node2vec: Scalable Feature Learning for Networks

Motivation

- Discriminative features needed for many applications involving nodes (e.g., multi-label classification) and edges (e.g., link prediction)
- Hand-crafted features take time, may not generalize well across tasks.
- Representation learning:
  - Trade-off between computational efficiency and prediction accuracy
  - Option 1: Directly optimize for high accuracy for a downstream task
    - Large number of unknowns, high training time
  - Option 2: Keep the objective function independent of the downstream task
    - A carefully designed objective can match the prediction accuracy
    - Current approaches don’t capture the diversity of connectivity patterns

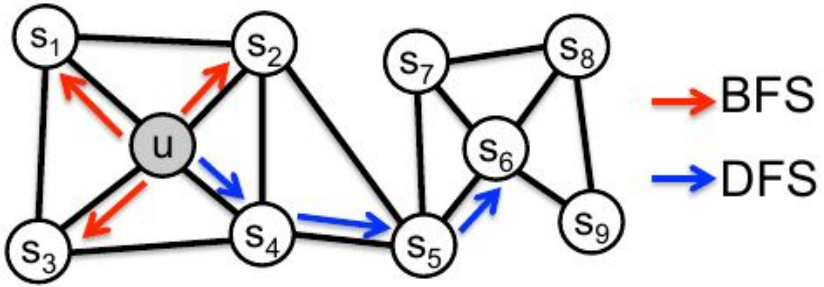
Overview of contributions

- An efficient scalable algorithm for feature learning in networks that efficiently optimizes a novel, neighborhood preserving objective using SGD.
- *Flexibility* in discovering representations conforming to different equivalences
- Feature representations of individual nodes can be extended to pairs of nodes (i.e., edges)
- Empirical validation for multi-class classifications and link prediction

Local neighbourhoods of nodes

- Like DeepWalk and LINE, the node2vec objective aims to preserve local neighbourhoods of nodes.
- Efficient minimization through SGD
- DeepWalk and LINE involve a rigid notion of network neighbourhoods
  - Insensitive to peculiar, mixed connective patterns

Connectivity patterns



- Nodes could be organized based on:
  - Communities they belong to (i.e. homophily)
  - Structural roles (i.e. structural equivalence)
- Real world networks display a mixture of such patterns

- Desirability: A flexible algorithm that can learn node representations obeying both principles:
  - Embed nodes from the same network community closely together
  - Nodes that share similar roles should have similar embeddings.

Related work

- Dimensionality reduction techniques
  - Linear (e.g., PCA)
  - Non-linear (e.g., IsoMap)
  - Computationally expensive
  - The objectives not robust to diverse patterns (homophily and structural equivalence)
- Skipgram model: Continuous word embeddings by optimizing a neighbourhood preserving likelihood
- DeepWalk and LINE represent a network as a “document”
  - Sampling of nodes devoid of flexibility
  - Node2vec overcomes that drawback; provides parameters to tune the explored seach space
- Deep network approaches directly optimize for downstream tasks
  - Less scalable owing to high training times

Feature Learning Framework

Let G = (V, E) be a given network.

We define a mapping f: V -> R^d

Optimization Function:

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

In order to make the optimization problem tractable, we make two standard assumptions:

Conditional Independence

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

Symmetry in Feature Space

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

$$\max_f \quad \sum_{u \in V} \left[-\log Z_u + \sum_{n_i \in N_S(u)} f(n_i) \cdot f(u) \right].$$

Classic Search Strategies

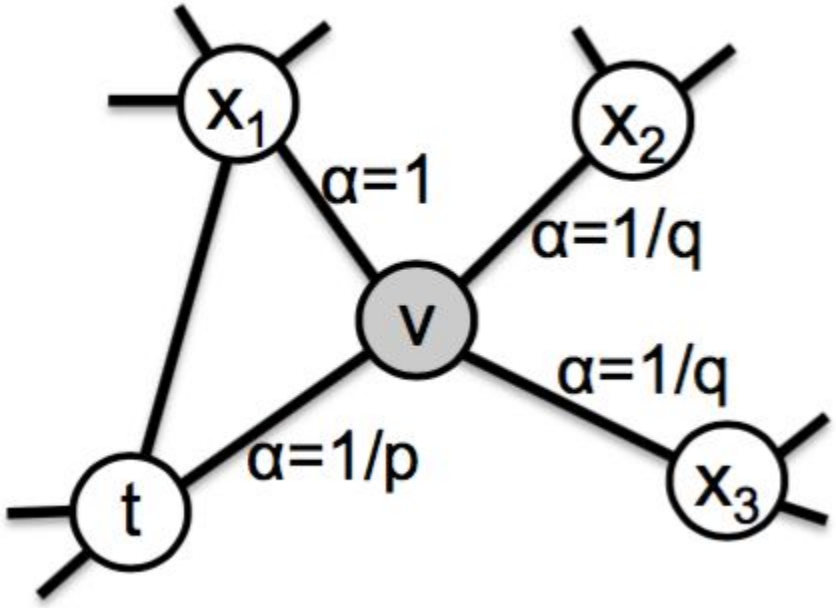
- Depth First Sampling
    - The neighborhood NS is restricted to nodes which are immediate neighbors of the source.
- Breadth First Sampling
    - The neighborhood consists of nodes sequentially sampled at increasing distances from the source node.

Random Walk

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

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

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

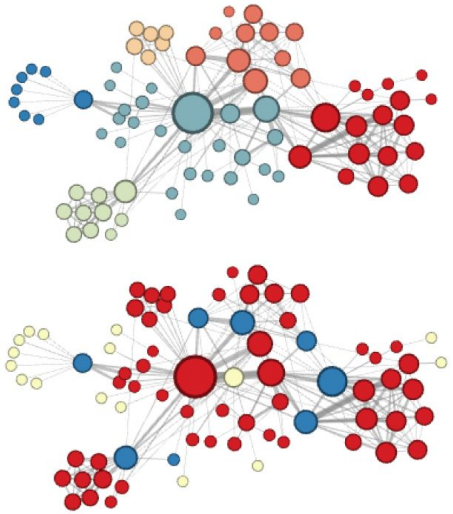

x1
x2
α=1
α=1/q
v
α=1/q
α=1/p
t
x3

Learning Edge Features

Operator	Symbol	Definition
Average	$\boxplus$	$[f(u) \boxplus f(v)]_i = \frac{f_i(u)+f_i(v)}{2}$
Hadamard	$\boxdot$	$[f(u) \boxdot f(v)]_i = f_i(u) * f_i(v)$
Weighted-L1	$\|\cdot\|_{\bar{1}}$	$\|f(u) \cdot f(v)\|_{\bar{1}i} = \|f_i(u) - f_i(v)\|$
Weighted-L2	$\|\cdot\|_{\bar{2}}$	$\|f(u) \cdot f(v)\|_{\bar{2}i} = \|f_i(u) - f_i(v)\|^2$

Testing on Les Miserables Network

- Top figure : p = 1, q = 0.5
- Bottom figure : p = 1, q = 2
- Top figure represents homophily
- Bottom figure represents structural equivalence



Experimental Setup

- Based on multi-label classification for nodes
- Link Prediction for Edges
- Performance evaluated against
    - **Spectral clustering** : This is a matrix factorization approach in which we take the top d eigenvectors of the normalized Laplacian matrix of graph G as the feature vector representations for nodes.
    - **DeepWalk** : This approach learns d-dimensional feature representation by simulating uniform random walks. The sampling strategy in DeepWalk can be seen as a special case of node2vec with p = 1, q = 1
    - **LINE** : learns d-dimensional representation in two separate phases. In the first phase, it learns d/2 dimension by BFS-style simulation over immediate neighbors of nodes. In the second phase, it learns d/2 dimensions by sampling nodes strictly at a 2-hop distance from source nodes.

Multi-Label Classification

Certain fraction of nodes and their labels are observed. The task is to predict the labels for the remaining nodes.
Following datasets used for this:

- **BlogCatalog** : This is a network of social relationships of the bloggers listed on the BlogCatalog website. The labels represent blogger interests inferred through the metadata provided by the bloggers. The network has 10,312 nodes, 333,983 edges, and 39 different labels.
- **Protein-Protein Interactions (PPI)** : We use a subgraph of the PPI network for Homo Sapiens. The subgraph corresponds to the graph induced by nodes for which we could obtain labels from the hallmark gene sets and represent biological states. The network has 3,890 nodes, 76,584 edges, and 50 different labels.
- **Wikipedia** : This is a co-occurrence network of words appearing in the first million bytes of the Wikipedia dump. The labels represent the Part-of-Speech (POS) tags inferred using the Stanford POS-Tagger. The network has 4,777 nodes, 184,812 edges, and 40 different labels.

All the networks exhibit a fair mix of homophilic and structural equivalences.

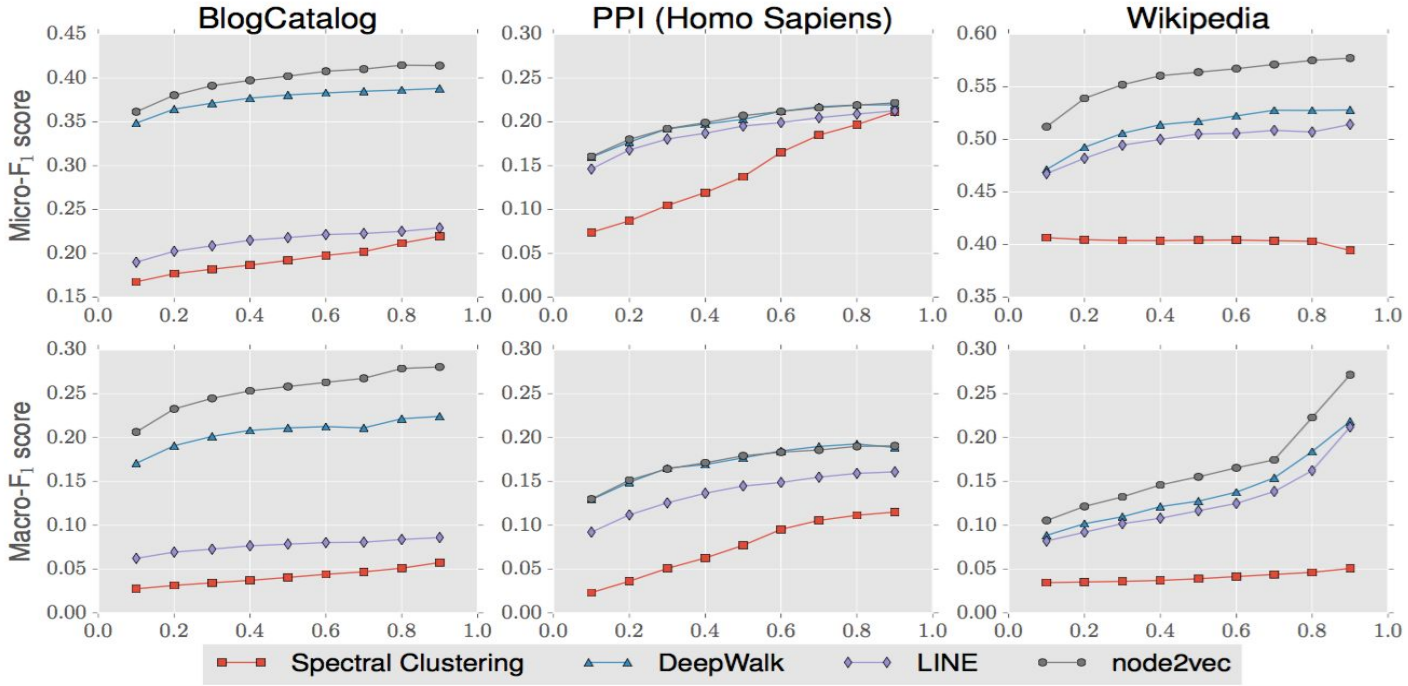


Figure 4: Performance evaluation of different benchmarks on varying the amount of labeled data used for training. The x axis denotes the fraction of labeled data, whereas the y axis in the top and bottom rows denote the Micro-F_1 and Macro-F_1 scores respectively. DeepWalk and *node2vec* give comparable performance on PPI. In all other networks, across all fractions of labeled data *node2vec* performs best.

Parameter Sensitivity (on BlogCatalog data)

- Performance of node2vec improves as the in-out parameter p and the return parameter q decrease.
- While a low q encourages outward exploration, low p ensures that the walk does not go too far from the start node.
- Effect of number of features d and the node’s neighborhood parameter (number of walks r, walk length l, neighborhood size k) observed as well.
- Performance saturates when dimension reaches around 100.
- Increasing number and length of walks per source improves performance.
- Increasing k also improves performance, but at the cost of increased optimization time.

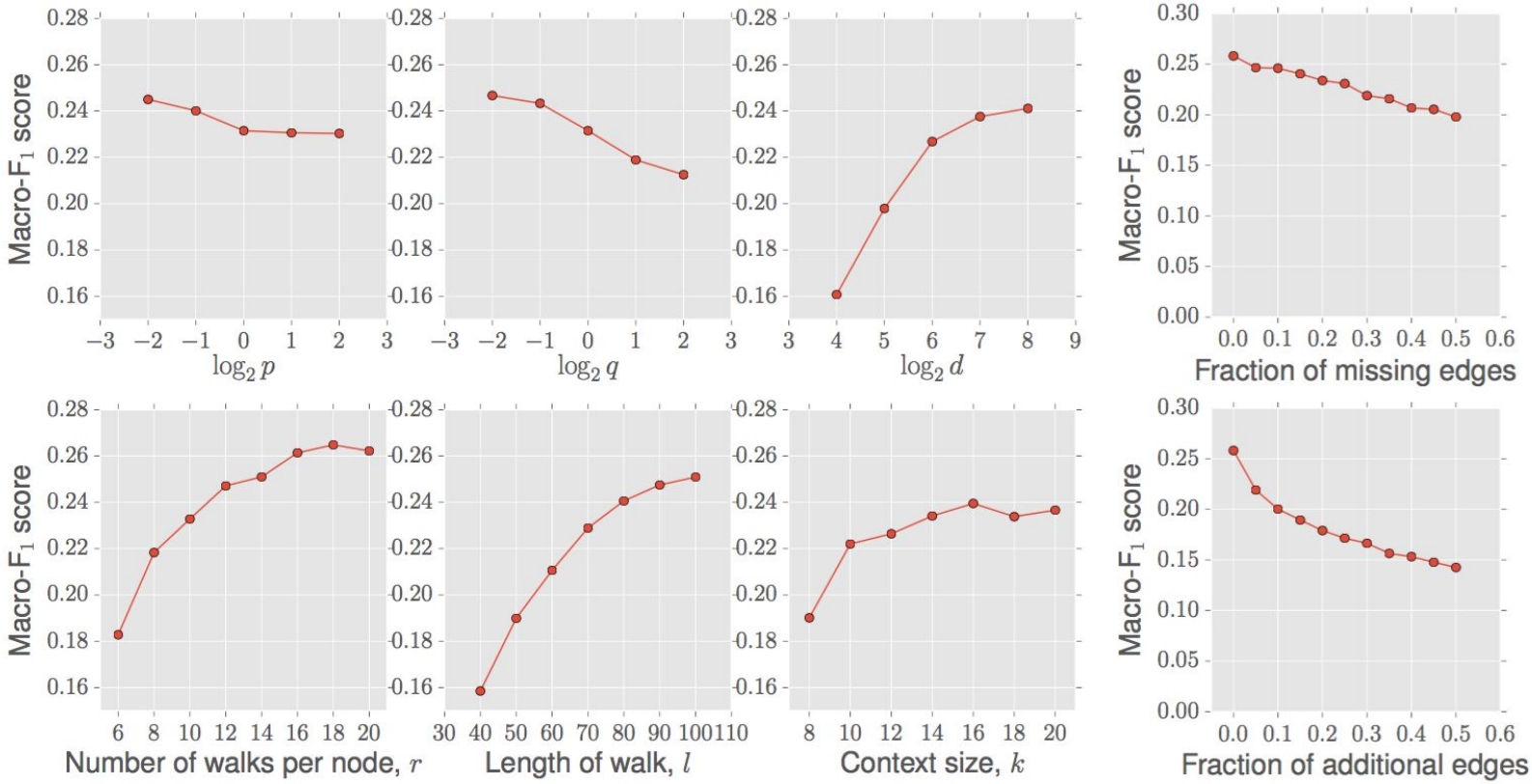


Figure 5: (a). Parameter sensitivity (b). Perturbation analysis for multilabel classification on the BlogCatalog network.

Perturbation Analysis

- Performance measured as a function of the fraction of missing edges
- Missing edges are chosen randomly, subject to the constraint that the number of connected components in the network remain fixed
- Decrease in Macro-F1 score as the fraction of missing edges increases is observed
- Robustness is important when a graph is evolving over time (eg. citation network)
- Performance measured as a function of the fraction of noisy edges added
- Noisy edges between randomly selected pairs in the network is added
- Robustness to false edges is useful (eg. sensor networks where the measurements used for constructing the network are noisy)

Scalability

- Node-representation learnt using node2vec with deafult parameter values for Erdos-Renyi graph
- Tested with node sizes from 100 to 1,000,000 nodes with constant average degree of 10

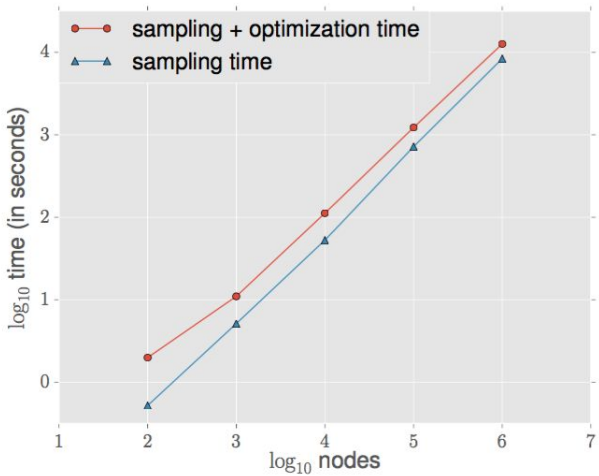


Figure 6: Scalability of *node2vec* on Erdos-Renyi graphs with an average degree of 10.

Link Prediction

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

Table 4: 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 (See Table 1 for definitions).

Discussion

- Feature learning as a search based optimisation problem
- It is observed that BFS can explore only limited neighborhoods. This makes BFS suitable for characterizing structural equivalences in network that rely on the immediate local structure of nodes.
- On the other hand, DFS can freely explore network neighborhoods which is important in discovering homophilous communities at the cost of high variance.