Universitat Pompeu Fabra Barcelona

Lecture 5 · Centrality Measures I

Networks, Crowds and Markets

Today's Lecture

1. Degree centrality
2. Closeness centrality
3. Betweenness centrality
4. `NetworkX` examples.
5. Linear Algebra tools for centrality
6. Eigenvector Centrality

Recall: Degree centrality

Definition

The degree centrality of a node v is its degree:

$$C_{\mathrm{deg}}(v) = \mathsf{deg}(v).$$

Interpretation:

- High degree node can directly influence/reach many others.
- In undirected networks: count of adjacent edges.
- In directed networks: sometimes split into in-degree and out-degree centrality, e.g. on Twitter in-degree centrality is more relevant.

Closeness centrality

Closeness Centrality

Given $v \in V$, its average distance to other nodes in the graph is

$$\overline{d}(v) \ := \ \frac{1}{N-1} \sum_{u \neq v} d(u, v).$$

Definition

The closeness centrality of $v \in V$ is

$$C_{\mathrm{close}}(v) = \frac{1}{\overline{d}(v)},$$

where $d(u, v)$ is the distance between u and v.

- Large if v is on average close to everyone else.
- Small if many nodes are far from v.

Distance matrix

Definition

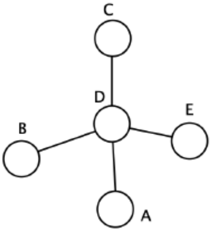
The distance matrix D_G has entries $D_G(i,j) = d(i,j)$.

Example:

$$D_G = \begin{pmatrix} 0 & 2 & 2 & 1 & 2 \\ 2 & 0 & 2 & 1 & 2 \\ 2 & 2 & 0 & 1 & 2 \\ 1 & 1 & 1 & 0 & 1 \\ 2 & 2 & 2 & 1 & 0 \end{pmatrix}.$$

Average distances from each node are computed as $\frac{1}{N-1} D_G \mathbf{1}$.

Closeness Centrality



$$\frac{1}{N-1} D_G \cdot \mathbf{1} = \frac{1}{4} \begin{pmatrix} 0 & 2 & 2 & 1 & 2 \\ 2 & 0 & 2 & 1 & 2 \\ 2 & 2 & 0 & 1 & 2 \\ 1 & 1 & 1 & 0 & 1 \\ 2 & 2 & 2 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \frac{1}{4} \begin{pmatrix} 7 \\ 7 \\ 7 \\ 4 \\ 7 \end{pmatrix} \rightarrow c = \begin{pmatrix} 4/7 \\ 4/7 \\ 4/7 \\ 1 \\ 4/7 \end{pmatrix}$$

Note that D is the most central also under degree centrality.

Eccentricity centrality

Recall: the **eccentricity** of a node v is

$$\mathrm{ecc}(v) = \max_{u \in V} d(u, v).$$

Definition

The **eccentricity centrality** of v is inversely proportional to its eccentricity:

$$C_{\mathrm{ecc}}(v) = \frac{1}{\mathrm{ecc}(v)}.$$

Note

To see how this differs from closeness centrality, imagine a dense “core” graph with a long chain of nodes attached at one end.

Betweenness centrality

Definition

The betweenness centrality of a node u measures how often u lies on shortest paths between other pairs of nodes:

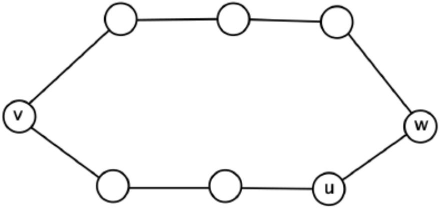
$$C_{\mathrm{betw}}(u) = \sum_{v \neq u \neq w} \frac{\sigma_{vw}(u)}{\sigma_{vw}},$$

where σ_{vw} is the number of shortest paths from v to w, and $\sigma_{vw}(u)$ is the number of those paths that pass through u.

- $\frac{\sigma_{vw}(u)}{\sigma_{vw}}$ is the proportion of path containing u in the set of all shortest paths between v and w.
- Nodes on many shortest paths act as **bridges**.
- Captures the potential of u to control information flow.

Betweenness Centrality: Example

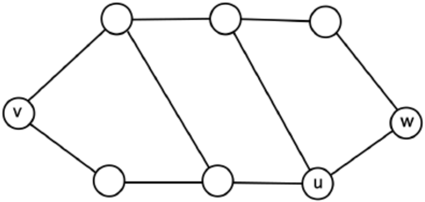
- Determine σ_{vw} and $\sigma_{vw}(u)$ in the following graph.



- $\sigma_{vw} = 2$
- $\sigma_{vw}(u) = 1$

Betweenness Centrality: Example

- Determine σ_{vw} and $\sigma_{vw}(u)$ in the following graph.



- $\sigma_{vw} = 4$
- $\sigma_{vw}(u) = 3$

Betweenness Centrality: Computing It Efficiently

Challenge: Directly counting all pairs of shortest paths costs $O(n^3)$.

Idea (Brandes, 2001): Each BFS from one source can capture *all shortest-path contributions* involving that source.

Key insight: Instead of computing all pairs (v, w), one BFS per node v is enough to accumulate betweenness scores for every other node.

Complexity: $O(nm)$ for unweighted graphs. Practical for graphs with up to $\sim 10^5$ edges.

Appendix: BFS Bookkeeping for Shortest Paths

Goal (unweighted graphs, source s): compute

- $d[v]$ = distance from s to v (in edges),
- $\sigma[v]$ = number of shortest $s \to v$ paths,
- $\mathrm{Pred}[v]$ = predecessors of v on shortest $s \to v$ paths.

Initialization:

- For all v: $d[v] = \infty$, $\sigma[v] = 0$, $\mathrm{Pred}[v] = \emptyset$.
- Set $d[s] = 0$, $\sigma[s] = 1$, push s in a queue Q.

BFS loop (standard queue):

- While Q not empty:
  - Pop v from Q.
  - For each neighbor w of v:
    - If $d[w] = \infty$ then
      $d[w] = d[v] + 1; \quad \sigma[w] = \sigma[v]; \quad \mathrm{Pred}[w] = \{v\};$ push w.
    - Else if $d[w] = d[v] + 1$ then
      $\sigma[w] \leftarrow \sigma[w] + \sigma[v];$ add v to $\mathrm{Pred}[w]$.

NetworkX examples

NetworkX quick start (Karate Club)

```
import networkx as nx

G = nx.karate_club_graph()
N, L = G.number_of_nodes(), G.number_of_edges()
print(f"N={N}, L={L}")
```

In `NetworkX`:

- `nx.degree_centrality` returns $\deg(v)/(N-1)$
- `nx.closeness_centrality(G, wf_improved=False)`
- `nx.betweenness_centrality(G)`

When plotting the network centrality measures can be used to color the nodes.

Real network #1: Florentine families (Renaissance credit/marriage)

```
F = nx.florentine_families_graph()   # N=16, classic network
print("Nodes:", F.nodes())

degF = dict(F.degree())
cloF = nx.closeness_centrality(F, wf_improved=False)
betF = nx.betweenness_centrality(F, normalized=True)

def top5(name, d):
    print(name, sorted(d.items(), key=lambda x: x[1], reverse=True)

top5("Degree:", degF)
top5("Closeness:", cloF)
top5("Betweenness:", betF)
```

Story: Medici emerge as top "brokers" by betweenness—consistent with their historical role in finance and politics.

Real network #2: Karate Club (community split)

```
G = nx.karate_club_graph()

deg  = nx.degree_centrality(G)
clo  = nx.closeness_centrality(G, wf_improved=False)
bet  = nx.betweenness_centrality(G, normalized=True)

def tab(name, d):
    rows = sorted(d.items(), key=lambda x: x[1], reverse=True)[:5]
    print(name, [(v, round(val,3)) for v,val in rows])

tab("Degree cent:", deg)
tab("Closeness cent:", clo)
tab("Betweenness cent:", bet)
```

Story: The two leaders (nodes usually labeled 0 and 33) rank highly; the broker between factions has high betweenness.

Basic spectral theory

Why Linear Algebra for Networks?

- Adjacency matrix A_G: encodes all links of G.
- Degree vector: $A_G\mathbf{1} = (\deg(v_1), \ldots, \deg(v_N))$.
- Laplacian $L = D - A_G$: central in diffusion, clustering, spanning trees.
- Many network measures (centrality, random walks, PageRank) reduce to eigenvalue/eigenvector problems.

Note

Eigenvalues of A_G reveal secrets of G.

- Google built its empire on one eigenvector (PageRank).
- Spotify/Youtube recommenders use eigenvector-like ideas.
- In social networks, eigenvector centrality captures being "friends with important people."

Recall: Eigenvalues and Eigenvectors

Definition

Let $A \in \mathbb{R}^{n \times n}$ then $\boldsymbol{v} \neq \mathbf{0}$ is called an eigenvector of A if

$$A\mathbf{v} = \lambda \mathbf{v}$$

for some λ, called eigenvalue. Assume $\|\boldsymbol{v}\| = \sqrt{\boldsymbol{v}^\top \boldsymbol{v}} = 1$.

If A has only real eigenvalues then it can be diagonalized: $\exists$ invertible P s.t.

$$A = P \Lambda P^{-1} \quad \text{with } \Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n).$$

The columns of P are the eigenvectors of A.

Note

If A is diagnosable then $A^k = P\Lambda^k P^{-1}$, $\Lambda^k = \mathrm{diag}(\lambda_1^k, \ldots, \lambda_n^k)$.

Spectral theorem

Theorem

If A is symmetric (i.e. $A = A^\top$), all eigenvalues are real, and eigenvectors form an orthogonal basis.

A is diagonalizable and for some orthogonal matrix U (i.e. $U^\top U = I_n$):

$$A \; = \; U \Lambda U^\top .$$

Spectral theorem

Theorem

If A is symmetric (i.e. $A = A^\top$), all eigenvalues are real, and eigenvectors form an orthogonal basis.

A is diagonalizable and for some orthogonal matrix U (i.e. $U^\top U = I_n$):

$$A \;=\; U \Lambda U^\top .$$

Note (Variational characterization of eigenvectors)

The eigenvectors are the saddle points of $\boldsymbol{x}^\top A \boldsymbol{x}$ subject to $\|\boldsymbol{x}\| = 1$:

- By KKT condition each optimum is a stationary point of

$$\text{Lagrangian} \;=\; \boldsymbol{x}^\top A \boldsymbol{x} - \lambda(\boldsymbol{x}^\top \boldsymbol{x} - 1).$$

- This gives $A\boldsymbol{x} = \lambda \boldsymbol{x}$. And for every such unit $\boldsymbol{x}$, $\boldsymbol{x}^\top A \boldsymbol{x} = \lambda$.

In particular, the maximal eigenvalue is $\lambda_{\max} = \max_{\|\boldsymbol{x}\|=1} \boldsymbol{x}^\top A \boldsymbol{x}$.

Eigenvalue centrality

Motivation

In degree centrality all neighbours are treated equally.

Now: a node is important if connected to other important nodes.

Motivation

In degree centrality all neighbours are treated equally.

Now: a node is important if connected to other important nodes.

- We try to define an importance measure x_v for $v \in V$ s.t.

$$x_v \;\propto\; \sum_{u \sim v} x_u.$$

In matrix form: there exists $\lambda > 0$ and a positive $\boldsymbol{x}$ s.t.

$$A_G \boldsymbol{x} = \lambda \boldsymbol{x}.$$

Motivation

In degree centrality all neighbours are treated equally.

Now: a node is important if connected to other important nodes.

- We try to define an importance measure x_v for $v \in V$ s.t.

$$x_v \propto \sum_{u \sim v} x_u.$$

In matrix form: there exists $\lambda > 0$ and a positive $\boldsymbol{x}$ s.t.

$$A_G \boldsymbol{x} = \lambda \boldsymbol{x}.$$

So centrality is given by an eigenvector of A_G with a positive eigenvalue.

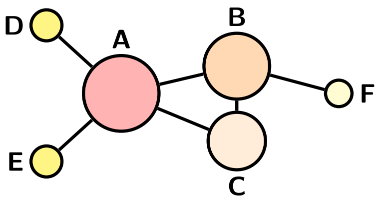
Theorem (special case of Perron-Frobenius)

As A_G has nonnegative entries, maximal eigenvalue is positive.

Since $\mathbf{1}^\top A_G \mathbf{1} = 2L > 0$ then $\lambda_{\max} > 0$.

The principal eigenvector has positive entries.

Eigenvector Centrality – Core–Periphery Example



Adjacency matrix (A):

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

Setup. A small core (A,B,C) connected as a triangle; three peripheral nodes (D,E,F) each attach to the core.

Why sizes differ.

- ***A*** connects to two central nodes (***B***,***C***) and two peripherals (***D***,***E***) — very central.
- ***B*** beats ***C*** because it also connects to ***F***.
- ***D***, ***E***, ***F*** are peripheral and get low scores.

Note (Potential problems)

- What if G is disconnected?
- What if λ_{max} has multiplicity ≥ 2?

Normalized ratios:

$$x_A : x_B : x_C : x_D : x_E : x_F \approx 1.00 : 0.87 : 0.76 : 0.41 : 0.41 : 0.35.$$