node2vec: Scalable Feature Learning for Networks

A paper by Aditya Grover and Jure Leskovec, presented at Knowledge Discovery and Data Mining '16.

11/27/2018

Presented by: Dharvi Verma

CS 848: Graph Database Management

UNIVERSITY OF WATERLOO
FACULTY OF MATHEMATICS

OVERVIEW

MOTIVATION

RELATED WORK

PROPOSED SOLUTION

EXPERIMENTS: EVALUATION OF node2vec

REFERENCES

MOTIVATION

Representational learning on graphs -> applications in Machine Learning

Increase in predictive power!

Reduction in Engineering effort

Can an algorithm capture both homophily & structural equivalence of nodes?

An approach which preserves neighbourhood of nodes?

RELATED WORK

RELATED WORK: A SURVEY

Conventional paradigm in feature extraction (for networks): involve hand-engineered features

Unsupervised feature learning approaches:-

Linear & Non-Linear dimensionality reduction techniques are computationally expensive, hard to scale & not effective in generalizing across diverse networks

LINE: Focus is on the vertices of neighbor nodes or Breadth-First-Search to capture local communities in 1st phase.

In 2nd phase, nodes are sampled at a 2-hop distance from source node.

Deepwalk: Feature representations using uniform random walks. Special case of node2vec where parameters p & q both equal 1.

RELATED WORK: A SURVEY

SKIP-GRAM MODEL

Hypothesis: Similar words tend to appear in similar word neighbourhood

"It scans over the words of a document, and for every word it aims to embed it such that the word's features can predict nearby words

The node2vec algorithm is inspired by the Skip-Gram Model & essentially extends it..

Multiple sampling strategies for nodes : There is no clear winning sampling strategy! Solution?

A flexible objective!

PROPOSED SOLUTION

..but wait, what are homophily & structural equivalence?

The homophily hypothesis-

Highly interconnected nodes that belong to the same communities or network clusters

The structural equivalence hypothesis-

Nodes with similar structural roles in the network

Embedded closely together

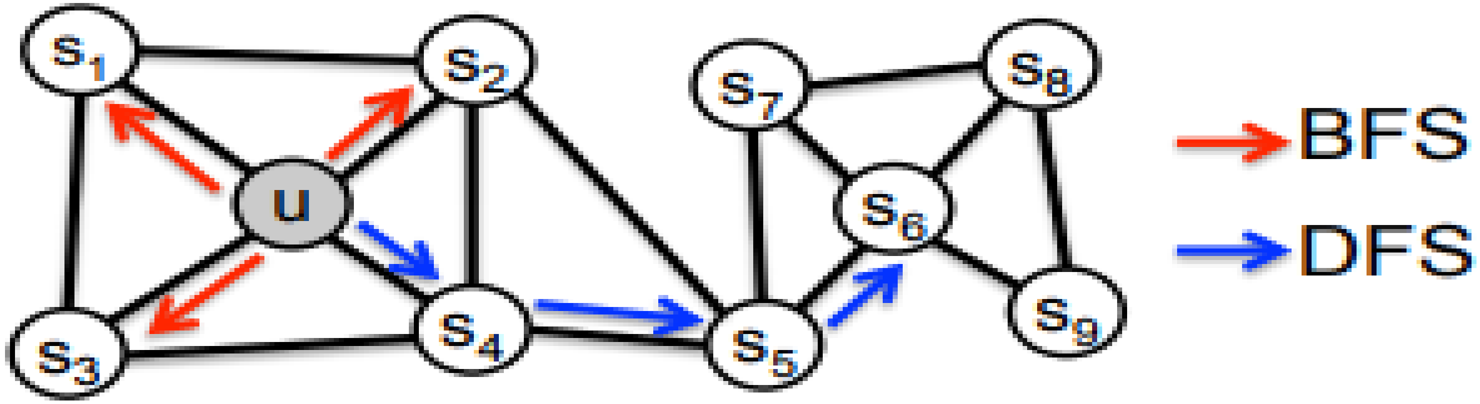

Figure 1: BFS & DFS strategies from node u for k=3 (Grover et al.)

FEATURE LEARNING FRAMEWORK

It is based on the Skip-Gram Model and applies to: any (un)directed, (un)weighted network

Let G = (V,E) be a given network and f: V -> R^d a mapping function from nodes to feature representations.

d= number of dimensions of feature representations, f is a matrix of size |V| X d parameters

For every source node $u \in V$, $N_S(u) \subset V$ is a network neighborhood of *node u* generated through a neighborhood sampling strategy S.

Objective function to be optimized:

$$\max_f \sum_{u \in V} \log Pr(N_S(u)|f(u)). \qquad (1)$$

FEATURE LEARNING FRAMEWORK

Assumptions for optimization:

A. Conditional Independence: "Likelihood of observing a neighborhood node is independent of observing any other neighborhood node given the feature representation of the source."

$$Pr(N_S(u)|f(u)) = \prod_{n_i \in N_S(u)} Pr(n_i|f(u)).$$

B. Symmetry in feature space: Between source node & neighbourhood node.

Hence, Conditional likelihood of every source-neighborhood node pair modelled as a softmax unit parametrized by a dot product of their features:

$$Pr(n_i|f(u)) = \frac{\exp(f(n_i) \cdot f(u))}{\sum_{v \in V} \exp(f(v) \cdot f(u))}.$$

FEATURE LEARNING FRAMEWORK

Using the assumptions, the objective function in (1) reduces to:

$$\max_f \sum_{u \in V} \Big[-\log Z_u + \sum_{n_i \in N_S(u)} f(n_i) \cdot f(u) \Big]. \qquad (2)$$

SAMPLING STRATEGIES

How does the skip-gram model extend to node2vec?

Networks aren't linear like text…so how can neighbourhood be sampled?

Randomized procedures: The neighborhoods $N_S(u)$ are not restricted to just immediate neighbors -> can have different structures depending on the sampling strategy S

Sampling strategies

a. Breadth-first Sampling (BFS): For structural equivalence

b. Depth-first Sampling (DFS): Obtains macro view of neighbourhood -> homophily

What is node2vec?

"**node2vec** is an algorithmic framework for learning continuous feature representations for nodes in networks"

- ❑ semi-supervised learning algorithm
- ❑ learns low-dimensional representations for nodes by optimizing neighbour preserving objective
- ❑ graph-based objective function customized using stochastic gradient descent (SGD)

How does it preserve neighborhood of nodes?

?

RANDOM WALKS TO CAPTURE DIVERSE NEIGHBOURHOODS

For a source node u such that c_o=u, c_i denotes the i^{th} node in the walk for a random walk of length l.

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

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

BIAS IN RANDOM WALKS

To enable flexibility, the random walks are biased using Search Bias parameter α.

Suppose a random walk that just traversed edge (t, v) and is currently at node v. To decide on the next step, the walk evaluates transition probability π_{vx} on edges (v,x) where v is the starting point.

Let $\pi_{vx} = \alpha_{pq}(t, x) \cdot w_{vx}$

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} is the shortest path between nodes t and x.

ILLUSTRATION OF BIAS IN RANDOM WALKS

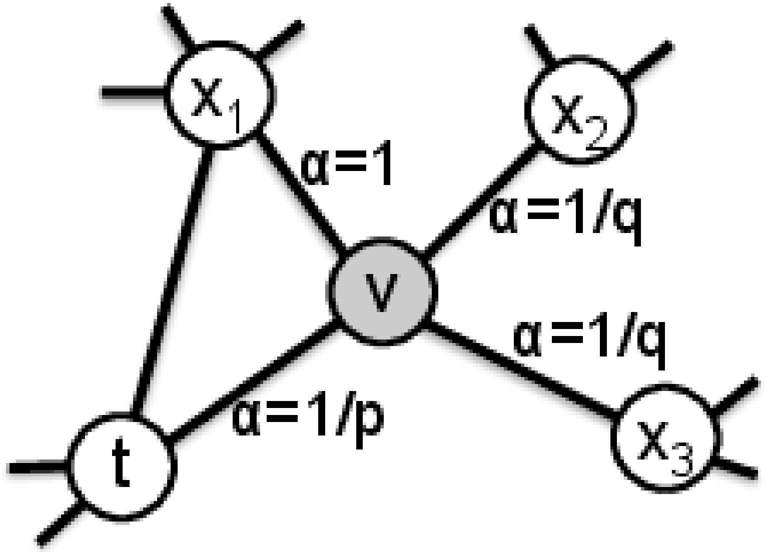

Figure 2: The walk just transitioned from t to v and is now evaluating its next step out of node v. Edge labels indicate search biases α (Grover et al.)

Significance of parameters p & q

Return parameter p: Controls the likelihood of immediately revisiting a node in the walk.

High value of p -> less likely to sample an already visited node, low value of p encourages a local walk

In-out parameter q: Allows the search to distinguish between inward & outward nodes.

For q>1, search is reflective of BFS (local view), for q <1, DFS-like behaviour due to outward exploration

The node2vec algorithm

Algorithm 1 The *node2vec* algorithm.

```
LearnFeatures (Graph G = (V, E, W), Dimensions d, Walks per
    node r, Walk length l, Context size k, Return p, In-out q)
    π = PreprocessModifiedWeights(G, p, q)
    G′ = (V, E, π)
    Initialize walks to Empty
    for iter = 1 to r do
        for all nodes u ∈ V do
            walk = node2vecWalk(G′, u, l)
            Append walk to walks
    f = StochasticGradientDescent(k, d, walks)
    return f
```

```
node2vecWalk (Graph G′ = (V, E, π), Start node u, Length l)
    Inititalize walk to [u]
    for walk_iter = 1 to l do
        curr = walk[−1]
        V_curr = GetNeighbors(curr, G′)
        s = AliasSample(V_curr, π)
        Append s to walk
    return walk
```

Figure 3: The node2vec algorithm (Grover et al)

EXPERIMENTS

1. Case Study: Les Misérables network

Description of the study: a network where nodes correspond to characters in the novel Les Misérables, edges connect coappearing characters. Number of nodes= 77, number of edges=254, d = 16. node2vec is implemented to learn feature representation for every node in the network.

For p = 1; q = 0.5 -> relates to homophily, for p=1, q=2, colours correspond to structural equivalence.

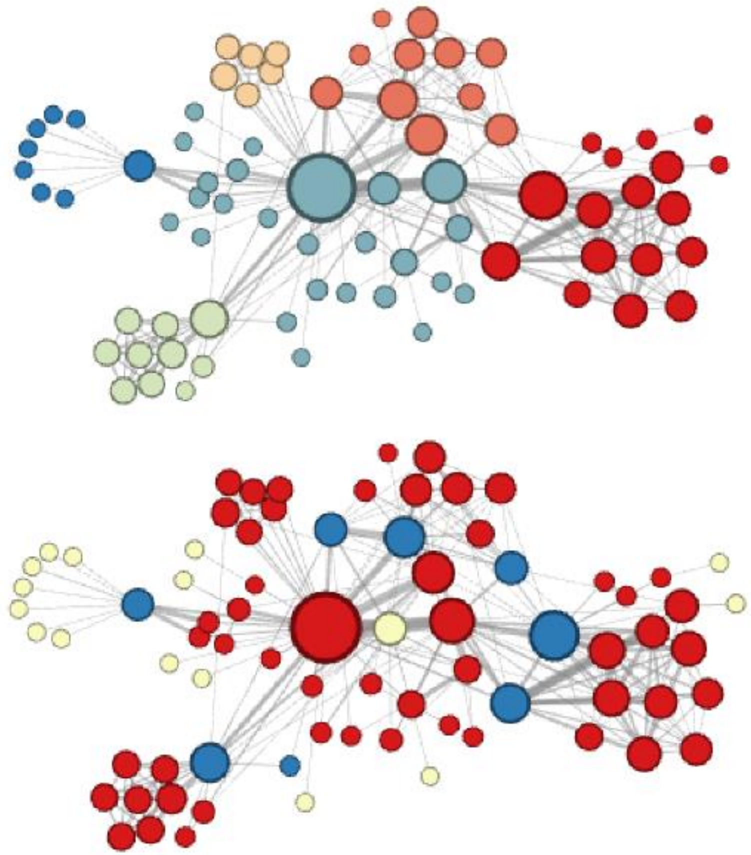

Figure 4: Complementary visualizations of Les Misérables coappearance network generated by node2vec with label colors reflecting homophily (top) and structural equivalence (bottom) (Grover et al).

2. Multi-label Classification

The node feature representations are input to a one-vs-rest logistic regression classifier with L2 regularization. The train and test data is split equally over 10 random instances.

Algorithm	Dataset		
	BlogCatalog	PPI	Wikipedia
Spectral Clustering	0.0405	0.0681	0.0395
DeepWalk	0.2110	0.1768	0.1274
LINE	0.0784	0.1447	0.1164
node2vec	**0.2581**	**0.1791**	**0.1552**
node2vec settings (p,q)	0.25, 0.25	4, 1	4, 0.5
Gain of *node2vec* [%]	**22.3**	**1.3**	**21.8**

Table 1: Macro-F1 scores for multilabel classification on BlogCat-alog, PPI (Homo sapiens) and Wikipedia word cooccurrence networks with 50% of the nodes labeled for training.

Note: The F1 score is the harmonic average of the precision and recall, where an F1 score reaches its best value at 1 (perfect precision and recall) and worst at 0.

2. Multi-label Classification

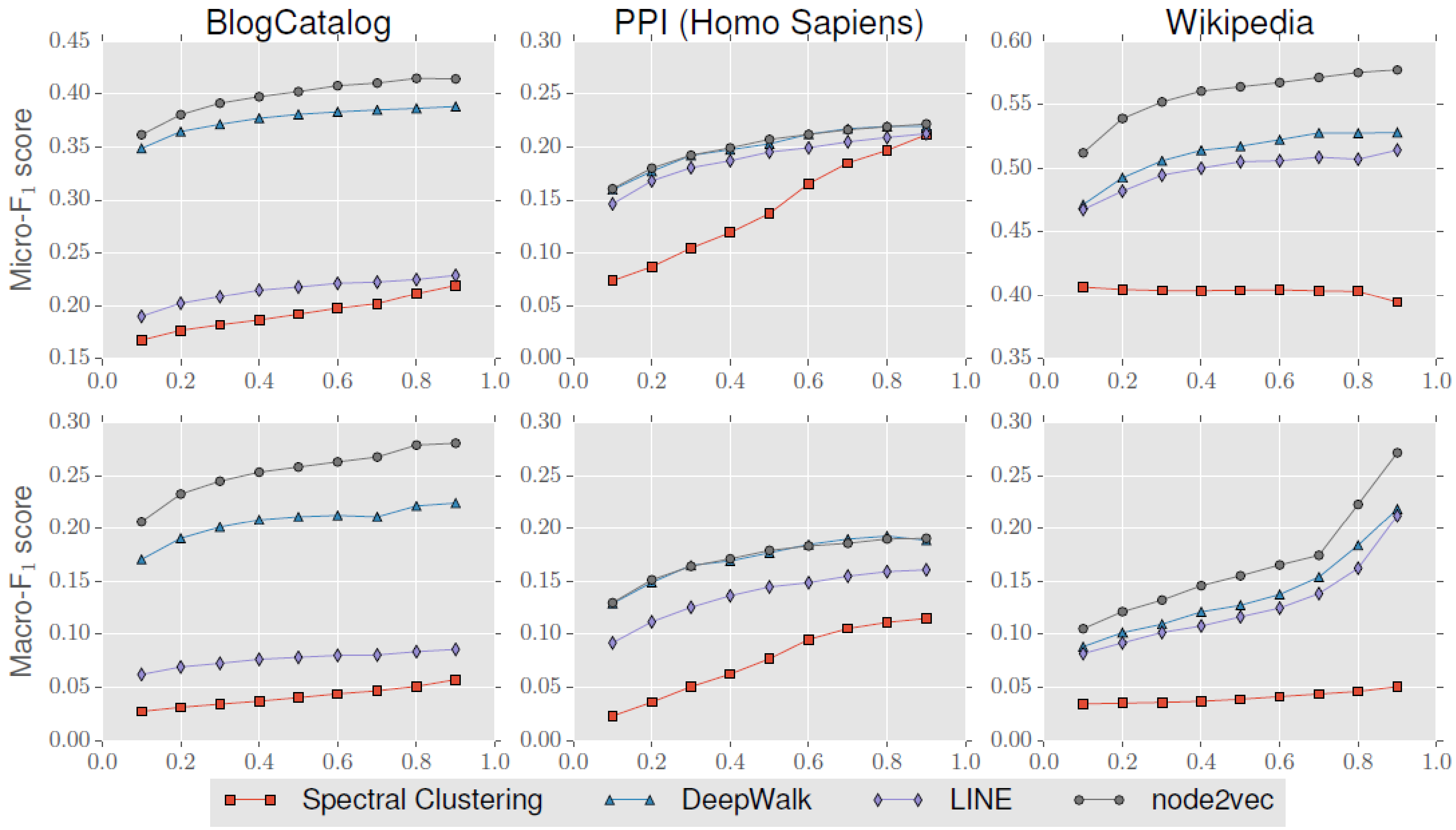

Figure 5: Performance evaluation of different benchmarks on varying the amount of labeled data used for training. The x axis denotes the fraction of labeled data, whereas the y axis in the top and bottom rows denote the Micro-F1 and Macro-F1 scores respectively (Grover et al).

3. Parameter Sensitivity

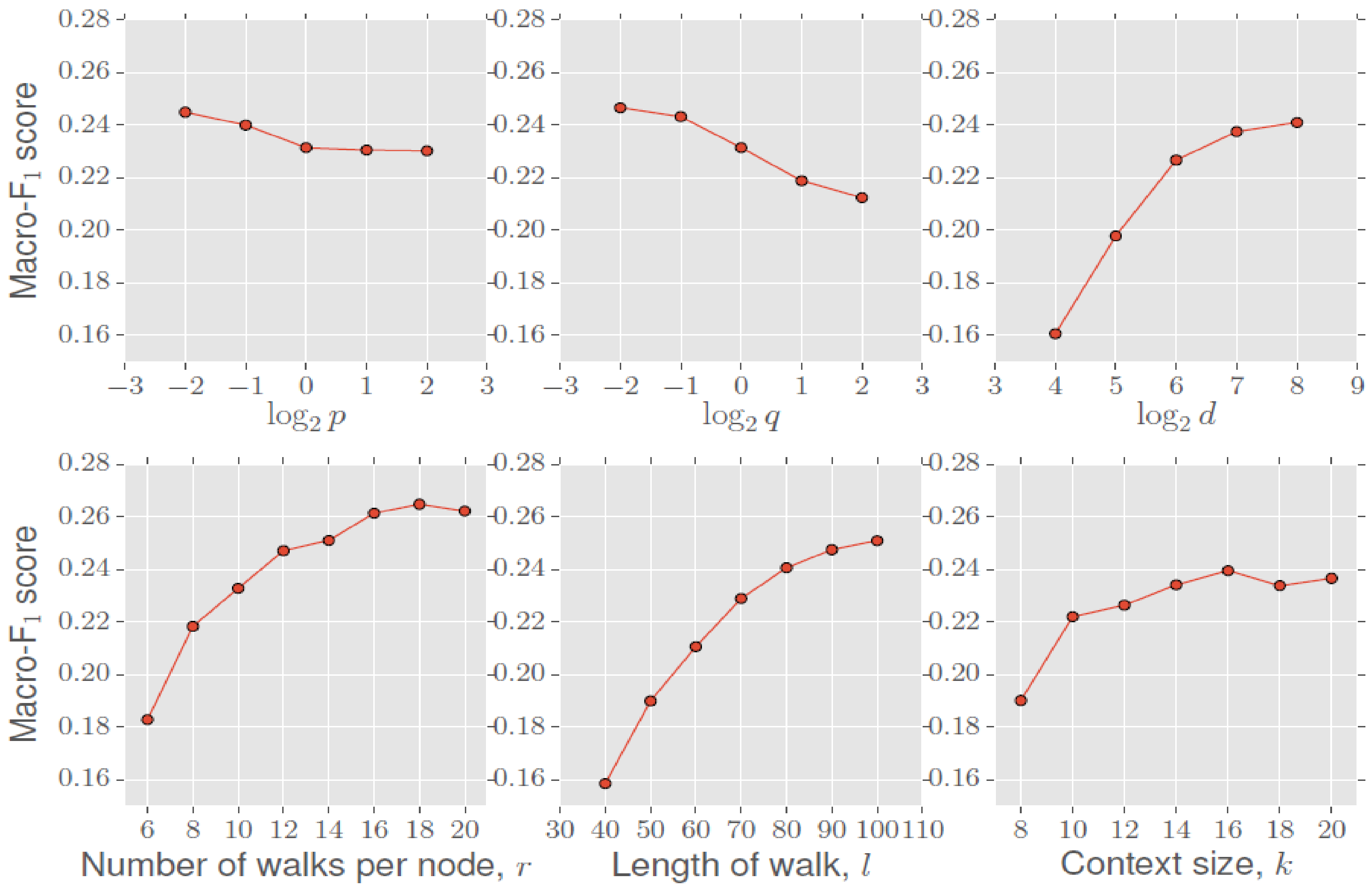

Figure 6: Parameter Sensitivity

4. Perturbation Analysis

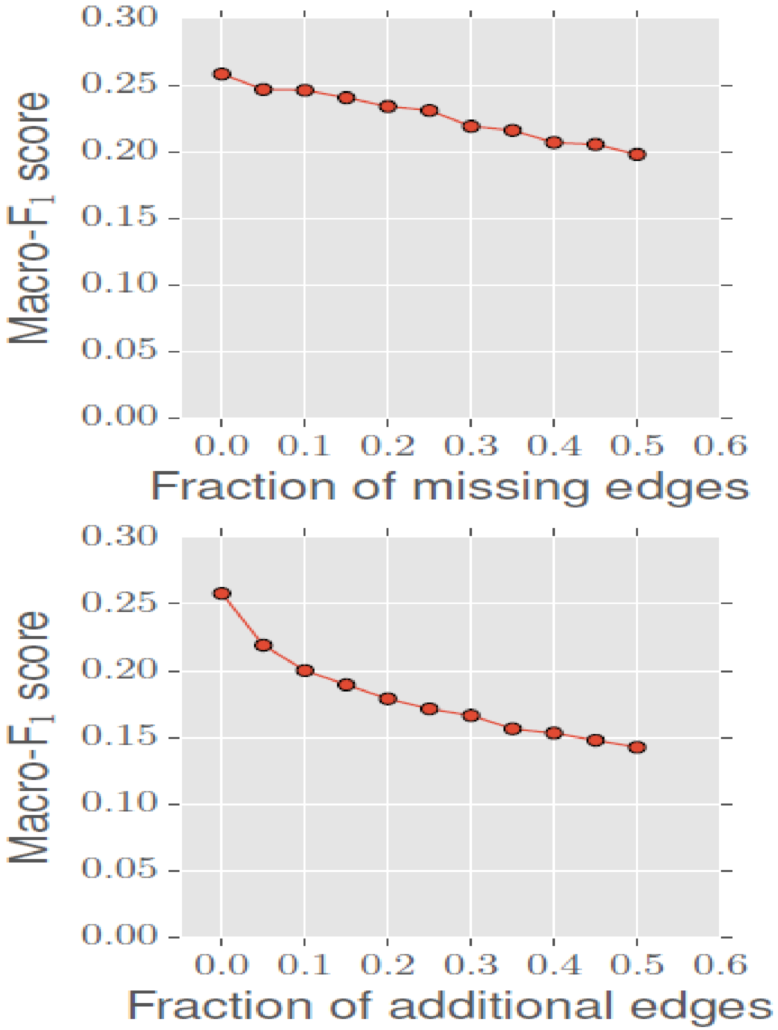

Figure 7: Perturbation analysis for multilabel classification on the BlogCatalog network.

5. Scalability

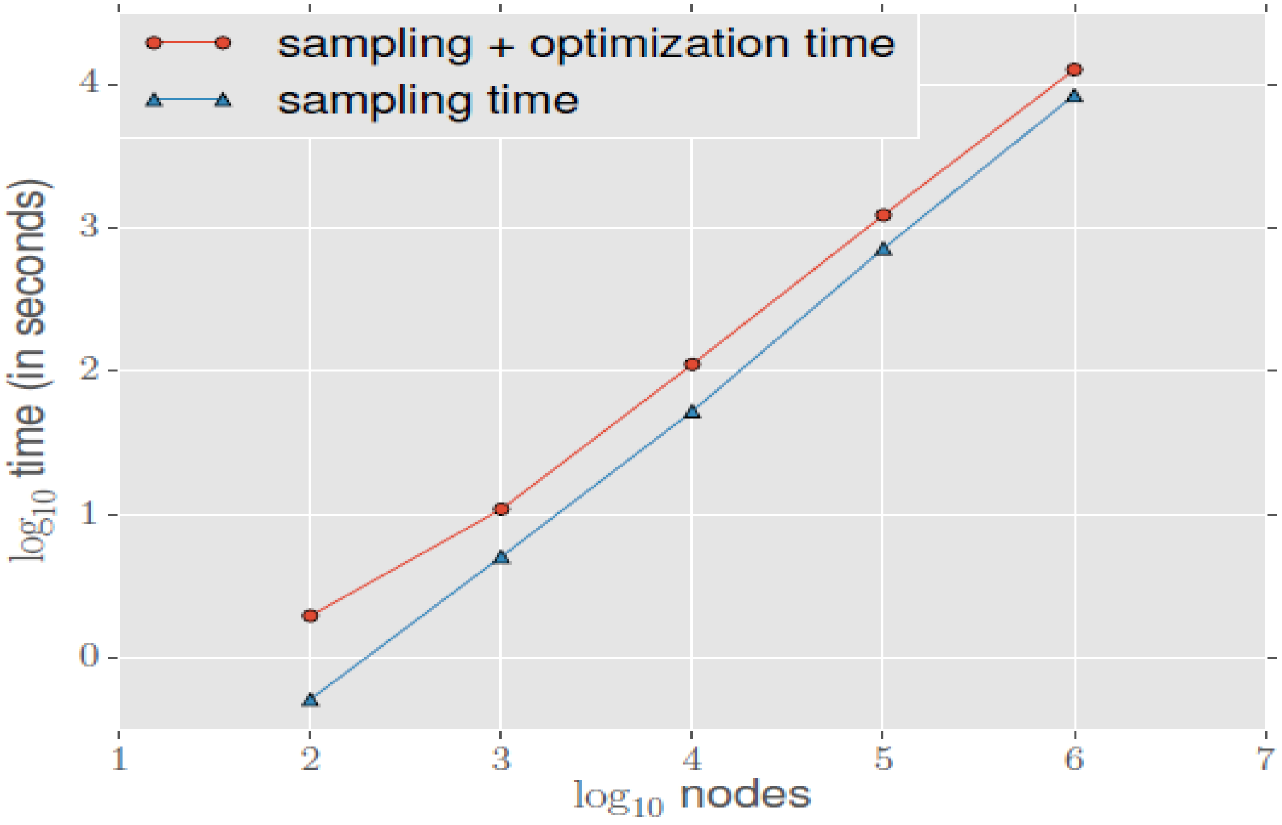

Figure 8: Scalability of node2vec on Erdos-Renyi graphs with an average degree of 10.

6. Link Prediction

Observation: The learned feature representations for node pairs significantly outperform the heuristic benchmark scores with node2vec achieving the best AUC improvement.

Amongst the feature learning algorithms, node2vec >> DeepWalk and LINE in all networks

Op	Algorithm	Dataset		
		Facebook	PPI	arXiv
	Common Neighbors	0.8100	0.7142	0.8153
	Jaccard's Coefficient	0.8880	0.7018	0.8067
	Adamic-Adar	0.8289	0.7126	0.8315
	Pref. Attachment	0.7137	0.6670	0.6996
(a)	Spectral Clustering	0.5960	0.6588	0.5812
	DeepWalk	0.7238	0.6923	0.7066
	LINE	0.7029	0.6330	0.6516
	node2vec	0.7266	0.7543	0.7221
(b)	Spectral Clustering	0.6192	0.4920	0.5740
	DeepWalk	**0.9680**	0.7441	0.9340
	LINE	0.9490	0.7249	0.8902
	node2vec	**0.9680**	**0.7719**	**0.9366**
(c)	Spectral Clustering	0.7200	0.6356	0.7099
	DeepWalk	0.9574	0.6026	0.8282
	LINE	0.9483	0.7024	0.8809
	node2vec	0.9602	0.6292	0.8468
(d)	Spectral Clustering	0.7107	0.6026	0.6765
	DeepWalk	0.9584	0.6118	0.8305
	LINE	0.9460	0.7106	0.8862
	node2vec	0.9606	0.6236	0.8477

Figure 9: Area Under Curve (AUC) scores for link prediction. Comparison with popular baselines and embedding based methods bootstapped using binary operators: (a) Average, (b) Hadamard, (c) Weighted-L1, and (d) Weighted-L2 (Grover et al.)

REFERENCE OF THE READING

node2vec: Scalable Feature Learning for Networks. A. Grover, J. Leskovec. *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2016.

THANK YOU