Betweenness Centrality

Assumption: important nodes connect other nodes.

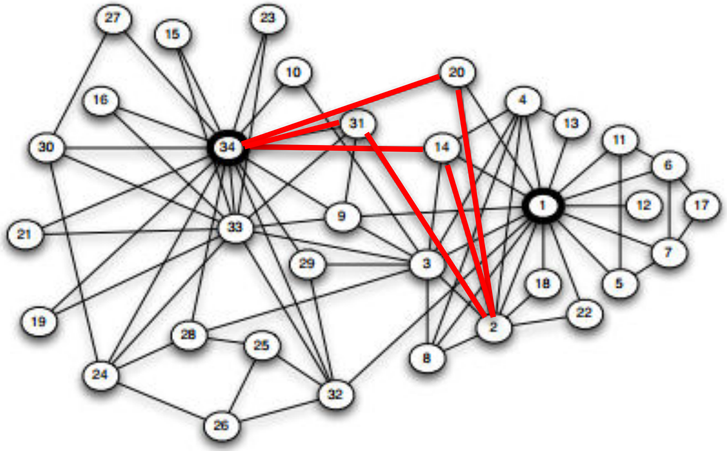
Recall: the distance between two nodes is the length of the shortest path between them.

Ex. The distance between nodes 34 and 2 is 2:

Path 1: 34 – 31 – 2

Path 2: 34 – 14 – 2

Path 3: 34 – 20 – 2



Friendship network in a 34-person karate club
[Zachary 1977]

Nodes 31, 14, and 20 are in a shortest path of between nodes 34 and 2.

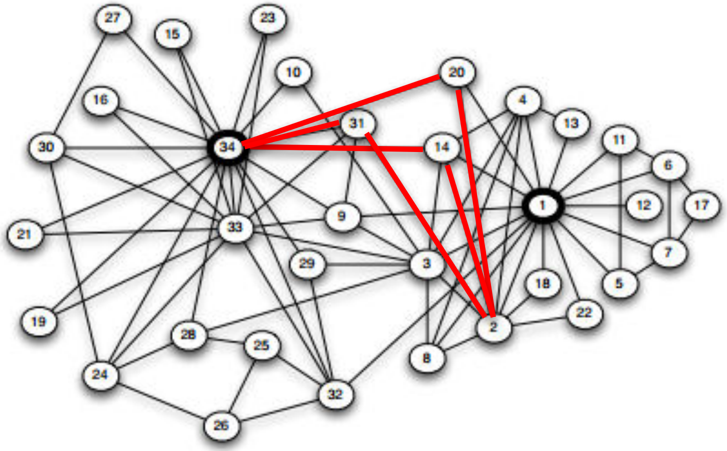
Betweenness Centrality

Assumption: important nodes connect other nodes.

$C_{btw}(v) = \sum_{s,t \in N} \frac{\sigma_{s,t}(v)}{\sigma_{s,t}}$, where

$\sigma_{s,t}$ = the number of shortest paths between nodes s and t.

$\sigma_{s,t}(v)$ = the number shortest paths between nodes s and t *that pass through node* v.

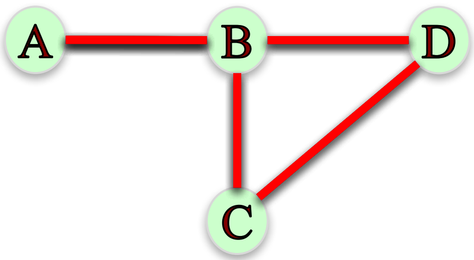


Friendship network in a 34-person karate club
[Zachary 1977]

Betweenness Centrality

Assumption: important nodes connect other nodes.

$$C_{btw}(v) = \sum_{s,t \in N} \frac{\sigma_{s,t}(v)}{\sigma_{s,t}}$$



Endpoints: we can either include or exclude node v as node s and t in the computation of $C_{btw}(v)$.

Ex. If we exclude node v, we have:

$$C_{btw}(B) = \boxed{\frac{\sigma_{A,D}(B)}{\sigma_{A,D}}} + \boxed{\frac{\sigma_{A,C}(B)}{\sigma_{A,C}}} + \boxed{\frac{\sigma_{C,D}(B)}{\sigma_{C,D}}} = \boxed{\frac{1}{1}} + \boxed{\frac{1}{1}} + \boxed{\frac{0}{1}} = 2$$

If we include node v, we have:

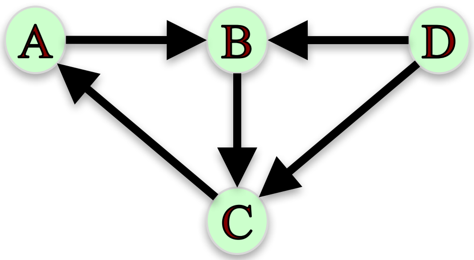
$$C_{btw}(B) = \boxed{\frac{\sigma_{A,B}(B)}{\sigma_{A,B}}} + \boxed{\frac{\sigma_{A,C}(B)}{\sigma_{A,C}}} + \boxed{\frac{\sigma_{A,D}(B)}{\sigma_{A,D}}} + \boxed{\frac{\sigma_{B,C}(B)}{\sigma_{B,C}}} + \boxed{\frac{\sigma_{B,D}(B)}{\sigma_{B,D}}} + \boxed{\frac{\sigma_{C,D}(B)}{\sigma_{C,D}}} = \boxed{\frac{1}{1}} + \boxed{\frac{1}{1}} + \boxed{\frac{1}{1}} + \boxed{\frac{1}{1}} + \boxed{\frac{1}{1}} + \boxed{\frac{0}{1}} = 5$$

Disconnected Nodes

Assumption: important nodes connect other nodes.

$$C_{btw}(v) = \sum_{s,t\in N} \frac{\sigma_{s,t}(v)}{\sigma_{s,t}}$$

What if not all nodes can reach each other?



Node D cannot be reached by any other node.

Hence, $\sigma_{A,D} = 0$, making the above definition undefined.

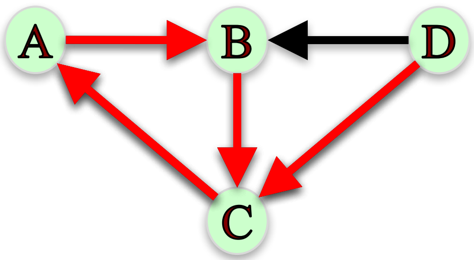
When computing betweenness centrality, we only consider nodes s, t such that there is at least one path between them.

Disconnected Nodes

Assumption: important nodes connect other nodes.

$C_{btw}(v) = \sum_{s,t \in N} \frac{\sigma_{s,t}(v)}{\sigma_{s,t}}$

What if not all nodes can reach each other?



Node D cannot be reached by any other node.

Hence, $\sigma_{A,D} = 0$, making the above definition undefined.

Ex. What is the betweenness centrality of node B, without including it as endpoint?

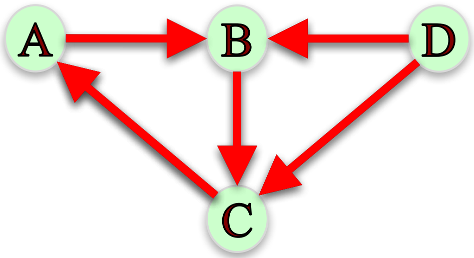
$$C_{btw}(B) = \boxed{\frac{\sigma_{A,C}(B)}{\sigma_{A,C}}} + \boxed{\frac{\sigma_{C,A}(B)}{\sigma_{C,A}}} + \boxed{\frac{\sigma_{D,C}(B)}{\sigma_{D,C}}} + \boxed{\frac{\sigma_{D,A}(B)}{\sigma_{D,A}}} = \boxed{\frac{1}{1}} + \boxed{\frac{0}{1}} + \boxed{\frac{0}{1}} + \boxed{\frac{0}{1}} = 1$$

Disconnected Nodes

Assumption: important nodes connect other nodes.

$$C_{btw}(v) = \sum_{s,t \in N} \frac{\sigma_{s,t}(v)}{\sigma_{s,t}}$$

What if not all nodes can reach each other?



Node D cannot be reached by any other node.

Hence, $\sigma_{A,D} = 0$, making the above definition undefined.

Ex. What is the betweenness centrality of node C, without including it as endpoint?

$$C_{btw}(C) = \boxed{\frac{\sigma_{A,B}(C)}{\sigma_{A,B}}} + \boxed{\frac{\sigma_{B,A}(C)}{\sigma_{B,A}}} + \boxed{\frac{\sigma_{D,B}(C)}{\sigma_{D,B}}} + \boxed{\frac{\sigma_{D,A}(C)}{\sigma_{D,A}}} = \boxed{\frac{0}{1}} + \boxed{\frac{1}{1}} + \boxed{\frac{0}{1}} + \boxed{\frac{1}{1}} = 2$$

Betweenness Centrality – Normalization

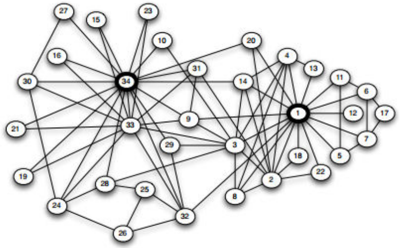
Assumption: important nodes connect other nodes.

$$C_{btw}(v) = \sum_{s,t \in N} \frac{\sigma_{s,t}(v)}{\sigma_{s,t}}$$

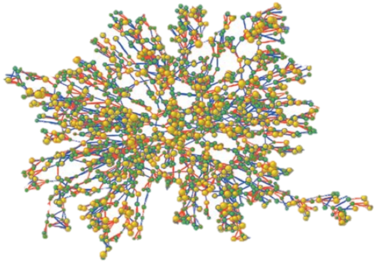
Normalization: betwenness centrality values will be larger in graphs with many nodes. To control for this, we divide centrality values by the number of pairs of nodes in the graph (excluding v):

$\frac{1}{2}(|N|-1)(|N|-2)$ in undirected graphs

$(|N|-1)(|N|-2)$ in directed graphs



Friendship network in a 34-person karate club [Zachary 1977]



Network of friendship, marital tie, and family tie among 2200 people [Christakis & Fowler 2007]

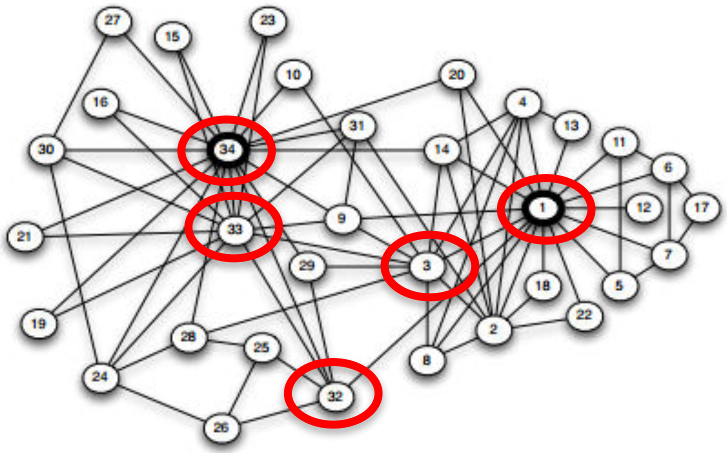
Betweenness Centrality

In: btwnCent = nx.betweenness_centrality(G, **normalized** = True, **endpoints** = False)

In: import operator

In: sorted(btwnCent.items(), key=operator.itemgetter(1), reverse = True)[0:5]

Out: [(1, 0.43763528138528146),
(34, 0.30407497594997596),
(33, 0.14524711399711399),
(3, 0.14365680615680618),
(32, 0.13827561327561325)]



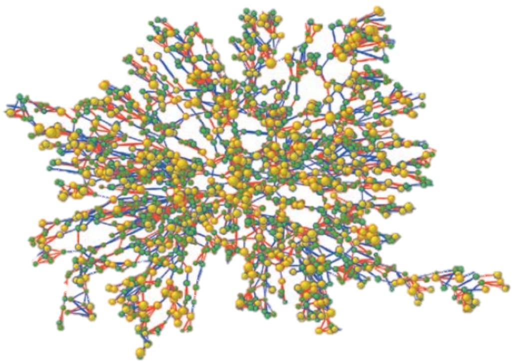
Friendship network in a 34-person karate club
[Zachary 1977]

Betweenness Centrality – Complexity

Computing betweenness centrality of all nodes can be very computationally expensive.

Depending on the algorithm, this computation can take up to $O(|N|^3)$ time.

Approximation: rather can computing betweenness centrality based on all pairs of nodes s, t, we can approximate it based on a sample of nodes.



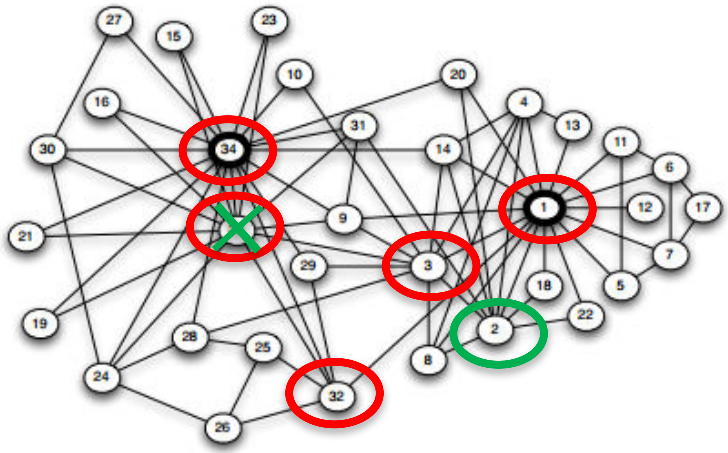
Network of friendship, marital tie, and family tie among 2200 people [Christakis & Fowler 2007]

N = 2200 nodes →
~4.8million pairs of nodes

Betweenness Centrality – Approximation

```
In: btwnCent_approx = nx.betweenness_centrality(G, normalized = True, endpoints = False, k = 10)

In: sorted(btwnCent_approx.items(), key=operator.itemgetter(1), reverse = True)[0:5]
Out: [(1, 0.48269390331890333),
(34, 0.27564694564694564),
(32, 0.20863636363636362),
(3, 0.1697598003848004),
(2, 0.13194624819624817)]
```



Friendship network in a 34-person karate club
[Zachary 1977]

Betweenness Centrality – Subsets

In: btwnCent_subset = nx.betweenness_centrality_subset(G, [34, 33, 21, 30, 16, 27, 15, 23, 10], [1, 4, 13, 11, 6, 12, 17, 7], normalized=True)

In:sorted(btwnCent_subset.items(),key=operator.itemgetter(1), reverse=True)[0:5]

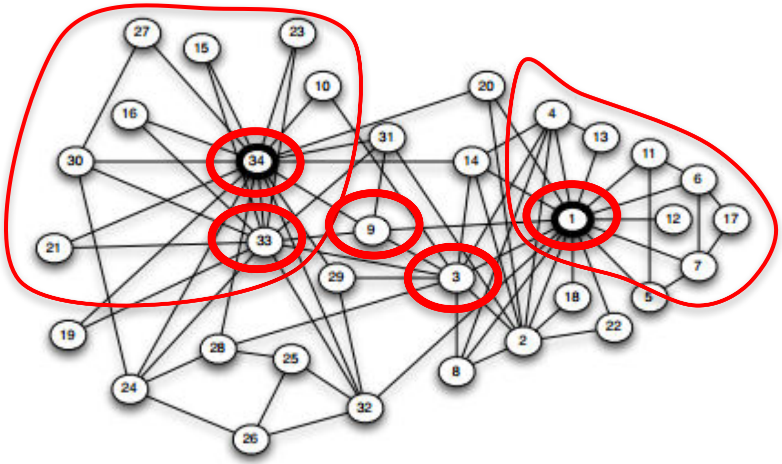
Out: [(1, 0.04899515993265994),

(34, 0.028807419432419434),

(3, 0.018368205868205867),

(33, 0.01664712602212602),

(9, 0.014519450456950456)]



Friendship network in a 34-person karate club
[Zachary 1977]

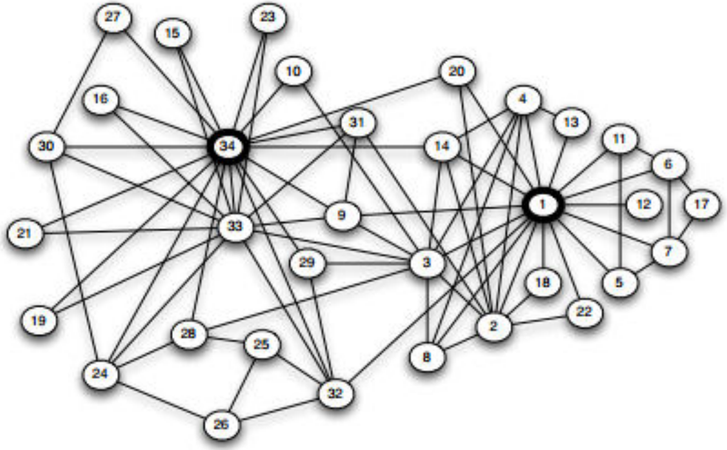
Betweenness Centrality – Edges

We can use betweenness centrality to find important edges instead of nodes:

$C_{btw}(e) = \sum_{s,t \in N} \frac{\sigma_{s,t}(e)}{\sigma_{s,t}}$, where

$\sigma_{s,t}$ = the number of shortest paths between nodes s and t.

$\sigma_{s,t}(e)$ = the number shortest paths between nodes s and t *that pass through edge* e.

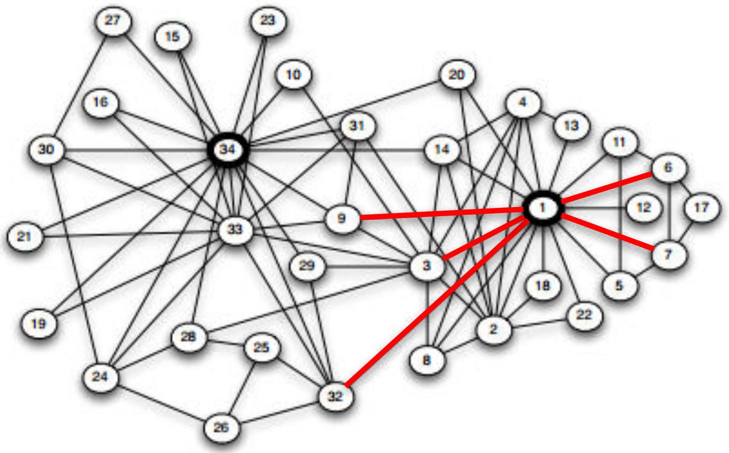


Friendship network in a 34-person karate club
[Zachary 1977]

Betweenness Centrality – Edges

```
In: btwnCent_edge = nx.edge_betweenness_centrality(G, normalized=True)

In: sorted(btwnCent_edge.items(), key=operator.itemgetter(1), reverse = True)[0:5]
Out: [((1, 32), 0.12725999490705373),
 ((1, 7), 0.07813428401663694),
((1, 6), 0.07813428401663694),
((1, 3), 0.0777876807288572),
((1, 9), 0.07423959482783014)]
```



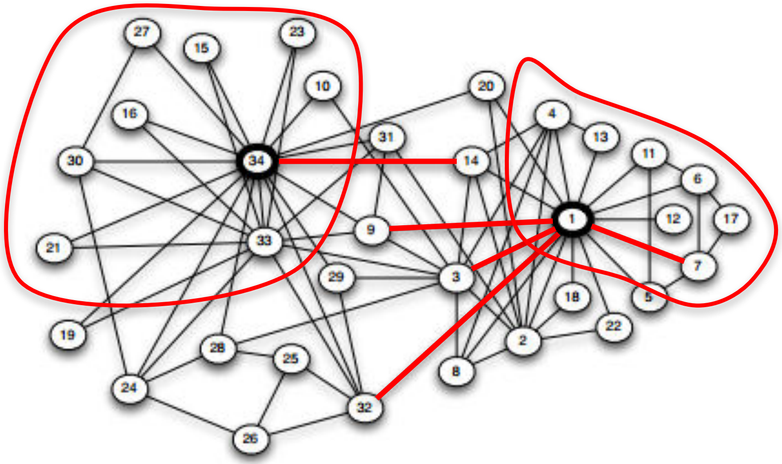
Friendship network in a 34-person karate club
[Zachary 1977]

Betweenness Centrality – Edges

In: btwnCent_edge_subset = nx.edge_betweenness_centrality_subset(G, [34, 33, 21, 30, 16, 27, 15, 23, 10], [1, 4, 13, 11, 6, 12, 17, 7], normalized=True)

In: sorted(btwnCent_edge_subset.items(), key=operator.itemgetter(1), reverse = True)[0:5]

Out: [((1, 32), 0.01366536513595337),
((1, 9), 0.01366536513595337),
((14, 34), 0.012207509266332794),
((1, 3), 0.01211343123107829),
((1, 7), 0.012032085561497326)]



Friendship network in a 34-person karate club
[Zachary 1977]

Summary

Betweenness centrality assumption: important nodes connect other nodes.

$$C_{btw}(v) = \sum_{s,t \in N} \frac{\sigma_{s,t}(v)}{\sigma_{s,t}}$$

Normalization: Divide by number of pairs of nodes.

Approximation: Computing betweenness centrality can be computationally expensive. We can approximate computation by taking a subset of nodes.

Subsets: We can define subsets of source and target nodes to compute betweenness centrality.

Edge betweenness centrality: We can apply the same framework to find important edges instead of nodes.