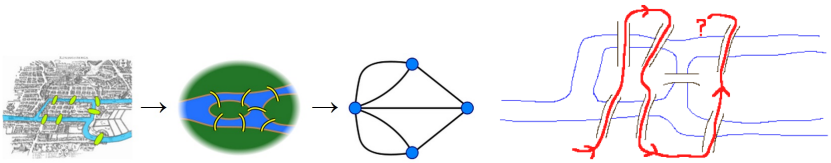
Chapter 1

Basics of Graph theory

1.1 Eulerian Beginning

Problem : Find a way to walk about the city so as to cross each of the 7 bridges exactly once and then return to the starting point : Eulerian path . Its negative resolution by Leonhard Euler in 1735 laid the foundations of graph theory and prefigured the idea of topology.



1.2 Graphs : A Set-Theoretic Definition

A graph G consists of an ordered tuple $G = (V, E)$, where V is a set of nodes, points, or vertices; E is a set whose elements are known as edges or lines. $E \subseteq V \times V$.If E equals $V \times V$, then the graph is complete.

1.3 Adjacency Matrix and List

A graph can be represented in the following two ways: **adjacency matrix** and **adjacency list**. Properties of an adjacency matrix A : $A = \{a_{ij}\}$, where i and j are nodes, and $a_{ij} = 1$ if there is an edge between i and j, else it is 0. The entries of the matrix A^2 denote the number of paths of length 2 between nodes in the graph. Similarly, entries of A^n denotes the number of paths of length n. Note that the trace (sum of the diagonal elements) of the matrix A^3 is equal to the number of triangles in the graph.

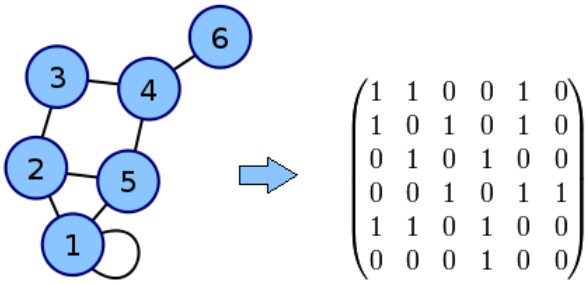


Figure 1.1: Adjacency matrix representation

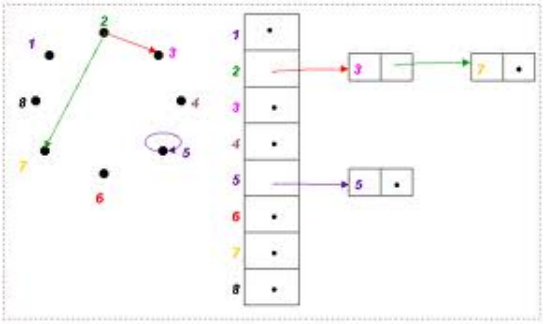


Figure 1.2: Adjacency List Representation

1.4 Paths, Walks and Trails

Def :A **path** in a graph is a single vertex or an ordered list of distinct vertices $v_i....v_k$ such that $v_{i-1}v_i$ is an edge for all $2 \leq i \leq k$. No vertex may be repeated.

Def :A **walk** of length k is a sequence $v_0, v_1, \cdots, v_k$ of vertices and edges such that (v_{i-1}, v_i) is an edge for $1 \leq i \leq k$.

Def :A **trail** is a walk with no repeated edge.

1.5 Components of a Graph

Let $G = (V, E)$ be an undirected graph. G is said to be connected if there exists a path between any two distinct vertices of G.

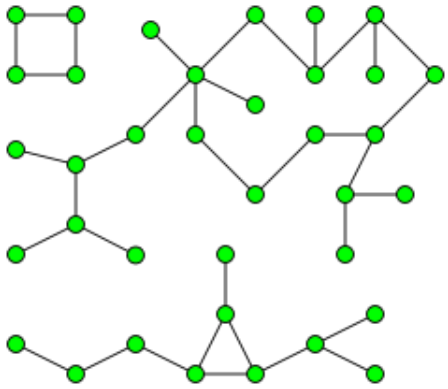


Figure 1.3: A graph with 3 components

1.6 Complete and Complement Graphs

A complete graph is one in which an edge exists between any 2 vertices, that is, all the entries in the adjancency matrix are 1.

A complete graph with n vertices is denoted as K_n. The complement graph G' of a graph G is a graph such that $V(G') = V(G)$, and an edge exists between 2 nodes v_i, v_j in G' if there exists no edge between them in G.

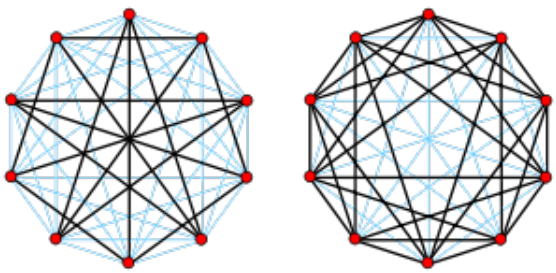


Figure 1.4: The Petersen graph on the left, and its complement graph on the right

1.7 Sparse and Dense Graphs

A graph $G(V, E)$ is called **sparse** if $|E| \approx |V|$; and it is called **dense** if $|E| \approx |V|^2$.

1.8 Planar Graphs

A graph G is called planar if G can be drawn in the plane with its edges intersecting only at vertices of G. Such a drawing of G is called an **embedding** of G in the plane. Note that of all complete graphs K_n, only K_1, K_2, K_3 and K_4 are planar.

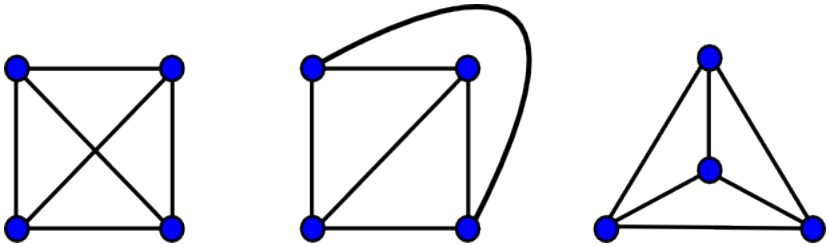


Figure 1.5: The graph K_4 (extreme left) and its planar embeddings

1.9 Regular Graphs and Lattices

A **regular** graph is one in which all the nodes have the same **degree**, that is, the same number of edges emanating from the node.A **lattice** is a regular graph with vertices coupled to their k nearest neighbours.

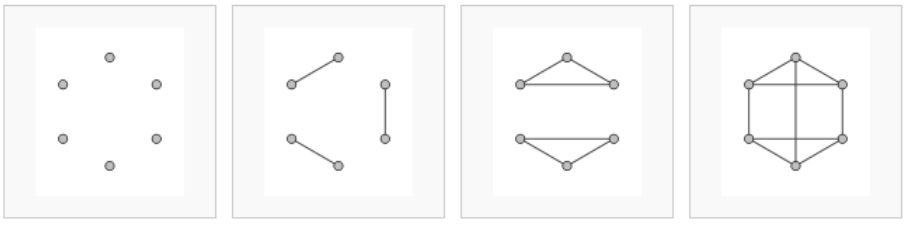


Figure 1.6: 0-regular, 1-regular, 2-regular and 3-regular graphs

1.10 Geodesics

A **geodesic** from vertex a to vertex b is a path of minimum length between the nodes. The length of this path is called the **geodesic distance** between a and b.

The eccentricity of a vertex v is the greatest geodesic distance between v and any other vertex. The largest eccentricity of any vertex in the graph is called the **diameter** (d) of the graph. The **radius** (r) of a graph is the minimum eccentricity of any vertex.
A **central** vertex in a graph of radius r is one whose distance from every other vertex in the graph is at most r.

A **peripheral** vertex in a graph of diameter d is one that is at a distance d from some other vertex in the graph.

1.11 Average Path Length

The average path length l is defined as the average of the shortest paths between all nodes in the network, i.e.,

$$l = \langle d_{ij} \rangle = \frac{1}{N(N-1)} \sum_{i \neq j} d_{ij}$$

If the graph is disconnected, it makes sense to consider the reciprocal of the harmonic mean; this is because the distance between two nodes belonging to separate components is infinite, the reciprocal being 0.

$$l = \langle \frac{1}{d_{ij}^{-1}} \rangle = \left(\frac{1}{N(N-1)} \sum_{i \neq j} d_{ij} \right)^{-1}$$

1.12 Cut Points

A **cut point** is a vertex whose removal increases the number of components in the graph. Such points are called **brokers** in social networks. Removal of brokers creates communities that are totally isolated from each other.

1.13 Bridges

An edge is called a **bridge** if its removal increases the number of components in the graph.

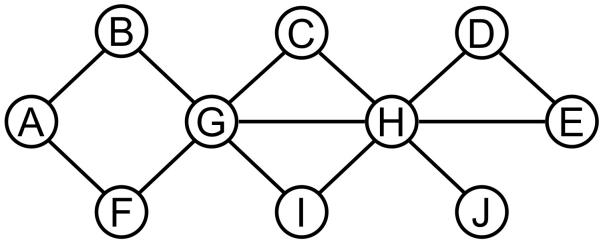
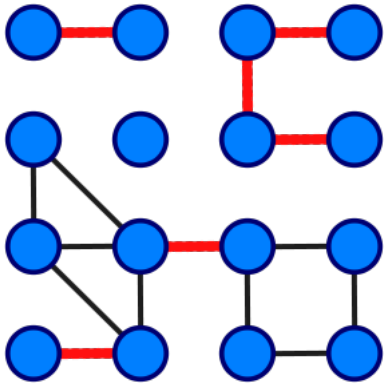


Figure 1.7: Nodes G and H are cut points



1.14 Connection Density

The **connection density** in a graph is defined as the ratio of the number of edges actually present in the graph and the maximum number of edges possible.

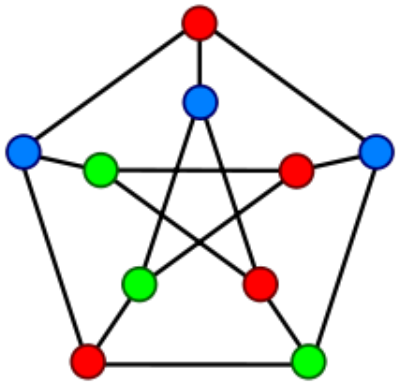
$$cd = \frac{|E|}{\binom{n}{2}} = \frac{2|E|}{N(N-1)}$$

It can also be thought of as the probability of existence of an edge between a randomly chosen pair of vertices.

1.15 Connected Components

A **strongly connected** directed graph is one where each node belonging to the graph can be reached from every other node via directed paths. A **weakly connected** directed graph is one where each node belonging to the graph can be reached from every other node, disregarding edge directions.

1.16 Chromatic Number



A **proper colouring** of a graph is an assignment of labels to each vertex of the graph such that no two adjacent vertices receive the same label. The **chromatic number** of a graph is the minimum number of colours required to achieve a proper colouring.

1.17 Chordal Graphs

A graph is **chordal** if each of its cycles of four or more nodes has a chord, which is an edge joining two nodes that are not adjacent in the cycle.

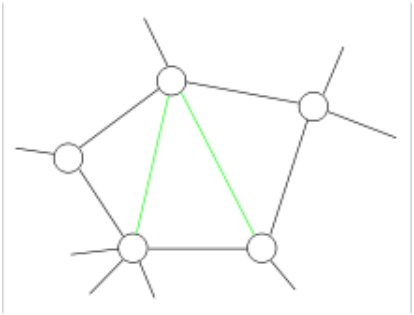


Figure 1.8: A cycle (black) with two chords (green)

Chapter 2

Metrics of Network

2.1 Eccentricity,Radius and Diameter

Eccentricity of a vertex v in a graph is the greatest geodesic(shortest) distance between v and any other vertex in the graph. **Radius** of a graph is the minimum eccentricity of any vertex similarly **Diameter** is the maximum eccentricity of any vertex in the graph.

2.2 Citation Network

A research paper always refers to earlier works on the related or similar topics . This reference is called citation. We can form a network with research papers as nodes and the citations as edges . Such a network is a citation network. As a research paper can only cite a paper which has been published previously hence the edges always point backward. It also follows that such a network is also acyclic as no forward edges are possible.

Alfred Lotka(1926) studied citation networks and concluded that: the number of scientists who have k citations falls off as $k^{-\alpha}$ for some constant α.

2.3 Degree Distribution

2.3.1 Definition

Let p_k is the probability that a vertex chosen uniformly at random has a degree k. Hence p_k is basically the fraction of vertices having degree k. The plot of k versus p_k is called the degree distribution of the network. What Lotka observed for citation network is true for most real world networks - p_k varies as $k^{-\alpha}$. That means the distribution is right skewed.

$$P_k \alpha k^{-\alpha}$$

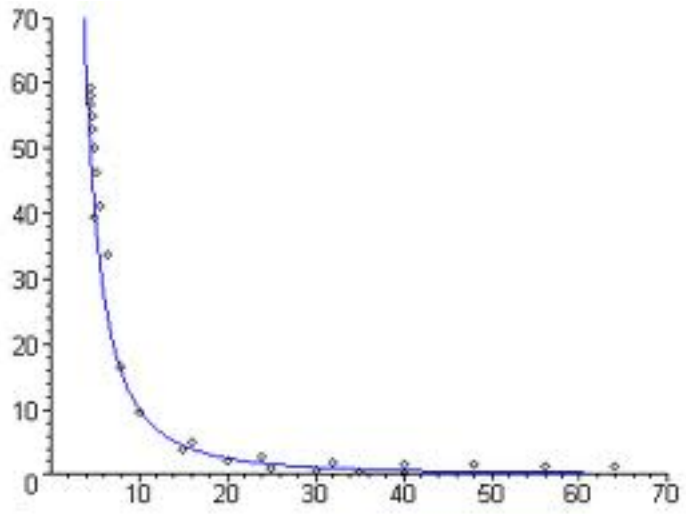


Figure 2.1: An example power-law graph, being used to demonstrate ranking of popularity. To the right is the long tail, and to the left are the few that dominate (also known as the 80-20 rule)

The network has very few nodes of high degree and large number of nodes of low degree. Due to noisy and insufficient data sometimes the definition is slightly modified . It is defined as the probability that a node has degree greater than or equal to k. (cumulative distribution is plotted.)

$P_k = \sum_{k'=k}^{\infty} p_{k'}$

for discrete distributions, while for continuous distributions we have

$P_k = \int_{k'=k}^{\infty} p_{k'} dk'$

So, P_k can also be interpreted as the probability that the degree of a node selected uniformly at random is greater than or equal to k.

2.3.2 Scale-Free Functions

A scale-free function $f(x)$is one in which the independent variable x when rescaled does not affect the functional form of the original function. Mathematically,

$f(ax) = bf(x)$

Power Laws are scale-free functions, that is, at any scale, they still show power law behaviour. Other examples where such behaviour is manifested include fractals.

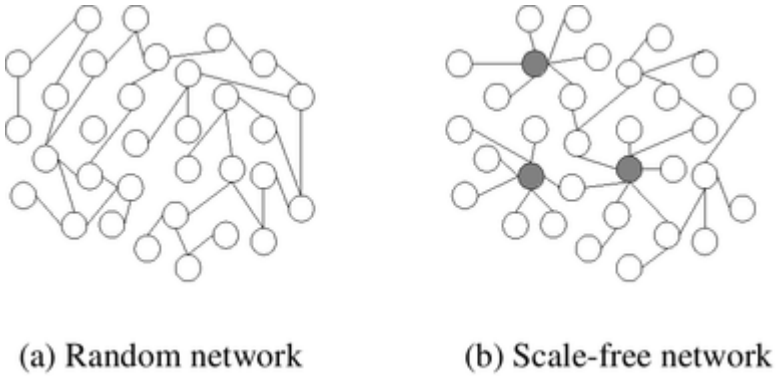


Figure 2.2: Random versus Scale-free network

2.4 Clustering Coefficient

The **clustering coefficient** for a vertex v in a network is defined as the ratio between the total number of connections among the neighbors of v to the total number of possible connections between the neighbours. Mathematically, $C_v = \frac{L}{\binom{n}{2}}$ where L = the number of actual links between the neighbours of v, and n = the number of neighbours of v.

The **clustering index** of the whole network is the average of the clustering coefficients of all the vertices. That is, $C = \frac{1}{N}\sum C_v$ Note that higher the clustering index, larger the number of triangles in the network.

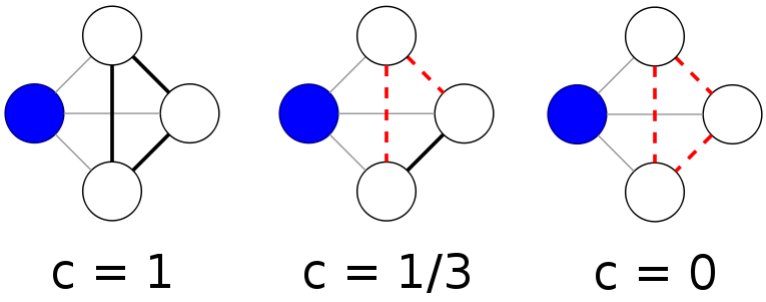


Figure 2.3: Local clustering coefficient values

2.5 Centrality

Centrality is a measure indicating the *importance* of node in the network. Commonly, it measures the 4 P's - prestige, prominence, (im)portance and

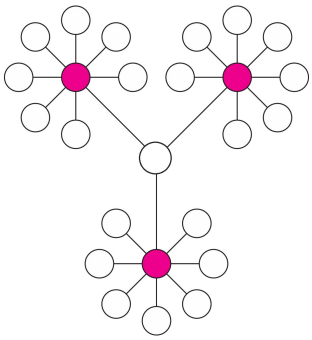


Figure 2.4: The most important vertices according to degree-centrality (red)

power. We wll get a better idea of what is meant by importance as the section progresses.

2.5.1 Degree Centrality

Degree centrality is defined as the ratio of the number of neighbours of a vertex with the total number of neighbours possible. Mathematically, Degree Centrality = $\frac{k}{N-1}$ where k is the degree of the vertex, and N is the total number of nodes in the network.

The variance of the distribution of degree centrality in a network gives us the **centralization** of the network. One can see that a *star network* is an ideal centralized network, whereas a *line network* is less centralized.

Figure 2.5: Star Network and Line Network

2.5.2 Betweenness Centrality

The degree of a node is not the only measure of the importance of a node in the network, and this centrality measure addresses this fact. This concept was introduced by Linton Freeman. In his conception, vertices that have a high probability of occuring on a randomly chosen shortest path between two nodes are said to have high **betweenness centrality**.

Formally, centrality of a vertex v is defined as the summation of the geodesic

path between any two nodes s and t via v, expressed as a fraction of the total number of geodesic paths between s and t. Mathematically,

$$g(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

If v is an articulation point, then we can further simplify this as follows. Let the two components that the removal of v divides the graph into be C_1 and C_2 with N_1 and N_2 nodes respectively. Then,

$$\begin{aligned} g(v) &= 2 \sum_{s \in C_1, t \in C_2} \frac{\sigma_{st}(v)}{\sigma_{st}} \\ &= 2 \sum_{s \in C_1, t \in C_2} 1 \\ &= 2N_1N_2 \end{aligned}$$

Removal of a node with high betweenness centrality can lead to increase in the geodesic path lengths, and in the extreme case, the network might even get disconnected as exhibited in the case above. In real world networks, this can be important; for example, to prevent the spread of a disease in an epidemic network.

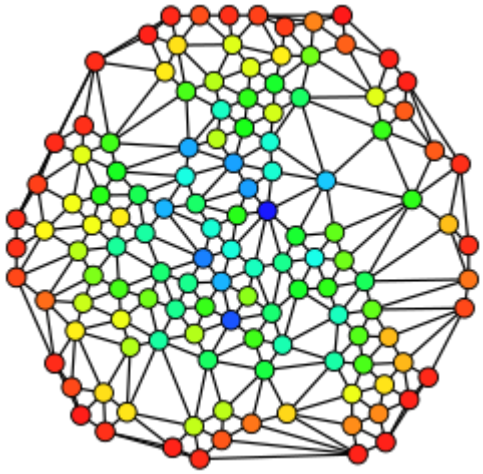


Figure 2.6: Hue (from red=0 to blue=max) shows the node betweenness.

2.5.3 Flow Betweenness

Suppose two nodes are connected by a **reluctant broker** (cut vertex), that is, the shortest path between them is blocked. Then, the nodes should use another pathway which is connecting them, rather than simply using the geodesic path.

The **flow betweenness** measure thus expands the notion of betweenness centrality. It assumes that any two nodes would use all the paths connecting them, instead of only using shortest path. However, it is to be noted that calculating flow betweenness is computationally intractable.

2.5.4 Eigenvector Centrality

Eigen Vector

The value λ is an eigenvalue of matrix A if there exists a non-zero vector x, such that $Ax = \lambda x$. Vector x is an eigenvector of matrix A

The largest eigenvalue is called the principal eigenvalue The corresponding eigenvector is the principal eigenvector Corresponds to the direction of maximum change

Eigenvector centrality is another measure of influence of a node in a network. It assigns relative scores to all nodes in the network based on the concept that connections to high-scoring nodes contribute more to the score of the node in question than equal connections to low-scoring nodes.

The idea is to define centrality of the vertex as the sum of centralities of its neighbours.

We now proceed to define the centrality value of a vertex as a sum of centralities of its neighbours. To begin with, we initially guess that a vertex i has centrality $x_i(0)$. We gradually improve this estimate by employing a Markov model, and continue in this manner until no more improvement is observed. The improvement made at step t is defined as,

$$\begin{aligned} x_i(t) &= \sum_j A_{ij} x_j(t-1) \\ \Rightarrow x(t) &= \mathbf{A}x(t-1) \\ &= \mathbf{A}^t x(0) \end{aligned}$$

This is known as the Power Iteration method proposed by Hotelling.
Now, express $x(0)$ as a linear combination of eigenvectors v_i of the adjacency matrix $\mathbf{A}$

$$\begin{aligned} x(0) &= \sum_i c_i v_i \\ \Rightarrow x(t) &= \mathbf{A}^t \sum_i c_i v_i \end{aligned}$$

We know from our knowledge of eigenvectors that $\mathbf{A}^t x = \lambda^t x$ holds, where λ is an eigenvalue. Using this with the equation above, we have

$$\begin{aligned} x(t) &= \sum_i \lambda_i^t c_i v_i \\ &= \lambda_1^t \sum_i \left(\frac{\lambda_i}{\lambda_1}\right)^t c_i v_i \\ \Rightarrow \frac{x(t)}{\lambda_1^t} &= \sum_i \left(\frac{\lambda_i}{\lambda_1}\right)^t c_i v_i \end{aligned}$$

In the limit $t \to \infty$, $\left(\frac{\lambda_i}{\lambda_1}\right)^t$ remains only for $i = 1$. Thus,

$$\lim_{t\to\infty} \frac{x(t)}{\lambda_1^t} = c_1 v_1$$

Thus, we get that the limiting centrality is proportional to the principal eigenvector v_1.

Note that directed acyclic networks suffer from the problem of **zero centrality**. If there exists a node A with no incoming edges, then this node has zero centrality (the assumption seems reasonable for a web page). Consider another node B that has one incoming edge from A. Then the eigenvector centrality of B is 0 because A the centrality of A is 0. Hence, in a similar fashion, all the centralities in an acyclic network become 0. We will see how this problem is remedied by the Katz Centrality metric.

2.5.5 Katz Centrality

Katz centrality can be used to compute centrality in directed networks such as citation networks and the World Wide Web. Katz centrality is more suitable in the analysis of directed acyclic graphs where traditionally used measures like Eigenvector centrality are rendered useless. Katz centrality can also be used in estimating the relative status or influence of actors in a social network. Each node is provided a small amount of centrality irrespective of its position. hence

$$x_i = \alpha \sum_j A_{ij} x_j + \beta$$

Note that $\alpha, \beta > 0$. In matrix terms, the above equation is equivalent to

$$\mathbf{x} = \alpha \mathbf{A}\mathbf{x} + \beta \mathbf{1}$$

where $\mathbf{1} = (1, 1, \cdots, 1)^T$. On simplifying, we obtain

$$\mathbf{x} = \beta(\mathbf{I} - \alpha \mathbf{A})^{-1}\mathbf{1}$$

Instead of inverting the matrix as above, we can alternatively iterate over the following equation until convergence

$$\mathbf{x}(t) = \alpha \mathbf{A}\mathbf{x}(t-1) + \beta \mathbf{1}$$

2.5.6 PageRank

Page rank algorithm is a link analysis algorithm used by google search engine that assigns a numerical weighting to each element of a hyperlinked set of documents such as world wide web.

The Google theory goes that if Page A links to Page B, then Page A is saying that Page B is an important page. PageRank also factors in the importance of the links pointing to a page. If a page has important links pointing to it, then its links to other pages also become important.

Essentially, PageRank is nothing but a variant of Katz Centrality. It can be mathematically expressed as follows.

$$x_i = \alpha \sum_j A_{ij} \frac{x_j}{k_j^{out}} + \beta$$

where k_j^{out} is the out-degree of node j. This normalization is done to obtain a stochastic matrix (a matrix where either all the rows or all the columns sum to one). Note that the above definition does not take into account the possibility of $k_j^{out} = 0$. To solve this problem, set $k_j^{out} = 1$ in the above calculation, since a vertex with zero out-degree contributes zero to centralities of other vertices. In matrix terms, we have

$$\begin{aligned} \mathbf{x} &= \alpha \mathbf{A} \mathbf{D}^{-\mathbf{1}} \mathbf{x} + \beta \mathbf{1} \\ \Rightarrow \mathbf{x} &= \beta (\mathbf{I} - \alpha \mathbf{A} \mathbf{D}^{-\mathbf{1}})^{-1} \mathbf{1} \end{aligned}$$

where $\mathbf{D}$ is a diagonal matrix such that

$$D_{ii} = \max \{k_i^{out}, 1\}$$

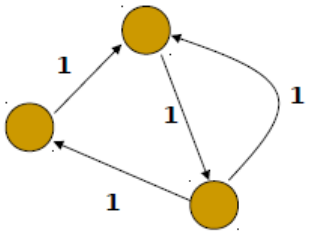
Random Walks

A **random walk** is a mathematical formalisation of a trajectory that consists of taking successive random steps. It was introduced by Karl Pearson in 1905.

Random walks are useful to analyze web surfing and to calculate PageRank values. Consider web surfing, initially, every page is chosen uniformly at random. With probability α, the surfer performs random walk by randomly choosing the hyperlinks in that page, and with probability $1 - \alpha$, the surfer stops the random walk. We already know that the steady state probabiliity that a web page is visited during web surfing represents its PageRank.

The transition matrix for web surfing is obtained from the adjacency matrix representing the underlying graph structure. The transition matrix is a stochastic matrix, all rows sum to 1, and is thus obtained by dividing each number in each row by the sum of the elements in that row in the adjacency matrix. Essentially, an entry in the transition matrix represents the probability with which that link is chosen.

As an example, consider the following graph and its equivalent adjacency matrix



$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

For the above graph, the transition matrix is given as,

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1/2 & 1/2 & 0 \end{bmatrix}$$

Here, we pictorially show a random walk on this network.

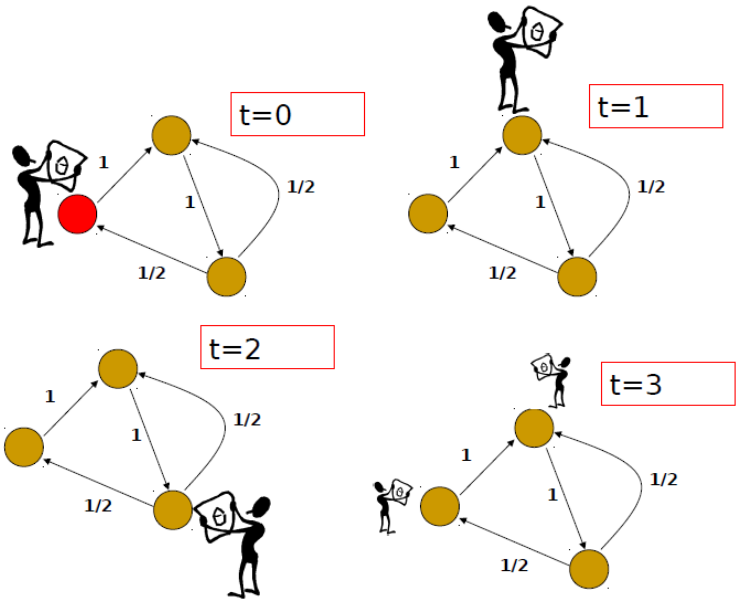


Figure 2.7: Random walk on the graph

Steady-state Calculation

We put $\beta = \alpha - 1$ in the pagerank expression. So,

$X(t) = \alpha AD^{-1}X(t-1) + (1-\alpha)1$

Now $\Sigma_{1..n}X_i(t) = 1$

So $X(t) = \alpha AD^{-1}X(t-1) + (1-\alpha)11^T X(t-1)$

$X(t) = PX(t-1)$

where $P = \alpha AD^{-1} + (1-\alpha)11^T$

P^T is the probability transition matrix.

Steady state probability: $\lim_{m\to\infty} P^{T^m}$

2.6 Hubs and Authorities

The idea behind Hubs and Authorities is rooted in a particular insight into the creation of web pages when the Internet was originally forming; that is, certain web pages, known as hubs, served as large directories that were not actually authoritative in the information that it held, but were used as compilations of a broad catalog of information that led users directly to other authoritative pages. A good hub represents a page that pointed to many other pages, and a good authority represents a page that was linked by many different hubs.

The scheme therefore assigns two scores for each page: its **authority**, which estimates the value of the content of the page, and its **hub value**, which estimates the value of its links to other pages. Mathematically, these two centrality values are expressed as follows. The **authority centrality** of a node (x_i) is proportional to the sum of hub centralities of nodes (y_j) pointing to it, and is defined as

$$x_i = \alpha \sum_j A_{ji} y_j$$

The **hub centrality** of a node is proportional to the sum of authority centralities of nodes pointing to it, and is defined as

$$y_i = \beta \sum_j A_{ij} x_j$$

In matrix terms, $\mathbf{x} = \alpha \mathbf{A}^T \mathbf{y}$, and $\mathbf{y} = \beta \mathbf{A}\mathbf{x}$. Solving these two equations gives us

$$\mathbf{x} = \alpha\beta \mathbf{A}^T \mathbf{A}\mathbf{x}$$

$$\mathbf{y} = \alpha\beta \mathbf{A}\mathbf{A}^T \mathbf{y}$$

where $\mathbf{x}$ converges to the prinicipal eigenvector of $\mathbf{A}^T\mathbf{A}$, and $\mathbf{y}$ converges to the principal eigenvector of $\mathbf{A}\mathbf{A}^T$.

2.7 Rich-Club Coefficient

For a given network , when influential nodes come together to collaborate on something, they form what is called a **Rich club**. As an example, *hubs* in a network are generally densely connected, and form a rich-club.

Formally, the rich-club of degree k of a network $\mathbf{G} = (V, E)$ is the set of vertices with degree greater than k. This can be mathematically expressed as,

$$R(k) = \{v \in V | k_v > k\}$$

The rich-club coefficient of degree k is given by,

$$\frac{\#\text{edge}(i,j)}{|R(k)||R(k)-1|}, \text{ where } (i,j) \in R(k)$$

Chapter 3

Social networks

3.1 Matching Index

Matching index is assigned to each edge in a network in order to quantify the similarity between the connectivity pattern of the two verticces adjacent to that edge.A low value of matching index would indicate dissimilar regions of the network, with theedge serving as a shortcut between distant regions in the network.

Formally,matching index of an edge(i,j) is defined as

$$\mu_{i,j} = \frac{\Sigma_{k\neq i,j} a_{ikj} a_{kj}}{\Sigma_{k\neq j} a_{i,k} + \Sigma_{k\neq i} a_{j,k}}$$

The value of $\mu_{i,j}$ varies between 0 and 1/2.

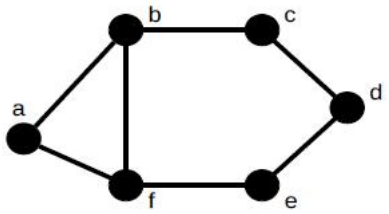
3.2 Social cohesiveness

Social cohesiveness refers to the closeness of members in a social network. a cohesive subgroup consists of actors connected through dense ,direct,reciprocated choice relations that enable members to share information, create solidarity,act etc.Numerous direct contacts among all subgroup members, combined with few or no ties to outsiders, dispose a group toward homogeneity of thought, identity, and behavior.

3.2.1 Clique

Clique is undirected maximal complete subgraph consisting of atleast three nodes.A clique consists of the largest number of nodes of the graph with ties to all other nodes of the clique.

The clique density is always 1.



A node can be a member of more than one clique; atlest one node differs in every clique.

A clique imposes the most stingent definition of cohesiveness,because its complete adjacency requirement means a single absent edge may evict a node from a group.more lenient membership criteria permit less-than-complete connections within a subgroup,thus allowing some differentaition in its internal structure.Some of these quasi-clique measures are-

3.2.2 K-Clique

A K-clique of a graph is a maximal subset S such that geodesic distance between every pair of vertices in the set is less than or equal to k. That means no two nodes in the set can be more than k steps away from each other.

in the given graph of fig-1 {a,b,c,f,e} forms a 2-clique.

fig1

3.2.3 K-Clan

A k-clan of a graph is a k-clique in which the subgraph induced by S has diameter less than or equal to k. So a subset to be a k-clan-

1. should be a k-clique.
2. all nodes are connected by a path less than or equal to k.

In the graph of fig-1 {b,c,d,e,f} forms a k-clan.

3.2.4 K-Plex

K-plex of a graph is a maximal subgraph with the following property: each vertex of the induced subgraph is connected to at least n-k other vertices, where n is the number of vertices in the induced subgraph.

A k-core of a graph is a maximal subgraph such that each node in the subgraph has at least degree k.

3.3 Equivalence

In a social network the positions or roles or social categories are defined by the relations among actors. Depending on the pattern of relationship with other

nodes in the network, two nodes can have same position or role in the network. The similarity or equvalence of two nodes in a network can be defined in several ways. Of them three are particularly important.

They are 1.Structural equivalence 2.Automorphic equivalence 3. Regular equivalence.

3.3.1 Structural Equivalence

Two nodes are structurally eqivalent if they have same relation with other nodes in the network i.e, they are perfectly substituable. Each of them are connected to exactly same set of neighbors and replacing one with the other doesnot change the network at all. But given a large complex network such equivalence can rarely be seen. Hence , there is a need to examine the degree of structural equivalence rather than mere presence of exact equivalence in a network.

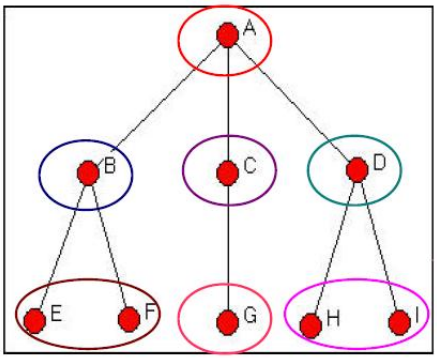


fig-3 Different structural equivalence classes in a network

The degree of structural equivalence between two nodes i,jcan be measured by examining the number of common neighbors,which can be expressed as

$$n_{i,j} = \Sigma_k A_{ik} A_{kj}$$

which is basically the ijth element of the matrix A^2. But it needs to be appropriately normalized since we are measuring the extent of similarity. The mesured has been refined by alternate considerations-

Cosine Similarity

cosine similarity measure is defined as the inner product of two vectors. That is,

$$\text{cosine similarity}=Cos\theta=\frac{x.y}{||x||*||y||}$$

If we consider ith and jth rows of an adjacency matrix A, then Cosine similarity between vertices i and j is-

$$\sigma_{ij} = \frac{\Sigma_k A_{ik} A_{kj}}{\sqrt{A_{ik}^2}\sqrt{A_{kj}^2}} = \frac{n_{ij}}{\sqrt{K_i K_j}}$$

Pearson Correlation coefficient

Pearson correlation coefficient between ith and jth rows of an adjacency matrux are-

$$r_{ij} = \frac{\Sigma_k A_{ik} A_{kj} - \frac{K_i k_j}{n}}{\sqrt{K_i - \frac{K_i^2}{n}}\sqrt{K_j - \frac{K_j^2}{n}}}$$

3.3.2 Automorphic Equivalence

Formally "Two vertices u and v of a labeled graph G are automorphically equivalent if all the vertices can be re-labeled to form an isomorphic graph with the labels of u and v interchanged. Two automorphically equivalent vertices share exactly the same label-independent properties." (Borgatti, Everett, and Freeman, 1996: 119).

More intuitively, actors are automorphically equivalent if we can permute the graph in such a way that exchanging the two actors has no effect on the distances among all actors in the graph. If we want to assess whether two actors are automorphically equivalent, we first imagine exchanging their positions in the network. Then, we look and see if, by changing some other actors as well, we can create a graph in which all of the actors are the same distance that they were from one another in the original graph.

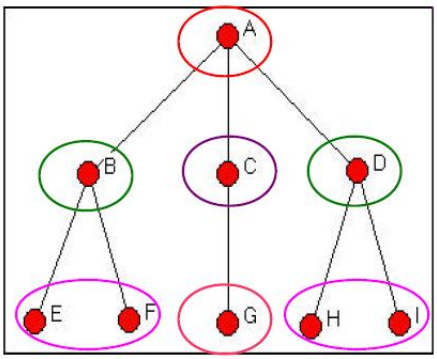


fig-4 Different structural equivalence classes in a network

In structural equivalence we are trying to find actors which are clones or substitutes. On the other hand automorphic equivalence asks if the whole network can be re-arranged, putting different actors at different nodes, but leaving the relational structure or skeleton of the network intact.

3.3.3 Regular Equivalence

Regularly equivalent vertices are vertices that, while they do not necessarily share neighbors, have neighbors who are themselves similar.Quantitative measures of regular equivalence are less well developed than measures of structural equivalence.The basic idea is to define a similarity score σ_{ij} such that i and j have high similarity if they have neighbors k and I that themselves have high similarity. For an undirected network we can write this as

$$\sigma_{ij} = \alpha \Sigma_k A_{ik} \sigma_{kj} + \delta_{ij}$$

or, in matrix form it can be written as

$$\sigma = \alpha A \sigma + I$$
$$\sigma = (I - \alpha A)^{-1}$$

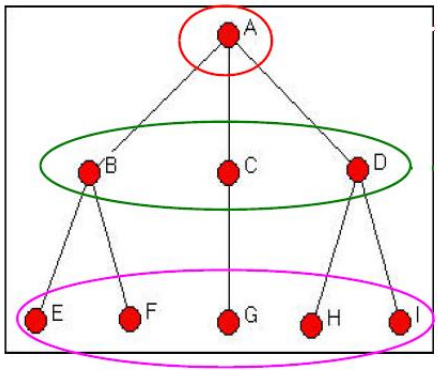


fig-5 Different regular equivalence classes in a network

If two vertices in a graph are structurally equivalent then they are also automorphically equivalent but the reverse is not true Again if two vertices are structurally equivalent then they are also regular equvalent but the reverse is not true.Also if two vertices are automorphic equivalent then they are also regularl equivalent but the reverse is not true. structural equivalence is the most strict equivalence measure. Automorphic equivalence is more strict than regular equivalence but less compared to structural equivalence.

3.4 Assortativity

Assortativity or homophily is the tendency of the nodes in a netwirk to attach with similar nodes. People are found to form friendships, acquaintances, business relations, and many other types of tie based on all sorts of characteristics, including age, nationality, language, in- come, educational level, and many others. Almost any social parameter one can imagine plays into people's selection of their friends. People have, it ap- pears, a strong tendency to associate with others whom they perceive as being similar to themselves in some way. This tendency is called homophily or assortativity.

Disassortativity is just the opposite of assortativity. Heare the like nodes link with unlike nodes. Protein-Protein interaction graph is a disassortative graph.

3.4.1 Measuring Assortativity

One way of capturing the degree correlation is by examining the properties of knn , or the average degree of neighbors of a node with degree k. This term is formally defined as

$$< k_{nn} >= \Sigma_{k'} k' p(k'|k)$$

Where $p(k_{'}|k)$ is the conditional probability that an edge of node degree k points to a node of degree $k^{'}$. If this function is increasing, the network is assortative, since it shows that nodes of high degree connect, on average, to nodes of high degree. Alternatively, if the function is decreasing, the network is dissortative, since nodes of high degree tend to connect to nodes of lower degree.

3.5 Signed Graph

A signed graph is a graph in which each edge has a positive or negative sign. Such graphs have been used to model social situations, with positive edges representing friendships and negative edges representing enmities between nodes, which represent people.

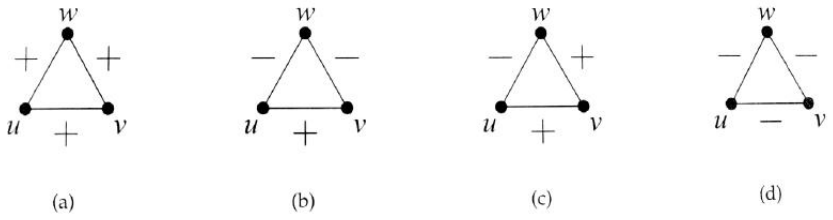


fig-6 triads (a) and (b) are stable configurations whlile (c) and (d) are unstable

The sign of a cycle in the graph is defined to be the product of the signs of its edges; in other words, a cycle is positive if it contains an even number of negative edges and negative if it contains an odd number of negative edges. A signed graph, or a subgraph or edge set, is called balanced if every cycle in it is positive.

Chapter 4

Community identification and Clustering

Community structures are said to be present in a network if it can be divided into sets of nodes such that they are densely connected among themselves and are sparsely connected to the rest of the network. Identifying community structures inside a network finds application in studying drug interactions,disease spreading, CPU optimization and many more.

Based on the approach employed, clustering techniques can be broadly categorized into the following computational methods, agglomerative, divisive and spectral.

Agglomerative techniques make use of a bottom-up approach for clustering. Start- ing with an empty graph G with N nodes and no edges, edges are iteratively added to the graph, while maximizing some quantity in the original network.

Divisive techniques make use of a top-down approach, removing certain edges from the original network so that separate community structures are obtained.

Spectral techniques split the graph into community structures based on eigenvalues / eigenvectors of the Graph Laplacian.

4.1 Similarity measure

A crucial step in any algorithm to identify community structures is to select suitable metrics to measure similarity or dissimilarity between nodes. The goal remains to to group similar data together, which would constitute a community. However, there is no single method that works equally well in all applications; it depends on what we want to find or emphasize in the data. Therefore, correct choice of a similarity measure is often more important than the clustering algorithm. As discussed in previous chapters, similarity measures could be obtained as Cosine Similarity, Jaccards Coefficient, etc.

4.2 Agglomerative Methods

Some of the agglomerative approaches are as follows-

4.2.1 Heirarchical clustering

Heirarchical clustering is also known as bottom-up clustering. The steps followed in this approach are-

1. Start with every data points in a separate cluster.
2. Merge the most similar pairs of data points or clusters.
3. Continue step 2 until we get one large cluster.

The output of the above method is a binary tree, called a dendogram. The root of this tree is the final cluster, and each original data item is a leaf. Initially, the tree is empty, containing only the original data items as leaves. Whenever data items / clusters are merged together, a node is added to the tree (representing this new cluster) with edges between this new node and its constituent clusters. Clusters can be obtained by cuttimg the dendogram at desired level.

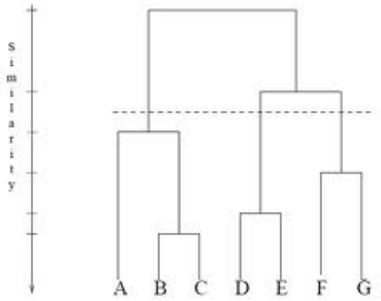


fig-6 Heirarchical clustering

As already mentioned, we could have used any of the previously defined measures of similarity to estimate the distance between data items. However,we need to define a linkage method that can estimate the distance between clusters. Since a data item can be thought of as a cluster with a single node, this linkage method will suffice for data items as well. Here we enumerate the different types of linkages that might be followed while merging any two clusters:

Single Linkage: The minimum of all pairwise distances between points in the two clusters

Complete Linkage: The maximum of all pairwise distances between points in the two clusters

Average Linkage: The average of all pairwise distances between points in the two clusters

Despite its simplicity, this approach does not scale to large graphs, owing to its O(n3) time complexity in the worst case. Also, the method is not flexible;

steps once taken cannot be undone. Another problem this approach suffers from is that arbitrary cut-offs need to be set to arrive at a community structure.

4.2.2 Local algorithm based on agglomeration

This algorithm was proposed by James.P.Bagrow. Apart from agglomerating one node at a time This method maintains two groups -a community B and a border C with B consisting of a set of nodes adjacent to the community C.At each step one node is picked from B and added to C.This continues until a stopping criteria is reached.

Define the outwardness v (C) of a node v B from community C as

$$\Omega_v(C) = \frac{\#ofneighborsofvoutsideC - \#ofneighborsofvinsideC}{K_v}$$

The algorithm is as follows-
1. Choose starting node s: C = {s};B = {neighbors of s};
2. Add v B to C, where $\Omega_v = \min\{\Omega\}$;
3. Update B, $\Omega^{'}$s, repeat from 2;

4.2.3 Modularity

Modularity measures the strength of division of a network into clusters or communities. Networks with high modularity have dense connections between the nodes within the modules and sparse connections with the nodes in different modules. Formally, modularity is the fraction of the edges that fall within the given groups minus the expected such fraction if edges were distributed at random. Modularity is expressed as

$$Q = \frac{1}{2m}\Sigma_{vw}(A_{vw} - \frac{k_w k_v}{2m})\delta_{C_v C_w}$$

Optimizing modularity and Louvain method

Modularity is often used in optimization methods for detecting community structure in networks.

The Louvain method is a simple, efficient and easy-to-implement method for identifying communities in large networks. It is a greedy optimization method that attempts to optimize the modularity of a partition of the network.The optimization is performed in two steps. First, the method looks for "small" communities by optimizing modularity locally. Second, it aggregates nodes belonging to the same community and builds a new network whose nodes are the communities. These steps are repeated iteratively until a maximum of modularity is attained and a hierarchy of communities is produced. Although the exact computational complexity of the method is not known, the method seems to run in time O(n log n) with most of the computational effort spent on the optimization at the first level. Exact modularity optimization is known to be NP-hard.

4.3 Divisive method

4.3.1 Girvan-Newman Algorithm

Previously we have studied the betweenness centrality of a vertex in a network.For any node i, we defined vertex betweenness as the number of shortest paths between pairs of nodes that run through it. The GirvanNewman algorithm extends this definition to the case of edges, defining the "edge betweenness" of an edge as the number of shortest paths between pairs of nodes that run along it. If there is more than one shortest path between a pair of nodes, each path is assigned equal weight such that the total weight of all of the paths is equal to unity. If a network contains communities or groups that are only loosely connected by a few intergroup edges, then all shortest paths between different communities must go along one of these few edges. Thus, the edges connecting communities will have high edge betweenness (at least one of them). By removing these edges, the groups are separated from one another and so the underlying community structure of the network is revealed.

The algorithm's steps for community detection are summarized below-

1. The betweenness of all existing edges in the network is calculated first.
2. The edge with the highest betweenness is removed.
3. Remove it from the network.
4. Recalculate the scores.
5. Repeat Steps 2,3 and 4 until no edges are left in the graph.

The crux of this method lies in the computation of the shortest paths. If we use simple BFS traversal for this computation, then, this can be done in $O(m)$ time for each source node, totalling to $O(mn)$ time for all the nodes, where m is the number of edges in the graph. In the worst case, $O(m)$ edges are removed, therefore, the total complexity of the algorithm is $O(m^2n)$, which is equivalent to $O(n^3)$ for sparse graphs, and $O(n^5)$ for dense graphs.

4.3.2 Radicchi's Algorithm

This algorithm is a divisive algorithm that is based on the notion that the number of triangles formed within communities is much higher than the number of triangles across communities. The algorithm tries to find the edge clustering coefficient of each edge; we remove the edge with the smallest value of the coefficient from the network. This coefficient is a measure of the number of triangles a particular edge ij is a part of, and is defined as:

$$C_{ij} = \frac{Z_{ij}}{min[(K_i-1)(K_j-1)]}$$

where Z_{ij} = Number of triangles ij is a part of. Note that the denominator of the ex- pression denotes the maximum number of triangles of which ij could possibly be a part of.The algorithm runs in $O(mn)$.