Centrality

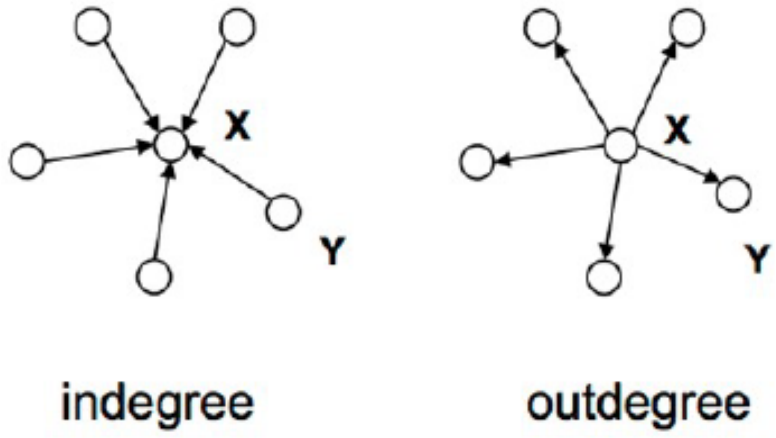
Centrality

- Introduction
- Degree Centrality
- Betweenness Centrality
- Closeness Centrality
- Eigenvector Centrality
- PageRank

Centrality

Finding out which is the most central node is important because it can help:

- disseminating information in the network faster
- stopping epidemics
- protecting the network from breaking



Degree Centrality

Degree centrality is a measure of how many connections or ties one node has to other nodes.

Actors who have more ties may have multiple alternative ways and resources to reach goals—and thus be relatively advantaged.

Degree Centrality is helpful if importance means:

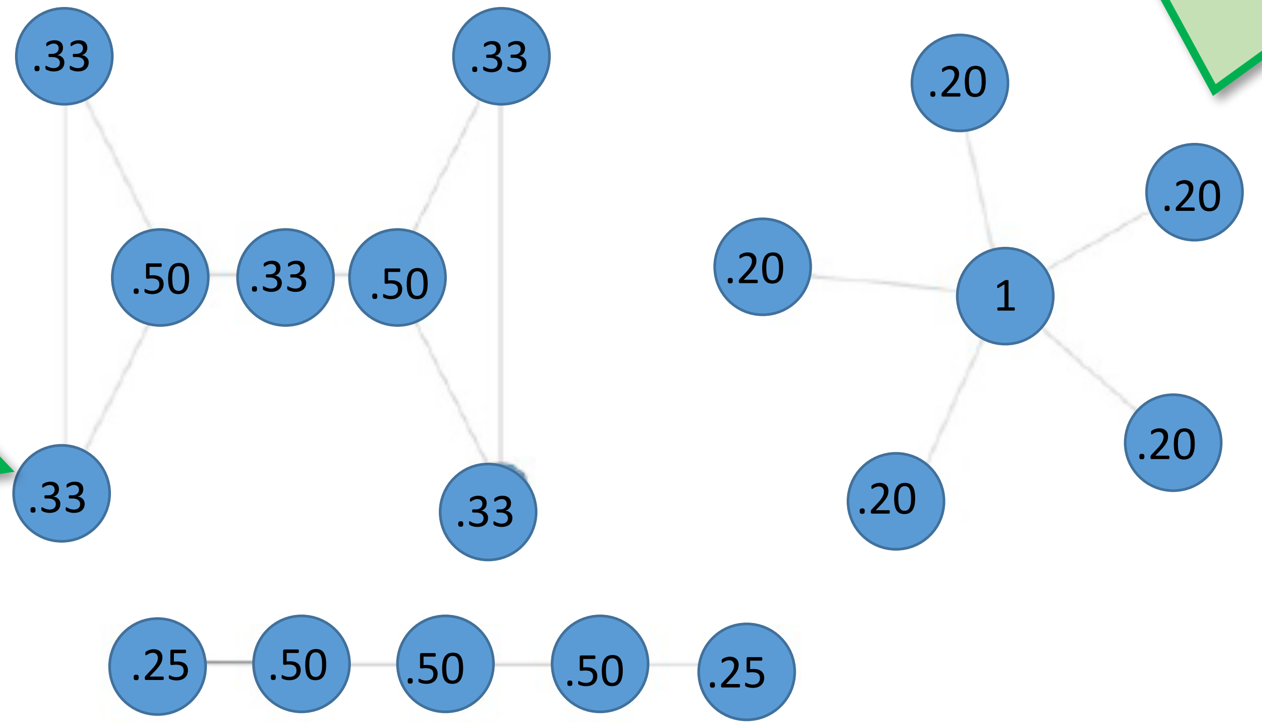
- How popular you are
- How many people you know
- People who will do favors to you
- People you can talk to / have beer with

Degree Centrality

In star network the hub has degree centrality of 1, since total links connected to the hub (k_i) = 5, and N = 6 thus 5/(6-1)=5/5=1

Degree centrality for Node i =
Cdi = Ki / (N-1)

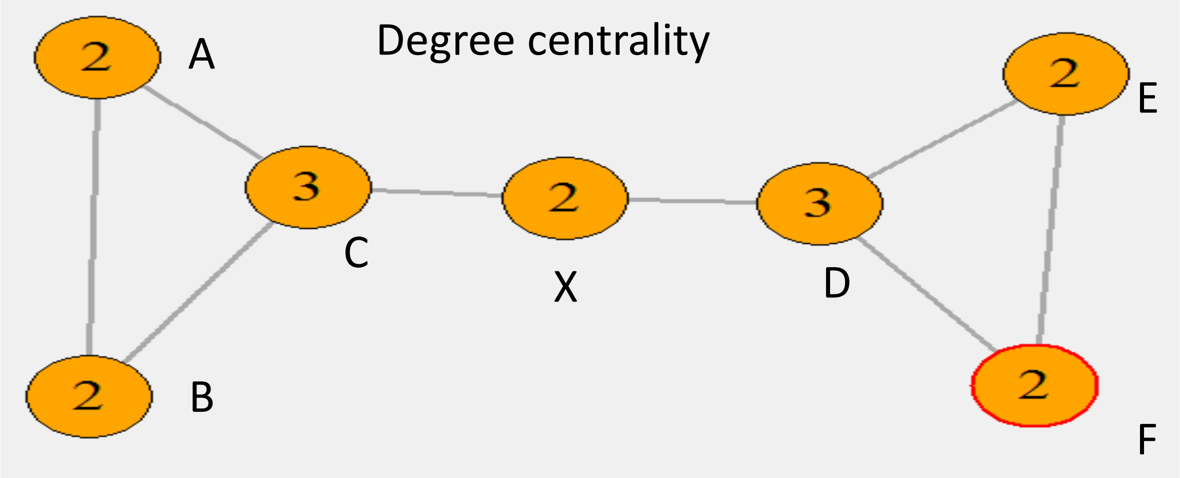
This node has k_i=2,
Total Nodes = 7,
hence Degree Centrality =
2/(7-1)=2/6=1/3=.33



Ki is number of links a given node has.
N is total number of nodes in the network.

M.E.J. Newman. (2010). Networks: An Introduction. Oxford University Press.

Degree Centrality



Degree Centrality for Node X = 2

Since, it has total 2 links (C—X, and X—D)

Normalized Degree Centrality for Node X = 2 / (7-1) = 2/6 = 0.33

Degree Centrality: Directed Graphs

Prominent Actors

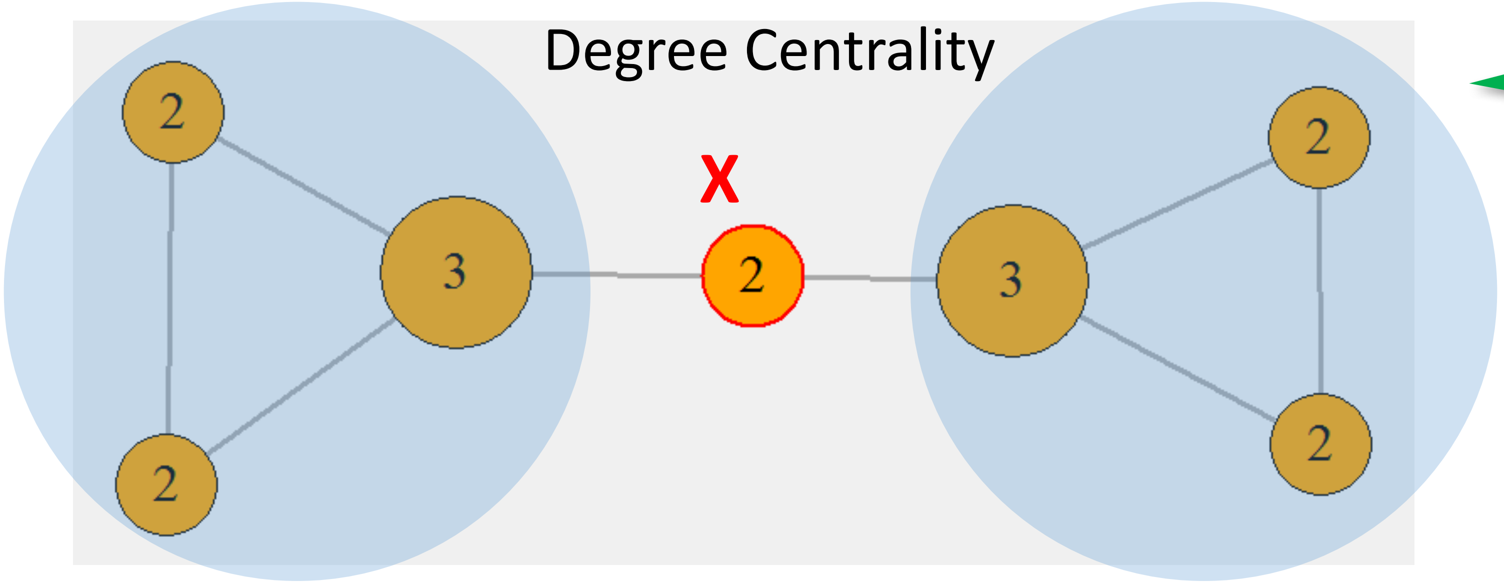
- In-degree Centrality.
- The basic idea is that many actors seek to direct ties to them, and so this may be regarded as a measure of importance.

Influential Actors

- Out-degree centrality.
- Actors who have high out-degree centrality may be relatively able to exchange with others, or disperse information quickly to many others.

Limits to Degrees

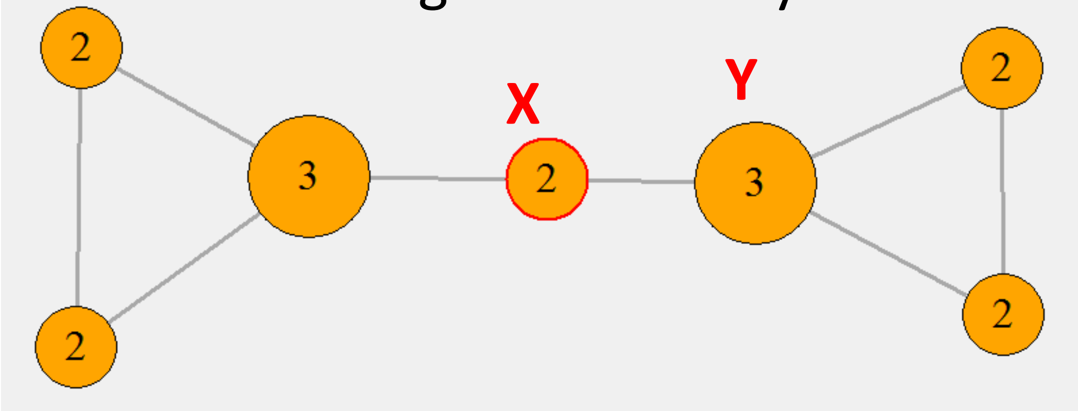
Degree is not everything when we are interested in measuring a node's ability to broker between groups, and or to control flow of Information.



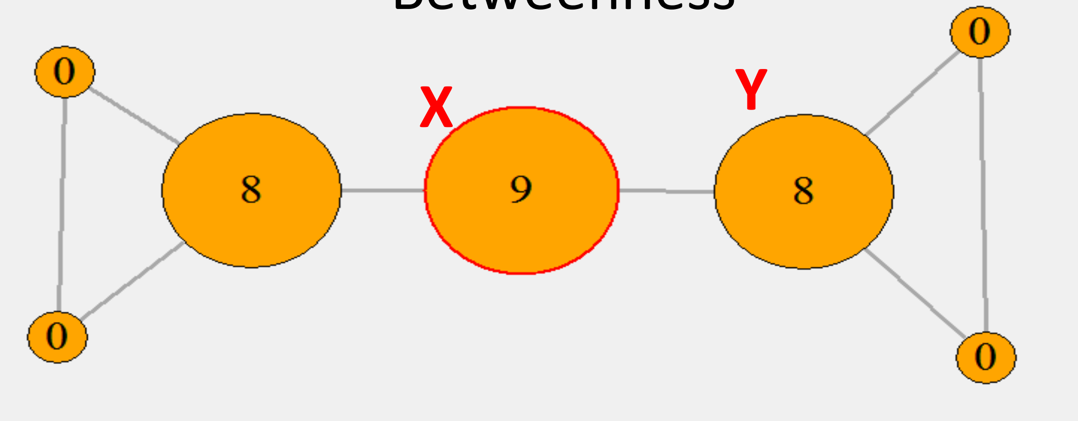
Node X has just 2 **degrees** even though it is performing a critical role of "connecting" two "communities"

Who is more central, “X” or “Y”?

Degree Centrality

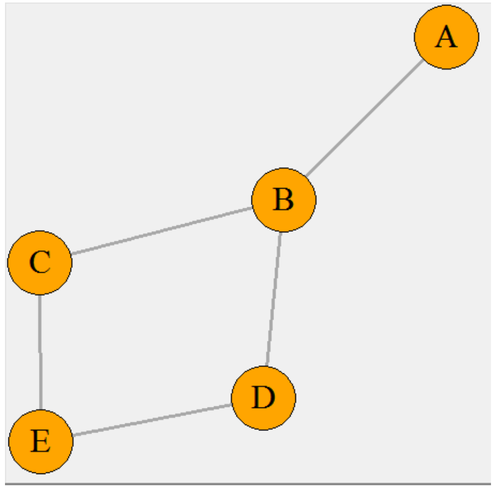


Betweenness



Note: Y has more degrees!

Betweenness Centrality



Betweenness Centrality for Node b:

There is only shortest path between a and c, and b sits in that (1/1).

There is only one shortest path between a and d, and b sits in that (1/1).

There are only two shortest paths between a and e, and b sits in both of them (2/2).

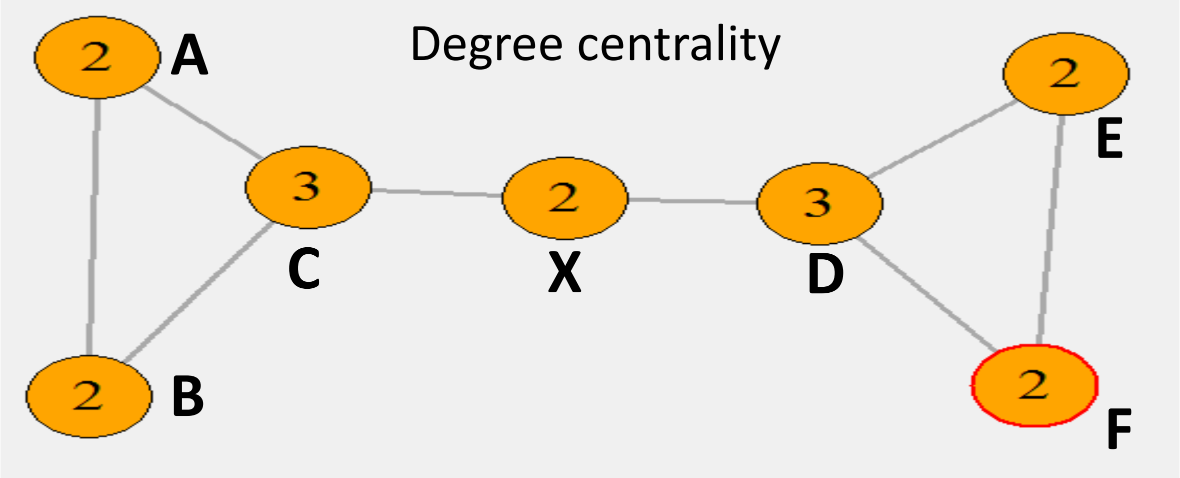
There are two shortest paths between c and d, and b sits in one of them (1/2).

There are no other shortest paths between any two nodes, where b sits in between.

Normalization factor is 4 x 3 /2 = 6

Betweenness Centrality = (1/1 + 1/1 + 2/2 + 1/2)/6 = 3.5 /6

Betweenness Centrality



Betweenness centrality for Node **X**: 9

Since number of shortest paths that go through **X** = 9

A--D, A--E, A--F = 3

B--D, B--E, B--F = 3

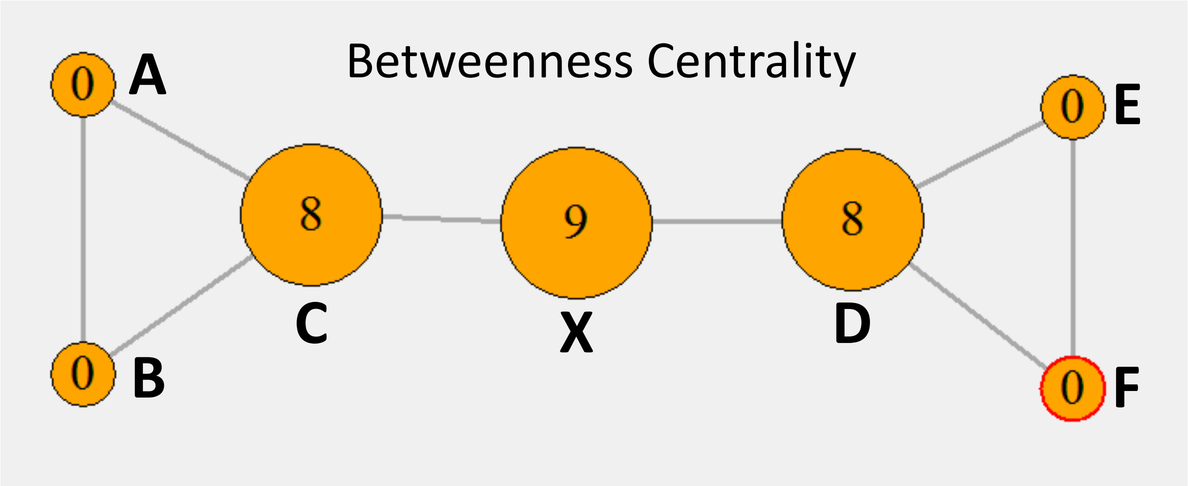
C--D, C--E, C--F = 3

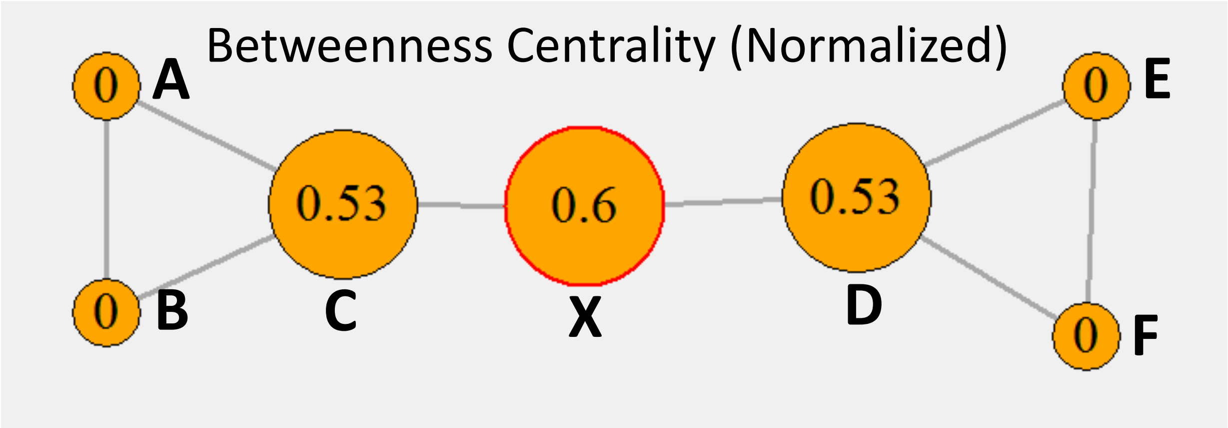
Normalized Betweenness Centrality for Node **X**

= 9 / [(N-1)(N-2)/2]

=9 / (6 X 6 / 2)

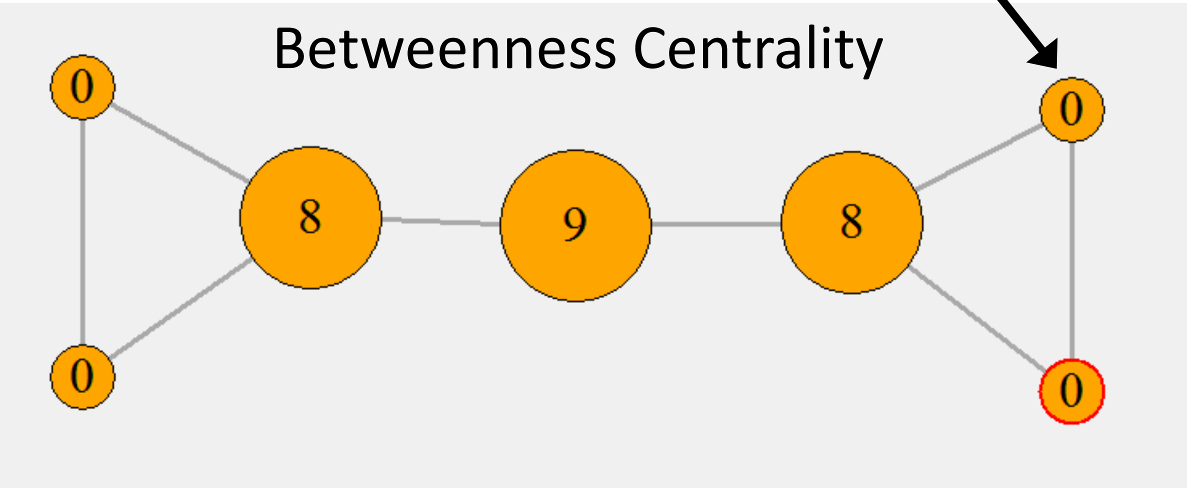
= 9/15 = 0.60



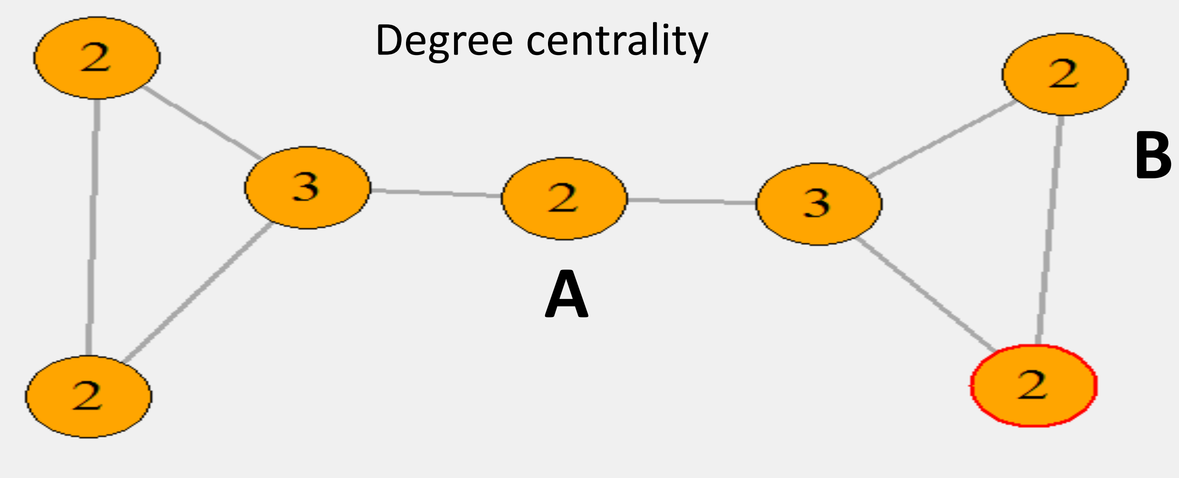


Limits to Betweenness

Betweenness is not everything when we are interested in measuring how efficiently a node can obtain information



Closeness Centrality



Closeness centrality for middle node **A**:

Sum of all geodesic distances between node **A** and all other nodes = 1+2+2+2+2+1=10

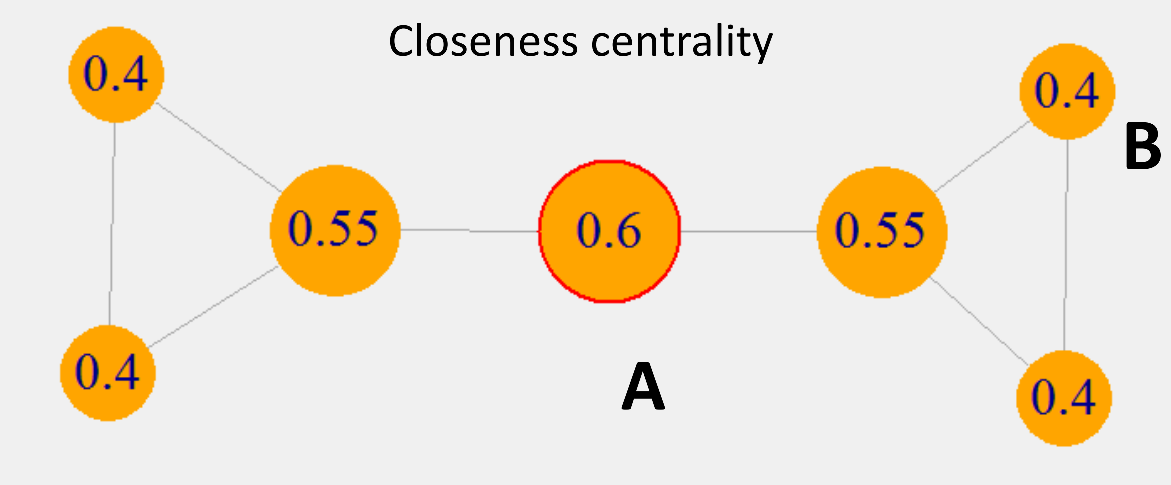
Number of all other nodes (i.e. N-1) = 6

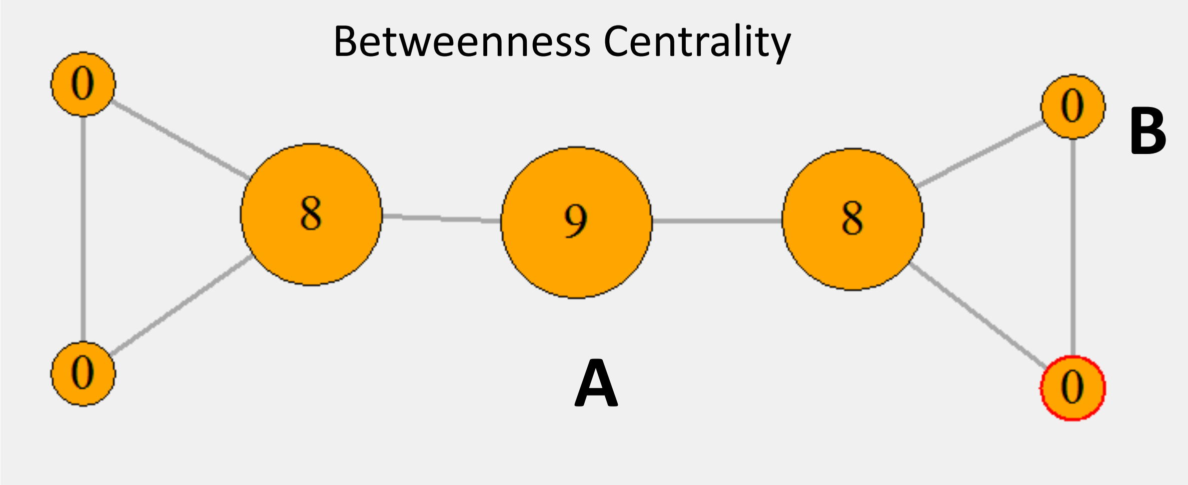
Average geodesic distance = 10/6

Closeness centrality = 1/average geodesic distance = 6/10=.6

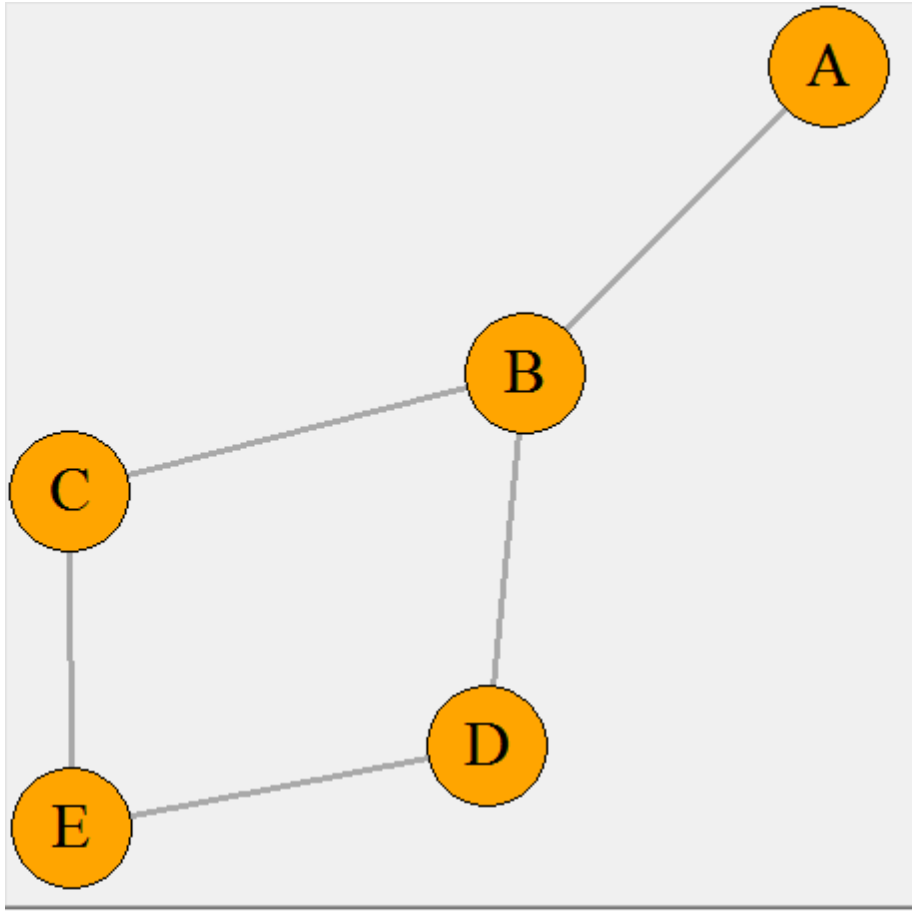
Node A has lower degree centrality because of just 2 links, but high betweenness centrality.

Node B, on the other hand, has higher degree centrality than betweenness centrality, since B does not sit in between any two nodes.





Closeness Centrality: Example



Closeness Centrality for Node **B**:

Geodesic distance between **B** and **A** = 1

Geodesic distance between **B** and **C** = 1

Geodesic distance between **B** and **D** = 1

Geodesic distance between **B** and **E** = 2

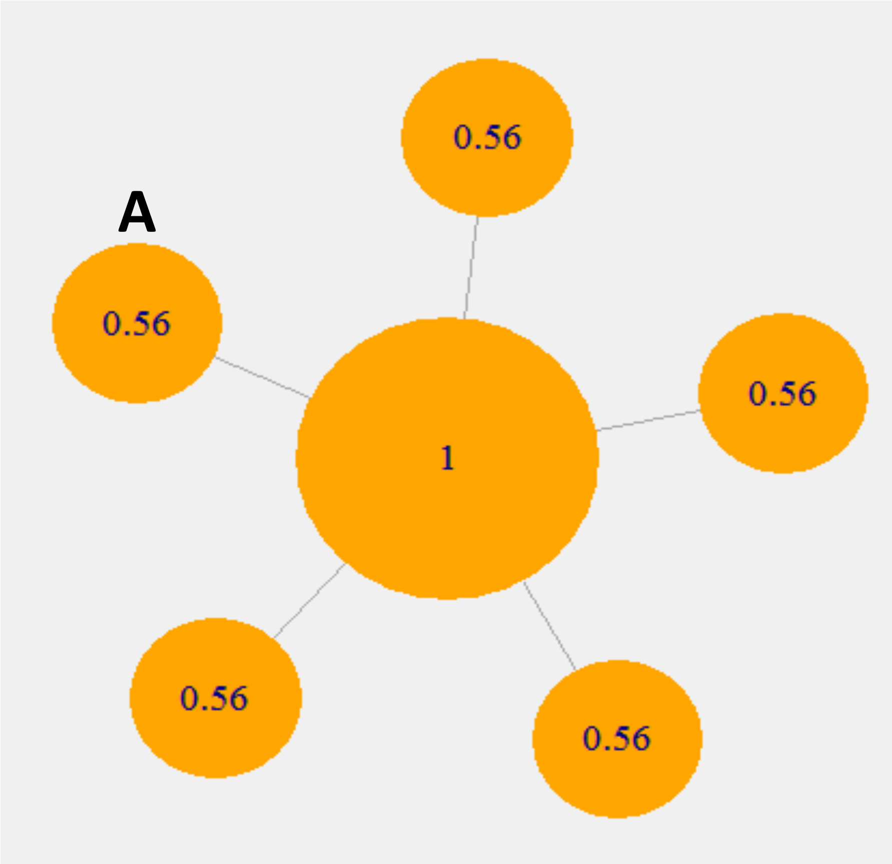
Total geodesic distance between **B** and other nodes = 5

Normalization factor: number of other nodes = 4

Average shortest distance = 5/4

Closeness Centrality = 4/5 = .8

Closeness Centrality:
1 / average shortest distance



Node **X**:
Average Shortest Path = (1+1+2+3+4+4)/6
= 6/15 = 2.5

Closeness Centrality = 1/2.5 = .40

Node **A**:
Average Shortest Path =
(1+2+2+2+2)/5 = 9/5 = 1.8

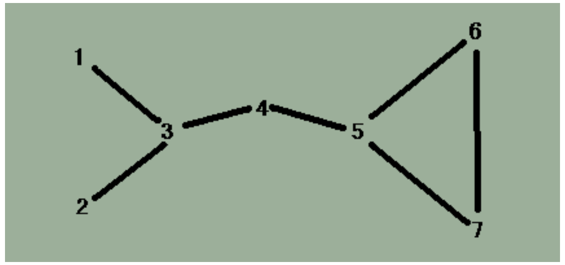
Closeness Centrality = 1/1.8 = .56

Generally, the 3 centrality types will be positively correlated.
When they are not (low) correlated, it probably tells you something interesting about the network:

	Low Degree	Low Closeness	Low Betweenness
High Degree		Embedded in cluster that is far from the rest of the network.	Ego's connectations are redundant; communication bypasses him/her.
High Closeness	Key player tied to important/active alters		Probably multiple paths in the network, ego is near many people, but so are many others.
High Betweenness	Ego's few ties are crucial for network flow	Ego monopolizes the ties from a small number of people to many others.	

http://www2.unb.ca/~ddu/6634/Lecture_notes/Lecture_4_centrality_measure.pdf

Centrality: Importance



Degree Centrality:

- Popularity in Friendship Networks.
- **The most popular person should have the highest number of friends.**

Betweenness Centrality:

- Ability to control information flow, and broker.
- **To control information flow, a node should be between other nodes because the node can interrupt information flow between them.**

Closeness Centrality:

- Ability to get information.
- **The node in the nearest position on average can most efficiently obtain information.**

http://www.sscnet.ucla.edu/soc/faculty/mcfarland/soc112/cent-ans.htm

Eigenvector Centrality

A node is important if it is connected to important nodes.

$$X_i = \sum_{j \in \Lambda(i)} X_j \qquad X_i = \sum_{j=1}^{N} A_{ij} X_j \qquad \boxed{AX = \lambda X}$$

The solution (when exists) gives the node centrality. We take the highest λ.

This concept is at the core of the ranking algorithm of Google.

Source: M. E. J. Newman. (2010). *Networks: An Introduction*. Oxford University Press.

Eigenvector Centrality (Continued)

Degree Centrality depends on having many connections, but what if these connections are pretty isolated?

A central node should be one connected to powerful nodes.

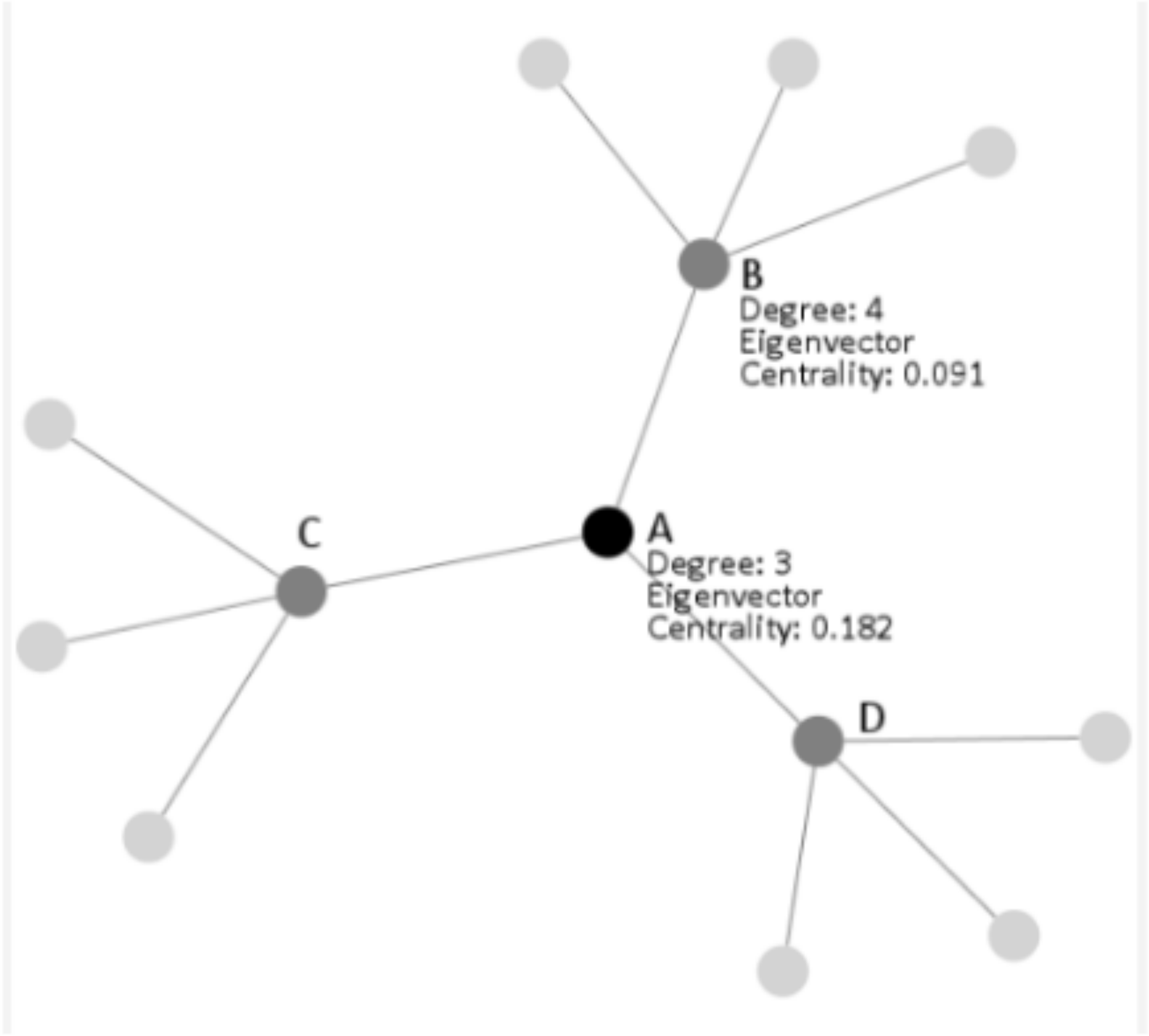
$$x_v = \frac{1}{\lambda} \sum_{t \in M(v)} x_t = \frac{1}{\lambda} \sum_{t \in G} a_v, x_t$$

Neighborhood of Xv

Adjacency Matrix of the graph

Moreover, a node with high eigenvector centrality is not necessarily highly linked (the node might have few but important linkers).

Degree Centrality | Betweenness Centrality | Closeness Centrality | **Eigenvector Centrality**



https://www.cl.cam.ac.uk/teaching/1314/L109/stna-lecture3.pdf

Which Nodes are Most "Central"?

Definition of "central" varies by context/purpose.

Local measure:

- Degree

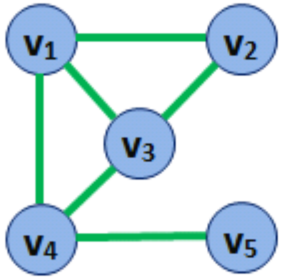
Relative to the rest of the network:

- Betweenness
- Closeness
- Eigenvector (Bonacich power centrality)

Source: M. E. J. Newman. (2010). *Networks: An Introduction*. Oxford University Press.

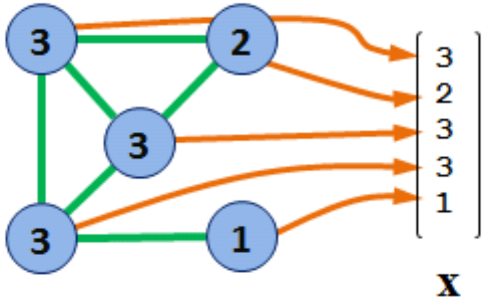
Eigenvector Centrality: Example

1 Consider the graph below and its 5x5 adjacency matrix **A**:



$$\mathbf{A} = \begin{pmatrix} - & 1 & 1 & 1 & 0 \\ & - & 1 & 0 & 0 \\ & & - & 1 & 0 \\ & & & - & 1 \\ & & & & - \end{pmatrix}$$

2 And then consider, x, a 5x1 vector of values, one for each vertex in the graph. In this case, we've used the degree centrality of each vertex.



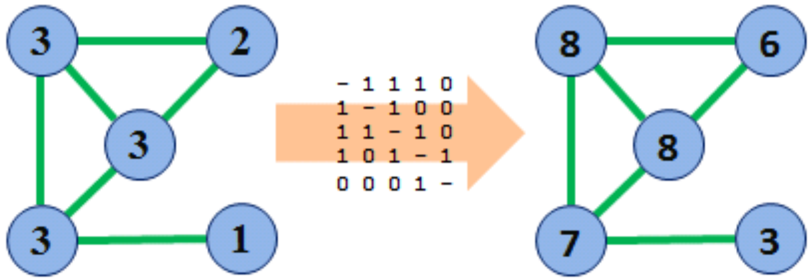
3 Now let's look at what happens when we multiply the vector x by the matrix A. The result, of course, is another 5x1 vector.

$$\mathbf{A} \times \mathbf{x} = \begin{pmatrix} - & 1 & 1 & 1 & 0 \\ 1 & - & 1 & 0 & 0 \\ 1 & 1 & - & 1 & 0 \\ 1 & 0 & 1 & - & 1 \\ 0 & 0 & 0 & 1 & - \end{pmatrix} \begin{pmatrix} 3 \\ 2 \\ 3 \\ 3 \\ 1 \end{pmatrix} = \begin{pmatrix} 0\text{x}3 + 1\text{x}2 + 1\text{x}3 + 1\text{x}3 + 0\text{x}1 \\ 1\text{x}3 + 0\text{x}2 + 1\text{x}3 + 0\text{x}3 + 0\text{x}1 \\ 1\text{x}3 + 1\text{x}2 + 0\text{x}3 + 1\text{x}3 + 0\text{x}1 \\ 1\text{x}3 + 0\text{x}2 + 1\text{x}3 + 0\text{x}3 + 1\text{x}1 \\ 0\text{x}3 + 0\text{x}2 + 0\text{x}3 + 1\text{x}3 + 0\text{x}1 \end{pmatrix} = \begin{pmatrix} 8 \\ 6 \\ 8 \\ 7 \\ 3 \end{pmatrix}$$

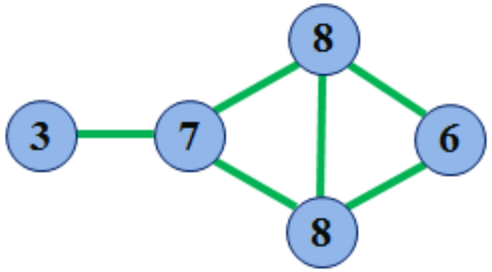
http://djjr-courses.wikidot.com/soc180:eigenvector-centrality

Eigenvector Centrality: Example Continued

4. In other words, what multiplication by the adjacency matrix does, is reassign each vertex the sum of the values of its neighbor vertices.



5. This has, in effect, "spread out" the degree centrality. That this is moving in the direction of a reasonable metric for centrality can be seen better if we rearrange the graph a little bit:



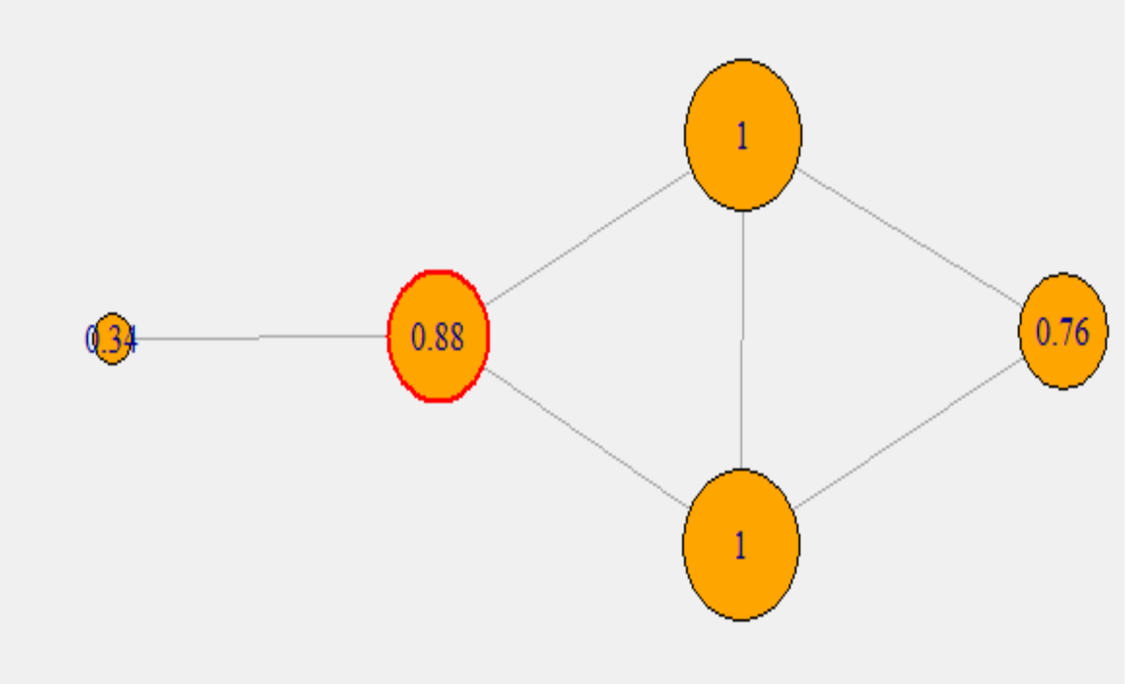
http://djjr-courses.wikidot.com/soc180:eigenvector-centrality

Eigenvector vs. Degree centrality

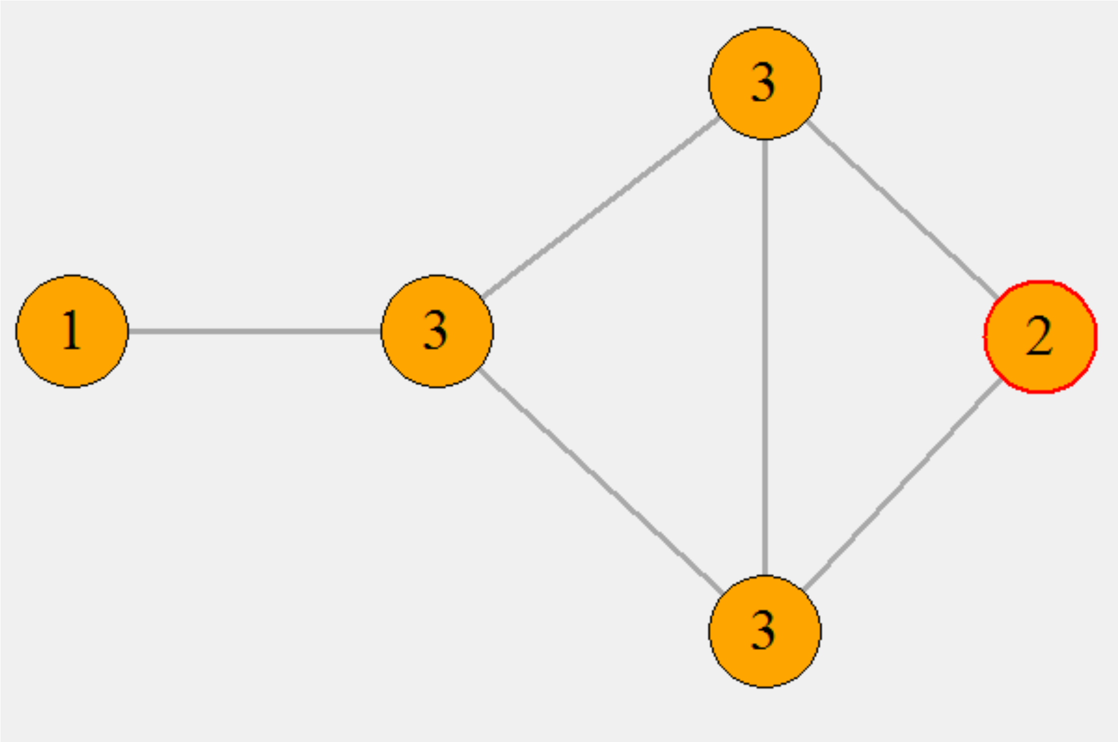
A node is important if it is linked to by other **important** nodes.

Eigenvector

Size proportional to eigenvector value



Degree centrality



Eigenvector Centrality

Suppose we multiplied the resulting vector by A again, how might we interpret what that meant? In effect, we'd be allowing this centrality value to once again "spread" across the edges of the graph. And we'd notice that the spread is in both directions (vertices both give to and get from their neighbors). We might speculate that this process might eventually reach an equilibrium when the amount coming into a given vertex would be in balance with the amount going out to its neighbors. Since we are just adding things up, the numbers would keep getting bigger, but we could reach a point where the share of the total at each node would remain stable. At that point we might imagine that all of the "centrality-ness" of the graph had equilibrated and the value of each node completely captured the centrality of all of its neighbors, all the way out to the edges of the graph.

Eigenvector Centrality, Continued

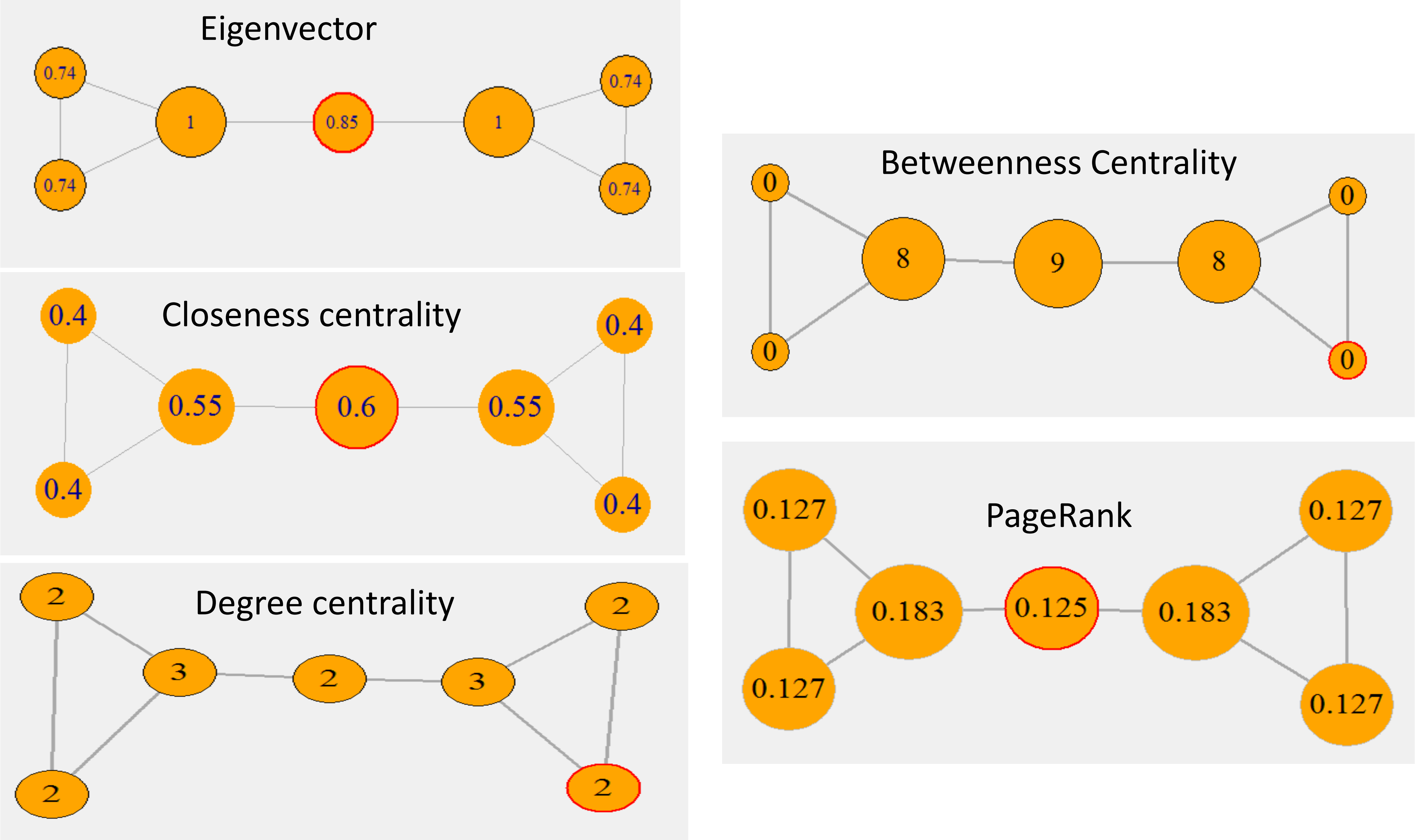
So, we can write this last equation as MX = Y. But our point was that there might be some set of values on our vertices, (that is to say, some {\bf x}), for which multiplication by A would not change the relative sizes of the vector's component values — the numbers would get bigger, but all by the same factor. We can express this like this:

$$\mathbf{Mx} = \lambda \mathbf{x}$$

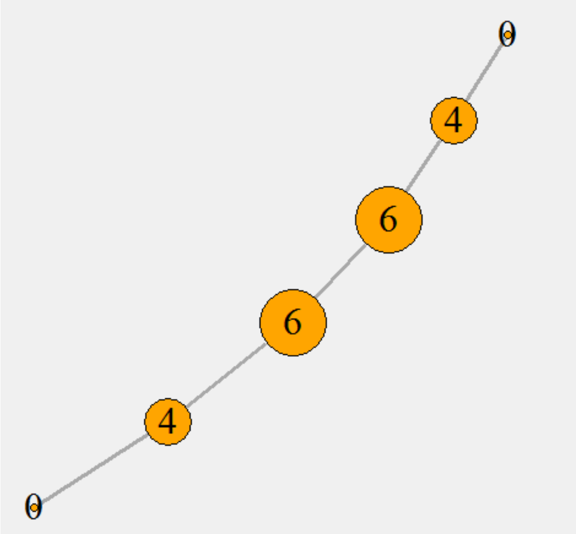
$$\mathbf{M} \times \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix} = \begin{pmatrix} \lambda x_1 \\ \lambda x_2 \\ \dots \\ \lambda x_n \end{pmatrix}$$

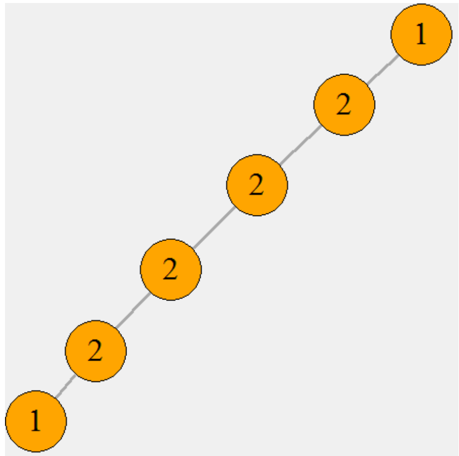
A vector (x) with this property — it can be multiplied by the adjacency matrix (M) for a graph and return itself multiplied by a scalar — is a characteristic of this particular adjacency matrix. In other words, there is something about the particular pattern of connections in this graph that leads to a specific set of vertex values that will have the equilibrium property described above.

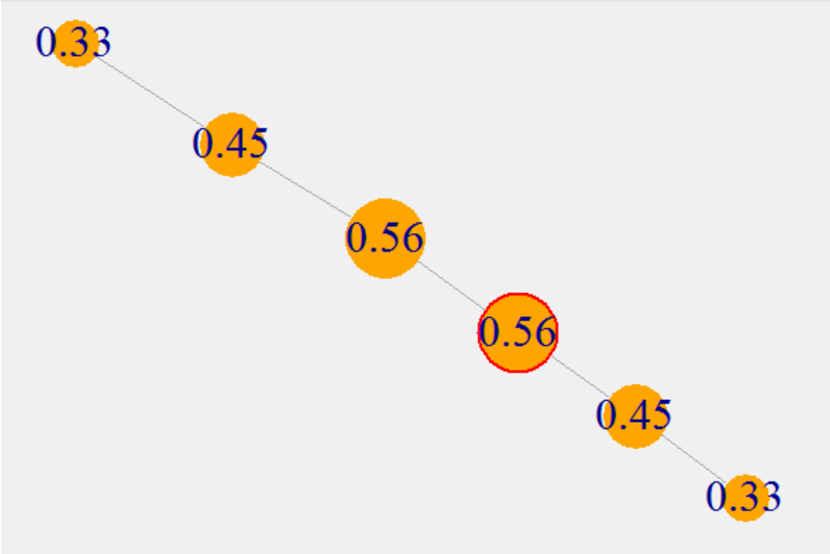
This vector is called an Eigenvector of the matrix **A**. The elements of this vector are the Eigenvector centralities of the vertices of the graph.

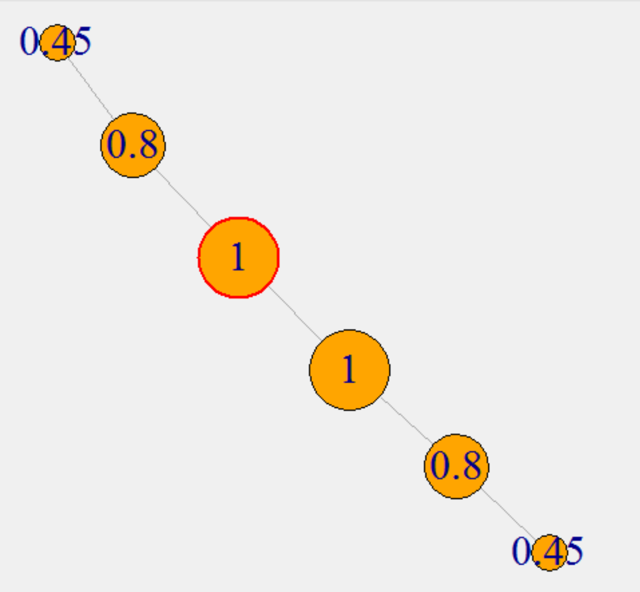

Eigenvector
0.74
1
0.85
1
0.74
0.74
0.74
Closeness centrality
0.4
0.55
0.6
0.55
0.4
0.4
0.4
Degree centrality
2
3
2
3
2
2
2
Betweenness Centrality
0
8
9
8
0
0
0
PageRank
0.127
0.183
0.125
0.183
0.127
0.127
0.127

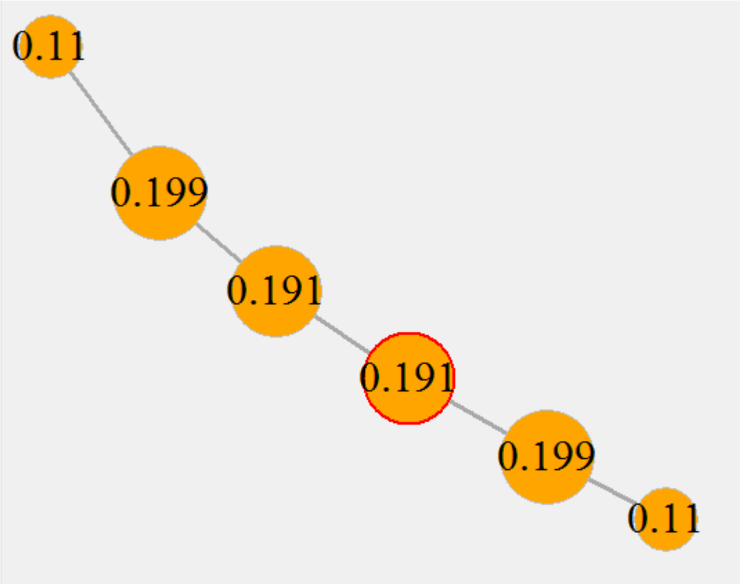
Betweenness Centrality, Degree Centrality, Closeness Centrality, Eigenvector Centrality, PageRank Centrality



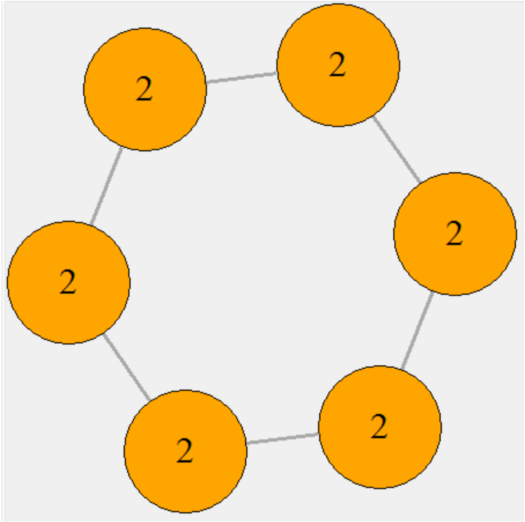


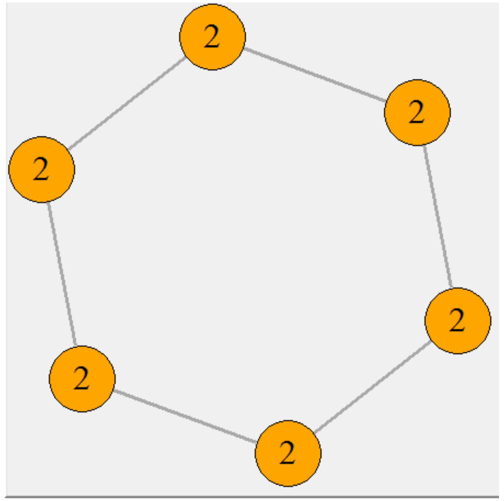


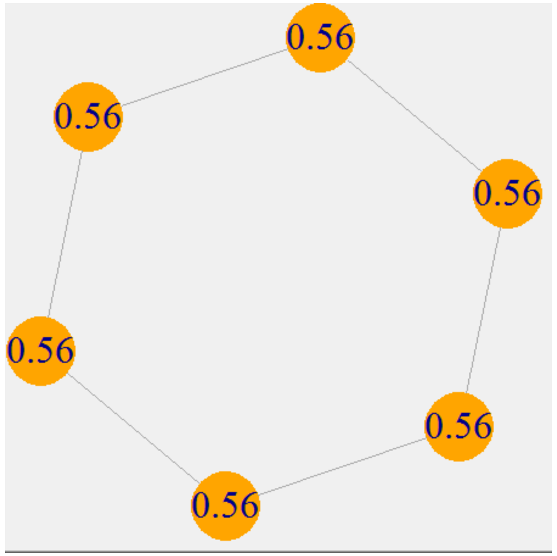


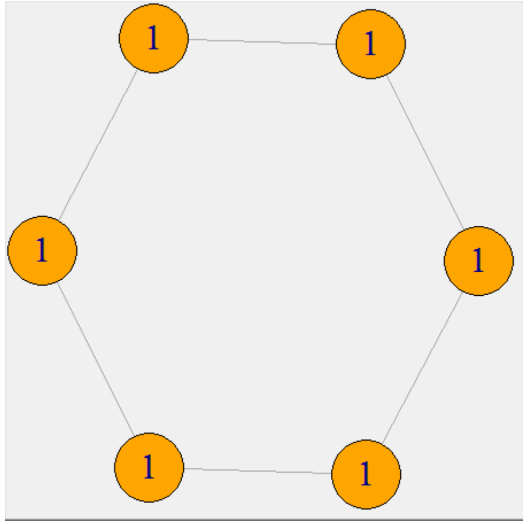


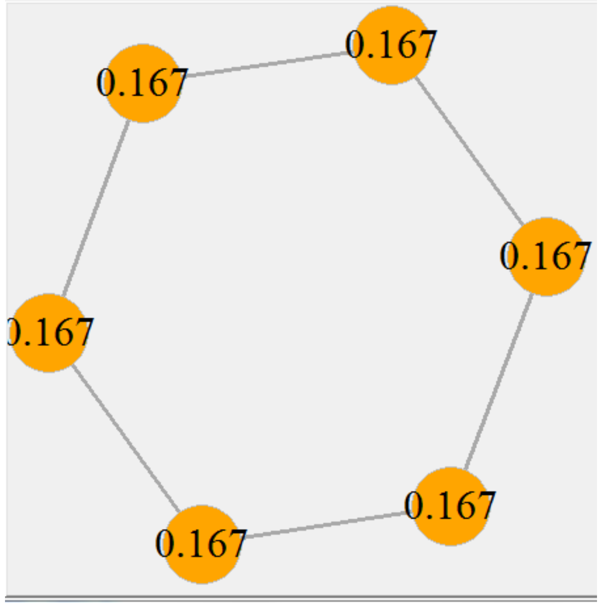
Betweenness Centrality, Degree Centrality, Closeness Centrality, Eigenvector Centrality, PageRank Centrality



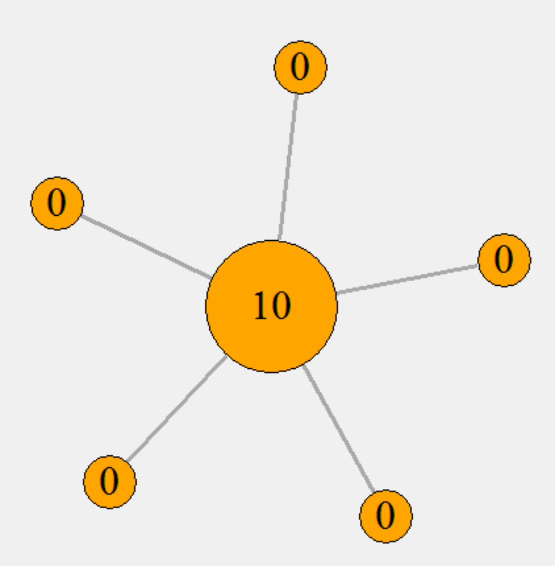


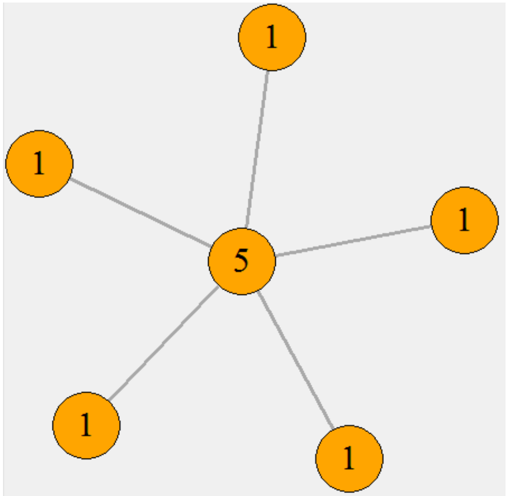


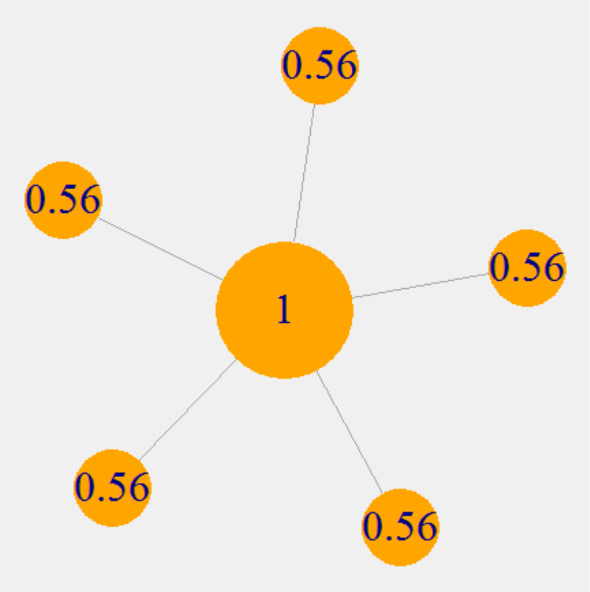


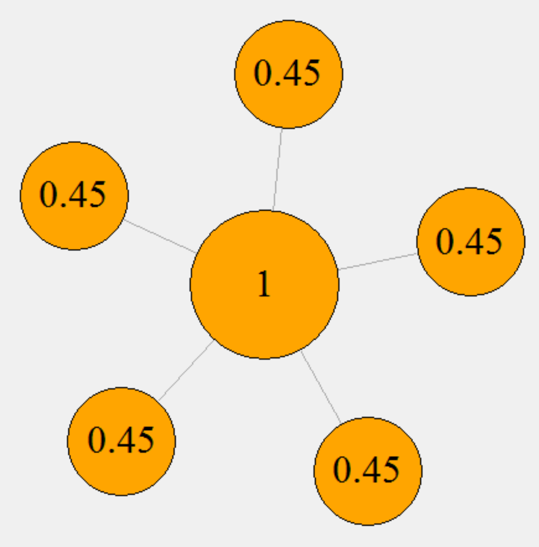


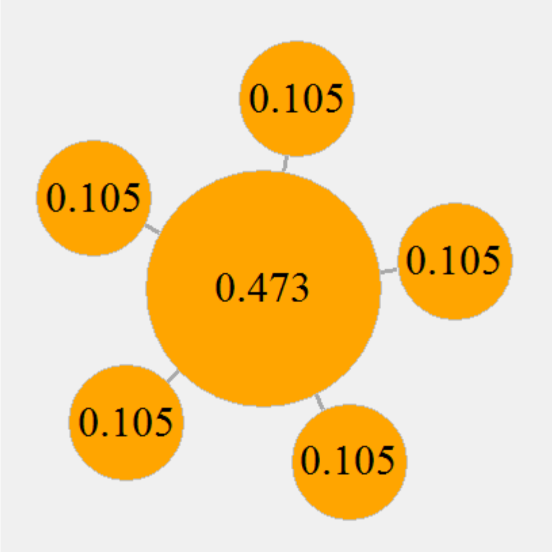
Betweenness Centrality, Degree Centrality, Closeness Centrality, Eigenvector Centrality, PageRank Centrality



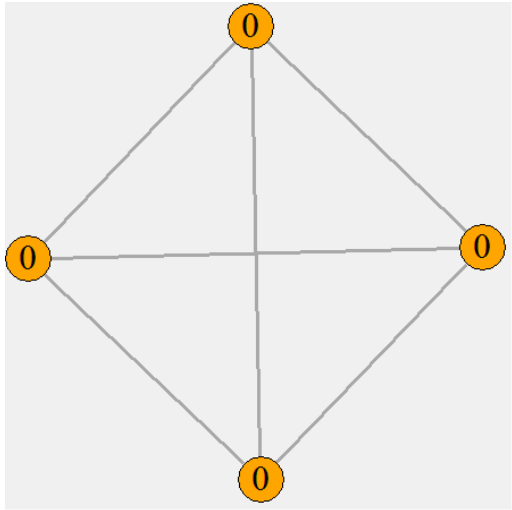


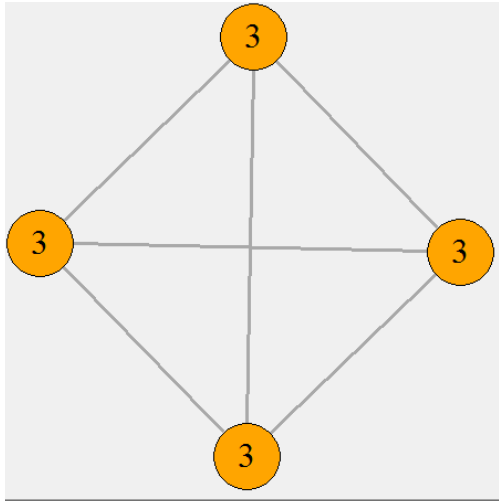


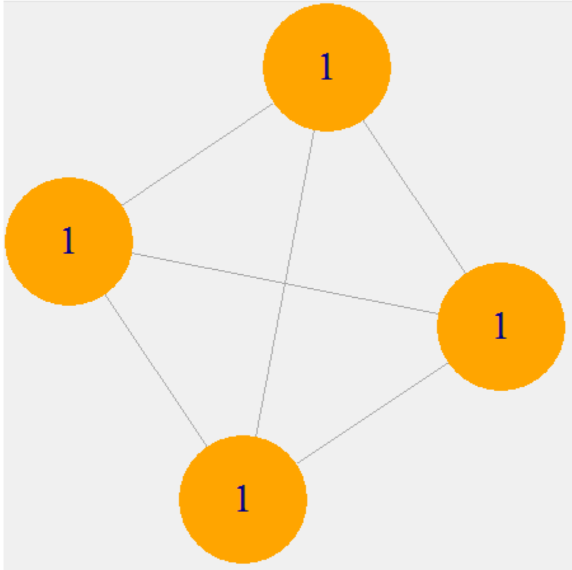


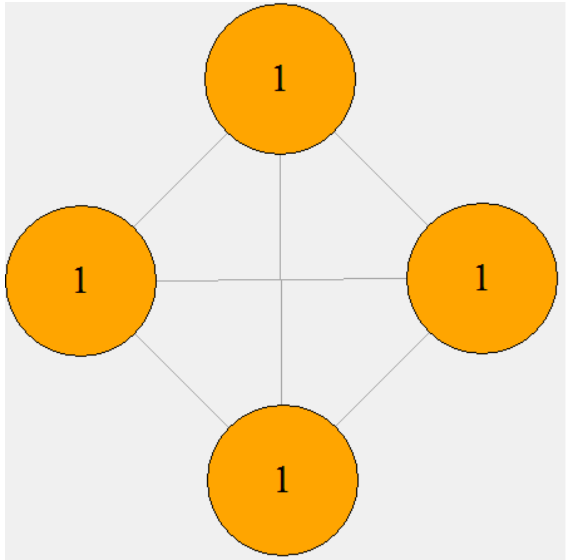


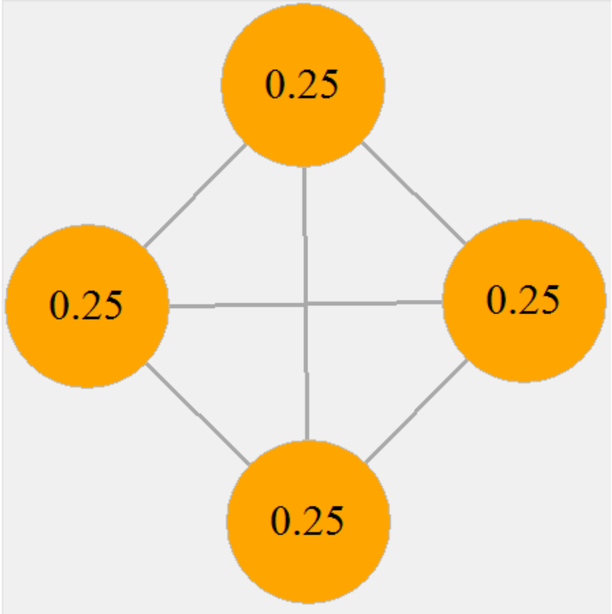
Betweenness Centrality, Degree Centrality, Closeness Centrality, Eigenvector Centrality, PageRank Centrality



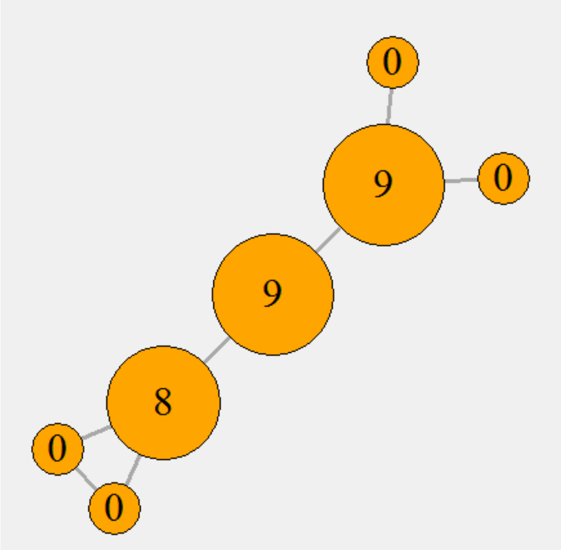


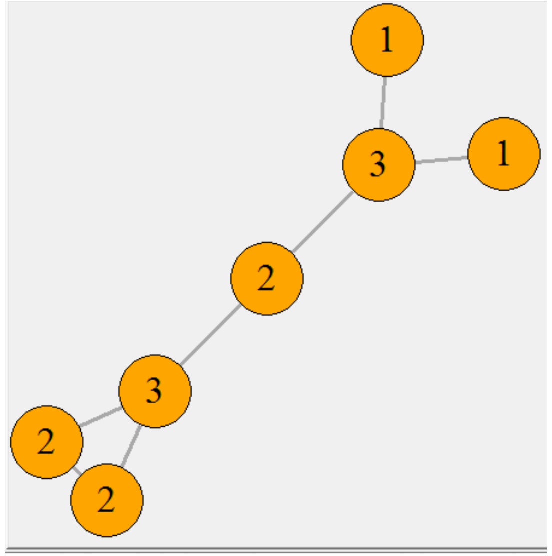


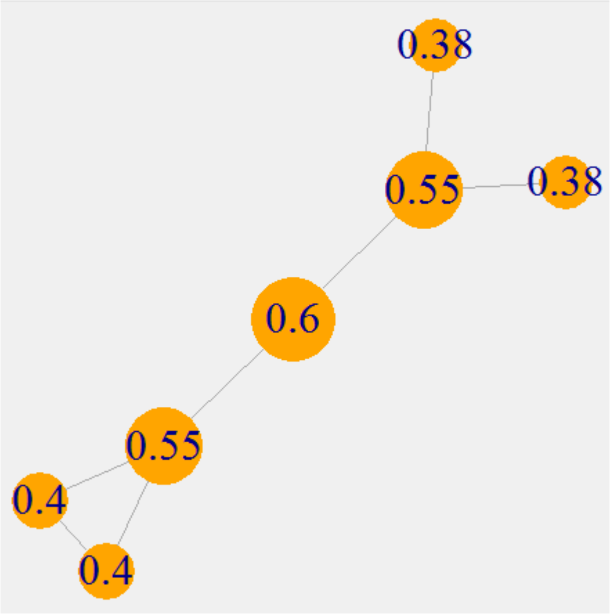


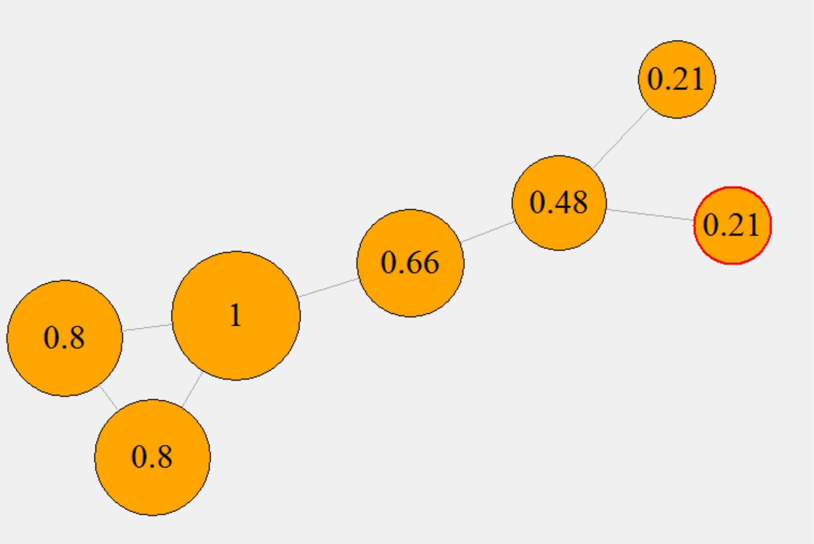


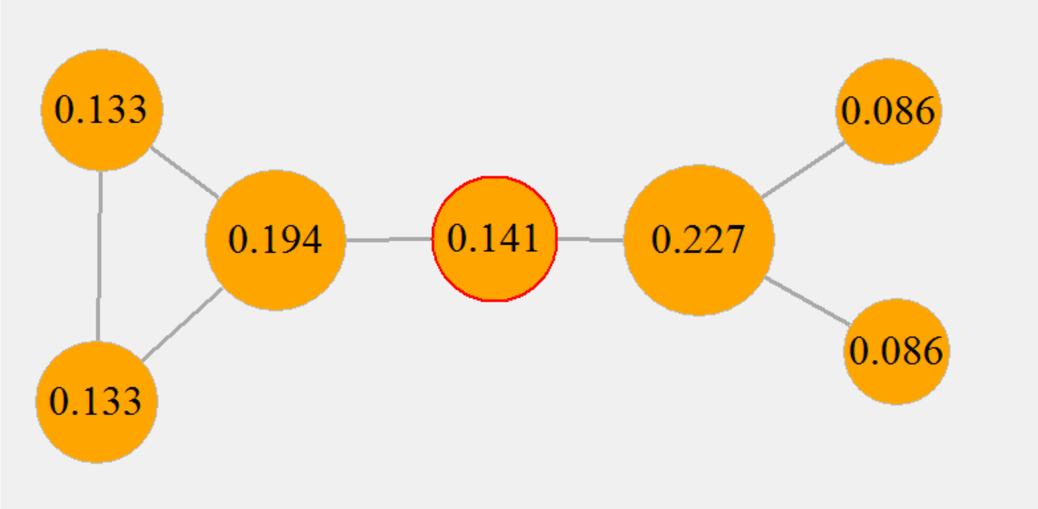
Betweenness Centrality, Degree Centrality, Closeness Centrality, Eigenvector Centrality, PageRank Centrality



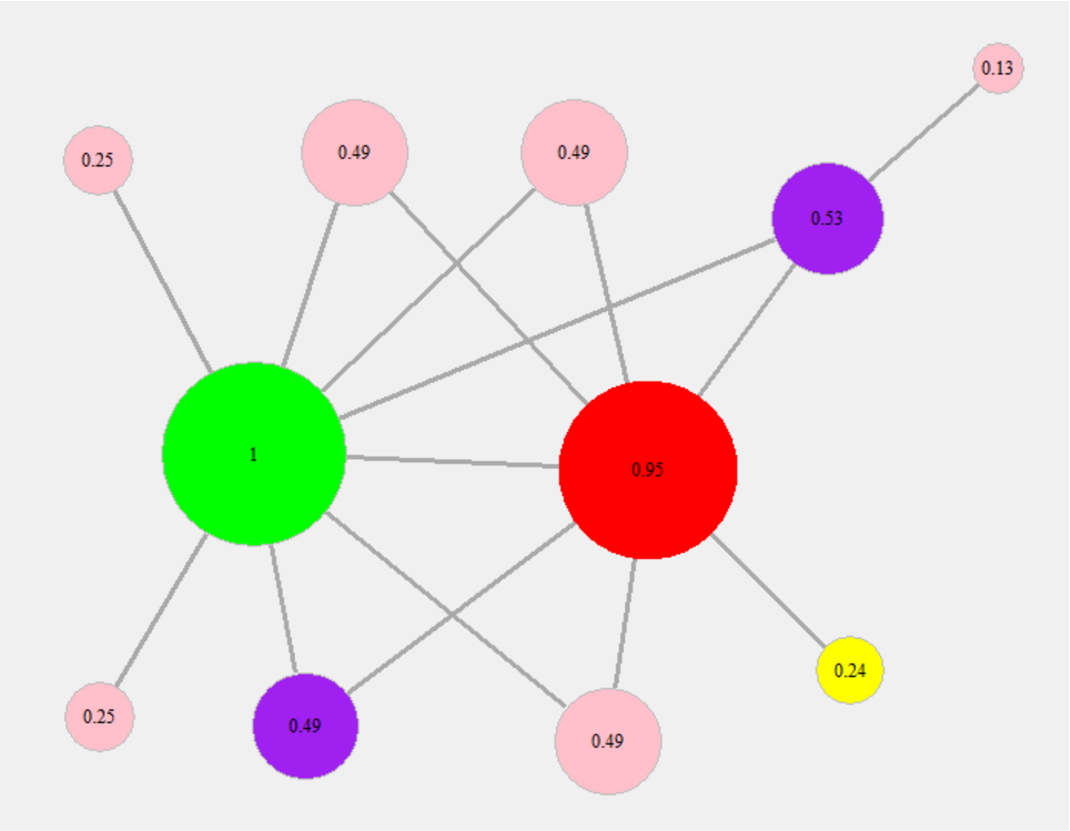


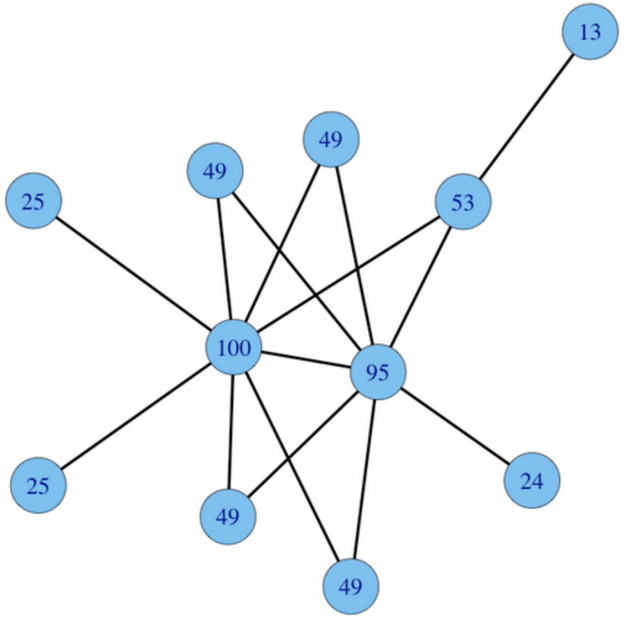






Graph showing eigenvector centrality values for each node.

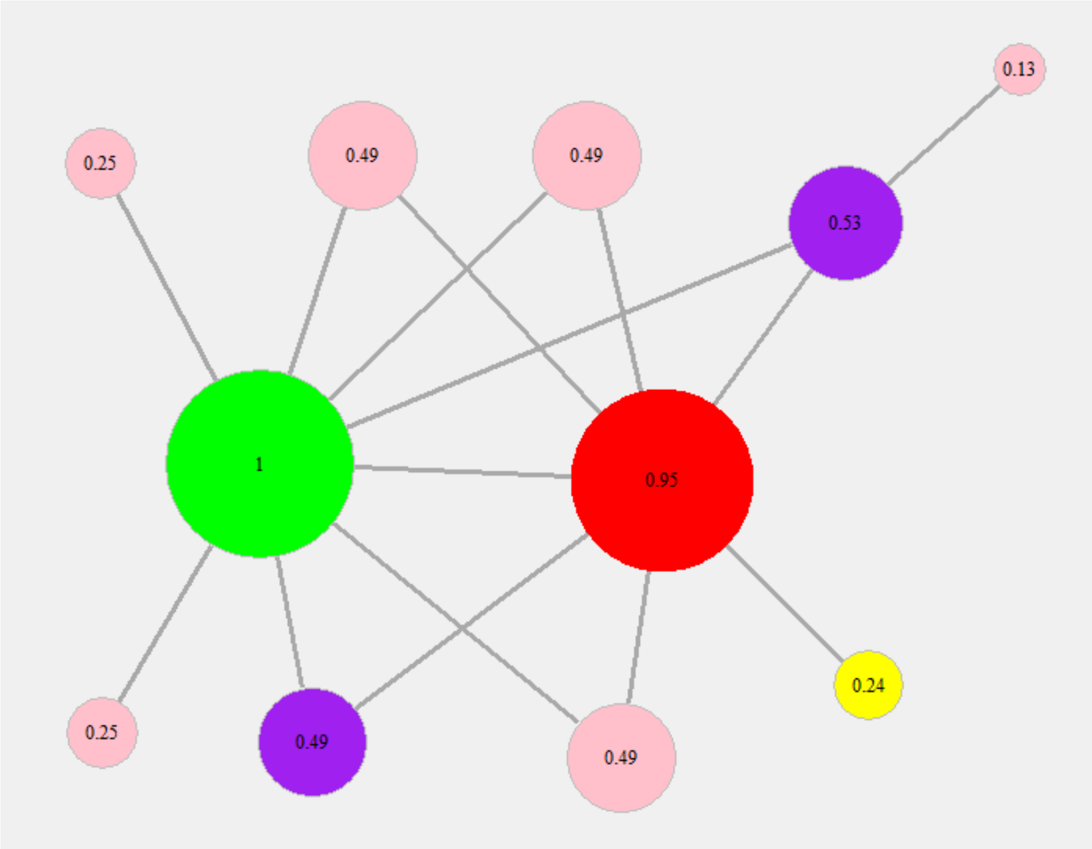


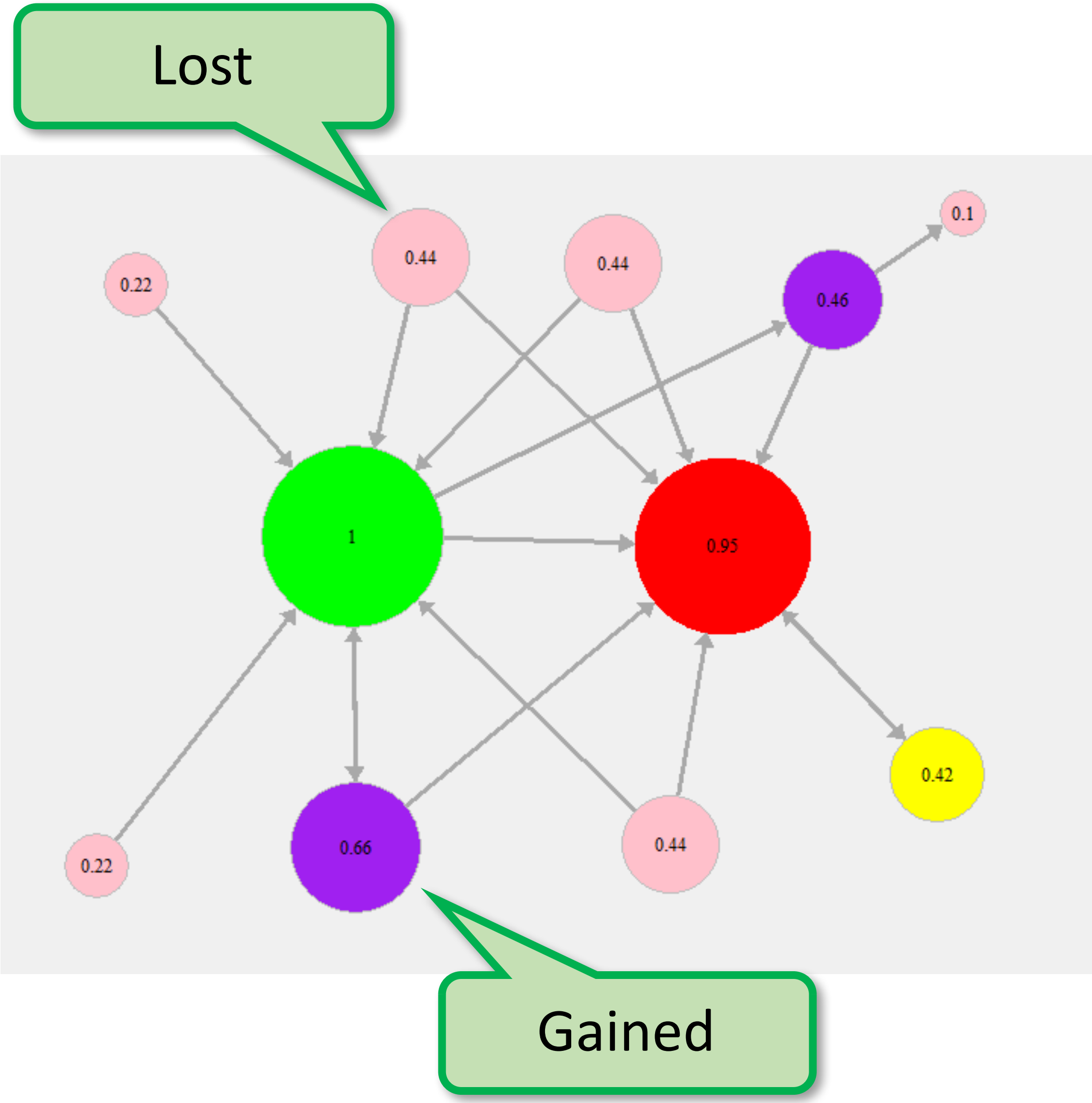


http://www.sci.unich.it/~francesc/teaching/network/eigenvector.html

Degree Centrality | Betweenness Centrality | Closeness Centrality | Eigenvector Centrality

How eigenvector changes for directed and non-directed graphs





Summarizing

Centrality indices are answers to the question, "**What characterizes an important node?**"

The word "importance" has a **wide number of meanings**, leading to **many different definitions of centrality**.

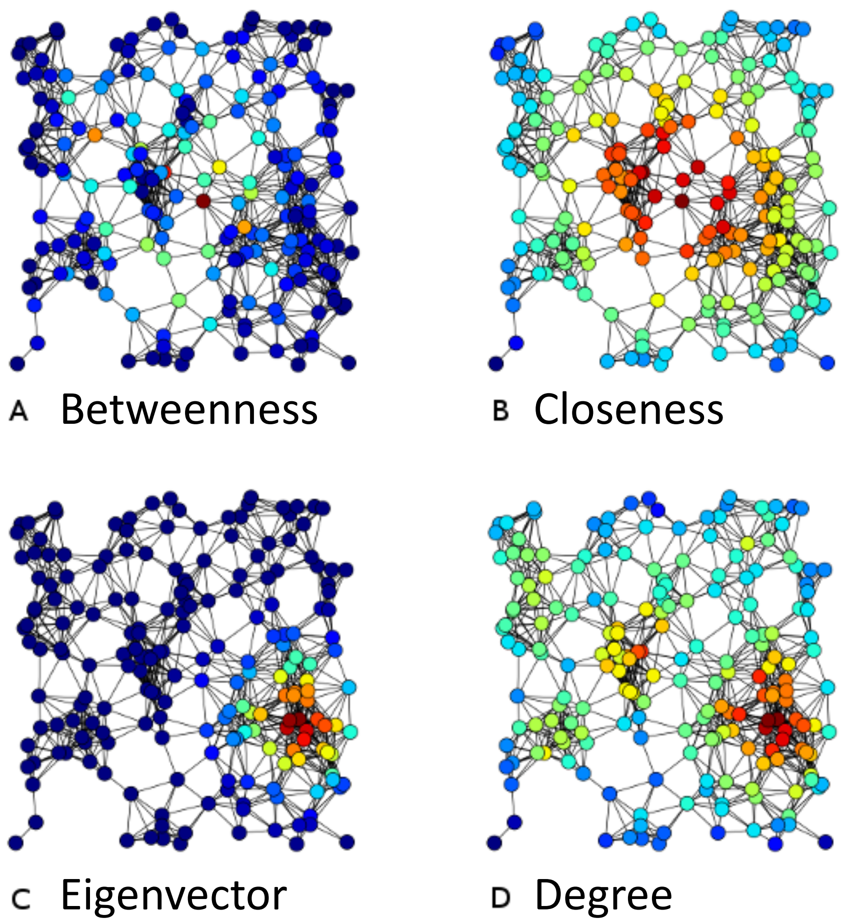


Image source: https://en.wikipedia.org/wiki/Centrality#/media/File:6_centrality_measures.png

Page Rank

Extension of Katz Centrality.

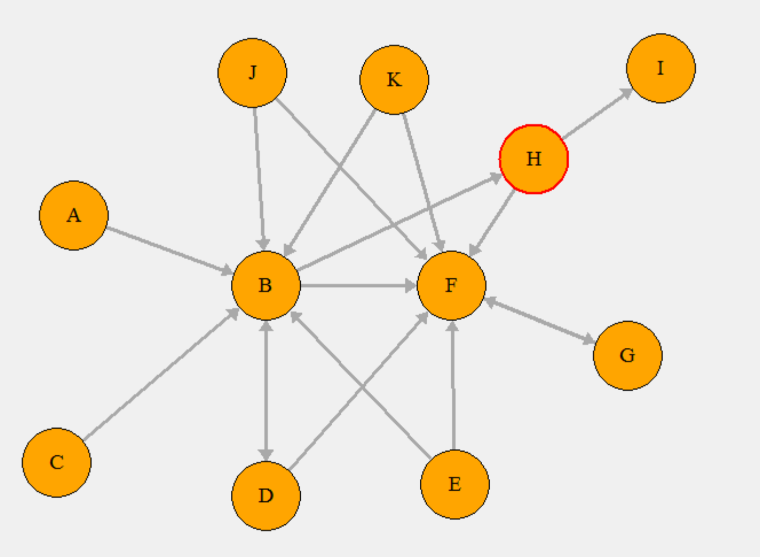
There are three distinct factors that determine the PageRank of a node: (i) the number of links it receives, (ii) the link propensity of the linkers, and (iii) the centrality of the linkers.

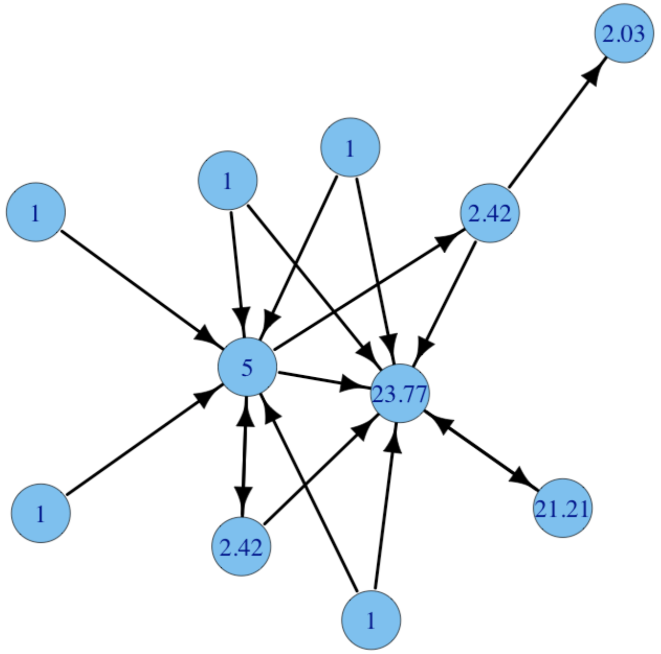
The first factor is not surprising: the more links a node attracts, the more important it is perceived. Reasonably, the value of the endorsement depreciates proportionally to the number of links given out by the endorsing node: links coming from parsimonious nodes are worthier than those emanated by spendthrift ones. Finally, not all nodes are created equal: links from important vertices are more valuable than those from obscure ones. This method has been coined (and patented) by Sergey Brin and Larry Page.

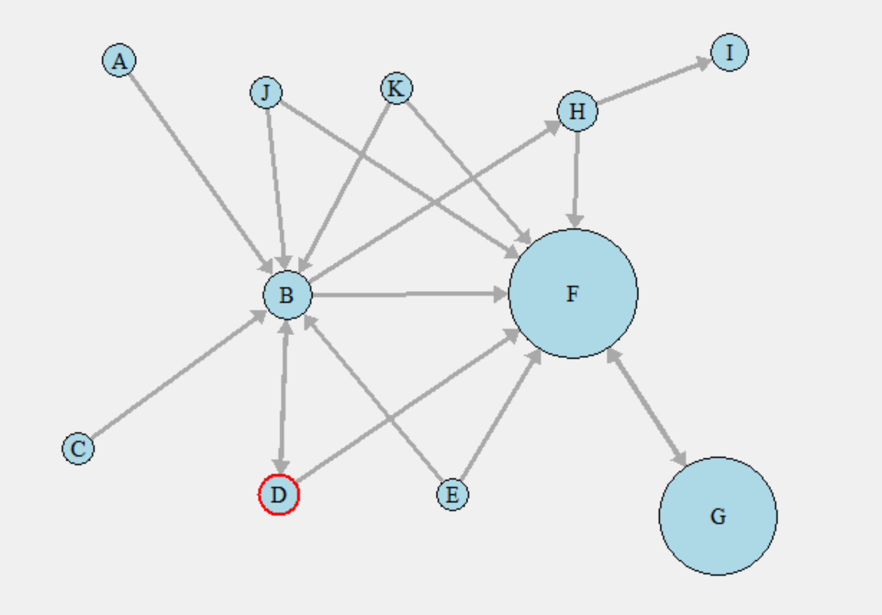
http://www.cs.princeton.edu/~chazelle/courses/BIB/pagerank.htm

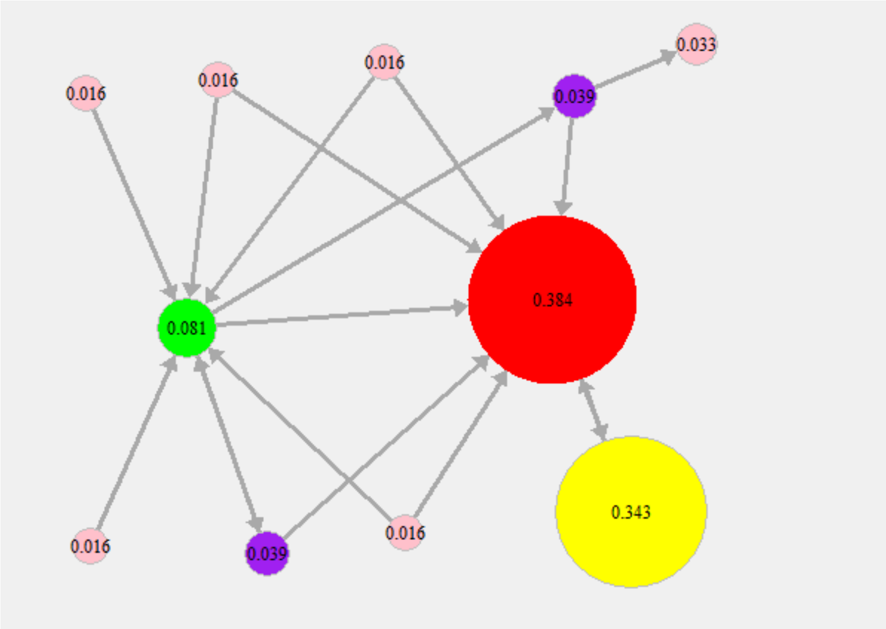
http://www.sci.unich.it/~francesc/teaching/network/pagerank

Degree Centrality	Betweenness Centrality	Closeness Centrality	Eigenvector Centrality	PageRank Centrality



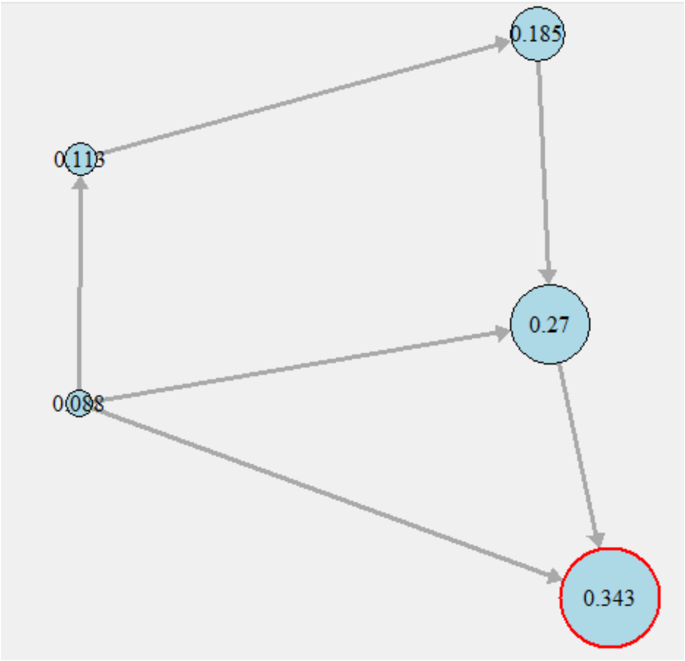


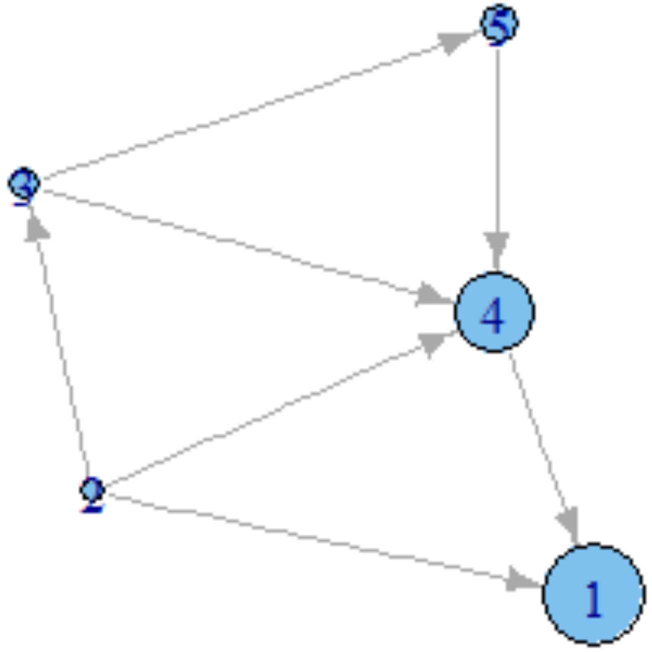




Page Rank

Simple graph with vertex size proportional to page.rank



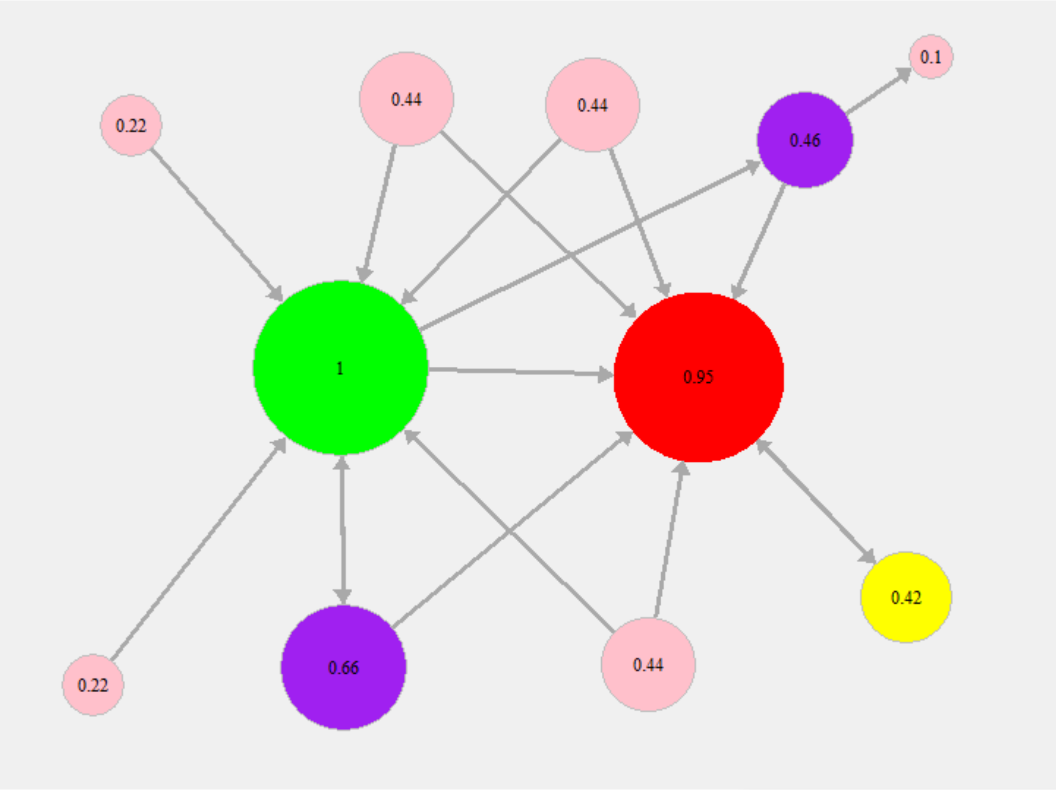


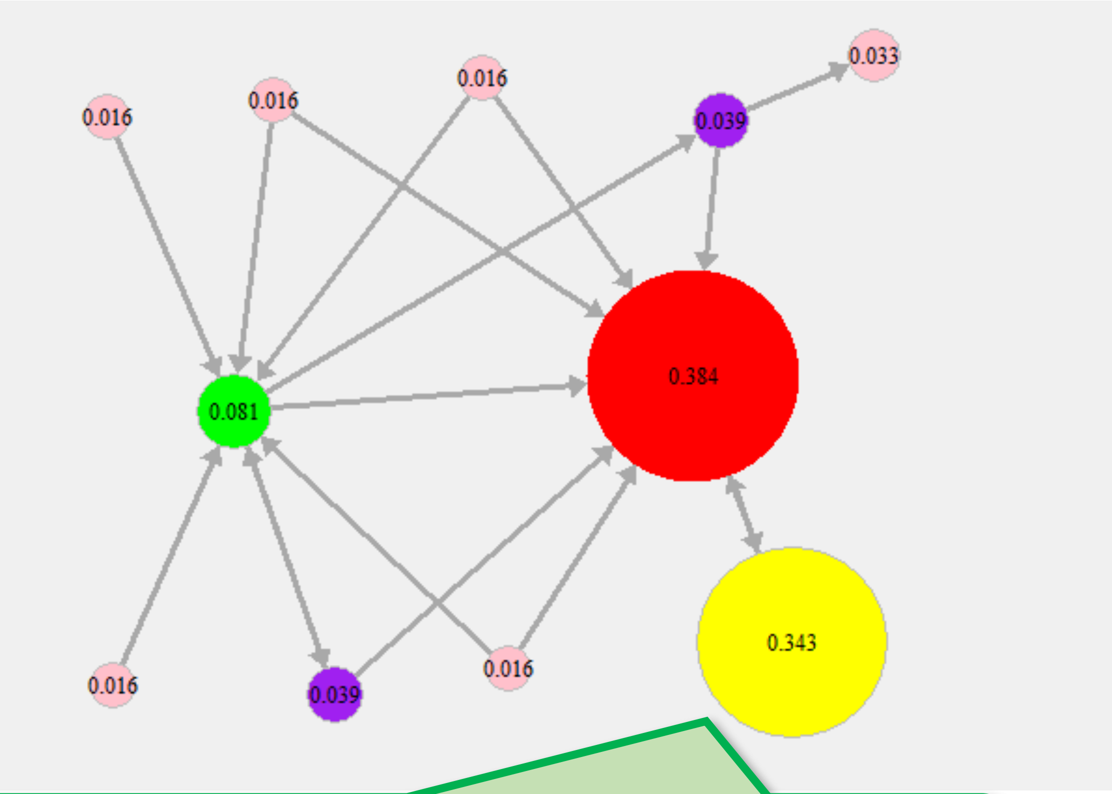

```
library(igraph)
pagerank_graph2 <- graph_from_literal(B-
+A,A-+E,E-+C,C-+D,B-+C,B-+D)
tkplot(pagerank_graph2)
tkplot(pagerank_graph2, edge.width=3,
       vertex.color="orange",
       vertex.label.color="black",
       vertex.label.cex=1.2,
       vertex.size=30)
pr<-page.rank(pagerank_graph2)$vector
pr<-round(pr,digits=3)
tkplot(pagerank_graph2, edge.width=3,
       vertex.color="lightblue",
       vertex.label.color="black",
       vertex.size=pr*100,
       vertex.label=pr)
```

http://blog.revolutionanalytics.com/2014/12/a-reproducible-r-example-finding-the-most-popular-packages-using-the-pagerank-algorithm.html

Eigenvector and PageRank compared

(node colors are the same in both the images)

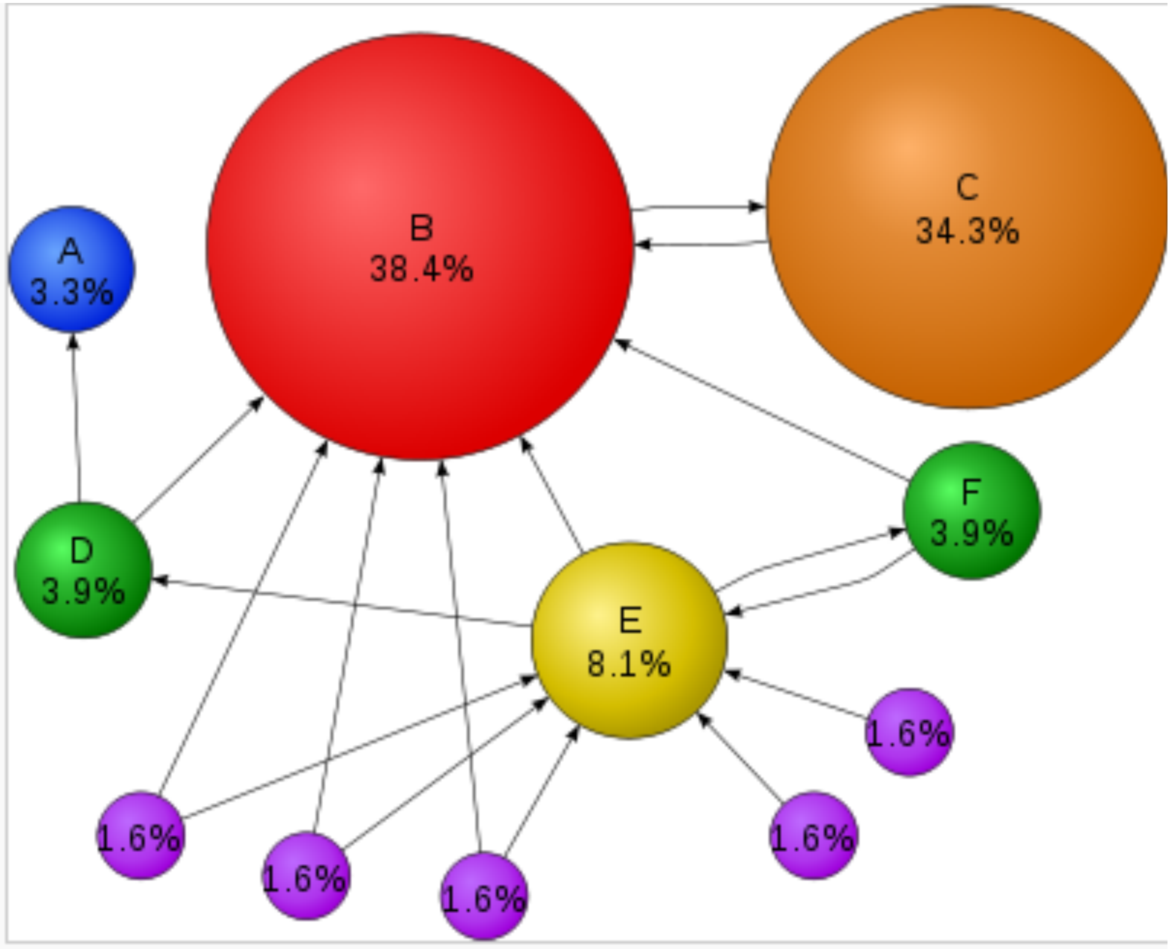




Getting a link from influential and spendthrift node!

Page Rank

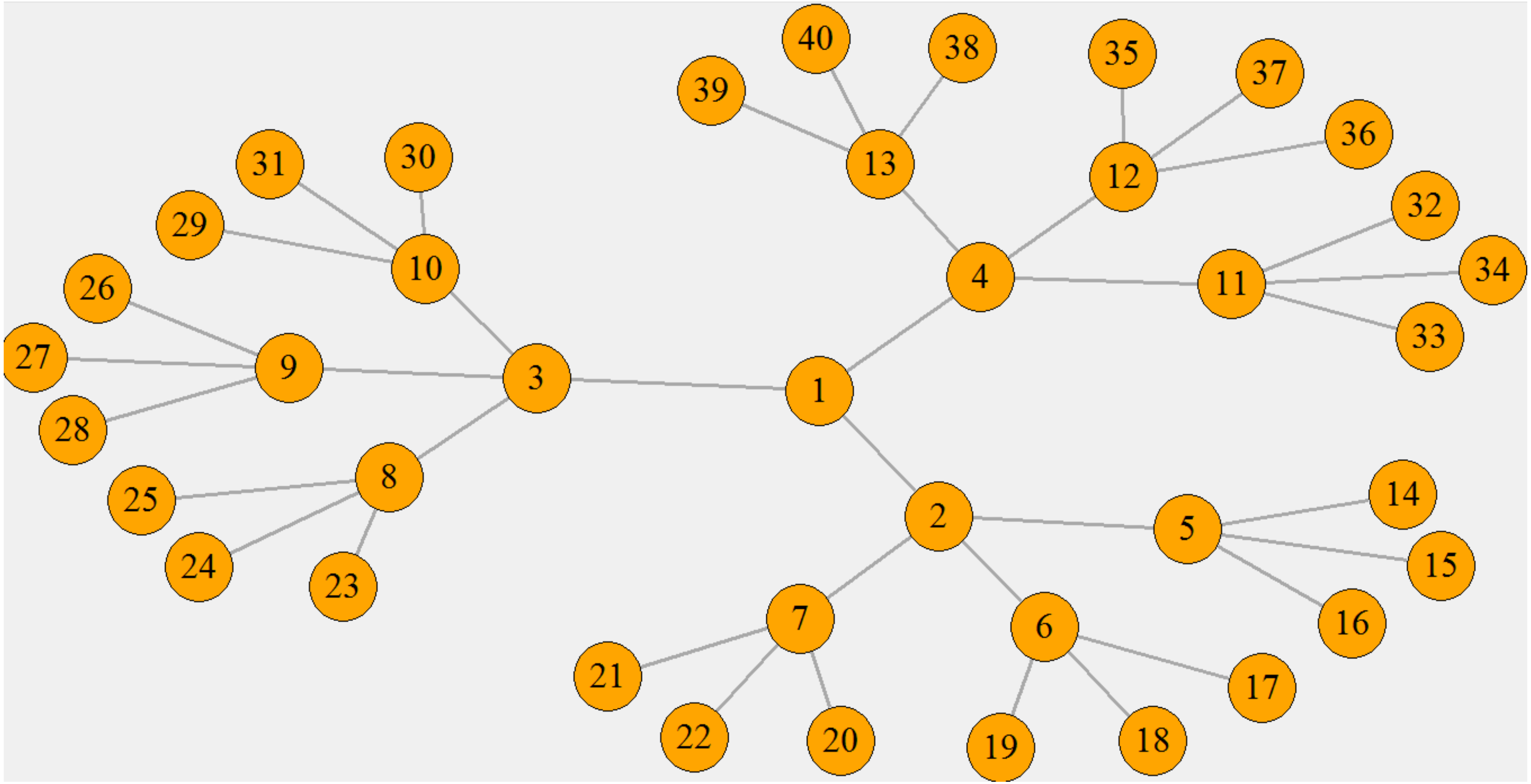
Mathematical **PageRanks** for a simple network, expressed as percentages. (Google uses a logarithmic scale.) Page **C** has a higher PageRank than Page **E**, even though there are fewer links to **C**; the one link to **C** comes from an important page and hence is of high value. If web surfers who start on a random page have an 85% likelihood of choosing a random link from the page they are currently visiting, and a 15% likelihood of jumping to a page chosen at random from the entire web, they will reach Page E 8.1% of the time. (The 15% likelihood of jumping to an arbitrary page corresponds to a damping factor of 85%.) Without damping, all web surfers would eventually end up on Pages **A**, **B**, or **C**, and all other pages would have PageRank zero. In the presence of damping, Page **A** effectively links to all pages in the web, even though it has no outgoing links of its own.



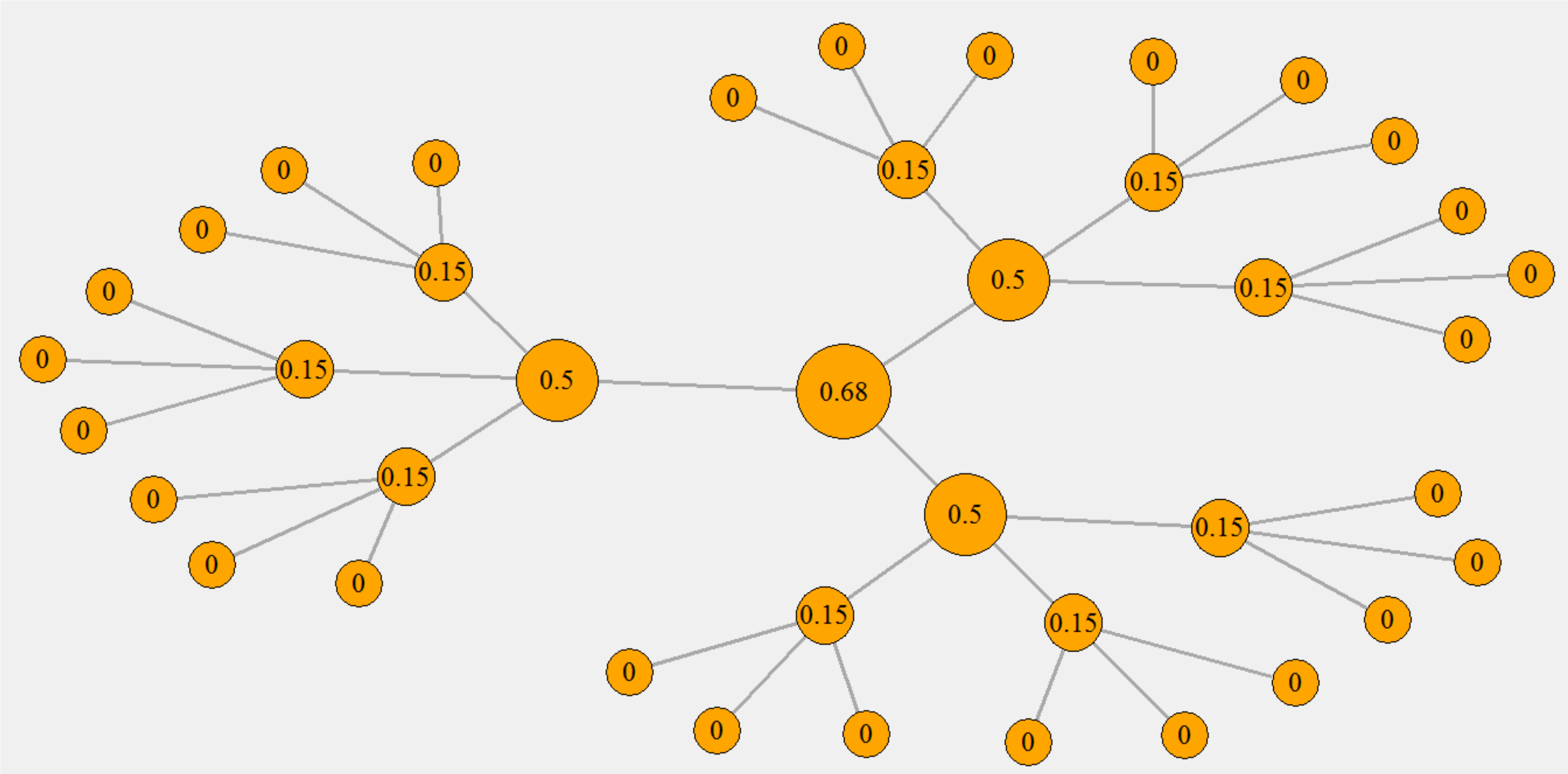
https://en.wikipedia.org/wiki/PageRank

Tree Graph

All nodes have degree centrality = 3; except for terminal nodes



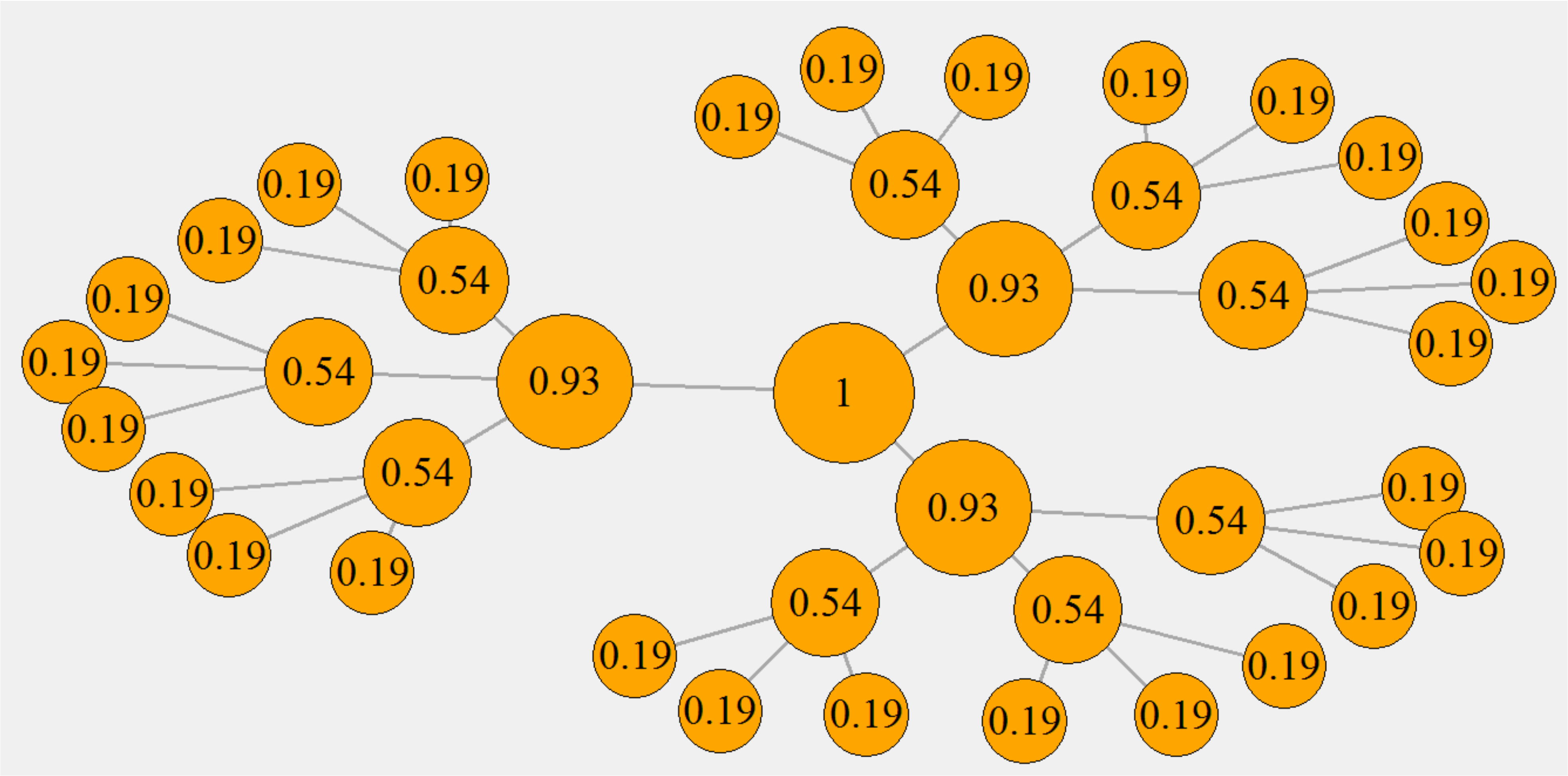
Betweenness Centrality (Normalized)



Moving away from the hub, “betweenness” lowers drastically, and becomes 0 at the terminal.

Interestingly, the hubs do not have absolute “betweenness”, (as shown by .68) but they are “closest” to every other node (as shown by closeness centrality of 1 on another slide).

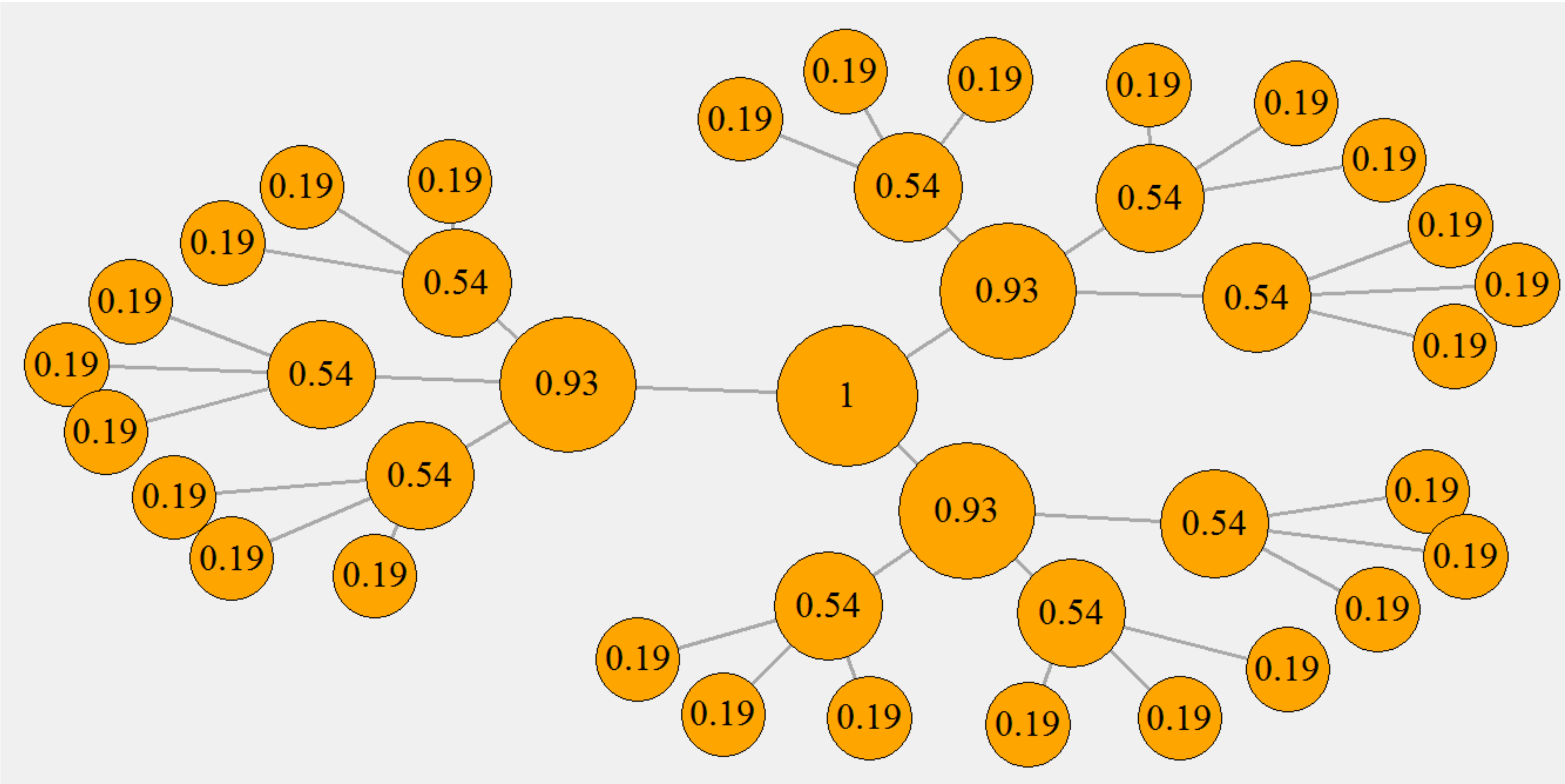
Closeness Centrality



Since all nodes are "closely" connected to each other (via central hub), there is not a huge variation in closeness centrality among the nodes.

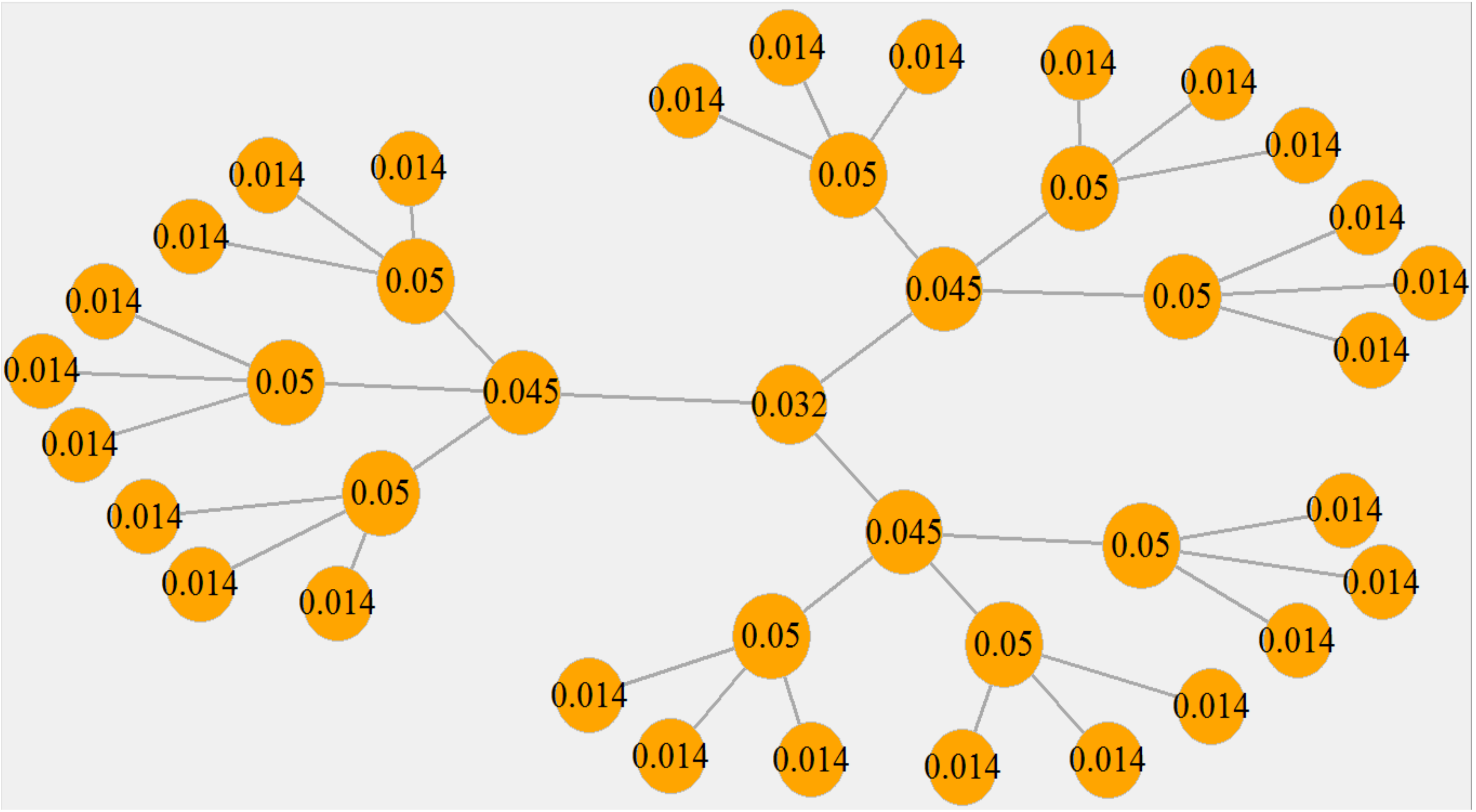
The hub-spoke system has lowered overall closeness for all nodes in general.

Eigenvector Centrality



As you go away from the hub, Eigenvector centrality lowers but the decline is much smoother since the influence wanes away slowly as compared to betweenness.

PageRank



Interestingly the hub here does not have the highest PageRank centrality value.

It is the secondary nodes that have higher PageRank value.

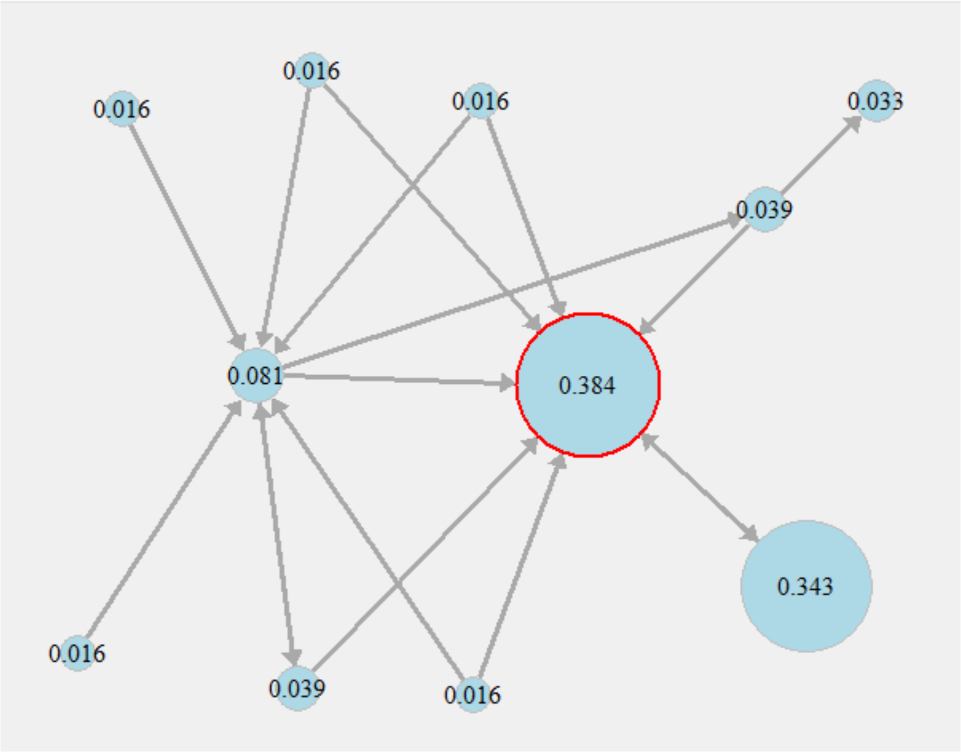
Perhaps number of links outweighs the effect of few influential links.

References

- M. Newmann. Networks. Oxford University Press. April 2010.
- http://www.slideshare.net/maksim2042/3-centrality
- https://www.cl.cam.ac.uk/teaching/1314/L109/stna-lecture3.pdf
- Barabasilab.com Class 11 (Fall 2016)
- http://djjr-courses.wikidot.com/soc180:eigenvector-centrality
- http://www.sci.unich.it/~francesc/teaching/network/pagerank
- http://www.cs.princeton.edu/~chazelle/courses/BIB/pagerank.htm

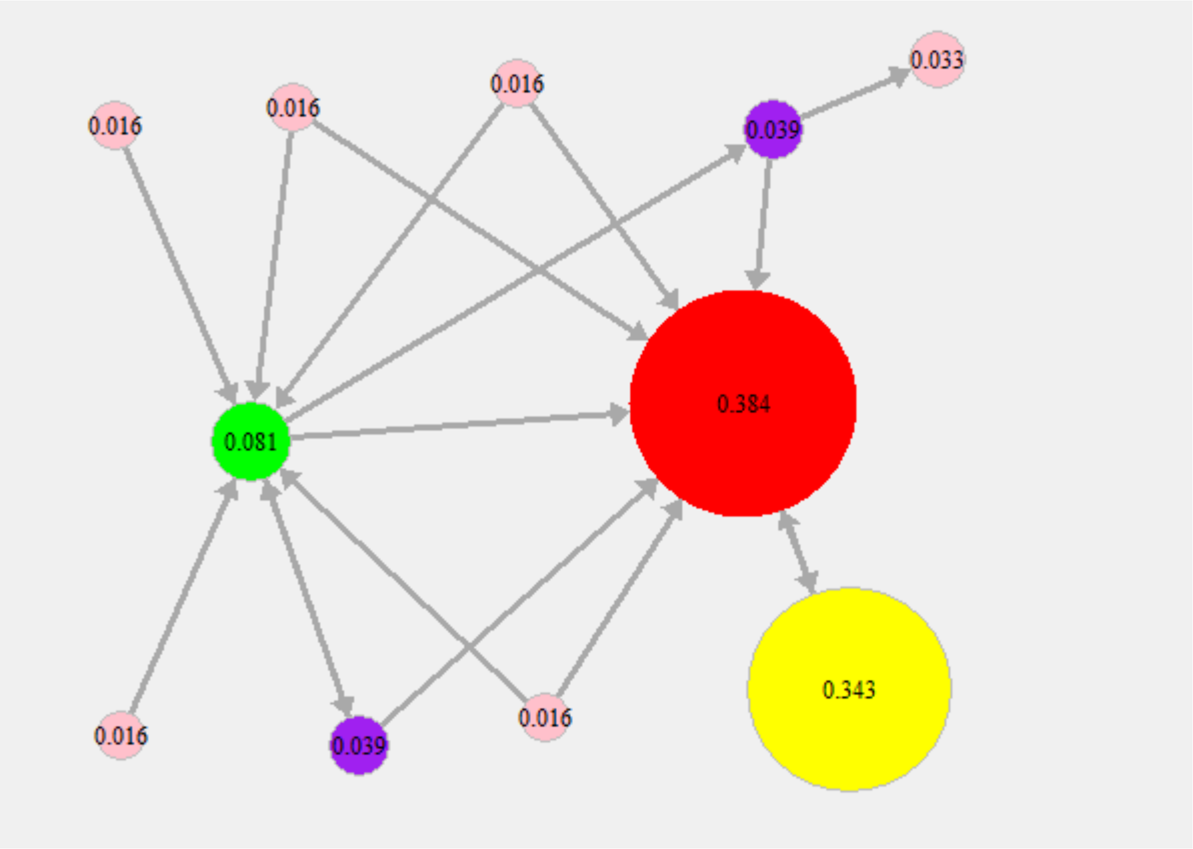
Questions? Thoughts? Visit the course's online discussion forum.

Appendix: PageRank



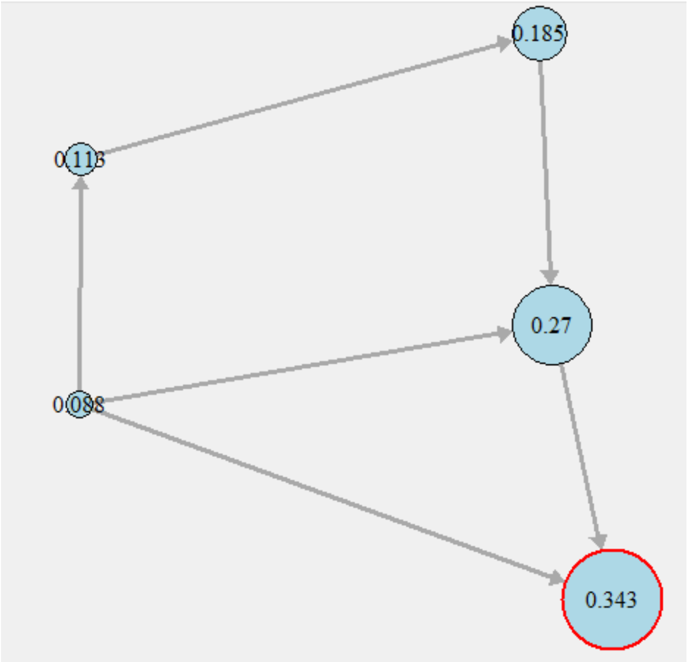

```
#page rank example
library(igraph)
pagerank_graph <- graph_from_literal(A-+B,C-+B,D-+F,D++B,E-+B,
                                     E-+F,B-+F,F++G,H-+F,H-+I,
                                     K-+B,K-+F,J-+B,J-+F,B-+H)
tkplot(pagerank_graph, edge.width=3,
       vertex.color="orange",
       vertex.label.color="black",
       vertex.size=30)
pr<-page_rank(pagerank_graph)$vector
pr<-round(pr,digits=3)
tkplot(pagerank_graph, edge.width=3,
       vertex.color="lightblue",
       vertex.label.color="black",
       vertex.frame.color="gray",
       vertex.size=pr*90+10,
       vertex.label=pr)
```

Appendix: PageRank




```
#page rank example
library(igraph)
pagerank_graph <- graph_from_literal(A-+B,C-+B,D-+F,D++B,E-+B,
                                     E-+F,B-+F,F++G,H-+F,H-+I,
                                     K-+B,K-+F,J-+B,J-+F,B-+H)
tkplot(pagerank_graph, edge.width=3,
       vertex.color="orange",
       vertex.label.color="black",
       vertex.size=30)
pr<-page_rank(pagerank_graph)$vector
pr<-round(pr,digits=3)
tkplot(pagerank_graph, edge.width=3,
       #vertex.color="lightblue",
       vertex.color=c("pink","green","pink","purple",
                      "red",
                      "pink",
                      "yellow",
                      "purple","pink",
                      "pink","pink"),
       vertex.label.color="black",
       vertex.frame.color="gray",
       vertex.size=pr*120+10,
       vertex.label.cex=.9,
       vertex.label=pr)
```

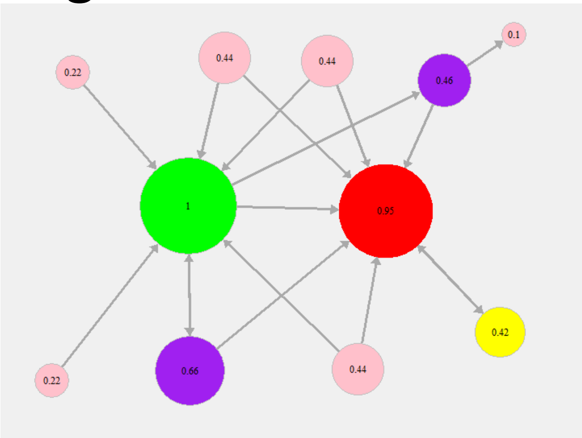
Appendix: PageRank



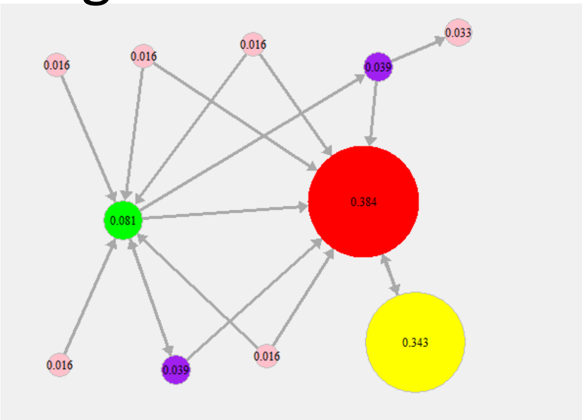

```
library(igraph)
pagerank_graph2 <- graph_from_literal(B-+A,A-+E,E-+C,C-+D,B-+C,B-+D)
tkplot(pagerank_graph2)
tkplot(pagerank_graph2, edge.width=3,
       vertex.color="orange",
       vertex.label.color="black",
       vertex.label.cex=1.2,
       vertex.size=30)
pr<-page.rank(pagerank_graph2)$vector
pr<-round(pr,digits=3)
tkplot(pagerank_graph2, edge.width=3,
       vertex.color="lightblue",
       vertex.label.color="black",
       vertex.size=pr*100,
       vertex.label=pr)
```

Appendix: PageRank and Eigenvalue

Eigenvalue



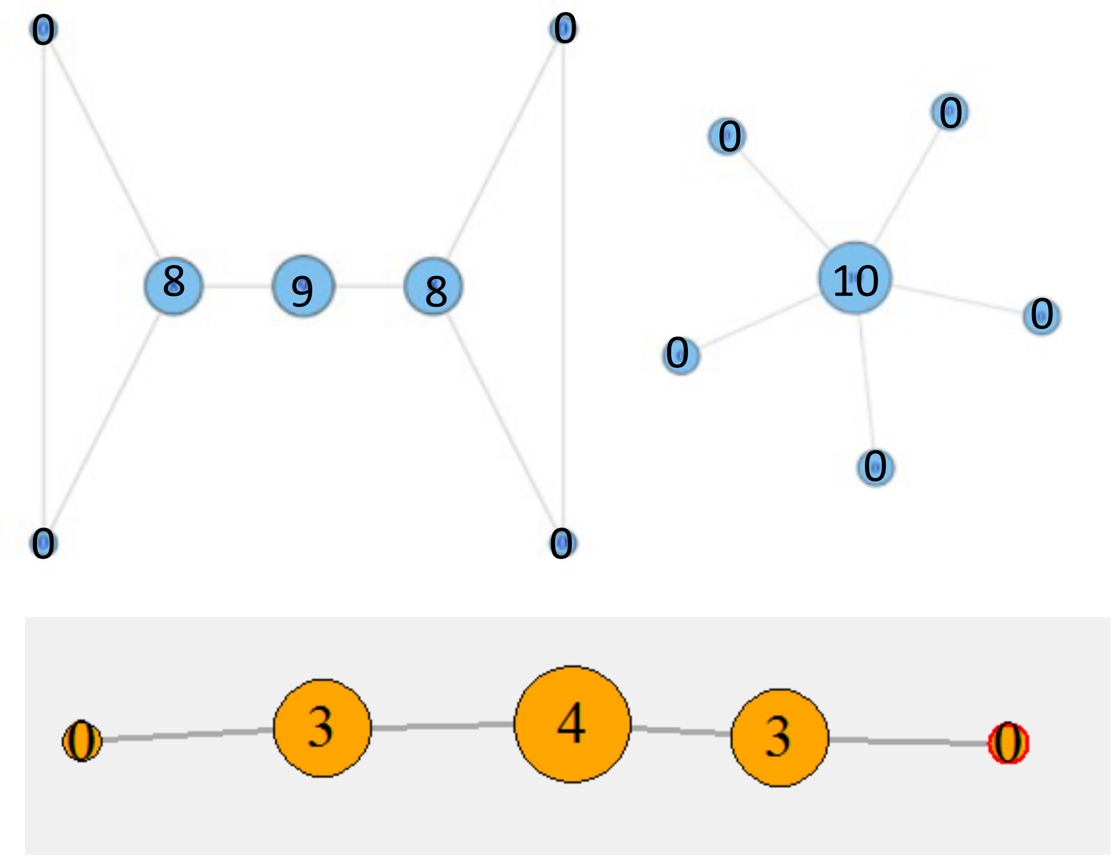
PageRank



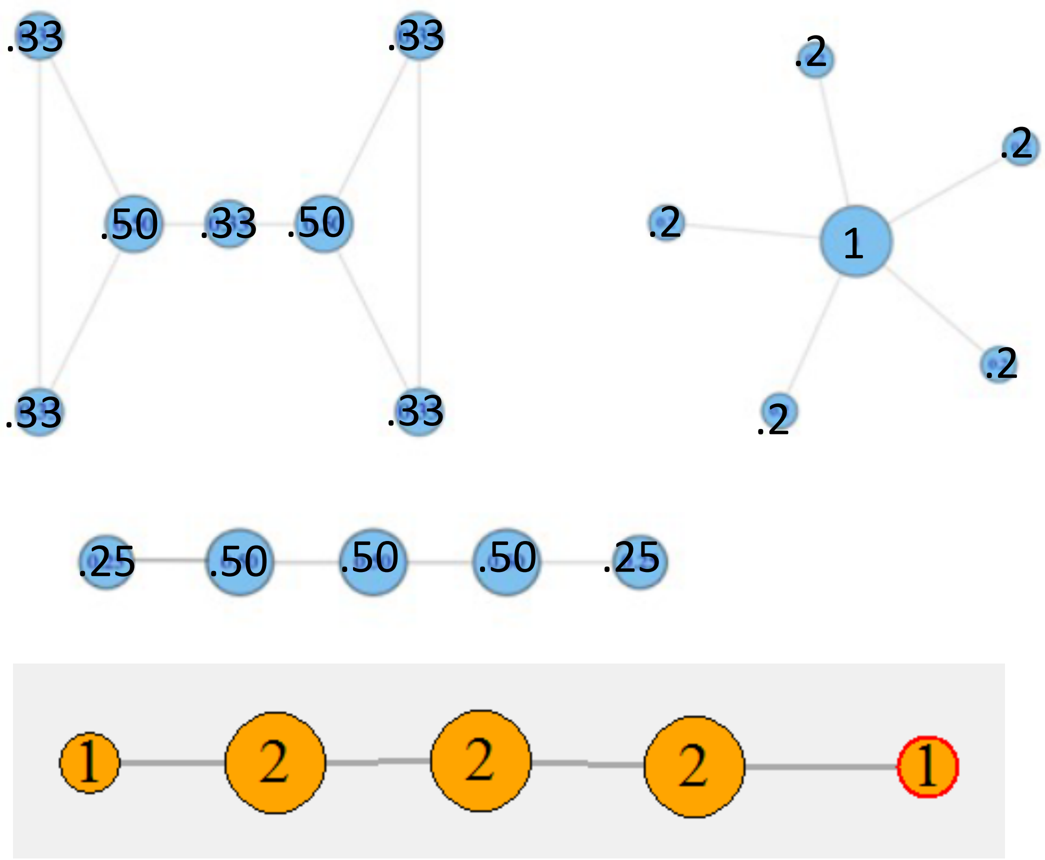

```
#page rank example
library(igraph)
pagerank_graph <- graph_from_literal(A-+B,C-+B,D-+F,D++B,E-+B,
                                     E-+F,B-+F,F++G,H-+F,H-+I,
                                     K-+B,K-+F,J-+B,J-+F,B-+H)
tkplot(pagerank_graph, edge.width=3,
       vertex.color="orange",
       vertex.label.color="black",
       vertex.size=30)
pr<-page_rank(pagerank_graph)$vector
pr<-round(pr,digits=3)
tkplot(pagerank_graph, edge.width=3,
       #vertex.color="lightblue",
       vertex.color=c("pink","green","pink","purple",
                      "red",
                      "pink",
                      "yellow",
                      "purple","pink",
                      "pink","pink"),
       vertex.label.color="black",
       vertex.frame.color="gray",
       vertex.size=pr*120+10,
       vertex.label.cex=.9,
       vertex.label=pr)
ei<-centr_eigen(pagerank_graph, scale = TRUE,
                options = arpack_defaults, normalized = TRUE)$vector
ei<-round(ei,digits=2)
tkplot(pagerank_graph, edge.width=3,
       #vertex.color="lightblue",
       vertex.color=c("pink","green","pink","purple",
                      "red",
                      "pink",
                      "yellow",
                      "purple","pink",
                      "pink","pink"),
       vertex.label.color="black",
       vertex.frame.color="gray",
       vertex.size=ei*50+10,
       vertex.label.cex=.9,
       vertex.label=ei)
```

Appendix: Betweenness Centrality

Betweenness centrality (not normalized)



Degree centrality

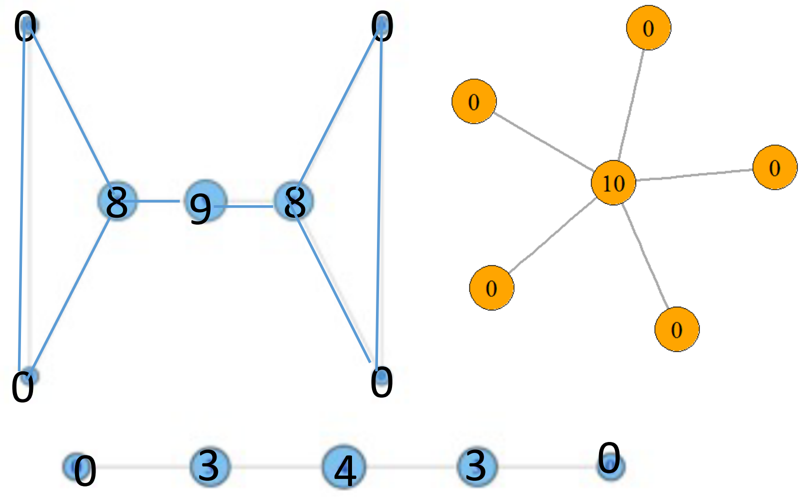


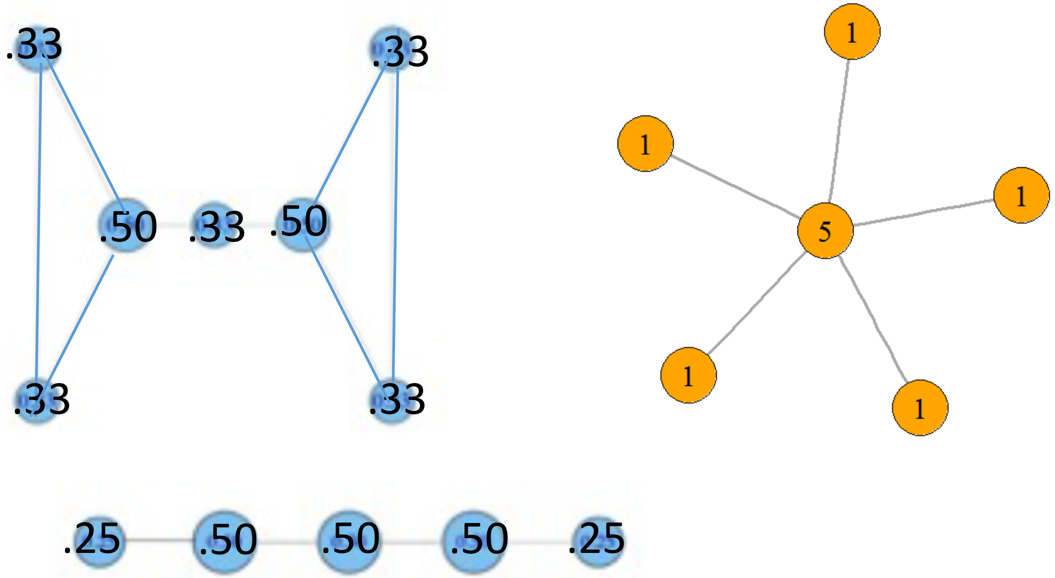
Appendix: Closeness Centrality

Closeness Centrality

Degree centrality

Betweenness centrality (not normalized)

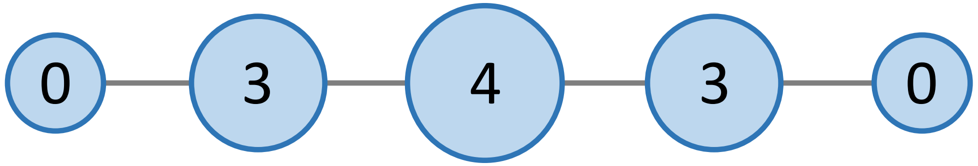




Appendix: Closeness Centrality

$$\tilde{C}^C(i) = \left[\sum_{j=1}^{N} d(i,j)\right]^{-1}$$

Closeness is based on the length of the average shortest path between a vertex and all vertices in the graph.



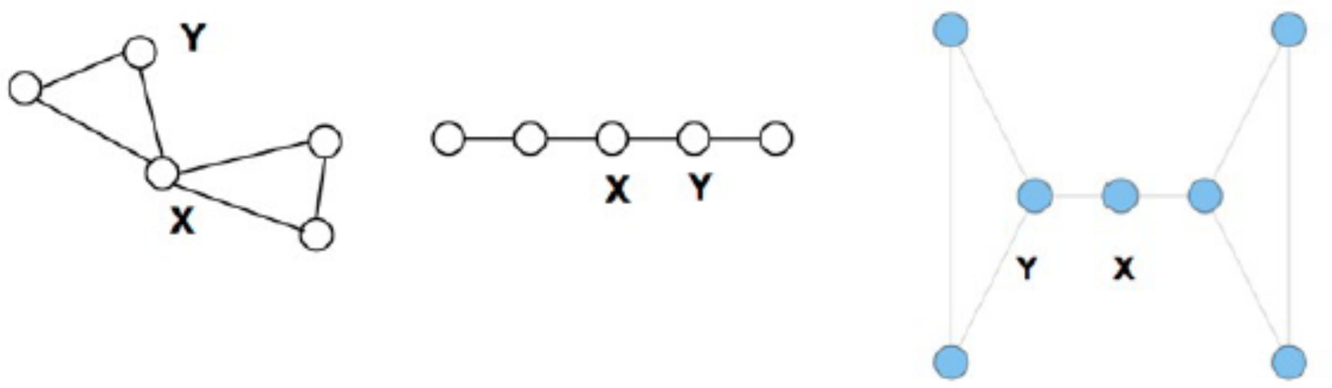
Normalized

$$C^C(i) = \frac{\tilde{C}^C(i)}{\boxed{N-1}}$$

All other nodes in the network

Source: M. E. J. Newman. (2010). *Networks: An Introduction*. Oxford University Press.

Appendix: Betweenness Centrality



$$\tilde{C}^B(i) = \sum_{j<k} \frac{d_{jk}(i)}{d_{jk}}$$

d_{jk} = # of shortest paths between *j* and *k*

$d_{jk}(i)$ = # of shortest paths between *j* and *k* that go through *i*

Normalized

$$C^B(i) = \frac{\tilde{C}^B}{\boxed{(N-1)(N-2)/2}}$$

Number of pairs of vertices excluding *i*

Source: M. E. J. Newman. (2010). *Networks: An Introduction*. Oxford University Press.