Centrality Measures.

Goal: Identify the most important or central nodes in a graph by assigning each vertex v a centrality $c(v)$.

For most measures we want $0 \leq c(v) \leq 1$. For some measures we ~~have~~ also have $\sum_{v \in V(G)} c(v) = 1$, in which case we can interpret $c: V(G) \to [0,1]$ as a probability measure over $V(G)$.

Sometimes we can avoid computing $c(v)$ for every node v:

- If we only want the largest K values (most central nodes).
- If we don't care about nodes with $c(v) < \varepsilon$ (least central nodes).

Degree Centrality

For undirected G ~~and~~ $v \in V(G)$, the degree-centrality of v is $\frac{\deg_G(v)}{n-1} = c_d(v)$.

In matrix language:

$$c_d = \frac{1}{n-1} \mathbb{1} \cdot A = \left[\frac{\deg(v_1)}{n-1}, \frac{\deg(v_2)}{n-1}, \dots, \frac{\deg(v_n)}{n-1} \right].$$

For directed graphs, there are three options

$$c_d^{in}(i) = \frac{1}{n-1} \cdot \sum_{k=1}^{n} a(k,i) = \frac{1}{n-1} \cdot \deg_G^-(i) \qquad c_d^{in} = \mathbb{1} \cdot A^T$$

$$c_d^{out}(i) = \frac{1}{n-1} \sum_{k=1}^{n} a(i,k) = \frac{1}{n-1} \cdot \deg_G^+(i) \qquad c_d^{out} = \mathbb{1} \cdot A$$

$$c_d^{tot}(i) = \frac{1}{2} \cdot \left(c_d^{in}(i) + c_d^{out}(i) \right) = \frac{1}{2(n-1)} \left(\deg_G^-(i) + \deg_G^+(i) \right).$$

$$c_d^{tot} = \frac{1}{2}\left(c_d^{in} + c_d^{out} \right).$$

- c_d^{in} is useful for Twitter-like networks, where an edge ij ~~represent~~ means user i follows user j. Then $c_d^{in}(i)$ = # followers of user i.

Eigenvector Centrality

For an undirected graph G, we might like the centrality of v to be the (scaled) ~~average~~ sum of the centrality of v's neighbours

$$c(v) = \frac{1}{\lambda} \cdot \sum_{w \in N_G(v)} c(w) \qquad (*).$$

where λ is some scaling factor used to make sure that $0 \leq c(v) \leq 1$.

Rewrite $(*)$ to get $\displaystyle\sum_{w \in N(v)} c(w) - \lambda c(v) = 0. \qquad (**)$.

We want ~~thi~~ $(**)$ to hold for every $v \in V(G)$, which gives the matrix equation

$$(A - \lambda I) \cdot \mathbf{c} = 0$$

This might look familiar:

- λ is called an <u>eigenvalue</u> of A
- $\mathbf{c}$ is the corresponding <u>eigenvector</u> of A

Not all eigenvalues/eigenvector pairs are interesting.

Example: $c = (0, 0, \ldots, 0)$ solves this for any $\lambda \in \mathbb{R}$.

We also never want c to have negative values, since these correspond to negative centralities.

To get a non-trivial vector c, we need to choose λ so that

$$p(\lambda) \overset{\text{def}}{=} \det(A - \lambda I) = 0$$

- $p(\lambda)$ is degree-n polynomial in the variable λ, so it has n (complex, possibly) solutions (Fundamental Theorem of Algebra)
- Since A is real and symmetric all solutions have $\lambda \in \mathbb{R}$, i.e. the eigenvalues $\lambda_1, \ldots, \lambda_n$ are all real.

- Perron Frobenius Theorem: If A is real, symmetric, non-negative and irreducible*, then A has an eigenvalue λ_1 with

$$\lambda_1 \geq \max\{|\lambda_i| : i \in \{1 \ldots n\}\}$$

and the solution c to $(A - \lambda_1 I) \cdot c = 0$ has only non-negative values.

*A is irreducible as long as G is connected.

Eigenvalue Centality (for connected undirected G)

Let λ_1 be the largest eigenvalue of adjacency matrix A, and let $\mathbf{c}$ be the solution to

$$(A - \lambda_1 I)\cdot \mathbf{c} = 0$$

(so $\mathbf{c}$ is an n-vector indexed by $V(G)$). Then the eigenvector centality of $v \in V(G)$ is

$$C_e(v) \stackrel{\text{def}}{=} \frac{c(v)}{\max\{c(w) : w \in V(G)\}}$$

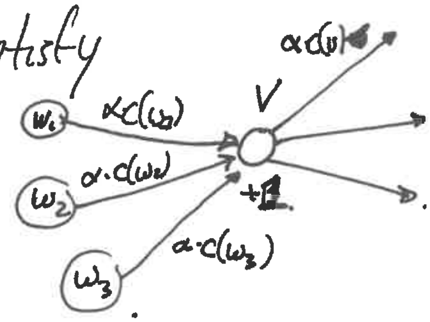
Generalization to directed graphs is problematic:

Two versions: $C_e^+(v) = \frac{1}{\lambda} \sum_{\underbrace{w \in N^+(v)}_{\text{out-neighbours}}} c(w)$ $\qquad$ $C_e^-(v) = \frac{1}{\lambda} \sum_{\underbrace{w \in N^-(v)}_{\text{in-neighbours.}}} c(w)$

If we choose $C_e^-(v)$ for something like Twitter, then users with no followers don't count at all!

Katz Centrality

Addresses main problem with (directed) eigenvector centrality. Every ~~user~~ vertex v gets a small centrality β "for free", even if $\deg^-(v) = 0$. Then we want centrality to satisfy



$$c(v) = \alpha \cdot \sum_{w \in V(G)} c(w) \cdot a(w,v) + \beta.$$

We will eventually normalize everything, so we can set $\beta = 1$, then we get the matrix equation.

$$c_\alpha = (I - \alpha A^T)^{-1} \mathbb{1} \qquad \text{[explain].}$$

We choose $\alpha < 1$, and need $\alpha < 1/\lambda$ where λ is maximum eigenvalue of A.

Amazing (?) fact:

$$c_\alpha(v) = \overbrace{\sum_{k=0}^{\infty} \underbrace{\sum_{w \in V(G)} \alpha^k \cdot \underbrace{A^k(w,v)}_{\text{\# K-edge paths from } w \text{ to } v.}}_{\text{\# K-edge paths that end at } v \text{ (each path is worth } \alpha^k)}}^{\text{each K-edge path is worth } \alpha^K.}$$

$\alpha < 1/\lambda$ is needed to avoid $c_\alpha(v) = \infty$

PageRank Centrality (Google).

Paige & Brin's Billion Dollar Idea.

Want to satisfy $c(v) = \alpha \cdot \sum_{w \in V(G)} c(w) \cdot \hat{a}(w,v) + \beta$.

where $\hat{a}(w,v) = \begin{cases} 1/\deg^+(w) & \text{if } \deg^+(w) > 0. \\ \frac{1}{n} & \text{if } \deg^+(w) = 0. \end{cases}$

<u>Intuition</u>: Each node is a webpage. Edge wv represents link on page w to page v.

- User chooses random link ~~to~~ follow on page w (each link is chosen with probability $\frac{1}{\deg^+(w)}$.)
- If w has no links, user chooses uniformly random page (from the entire WWW).

Then $c(v)$ is the probability that a user is looking at page v, at a particular point in time.
(called the stationary probability of v.)

Each row of $\hat{A}$ is a probability distribution over $V(G)$.

$$\begin{array}{l} a(i,1)\,a(i,2)\,a(i,3)\ \cdots\cdots\ a(i,n) \\ \frac{1}{d}\,0\,0\,0\,\frac{1}{d}\,0\,0\,0\,\frac{1}{d}\,0\,0\,\frac{1}{d}\cdots\frac{1}{d}. \\ \text{OR.} \\ \frac{1}{n}\,\frac{1}{n}\,\frac{1}{n}\cdots\cdots\cdots 1/n. \end{array}$$

If we are standing at vertex i, then the probability that we move from i to j is $\hat{a}(i,j)$.

Also $c \stackrel{def}{=} c(1) \dots c(n)$ is a probability distribution over $V(G)$.

$$c \cdot \hat{A}^T = c$$

Interpretation: If we are dropped at vertex i with probability $c(i)$ then take a random walk for any number k of steps, the probability that we finish at vertex j is $c(j)$.