MODELLI 27/10/26



Graph Lapl. $L_W = B^T C B$

Thm $\ker(L_W) = \text{span}\{1\!\!1_{V_1}, \dots, 1\!\!1_{V_q}\}$
V_i = connect. comp

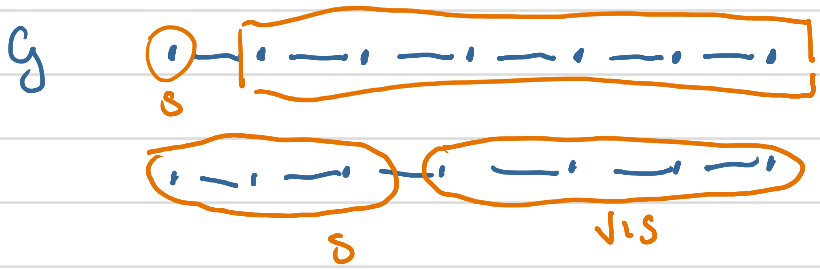
$$\ker(L_W) = \text{span}\left\{\begin{bmatrix}1\\1\\\vdots\\1\end{bmatrix}, \begin{bmatrix}0\\0\\\vdots\\1\end{bmatrix}\right\} = \text{span}\left\{\begin{bmatrix}1\\\vdots\\1\end{bmatrix}, \begin{bmatrix}1\\1\\1\\-1\\-1\end{bmatrix}\right\}$$

guaranteed — sign identify the connect. comp.

S a set of nodes ; we partition $G = (V, E)$ into S and $V \setminus S$

$$w(\partial S) = \sum_{\substack{i \in S \\ j \in V \setminus S}} w_{ij}$$ (w_{ij}: weight matrix)

Minimize $w(\partial S)$ ⇝ find a good cut



$w(\partial S) = 1$

$w(\partial S) = 1$

DEF The ratio-cut for S is $RC(S) = w(\partial S)\left(\frac{1}{|S|} + \frac{1}{|V \setminus S|}\right)$

↳ force a balanced partition

Task Find a good cut ⟷ minimize $RC(S)$

When we have a partition $S \sqcup (V \setminus S)$ $\begin{cases} |S| = a \\ |V \setminus S| = n - a = b \end{cases}$

we can def $(z_S)_i = \begin{cases} \sqrt{b/na} & i \in S \\ -\sqrt{a/nb} & i \notin S \end{cases}$

CHECK $\|z_S\|_2 = 1$

$z_S^T L_\omega z_S = z_S B^T C B z_S$ $\quad B = \boxed{\#-1}$ $\quad (Bz_S)_e = \begin{cases} 0 & e \text{ in } S \text{ or } V\setminus S \\ \neq 0 & e \text{ connects } S \text{ with } V\setminus S \end{cases}$

Up to choosing the signs: $\sqrt{\frac{b}{na}} + \sqrt{\frac{a}{nb}}$

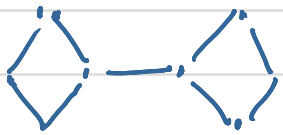
By a direct computation $z_S^T L_\omega z_S = w(\partial S)\frac{n}{ab} = Rc(S)$ $\quad \left(\frac{1}{a} + \frac{1}{n-a}\right)$

We can minimize the Rayleigh quotient:

$\min_{z \in \mathbb{C}^n,\ z^T \mathbb{1} = 0,\ \|z\|_2 = 1} z^T L_\omega z = \lambda_2$ $\quad Sp(L_\omega) = \{\lambda_1, \lambda_2, \dots, \lambda_n\}$, $\lambda_1 = 0$, $v_1 = \mathbb{1}$

↳ Courant-Fisher (minimax thm)

$S = \{ i \mid z_i > 0 \}$ $\quad V\setminus S = \{ \dots \}$ $\quad z$ eigv of λ_2 — Fiedler vector



= Centrality measures ⇝ rank the nodes (or the edges) in a graph

↳ DEF $C(i)$ = "importance of node i (edge)"



Properties:
-) $C(i) \geq 0$
-) Invariant under permutation

Practical note: we will rely on NetworkX in Python → many of these measures are available!

DEGREE CENTRALITY $\quad C_D(i) = \frac{d_i}{n-1}$ $\quad d_i = \#$ edges connected to i

For directed graphs: we can count incoming or outgoing edges
Limitation: very local measure

CLOSENESS CENTRALITY Rank a node higher if it's close to everybody else



$d(i,j)$ = dist from i to j = length of the shortest path

$$C_C(i) = \frac{n-1}{\sum_{j \neq i} d(i,j)}$$

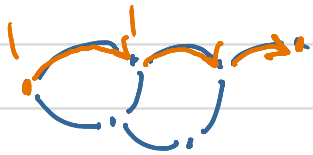
↝ More expensive
↝ Prone to change quickly by node addition or removal

HARMONIC CENTRALITY

$$C_H(i) = \sum_{j \neq i} \frac{1}{d(i,j)}$$

↝ Similar to closeness, but also usable for disconnected graphs
↝ Same comp. cost

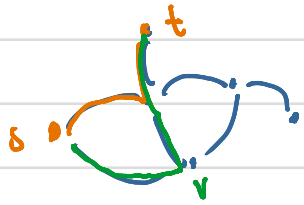
Note



Do not need to compute all pairwise shortest paths. You can find it in NetworkX ~ idea based on Dijkstra for all shortest paths at once

BETWEENNESS CENTRALITY Nodes are important if they belong to shortest paths...

σ_{st} = length of the shortest path from s to t = $d(s,t)$
$\sigma_{st}(v)$ = length of the shortest path from s to v that goes through v



$\sigma_{st} = 2$
$\sigma_{st}(v) = 3$

$$C_B(i) = \sum_{\substack{s,t \in V \\ s \neq t \\ s \neq i \\ t \neq i}} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

We can give the same def for edges, with $\sigma_{st}(e)$ = length ... that goes through an edge e

Length-based measures can be more expressive (less local!) but they are also expensive

EIGENVECTOR CENTRALITIES

Idea: a node distributes the importance to the neighbors

$x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$ x_i = ranking of node i

In the undirected case $x = \lambda A x$

If we have W (weighted graph) $\rightsquigarrow$ $x = \lambda W x$ (ranking is distributed proportionally to w_i)

$\rightsquigarrow$ λ is an eigenvalue: which one do we pick?
→ Choose λ "as large as possible"

A, W are non-negative $\Rightarrow$ Perron-Frobenius guarantees $\lambda = \rho(A)$
If the graph is connected $\rightsquigarrow$ $\lambda = \rho(A)$ is the unique eigenvalue with $|\lambda| = \rho(A)$ $\quad x \geq 0$

POWER METHOD is a good choice for computing x

$$\begin{cases} x^{(0)} = \dots \\ x^{(k+1)} = \dfrac{Ax^{(k)}}{\|Ax^{(k)}\|_\infty} \end{cases}$$

$x^{(k)} \to x$ eigenvector for the dominant eigenvalue with speed $O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^k\right)$

Minor modification: assign a baseline rank β

$x = \kappa A x + \beta \mathbb{1}$ $\rightsquigarrow$ $(I - \kappa A)x = \beta \cdot \mathbb{1}$

$\beta \mathbb{1}$ ▷ baseline
$\kappa A x$ ▷ neighbor distribution

$x = (I - \kappa A)^{-1} \mathbb{1}$ ($\beta = 1$ for simplicity)

If we pick $0 < \kappa < \rho(A)^{-1}$ $\quad x = (I - \kappa A)^{-1} \mathbb{1}$

$$\stackrel{!}{=} \sum_{j \geq 0} \kappa^j A^j \mathbb{1}$$

$(A^j)_{st}$ = # walks from s to t of length j

$(A^j \mathbb{1})_s$ = # walks of length j starting from s

Interpretation: we are counting outgoing paths, penalizing the long ones

Depending on the choice of α:

α small $\quad x \cong (I + \alpha A + \dots)\mathbb{1} = \mathbb{1} + \alpha A \mathbb{1}$

$A\mathbb{1}$ ▷ degrees

$\mathbb{1}$ ▷ constant

$\alpha \to \frac{1}{\rho(A)}$ $\quad x \cong (I - \alpha A)^{-1}\mathbb{1} \cong$ proportional to the eigv. centrality mentioned above

$(I - \alpha A)^{-1}$ ▷ almost singular, because α close to λ_1

$$D = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix} - \alpha I = \begin{bmatrix} \lambda_1 - \alpha & & \\ & \lambda_2 - \alpha & \\ & & \ddots \\ & & & \lambda_n - \alpha \end{bmatrix} \sim \begin{bmatrix} \frac{1}{\lambda_1 - \alpha} & & & \\ & \frac{1}{\lambda_2 - \alpha} & & \\ & & \ddots & \\ & & & \frac{1}{\lambda_n - \alpha} \end{bmatrix}^{-1}$$

$\cong v_1 v_1^T$

SUBGRAPHS CENTRALITIES