THE UNIVERSITY OF TEXAS AT AUSTIN
Department of Statistics and Data Sciences
College of Natural Sciences

Stat Models for Big Data

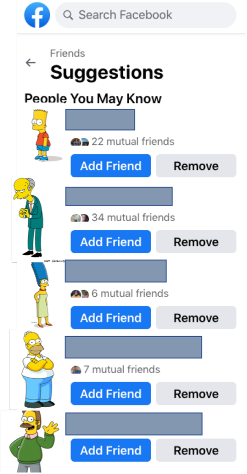
Graph representation learning - node2vec

Purnamrita Sarkar
Department of Statistics and Data Science
The University of Texas at Austin

https://psarkar.github.io/teaching

Machine learning tasks with networks

- Link prediction:
  - Given the network structure, can we predict missing edges?
  - Useful for suggesting new friends, new products, etc.



Machine learning tasks with networks

- Semi-supervised learning:
  - Given some node labels and the network structure, can we predict the labels of other nodes?
  - Given the identities of some websites which are link farms and some that are not, can we detect other link farms?

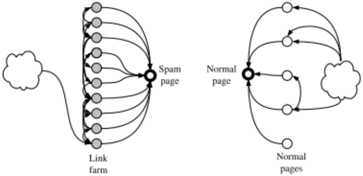
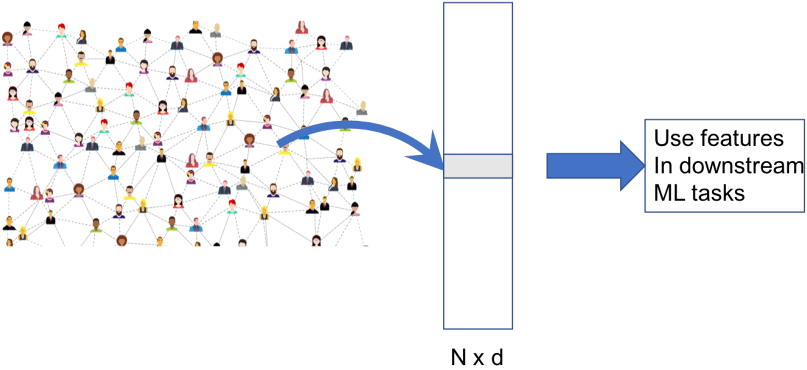


Figure 1: Schematic depiction of the neighborhood of a page participating in a link farm (left) and a normal page (right).

Figure 1: Becchetti et al, Link-Based Characterization and Detection of Web Spam, AIRWEB'06

How to engineer features from graphs



- Learn function $f : u \rightarrow \mathbb{R}^d$, where u is a node to map a network node into a d dimensional space.

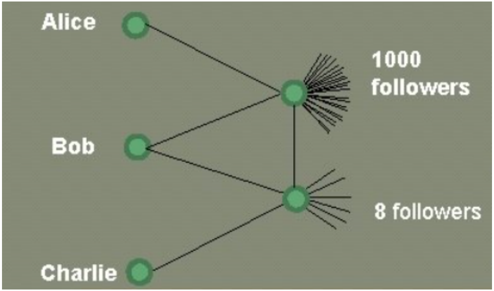
We have seen similar ideas before

- We have looked at network embeddings before.
- Spectral clustering of various types lead to network embeddings
- We have looked at statistical network models, where the nodes can be represented by latent vectors, and the goal is to learn them.

Graph embedding: main idea

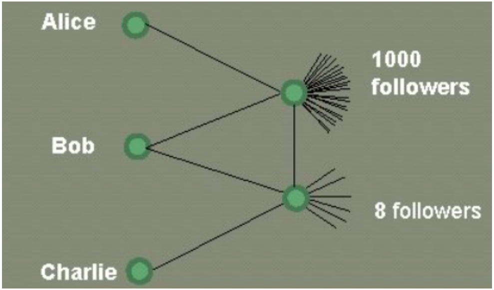
- Embed nodes such that similarity in embedding space reflects similarity in the network.
- What was the network similarity for spectral clustering?

Different network similarity measures



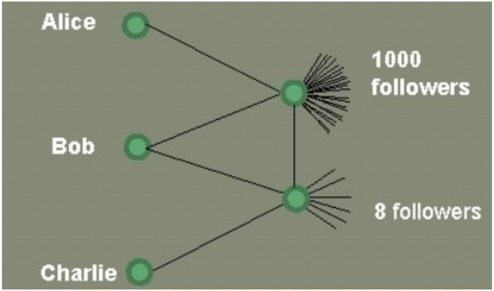
- Shortest path

Different network similarity measures



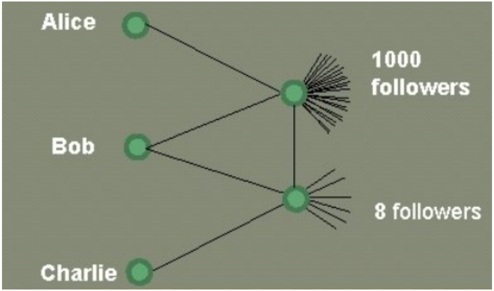
- Common neighbors: $\sum_i A(x,i)A(i,y)$

Different network similarity measures



- Jaccard score: $\dfrac{|N(x) \cap N(y)|}{|N(x) \cup N(y)|}$

Different network similarity measures



- Adamic Adar : $\displaystyle\sum_{u \in N(x) \cap N(y)} \frac{1}{\log |N(u)|}$

Graph embedding: main idea

- Define a mapping of nodes, or an embedding function f.

Graph embedding: main idea

- Define a mapping of nodes, or an embedding function f.
- Define a similarity or proximity in the network, call this $\mathrm{sim}(u, v)$

Graph embedding: main idea

- Define a mapping of nodes, or an embedding function f.
- Define a similarity or proximity in the network, call this $\mathrm{sim}(u, v)$
- Optimize over f such that $\mathrm{sim}(u, v) \approx f(u)^T f(v)$

Random walk based similarity measures

- Learn embedding that preserves random walk based similarity

Random walk based similarity measures

- Learn embedding that preserves random walk based similarity
- "Nearby" nodes in the embedding space should be "close" in the network

Random walk based similarity measures

- Learn embedding that preserves random walk based similarity
- “Nearby” nodes in the embedding space should be “close” in the network
- Define closeness in network by $N_R(u)$, which is a set of nodes visited by some random walk strategy R from u

Random walks based similarity measures

- Maximize $\max_{z} \sum_{u} \log P\,(\text{embeddings of neighbors of } u | z_u)$

Random walks based similarity measures

- Maximize $\max_{z} \sum_{u} \log P\,(\text{embeddings of neighbors of } u | z_u)$

- First thing: what is neighborhood of u
  - Run many short random walks from u
  - Store the nodes that were visited on the walks - a node can be visited multiple times

Random walks based similarity measures

- $\log P\left(\text{embeddings of neighbors of } u|z_u\right) = \sum_{v \in N_R(u)} \log P\left(z_v|z_u\right)$
- $P\left(z_v|z_u\right) = \frac{\exp(z_v^T z_u)}{\sum_i \exp(z_i^T z_u)}$
- This is a distribution over all nodes, and the hope is if v is a neighbor of u, then it is much more similar than those who are not.
- This is computationally extensive, since the computation of the denominator is $O(n)$ for each node.
    - Can be avoided using negative sampling.

Recall BFS (breadth first) and DFS (depth first) ?

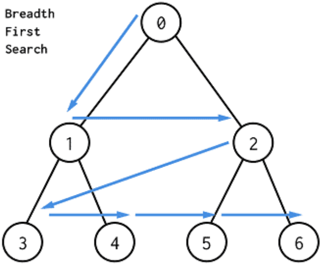


Figure 2: www.freelancinggig.com

Random walk

- Biased second order random walks.

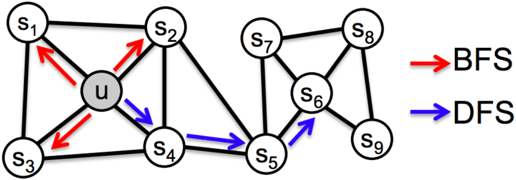


Figure 3: BFS-local traversal. DFS-global traversal. Courtesy - Grover and Leskovec 2016

- Two parameters for biased random walk:
  - p - if high, we do more DFS type traversal than BFS type
  - q - if high, we do more BFS type traversal than DFS type traversal

Random walk

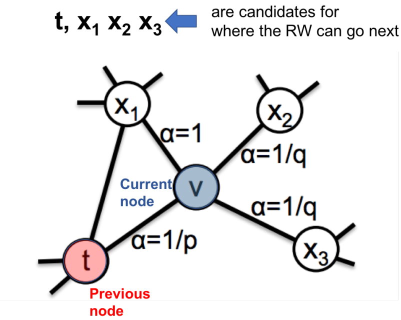


Figure 4: Courtesy - Grover and Leskovec 2016

$$P(x|v) \propto \begin{cases} 1 & x \text{ is 1 hop from } t \\ 1/p & x = t \\ 1/q & x \text{ is 2 hops from } t \end{cases}$$

Random walk

- Simulate r random walks of length ℓ for each node
- Optimize the objective with these $N_R(u)$ neighborhoods using Stochastic Gradient Descent.
- Can parallelize each step.

Real data examples

- Network built from characters in Les Miserable
- It contains 77 nodes corresponding to characters of the novel
- 254 edges connecting two characters whenever they appear in the same chapter.

Real data examples

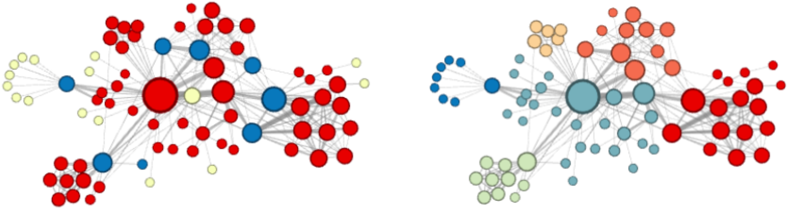


Figure 5: Left: $p = 1, q = 2$ - Structural similarities. Right: $p = 1, q = .5$ - Homophily

Acknowledgments

- Jure leskovec's lecture notes.