Precog Recruitment Task
node2vec: Scalable Feature Learning for Networks

Chetan Mahipal

Author of Paper - Aditya Grover, Jure Leskovec

IIIT Hyderabad - February 14, 2024

"Feature learning is the cornerstone of modern machine learning, enabling systems to autonomously extract meaningful representations from raw data." - Anonymous.

1 Summarizing the Research

1.1 Introduction

1.1.1 Use of Feature Learning

Many important tasks in network analysis involve predictions over nodes and edges. In a typical node classification task, we are interested in predicting the most probable labels of nodes in a network. For example, in a social network, we might be interested in predicting interests of users, or in a protein-protein interaction network we might be interested in predicting functional labels of proteins. Similarly, in link prediction, we wish to predict whether a pair of nodes in a network should have an edge connecting them . Link prediction is useful in a wide variety of domains; for instance, in genomics, it helps us discover novel interactions between genes, and in social networks, it can identify real-world friends.

1.1.2 Challenges with Classical Stratigies

Classical strategies like Breadth-First Search (BFS) and Depth-First Search (DFS) may struggle to capture and represent two key notions of similarity in networks: homophily and structural equivalence. While BFS and DFS explore networks based on connectivity, they may overlook node attributes important for homophily. Additionally, they may not effectively capture structural roles crucial for structural equivalence. As networks often exhibit both behaviors simultaneously, these strategies may not fully capture the nuanced similarities present in real-world networks.

1.1.3 Introduction of node2vec

node2vec, a semi-supervised algorithm for scalable feature learning in networks. We optimize a custom graph-based objective function using SGD motivated by prior work on natural language processing. Intuitively, our approach returns feature representations that maximize the likelihood of preserving network neighborhoods of nodes in a d-dimensional feature space. We use a 2nd order random walk approach to generate (sample) network neighborhoods for nodes. node2vec can learn

representations that organize nodes based on their network roles and/or communities they belong to. We achieve this by developing a family of biased random walks, which efficiently explore diverse neighborhoods of a given node.

1.2 Methodology

1.2.1 Biased Random Walk

Formally, given a source node u, we simulate a random walk of fixed length l. Let c_i denote the ith node in the walk, starting with $c_0 = u$. Nodes c_i are generated by the following distribution:

$$P(c_i = x \mid c_{i-1} = v) = \begin{cases} \frac{\pi_{vx}}{Z} & \text{if } (v, x) \in E \\ 0 & \text{otherwise} \end{cases}$$

where π_{vx} is the unnormalized transition probability between nodes v and x, and Z is the normalizing constant.

1.2.2 Search bias α

We set the unnormalized transition probability to $\pi_{vx} = \alpha_{pq}(t, x) \cdot w_{vx}$, where

$$\alpha_{pq}(t, x) = \begin{cases} \frac{1}{p} & \text{if } d_{tx} = 0 \\ 1 & \text{if } d_{tx} = 1 \\ \frac{1}{q} & \text{if } d_{tx} = 2 \end{cases}$$

and d_{tx} denotes the shortest distance between nodes t and x.

1.2.3 Parameters

1. **Return Parameter** p: This parameter controls the likelihood. High value of p leads to aversion of 2-hop as tendency to revisit sampled node decreases. Low value of p leads to backtrack the step and would keep the walk to local to the node (in neighbours of nodes).

2. **In-out Parameter** q: This parameter allows to differentiate searches. If $q > 1$, the random walk tends to show BFS behaviour. In contrast, if $q < 1$, the walk is more inclined to farther nodes, showing DFS behaviour.

1.3 Outcomes of node2vec

1.3.1 Performance

1. **Multi-Label Classification**: Networks in which we have a mix of equivalences present, the semi supervised nature of node2vec can help us infer the appropriate degree of exploration necessary for feature learning. In the case of PPI network, the best exploration strategy is node2vec(p = 4, q = 1).

2. **Link Prediction**: A general observation we can draw from the results is that the learned feature representations for node pairs significantly outperform the heuristic benchmark scores with node2vec achieving the best AUC improvement on 12.6% on the arXiv dataset over the best performing baseline.

1.3.2 Future Extensions

1. Using node2vec on networks with special structure such as heterogeneous information networks.
2. Applying node2vec on networks with explicit domain features for nodes and edges and signed-edge networks.
3. Yes node2vec representations as building blocks for end-to-end deep learning on graphs.

2 Strength of The Research Paper

2.1 Scalability

One of the key strengths of node2vec is its scalability. It efficiently learns embeddings for large-scale networks by employing a second order biased random walk strategy. Random walks are computationally efficient in terms of both space and time requirements.

1. **Space Complexity**: The space complexity to store the immediate neighbors of every node in the graph is $O(|E|)$. For 2^{nd} order random walks, it is helpful to store the interconnections between the neighbors of every node, which incurs a space complexity of $O(a^2|V|)$ where a is the average degree of the graph and is usually small for real-world networks.
2. **Time Complexity**:They provide a convenient mechanism to increase the effective sampling rate by reusing samples across different source nodes.By simulating a random walk of length $l > k$, we can generate k samples for $l - k$ nodes at once due to the Markovian nature of the random walk. Hence, our effective complexity is $O\left(\frac{l}{k(l-k)}\right)$ per sample.

2.2 Flexibility

The parameters p & q allows our search procedure to interpolate between BFS and DFS which gives us liberty of exploring different node equivalences (mainly homophily and structural equivalence).

1. **Return Parameter** p: This parameter controls the likelihood. High value of p leads to aversion of 2-hop as tendency to revisit sampled node decreases. Low value of p leads to backtrack the step and would keep the walk to local to the node (in neighbours of nodes).
2. **In-out Parameter** q: This parameter allows to differentiate searches. If $q > 1$, the random walk tends to show BFS behaviour. In contrast, if $q < 1$, the walk is more inclined to farther nodes, showing DFS behaviour.

2.3 Performance

A general observation can be drawn from the results given in the paper is that the learned feature representations for node pairs significantly outperform the heuristic benchmark scores with node2vec. The paper shows how extensions of node embeddings to link prediction can give robust result, especially important in cases where the graphs are evolving over time.

3 Weakness of Research Paper

3.1 Parameter Tuning

Setting the parameters p and q appropriately can significantly impact the quality of the learned

embeddings, and determining the optimal values might require experimentation. The performance of node2vec improves as the in-out parameter q and the return parameter p decrease, but if the value of parameters are not chosen wisely the results may be biased and leads to bad results.

3.2 Limited Theoretical Justification

I felt the paper lacked theoretical and mathematical proofs and results. I felt the need for mathematical reasoning the selection of 2 nd Order random walk. For optimization, a brief comparison of results with different techniques negative sampling, SGD etc. and a overview of these algorithms and steps at which they are implied.

3.3 Computation Overhead

Lack of detailed discussion or exploration of the computational overhead associated with the method. Setting the search parameters based on the underlying task and domain at no additional cost, learning the best settings of our search parameters adds an overhead. These overheads will give a bigger overview how the model performs as network expands and becomes dense which will help in practical use of algorithm.

Note: Found some typing errors in the paper which leads to abuse of notation.

4 Improvements to Paper

4.1 Automated Parameter Tuning

Developing automated techniques or algorithms for tuning the parameters p and q in node2vec. This could involve leveraging machine learning approaches or optimization algorithms to find optimal parameter settings based on the characteristics of the input network. Providing a systematic way to determine these parameters could simplify the application of node2vec and improve its effectiveness across a wide range of networks.

4.2 Theoretical Analysis

Provide more rigorous proofs for important choices like 2^{nd} Order random walk or using SGD (Stochastic Gradient Descent). A little reasoning can be provided for topics which come along the way to give a better understanding of it's application and use case in the node2vec.

4.3 Analysis of Embedding

The paper can also include visual interpretations of node embedding formed on running the algorithm. It also include some analysis that can be drawn from the method which help in better relationship understanding of the nodes and also the helps in understanding characteristic feature that governs the embedding.

References

[1] Aditya Grover & Jure Leskovec(2016). node2vec: Scalable Feature Learning for Networks