node2vec: Scalable Feature Learning for Networks

KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, August 2016

Jade F. Preston
PhD Student
School of Data Science
yry2jy@virginia.edu

Background

- With network analysis, we are interested in classifying nodes and predicting their links
  - Example 1: Social network
    - Classify user interest in products/predict the connection that creates their interest
  - Example 2: Supply chain network
    - Classify demand nodes/predict the route or mode of transportation between locations

- Two ways to classify nodes: <u>supervised</u> and <u>unsupervised</u> learning

Problem Solved (1/2)

- Trade-offs for supervised and unsupervised machine learning (ML) algorithms for networks:

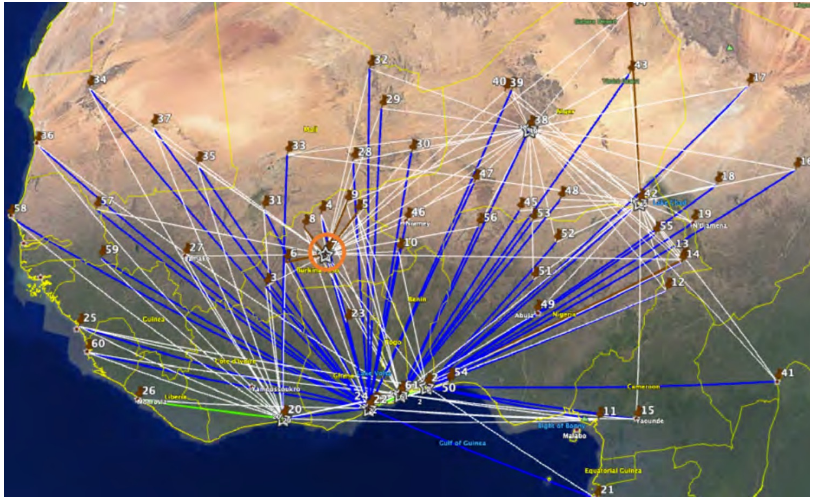
Supervised Learning	Unsupervised Learning
Involves high training time complexity Requires domain specific expert knowledge	Less training time complexity
Can have high classification accuracy	Not scalable to large, real-world problems Yields low classification performance

No optimizable objective function for scalable, unsupervised, feature learning in networks

Problem Solved (2/2)

- Build framework based on requirements/objectives:
  - Scale for large networks
  - Feature learn
  - Predict network relationships (nodes and edges)

- Compare performance to other feature learning algorithms
  - Speed/efficiency
  - Classification accuracy



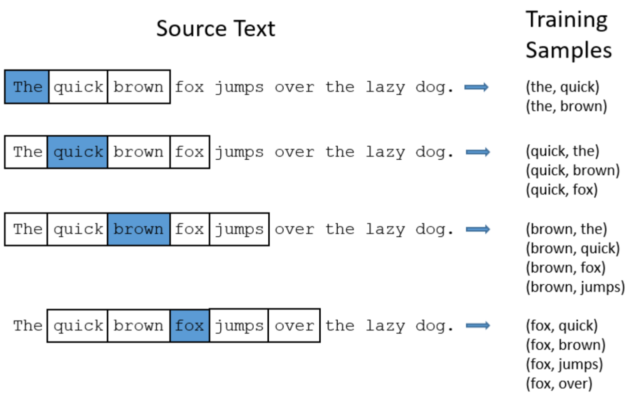
Solution and Contributions (1/2)

Introduce a semi-supervised algorithm for scalable feature learning in networks

- Optimize maximum likelihood function using stochastic gradient decent with negative sampling:

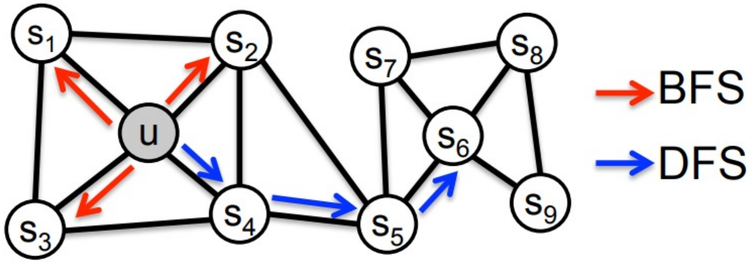
$$\max_{f} \sum_{u \in V} \log Pr(N_S(u)|f(u))$$

- Implement feature learning framework for nodes - based on skip-gram architecture
  - Similar words appear in the same neighborhood



Solution and Contributions (2/2)

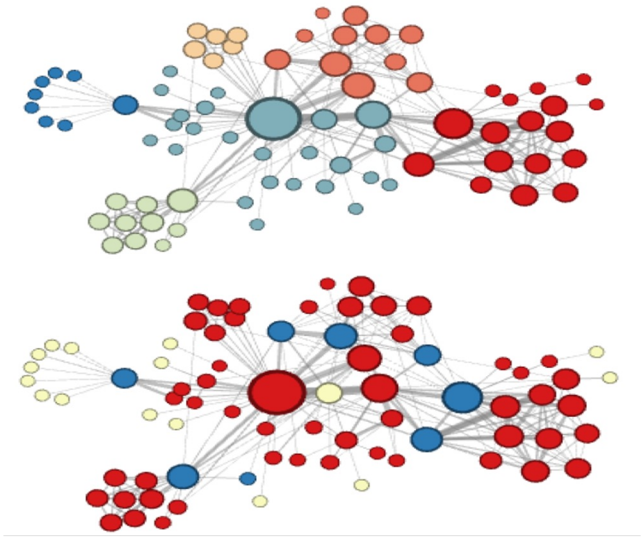
- Implement biased random walk
- Strike balance between search strategies:
    - Breadth First Sampling (BFS) – homophily (p)
    - Depth First Sampling (DFS) – structural equivalence (q)
- Feature engineer edges- based on binary operators between nodes



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	$\|\cdot\|_{\bar{1}}$	$\|f(u) \cdot f(v)\|_{\bar{1}i} = \lvert f_i(u) - f_i(v)\rvert$
Weighted-L2	$\|\cdot\|_{\bar{2}}$	$\|f(u) \cdot f(v)\|_{\bar{2}i} = \lvert f_i(u) - f_i(v)\rvert^2$

Evaluation

- Les Misérables Case Study
- Node classification comparison
  - Spectral Clustering, Deep Walk, LINE
- Edge prediction comparison
  - Common Neighbors, Jaccard's Coefficient, Adamic-Adar Score, Preferential Attachment



Based on the Macro-F1 score, node2vec outperforms well known prediction algorithms in node and edge prediction by 27% and 13% respectively

Key Insights and Takeaways

- Parameter Sensitivity
  - Performance fluctuates as parameters change
- Perturbation Analysis
  - Robust to missing edges
- Scalability
  - Generating one million node representations
- Parallelizability
  - Phase can occur in tandem and executed asynchronously

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)
- $G' = (V, E, \pi)$
- Initialize $walks$ to Empty
- **for** $iter = 1$ **to** r **do**
  - **for all** nodes $u \in V$ **do**
    - $walk$ = node2vecWalk(G', u, l)
    - Append $walk$ to $walks$
- f = StochasticGradientDescent$(k, d, walks)$
- **return** f

node2vecWalk (Graph $G' = (V, E, \pi)$, 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** $walk$

Advocate / Critic

- Advocate: Developed multiple novel ideas
  - scalable and efficient algorithm
  - Outperforms comparable methodologies

- Critic: Many undefined technical concepts and assumptions
  - Left too much interpretation up to the reader
  - Example: Define what they meant by semi-supervised

Related Work

- struc2vec: Learning Node Representations from Structural Identity
  - learning framework for nodes structural equivalences
- Learning edge representations via low-rank asymmetric projections
  - Learning algorithm for edge classification using node embeddings
- sub2Vec: Feature Learning for Subgraphs
  - Learning algorithm for feature representations of subgraphs

THANKS!

Any questions?

yry2jy@virginia.edu