Representation Learning on Dynamic Graphs

Erfan Loghmani

Sharif University of Technology

Abstract

Graphs are a common language in modeling several problems, from social and economic networks to interactions in cells and brain neurons.
According to the availability of an enormous amount of data from graphs, Machine Learning algorithms gained lots of attention in this area. But the main challenge is how to represent and encode nodes so that such models could interpret the data and the representation preserve the main features of nodes.
There are some methods for representation learning on static graphs, but in most of the real problems our graph evolves in time and we have both arrival/deletion of nodes and edges as well as interactions between nodes. The main goals of an online representation learning method is to save time and computation and avoid to run the method for entire graph in each time-step and also could adjust the embedding of nodes smoothly in time.

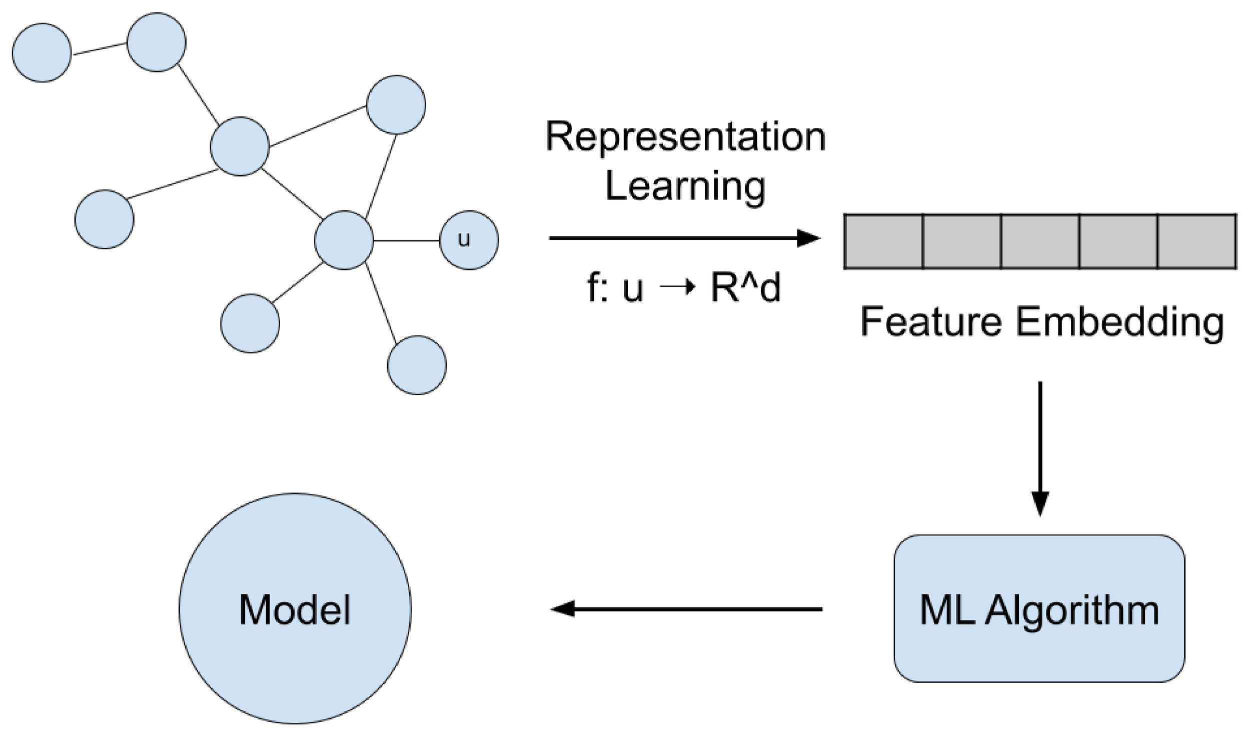


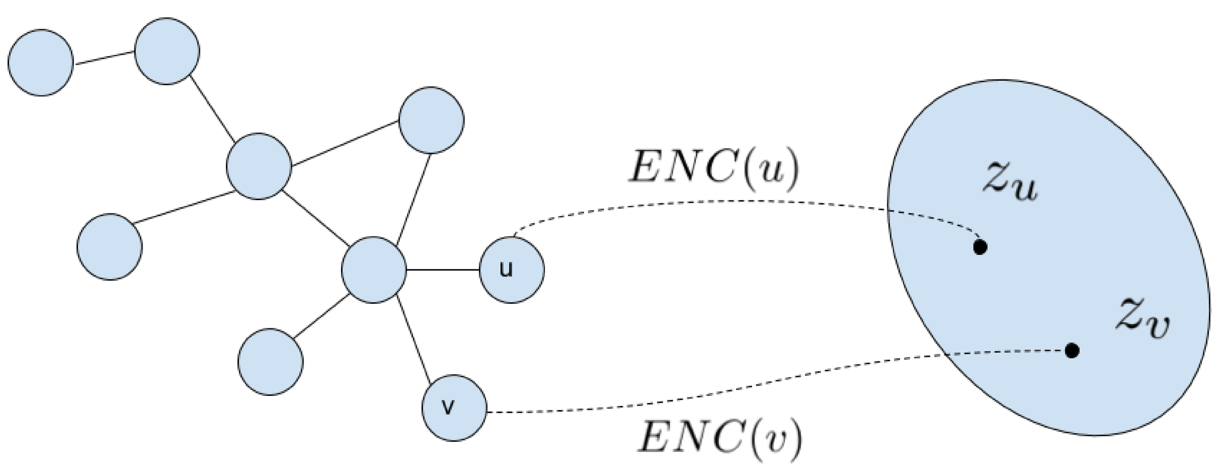
Figure 1: Representation learning position in machine learning pipeline.

Methods on Static Graphs [1]

1. Encoder-Decoder Approach

This approach tries to encode nodes to a vector space so that close nodes in the graph be also close in the vector space.
For this approach we should define these four components:

- A similarity function between nodes on graph, e.g. number of common neighbors of two nodes
- An encoder function, ENC that generates the representations for each node in our embedding space
- A decoder function, DEC that gives us the similarity of two embedded nodes
- A loss function that we want to minimize.



$$Goal:\ similarity(u,v) \approx DEC(z_u, z_v) = z_u^T z_v$$

Figure 2: Encoder Decoder Approach

2. Graph Neural Networks

The general neural network approach on graphs is to have different computation graphs for each node, and in each layer aggregate, each node neighbors information and pass through one neuron.
This approach has some advantages over the encoder-decoder approach:

- Could directly be used for our learning problem
- We can use other features we have on nodes
- No need to know all of the node labels

There are different methods based on this idea that currently are being used like:

- Graph Convolutional Networks
- GraphSAGE
- Gated Graph Neural Networks
- Graph Attention Networks

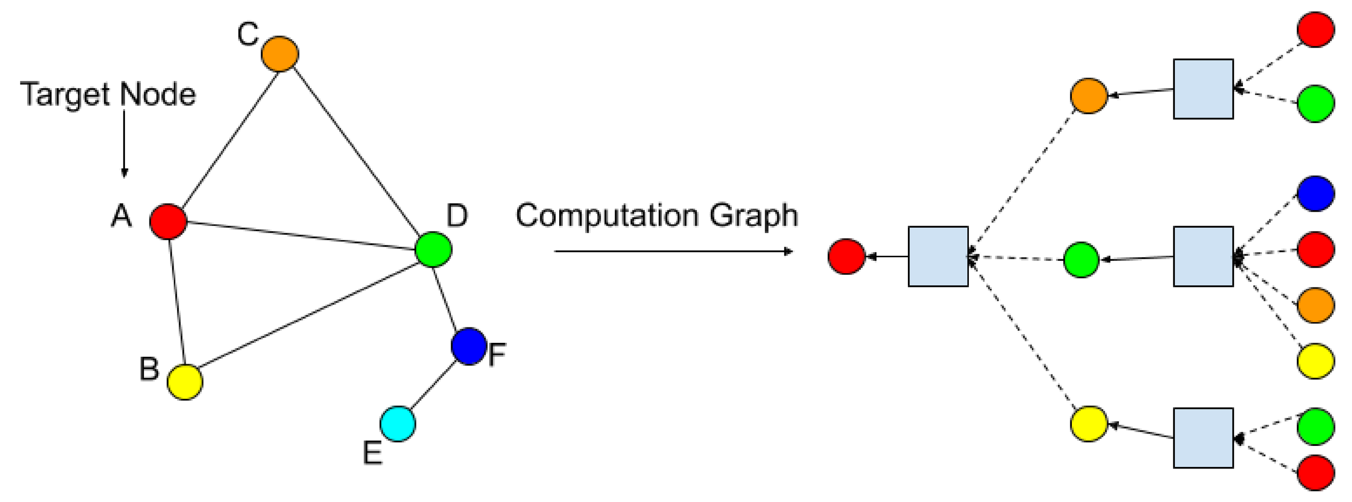


Figure 3: General neural network schema

Methods on Dynamic Graphs

There are some works on this problem:

- DyRep [5] uses point processes to model interaction between nodes.
- Sankar et. al. [3] use self-attention networks
- Taheri et. al. [4] use recurrent models
- Kumar et. al. [2] use both RNN and attention mechanism on user-item interaction graph

Our Direction

These methods are based on only graphs and see edge additions or interactions between nodes, but can't encounter interaction features like the text communicated or the text posted by a user. We think that these features also have information about the representation of each node and could help tasks like edge prediction.

References

[1] W. L. Hamilton, R. Ying, and J. Leskovec. Representation learning on graphs: Methods and applications. *CoRR*, abs/1709.05584, 2017. URL `http://arxiv.org/abs/1709.05584`.

[2] S. Kumar, X. Zhang, and J. Leskovec. Predicting dynamic embedding trajectory in temporal interaction networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1269–1278. ACM, 2019.

[3] A. Sankar, Y. Wu, L. Gou, W. Zhang, and H. Yang. Dynamic graph representation learning via self-attention networks. *CoRR*, abs/1812.09430, 2018. URL `http://arxiv.org/abs/1812.09430`.

[4] A. Taheri, K. Gimpel, and T. Y. Berger-Wolf. Learning to represent the evolution of dynamic graphs with recurrent models. In *WWW*, 2019.

[5] R. Trivedi, M. Farajtabar, P. Biswal, and H. Zha. Representation learning over dynamic graphs. *CoRR*, abs/1803.04051, 2018. URL `http://arxiv.org/abs/1803.04051`.

Contact Information

- erfan.loghmani@gmail.com - loghmani@ce.sharif.edu
- www.ce.sharif.edu/~loghmani