SOCIAL NETWORK ANALYTICS

Centrality Measures

Degree Centrality, Closeness Centrality, Betweenness Centrality and Eigenvector Centrality

Prakash C O

Department of Computer Science and Engineering

SOCIAL NETWORK ANALYTICS

Centrality Measures

Degree Centrality, Closeness Centrality, Betweenness Centrality and Eigenvector Centrality

Prakash C O

Department of Computer Science and Engineering

Centrality Measures

- **Centrality refers to a group of metrics that aim to quantify the**
  - **"importance"** or
  - **"influence"** or
  - **“power”** or
  - **other individual characteristics** (in a variety of senses)

  of a particular node (or group) **within a network**.

- **One of the first questions one asks when looking at a social network is**
  - Who is more important/influential in the network?
  - Who has the power?
  - Who is at the heart of a social network?
  - Who connects many social circles?

Centrality Measures

- In graph theory and network analysis, indicators of **centrality** identify the most important vertices within a graph/network.
- **Applications of centrality** include identifying
  - the most influential person(s) in a social network,
  - key infrastructure nodes in the Internet or urban networks, and
  - super-spreaders of disease.
- Centrality indicates one type of “importance” of actors in a network: in lay terms, these are the “key” players.

Centrality Measures

Centrality measure	Interpretation in social networks
Degree	How many people can this person reach directly?
Betweenness	How likely is this person to be the most direct route between two people in the network?
Closeness	How fast can this person reach everyone in the network?
Eigenvector	How well is this person connected to other well-connected people?

SOCIAL NETWORK ANALYTICS

Degree Centrality

Prakash C O
Department of Computer Science and Engineering

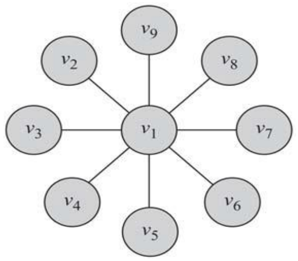
Degree Centrality

- **Degree:**
  - **The degree of a node in a network is the number of connections or edges the node has to other nodes.**
  - If a network is directed, nodes have two different degrees,
    - the **in-degree**, which is the number of incoming edges, and
    - the **out-degree**, which is the number of outgoing edges.
- **The assumption in Degree Centrality:**
  - **important nodes have many connections.**

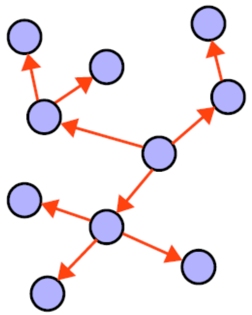
The most basic measure of centrality is:

- the number of neighbors/connections/edges.

- **Undirected networks: use degree**
- **Directed networks: use in-degree or out-degree**



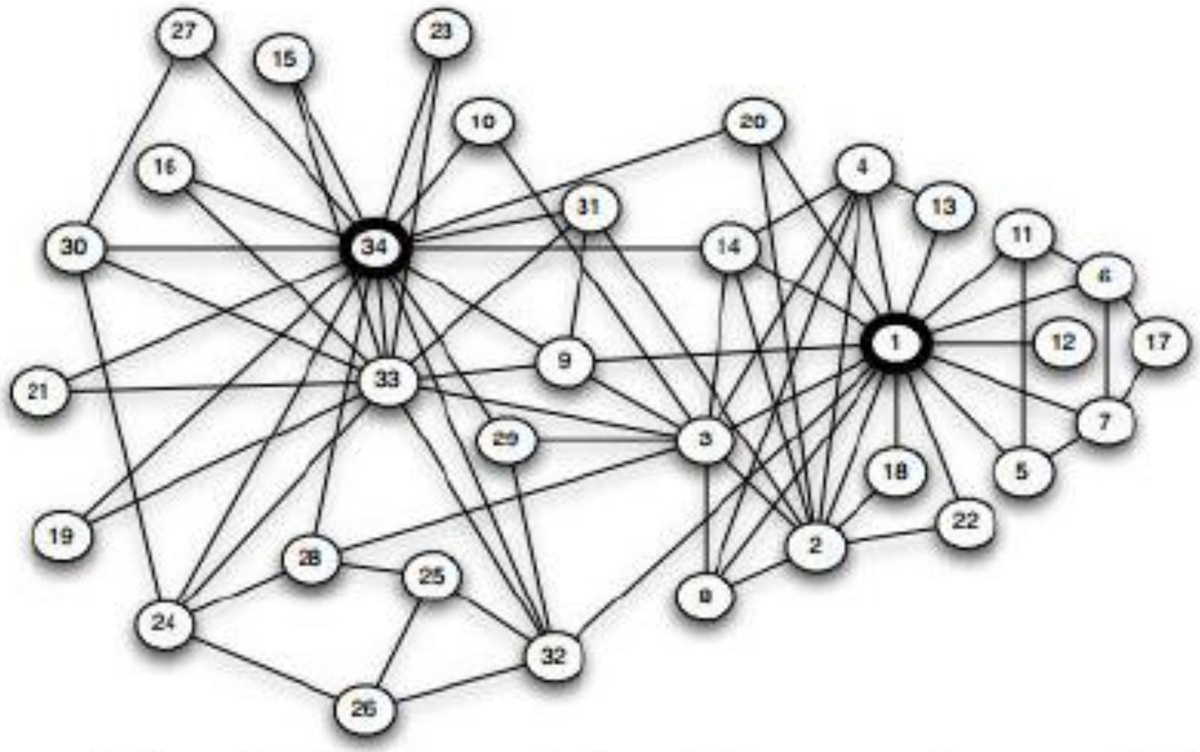
Undirected graph



Directed graph

Degree Centrality

- Based on the structure of the network, which are the 5 most important node in the Karate Club friendship network?

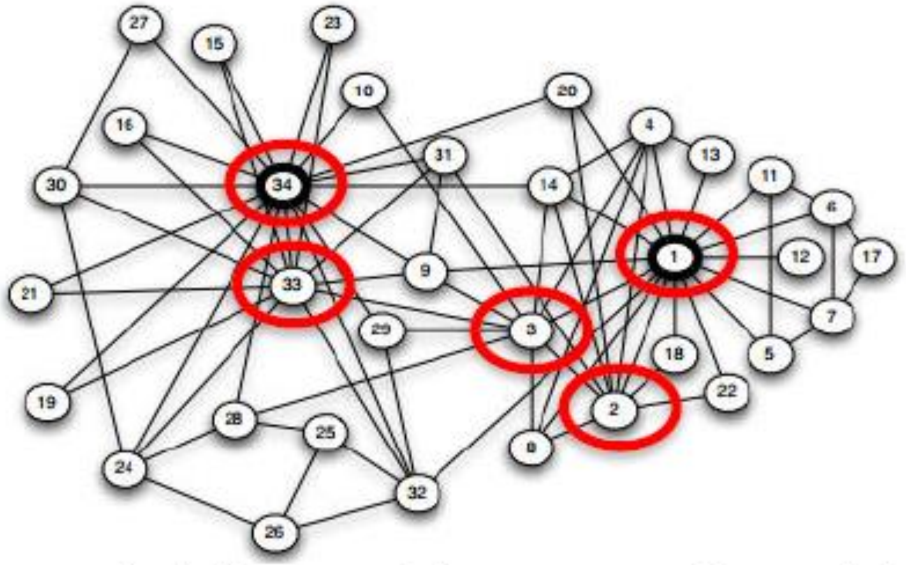


Friendship network in a 34-person karate club [Zachary 1977]

Degree Centrality

Solution:

- Different ways of thinking about “node importance”.
  - Node importance: **Degree(number of friends)**
  - 5 most important nodes are: 34, 1, 33, 3, 2



Friendship network in a 34-person karate club
[Zachary 1977]

Degree Centrality

- **Degree Centrality was proposed by Linton C. Freeman in his 1979 paper "Centrality in Social Networks Conceptual Clarification".**

- **The Degree Centrality algorithm can be used to find the popularity of individual nodes.**
  The Degree Centrality is often used as part of a global analysis where we calculate the minimum degree, maximum degree, mean degree, and standard deviation across the whole graph.

Degree Centrality

Application: Finding "Celebrities" in Communities.

- Every community has its own Virat Kohli, Ratan Tata, Amitabh bachchan, Sudha Murthy—people who are significantly more popular then others. There are usually very few of them, and they are orders of magnitude more popular then everyone else.

- The first and simplest metric that we use will help us find these local celebrities. This metric is called **degree centrality**.

Degree Centrality

Application: Finding "Celebrities" in Communities.

- **A node degree is simply the number of connections that a node has.**
  - On Twitter, it's the number of followers;
  - On Facebook, it's the number of friends; and
  - On Reddit, it could be interpreted as the number of upvotes ("link karma").

Degree Centrality

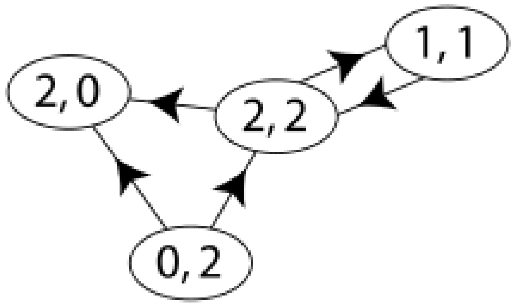
- **The degree centrality $C_d(v_i)$ for node v_i in an undirected graph is**

$$C_d(v_i) = d_i$$

where d_i is the degree (number of adjacent edges) of node v_i.

- **In directed graphs**, we can either use the in-degree, the out-degree, or the combination as the degree centrality value:

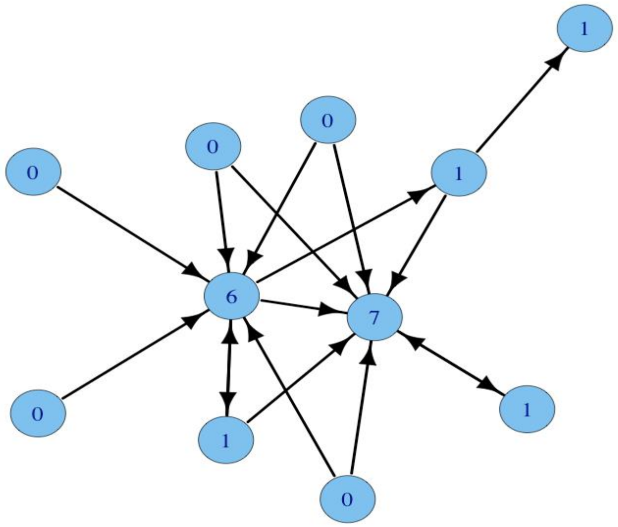
$$\begin{aligned} C_d(v_i) &= d_i^{\text{in}} && (prestige), \\ C_d(v_i) &= d_i^{\text{out}} && (gregariousness), \\ C_d(v_i) &= d_i^{\text{in}} + d_i^{\text{out}}. \end{aligned}$$



A directed graph with vertices labeled (indegree, outdegree)

Degree Centrality

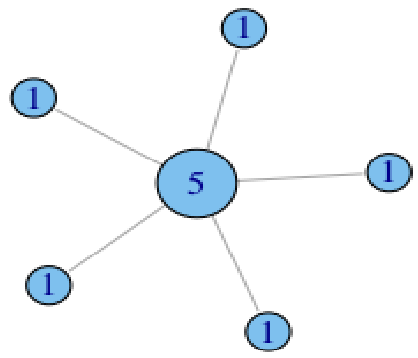
- **When using in-degrees, it measures how popular a node is and its value shows prominence or prestige.**



- **When using out-degrees, it measures the gregariousness of a node.**

Degree Centrality

- In social networks, Degree centrality measures
  - How many people can this person knows/reach directly?



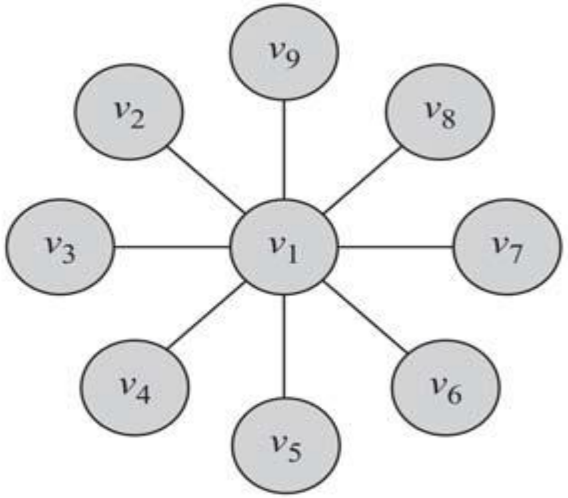
- Degree centrality measures number of nodes at distance one.

- In ego-centric networks, degree centrality is the **sum of all other actors who are directly connected to ego.**

- In degree centrality, each neighbor contributes equally to centrality.
- Degree Centrality **signifies activity or popularity**.

Degree Centrality

➢ **Problem 1:** Find degree centrality of all the nodes in the graph.



Normalized Degree Centrality

- The degree centrality measure does not allow for centrality values to be compared across networks (e.g., Facebook and Twitter).
  To overcome this problem, we can normalize the degree centrality values.

- **Normalized Degree Centrality** (Undirected Graph)
  - Simple normalization methods include normalizing by the maximum possible degree

$$C_d^{\text{norm}}(v_i) = \frac{d_i}{n-1}$$

where n is the number of nodes.

Normalized Degree Centrality

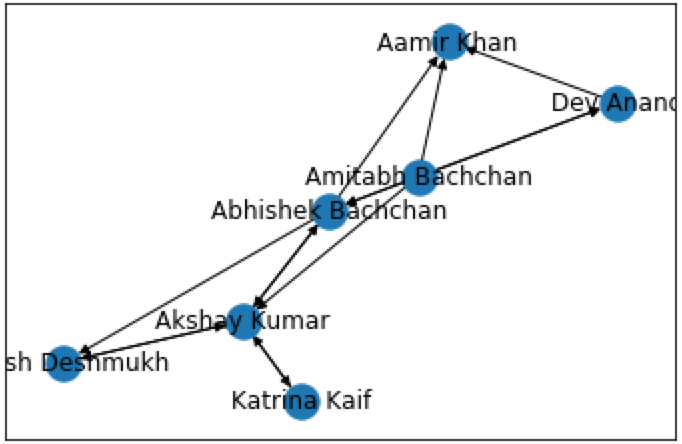
➢ Normalized Degree Centrality (Directed Graph)

- Simple normalization methods include normalizing by the maximum possible degree

$$C_{indeg}^{norm}(v_i) = \frac{d_i^{indeg}}{n-1}$$

$$C_{outdeg}^{norm}(v_i) = \frac{d_i^{outdeg}}{n-1}$$

where n is the number of nodes.

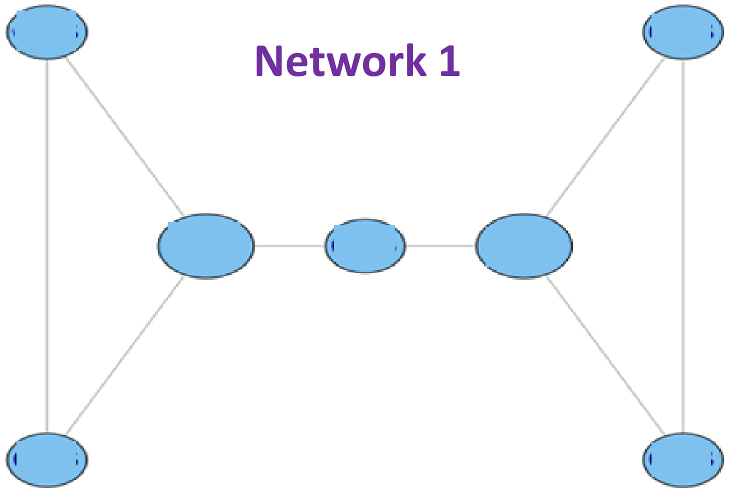



```
>>>indegCent = nx.in_degree_centrality(G)
>>>print(indegCent['Akshay Kumar'])  #(4/6)
>>>outdegCent = nx.out_degree_centrality(G)
>>>print(outdegCent["Akshay Kumar"])  #(3/6)
```

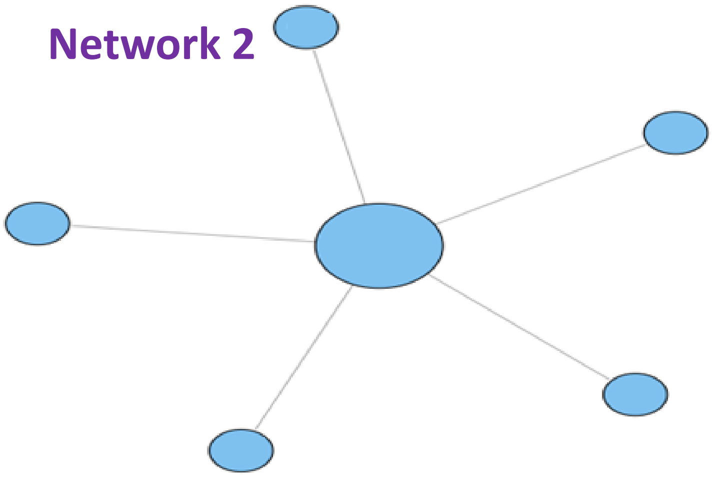
Normalized Degree Centrality

- **Problem 2:** Find normalized degree centrality of the nodes in the following graphs.

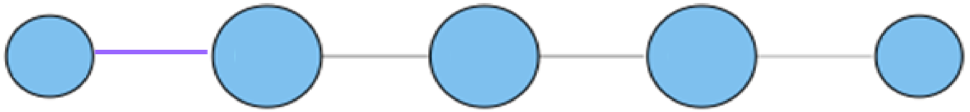
Network 1



Network 2



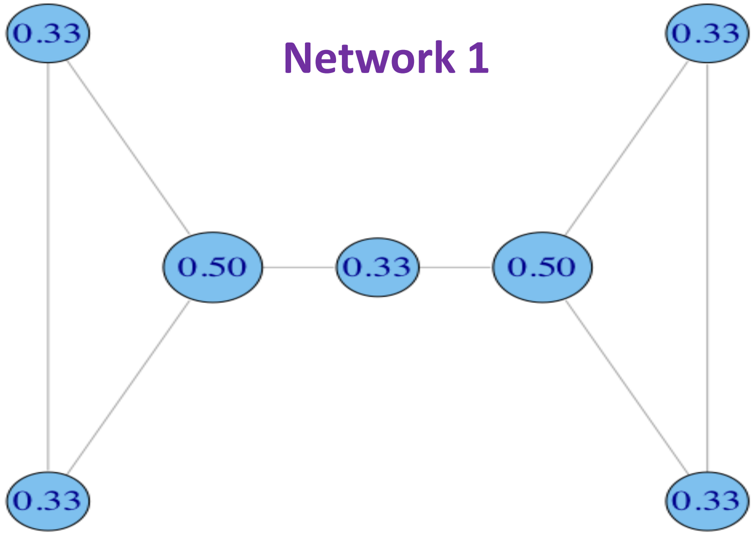
Network 3



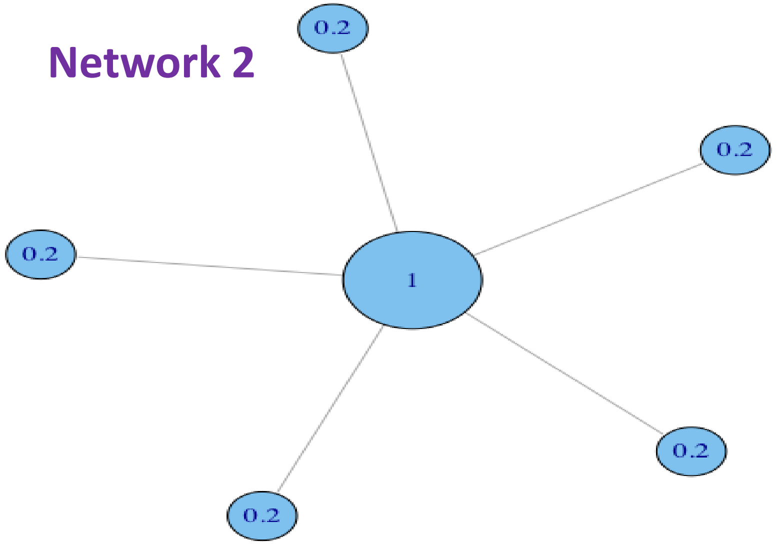
Normalized Degree Centrality

➢ **Problem 2:** Find normalized degree centrality of the nodes in the following graphs. **Solution:**

Network 1



Network 2



Network 3

Degree Centrality – Use cases

When to use the Degree Centrality algorithm

- **Degree centrality is an important component of any attempt to determine the most important people on a social network.**

  For example, in BrandWatch's most influential men and women on Twitter 2017 the top 5 people in each category have over 40m followers each.

- **Weighted degree centrality has been used to help separate fraudsters from legitimate users of an online auction.**

  The weighted centrality for fraudsters is significantly higher because they tend to collude with each other to artificially increase the price of items. Read more in Two Step graph-based semi-supervised Learning for Online Auction Fraud Detection.

Degree Centrality -Assignment

- **Find Top 10 Most popular Facebook fan pages, based on number of fans.**
- **Find Top 20 the Most Followed Accounts on Twitter.**
- **Find Top 10 popular people in the LiveJournal network.**

Degree Centrality

Demo using NetworkX

- DegreeCentrality.py
- In-Out-DegreeCentrality.py

Demo using Gephi

Demo using UCINET

SOCIAL NETWORK ANALYTICS

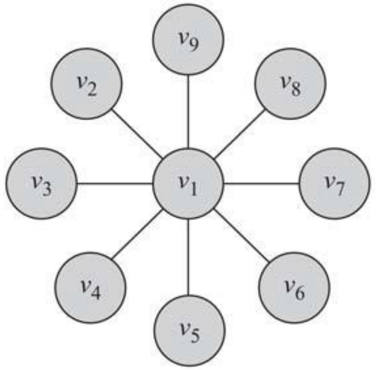
Closeness Centrality

Prakash C O

Department of Computer Science and Engineering

Closeness Centrality

- **Closeness centrality is based on the notion of distance.**
- If an actor is close to all others in the network, a distance of no more than one, then she or he is not dependent on any other to reach everyone in the network.



- **Closeness measures independence or efficiency**.
- With disconnected networks, closeness centrality must be calculated for each component.

Closeness Centrality

- **In social networks,**
  - **Closeness centrality (indicates who is at the heart of a social network)**
  - **Closeness centrality measures the proximity of an entity to the other entities in the social network.**

Closeness Centrality

- In closeness centrality, the question is how close is a node to all other nodes in a network.

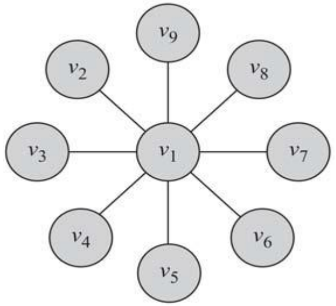
- **Closeness centrality measure is very important and commonly used in Social Network Analysis.**

  This closeness measure can be viewed as a time required to spread information from a given node to all other reachable nodes in a network.

- Closeness centrality is a **way of detecting nodes that are able to spread information very efficiently through a graph**.

Closeness Centrality

- The closeness centrality of a node measures its average farness (inverse distance) to all other nodes.
- Nodes with a high closeness score have the shortest distances to all other nodes.
- More closeness value, then its more easy(and faster) to spread information/technology/diseases/gossips in a network/community.



Closeness Centrality

- **Closeness centrality** of a node is calculated as **the reciprocal of the *sum of the length of the shortest paths between the node and all other nodes in the graph*.**
  Thus, the more central a node is, the *closer* it is to all other nodes.

$$C_c(i) = \left[\sum_{j=1}^{N} d(i,j)\right]^{-1} \qquad C_c(i) = \frac{1}{\sum_{j=1}^{N} d(i,j)}$$

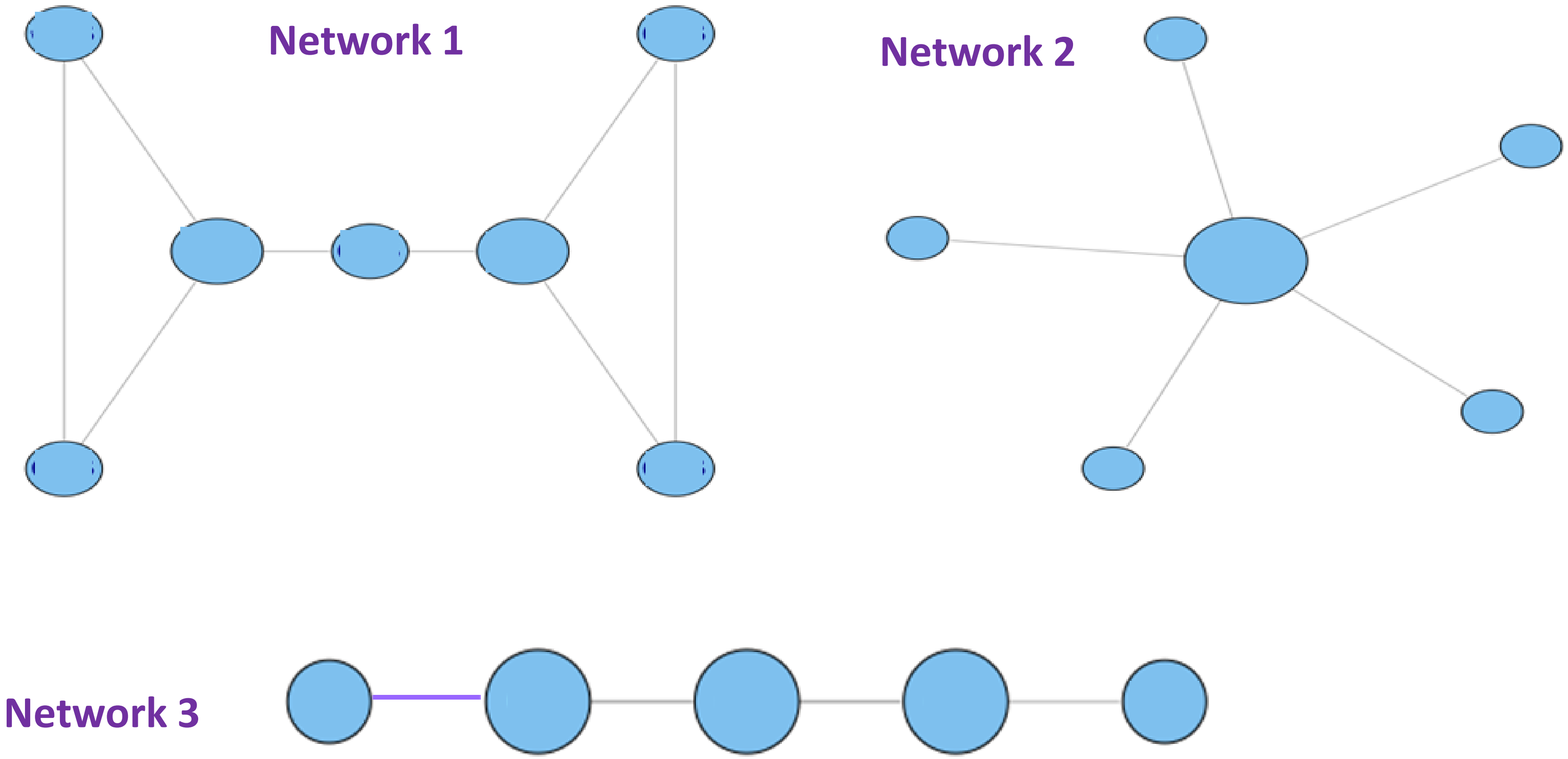
Usually normalized by:

$$C'_C(i) = \left[\frac{\sum_{j=1}^{N} d(i,j)}{(N-1)}\right]^{-1} = \left[\frac{(N-1)}{\sum_{j=1}^{N} d(i,j)}\right]$$

In closeness, we are taking inverse bcoz we want closeness to go high when avg shortest path length goes down.

Normalized Closeness Centrality

➢**Problem 1:** Find **normalized closeness centrality** of the nodes in the following networks.



Normalized Closeness Centrality

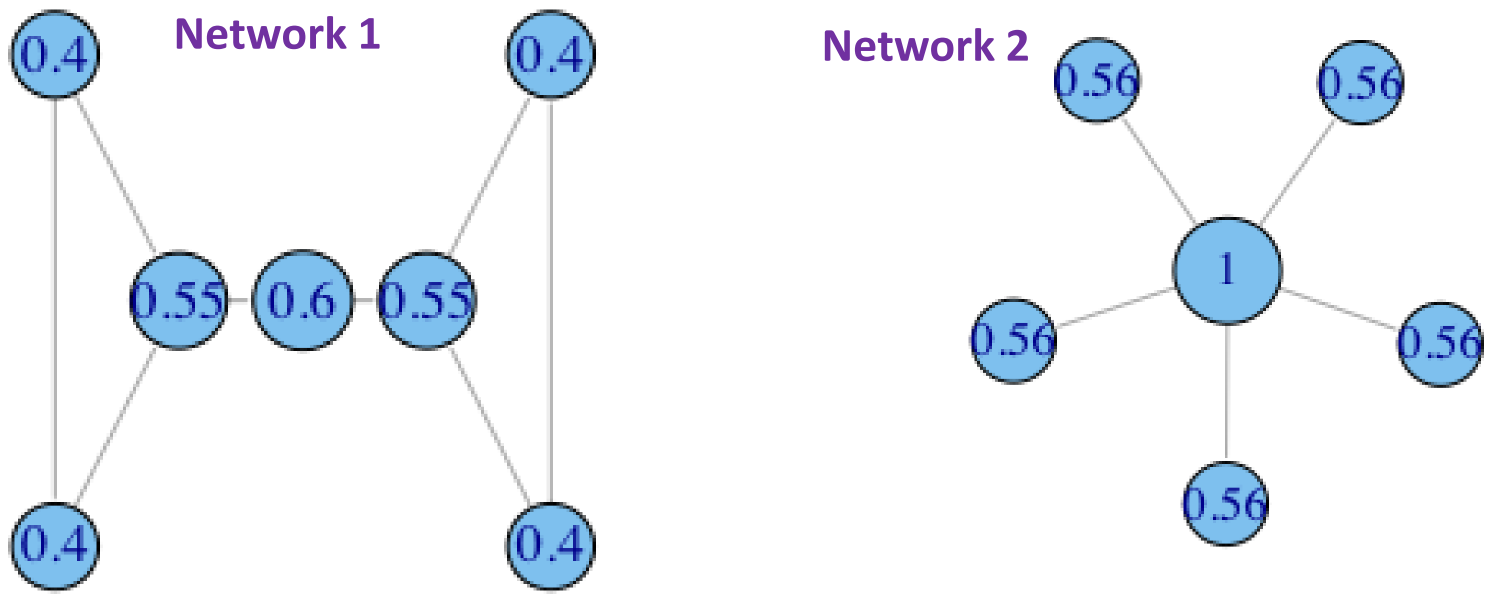
➢Problem 1:

Network 3

$$C_c'(A) = \left[\frac{\sum_{j=1}^{N} d(A,j)}{N-1} \right]^{-1} = \left[\frac{1+2+3+4}{4} \right]^{-1} = \left[\frac{10}{4} \right]^{-1} = 0.4$$

Normalized Closeness Centrality

➢Problem 1:



The measure will reach its maximum for a given network size when an actor is directly connected to all others in the network and its minimum when an actor is not connected to any others.

Closeness Centrality – Use cases

When to use the Closeness Centrality algorithm

- Closeness centrality is used to research organizational networks, where individuals with high closeness centrality are in a favourable position to control and acquire vital information and resources within the organization.
  One such study is "Mapping Networks of Terrorist Cells" by Valdis E. Krebs.

- Closeness centrality has been used to estimate the importance of words in a document, based on a graph-based keyphrase extraction process.
  This process is described by Florian Boudin in "A Comparison of Centrality Measures for Graph-Based Keyphrase Extraction“.

Closeness Centrality – Use cases

When to use the Closeness Centrality algorithm

- Closeness centrality can be interpreted as an estimated time of arrival of information flowing through telecommunications or package delivery networks where information flows through shortest paths to a predefined target.

It can also be used in networks where information spreads through all shortest paths simultaneously, such as infection spreading through a social network.

Find more details in "Centrality and network flow" by Stephen P. Borgatti.

Demo using NetworkX

- ClosenessCentrality.py

Demo using Gephi

Closeness Centrality - Assignment

- Find the possible top 10 quick Gossipmongers in the LiveJournal network

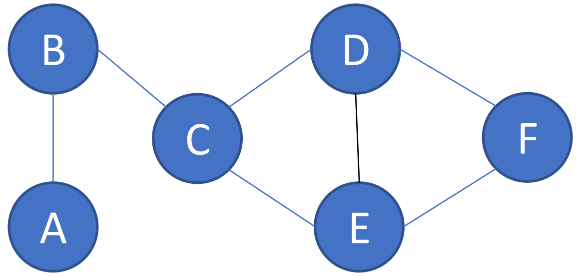
SOCIAL NETWORK ANALYTICS

Betweenness Centrality

Prakash C O

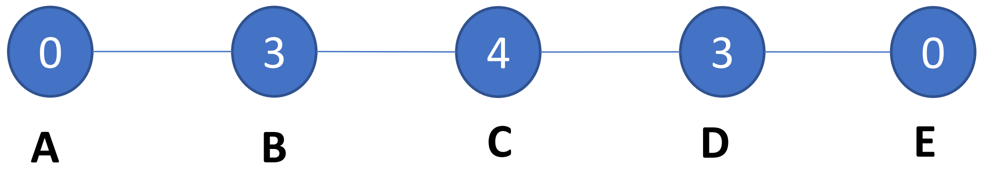
Department of Computer Science and Engineering

Betweenness Centrality



- Betweenness centrality quantifies the number of times a node acts as a bridge along the shortest path between two other nodes.
- **A node is important if it lies in many shortest-paths**
  - so it is essential in passing information through the network
- Betweenness centrality finds the Communication Bottlenecks and/or Community Bridges.

Betweenness Centrality



- In the above figure
  - **Node A lies between no two other nodes**
  - **Node B lies between A and three other nodes : C, D, and E**
  - **Node C lies between 4 pairs of nodes (A,D), (A,E), (B,D), (B,E).**

    Note that there are no alternative paths for these pairs to take, so C gets full credit.

Betweenness Centrality:

$$C_B(i) = \sum_{j \neq k} g_{jk}(i) / g_{jk}$$

Where g_{jk} (i)= the number of shortest paths connecting *jk* passing through i   **Note: i != j and i != k**

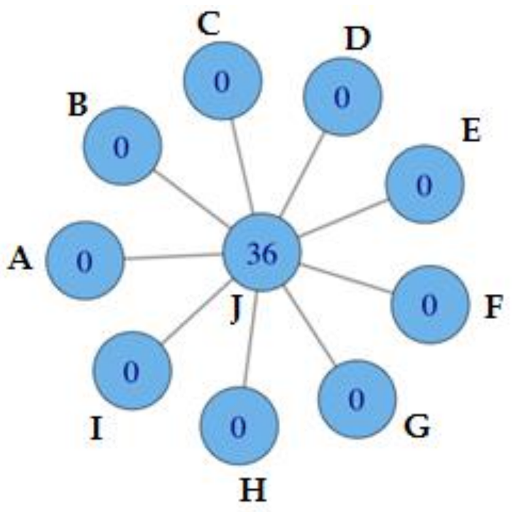
g_{jk} = total number of shortest paths **between nodes j and k**

Endpoints: we can either include or exclude node i as nodes j and k in the computation of C_B(i) .

Betweenness Centrality

Example 1: Exclude node i as nodes j and k in the computation of $C_B(i)$.

Note: Nodes that most frequently lie on the shortest paths will have a higher betweenness centrality score.



In Star Network, N=10,

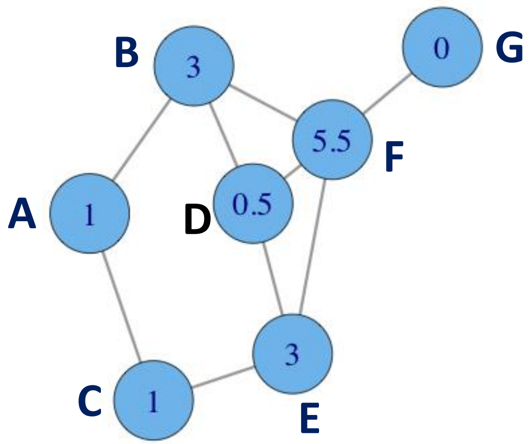
$C_B(J)$ =

$[(1/1)_{AB}+(1/1)_{AC}+(1/1)_{AD}+(1/1)_{AE}+(1/1)_{AF}+(1/1)_{AG}+(1/1)_{AH}+(1/1)_{AI}]+$
$[(1/1)_{BC}+(1/1)_{BD}+(1/1)_{BE}+(1/1)_{BF}+(1/1)_{BG}+(1/1)_{BH}+(1/1)_{BI}]+$
$[(1/1)_{CD}+(1/1)_{CE}+(1/1)_{CF}+(1/1)_{CG}+(1/1)_{CH}+(1/1)_{CI}]+$
$[(1/1)_{DE}+(1/1)_{DF}+(1/1)_{DG}+(1/1)_{DH}+(1/1)_{DI}]+$
$[(1/1)_{EF}+(1/1)_{EG}+(1/1)_{EH}+(1/1)_{EI}]+$
$[(1/1)_{FG}+(1/1)_{FH}+(1/1)_{FI}]+$
$[(1/1)_{GH}+(1/1)_{GI}]+$
$[(1/1)_{HI}] = 36$

Each node receives a score, based on the number of shortest paths that pass through the node.

Example 2:

Note: Nodes that most frequently lie on the shortest paths will have a higher betweenness centrality score.



N=7,

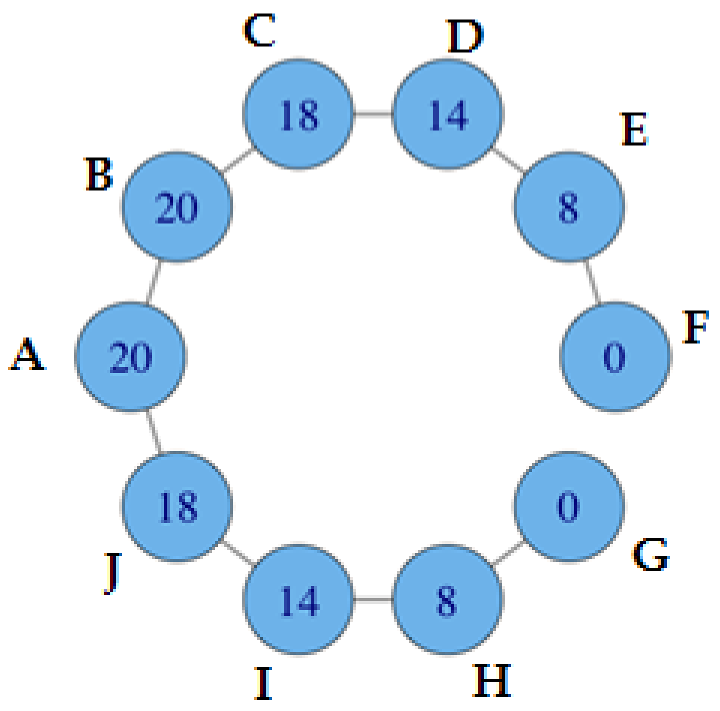
$C_B(F)$= [$(1/1)_{AG}$+ $(1/1)_{BG}$+$(1/1)_{CG}$+ $(1/1)_{DG}$+ $(1/1)_{EG}$+ $(1/2)_{BE}$] = **5.5**

$C_B(D)$= $(1/2)_{BE}$ = **0.5**

$C_B(E)$= [$(1/1)_{CD}$+ $(1/1)_{CF}$+$(1/1)_{CG}$] = **3**

$C_B(A)$= $(1/1)_{BC}$ = **1**

Example 3:

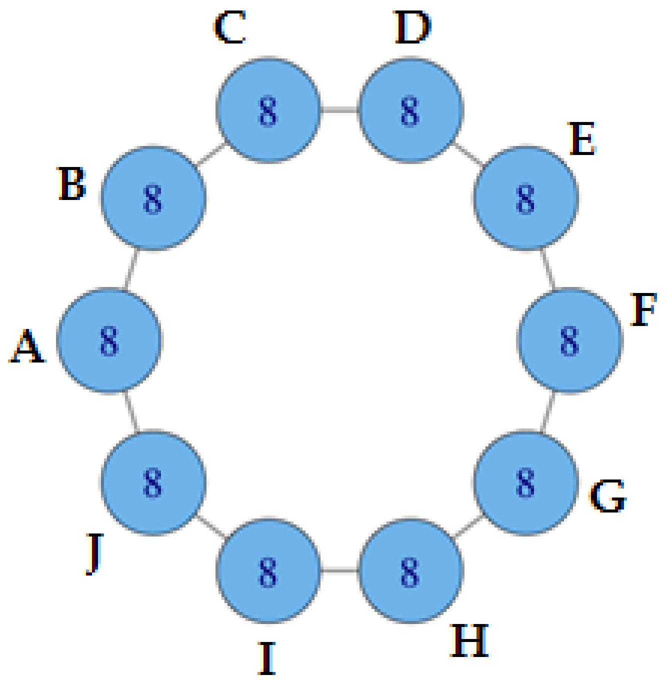


In linear Network, N=10,

$C_B(D)$= $[(0/1)_{FE}+ (1/1)_{FC}+(1/1)_{FB}+ (1/1)_{FA}+ (1/1)_{FJ}+ (1/1)_{FI}+ (1/1)_{FH}+ (1/1)_{FG}] + [(1/1)_{EC}+(1/1)_{EB}+ (1/1)_{EA}+ (1/1)_{EJ}+ (1/1)_{EI}+ (1/1)_{EH}+ (1/1)_{EG}] =$ **14**

$C_B(C)$= $[(0/1)_{FE}+ (0/1)_{FD}+(1/1)_{FB}+ (1/1)_{FA}+ (1/1)_{FJ}+ (1/1)_{FI}+ (1/1)_{FH}+ (1/1)_{FG}] + [(0/1)_{ED}+ (1/1)_{EB}+ (1/1)_{EA}+ (1/1)_{EJ}+ (1/1)_{EI}+ (1/1)_{EH}+ (1/1)_{EG}] + [(1/1)_{DB}+ (1/1)_{DA}+ (1/1)_{DJ}+ (1/1)_{DI}+ (1/1)_{DH}+ (1/1)_{DG}] =$ **18**

Example 4:



In Ring Network, N=10,

$C_B(A)=$

$[(1/1)_{BJ}+(1/1)_{BI}+(1/1)_{BH}+(1/2)_{BG}+(0/1)_{BF}+(0/1)_{BE}+(0/1)_{BD}+(0/1)_{BC}]+$

$[(1/1)_{CJ}+(1/1)_{CI}+(1/2)_{CH}+(0/1)_{CG}+(0/1)_{CF}+(0/1)_{CE}+(0/1)_{CD}]+$

$[(1/1)_{DJ}+(1/2)_{DI}+(0/1)_{DH}+(0/1)_{DG}+(0/1)_{DF}+(0/1)_{DE}]+$

$[(1/2)_{EJ}+(0/1)_{EI}+(0/1)_{EH}+(0/1)_{EG}+(0/1)_{EF}]+$

$[(0/1)_{FJ}+(0/1)_{FI}+(0/1)_{FH}+(0/1)_{FG}]+$

$[(0/1)_{GJ}+(0/1)_{GI}+(0/1)_{GH}]+$

$[(0/1)_{HJ}+(0/1)_{HI}]+$

$[(0/1)_{IJ}] = 8$

Betweenness Centrality - Use cases

- **Betweenness centrality finds wide application in network theory;** it represents the degree to which nodes stand between each other.
- **For example**, in a telecommunications network, a node with higher betweenness centrality would have more control over the network, because more information will pass through that node.

Betweenness Centrality - Use cases

When to use the Betweenness Centrality algorithm

- **Betweenness centrality is used to research the network flow in a package delivery process, or telecommunications network.**
  These networks are characterized by traffic that has a known target and takes the shortest path possible.
  This, and other scenarios, are described by Stephen P. Borgatti in "Centrality and network flow".

- **Betweenness centrality can be used to help microbloggers spread their reach on Twitter, with a recommendation engine that targets influencers that they should interact with in the future.**
  This approach is described in "Making Recommendations in a Microblog to Improve the Impact of a Focal User".

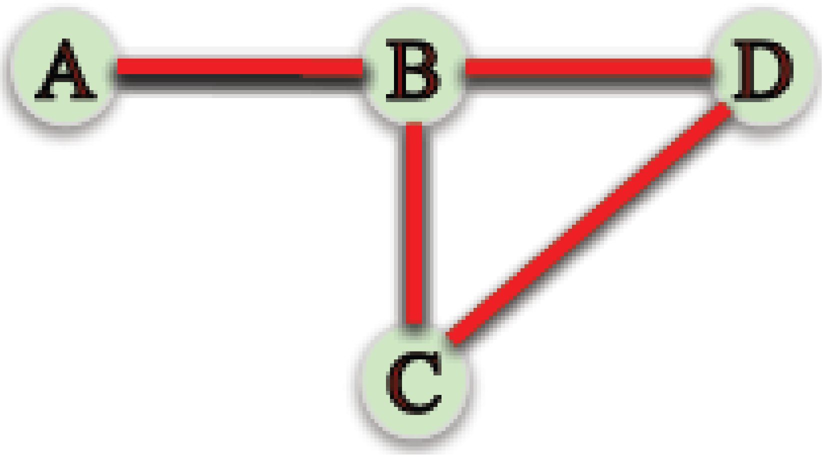
Betweenness Centrality - Use cases

When to use the Betweenness Centrality algorithm

- Betweenness centrality is used to identify influencers in legitimate, or criminal, organizations.
  Studies show that influencers in organizations are not necessarily in management positions, but instead can be found in brokerage positions of the organizational network.
  Removal of such influencers could seriously destabilize the organization.
  More detail can be found in "**Brokerage qualifications in ringing operations**", by Carlo Morselli and Julie Roy.

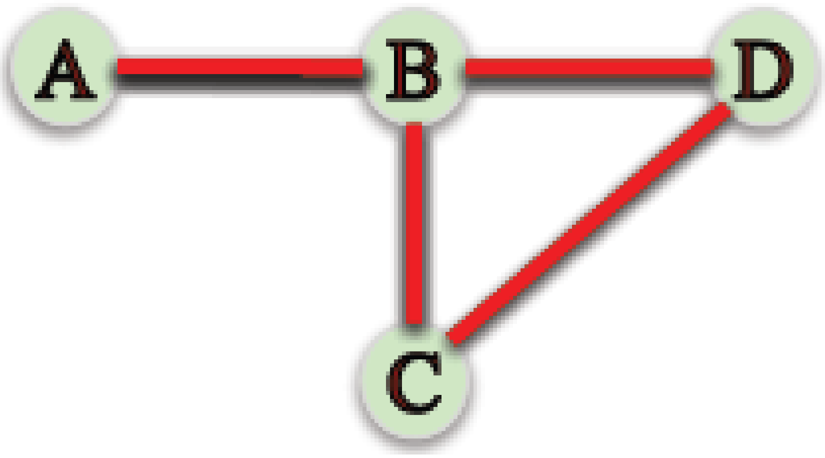
Exercise 1: **Compute Betweenness Centrality of node B**

(with and without including B as an endpoint)



Exercise 1: Compute Betweenness Centrality of node B

(with and without including B as an endpoint)

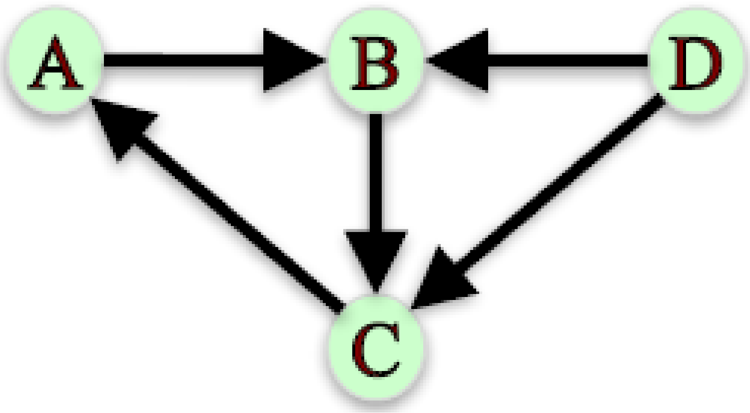


If we **exclude node B**, we have: $C_B(B) = \mathbf{2}$ $[(1/1)_{AC} + (1/1)_{AD} + (0/1)_{DC}]$

If we **include node B**, we have: $C_B(B) = \mathbf{5}$ $[(1/1)_{AB} + (1/1)_{AC} + (1/1)_{AD} + (1/1)_{CB} + (1/1)_{DB} + (0/1)_{DC}]$

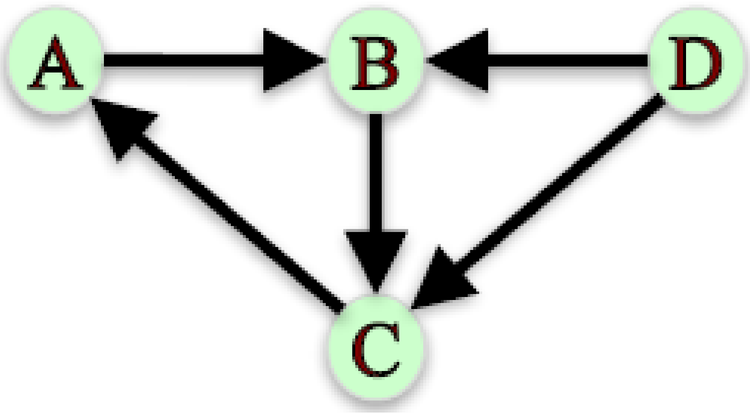
Exercise 2: **Compute Betweenness Centrality of nodes B and C**

(without including them as an endpoint)



Exercise 2: **Compute Betweenness Centrality of nodes B and C**

(without including them as an endpoint)



$C_B(B) = 1 \quad [\,(1/1)_{AC} + (0/1)_{CA} + (0/1)_{DC} + (0/1)_{DA}]$

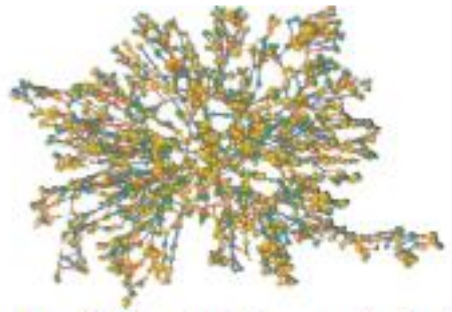
$C_B(C) = 2 \quad [\,(0/1)_{AB} + (1/1)_{BA} + (0/1)_{DB} + (1/1)_{DA}]$

Normalized Betweenness Centrality

- **Normalization:**
  Betweenness centrality values will be larger in graphs with many nodes. To control for this, we divide centrality values by the number of pairs of nodes in the graph (excluding i):

$$C_B'(i) = C_B(i)\boxed{/[(n-1)(n-2)/2]}$$

number of pairs of vertices
excluding the vertex itself



Network of friendship, marital tie, and family tie among 2200 people [Christakis & Fowler 2007]

Note: $[(n-1)(n-2)]/2$ in undirected graphs

$[(n-1)(n-2)]$ in directed graphs

Betweenness Centrality

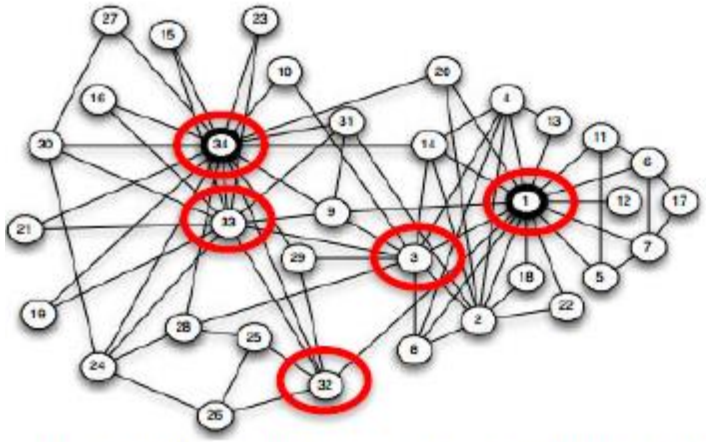
Demo using NetworkX: BetweennessCentrality.py

```
import networkx as nx
G = nx.karate_club_graph()
G = nx.convert_node_labels_to_integers(G,first_label=1)
nx.draw_networkx(G)

btwnCent = nx.betweenness_centrality(G,normalized=True,
  endpoints = False)
import operator
print(sorted(btwnCent.items(),key=operator.itemgetter(1),
  reverse = True)[0:5])
```

Output:

- [(**1**, 0.43763528138528146), (**34**, 0.30407497594997596), (**33**, 0.14524711399711 4), (**3**, 0.14365680615680618), (**32**, 0.13827561327561325)]



Friendship network in a 34-person karate club
[Zachary 1977]

SOCIAL NETWORK ANALYTICS

Eigenvector Centrality

Prakash C O

Department of Computer Science and Engineering

Eigenvector Centrality

- Eigenvector Centrality was proposed by Phillip Bonacich, in his 1986 paper ”Power and Centrality: A Family of Measures”.
- Eigenvector Centrality was the first of the centrality measures that considered the transitive importance of a node in a graph, rather than only considering its direct importance.
- **Eigenvector centrality measures a node’s importance while giving consideration to the importance of its neighbors.**

  For example, A node with 300 relatively unpopular friends on Facebook would have lower eigenvector centrality than someone with 300 very popular friends (like Narendra Modi, Amitabh Bachchan, Virat Kohli).

Eigenvector Centrality

- **Its a measure of the influence of a node in a network.**
- In Eigenvector Centrality, **relative scores are assigned to all nodes in the network based on the concept** that **connections from high-scoring nodes contribute more to the score of the node** in question than equal connections from low-scoring nodes.
- A **high eigenvector score** means that a node is connected by nodes who themselves have high scores.

  A node is important if it is linked to by other important nodes.

- Google's PageRank is a variant of the eigenvector centrality measure.

Eigenvector Centrality

- **Eigenvector centrality differs from in-degree centrality**:
  - **A node receiving many links does not necessarily have a high eigenvector centrality** (it might be that all linkers have low or null eigenvector centrality).
  - **A node with high eigenvector centrality is not necessarily highly linked** (the node might have few but important linkers).

Eigenvector Centrality to rank web pages

The basic idea.

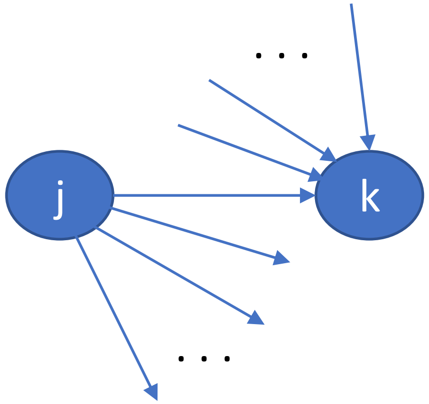
- A core idea in assigning a score to any given web page is that **the page's score is derived from the links made to that page from other web pages**.

- The **links to a given page** are called the **backlinks** for that page.
  - **A link to page k becomes a vote for page k's importance.**
  - **A link to page k from an important page should boost page k's importance score more than a link from an unimportant page.**

- The web thus becomes a democracy where pages vote for the importance of other pages by linking to them.

Eigenvector Centrality to rank web pages

- If **page j contains n_j links to other pages**, one of which links to page k, then we will boost page k's score by x_j/n_j , rather than by $\mathbf{x_j}$.
- To quantify this for a web of N pages, let $\mathbf{L_k} \subset \{1, 2, \ldots, m\}$ denote the set of pages with a link to page k, that is, **L_k is the set of page k's backlinks**.
- **For each page k we require**



$$x_k = \sum_{j \in L_k} \frac{x_j}{n_j},$$

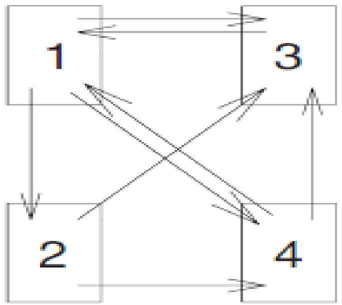
where n_j is the number of outgoing links from page j

Eigenvector Centrality to rank web pages

$$x_k = \sum_{j \in L_k} \frac{x_j}{n_j},$$

- Let's apply this approach to the four-page web of Figure below.
  - For page 1: $\mathbf{x_1 = x_3/1 + x_4/2}$
  - For page 2: $\mathbf{x_2 = x_1/3,}$
  - For page 3: $\mathbf{x_3 = x_1/3 + x_2/2 + x_4/2}$, and
  - For page 4: $\mathbf{x_4 = x_1/3 + x_2/2}$.
- These linear equations can be written **Ax**, where $x = [x_1\ x_2\ x_3\ x_4]^T$

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 1 & \frac{1}{2} \\ \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{2} & 0 & 0 \end{bmatrix}$$



Eigenvector Centrality to rank web pages

- This transforms the web ranking problem into the "standard" problem of finding an eigenvector for a square matrix.

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 1 & \frac{1}{2} \\ \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{2} & 0 & 0 \end{bmatrix}$$

$$\lambda_1 = 1$$

$$\lambda_2 = \frac{-\sqrt[3]{2} + \sqrt[3]{4} - 2}{6}$$

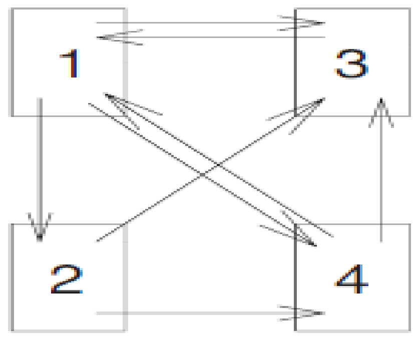
$$\lambda_3 = \frac{-i \times \sqrt{3} \times \sqrt[3]{2} + \sqrt[3]{2} - i \times \sqrt{3} \times \sqrt[3]{4} - \sqrt[3]{4} - 4}{12}$$

$$\lambda_4 = \frac{i \times \sqrt{3} \times \sqrt[3]{2} + \sqrt[3]{2} + i \times \sqrt{3} \times \sqrt[3]{4} - \sqrt[3]{4} - 4}{12}$$

- The greatest eigenvalue results in the desired centrality measure.

  For eigen value 1, the eigen vector is $[12\ \ 4\ \ 9\ \ 6]^T$ or $[2\ \ 2/3\ \ 3/2\ \ 1]^T$.

Eigenvector Centrality to rank web pages



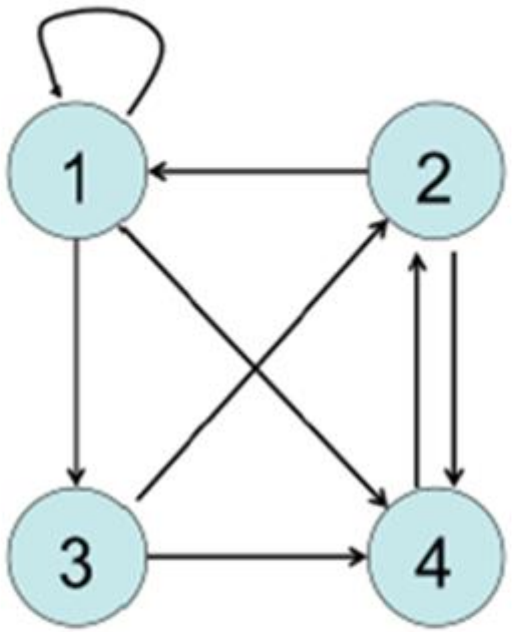
- To define an absolute score one must normalize the eigen vector e.g. such that the sum over all vertices is 1.

  In this case we obtain $\mathbf{x_1=12/31 \approx 0.387}$, $\mathbf{x_2 \approx 0.129}$, $\mathbf{x_3 \approx 0.290}$, and $\mathbf{x_4 \approx 0.194}$.

- It might **seem surprising that page 3, linked to by all other pages, is not the most important**. To understand this, **note that page 3 links only to page 1 and so casts its entire vote for page 1**. This, with the vote of page 3, results in page 1 getting the highest importance score.

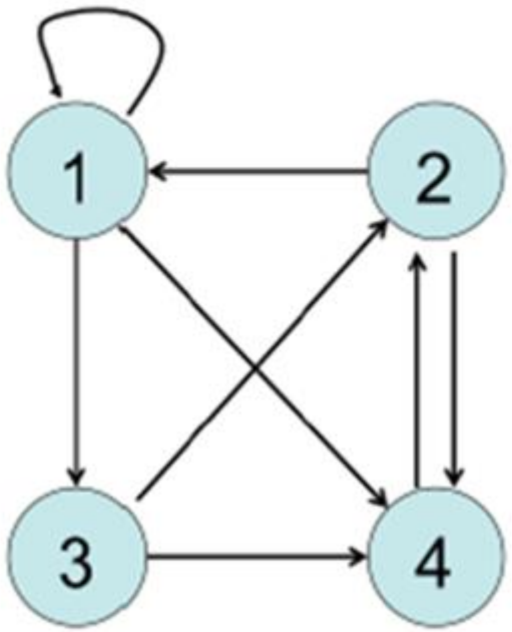
Eigenvector Centrality

- **Exercise:** Write linear equation to compute each page score.



Eigenvector Centrality

- **Exercise:** Write linear equation to compute each page score.



$$x_1 = \frac{x_1}{3} + \frac{x_2}{2}$$
$$x_2 = \frac{x_3}{2} + x_4$$
$$x_3 = \frac{x_1}{3}$$
$$x_4 = \frac{x_1}{3} + \frac{x_2}{2} + \frac{x_3}{2}$$
$$1 = x_1 + x_2 + x_3 + x_4$$

Eigenvector Centrality

Demo using NetworkX

```
>>> import networkx as nx
>>> G = nx.path_graph(4)
>>> centrality = nx.eigenvector_centrality(G)
>>> print(['%s %0.2f'%(node,centrality[node]) for node in
  centrality])
```

The output of the above code is:

['0 -> 0.37', '1 -> 0.60', '2 -> 0.60', '3 -> 0.37']

- **EigenvectorCentrality.py**

Eigenvector Centrality - Assignment

Paper Reading:

- The $25,000,000,000 Eigenvector the Linear Algebra behind Google - Kurt Bryan and Tanya Leise.

References

- Social Network Analysis: **Lada Adamic,** University of Michigan.
- Social Media Mining - Reza Zafarani
- The $25,000,000,000 Eigenvector the Linear Algebra behind Google - Kurt Bryan and Tanya Leise
- Wikipedia – Current Literature
- https://neo4j.com/docs/graph-algorithms/current/labs-algorithms/centrality/

THANK YOU

Prakash C O
Department of Computer Science and Engineering
copprakasha@pes.edu
+91 98 8059 1946