

Network Science Summer School



Universiteit Utrecht

Day program

09:30–10:00:

Introductions

10:00–11:00:

Introduction to network science

11:00–13:00:

Practical + discussion:

Network tools in Python and R

13:00–14:00

Lunch

14:00–15:00:

Network representation

Centrality

15:00–17:00:

Practical + discussion:

Centrality measures

Intro to linear algebra

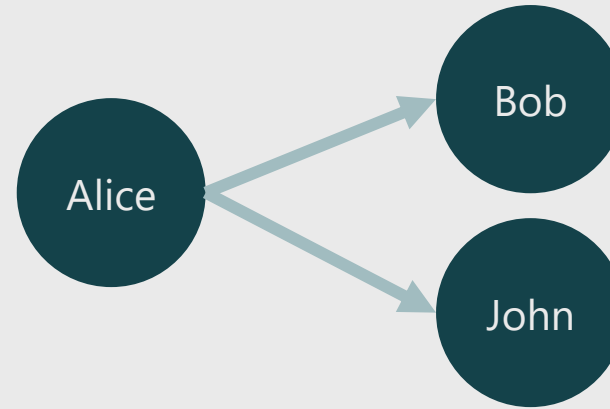
Why? Multiplying matrices is fast (relatively)

Network representation

Adjacency list:

A. It is dense: Only keeping edges

| Origin | Target | Weight |
|--------|--------|--------|
| Alice | Bob | 1 |
| Alice | John | 1 |



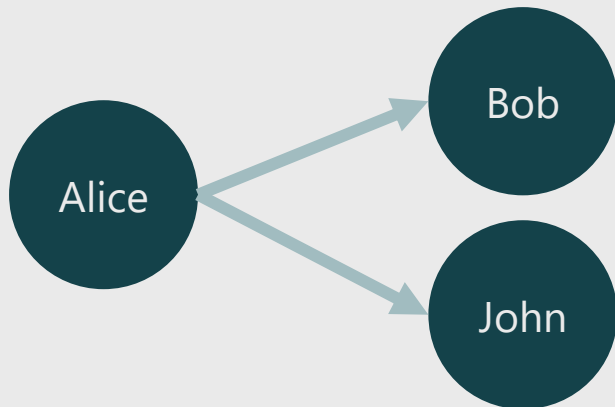
Adjacency matrix:

- A. Linear algebra is easy
- Sparse: Many zeros → 1E6 nodes/10 million edges → 1 trillion options

| Target → ↓ Source | Alice | Bob | John |
|----------------------|-------|-----|------|
| Alice | 0 | 1 | 1 |
| Bob | 0 | 0 | 0 |
| John | 0 | 0 | 0 |

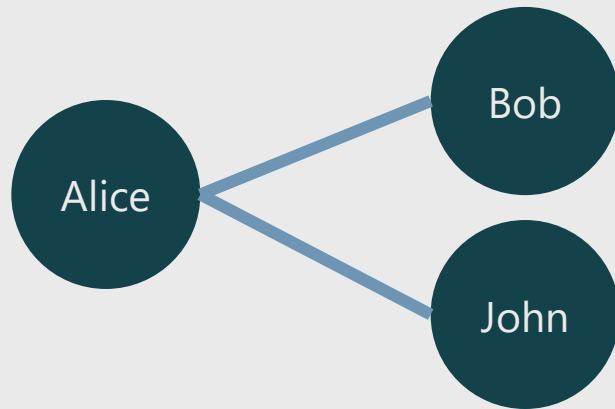
In computers → Sparse matrices: Best of both worlds

Directed networks



| Target → ↓ Source | Alice | Bob | John |
|----------------------|-------|-----|------|
| Alice | 0 | 1 | 1 |
| Bob | 0 | 0 | 0 |
| John | 0 | 0 | 0 |

Undirected networks



| Target → ↓ Source | Alice | Bob | John |
|----------------------|-------|-----|------|
| Alice | 0 | 1 | 1 |
| Bob | 1 | 0 | 0 |
| John | 1 | 0 | 0 |

Some terms

A =

| Target → ↓ Source | Alice | Bob | John |
|----------------------|-------|-----|------|
| Alice | 0 | 1 | 1 |
| Bob | 0 | 0 | 0 |
| John | 0 | 0 | 0 |

Diagonal

Trace = Sum of elements in the diagonal

Identity matrix (I) =

$$I @ A = A$$

| | 1 | 0 | 0 |
|--|---|---|---|
| | 0 | 1 | 0 |
| | 0 | 0 | 1 |

Transpose (A^T) =

(python) A.T

| Target → ↓ Source | Alice | Bob | John |
|----------------------|-------|-----|------|
| Alice | 0 | 0 | 0 |
| Bob | 1 | 0 | 0 |
| John | 1 | 0 | 0 |

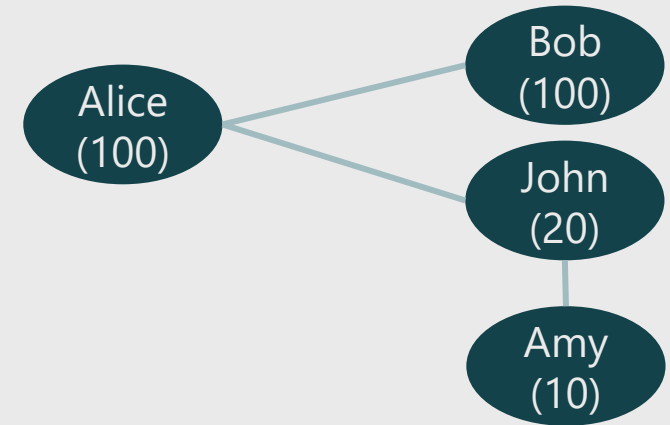
Symmetric matrix: $A = A.T$ (e.g. undirected network)

Python exercise notebook 2, ex.1

Python:

- Convert between formats
- Plot matrix

Matrix multiplication: sum



| Target → ↓ Source | Alice | Bob | John | Amy |
|----------------------|-------|-----|------|-----|
| Alice | 0 | 1 | 1 | 0 |
| Bob | 1 | 0 | 0 | 0 |
| John | 1 | 0 | 0 | 1 |
| Amy | 0 | 0 | 1 | 0 |

@

| Node | Income |
|-------|--------|
| Alice | 100 |
| Bob | 100 |
| John | 20 |
| Amy | 10 |

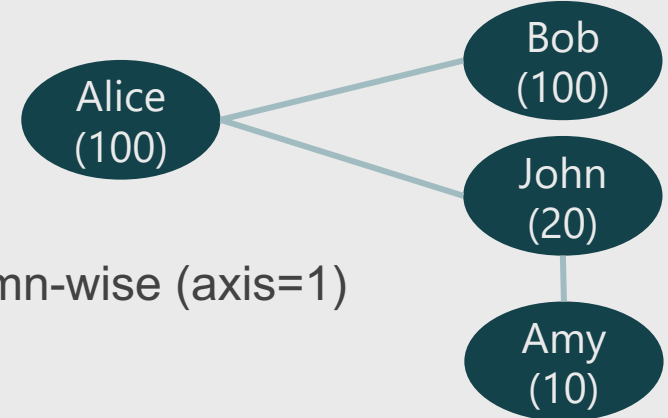
=

| Node | Income |
|-------|-------------------------------------|
| Alice | $0*100 + 1*100 + 1*20 + 0*10 = 120$ |
| Bob | $1*100 + 0*100 + 0*20 + 0*10 = 100$ |
| John | $1*100 + 0*100 + 0*20 + 1*10 = 110$ |
| Amy | $0*100 + 0*100 + 1*20 + 0*10 = 20$ |

$$A @ M = SM$$

$$(N \times N) @ (N \times 1) = (N \times 1)$$

Matrix multiplication: average



Divide by the degree. We get it by summing the adjacency elements column-wise (axis=1)

$$A @ M / A.sum(1) = \text{average}$$
$$(\text{N} \times \text{N}) @ (\text{N} \times \text{1}) / (\text{N} \times \text{1}) = (\text{N} \times \text{1}) / (\text{N} \times \text{1}) = (\text{N} \times \text{1})$$

| Target → ↓ Origin | Alice | Bob | John | Amy |
|----------------------|-------|-----|------|-----|
| Alice | 0 | 1 | 1 | 0 |
| Bob | 1 | 0 | 0 | 0 |
| John | 1 | 0 | 0 | 1 |
| Amy | 0 | 0 | 1 | 0 |

@

| Node | Income |
|-------|--------|
| Alice | 100 |
| Bob | 100 |
| John | 20 |
| Amy | 10 |

| Node | Income |
|-------|--------|
| Alice | 120 |
| Bob | 100 |
| John | 110 |
| Amy | 20 |

=

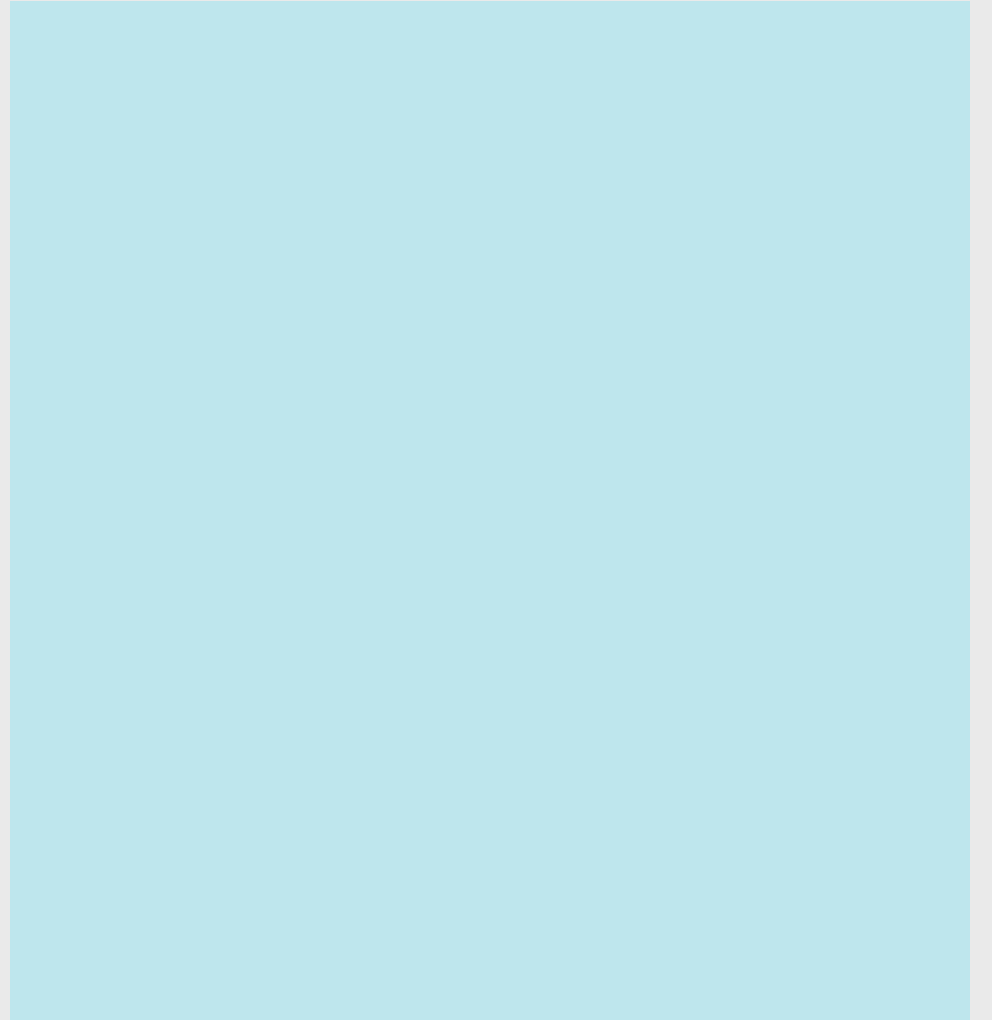
=

| Target → ↓ Source | Sum |
|----------------------|-----|
| Alice | 2 |
| Bob | 1 |
| John | 2 |
| Amy | 1 |

| Target → ↓ Source | Sum |
|----------------------|-----|
| Alice | 2 |
| Bob | 1 |
| John | 2 |
| Amy | 1 |

| Node | Income |
|-------|--------|
| Alice | 60 |
| Bob | 100 |
| John | 55 |
| Amy | 20 |

Python exercise notebook 2, ex.2



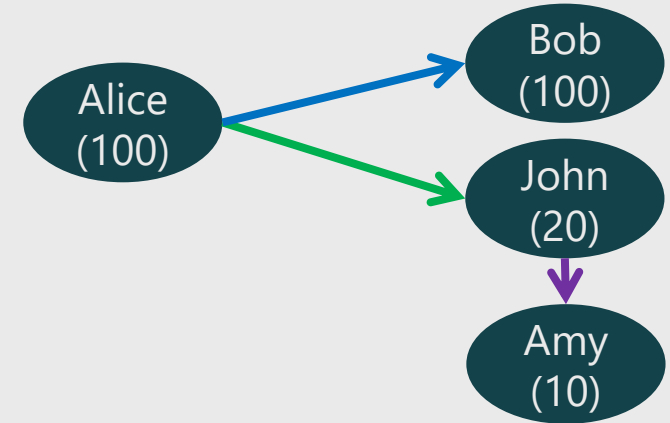
Matrix multiplication: paths

Interpretation A: Presence of path between node i and j

Interpretation A²: Number of path between node i and j in two steps

Interpretation A³: Number of path between node i and j in three steps

...



| Target → ↓ Source | Alice | Bob | John | Amy |
|----------------------|-------|-----|------|-----|
| Alice | 0 | 1 | 1 | 0 |
| Bob | 0 | 0 | 0 | 0 |
| John | 0 | 0 | 0 | 1 |
| Amy | 0 | 0 | 0 | 0 |

@

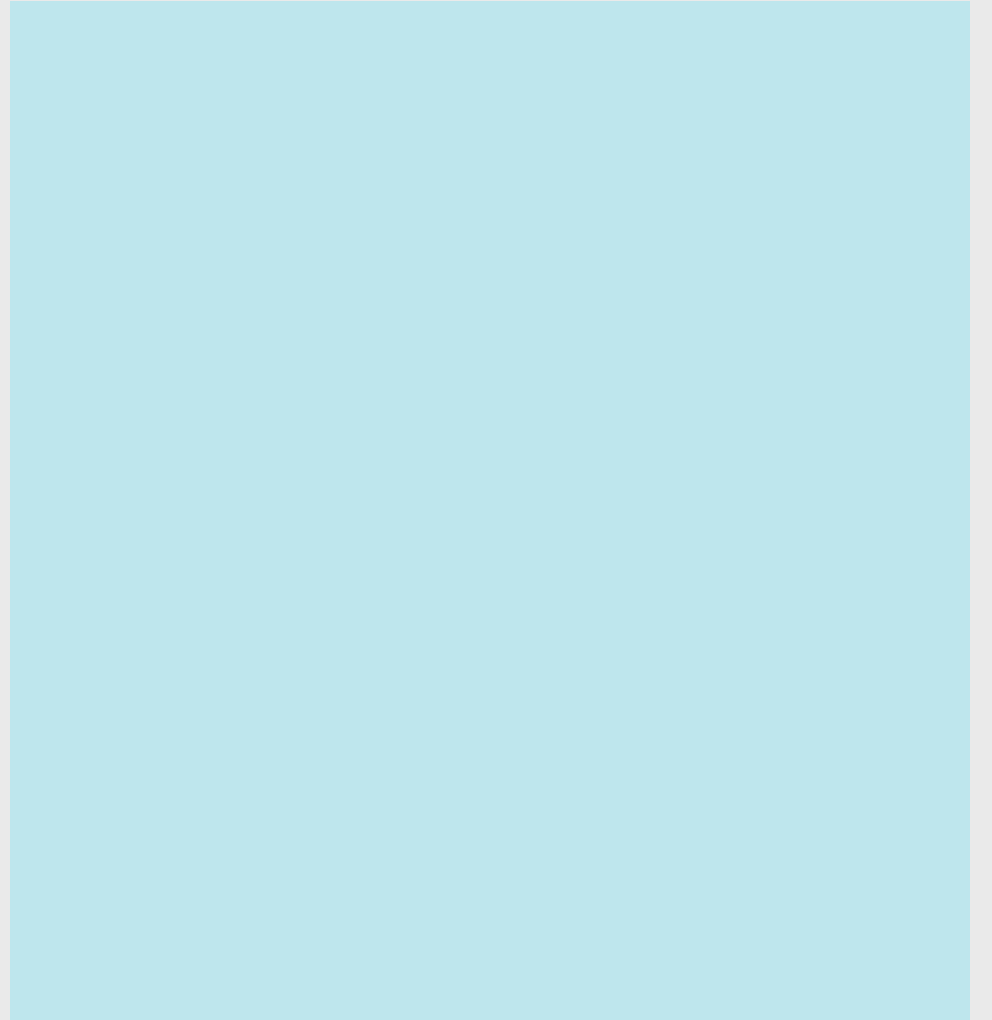
| Target → ↓ Source | Alice | Bob | John | Amy |
|----------------------|-------|-----|------|-----|
| Alice | 0 | 1 | 1 | 0 |
| Bob | 0 | 0 | 0 | 0 |
| John | 0 | 0 | 0 | 1 |
| Amy | 0 | 0 | 0 | 0 |

=

| Target → ↓ Source | Alice | Bob | John | Amy |
|----------------------|-------|-----|------|-----|
| Alice | 0 | 0 | 0 | 1 |
| Bob | 0 | 0 | 0 | 0 |
| John | 0 | 0 | 0 | 0 |
| Amy | 0 | 0 | 0 | 0 |

$$\begin{aligned} & \text{Alice} \rightarrow \text{Alice} (0) * \text{Alice} \rightarrow \text{Amy} (0) \\ & + \text{Alice} \rightarrow \text{Bob} (1) * \text{Bob} \rightarrow \text{Amy} (0) \\ & + \text{Alice} \rightarrow \text{John} (1) * \text{John} \rightarrow \text{Amy} (1) \\ & + \text{Alice} \rightarrow \text{Alice} (0) * \text{Alice} \rightarrow \text{Amy} (1) \end{aligned}$$

Python exercise notebook 2, ex.3a



Matrix multiplication: number of people reached in <3 steps

Number of paths in two or three steps from node i to node j: $N = A + A^2 + A^3$

We need to remove duplicate paths: $N = N > 0$

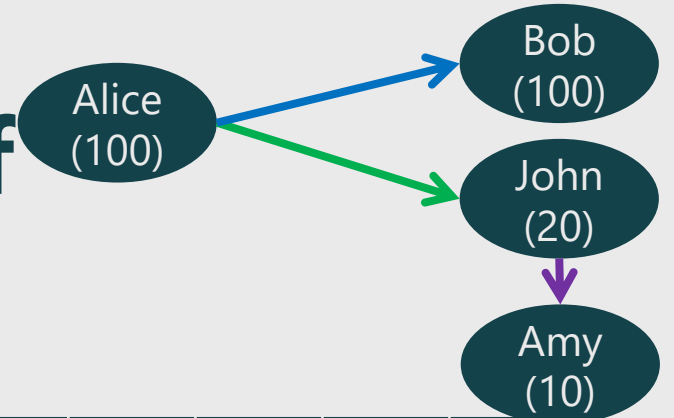
We need to remove paths from us to ourselves $N.setdiag(0)$

Matrix multiplication: number of triangles

Number of paths in two or three steps from node i to node j in three steps: A^3
We are interested in the diagonal

Undirected network? Divide the triangles by two (two directions)
Counting the total number of triangles? Divide the trace by 3

Matrix multiplication: number of triangles



A^2

| Target → ↓ Source | Alice | Bob | John | Amy |
|----------------------|-------|-----|------|-----|
| Alice | 0 | 1 | 1 | 0 |
| Bob | 0 | 0 | 0 | 0 |
| John | 0 | 0 | 0 | 1 |
| Amy | 0 | 0 | 0 | 0 |

@

| Target → ↓ Source | Alice | Bob | John | Amy |
|----------------------|-------|-----|------|-----|
| Alice | 0 | 1 | 1 | 0 |
| Bob | 0 | 0 | 0 | 0 |
| John | 0 | 0 | 0 | 1 |
| Amy | 0 | 0 | 0 | 0 |

=

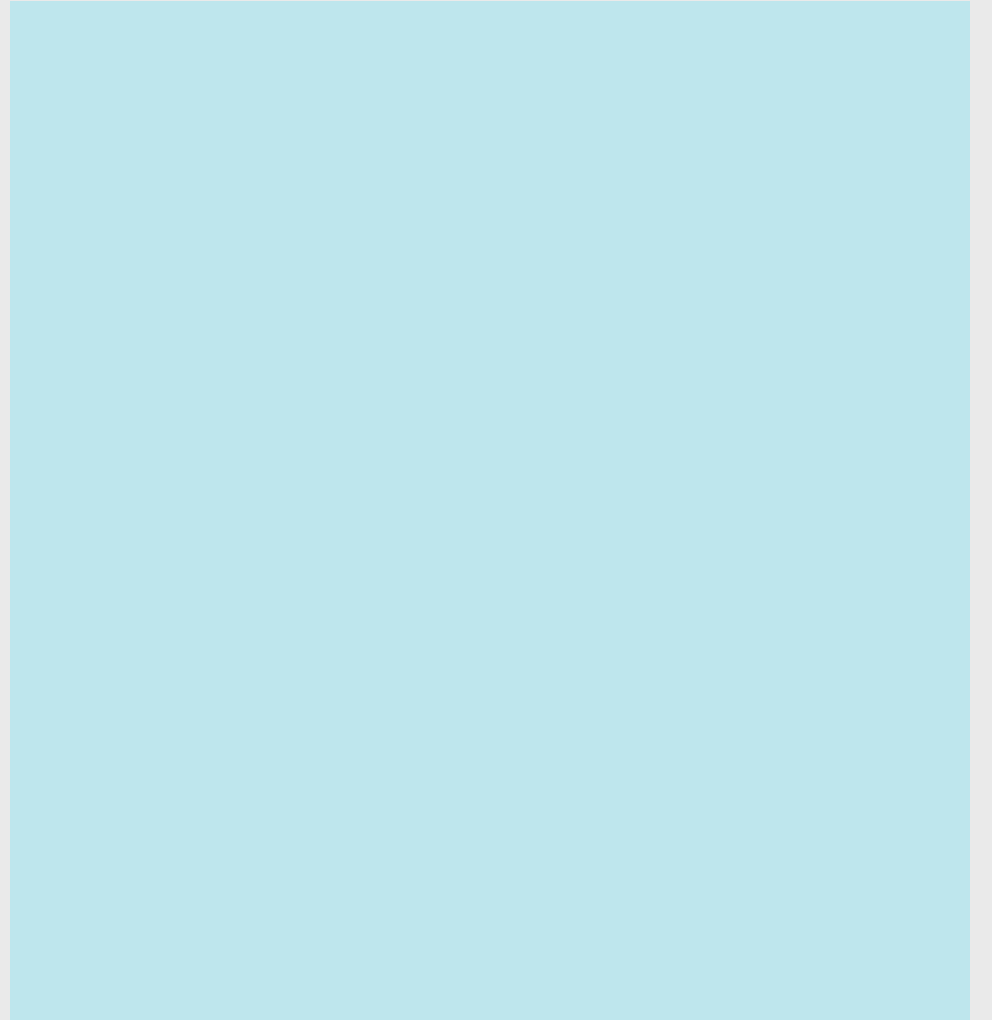
| Target → ↓ Source | Alice | Bob | John | Amy |
|----------------------|-------|-----|------|-----|
| Alice | 0 | 0 | 0 | 1 |
| Bob | 0 | 0 | 0 | 0 |
| John | 0 | 0 | 0 | 0 |
| Amy | 0 | 0 | 0 | 0 |

Alice → Alice (0) * Alice → Amy (0)
 + Alice → Bob (1) * Bob → Amy (0)
 + Alice → John (1) * John → Amy (1)
 + Alice → Alice (0) * Alice → Amy (1)

Diagonal of A^3

Alice → X_1 * X_1 → X_1 * X_1 → Alice +
 Alice → X_1 * X_1 → X_2 * X_2 → Alice +
 ...

Python exercise notebook 2, ex.3b



Centrality measures

Nice explanations:

<https://aksakalli.github.io/2017/07/17/network-centrality-measures-and-their-visualization.html>

Networks: an introduction (Newman)

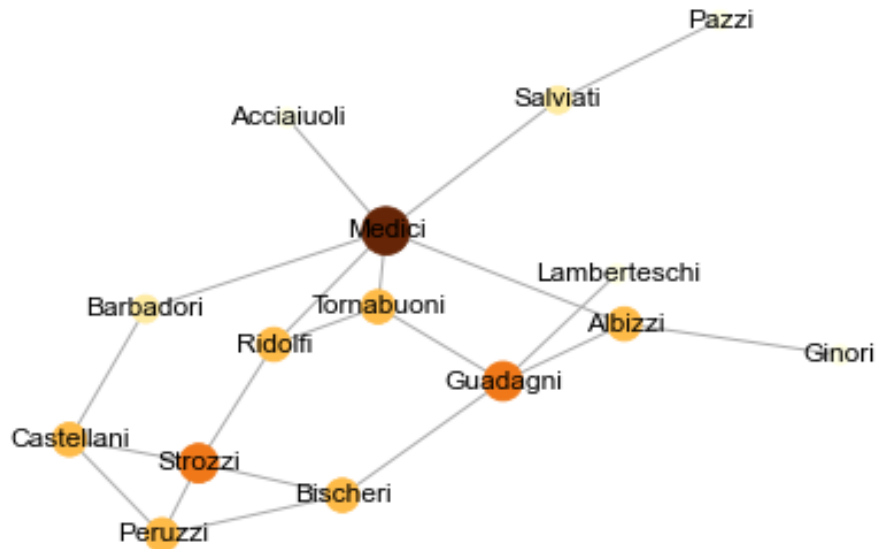
Degree centrality = $d_i / N - 1$

d_i = degree of node i

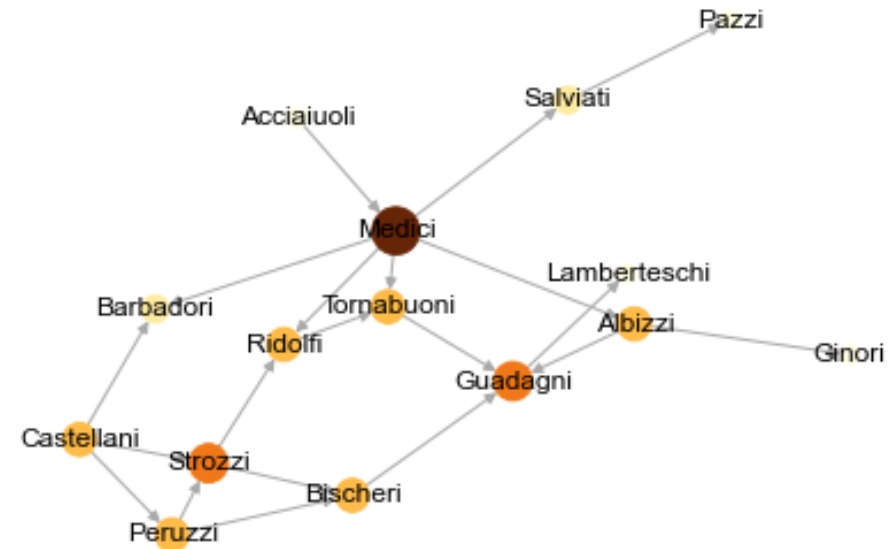
$N - 1$ = number of nodes - 1 (max. potential number of partners without self-edges/multi-edges)

Measures the **local** influence of the node

Undirected



Directed



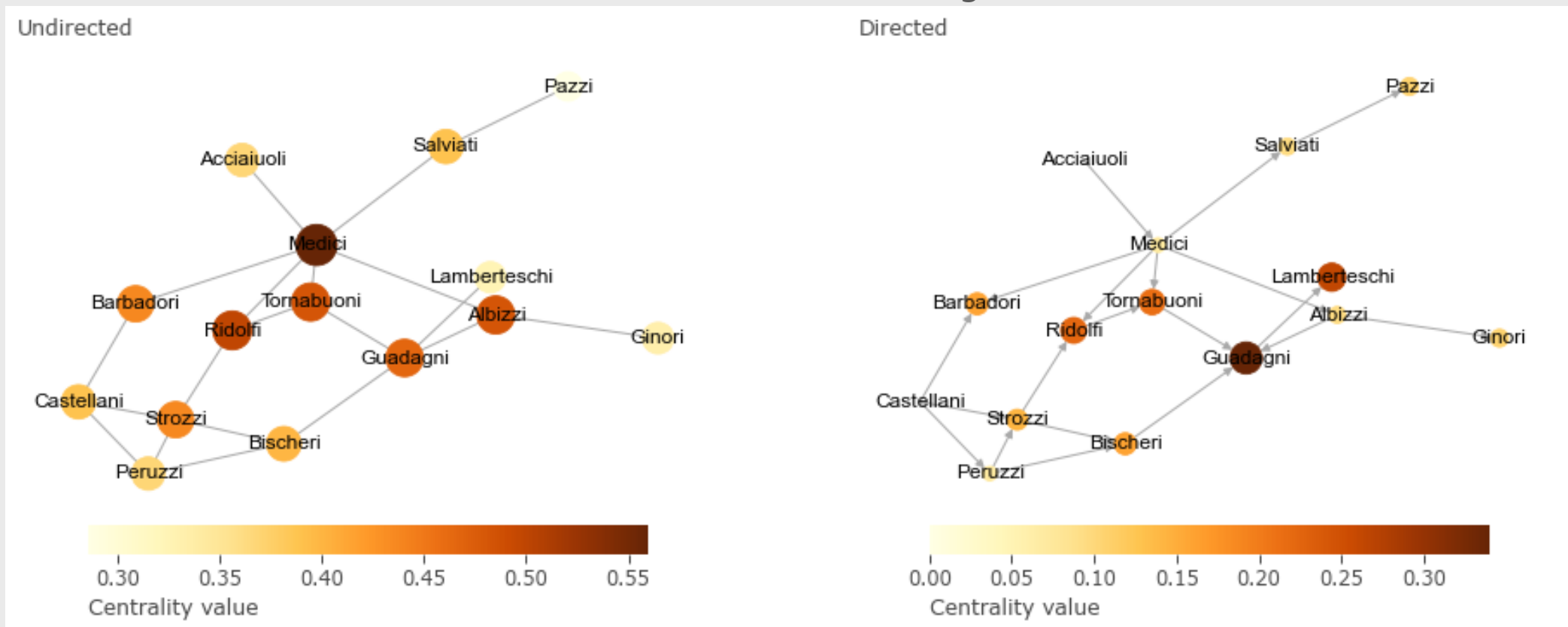
Closeness centrality = $1/l_i$

l_i = average distance of node i to all other nodes // $l_i = \frac{1}{N} \sum_j d_{ij}$

d_{ij} = shortest distance from node i to node j

Only useful in fully connected networks

Measures the **most central** node in the network (closest to get to all other nodes)

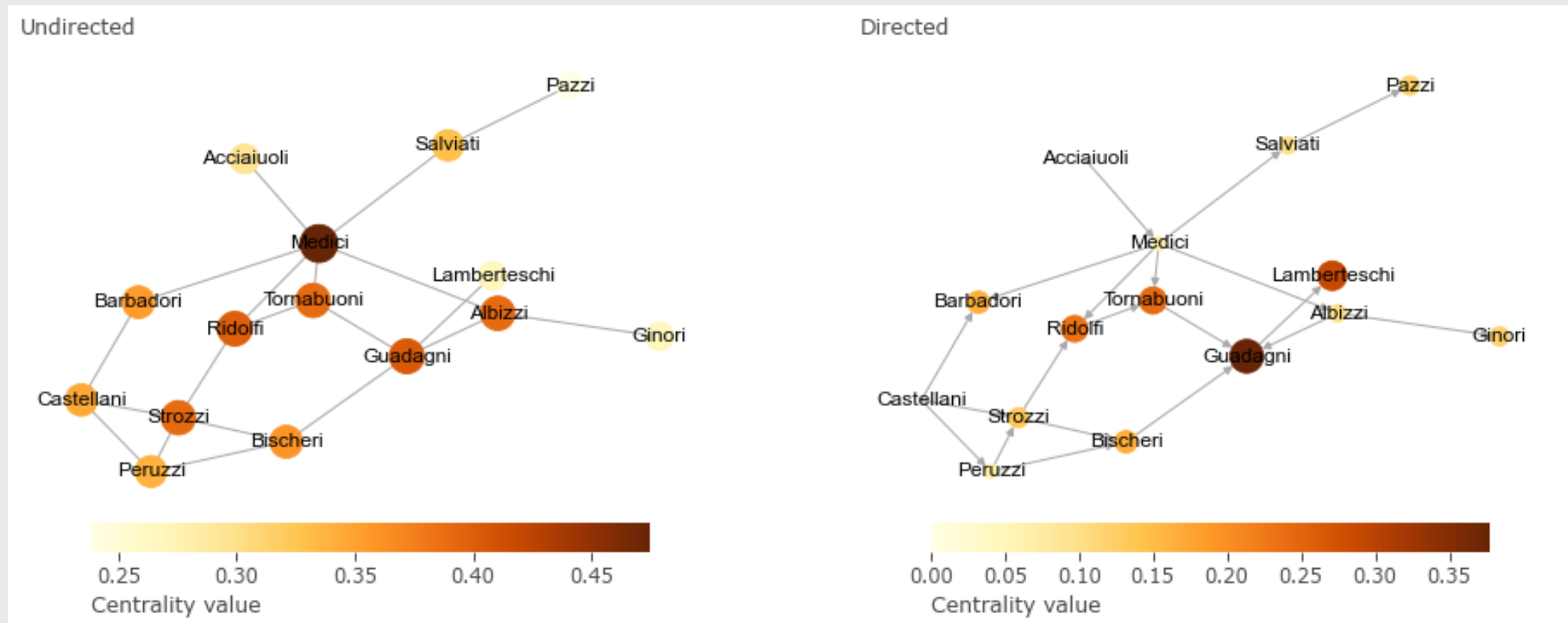


Harmonic closeness centrality = $\frac{1}{N-1} \sum_{ij|i \neq j} \frac{1}{d_{ij}}$

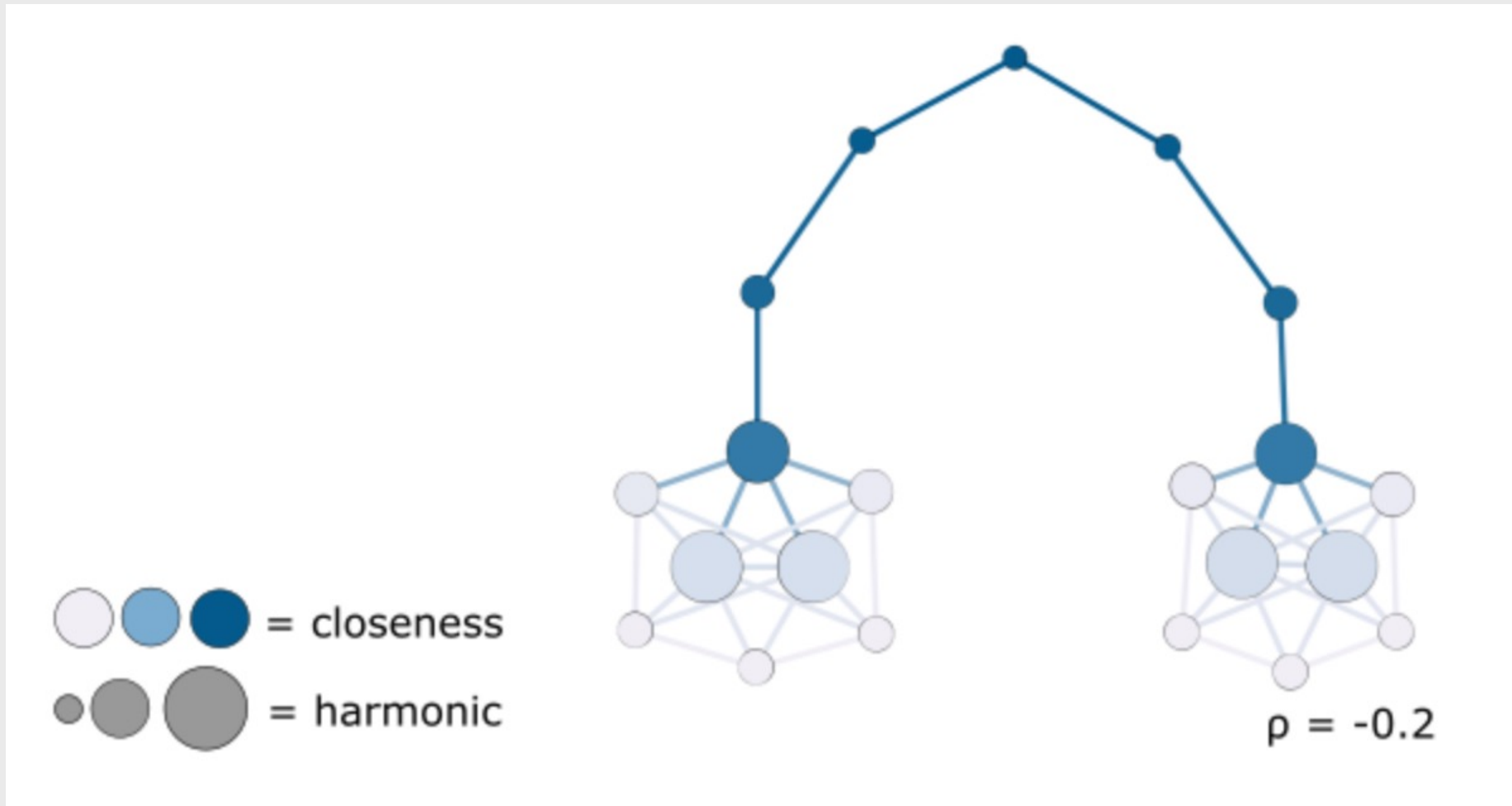
d_{ij} = shortest distance from node i to node j

Useful also in disconnected networks. Gives more weight to closer nodes.

Measures the **most central** node in the network (harmonic average)



Closeness vs harmonic



Betweenness centrality = $\frac{1}{n^2} \sum_{st} n_{st}^i$

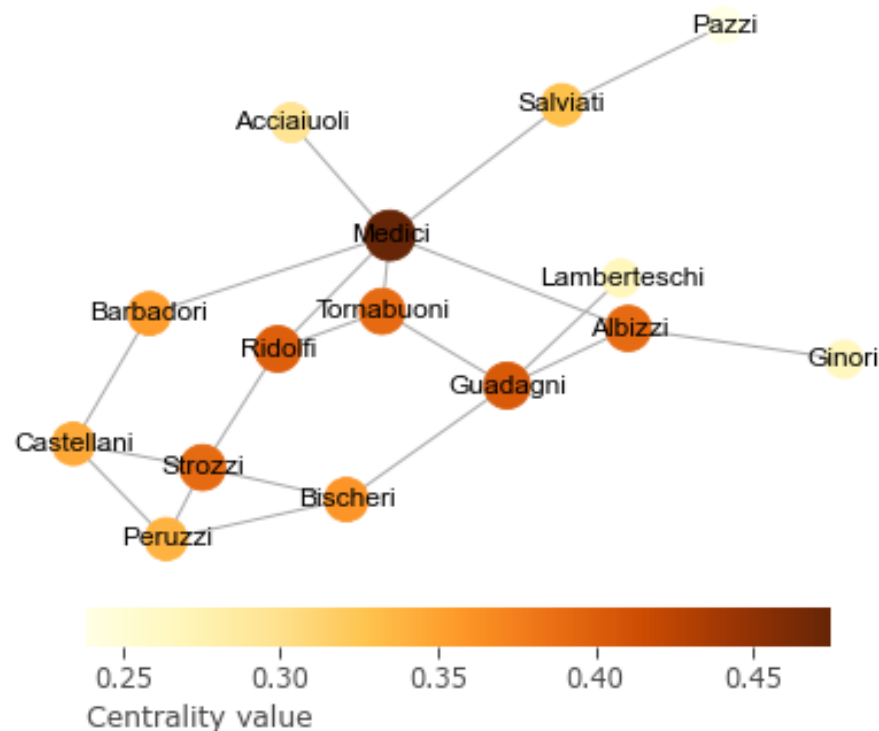
$n_{st}^i = 1/g$ if node i lies on the g shortest paths between nodes s and t

Assumptions:

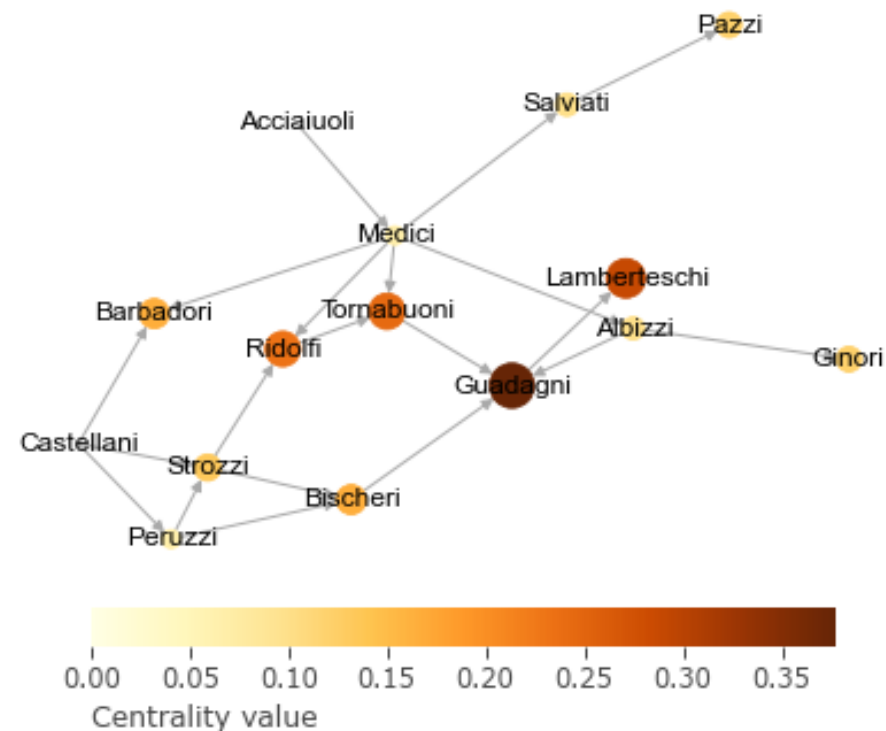
- every pair of nodes in the network exchanges messages at the same average rate
- messages always take the shortest available path through the network

Measures **brokerage** in the network → disruption of these nodes = disruption of communication

Undirected



Directed



Freeman (1977),
and Anthonisse
(1971, unpublished)

Eigenvector centrality = $\lambda^{-1} \sum_j A_{ij} e_j$

Takes into account how central your neighbors are.

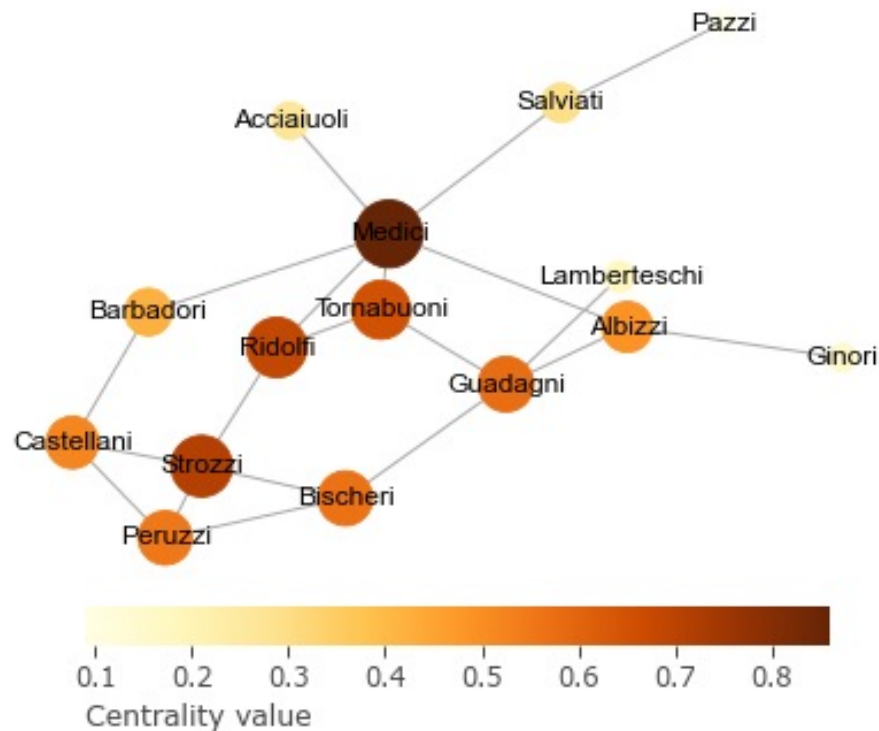
e_j = eigenvector centrality of node j

λ = largest eigenvalue

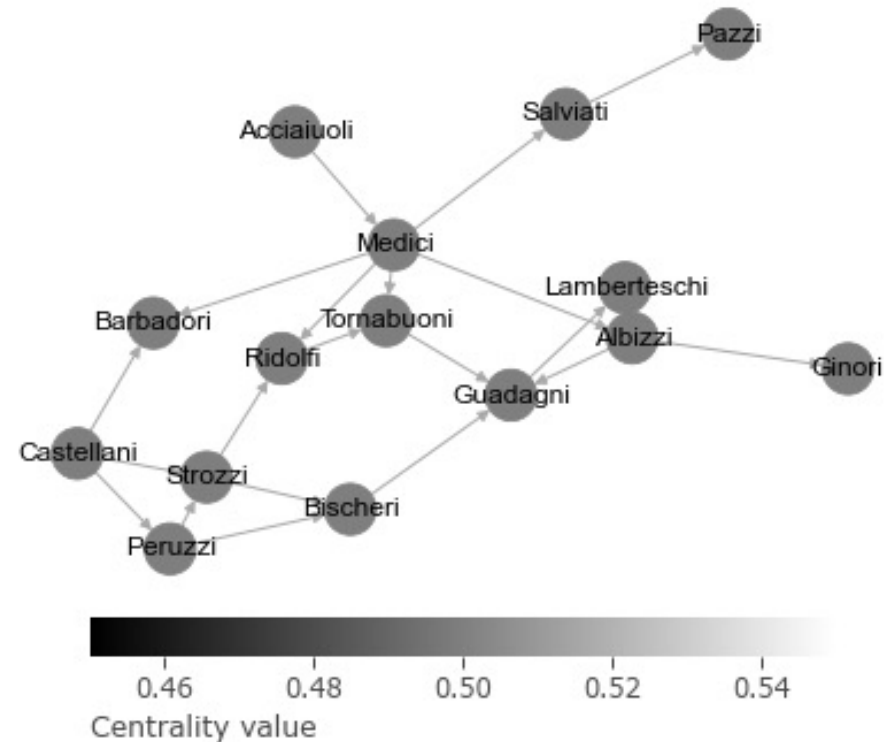
Measures total **influence** in the network (assuming all nodes are the same)

Only for undirected, fully-connected networks!

Undirected



Directed



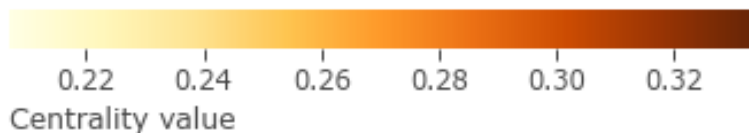
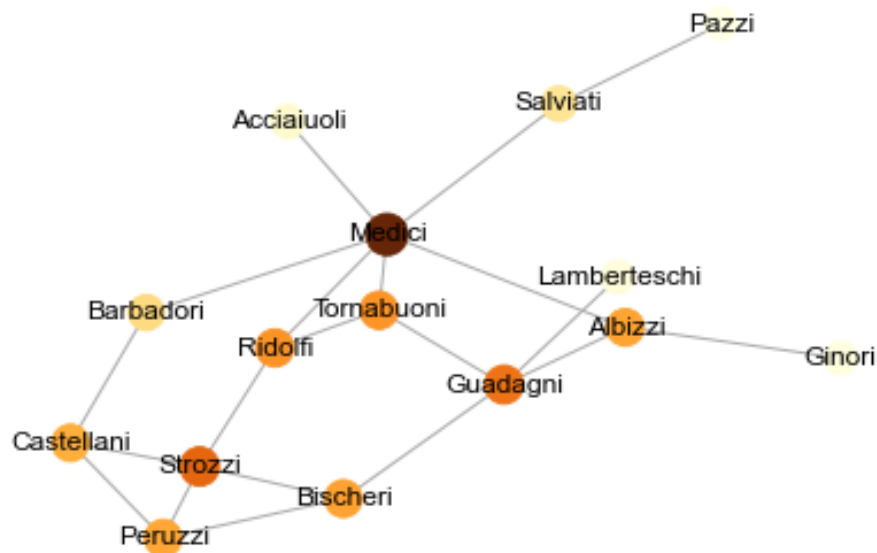
Katz centrality = $\alpha \sum_j A_{ij} k_j + \beta$

k_j = Katz centrality of node j

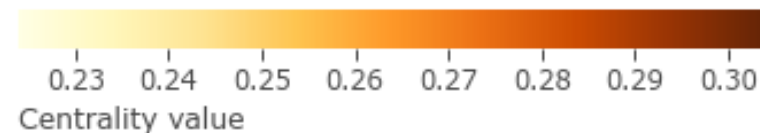
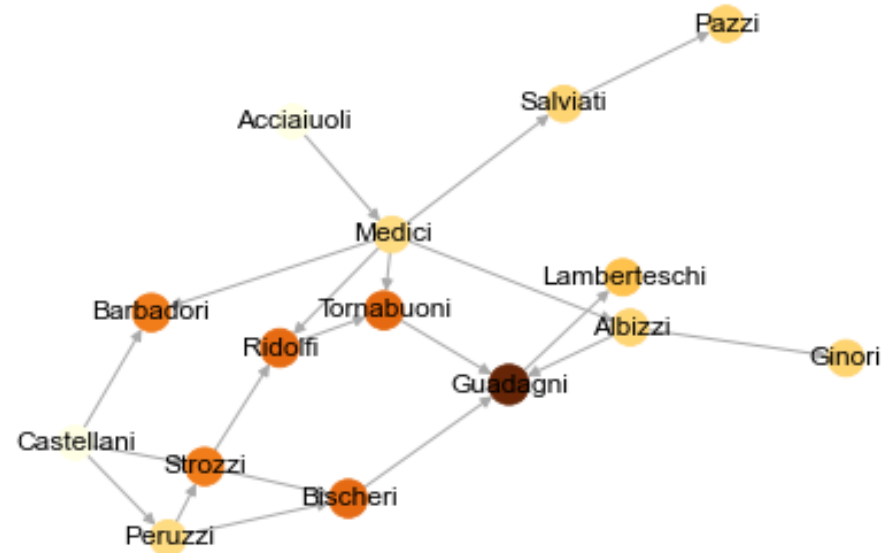
Takes into account how central your neighbors are, **each node has a minimum value of β** , and the balance between the constant and the eigenvector part is controlled by α

Measures total **influence** in the network (assuming all nodes are the same)

Undirected



Directed



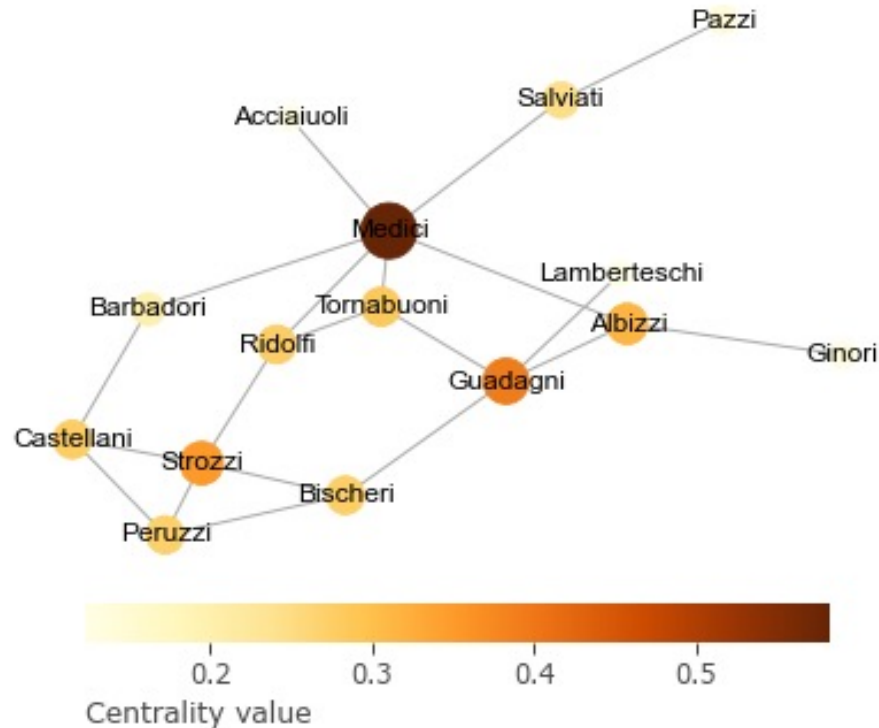
$$\text{Pagerank centrality} = \alpha \sum_j A_{ij} p_j / k_j + \beta$$

k_j = Degree of node j

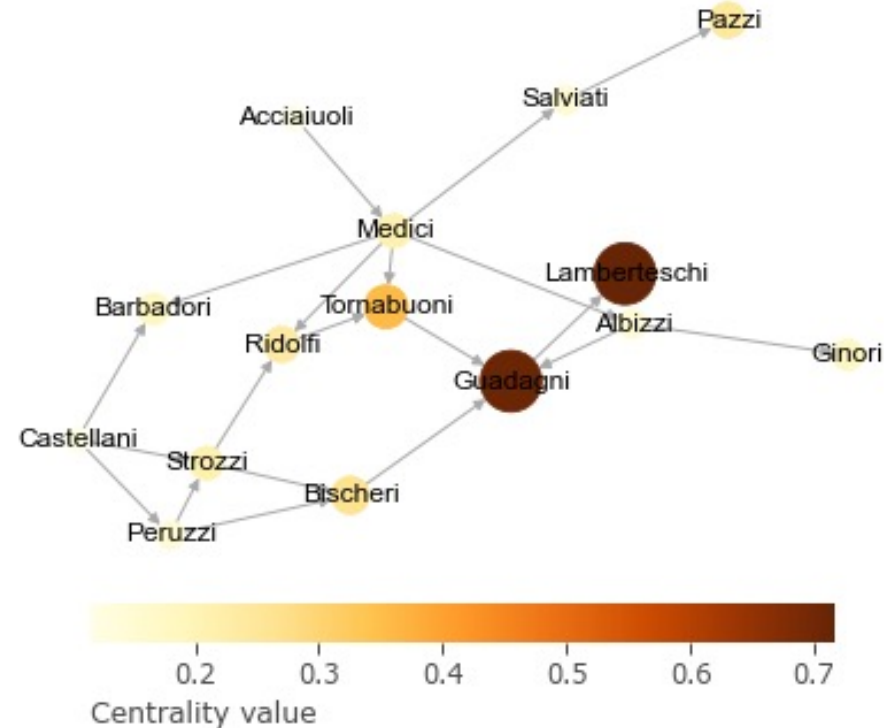
p_j = Pagerank centrality of node j

Takes into account how central your neighbors are. Each node has a minimum value of β , **the pagerank of your neighbours is normalized by their out-degree**, and the balance between the constant and the eigenvector part is controlled by α

Undirected



Directed



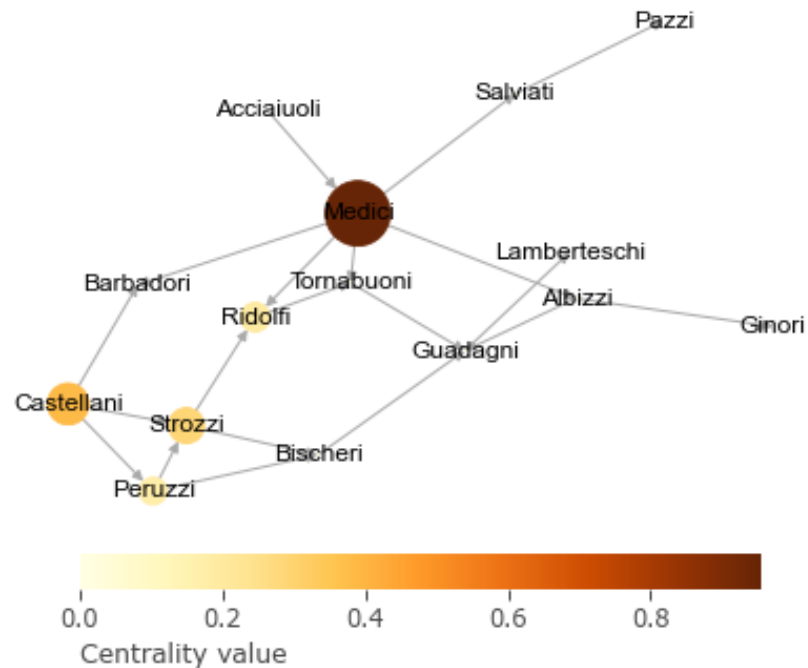
Hubs and authorities (HITS)

A node may be important if it points to others with high centrality, e.g., a review article pointing to prestigious articles

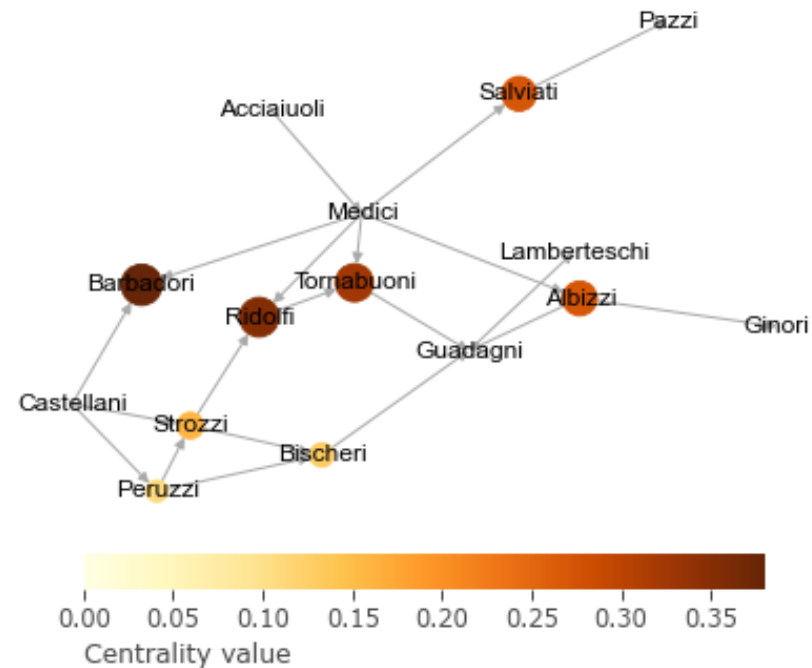
Authorities are nodes that contain useful information on a topic of interest and **hubs** are nodes that tell us where the best authorities are to be found (Newman). Two centralities: authority (a) and hub (h) centrality.

$$h_i = \alpha \sum_j A_{ij} a_j \text{ and } a_i = \alpha \sum_j A_{ij} h_j$$

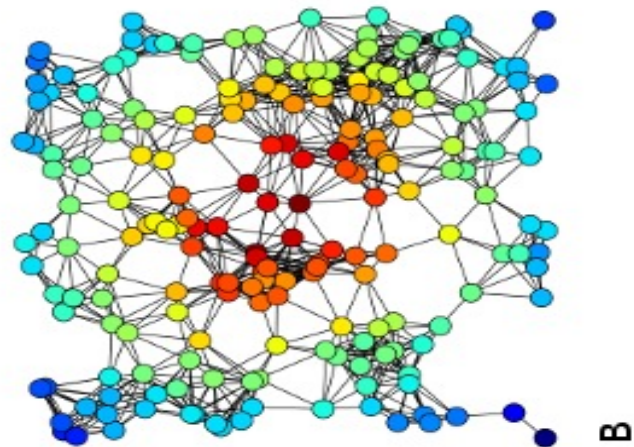
Hubs



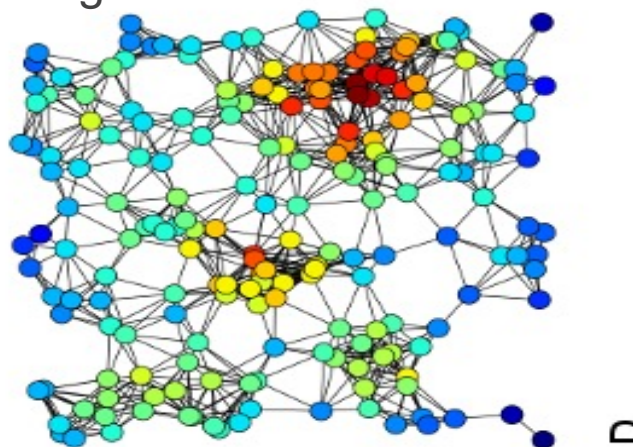
Authorities



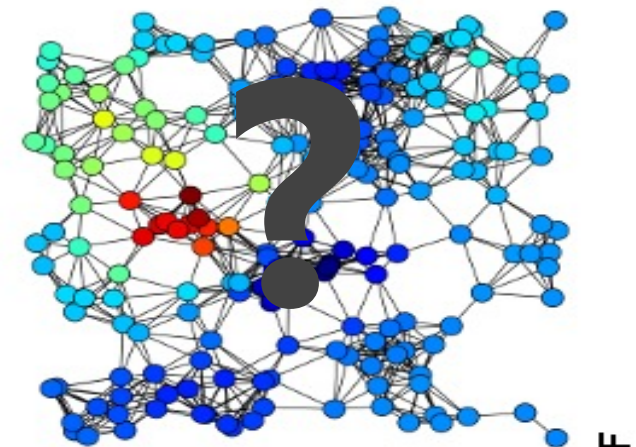
Closeness



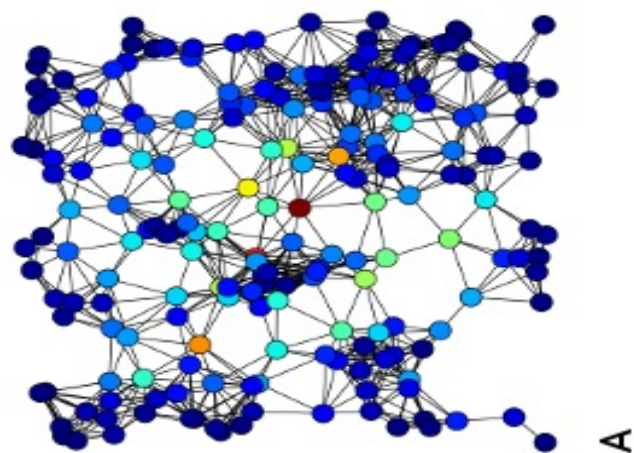
Degree



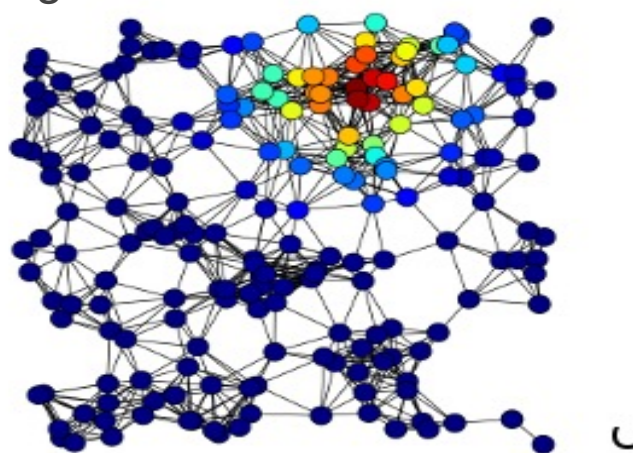
Katz



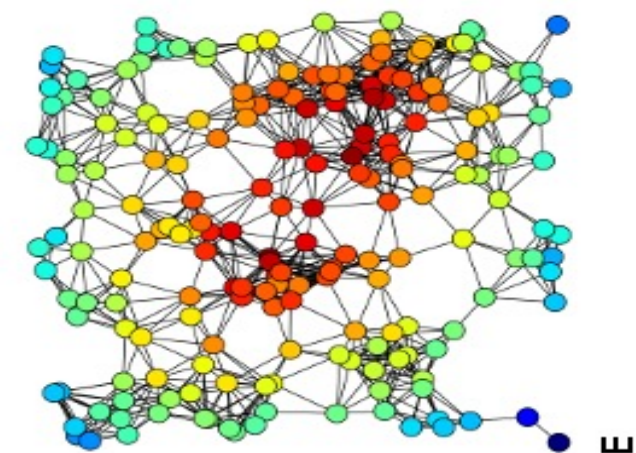
Betweenness



Eigenvector



Harmonic



Use a centrality measure that fits your theory, not the one that gives you the best results

Consider what is the real objective (e.g. is it to enable low-income individuals to increase their social capital?) (<https://petterhol.me/2019/01/11/the-importance-of-being-earnest-about-node-importance/>)

| Periodic Table of Network Centrality | | | | | | | | | | | | | | | | | | |
|--------------------------------------|--------------------------------|------------------------------------|------------------------------------|--|---------------------------------------|--|--|--|---------------------------------|-----------------------------------|---------------------------------|---------------------------------|-------------------------------------|--|-------------------------------------|-----------------------------------|---|-----------------------------------|
| 1 IA | | | | | | | | | | | | 18 VIIIA | | | | | | |
| 1 | 8000 1979 DC Degree | | | | | | | | | | | 518 1989 IC Information C | | | | | | |
| | | | | | | | | | | | | 13 IIIA | 14 IVA | 15 VA | 16 VIA | 17 VIIA | | |
| 2 | 224 1971 BC Betweenness | 239 2008 EBC Endpoint BC | | | | | | | | | | | 26 1989 kPC kPath C. | 275 2002 EGO Ego | 51 2004 HYPER Hypergraphs | 279 1997 AFF Affiliation C. | 399 2 001 α -C α -Cent. | 178 1995 ECC Eccentricity |
| | | | | | | | | | | | | | | | | | | |
| 3 | 942 1966 CC Closeness | 239 2008 PBC Proxy BC | 3 IIIA | 4 IVB | 5 VB | 6 VIB | 7 VIIB | 8 VIIIB | 9 VIIIB | 10 VIIIB | 11 IB | 12 IIB | 9068 1999 HITS Hubs/Authority | 573 2006 g-kPC geodesic kPath | 296 1999 GROUP Groups/Classes | 80 2006 HYPSC Hyperg. SC | 34 2010 t-SC t-Subgraph | 116 1998 RAD Radiality |
| | | | | | | | | | | | | | | | | | | |
| 4 | 1279 1972 EC Eigenvector | 239 2008 LSBC LscaledBC | 224 1971 EBC Edge BC | 53 2009 CBC Commun. BC | 236 2007 Δ C Delta Cent. | 5 2010 MDC MD Cent. | 0 2015 EYC Entropy C. | 2 2013 CAC Comm. Ability | 56 2007 EPTC Entropy PC | 281 1971 CCoef Clust. Coef. | 42 2012 PeC PeC | 427 2007 BN Bottleneck | 43 2009 EI Essentiality I. | 573 2006 e-kPC e-disjoint kPC | 573 2006 v-kPC v-disjoint kPC | 505 2010 WEIGHT Weighted C. | 17 2013 TCom Total Comm. | 116 1998 INT Integration |
| | | | | | | | | | | | | | | | | | | |
| 5 | 1306 1953 KS Katz Status | 239 2008 DBBC DBounded BC | 979 2005 RWBC RWalk BC | 477 1991 TEC Total Effects | 42 2009 LI Lobby Index | 11 2008 MC Mod Cent. | 0 2014 COMCC Community C. | 45 2012 ECCoef ECCoef | 0 2015 SMD Super Mediat. | 1 2014 UCC United Comp. | 4 2012 WDC WDC | 119 2008 MNC MNC | 43 2009 KL Clique Level | 179 2005 BIP Bipartivity | 426 1988 GPI GPI Power | 116 1991 kRPC Reachability | 58 2007 SCodd odd Subgraph | 586 2004 RWCC RWalk CC |
| | | | | | | | | | | | | | | | | | | |
| 6 | 8053 1999 PR Page Rank | 239 2008 DSBC DScaled BC | 291 1953 σ Stress | 477 1991 IEC Immediate Eff. | 1 2014 DM Degree Mass | 10 2012 LAPC Laplacian C. | 0 2012 ABC Attentive BC | 1699 2001 STRC Straightness C | 0 2015 SNR Silent Node R. | 15 2011 HPC Harm. Prot. | 26 2011 LAC Local Average | 119 2008 DMNC DMNC | 3 2013 LR Lurker Rank | 2457 1987 β -C β Cent. | X X HYP Hyperbolic C. | 27 2012 KEPC k-edge PC | 13 2007 FC Functional C. | 0 2014 HCC Hierar. CC |
| | | | | | | | | | | | | | | | | | | |
| 7 | 484 2005 SC Subgraph | 613 1991 FBC Flow BC | 14 2012 RLBC RLimited BC | 477 1991 MEC Mediative Eff. | 69 2010 LEVC Leverage Cent. | 35 2010 TC Topological C. | X X SDC Sphere Degree | 15 2010 ZC Zonal Cent. | 14 2013 CI Collab. Index | 11 2013 CoEWC CoEWC | 45 2012 NC NC | 108 2010 MLC Moduland C. | X X RSC Resolvent SC | 1 2014 SWIPD SWIPD | 36 2009 XXXX LinComb | 0 2014 BCPR BCPR | 0 2014 TPC Tunable PC | 0 2015 EDCC Effective Dist. |
| | | | | | | | | | | | | | | | | | | |
| citations year C Name | | | | | | | | | | | | | | | | | | |
| | | 8000 1979 Freeman Conceptual | 942 1966 Sabidussi Axiomatic | 573 2006 Borgatti/Everett Conceptual | 1130 2005 Borgatti Conceptual | 24 2014 Boldi/Vigna Axiomatic | 252 1974 Nieminen Axiomatic | 6 1981 Kishi Axiomatic | 3 2012 Kitti Axiomatic | 3 2009 Garg Axiomatic | | | | | | | | |
| | | 2065 1934 Moreno Historic | 1546 1950 Bavelas Historic | 780 1948 Bavelas Historic | 1475 1951 Leavitt Historic | 297 1992 Borgatti/Everett Conceptual | 3649 2001 Jeong et al. Empirical | 4167 1998 Tsai/Ghoshal Empirical | 961 1993 Ibarra Empirical | 71 2008 Valente Empirical | | | | | | | | |
| | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | |

“Traditional”

Betweenness-like

Friedkin Measures

Miscellaneous

Path-based

Specific Network Type

Spectral-based

Closeness-like

©David Schoch (University of Konstanz)

Chains

Sometimes data is represented as chains

- Life trajectory: Aranda → León → Vermont → Amsterdam
- Ownership chain: (right figure)

They allow you to do other analysis:

- Importance of the node based on how often it is found in between
- Importance of the node based on how many people jump to you

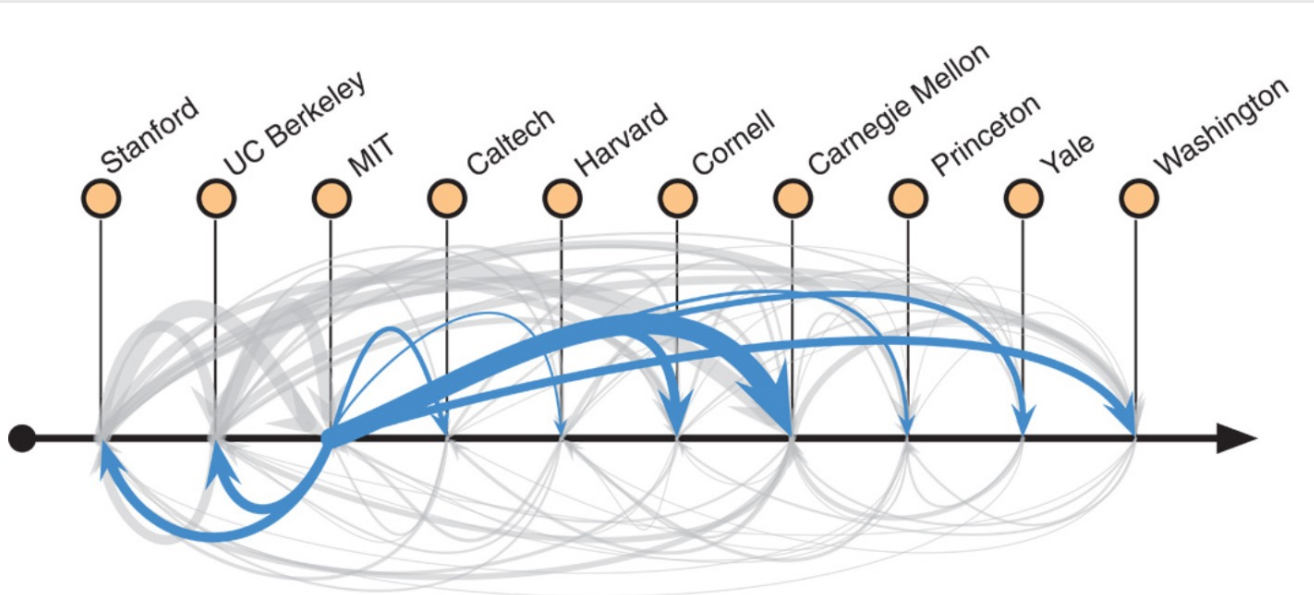


Fig. 1 Prestige hierarchies in faculty hiring networks.

Clauset, Arbesman and Larremore (2015)

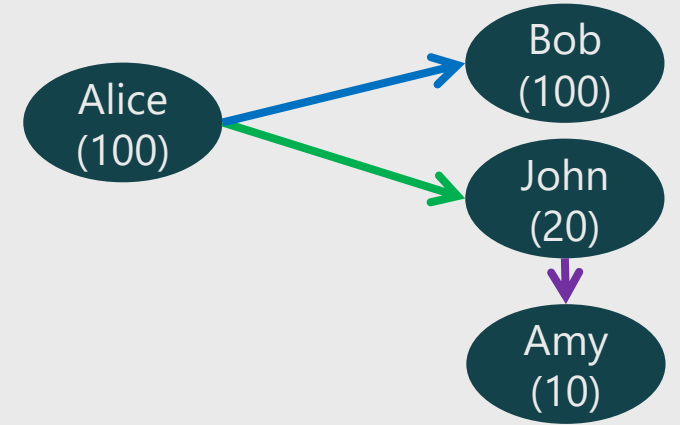


Practical 2:

Exercise 4 and 5

Linear algebra and centrality measures

Degree



| Target → ↓ Source | Alice | Bob | John | Amy |
|----------------------|-------|-----|------|-----|
| Alice | 0 | 1 | 1 | 0 |
| Bob | 0 | 0 | 0 | 0 |
| John | 0 | 0 | 0 | 1 |
| Amy | 0 | 0 | 0 | 0 |

@

| Alice | 1 |
|-------|---|
| Bob | 1 |
| John | 1 |
| Amy | 1 |

=

| | Out-Degree |
|-------|------------|
| Alice | 2 |
| Bob | 0 |
| John | 1 |
| Amy | 0 |

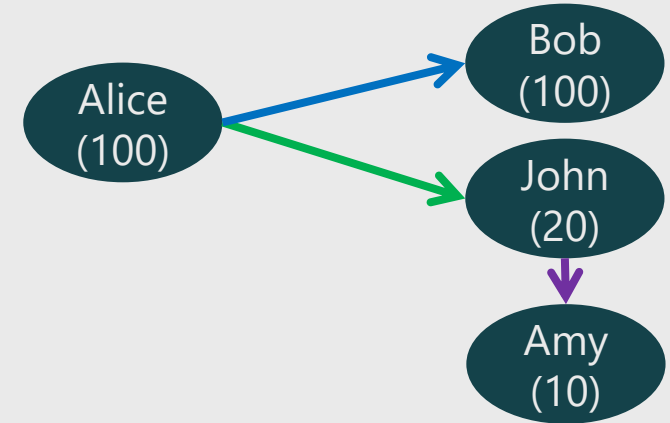
Matrix multiplication: paths

Interpretation A: Presence of path between node i and j

Interpretation A²: Number of path between node i and j in two steps

Interpretation A³: Number of path between node i and j in three steps

...



| Target → ↓ Source | Alice | Bob | John | Amy |
|----------------------|-------|-----|------|-----|
| Alice | 0 | 1 | 1 | 0 |
| Bob | 0 | 0 | 0 | 0 |
| John | 0 | 0 | 0 | 1 |
| Amy | 0 | 0 | 0 | 0 |

@

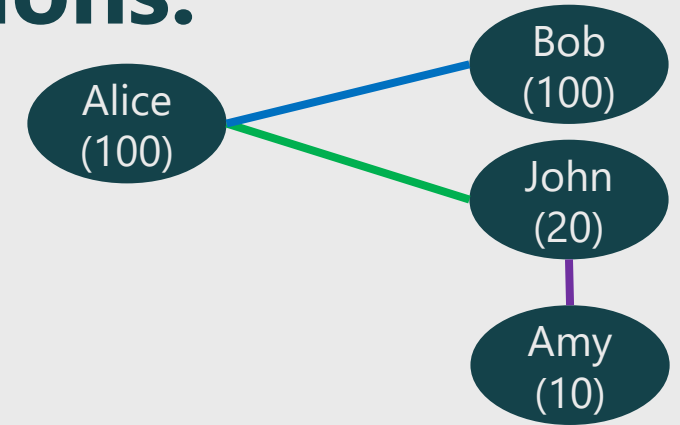
| Target → ↓ Source | Alice | Bob | John | Amy |
|----------------------|-------|-----|------|-----|
| Alice | 0 | 1 | 1 | 0 |
| Bob | 0 | 0 | 0 | 0 |
| John | 0 | 0 | 0 | 1 |
| Amy | 0 | 0 | 0 | 0 |

=

| Target → ↓ Source | Alice | Bob | John | Amy |
|----------------------|-------|-----|------|-----|
| Alice | 0 | 0 | 0 | 1 |
| Bob | 0 | 0 | 0 | 0 |
| John | 0 | 0 | 0 | 0 |
| Amy | 0 | 0 | 0 | 0 |

$$\begin{aligned} & \text{Alice} \rightarrow \text{Alice (0)} * \text{Alice} \rightarrow \text{Amy (0)} \\ & + \text{Alice} \rightarrow \text{Bob (1)} * \text{Bob} \rightarrow \text{Amy (0)} \\ & + \text{Alice} \rightarrow \text{John (1)} * \text{John} \rightarrow \text{Amy (1)} \\ & + \text{Alice} \rightarrow \text{Alice (0)} * \text{Alice} \rightarrow \text{Amy (1)} \end{aligned}$$

Another view on matrix multiplications: Random walks



Transition matrix (row-normalized A)

| Target → ↓ Source | Alice | Bob | John | Amy |
|----------------------|-------|-----|------|-----|
| Alice | 0 | 1 | 0.5 | 0 |
| Bob | 1 | 0 | 0 | 0 |
| John | 0.5 | 0 | 0 | 0.5 |
| Amy | 0 | 0 | 1 | 0 |

 @

| Target → ↓ Source | Alice | Bob | John | Amy |
|----------------------|-------|-----|------|-----|
| Alice | 0 | 0.5 | 0.5 | 0 |
| Bob | 1 | 0 | 0 | 0 |
| John | 0.5 | 0 | 0 | 0.5 |
| Amy | 0 | 0 | 1 | 0 |

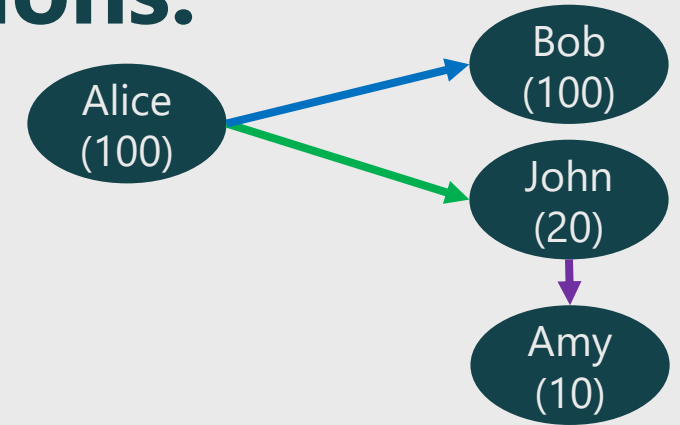
 =

A random walker starting in Alice will go 50% of the times to Bob, 50% of the times to John.

- From Bob it goes 100% of the times back to Alice
- From John it goes 50% of the times to John, 50% back to Alice

If we let the random walker walk forever → The fraction of time spend at each node converges to the **degree centrality** of the node

Another view on matrix multiplications: Random walks and Pagerank



Transition matrix (row-normalized A)

| Target → ↓ Source | Alice | Bob | John | Amy | | Target → ↓ Source | Alice | Bob | John | Amy |
|----------------------|-------|-----|------|-----|---|----------------------|-------|-----|------|-----|
| Alice | 0 | 0.5 | 0.5 | 0 | @ | Alice | 0 | 0.5 | 0.5 | 0 |
| Bob | 0 | 0 | 0 | 0 | | Bob | 0 | 0 | 0 | 0 |
| John | 0 | 0 | 0 | 1 | | John | 0 | 0 | 0 | 1 |
| Amy | 0 | 0 | 0 | 0 | | Amy | 0 | 0 | 0 | 0 |

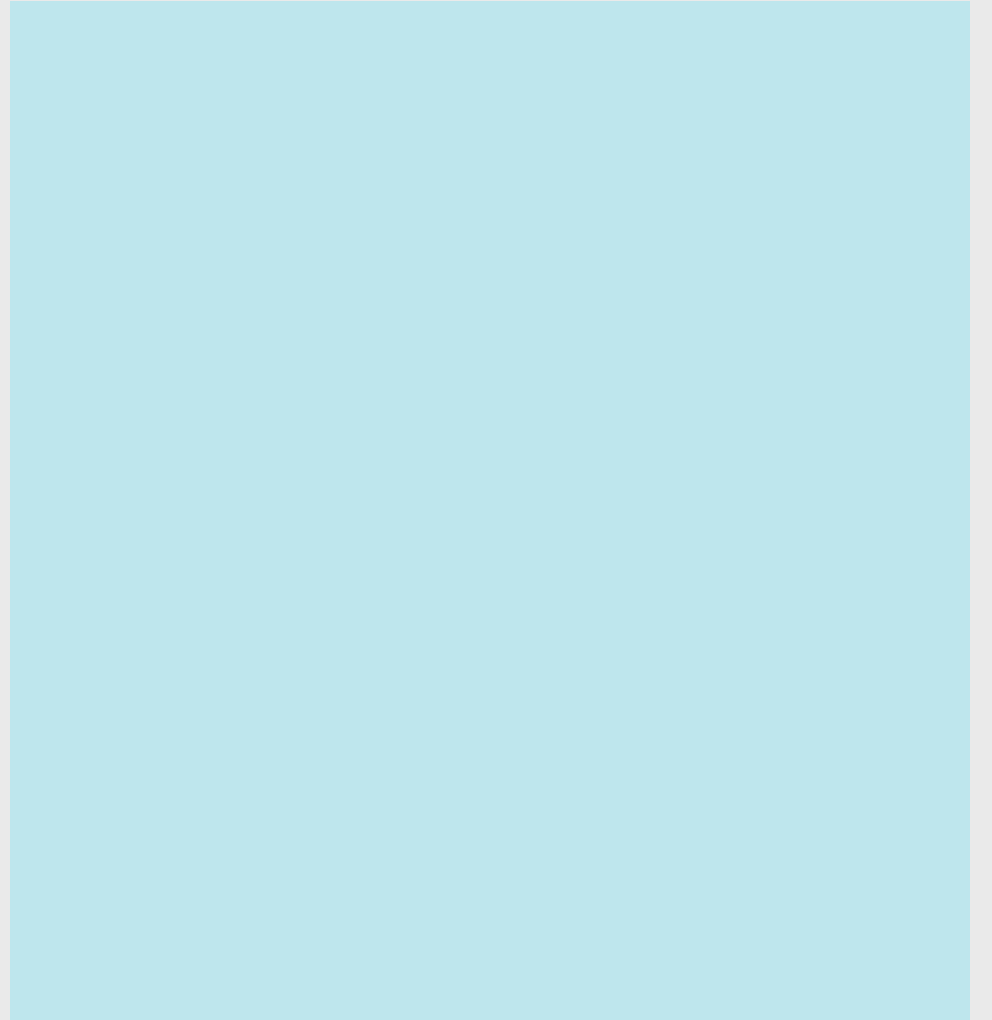
=

A random walker starting in Alice will go 50% of the times to Bob, 50% of the times to John.

- From Bob it gets trapped
- From John it goes 100% of the times to Amy and gets trapped

If we let random walkers walk forever → They get trapped in the extremes! Corrected using PageRank (the beta parameter can be understood as a teleportation probability)

Python exercise notebook 2, ex.7



Practical 3:

Working with networks using Gephi

Follow this tutorial (slides 1–23 only!): <https://gephi.org/users/quick-start/>

- In community detection use the “stochastic blockmodel” instead of modularity maximization (or try both)

You can choose to use our own data (<https://tinyurl.com/network-game>) or the Twitter data.