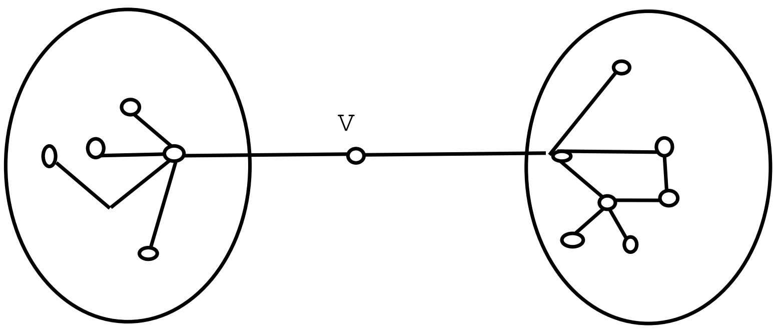
Social Networks Theory: Node Removal, Flow-Betweenness, Eigenvector Centrality, Page Rank Algorithm of Google -10.02.2006

Sandeep Karajgaonkar (05CS6018)

April 22, 2006

1 Node Removal

In centrality we mainly try to find out how important node is in network such as Betweenness centrality. Degree of a node doesnt alone does not determine the nodes importance in the network.



As shown in above figure the node might be acting as a bridge between two clusters or regions of a network and certainly has more importance than others.

Now what if we removed one or more nodes from the network?
Is it going to affect the network? What could be possible effects?

If we removed one or more nodes form the network certainly it is going to affect out network. As it is going to change its structure and can have considerable effects on the network.

- First effect that can be seen is increase in the geodesic path (shortest path between any two nodes).
- In major cases if the removed node was acting as bridge then, it causes the geodesic path to be infinity and makes network disconnected.

We can easily understand the impact of removal of the nodes with high betweenness if some of the day to day life network such as Traffic network and information Network.

2 Flow Betweenness

Centrality gives the importance of a node in the network. So the node with high centrality has more importance in the network.

Many times such node can behaves as a reluctant broker and do not allowing the two other nodes of different regions to get relationship established.

To tackle with this situation these two nodes must find the other ways to do so, which may be costly. This provides the main idea of flow betweenness.

Thus in flow betweenness we takes all paths in accounts along with all shortest paths between two nodes.

Thus the flow betweenness of a node v can be calculated as follows-

$$fb(v) = \sum_{s \neq v \neq t} \frac{path_{st}(v)}{path_{st}} \tag{1}$$

Where fb(v) = flow betwenness of a node.
$Path_{st}$(v) = no of paths between s and t which goes through v
$path_{st}$ = no of paths between s and t

The main problem with flow betweenness is that- for a large network it is not computationally feasible (tractable) to calculate as we have to consider all the paths between two nodes.

3 Eigenvector centrality

Eigenvector centrality in terms gives the major of Popularity. Which says that persons popularity or power is not just how many people knows him but also how many peoples knows the peoples who knows me and it is a recursive one.

This idea is used in estimating the transmission of HIV.

Definition: Eigenvector centrality is defines as the principle eigenvector of the adjacency matrix.

Eigenvector of any symmetric matrix A is a vector e such that

$$e_i = \lambda^{-1} \sum_j a_i j e_j \tag{2}$$

Where λ is a constant and ei is the centrality of the node i.

Thus the centrality of a node is proportional to the centrality of nodes it is connected and which is recursive.

Finding Eigenvector centrality

Hotelling proposed a power accelerated method as follows-

Step 0: Set e_i =1 for all i.

Step 1: Compute ${e_i}^* = \sum_j a_{ij} e_j$

Step 2: Set λ equal to the square root of the sum of squares of e^*.

Step 3: Set $e_i = e_i/\lambda$ for all i

Step 4: Repeat step 1 to 3 until λ stops changing.

From above it is clear that after execution of step one fro the first time e* is equal to the degree of vertices.

As step 4 states we repeat steps 1 to 3 unless converges i.e. stops changing.

4 PageRank Algorithm of Google

It is originally developed by Larry Page to give ranking of pages. Page Rank is a numeric value that represents how important a page is on the web, which is calculated by analyzing the link (hyperlink) structure of the web.
One page links (hyperlink) to another page, it is effectively casting a vote for the other page. Thus more votes that are cast for a page, the more important the page must be and this is recursive one. So that the pages linked with the pages with high PageRank (higher eigenvector centrality) has a higher PageRank.

Google has not disclosed his page ranking formula but mostly used formula is as follows-

$$PR(A) = (1 - d) + d(\mathrm{PR(t1)/C(t1)+...+PR(tn)/C(tn)})$$

Where "t1 - tn" are pages linking to page A
"C" is the number of outbound links that a page has
"d" is a damping factor, usually set to 0.85.

bove calculated rank is static rank i.e. it is once calculated periodically and same is used unless next time it is explicitly calculated to take care on changed link structure. Normally Google recalculates its PageRank on monthly basis by doing crawling of the web structure.