Network 2 – Centrality Measures

Giorgio Ricchiuti
www.grarchive.net
giorgio.ricchiuti@unifi.it

UNIVERSITÀ DEGLI STUDI FIRENZE

UNIVERSITÀ CATTOLICA del Sacro Cuore
CLE
Complexity Lab in Economics

CompEc

Quick Recap

Last lecture:

- A network $G = (V, E)$: nodes and edges
- Adjacency matrix A: $a_{ij} = 1$ if link exists
- Directed vs. undirected; weighted networks
- Degree: $k_i = \sum_j a_{ij}$
- Density: $c = \frac{2L}{n(n-1)}$
- Paths, distance, diameter

Today:

- Local measures: degree centrality, average neighbor degree, clustering
- Higher-order measures: betweenness, closeness, eigenvector, eccentricity
- Katz centrality and PageRank
- A numerical example
- Python: computing centrality on real networks

What Can We Learn from Networks?

- Networks capture the pattern of interactions between the parts of a system.
- A net is a simplified representation that reduces a system to an abstract topology.
- It creates a bridge between empirical data and a large toolkit of powerful analysis techniques.

Given the topology we can calculate local, meso, and large-scale topological measurements:

A) Local:

- 1st order: **degree**
- 2nd order: avg. neighbor degree
- 3rd order: **clustering**
- 4th order: squared clustering

B) Higher order:

- betweenness
- closeness
- eigenvector centrality
- eccentricity

Centrality: the Big Picture

What can we do with networks?

The **centrality** of a node measures its **importance**:

- how influential is a person in a social network?
- how critical is an element in an infrastructure network?
- what is the disease-spreading capacity of an individual?
- what is the most systemically important financial institution?

- **Degree centrality** $\rightarrow$ **POPULARITY**
- **Betweenness centrality** $\rightarrow$ **BRIDGE**
- **Closeness centrality** $\rightarrow$ **CENTRALNESS**
- **Eigenvector centrality** $\rightarrow$ **INFLUENCE**

Degree and Degree Centrality

- The **degree** of node i is the number of its links:

$$k_i = \sum_j a_{ij}$$

- The **degree centrality** normalises by the maximum possible degree:

$$dc_i = \frac{k_i}{N-1} \tag{1}$$

  where N is the total number of nodes.

- *Interpretation:* a node with high degree centrality is **popular** — it has many direct connections.

Average Neighbor Degree and Assortativity

Starting from degree, we can calculate the **average neighbor degree**:

$$k_i^{nn} = \frac{1}{k_i} \sum_{j \in \mathcal{N}(i)} k_j$$

where $\mathcal{N}(i)$ is the set of neighbors of i.

A high k_i^{nn} means that node i is linked to highly-connected nodes.

This indicator reveals whether a network is **assortative**:

- if k_i^{nn} *increases* with k_i: **assortative** (hubs connect to hubs)
- if k_i^{nn} *decreases* with k_i: **disassortative** (hubs connect to low-degree nodes)

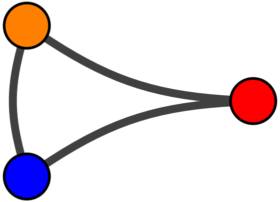
Example: social networks tend to be assortative; biological networks disassortative.

Clustering Coefficient

The **clustering coefficient** of node i measures the fraction of pairs of neighbors of i that are also connected to each other:

$$C_i = \frac{\text{number of triangles through } i}{\text{number of connected triples centred at } i} = \frac{2t_i}{k_i(k_i-1)}$$

where t_i is the number of triangles through node i.



Node A: $k_A = 2,\ t_A = 1 \Rightarrow C_A = \frac{2\cdot 1}{2\cdot 1} = 1.$ All neighbors of A are connected.

Squared Clustering

The **squared clustering** captures the density of squares (cycles of length 4) around a node.

A higher squared clustering coefficient means that nodes are strongly connected through common neighbors — reinforcing the possibility of **contagion** between a node and its neighbors.

While the standard clustering coefficient counts triangles, the squared clustering counts *quadrilaterals* — useful in bipartite or economic networks where direct triangles are rare.

Betweenness Centrality

The **betweenness centrality** is given by the number of times that node i lies on a shortest path between two other nodes j and l:

$$b_i = \sum_{\substack{j,l=1 \\ i \neq j \neq l}}^{N} \frac{d_{jl}(i)}{d_{jl}} \tag{2}$$

where d_{jl} is the total number of shortest paths from j to l and $d_{jl}(i)$ is the subset passing through i.

Interpretation: a node with high betweenness acts as a **bridge** — removing it disconnects large parts of the network.

Application: identifying systemically important banks; finding bottlenecks in supply chains.

Closeness Centrality

The **closeness centrality** is the reciprocal of the average distance from node i to all other nodes:

$$cl_i = \frac{N-1}{\sum_j d_{ij}} = \frac{1}{\bar{d}_i} \tag{3}$$

In order to be a hub, a node should not be very distant from all the others.

Interpretation: how quickly can information starting from i reach the rest of the network?

Application: optimal location of a warehouse, hospital, or emergency service.

Eigenvector Centrality

The **eigenvector centrality** measures the importance of a node based on the importance of its neighbors. In vector notation, it solves $W \cdot c = \lambda c$, i.e.:

$$c_i = \frac{1}{\lambda} \sum_j W_{ij} c_j \tag{4}$$

where λ is the largest eigenvalue of W.

It is intrinsically based on the **spectral properties** of the adjacency matrix.

Interpretation: being connected to well-connected nodes matters — this is the logic behind Google's **PageRank**.

Eccentricity

The **eccentricity** of a node is the inverse of its maximum distance from any other node:

$$e_i = \frac{1}{\max_{\forall j \in N} d_{ij}} \tag{5}$$

Related concepts:

- **Center**: set of nodes with minimum eccentricity.
- **Diameter**: maximum distance between any pair of vertices.
- **Radius**: minimum among all maximum distances, $r(G)$.

Katz Centrality

The **Katz centrality** (or alpha centrality) computes the relative influence of a node by counting *all* paths to it, penalising longer paths by an attenuation factor α:

$$x_i = \alpha^d \sum_j a_{ij} x_j + \beta$$

Each path of length d is weighted by α^d, with $0 < \alpha < 1/\lambda_{\text{max}}$.

- When $\alpha \to 0$: reduces to degree centrality
- When $\alpha \to 1/\lambda_{\text{max}}$: approaches eigenvector centrality

Useful when the network has nodes with zero degree (isolated nodes) where eigenvector centrality fails.

PageRank

PageRank is an algorithm developed by Google to rank web pages — a variant of eigenvector centrality with a *damping factor* d:

$$PR(i) = \frac{1-d}{N} + d \sum_{j \in \mathcal{N}(i)} \frac{PR(j)}{k_j^{out}}$$

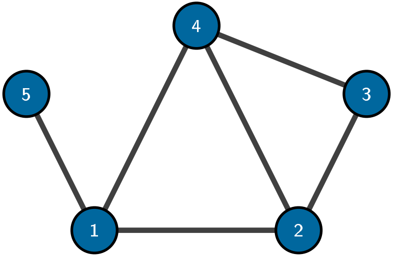
where $d \approx 0.85$ is the probability that a random surfer follows a link (rather than jumping to a random page).

- A page is important if **many important pages** link to it
- Outgoing links *dilute* the score: quality matters more than quantity

Beyond the web: used to rank scientific journals (impact factor), financial institutions (systemic risk), species in food webs.

A Worked Example

Consider the following undirected network with 5 nodes:



$$A = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Degrees: $k_1 = 3,\ k_2 = 3,\ k_3 = 2,\ k_4 = 3,\ k_5 = 1$

A Worked Example – Centrality Measures

	Node 1	Node 2	Node 3	Node 4	Node 5
Degree k_i	3	3	2	3	1
Degree centrality	0.75	0.75	0.50	0.75	0.25
Avg. neighbor deg.	2.33	2.67	3.00	2.67	3.00
Clustering C_i	0.33	0.33	1.00	0.33	0.00
Closeness cl_i	0.80	0.80	0.67	0.80	0.50
Betweenness b_i	2.0	2.0	0.0	0.0	0.0

- Nodes 1, 2, 4 are equally central by degree and closeness
- But nodes 1 and 2 are the **bridges**: remove them and the network fragments
- Node 3 has perfect clustering ($C_3 = 1$) but low betweenness — it is in a tight local cluster, not a bridge
- Node 5 is peripheral: degree 1, zero clustering, low closeness

The Zachary Karate Club

Context:

- Wayne Zachary (1977) observed a university karate club for 3 years
- 34 members, 78 friendships (edges)
- During the study, a conflict arose between the instructor (**node 0**, Mr. Hi) and the administrator (**node 33**, Officer)
- The club eventually **split into two factions**

Why it matters:

- Classic benchmark for **community detection**
- Zachary predicted the split using only network structure — before it happened
- The two factions correspond almost perfectly to the two communities identified by graph algorithms

Network statistics

Nodes	34
Edges	78
Density	0.139
Avg. degree	4.59
Diameter	5
Avg. path	2.41
Avg. clust.	0.571

Most central nodes

Degree	0, 33, 32
Betweenness	0, 33, 32
Eigenvector	33, 0, 32

The two "leaders" (0 and 33) dominate all centrality measures.

Python: What We Will Do

1. Build a small network and compute all centrality measures
2. Visualise the network with node size $\propto$ betweenness centrality
3. Compare degree vs. betweenness: do they agree?
4. Load the **Zachary Karate Club** network and repeat

Key NetworkX functions

- `nx.degree_centrality(G)`
- `nx.betweenness_centrality(G)`
- `nx.closeness_centrality(G)`
- `nx.eigenvector_centrality(G)`
- `nx.average_neighbor_degree(G)`
- `nx.clustering(G)`