Principles/Social Media Mining

CIS 600

Week 8: Centrality Analysis

`

Edmund Yu, PhD
Associate Teaching Professor
esyu@syr.edu

October 15, 2020

Influence

- In a social network, nodes are usually not independent of each other
  - They are connected by ties (or edges, links, connections…)
- When nodes are connected, they could influence each other, via the connections
  - In a broader sense, influence is a form of contagion that moves in a network of connected nodes, and it can be amplified or attenuated.
- Not all nodes are created equal in a social network. Some nodes are more important than others.
  - "All animals are equal, but some animals are more equal than others." - George Orwell, **Animal Farm**

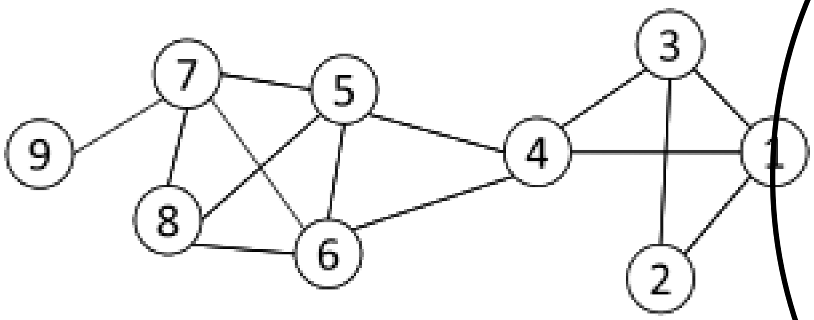
Centrality Analysis

- How do we find those more important nodes?
  - Centrality Analysis - commonly used measures include:
    - Degree Centrality
    - Closeness Centrality
    - Betweenness Centrality
    - Eigenvector Centrality

Networks and Representation

Social Network: A social structure made of nodes (individuals or organizations) and edges that connect nodes in various relationships like friendship, kinship etc.

❖ Graph Representation



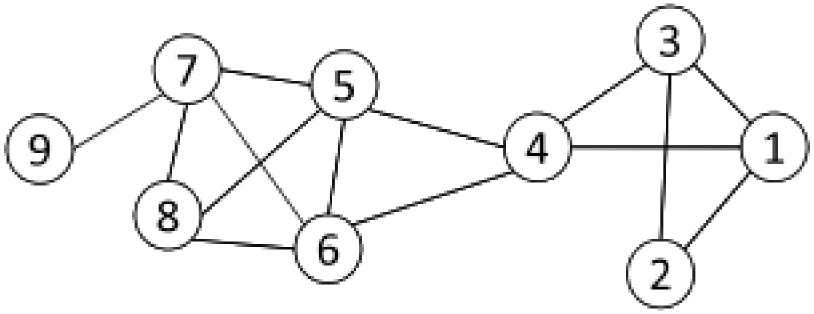
❖ Adjacency Matrix, A

Node	1	2	3	4	5	6	7	8	9
1	-	1	1	1	0	0	0	0	0
2	1	-	1	0	0	0	0	0	0
3	1	1	-	1	0	0	0	0	0
4	1	0	1	-	1	1	0	0	0
5	0	0	0	1	-	1	1	1	0
6	0	0	0	1	1	-	1	1	0
7	0	0	0	0	1	1	-	1	1
8	0	0	0	0	1	1	1	-	0
9	0	0	0	0	0	0	1	0	-

Degree Centrality

- The importance of a node is determined by the number of nodes adjacent to it
  - The larger the degree, the more important the node is
  - Only a small number of nodes have high degrees in many real-life networks (power law)
- **Degree Centrality:** $C_D(v_i) = d_i = \sum_j A_{ij}$
- **Normalized Degree Centrality:** $C'_D(v_i) = d_i / (n-1)$

n = total number of nodes



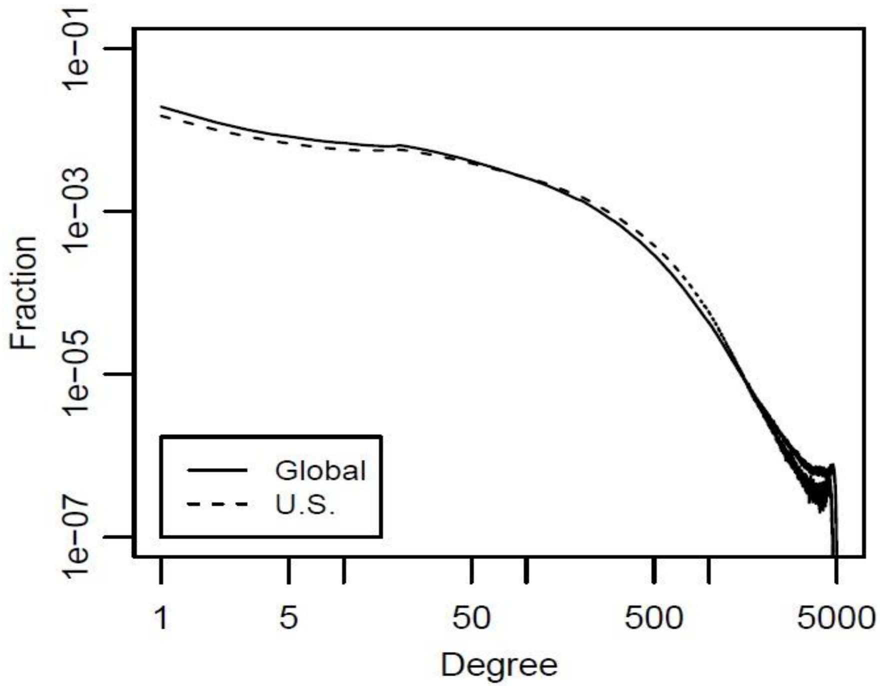
For node 1, degree centrality is 3;
Normalized degree centrality is
3/(9-1)=3/8.

Degree Distribution Plot (Revisited)

- On social networking sites, there exist many users with few connections and there exist a handful of users with very large numbers of friends. (**Power-law degree distribution**)

The x-axis represents the degree and the y-axis represents the fraction of nodes having that degree

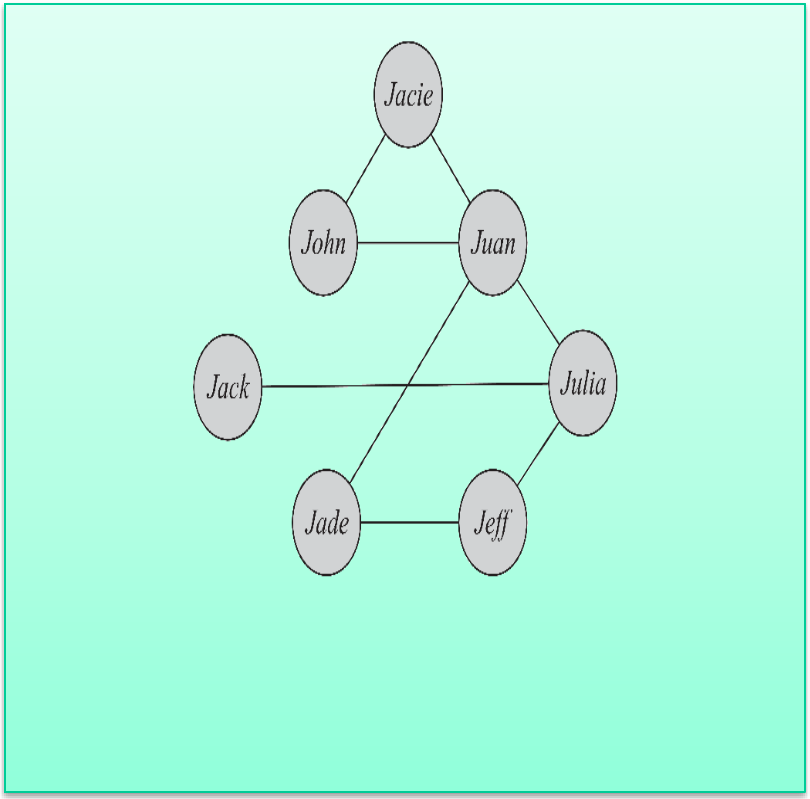
Facebook Degree Distribution



Degree Distribution (Revisited)

- When dealing with very large graphs, how nodes' degrees are distributed is an important concept to analyze and is called **Degree Distribution**

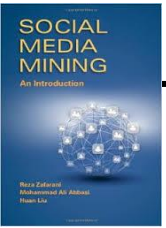
$$\pi(d) = \{d_1, d_2, \ldots, d_n\}$$



n_d is the number of nodes with degree d

$$p_d = \frac{n_d}{n}$$

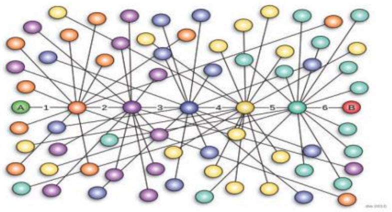
$$\sum_{d=0}^{\infty} p_d = 1$$



Network Modeling (Revisited)

- Large Networks demonstrate statistical patterns:
  - Small-world phenomenon (6 degrees of separation)
  - **Power-law distribution** (a.k.a. scale-free distribution)

$$f(x) = ax^k \qquad f(cx) = a(cx)^k = c^k f(x) \propto f(x).$$



  - Community structure (high clustering coefficient)
- Model the network dynamics
  - Find a mechanism such that the statistical patterns observed in large-scale networks can be reproduced.
  - Examples: random graph, preferential attachment process, Watts and Strogatz model
- Used for simulation to understand network properties
  - Thomas Shelling's famous simulation: What could cause the segregation
  - Network robustness under attack
  - Information diffusion within a given network structure

Power-Law Distribution

- When the frequency of an event changes as a **power** of an attribute
  - The frequency follows a **power-law**

Power-law intercept

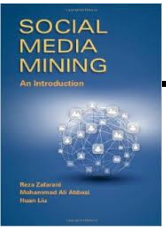
The power-law exponent and its value is typically in the range of **[2, 3]**

$$p_k = ak^{-b}$$

Fraction of users with degree k

Node degree

$$\ln p_k = -b \ln k + \ln a$$



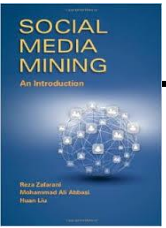
Power-Law Distribution: Examples

- **Call networks:**
  - The fraction of telephone numbers that receive k calls per day is roughly proportional to $\mathbf{1/k^2}$

- **Book Purchasing:**
  - The fraction of books that are bought by k people is roughly proportional to $\mathbf{1/k^3}$

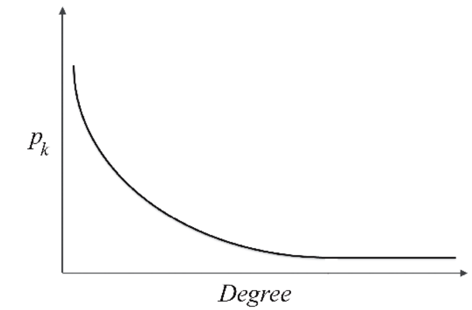
- **Scientific Papers:**
  - The fraction of scientific papers that receive k citations in total is roughly proportional to $\mathbf{1/k^3}$

- **Social Networks:**
  - The fraction of users that have in-degrees of k is roughly proportional to $\mathbf{1/k^2}$

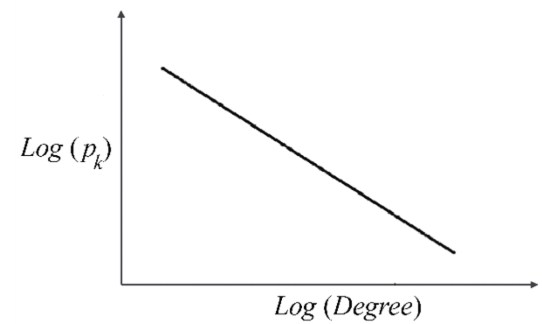


Power-Law Distribution

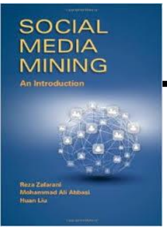
- Many real-world networks exhibit a power-law distribution.
- Power-laws seem to dominate
  - When the quantity being measured can be viewed as a type of **popularity**.
- A power-law distribution
  - **Small occurrences**: common
  - **Large instances**: extremely rare



(a) Power-Law Degree Distribution



(b) Log-Log Plot of Power-Law Degree Distribution



Power-Law Distribution: A Test

To test whether a network exhibits a power-law distribution

1. Pick a popularity measure and compute it for the whole network

   Example: number of friends for all nodes

2. Compute p_k, the fraction of individuals having popularity k.

3. Plot a log-log graph, where the x-axis represents $\ln k$ and the y-axis represents $\ln p_k$.

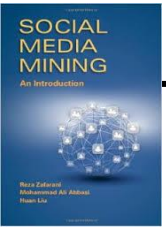
4. If a power-law distribution exists, we should observe a straight line

This is not a systematic approach!

1. Other distributions could also exhibit this pattern
2. The results [estimations for parameters] can be biased and incorrect

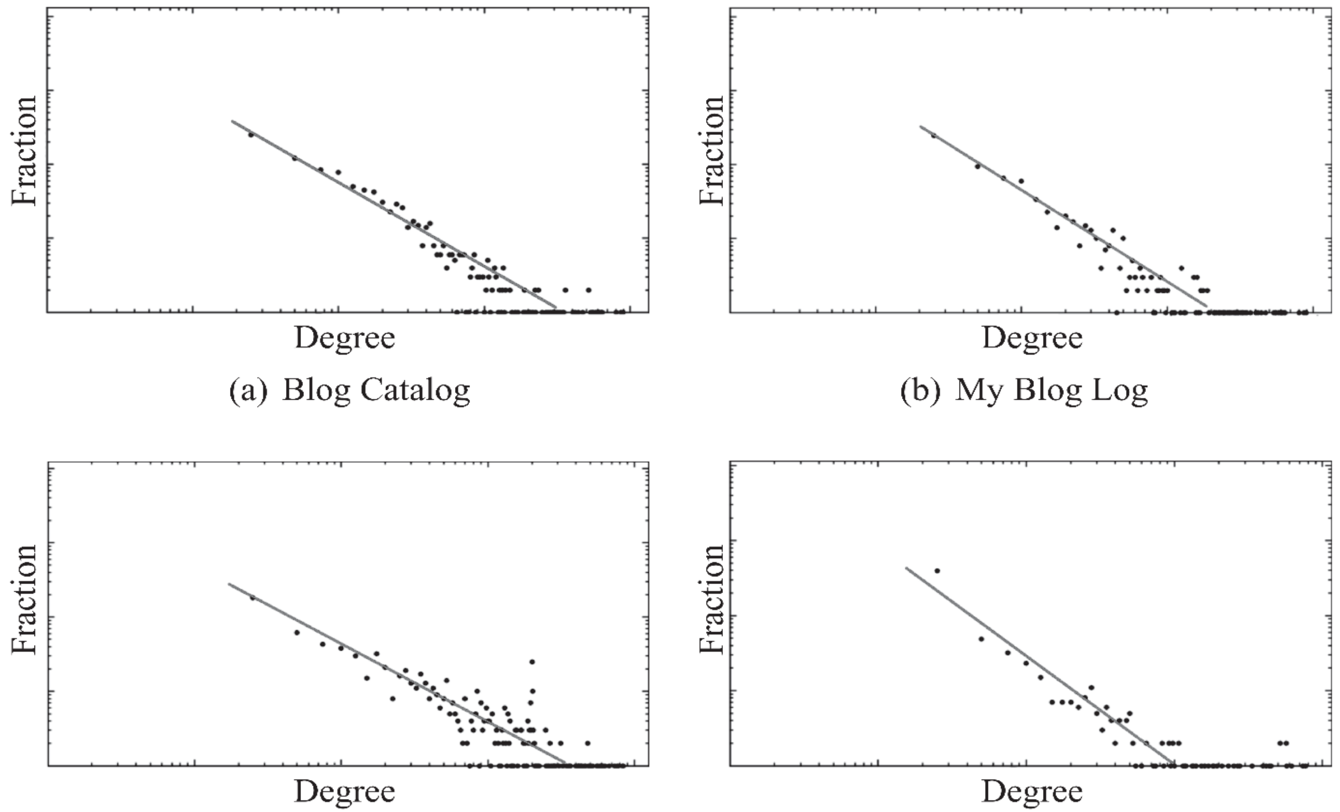
For a systematic approach see:

Clauset, Aaron, Cosma Rohilla Shalizi, and Mark EJ Newman. "Power-law distributions in empirical data." *SIAM review 51(4)* (2009): 661-703.

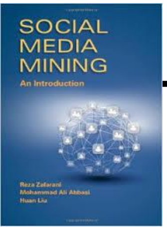


Real World Networks

- Networks with a power-law degree distribution are called **Scale-Free** networks

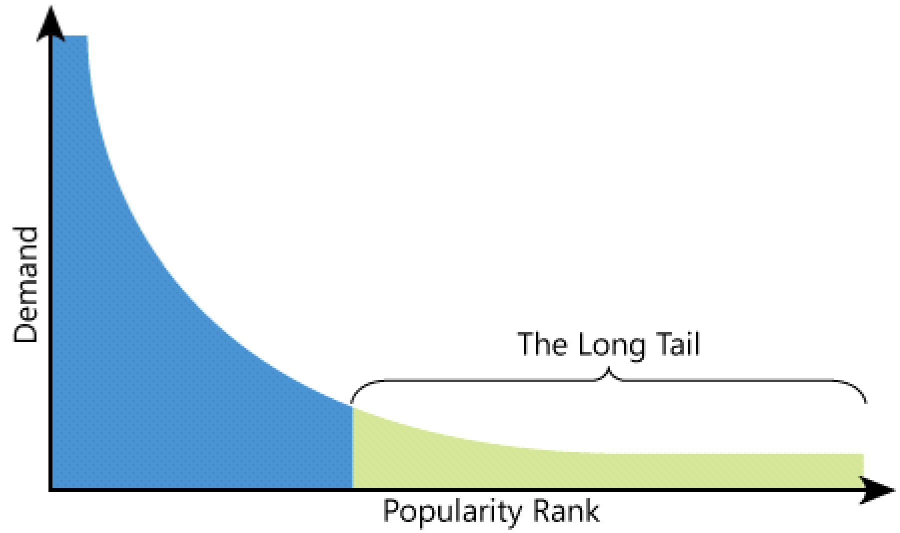


(a) Blog Catalog
(b) My Blog Log
(c) Twitter
(d) My Space



The Loooooong Tail

Are most sales being generated by a small set of items that are enormously popular?

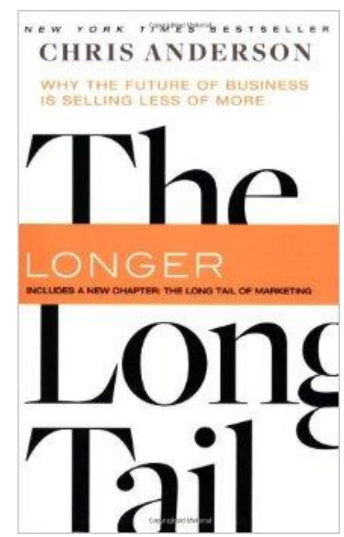


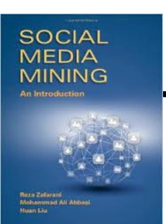
OR

By a much larger population of items that are each individually less popular?

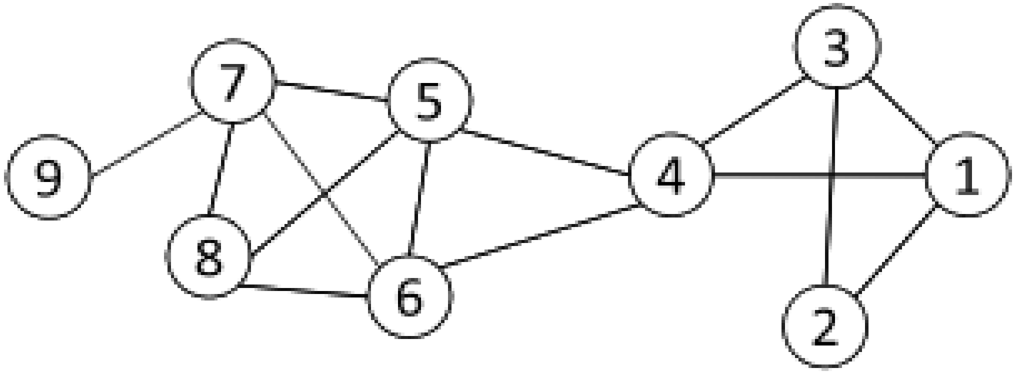
The total sales volume of unpopular items, taken together, is very significant.

- 57% of Amazon's sales is from the long tail





Degree Centrality



Node	1	2	3	4	5	6	7	8	9
1	-	1	1	1	0	0	0	0	0
2	1	-	1	0	0	0	0	0	0
3	1	1	-	1	0	0	0	0	0
4	1	0	1	-	1	1	0	0	0
5	0	0	0	1	-	1	1	1	0
6	0	0	0	1	1	-	1	1	0
7	0	0	0	0	1	1	-	1	1
8	0	0	0	0	1	1	1	-	0
9	0	0	0	0	0	0	1	0	-

Node	Degree Centrality	Node	Degree Centrality
1	0.375	6	**0.5**
2	0.25	7	**0.5**
3	0.375	8	0.375
4	**0.5**	9	0.125
5	**0.5**		

3.7 Centrality

3.7.1 Degree

degree_centrality(G)	Compute the degree centrality for nodes.
in_degree_centrality(G)	Compute the in-degree centrality for nodes.
out_degree_centrality(G)	Compute the out-degree centrality for nodes.

networkx.algorithms.centrality.degree_centrality

degree_centrality(*G*)

Compute the degree centrality for nodes.

The degree centrality for a node v is the fraction of nodes it is connected to.

Parameters **G** (*graph*) – A networkx graph

Returns **nodes** – Dictionary of nodes with degree centrality as the value.

Return type dictionary

See also:

betweenness_centrality(), *load_centrality()*, *eigenvector_centrality()*

Closeness Centrality

- "Central" nodes are important, as they can reach the whole network more quickly than non-central nodes
- Importance measured by how close a node is to other nodes
- Average Distance: $D_{avg}(v_i) = \frac{1}{n-1}\sum_{j \neq i}^{n} g(v_i, v_j)$

  $g(v_i, v_j)$ is the **geodesic (i.e. shortest)** distance between v_i and v_j
- **Closeness Centrality**

$$C_C(v_i) = \left[\frac{1}{n-1}\sum_{j \neq i}^{n} g(v_i, v_j)\right]^{-1} = \frac{n-1}{\sum_{j \neq i}^{n} g(v_i, v_j)}$$

Closeness Centrality Example

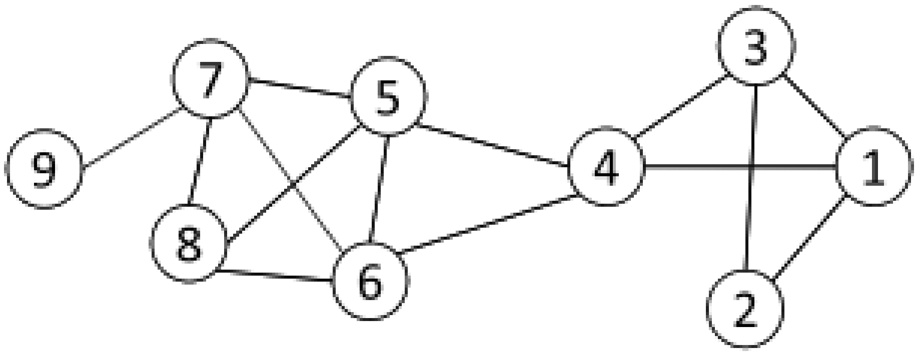


Table 2.1: Pairwise geodesic distance

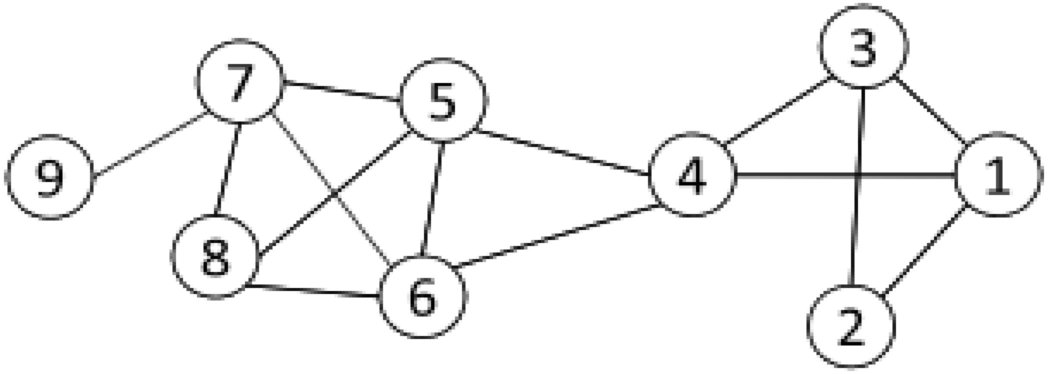
Node	1	2	3	4	5	6	7	8	9
1	0	1	1	1	2	2	3	3	4
2	1	0	1	2	3	3	4	4	5
3	1	1	0	1	2	2	3	3	4
4	1	2	1	0	1	1	2	2	3
5	2	3	2	1	0	1	1	1	2
6	2	3	2	1	1	0	1	1	2
7	3	4	3	2	1	1	0	1	1
8	3	4	3	2	1	1	1	0	2
9	4	5	4	3	2	2	1	2	0

$$C_C(3) = \frac{9-1}{1+1+1+2+2+3+3+4} = 8/17 = 0.47,$$

$$C_C(4) = \frac{9-1}{1+2+1+1+1+2+2+3} = 8/13 = 0.62.$$

Node 4 is more central than node 3

Closeness Centrality



Node	Closeness	Node	Closeness
1	0.470588235294	6	**0.615384615385**
2	0.347826086957	7	0.5
3	0.470588235294	8	0.470588235294
4	**0.615384615385**	9	0.347826086957
5	**0.615384615385**		

References

3.7.3 Closeness

closeness_centrality(G[, u, distance, …])	Compute closeness centrality for nodes.
incremental_closeness_centrality(G, edge[, …])	Incremental closeness centrality for nodes.

networkx.algorithms.centrality.closeness_centrality

closeness_centrality(*G*, *u=None*, *distance=None*, *wf_improved=True*)

Compute closeness centrality for nodes.

Closeness centrality[1] of a node `u` is the reciprocal of the average shortest path distance to `u` over all `n-1` reachable nodes.

$$C(u) = \frac{n - 1}{\sum_{v=1}^{n-1} d(v, u)},$$

where `d(v, u)` is the shortest-path distance between `v` and `u`, and `n` is the number of nodes that can reach `u`. Notice that the closeness distance function computes the incoming distance to `u` for directed graphs. To use outward distance, act on `G.reverse()`.

Notice that higher values of closeness indicate higher centrality.

Betweenness Centrality

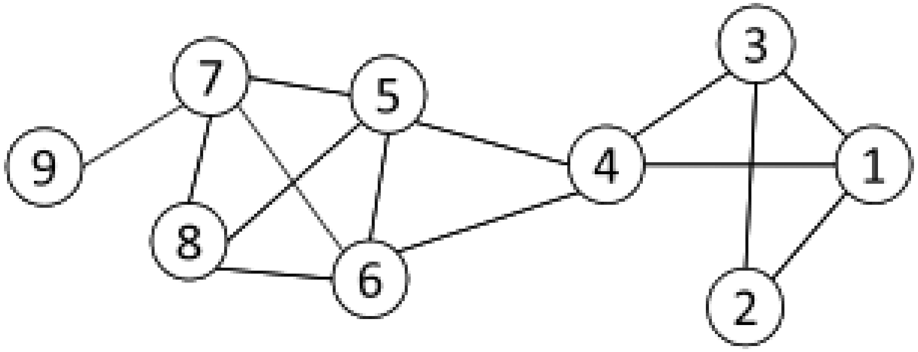
- Betweenness (for nodes) counts <u>the number of shortest paths that pass one node</u>
  - Nodes with high betweenness are important in communication and information diffusion
- **Betweenness Centrality**

$$C_B(v_i) = \sum_{v_s \neq v_i \neq v_t \in V, s<t} \frac{\sigma_{st}(v_i)}{\sigma_{st}}$$

σ_{st} : The number of shortest paths between s and t

$\sigma_{st}(v_i)$: The number of shortest paths between s and t that pass v_i

Betweenness Centrality Example



$$C_B(4) = 15$$

Table 2.2: $\sigma_{st}(4)/\sigma_{st}$

	$s=1$	$s=2$	$s=3$
$t=5$	1/1	2/2	1/1
$t=6$	1/1	2/2	1/1
$t=7$	2/2	4/4	2/2
$t=8$	2/2	4/4	2/2
$t=9$	2/2	4/4	2/2

What's the betweenness centrality for node 5?

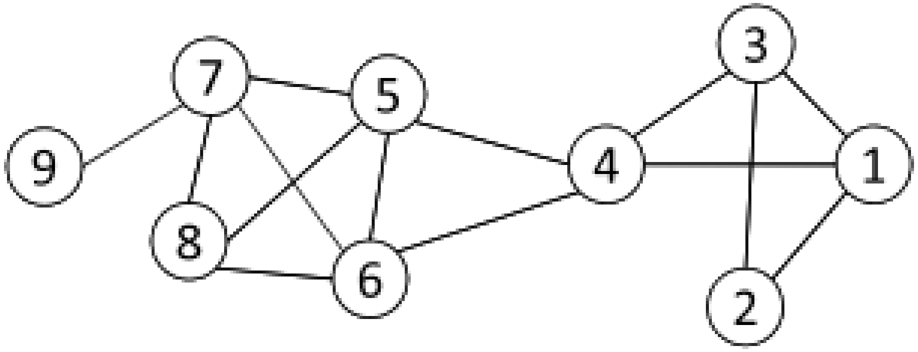
σ_{st} : The number of shortest paths between s and t

$\sigma_{st}(v_i)$: The number of shortest paths between s and t that pass v_i

$$C_B(v_i) = \sum_{v_s \neq v_i \neq v_t \in V, s<t} \frac{\sigma_{st}(v_i)}{\sigma_{st}}$$

Betweenness Centrality – Node 5

	s=1	s=2	s=3	s=4
t=7	1/2	2/4	1/2	1/2
t=8	1/2	2/4	1/2	1/2
t=9	1/2	2/4	1/2	1/2

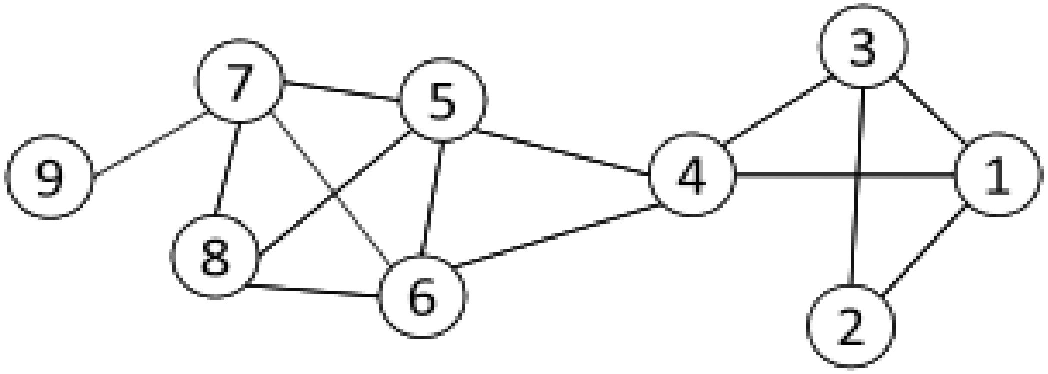


σ_{st} : The number of shortest paths between s and t

$\sigma_{st}(v_i)$: The number of shortest paths between s and t that pass v_i

$$C_B(v_i) = \sum_{v_s \neq v_i \neq v_t \in V, s<t} \frac{\sigma_{st}(v_i)}{\sigma_{st}}$$

Betweenness Centrality



Node	Betweenness	Node	Betweenness
1	3	6	6
2	0	7	7
3	3	8	0
4	**15**	9	0
5	6		

Normalized Betweenness Centrality

- ❖ Since the maximum value of $C_B(v_i)$ in an undirected network can be:

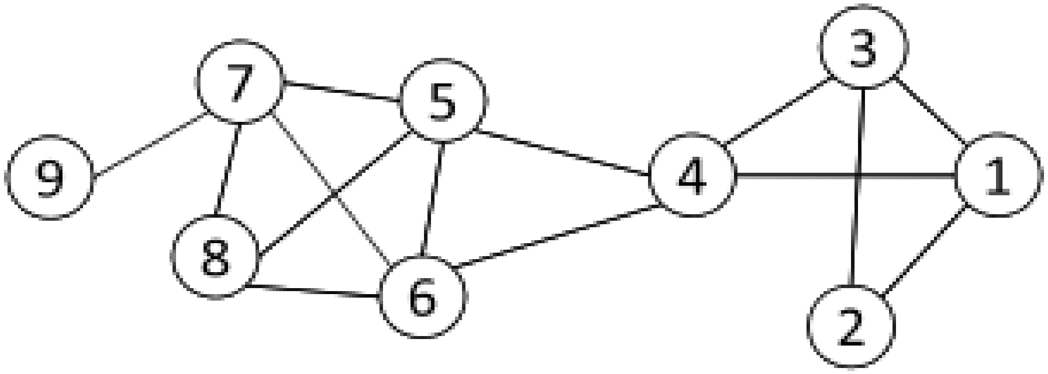
$$\binom{n-1}{2} = (n-1)(n-2)/2$$

- ❖ We can normalize the betweenness centrality as:

$$C_B^{'}(v_i) = \frac{C_B(v_i)}{(n-1)(n-2)/2}$$

$$C_B(v_i) = \sum_{v_s \neq v_i \neq v_t \in V, s<t} \frac{\sigma_{st}(v_i)}{\sigma_{st}}$$

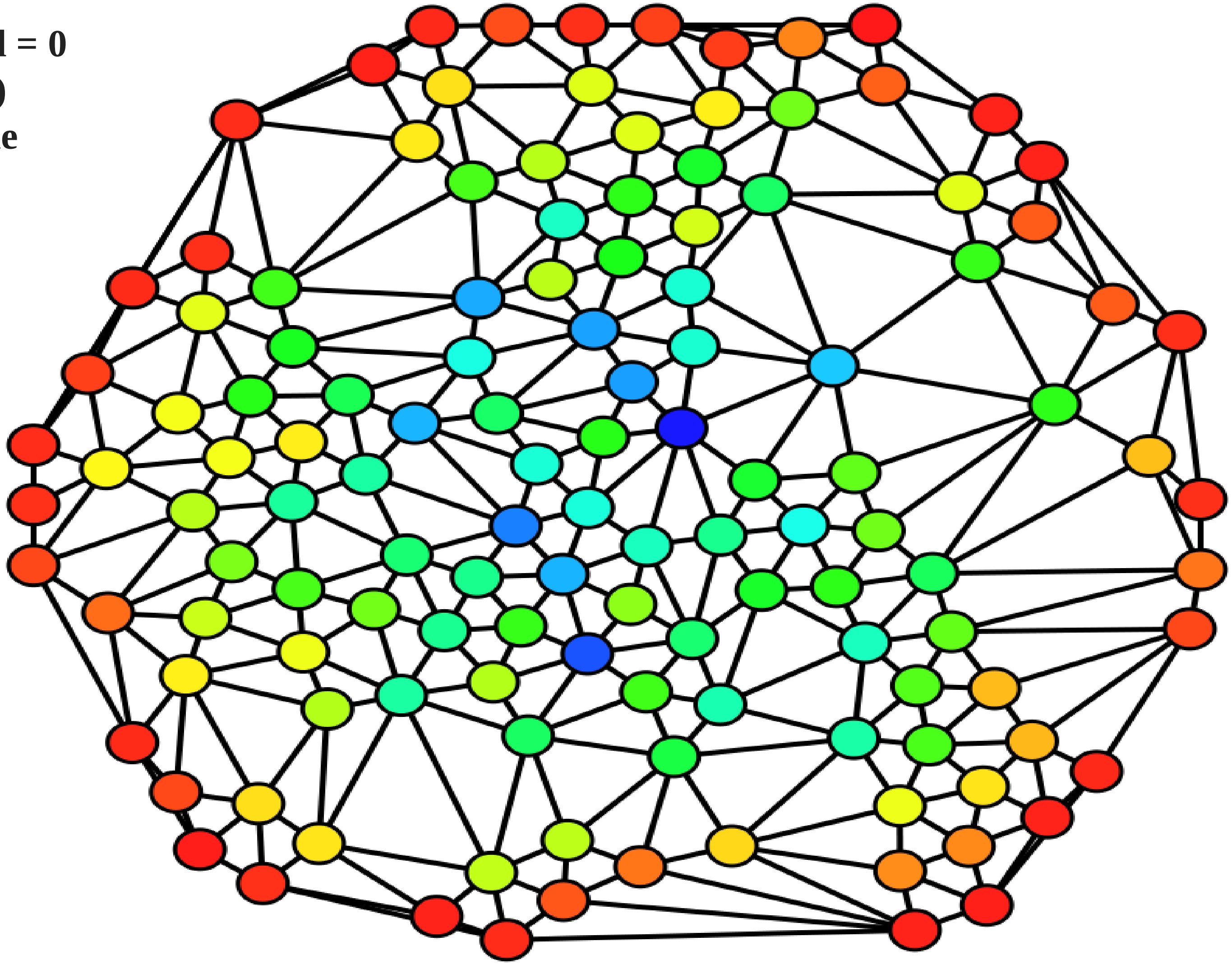
Normalized Betweenness Centrality



Node	Betweenness	Node	Betweenness
1	0.107142857143	6	0.214285714286
2	0	7	0.25
3	0.107142857143	8	0
4	**0.535714285714**	9	0
5	0.214285714286		

Normalized Betweenness Centrality

Hue (from red = 0 to blue = max) shows the node betweenness.



https://en.wikipedia.org/wiki/Centrality

References

3.7.5 (Shortest Path) Betweenness

betweenness_centrality(G[, k, normalized, ...])	Compute the shortest-path betweenness centrality for nodes.
edge_betweenness_centrality(G[, k, ...])	Compute betweenness centrality for edges.
betweenness_centrality_subset(G, sources, ...)	Compute betweenness centrality for a subset of nodes.
edge_betweenness_centrality_subset(G, ...[, ...])	Compute betweenness centrality for edges for a subset of nodes.

networkx.algorithms.centrality.betweenness_centrality

betweenness_centrality (*G, k=None, normalized=True, weight=None, endpoints=False, seed=None*)

Compute the shortest-path betweenness centrality for nodes.

Betweenness centrality of a node v is the sum of the fraction of all-pairs shortest paths that pass through v

$$c_B(v) = \sum_{s,t \in V} \frac{\sigma(s,t|v)}{\sigma(s,t)}$$

where V is the set of nodes, $\sigma(s,t)$ is the number of shortest (s,t)-paths, and $\sigma(s,t|v)$ is the number of those paths passing through some node v other than s,t. If $s = t$, $\sigma(s,t) = 1$, and if $v \in s,t$, $\sigma(s,t|v) = 0^2$.

`betweenness_centrality` (*G*, *k=None*, *normalized=True*, *weight=None*, *endpoints=False*, *seed=None*)

Compute the shortest-path betweenness centrality for nodes.

Betweenness centrality of a node v is the sum of the fraction of all-pairs shortest paths that pass through v

$$c_B(v) = \sum_{s,t \in V} \frac{\sigma(s,t|v)}{\sigma(s,t)}$$

where V is the set of nodes, $\sigma(s,t)$ is the number of shortest (s,t)-paths, and $\sigma(s,t|v)$ is the number of those paths passing through some node v other than s, t. If $s = t$, $\sigma(s,t) = 1$, and if $v \in s, t$, $\sigma(s,t|v) = 0^2$.

Parameters

- **G** (*graph*) – A NetworkX graph.
- **k** (*int, optional (default=None)*) – If k is not None use k node samples to estimate betweenness. The value of k <= n where n is the number of nodes in the graph. Higher values give better approximation.
- **normalized** (*bool, optional*) – If True the betweenness values are normalized by `2/((n-1)(n-2))` for graphs, and `1/((n-1)(n-2))` for directed graphs where `n` is the number of nodes in G.
- **weight** (*None or string, optional (default=None)*) – If None, all edge weights are considered equal. Otherwise holds the name of the edge attribute used as weight.
- **endpoints** (*bool, optional*) – If True include the endpoints in the shortest path counts.
- **seed** (*integer, random_state, or None (default)*) – Indicator of random number generation state. See *Randomness*. Note that this is only used if k is not None.

Returns **nodes** – Dictionary of nodes with betweenness centrality as the value.

Eigenvector Centrality

- One's importance is determined by his friends' importance
- If one has many important friends, he should be important as well.

Node	1	2	3	4	5	6	7	8	9
1	-	1	1	1	0	0	0	0	0
2	1	-	1	0	0	0	0	0	0
3	1	1	-	1	0	0	0	0	0
4	1	0	1	-	1	1	0	0	0
5	0	0	0	1	-	1	1	1	0
6	0	0	0	1	1	-	1	1	0
7	0	0	0	0	1	1	-	1	1
8	0	0	0	0	1	1	1	-	0
9	0	0	0	0	0	0	1	0	-

$$C_E(v_i) \propto \sum_{v_j \in N_i} A_{ij} C_E(v_j)$$

$$\mathbf{x} \propto A\mathbf{x} \Rightarrow A\mathbf{x} = \lambda \mathbf{x}.$$

- The centrality corresponds to the principal eigenvector of the adjacency matrix A.
- A variant of this eigenvector centrality is the PageRank score.

Gray Cardinal

- An éminence grise (French for “gray cardinal”) is a powerful advisor who operates secretly and unofficially.
  - This phrase originally referred to François Leclerc du Tremblay, the right-hand man of Cardinal Richelieu.
  - Although he never achieved that rank (cardinal), people around him may have addressed him thus in reference to the considerable influence over the real Cardinal.
- Grey cardinals exist in social networks, so we must find a way to find them!
- Philip Bonacich proposed that instead of simply adding the number of links to compute degree, <u>one should weight each of the links by the degree of the node at the other end of the link</u>

(Bonacich, P. "Some unique properties of eigenvector centrality." *Social Networks 29 (2007), 555-564)*

Éminence grise

From Wikipedia, the free encyclopedia

An ***éminence grise*** (French pronunciation: [eminɑ̃s ɡʁiz]) or **grey eminence** is a powerful decision-maker or adviser who operates "behind the scenes", or in a non-public or unofficial capacity.

François Leclerc du Tremblay is the figure in black, depicted descending the staircase in this oil painting (1873) by Jean-Léon Gérôme.

Contents [hide]

Origin [edit]

This phrase originally referred to François Leclerc du Tremblay, the right-hand man of Cardinal Richelieu.[1] Leclerc was a Capuchin friar who was renowned for his beige robe attire, as beige was termed "grey" in that era. The style *His Eminence* is used to address or refer to a cardinal in the Roman Catholic Church.[2] Although Leclerc never achieved the rank of cardinal, those around him addressed him as such in deference to the considerable influence this "grey" friar held over "His Eminence the Cardinal".[3]

Leclerc is referred to in several popular works. Aldous Huxley wrote an English biography of Leclerc entitled *Grey Eminence*. There is also an 1873 painting by Jean-Léon Gérôme, *L'Éminence Grise*, which depicts him descending the grand staircase of the Palais Cardinal and the deference shown to him by others present. Leclerc is referred to in Alexandre Dumas' *The Three Musketeers* as the character Father Joseph, a powerful associate of Richelieu and one to be feared.

Historical examples [edit]

- Empress Jia and Empress Dowager Cixi are two examples of women who essentially ruled Imperial Chinese dynasties. As women were barred from reigning in their

Eigenvector Centrality

❖ One's importance is determined by his friends' importance

❖ If one has many important friends, he should be important as well.

$$C_E(v_i) \propto \sum_{v_j \in N_i} A_{ij} C_E(v_j)$$

$$\mathbf{x} \propto A\mathbf{x} \quad \Rightarrow \quad A\mathbf{x} = \lambda\mathbf{x}.$$

Node	1	2	3	4	5	6	7	8	9
1	-	1	1	1	0	0	0	0	0
2	1	-	1	0	0	0	0	0	0
3	1	1	-	1	0	0	0	0	0
4	1	0	1	-	1	1	0	0	0
5	0	0	0	1	-	1	1	1	0
6	0	0	0	1	1	-	1	1	0
7	0	0	0	0	1	1	-	1	1
8	0	0	0	0	1	1	1	-	0
9	0	0	0	0	0	0	1	0	-

❖ The centrality corresponds to the principal eigenvector of the adjacency matrix A.

❖ A variant of this eigenvector centrality is the PageRank score.

Principal Eigenvector

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

From A$\mathbf{x} = \lambda\mathbf{x}$

We have $(A - \lambda I)\mathbf{x} = 0$, where I is the identity matrix

$\rightarrow \quad |(A - \lambda I)| = 0$, where | . | is its determinant

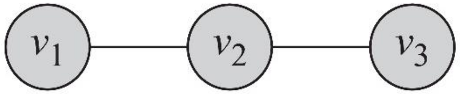
$$\rightarrow \quad \begin{vmatrix} -\lambda & \mathbf{1} \\ \mathbf{1} & -\lambda \end{vmatrix}$$

$\rightarrow \quad \lambda^2 - 1 = 0$

$\rightarrow \quad \lambda = \mathbf{1}$, or -1

$\rightarrow \quad$ the principal eigenvector $\mathbf{x}$ (for 1) is $\begin{pmatrix} \sqrt{1/2} \\ \sqrt{1/2} \end{pmatrix}$

Principal Eigenvector



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

$$\lambda \mathbf{C}_e = A\mathbf{C}_e \qquad (A - \lambda I)\mathbf{C}_e = 0 \quad \mathbf{C}_e = [u_1\ u_2\ u_3]^T$$

$$\begin{bmatrix} 0-\lambda & 1 & 0 \\ 1 & 0-\lambda & 1 \\ 0 & 1 & 0-\lambda \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$det(A - \lambda I) = \begin{vmatrix} 0-\lambda & 1 & 0 \\ 1 & 0-\lambda & 1 \\ 0 & 1 & 0-\lambda \end{vmatrix} = 0$$

$$(-\lambda)(\lambda^2 - 1) - 1(-\lambda) = 2\lambda - \lambda^3 = \lambda(2 - \lambda^2) = 0$$

Eigenvalues are

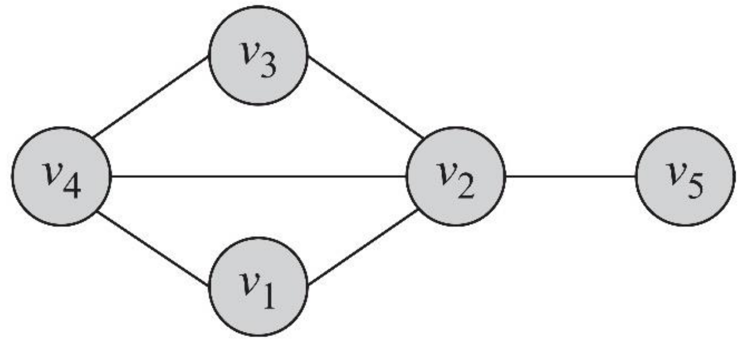
$$(-\sqrt{2}, 0, \boxed{+\sqrt{2}})$$

Largest Eigenvalue

Corresponding eigenvector (assuming $\mathbf{C_e}$ has norm 1)

$$\begin{bmatrix} 0-\sqrt{2} & 1 & 0 \\ 1 & 0-\sqrt{2} & 1 \\ 0 & 1 & 0-\sqrt{2} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \qquad \mathbf{C}_e = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} 1/2 \\ \sqrt{2}/2 \\ 1/2 \end{bmatrix}$$

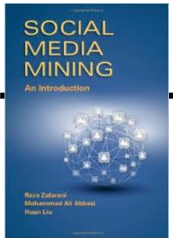
Principal Eigenvector



$$A = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \Rightarrow \lambda = (2.68,\ -1.74,\ -1.27,\ 0.33,\ 0.00)$$

Eigenvalues Vector

$$\lambda_{max} = 2.68 \Rightarrow C_e = \begin{bmatrix} 0.4119 \\ 0.5825 \\ 0.4119 \\ 0.5237 \\ 0.2169 \end{bmatrix}$$



3.6.2 Eigenvector

eigenvector_centrality(G[, max_iter, tol, …])	Compute the eigenvector centrality for the graph G.
eigenvector_centrality_numpy(G[, weight, …])	Compute the eigenvector centrality for the graph G.
katz_centrality(G[, alpha, beta, max_iter, …])	Compute the Katz centrality for the nodes of the graph G.
katz_centrality_numpy(G[, alpha, beta, …])	Compute the Katz centrality for the graph G.

networkx.algorithms.centrality.eigenvector_centrality

eigenvector_centrality(*G*, *max_iter=100*, *tol=1e-06*, *nstart=None*, *weight=None*)

Compute the eigenvector centrality for the graph `G`.

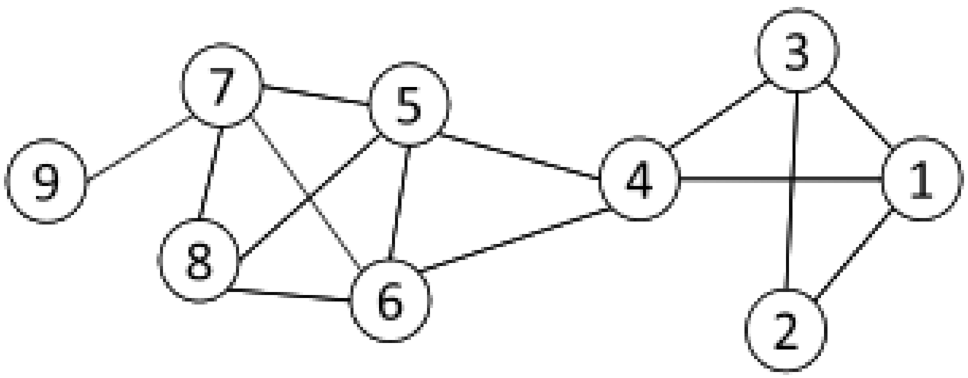
Eigenvector centrality computes the centrality for a node based on the centrality of its neighbors. The eigenvector centrality for node i is the i-th element of the vector x defined by the equation

$$Ax = \lambda x$$

where A is the adjacency matrix of the graph `G` with eigenvalue λ. By virtue of the Perron–Frobenius theorem, there is a unique solution x, all of whose entries are positive, if λ is the largest eigenvalue of the adjacency matrix A ([2]).

Parameters

Eigenvector Centrality



Node	1	2	3	4	5	6	7	8	9
1	-	1	1	1	0	0	0	0	0
2	1	-	1	0	0	0	0	0	0
3	1	1	-	1	0	0	0	0	0
4	1	0	1	-	1	1	0	0	0
5	0	0	0	1	-	1	1	1	0
6	0	0	0	1	1	-	1	1	0
7	0	0	0	0	1	1	-	1	1
8	0	0	0	0	1	1	1	-	0
9	0	0	0	0	0	0	1	0	-

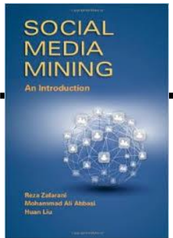
Node	Eigenvector Centrality	Node	Eigenvector Centrality
1	0.195751798216	6	**0.468084576647**
2	0.111686197298	7	0.409977787365
3	0.195751798216	8	0.384018975728
4	0.37874977785	9	0.116955395176
5	**0.468084576647**		

Eigenvector Centrality

Theorem 1 (Perron-Frobenius Theorem). *Let $A \in \mathbb{R}^{n \times n}$ represent the adjacency matrix for a [strongly] connected graph or $A : A_{i,j} > 0$ (i.e. a positive n by n matrix). There exists a positive real number (Perron-Frobenius eigenvalue) $\lambda_{\max}$, such that $\lambda_{\max}$ is an eigenvalue of A and any other eigenvalue of A is strictly smaller than $\lambda_{\max}$. Furthermore, there exists a corresponding eigenvector $\mathbf{v} = (v_1, v_2, \ldots, v_n)$ of A with eigenvalue $\lambda_{\max}$ such that $\forall v_i > 0$.*

So, to compute eigenvector centrality of A,

1. We compute the eigenvalues of A
2. Select the largest eigenvalue λ
3. The corresponding eigenvector of λ is $\mathbf{C_e}$.
4. Based on the Perron-Frobenius theorem, all the components of $\mathbf{C_e}$ will be positive
5. The components of $\mathbf{C_e}$ are the eigenvector centralities for the graph.



Eigenvector Centrality

- One's importance is determined by his friends' importance

- If one has many important friends, he should be important as well.

$$C_E(v_i) \propto \sum_{v_j \in N_i} A_{ij} C_E(v_j)$$

$$\mathbf{x} \propto A\mathbf{x} \quad \Rightarrow \quad A\mathbf{x} = \lambda\mathbf{x}.$$

Node	1	2	3	4	5	6	7	8	9
1	-	1	1	1	0	0	0	0	0
2	1	-	1	0	0	0	0	0	0
3	1	1	-	1	0	0	0	0	0
4	1	0	1	-	1	1	0	0	0
5	0	0	0	1	-	1	1	1	0
6	0	0	0	1	1	-	1	1	0
7	0	0	0	0	1	1	-	1	1
8	0	0	0	0	1	1	1	-	0
9	0	0	0	0	0	0	1	0	-

- The centrality corresponds to the <u>principal</u> eigenvector of the adjacency matrix A.

- A variant of this eigenvector centrality is the **PageRank** score.

PageRank

Table 2.5: Column-Normalized Adjacency Matrix

Node	1	2	3	4	5	6	7	8	9
1	0	1/2	1/3	1/4	0	0	0	0	0
2	1/3	0	1/3	0	0	0	0	0	0
3	1/3	1/2	0	1/4	0	0	0	0	0
4	1/3	0	1/3	0	1/4	1/4	0	0	0
5	0	0	0	1/4	0	1/4	1/4	1/3	0
6	0	0	0	1/4	1/4	0	1/4	1/3	0
7	0	0	0	0	1/4	1/4	0	1/3	1
8	0	0	0	0	1/4	1/4	1/4	0	0
9	0	0	0	0	0	0	1/4	0	0

Suppose we start from $\mathbf{x}^{(0)} = \mathbf{1}$, then $x^{(1)} \propto \widetilde{A}x^{(0)}$, $x^{(2)} \propto \widetilde{A}x^{(1)}$, etc. Typically, the vector $\mathbf{x}$ is normalized to the unit length. Below, we show the values of $\mathbf{x}$ in the first seven iterations.

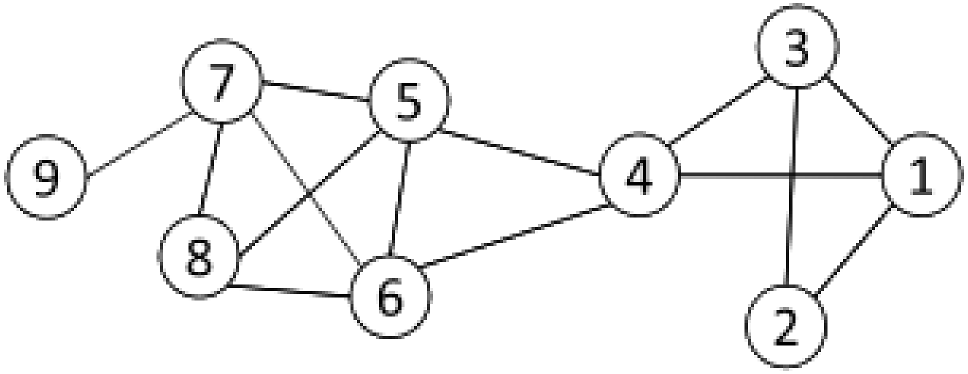
The power method

$$\mathbf{x}^{(0)} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \quad \mathbf{x}^{(1)} = \begin{bmatrix} 0.33 \\ 0.21 \\ 0.33 \\ 0.36 \\ 0.33 \\ 0.33 \\ 0.57 \\ 0.23 \\ 0.08 \end{bmatrix}, \quad \mathbf{x}^{(2)} = \begin{bmatrix} 0.32 \\ 0.23 \\ 0.32 \\ 0.41 \\ 0.41 \\ 0.41 \\ 0.34 \\ 0.32 \\ 0.15 \end{bmatrix}, \quad \mathbf{x}^{(3)} = \begin{bmatrix} 0.32 \\ 0.21 \\ 0.32 \\ 0.41 \\ 0.39 \\ 0.39 \\ 0.45 \\ 0.28 \\ 0.08 \end{bmatrix},$$

$$\mathbf{x}^{(4)} = \begin{bmatrix} 0.32 \\ 0.21 \\ 0.32 \\ 0.41 \\ 0.41 \\ 0.41 \\ 0.37 \\ 0.31 \\ 0.11 \end{bmatrix}, \quad \mathbf{x}^{(5)} = \begin{bmatrix} 0.31 \\ 0.21 \\ 0.31 \\ 0.41 \\ 0.40 \\ 0.40 \\ 0.42 \\ 0.30 \\ 0.10 \end{bmatrix}, \quad \mathbf{x}^{(6)} = \begin{bmatrix} 0.31 \\ 0.21 \\ 0.31 \\ 0.41 \\ 0.41 \\ 0.41 \\ 0.39 \\ 0.31 \\ 0.11 \end{bmatrix}, \quad \mathbf{x}^{(7)} = \begin{bmatrix} 0.31 \\ 0.21 \\ 0.31 \\ 0.41 \\ 0.40 \\ 0.40 \\ 0.41 \\ 0.30 \\ 0.10 \end{bmatrix}.$$

After convergence, we have the Pagerank scores for each node listed in Table 2.6. Based on

Eigenvector Centrality – PageRank



Node	PageRank	Node	PageRank
1	0.31	6	**0.41**
2	0.20	7	**0.41**
3	0.31	8	0.31
4	**0.41**	9	0.10
5	**0.41**		

3.34 Link Analysis

3.34.1 PageRank

PageRank analysis of graph structure.

pagerank(G[, alpha, personalization, …])	Return the PageRank of the nodes in the graph.
pagerank_numpy(G[, alpha, personalization, …])	Return the PageRank of the nodes in the graph.
pagerank_scipy(G[, alpha, personalization, …])	Return the PageRank of the nodes in the graph.
google_matrix(G[, alpha, personalization, …])	Return the Google matrix of the graph.

networkx.algorithms.link_analysis.pagerank_alg.pagerank

pagerank (*G*, *alpha=0.85*, *personalization=None*, *max_iter=100*, *tol=1e-06*, *nstart=None*, *weight='weight'*, *dangling=None*)

Return the PageRank of the nodes in the graph.

PageRank computes a ranking of the nodes in the graph G based on the structure of the incoming links. It was originally designed as an algorithm to rank web pages.

Parameters

- **G** (*graph*) – A NetworkX graph. Undirected graphs will be converted to a directed graph with two directed edges for each undirected edge.

A Variation of EC: Katz Centrality

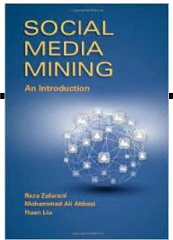
- A major problem with eigenvector centrality arises when it deals with <u>directed graphs</u>
- Centrality only passes over *outgoing* edges and in special cases such as when a node is in a directed acyclic graph centrality becomes zero
  - The node can have many edge connected to it

- *Elihu Katz*

- To resolve this problem we add bias term β to the centrality values for all nodes

Eigenvector Centrality

$$C_{\text{Katz}}(v_i) = \boxed{\alpha \sum_{j=1}^{n} A_{j,i} C_{\text{Katz}}(v_j)} + \beta$$



katz_centrality (*G*, *alpha=0.1*, *beta=1.0*, *max_iter=1000*, *tol=1e-06*, *nstart=None*, *normalized=True*, *weight=None*)

Compute the Katz centrality for the nodes of the graph G.

Katz centrality computes the centrality for a node based on the centrality of its neighbors. It is a generalization of the eigenvector centrality. The Katz centrality for node i is

$$x_i = \alpha \sum_j A_{ij} x_j + \beta,$$

where A is the adjacency matrix of graph G with eigenvalues λ.

The parameter β controls the initial centrality and

$$\alpha < \frac{1}{\lambda_{\max}}.$$

[1] Phillip Bonacich: Power and Centrality: A Family of Measures. American Journal of Sociology 92(5):1170–1182, 1986 http://www.leonidzhukov.net/hse/2014/socialnetworks/papers/Bonacich-Centrality.pdf

Klout
(Defunct)

Klout

From Wikipedia, the free encyclopedia

Klout was a website and mobile app that used social media analytics to rate its users according to online social influence via the "Klout Score", which was a numerical value between 1 and 100. In determining the user score, Klout measured the size of a user's social media network and correlated the content created to measure how other users interact with that content.[3] Klout launched in 2008.[4]

Lithium Technologies, who acquired the site in March 2014, closed the service on May 25, 2018.[5][6]

Klout used Bing, Facebook, Foursquare, Google+, Instagram, LinkedIn, Twitter, YouTube, and Wikipedia data to create Klout user profiles that were assigned a unique "Klout Score".[7] Klout scores ranged from 1 to 100, with higher scores corresponding to a higher ranking of the breadth and strength of one's online social influence. While all Twitter users were assigned a score, users who registered at Klout could link multiple social networks, of which network data was then aggregated to influence the user's Klout Score.

Klout

KLOUT

Type of business	Subsidiary
Type of site	Social Networking
Available in	English
Headquarters	San Francisco, California, United States
Area served	Worldwide
Owner	Lithium Technologies
Founder(s)	Joe Fernandez Binh Tran
Key people	Joe Fernandez (CEO) Emil Michael (COO)
Employees	40[1]
Website	klout.com
Alexa rank	▲ 27,297 (April 2018)[2]
Advertising	No
Registration	Optional
Launched	2008

Contents [hide]

The Klout Score

- Klout is a real-world example of <u>centrality analysis</u>
  - Klout Score (www.klout.com) is a metric computed upon all of your social media activity
  - Klout samples a variety of activities that a person can engage in—from Twitter posts and number of followers, to activity on Facebook and a number of other networks
    - Klout has access to the full Twitter "firehose" feed
  - Klout applies a proprietary formula to weigh all of its users on a percentage scale, with celebrities topping the scale at 100%, a strong "middle class" inhabiting the 20%-50% range, and the majority of casual Twitter users located way below this mark

https://klout.com/corp/score

Klout Score

THE KLOUT SCORE

The Klout Score is a number between 1-100 that represents your influence. The more influential you are, the higher your Klout Score.






The Klout Score

- The Klout Score currently incorporates more than 400 signals from seven different networks, including:

The Klout Score

Facebook:

- **Mentions:** A mention of your name in a post indicates an effort to engage with you directly.
- **Likes:** The simplest action that shows engagement with the content you create.
- **Comments:** As a reaction to content you share, comments also reflect direct engagement by your network.
- **Subscribers:** Subscriber count is a more persistent measure of influence that grows over time.
- **Wall Posts:** Posts to your wall indicate both influence and engagement.
- **Friends:** Friend count measures the reach of your network but is less important than how your network engages with your content.

The Klout Score

Twitter

- ❖ **Retweets:** Retweets increase your influence by exposing your content to extended follower networks.
- ❖ **Mentions:** People seeking your attention by mentioning you is a strong signal of influence.
- ❖ **List Memberships:** Being included on lists curated by other users demonstrates your areas of influence.
- ❖ **Followers:** Follower count is one factor in your Score, but Klout heavily favors engagement over size of audience.
- ❖ **Replies:** Replies show that you are consistently engaging your network with quality content.

The Klout Score

Google+

- ❖ **Comments:** As a reaction to content you share, comments also reflect direct engagement by your network.
- ❖ **+1′s:** The simplest action that shows engagement with the content you create.
- ❖ **Reshares:** Reshares increase your influence by exposing your content to extended networks on Google+.

The Klout Score

LinkedIn

- **Title:** Your reported title on LinkedIn is a signal of your real-world influence and is persistent.
- **Connections:** Your connection graph helps validate your real-world influence.
- **Recommenders:** The recommenders in your network add additional signals to the contribution LinkedIn makes to your Score.
- **Comments:** As a reaction to content you share, comments also reflect direct engagement by your network.

The Klout Score

Wikipedia

- **Page Importance:** Measured by applying a PageRank algorithm against the Wikipedia page graph.
- **Number of Inlinks:** Measures the total number of inbound links to a page.
- **Inlinks to Outlinks Ratio:** Compares the number of inbound links to a page to the number of outbound links.

The Klout Score

Foursquare

- ❖ **Tips Done:** The number of suggestions you’ve left that have been <u>completed</u> indicate your ability to influence others on foursquare.

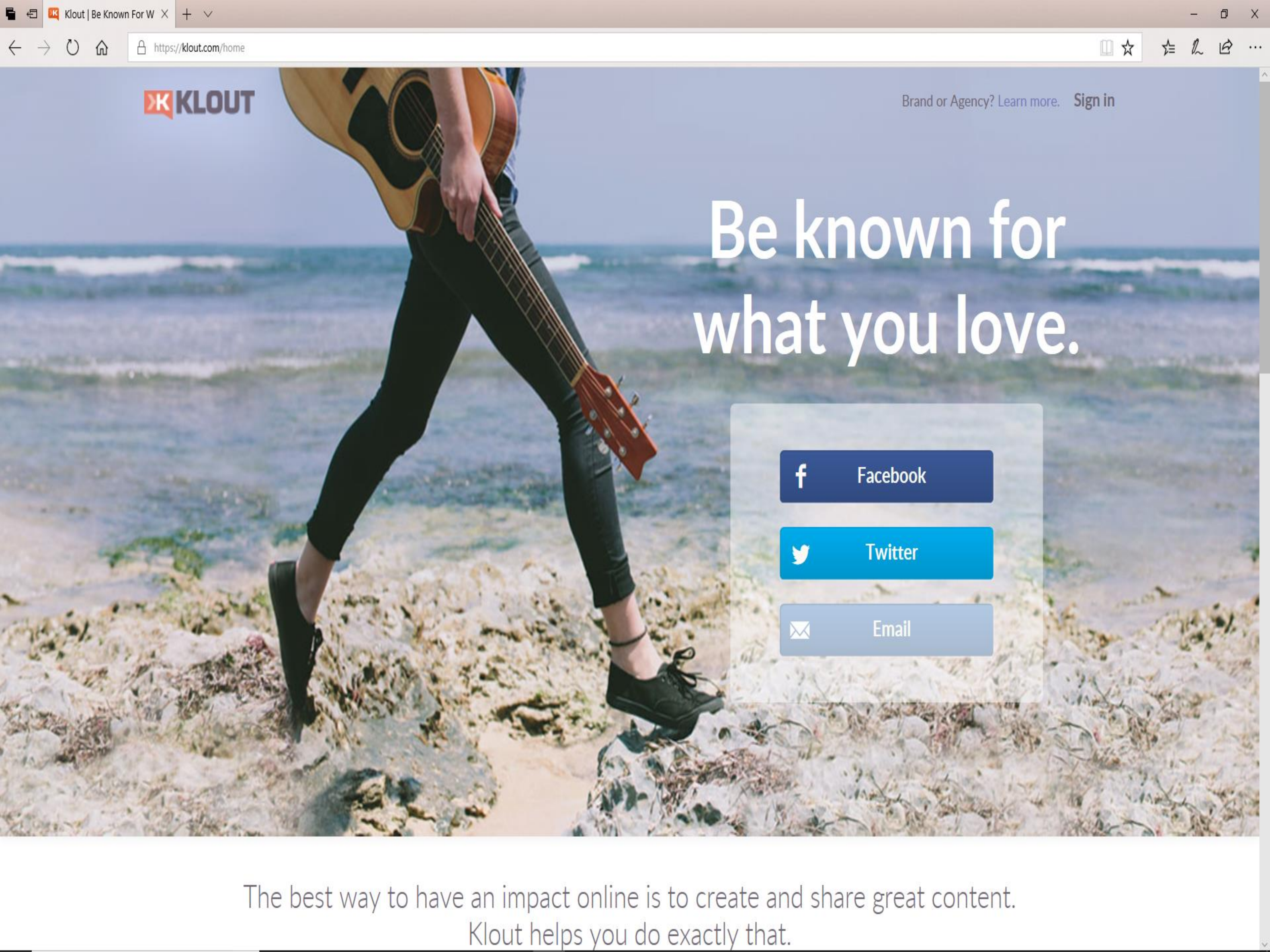
Klout

- ❖ **+K received:** Receiving +K increases your Klout Score by an amount that is capped in every 90-day measurement cycle to protect the integrity of the Score.

The Klout Score

…

In spite of the controversy, some employers are making hiring decisions based on Klout scores. In an article for Wired, a man recruited for a VP position with fifteen years of experience consulting for companies including America Online, Ford and Kraft was eliminated as a candidate specifically because of his Klout score, which at the time was 34, in favour of a candidate with a score of 67.[22]



Klout | Be Known For W
https://klout.com/home
KLOUT
Brand or Agency? Learn more.
Sign in
Be known for
what you love.
Facebook
Twitter
Email
The best way to have an impact online is to create and share great content.
Klout helps you do exactly that.

Klout | Developer Portal

https://klout.com/s/developers/home

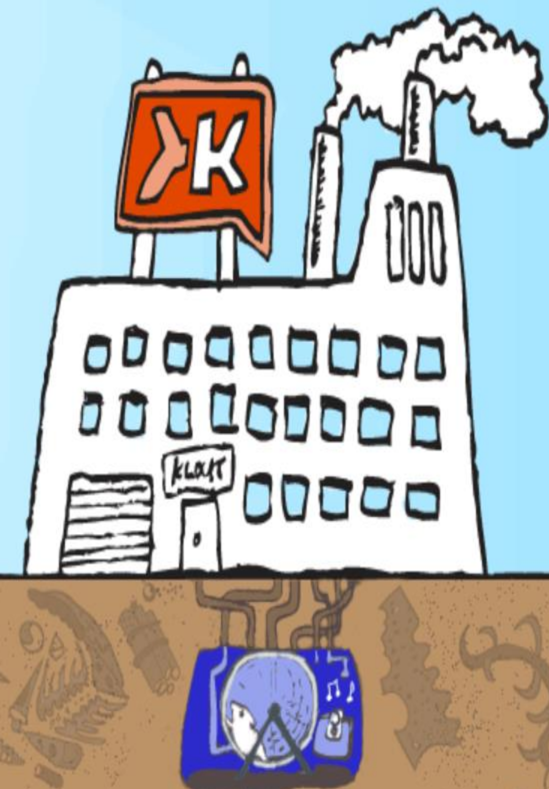
KLOUT DEVELOPERS

Don't have an account? Sign up here. Sign in

Harness the Power of Klout

Take the next step with our API

Register to build your awesome app



Documentation

Enterprise Data

@KloutAPI Updates

Klout

developer.klout.com/member/register

DEVELOPER PORTAL

Register

Sign In   Register

- INTERACTIVE CONSOLE
- TERMS OF SERVICE
- CODE LIBRARY
- DOCS
- - VERSION 2
- REGISTER AN APP

Register a new Mashery ID to access developer.klout.com

Username

Display Name

This is the name which other users will see

Email

A validation E-mail will be sent to this address. Validation is required to complete registration.

Confirm E-mail

Please re-enter your e-mail address.

Password Requirements

- At least one letter
- At least one number
- At least eight characters

Documentation

Interactive Console
Terms of Service
Code Library
Style Guide
Docs
- Version 2
- KloutPass
Register An App

The best documentation is the Interactive Console

Partner API v2
KloutPass
Style Guide
Terms of Service

V2

Version 2 of the API represents a distinct departure from the previous API. We are regularly adding new features and strive to make no breaking changes. Those hamsters are busy at work.

In order to support oft-requested features, like lookup by multiple networks, and to improve the performance at scale, we've abstracted an identity lookup service. Details are below.

- All calls to the API now require a **kloutId**.
- Use the **identity.json** resource to translate a twitter_screen_name or id to a **kloutId**
- Lookup of multiple users simultaneously has been removed. Rate limits will compensate.
- XML support has been removed.

More information...

Documentation: v2

Interactive Console

Terms of Service

Code Library

Style Guide

Docs

- Version 2

- KloutPass

Register An App

Version 2 of the API represents a distinct departure from the previous API. We are regularly adding new features and strive to make no breaking changes. Those hamsters are busy at work.

In order to support oft-requested features, like lookup by multiple networks, and to improve the performance at scale, we've abstracted an identity lookup service. Details are below.

You may also be interested in our OAuth 2.0 API.

An Introduction to Calling the new Klout API

Be sure to always include your API Key in the query string for requests to the API. (key=)

All calls to the Klout API now require a unique *kloutId*. To facilitate this, you must first translate a

PACKAGE INDEX
Browse packages
Package submission
List trove classifiers
List packages
RSS (latest 40 updates)
RSS (newest 40 packages)
Python 3 Packages
PyPI Tutorial
PyPI Security
PyPI Support
PyPI Bug Reports
PyPI Discussion
PyPI Developer Info
ABOUT
NEWS
DOCUMENTATION
DOWNLOAD
COMMUNITY
FOUNDATION
CORE DEVELOPMENT
LINKS

Klout 0.1.3

Minimalist Klout API interface.

Download
Klout-0.1.3.zip

Package Documentation

Not Logged In
Login
Register
Lost Login?
Use OpenID

Status
Nothing to report

Klout API

build passing

A minimalist klout API interface. Use of this API requires klout *developer key*. You can get registered and get a key at

> <http://klout.com/s/developers/v2>

Complete documentation is available at:

> <https://klout.readthedocs.org/en/latest/>

Design Philosoph

See Design Philosophy

Quickstart

Install the PyPi package:

```
pip install Klout
```

This short example shows how to get a kloutId first and fetch user's score using that kloutId:

Klout: An Example

```
from klout import *
import json

# Create the Klout object
k = Klout('**********')

# Get kloutId of the user by inputting a twitter screenName
kloutId = k.identity.klout(screenName="katyperry").get('id')
print "User's klout id is: %s" % (kloutId)

# Get klout score of the user
score = k.user.score(kloutId=kloutId).get('score')
print "User's klout score is: %s" % (score)
```

Klout: An Example (cont.)

By default all communication is not secure (HTTP).
An optional secure parameter can be specified for secure (HTTPS)
communication
k = Klout('sjgykfxs6tqfp33nm9j9kv9t', secure=True)

Optionally a timeout parameter (in seconds) can also be sent
with all calls
score = k.user.score(kloutId=kloutId, timeout=5).get('score')
print "User's klout score is: %s (secure)" % (score)

Get influencers/influencees
inf = k.user.influence(kloutId=kloutId)
print json.dumps(inf, indent=1)

Social Listening

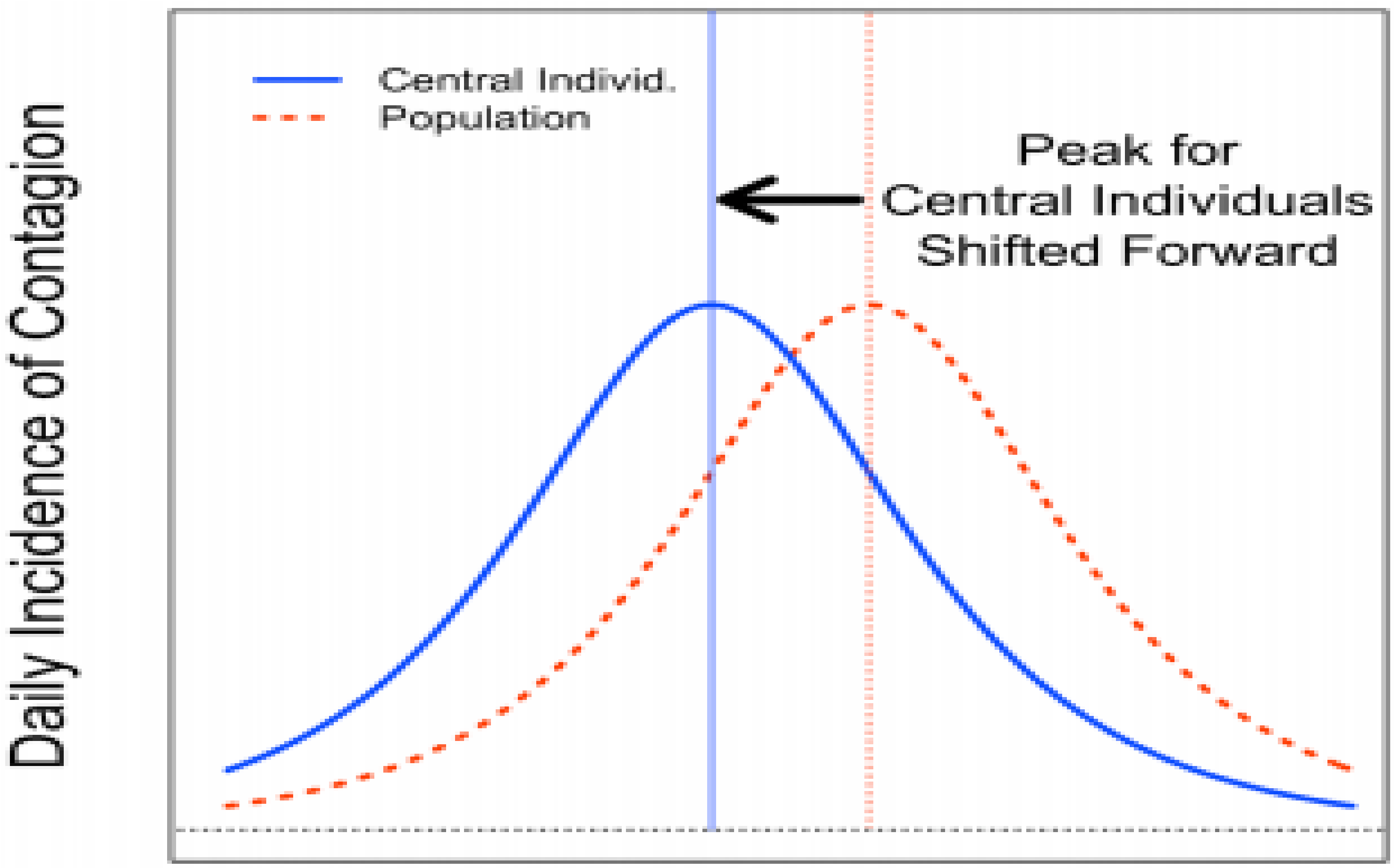
Early Warning Social Listening

- Epidemics are difficult to control because by the time you know you have one, it's usually too late.
  - The infection has taken hold in a sizable proportion of the population and is spreading exponentially.
  - The time for crises prevention has passed and you move straight into full-fledged crises management.
- Wouldn't it be great if we could find a way get a head start?
  - That's a question tackled by Christakis and Fowler, leaders in the field of social network science:
    - In a study of 744 Harvard undergraduates, they were able to isolate a **network central group** that predicted the outbreak of the H1N1 influenza virus on campus in 2009.

Connected: The Surprising Power of Our Social Networks and How They Shape Our Lives -- How Your Friends' Friends' Friends Affect Everything You Feel, Think, and Do, Christakis and Fowler, 2011

Early Warning Social Listening

❖They found that they could gain a full two weeks.



Early Warning Social Listening

- If we adapt the same approach to **social listening**, we could get a similar head start on outbreaks of sentiment.
  - Some important PR crises could be averted and important opportunities could be uncovered and acted upon earlier.
- We could also adapt the same approach to reducing information overload, which is also one of the biggest obstacles to effective social listening.
  - We should focus on network central group, which would help us cut down on the information we have to gather.