Exam III syllabus

→ K-means | EM clustering
→ hierarchical clustering
→ density-based
→ spectral / graph
→ Evaluation of clustering

graph analysis
→ graph mining
→ graph data (chap 4)
→ graph kernels (chap 5, sec. 5.4.2)

Centrality : $\underbrace{C(v) \to \mathbb{R}}_{\text{function to rank nodes}}$

degree centrality

$d(v_i)$ = degree

Eccentricity centrality

$\frac{1}{e(v_i)}$

$e(v_i) = \max_{v_j} \{ d(v_i, v_j) \}$

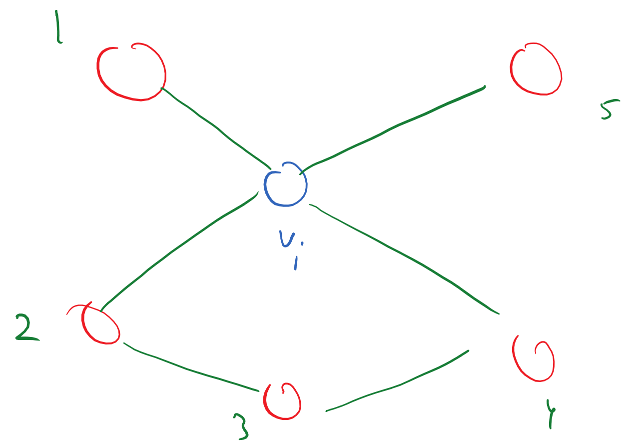
eccentricity

"effective"

closeness centrality

$$cc(v_i) = \frac{1}{\sum_{v_j} d(v_i, v_j)}$$

Betweenness centrality



$$\gamma_{jk}(v_i) = \frac{\eta_{jk}(v_i)}{\eta_{jk}}$$

$$= \frac{\text{\# of shortest paths between } j \text{ \& } k \text{ that go through } v_i}{\text{\# of SPs between } j \text{ \& } k}$$

$$\gamma_{24}(v_i) = \frac{1}{2} \qquad \gamma_{15}(v_i) = \frac{1}{1} = 1$$

$$b(v_i) = \sum_{j} \sum_{k>j} \gamma_{jk}(v_i)$$

$j \neq i$

$k \neq i$

distance

= "shortest" distance between two entities

→ Euclidean

Geodesic distance

for graphs

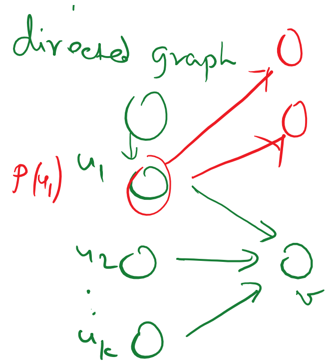
→ shortest path length

Prestige

$G \rightarrow A$

adjacency matrix

directed graph



$$p(v) = \sum_{u_i} \underline{\underline{A(u_i, v)}} \cdot p(u_i)$$

prestige of v

$$\boxed{\underbrace{\vec{p}}_{\text{prestige over all nodes}} = A^T \vec{p}}$$

$$\vec{p}_0 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}$$

$$\underbrace{\vec{p}_{i+1} = A^T \vec{p}_i}_{\text{Converge to dominant eigenvector}}$$

iterative method until convergence

Google PageRank

1) <u>authority</u> $\equiv$ normalized prestige

Prestige	Authority
A	$M = \Delta^{-1} A$
adjacency	normalized A

out degree

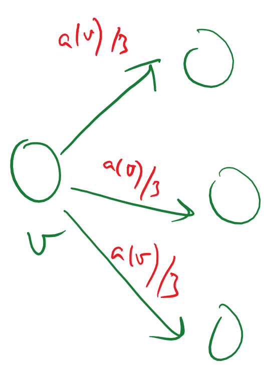
$$\Delta = \text{degree matrix} = \begin{pmatrix} d_1 & & & \\ & d_2 & & \\ & & \ddots & \\ & & & d_n \end{pmatrix}$$

$$\Delta^{-1} = \begin{pmatrix} 1/d_1 & & \\ & 1/d_2 & \\ & & \ddots \end{pmatrix}$$

adjacency | normalized A

$D^{-1} = \begin{pmatrix} 1/d_1 & & & \\ & 1/d_2 & & \\ & & \ddots & \\ & & & 1/d_n \end{pmatrix}$

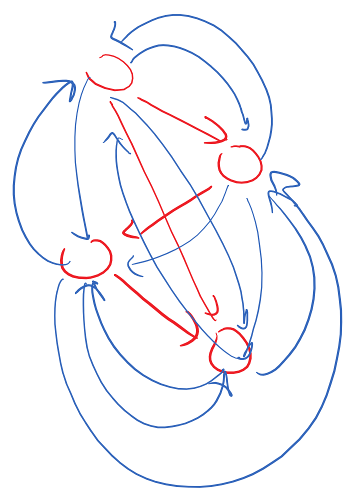
$\underline{\underline{a(v)}}$



$$\boxed{\vec{a} = M^T \vec{a}}$$

+

G



$|G| = n$ vertices

random jumps

$$N_r = \left(\frac{1}{n} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & \cdots & 1 \\ \vdots & & & & \vdots \\ 1 & 1 & 1 & \cdots & 1 \end{bmatrix} \right)$$

random

$n \times n$ matrix

$$\boxed{M' = \underline{\underline{\alpha N_r}} + \underline{(1-\alpha) M}}$$

$\alpha = 0.1$ or $\alpha = 0.05$

$$\boxed{\vec{p} = (M')^T \vec{p}}$$

Pagerank is the dominant eigenvector of $(M')^T$

---

authorities & hubs (HITS)

authorities & hubs (HITS)

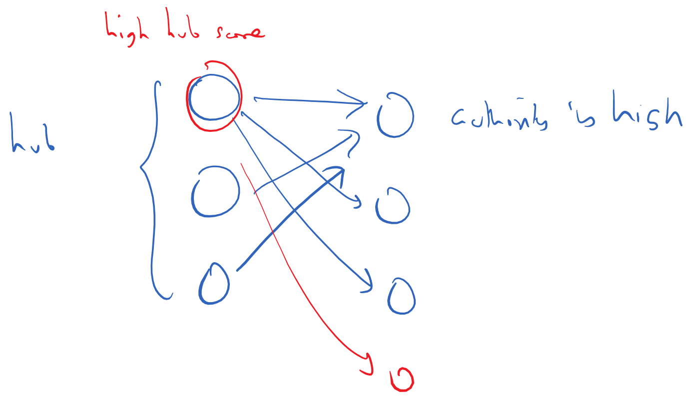
authority : $a(v)$

hub : $h(v)$

v

hub score: how many high authority pages/nodes does v link to

authority score: how many high "hub" node point to v



$$\boxed{\begin{aligned}\vec{a} &= A^T \vec{h} \\ \vec{h} &= A\vec{a}\end{aligned}}$$

$$\vec{a} = A^T(A\vec{a}) = (A^TA)\vec{a}$$

$$\boxed{\vec{a} = (A^TA)\vec{a}}$$

$$\vec{h} = A\vec{a} = A(A^T\vec{h})$$

$$\boxed{\vec{h} = (AA^T)\vec{h}}$$

A^TA

if you do SVD of A

of you do SVD of A

Going to give you $\vec{a}$ & $\vec{h}$ vectors

$\vec{a}$: right singular vector

$\vec{h}$: left singular vector

Kernel function between graph nodes

$K(u,v)$ = similarity between u & v

G is undirected

→ A is symmetric

adjacency matrix

$A = \{ a_{ij} \geq 0 \}$

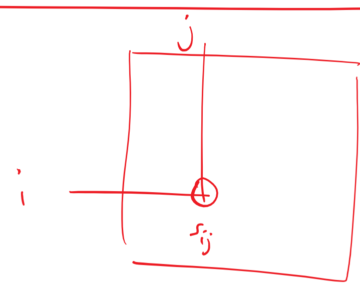
weight on edge (i,j)

→ $L = \Delta - A$

Laplacian

Positive semi-definite

S : base similarity matrix



S^2 = 2-step similarity

S^l = l-step similarity

① Power-kernel

l

Power-kernel

??? $\boxed{K(u,v) = S^{l}(u,v)}$ ← Similarity after a l-length walk.

This has to be PSD ← positive semi-definite

$$A = ULU^T$$

$$= \underbrace{\begin{pmatrix} | & | & & | \\ u_1 & u_2 & \cdots & u_n \\ | & | & & | \end{pmatrix}}_{U} \underbrace{\begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{pmatrix}}_{L} \underbrace{\begin{pmatrix} - & u_1^T & - \\ - & u_2^T & - \\ & \vdots & \\ - & u_n^T & - \end{pmatrix}}_{U^T}$$

eigen vectors   eigenvalue

$$\lambda_1 \geq \lambda_2 \ldots \geq \lambda_n$$

$$S^{l} = A^{l} = U \underbrace{\begin{pmatrix} \lambda_1^{l} & & & \\ & \lambda_2^{l} & & \\ & & \ddots & \\ & & & \lambda_n^{l} \end{pmatrix}}_{} U^T$$

A can have $-$ve λ_i

all even-length walks (l is even) $\underbrace{\text{lead to } K}_{\text{guaranteed to be PSD}}$

(1) $S = A$

→ even powers are kernels

$$K = S^2, \quad K = S^4, \ldots$$

$K = S^2, \quad K = S^4, \ldots$

$S = L = \Delta - A$

$\rightarrow$ all l values lead to PSD K.

Von Neumann Diffusion kernel

$$K = \sum_{l=0}^{\infty} \beta^l S^l$$

$S^0 = I$

$\beta \geq 0$ is some $\underbrace{\text{constant}}_{\text{damping}}$.

$K(u,v)$ = influence / diffusion over walks of all lengths l

$$K = I + \beta S + \beta^2 S^2 + \beta^3 S^3 + \ldots$$

$$K = I + \beta S \left(\underbrace{I + \beta S + \beta^2 S^2 + \ldots}_{} \right)$$

$$K = I + \beta S K$$

$$(I - \beta S) K = I$$

$$\boxed{K = (I - \beta S)^{-1}}$$

$$\boxed{K = (I - \beta S)^{-1}}$$

for K to be a kernel (PSD), we have to restrict β.

$$\boxed{\beta < 1/\max\{|\lambda_i|\}}$$

Exponential Diffusion kernel

$$K = \underbrace{\sum_{l=0}^{\infty} \left(\frac{\beta^l}{l!}\right) S^l}$$

$$\boxed{K = \exp(\beta S)}$$

matrix exponential

$$e^x = \underbrace{1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots}$$

exponential

$$e^{\beta x} = 1 + \beta x + \frac{(\beta x)^2}{2!} + \frac{(\beta x)^3}{3!} + \cdots$$

$$S = U L U^T$$

eigenvectors eigenvalues

$$\exp(S) = U \begin{pmatrix} e^{\lambda_1} & & 0 \\ & e^{\lambda_2} & \\ & & \end{pmatrix} U^T$$

$$\exp(S) = U \begin{pmatrix} e^{\lambda_1} & & & 0 \\ & e^{\lambda_2} & & \\ & & \ddots & \\ 0 & & & e^{\lambda_n} \end{pmatrix} U^T$$

$$K = \boxed{\exp(\beta S) = U \begin{pmatrix} e^{\beta\lambda_1} & & & 0 \\ & e^{\beta\lambda_2} & & \\ & & \ddots & \\ 0 & & & e^{\beta\lambda_n} \end{pmatrix} U^T}$$

$\underbrace{\text{well defined}}_{\text{will lead to a PSD kernel}}$ for all β!

$\beta \geq 0$ ← Preferred.

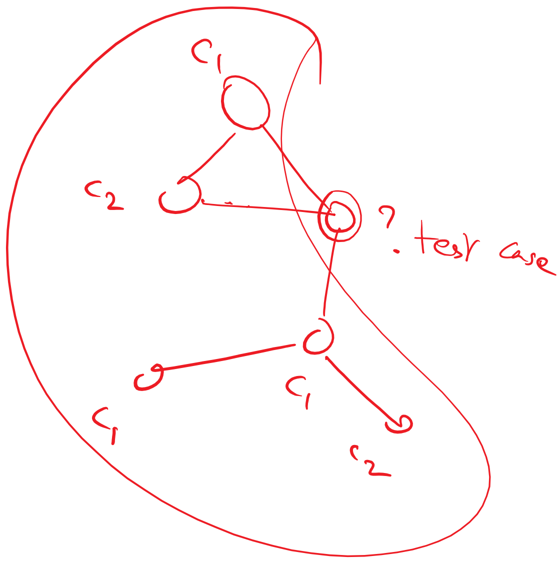
$$G \longrightarrow S \longrightarrow K$$

base matrix

kernel matrix

Application:

1) Graph / node classification



e.g. use SVM
via K

2) graph clustering

→ Kernel K-means

→ Spectral