Graph Auto Encoder: this algorithm uses Variational Autoencoder. All the details can be found in the paper .Link to the code
7. CapsGNN: it introduces the concept of capsule network to the graph. By extracting node features in the form of capsules, routing mechanism can be utilized to capture important information at the graph level. As a result, the model generates multiple embeddings for each graph to capture graph properties from different aspects. The attention module is used to tackle graphs with various sizes which also enables the model to focus on critical parts of the graphs. Link to the article here, to the code here.
8. AttentionWalk: this algorithm is based on an attention model. A detailed paper can be found here. The implementation here
9. GraphSAGE: a general, inductive framework that leverages node feature information to efficiently generate node embeddings for previously unseen data. Instead of training individual embeddings for each node, the model learns a function that generates embeddings by sampling and aggregating features from a node's local neighborhood. The paper is available here and the implementation here.

- This above is a general list of algorithms that we find interesting for network embedding, however we will choose at least 4 algorithms for our final project and in the mid term proposal we will list the algorithms we have chosen with the explanation for our choices.
- As regards the metrics for the evaluation of the performance of the algorithms, we have chosen:
    1. Link prediction: test if the embeddings can predict if there is an edge between two nodes
    2. Node classification.
    3. Node clustering: evaluate if the embeddings can represent the clustering of the nodes present in the original graph
    4. Preservation of the topology: measure how much the embeddings reflect the structure of the original graph(reconstruction error, neighborhood preservation).
    5. Distance preservation: compute the distances between nodes in the original graph(shortest path) and in the embedding

Intended experiments:

- For our experiments we will use the implementation available on GitHub and in particular the one that we have presented in the link attached to the algorithm presentation.
- machines for experiments: mainly Google Colab for the deep learning approaches and our PCs for the others.

References

- Donnat, Zitnik, Hallac, Leskovec, (2018, July), Learning structural node embeddings via diffusion wavelets. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*
- Grover A., Leskovec J., (2016, August), node2vec: Scalable Feature Learning for Networks. In *Proceedings of the 22th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*
- Perozzi B., Al-Rfou R., Skiena S., (2014, August), DeepWalk: online learning for social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*
- Tang J., Qu M., Wang M., Zhang M., Yan J., Mei Q., (2015, May), LINE: Large-scale Information Network Embedding. In *Proceedings of the 24th International Conference on World Wide Web*
- Kipf T. N., Welling M., (2016, November), Variational Auto-Encoders. In *NIPS Workshop on Bayesian Deep Learning.*
- Abu-El-Haija S., Perozzi B., Al-Rfou R., Alemi A., (2018, September), Watch Your Step: Learning Node Embeddings via Graph Attention. In *32nd Conference on Neural Information Processing Systems.*
- Hamilton W. L., Ying R., Leskovec J., (2017, June), Inductive Representation Learning on Large Graphs. In *NIPS 2017*.
- Zhang X., Lihui C., (2018, September), Capsule Graph Neural Network. In *ICLR 2019 Conference*

Contributions

Pietro Volpato: 50%, Filippo D'Emilio: 50%