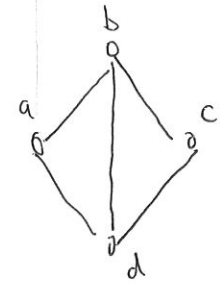
[Leskovec et al., Chapter 10.2, 10.4]

Betweenness Centrality : Measurement how much importance a link is

- All pairs of nodes have 1 communication.
- They will use a shortest path for a communication.
- If there are more than one shortest path, they will choose the communication path randomly.

Ex



communication between a → c

2 shortest paths

a → b → c → use with prob 0.5

a → d → c → use with prob 0.5

- Count expected number of communications on each link.

Ex expected #communications on link {a,b}

- from a → c → prob. 0.5
- from a → b → prob. 1
- from a → d → prob. 0
- from b → a → prob. 1
- from b → c → prob 0
- from b → d → prob. 0
- from c → a → prob. 0.5
- from c → b → prob. 0
- from c → d → prob. 0
- from d → a → prob. 0
- from d → b → prob. 0
- from d → c → prob. 0

Expected # communications

= betweenness of {a,b}

= 3

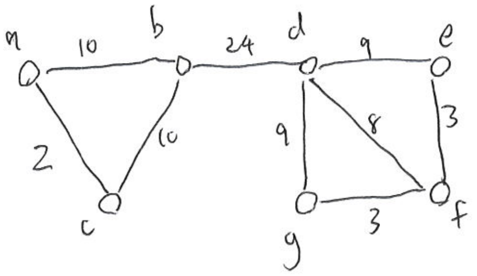
Betweenness of $\{a,d\}$, $\{b,c\}$, $\{c,d\}$ is 3.

Betweenness of $\{b,d\}$ is 2.

Formulation

$$\text{Betweenness of } \{u,v\} = \sum_{v_1, v_2 \in V} \frac{\#\text{ shortest paths using } \{u,v\} \text{ from } v_1 \to v_2}{\#\text{ shortest paths from } v_1 \to v_2}$$

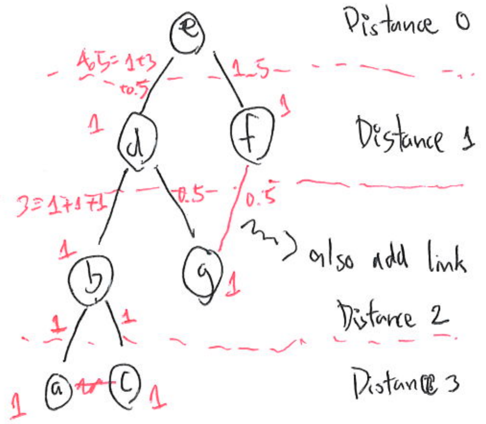
Exercise



Algorithms for calculating betweenness centrality

Technique : Breathe First Search (BFS)

Use for calculating # shortest paths from ⓔ to all other nodes.



Distance 0

Distance 1

also add link between distance i and i+1

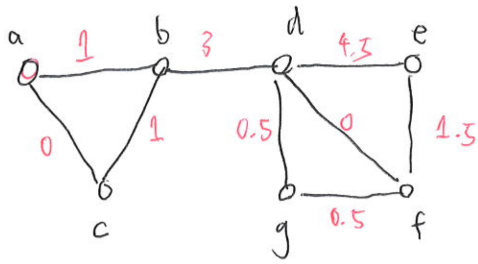
Distance 2

Distance 3

Question

How to calculate # shortest paths from ⓔ to other nodes?

BFS Tree



- Calculate the graph like this, but begin from all other nodes.
- Add the results together.

Algorithm

For all node i : $O(\|V\|)$

Perform a BFS to construct a BFS tree from i $O(\|E\|)$

Consider the BFS Bottom-up, Suppose that we are considering j at level d.

$$cen_{i,j} = \frac{\boxed{\sum_{j'\text{: incident edge at level } d+1} [cen_{i,j+1}] + 1}}{\#\text{ incident edges at level } d.}$$

Same for the edge incident to the same node
- Can be pre-calculated at nodes!

$O(\|E\|)$

$$\text{Betweenness of edge } j = \sum_i cen_{i,j}$$

Computation Time $O(\|V\| \|E\|)$

Graph Clustering Method (Divide a social network into groups)

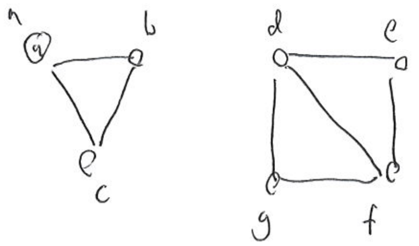
1: Remove the edge with highest betweenness centrality

2: If "nice groups" are obtained, terminated.

Otherwise, go to step 1

3: Nodes that can communicate together (after the removals) will be put into the same group.

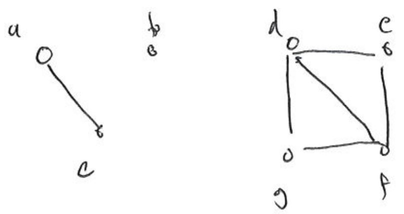
Remove betweenness ≥ 12



Group 1 : {a, b, c}

Group 2 : {d, e, f, g}

Remove betweenness ≥ 5



Group 1 : {a, c}

Group 2 : {b}

Group 3 : {d, e, f, g}

Create groups by removing as least as edges as possible

Partition a social network [k-way clustering]

- Divide a social network into 2 groups with similar sizes.

Laplacian Matrix

$$A = \begin{array}{c|ccccccc} & a & b & c & d & e & f & g \\ \hline a & 2 & -1 & -1 & 0 & 0 & 0 & 0 \\ b & -1 & 3 & -1 & -1 & 0 & 0 & 0 \\ c & -1 & -1 & 2 & 0 & 0 & 0 & 0 \\ d & 0 & -1 & 0 & 4 & -1 & -1 & -1 \\ e & 0 & 0 & 0 & -1 & 2 & -1 & 0 \\ f & 0 & 0 & 0 & -1 & -1 & 3 & -1 \\ g & 0 & 0 & 0 & -1 & 0 & -1 & 2 \end{array}$$

$A = (a_{ij})$ where

$$a_{ij} = \begin{cases} 0 & \text{when } i \neq j \text{ and } \{i,j\} \notin E \\ -1 & \text{when } i \neq j \text{ and } \{i,j\} \in E \\ \text{degree of } i & \text{when } i = j \end{cases}$$

$\Downarrow$ eigenvalue decomposition.

Find vector v and value λ such that $\lambda \cdot v = A \cdot v$

smallest eigenvalue $= 0$

$$\text{eigenvector} = \begin{array}{c} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \end{array} \begin{array}{c} a \\ \vdots \\ g \end{array}$$

$$\underbrace{0}_{\lambda} \cdot \underbrace{\begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}}_{v} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \underbrace{\begin{bmatrix} 2 & -1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & -1 & 0 & 0 & 0 \\ & & & \vdots & & & \end{bmatrix}}_{A} \cdot \underbrace{\begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}}_{V}$$

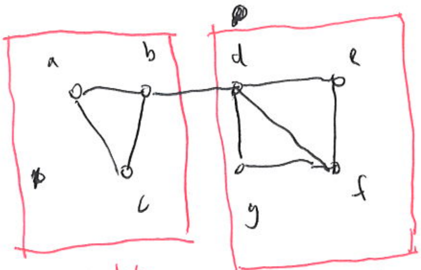
This is true for all Laplacian matrix

Second smallest eigenvalue $= 0,398$

$$\text{eigenvalue} = \begin{array}{c} a \\ b \\ c \\ d \\ e \\ f \\ g \end{array} \begin{bmatrix} -0.493 \\ -0.297 \\ -0.493 \\ 0.214 \\ 0.356 \\ 0.356 \\ 0.356 \end{bmatrix}$$

$a, b, c \rightarrow$ one partition

$d, e, f, g \rightarrow$ another partition



one partition   another partition