Summary Report: Node2Vec

Manas Mittal

1 Broad Area

Graph representational learning aims to eliminate the need for manual feature engineering by automatically learning node representations. The objective is to ensure that nodes with similar features are mapped to similar embeddings. Specifically, given a similarity measure in the embedding space such as cosine similarity, nodes that exhibit similarity in the graph structure should also have high cosine similarity in the embedding space.

2 Key Features of Interest for Nodes in a Graph

- Node features and labels
- Global and local structure of the graph

These learned representations can either be specific to a downstream task or agnostic to it. For instance, Node2Vec provides general-purpose embeddings that capture structural similarities in a graph independent of the downstream prediction task.

3 Node2Vec: An Overview

Node2Vec is an algorithmic framework designed to map nodes to a low-dimensional feature space, optimizing the likelihood of preserving network neighborhoods. It employs a flexible and biased second-order random walk mechanism to explore graph neighborhoods, ensuring a balance between local and global structure preservation.

4 Existing Techniques and Limitations

4.1 Manual Feature Engineering for Nodes and Edges

- Labor-intensive and dependent on the specific downstream task
- Lacks generalizability across different tasks

4.2 Optimization-Based Feature Learning

Supervised Learning: Achieves high accuracy but incurs high computational complexity. Specific to the downstream task at hand.
Unsupervised Learning: Reduces accuracy and also struggles with scalability due to expensive decompositions in large networks.

4.3 Neighborhood-Based Learning Objectives

- Prior methods focus on preserving local node neighborhoods
- Computationally efficient via stochastic gradient descent (SGD)
- However, rigid definitions of neighborhoods limit their effectiveness

5 Node2Vec Approach

Node2Vec overcomes these limitations by introducing a flexible notion of node neighborhoods. The framework adheres to two core principles:

- **Homophily:** Embedding nodes from the same network community closely together (BFS strategy)
- **Structural Similarity:** Ensuring nodes with similar roles have similar embeddings (DFS strategy)

The algorithm provides us with two hyperparameters p and q that can be tweaked to decide which notion should be captured more. These parameters can also be learned in a semi-supervised setting.

6 Technical Details of Node2Vec

- **Optimization Objective:** Optimizes a custom graph-based objective function using stochastic gradient descent (SGD). The function is similar to the original idea proposed in the DeepWalk paper and based on the ideas of the Skip-Gram model from the NLP domain.
- **Second-Order Biased Random Walks:** Generates sampled network neighborhoods while considering the previous node to guide exploration.
- **Flexible Neighborhood Exploration:** Different path probabilities allow for adaptable representation learning.
- **Generating Edge Embeddings:** Edge embeddings are generated by doing a binary operation on the participating nodes.
  - Hadamard operator: multiplication in corresponding dimensions.
  - Average

  - Weighted L1
  - Weighted L2

7 Experiments and Evaluation

7.1 Evaluation Tasks

- Multi-label Node Classification: Assigning multiple labels per node.
- Link Prediction: Predicting edges between node pairs.

7.2 Comparison with Baselines

Evaluated against LINE, Spectral Clustering, and DeepWalk. Node2Vec outperforms all baselines using macro F1 score.

7.3 Parameter Sensitivity (BlogCatalog Dataset)

- Macro F1 score decreases with higher p and q.
- Improves with increasing dimension, number of walks, walk length, and context window, before saturating.

7.4 Scalability

Linear scaling with the number of nodes.

7.5 Perturbation Analysis (BlogCatalog Dataset)

- **Missing Edges:** Performance (Macro F1 score) declines linearly with a small slope.
- **Noisy Edges:** Initial sharp drop in performance, then a slower decline.

8 Strengths of the Paper

- Provides an intuitive and practical approach to graph representation learning.
- Easy to understand and implement.
- Employs parallelizable algorithms for efficient scaling.
- Outperforms state-of-the-art (SOTA) algorithms significantly.
- Effectively integrates representation learning advancements from NLP, such as Skip-Gram, into graph-based tasks.

- Detailed parameter analysis: variation of different parameters of the BlogCatalog dataset.
- Robust to missing and noisy edges.

9 Limitations of the Paper

- **Limited Scalability:** Generating embeddings for large-scale graphs with millions of nodes poses computational challenges. Since every node's embedding has to be learned separately, this will cause storage issues.
- **Cannot Work with Dynamic Graphs:** This approach learns the node embedding at the start, and no new nodes can be added on the fly.
- **Ignores Node Features:** Node2Vec captures only the structure of the graph and does not consider node features, limiting its applicability in end-to-end systems like Graph Neural Networks.
- **Lack of GNN Comparison:** The paper does not compare Node2Vec with Graph Neural Networks, which are powerful and scalable tools.
- **Empirical Understanding of Edge Features:** No theoretical validation for the edge feature generation using node features.

10 Future Work and Improvements

- Comparison with Graph Neural Networks (GNN).
- Developing different notions of neighborhoods for richer representations.