Network Science

Institute of Informatics, University of Szeged
Department of Computational Optimization
Lecturer: András London

Lecture 2

Diameter, Average Path Length

- ℓ_{ij} – the shortest path in the network between nodes i and j
- $\Delta = \max_{i,j} \ell_{ij}$ – diameter: the maximum among all shortest paths

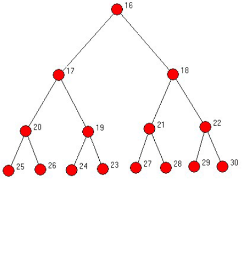


Figure: What are the diameters of an N-node ring and an N-node binary tree?

Average Path Length

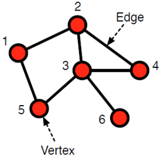
- The average length of shortest paths

$$\langle \ell \rangle = \frac{1}{\binom{n}{2}} \sum_{i,j} \ell_{ij}$$

- Why is it interesting in real networks, and what information does it provide?
- What algorithms are used to compute it?

Degree Distribution

- $A = [a_{ij}] \in \mathbb{R}^{n \times n}$ – adjacency matrix of G:
- Degree of vertex i: $k_i = \sum_{j=1}^{n} a_{ij}$
- Degree distribution: $\mathbb{P}$(degree of a randomly chosen vertex is k)



k	$\Pr(k)$
1	$1/6$
2	$3/6$
3	$1/6$
4	$1/6$

- Why is the degree distribution of a network interesting?
- What degree distributions do real networks follow?

$\longrightarrow$ Key concept, extensively discussed later on.

Which Nodes are "Important" in a Network?

- From a structural standpoint, for example:
  - High degree
  - Centrally located
  - Important for some dynamic process (e.g., spread of infection, random walks)

$\Longrightarrow$ Centrality

- "The more central, the more important; the less central, the less important."
- But how do we "measure" centrality?

Degree Centrality

- **Higher degree $\rightarrow$ more important node**
- $k_i = \sum_{j=1}^{n} a_{ij}$; directed: $k^{in} i = \sum j = 1^n a_{ji}$, $k^{out} i = \sum j = 1^n a_{ij}$

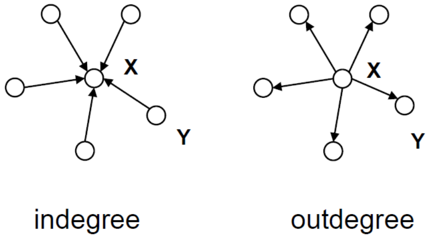


Figure: In-degree and out-degree centrality.

Betweenness Centrality

- Measures how many times a node lies on the shortest path between other nodes, **if one has to go through it**

$$BC(k) = \sum_{i \neq k \neq j} \frac{\sigma_{ij}(k)}{\sigma_{ij}},$$

  where σ_{ij} is the number of shortest paths between i and j, and $\sigma_{ij}(k)$ is the number of those shortest paths passing through k.
- Brandes Algorithm: $O(nm)$ time complexity algorithm to compute BC (m is the number of edges in the graph)

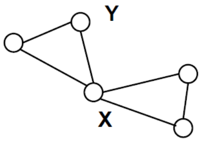


Figure: What are the betweenness values of X and Y?

Closeness Centrality

- Measures how close a node is to all other nodes $\rightarrow$ **average length of the shortest paths** from a node to all other nodes in the network

$$C(i) = \frac{n-1}{\sum_{i \neq j} \ell_{ij}},$$

where ℓ_{ij} is the length of the shortest path between i and j.

- Computation: Floyd-Warshall algorithm

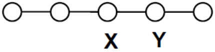


Figure: What are the closeness values of X and Y?

Harmonic Centrality

Two issues with closeness centrality:

- Real networks often have a small diameter $\rightarrow$ closeness centrality values vary within a narrow range.
- Cannot be computed for disconnected networks.

Harmonic Centrality

$$C^h(i) = \frac{1}{n-1} \sum_{j \neq i} \frac{1}{\ell_{ij}},$$

where $\ell_{ij} = \infty$ if there is no path from i to j.

Eigenvector Centrality

- Basic idea: **not all neighbors contribute equally** to the centrality calculation.
- Recursive formula:

$$x_i^{(t+1)} = \sum_{j=1}^{n} w_{ij} x_j^{(t)}$$

"The more important the neighbor, the more it contributes to the centrality of the node."

- Matrix form:

$$A\mathbf{x} = \lambda_1 \mathbf{x},$$

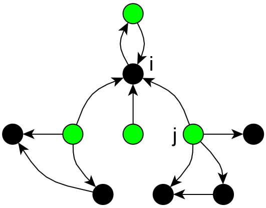
where λ_1 is the largest eigenvalue associated with matrix A (see Perron-Frobenius theorem).

PageRank

- **What if the graph is not connected?** $\Longrightarrow$ "Random surfer", see Google search engine [1]
- Recursion:

$$PR(i) = \frac{1-\lambda}{n} + \lambda \sum_{j \in N^+(i)} \frac{PR(j)}{k^{ki}(j)},$$

where $\lambda \in [0, 1]$ is a parameter (damping factor), $N^+(i)$ is the "in-neighborhood" of node i



[1] Brin & Page, *Computer networks and ISDN systems*, 1998

A Bit of Linear Algebra?

Expressing the PageRank recursion in vector equation form:

$$\mathbf{PR} = \mathbf{PR}R = \mathbf{PR}(\lambda P + (1-\lambda)U)$$

Rearranging this equation, we get:

$$\begin{aligned} \mathbf{PR} &= \mathbf{PR}R = \mathbf{PR}(\lambda P + (1-\lambda)U) = \lambda \mathbf{PR}P + (1-\lambda)\mathbf{PR}U = \\ &= \lambda \mathbf{PR}P + (1-\lambda)\mathbf{PR}\mathbb{1}\mathbb{1}^T \frac{1}{N} = \lambda \mathbf{PR}P + (1-\lambda)\mathbb{1}^T \frac{1}{N} \end{aligned}$$

Using the properties that $U = \mathbb{1}\mathbb{1}^T \frac{1}{N}$ and $\mathbf{PR}\mathbb{1} = 1$. From here, we derive:

$$\mathbf{PR} = \frac{1-\lambda}{N}\mathbb{1}(I - \lambda P)^{-1} = \frac{1-\lambda}{N}\mathbb{1}\sum_{n=0}^{\infty}(\lambda P)^n$$

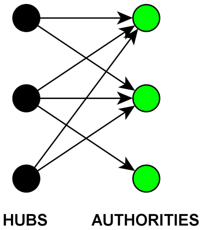
PageRank Algorithm

Input: Directed graph G
Output: PageRank score vector

1: Initialize $\mathbf{PR}0 = \frac{\lambda}{N}\mathbb{1}$
2: $k = 1$
3: **repeat**
4: $\quad \mathbf{PR}k + 1 := \frac{\lambda}{N}\mathbb{1} + \lambda A D^1 \mathbf{PR}k$
5: $\quad k = k + 1$
6: **until** $||\mathbf{PR}k + 1 - \mathbf{PR}_k||1$
7: return $\mathbf{PR}k + 1$

HITS (Hyperlink Induced Topic Search)

- Developed by Kleinberg[2], it's a refined version of the original PageRank algorithm.
- When ranking the nodes of the graph, it distinguishes between Hub and Authority nodes
  - A good Authority node is one that is pointed to by many links.
  - A good Hub node is one that points to many good Authority nodes.



[2] Kleinberg, *Journal of the ACM*, 1999

HITS Algorithm

Input: Directed graph G
Output: Hub and Authority scores for the nodes

Initially, set every node's score to 1
repeat
  for all hub $i \in H$ **do**
    $h_i = \sum_{j \in F(i)} a_j$ $\{F(i)$: *nodes pointing to* $i\}$
  end for
  for all authority $i \in A$ **do**
    $a_i = \sum_{j \in B(i)} h_j$ $\{B(i)$: *nodes pointed to by* $i\}$
  end for
until convergence
Normalize scores

Some Free Software

Network Visualization and Analysis

- Cytoscape (GUI)
- Gephi (GUI)
- iGraph (R, C++, Python)

Tasks:

- Centralit analysis and visualization of a network (e.g., the Zachary's karate club).
- Reflect on PageRank and HITS in matrix equation form.

Further Reading

- Jackson's book, Chapter 2