Laplacian Matrix of a Network and Applications

Alice Nanyanzi (alicenanyanzi@aims.ac.za)
African Institute for Mathematical Sciences (AIMS)

Supervised by: Dr Franck Kalala Mutombo
AIMS South Africa, University of Cape Town, University of Lubumbashi

19 May 2016

Submitted in partial fulfillment of a structured masters degree at AIMS-South Africa

AIMS | African Institute for Mathematical Sciences
SOUTH AFRICA

Abstract

In this project, the Laplacian matrix of a network is analysed. A network here is viewed as a graph with nodes and links between them. The Lapacian matrix is the difference between the degree matrix and the adjacency matrix of the network. We discuss two applications of the Laplacian matrix. One application is the Laplacian centrality, a new centrality measure for weighted networks. We also investigate and interpret the diffusion process over a network. We use the Python programming language for simulations and computations. In addition, we use network software such as Pajek, yED, etc. for visualisation and analysis of networks.

Declaration

I, the undersigned, hereby declare that the work contained in this research project is my original work, and that any work done by others or by myself previously has been acknowledged and referenced accordingly.

NAlice

Alice Nanyanzi, 19 May 2016

Contents

1. Introduction

The concept of network is very intuitive to everyone in our modern society. The word network literally means a group of interconnected items (Estrada, 2011). Many systems in real world are composed of components that are linked to each other. Since these systems are part of us, it is important to study different aspects of these systems, for instance, the nature of individual components, the connections or interactions between different components, etc. In the study of these systems, we often use networks to represent the skeleton of the systems. A network also called a graph in the mathematical literature is a collection of vertices joined by edges. Vertices and edges are also called nodes and links in computer science, sites and bonds in physics, and actors and ties in sociology (Newman, 2010). The history of network theory begins with the early work done by the Swiss mathematician Leonhard Euler about the Köningsberg's bridges problem (Euler, 1953), (Gribkovskaia et al., 2007).

Today, it is evident that networks have applications in a large number of fields ranging from biology, technology, physics, ecology, social sciences, etc. However, research shows that the topological structures of most real-world networks cannot be trivially described (Estrada, 2015). Such networks are therefore referred to as complex networks. Examples of complex networks include: internet, social networks, neural network, metabolic network, etc. The study of complex networks has recently become an interesting area of research especially the study of the topological structure of these networks. A structure is defined as the way in which components of a system are organised (Estrada, 2011).

The representation of a network plays a key role in the study of networks. Most often, networks are represented as matrices. This is because it is not only convenient but also, the properties of the matrix give us information about the network that the matrix represents (Molitierno, 2012). One of such matrices is the Laplacian matrix which is defined as the difference between the degree and the adjacency matrices of a network. Our motivation to study the Laplacian matrix is that its properties give more information about the structure of a network compared to other matrices. We will explore two of the applications of the Laplacian matrix. In network analysis, it is vital to identify which node(s) are more important (or central) in a network. The measure of this importance is known as centrality. One of the applications of Laplacian matrix is a new centrality measure for weighted networks called the Laplacian centrality, in which the importance of a node is related to the ability of the network to respond to the deactivation or removal of that node from the network. This centrality measure provides intermediate information about a particular node (Qi et al., 2013). We perform computations on datasets such as the Zachary karate club network to compare the Laplacian centrality to other standard centrality measurements. We also explore diffusion of a substance such as heat, information, epidemic, etc. over a network, which is one of the physical analogies of the Laplacian matrix.

This essay is organized as follows: Chapter 2 is a brief review of graph and network theory, characteristics of networks and network models. In chapter 3, we focus on the Laplacian matrix of a network and properties of its spectrum. The spectrum of the Laplacian matrix is the set of the eigenvalues together with their multiplicities. In Chapter 4, we discuss two applications of the Laplacian matrix which include the Laplacian centrality and the diffusion process over a network which is one of the physical analogies of the Laplacian matrix. Finally, we draw conclusions and discuss areas of further research in Chapter 5.

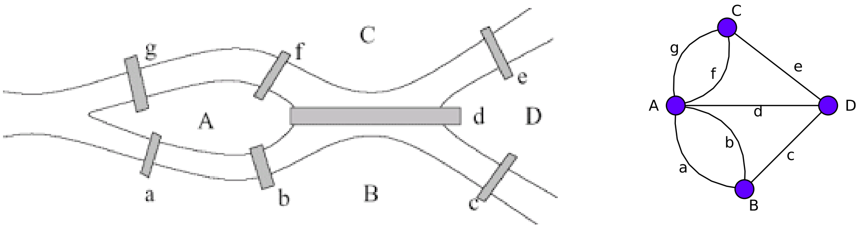
2. Review of Networks

In this chapter, we discuss various concepts of graph theory, characteristics of networks, and different models of networks.

2.1 Graphs and Networks

According to Estrada (2011), in mathematics the study of networks is known as graph theory. In this essay, we will use the two words: 'graph' and 'network' interchangeably. We consider the earliest representation of a graph by the famous Swiss mathematician, Euler.

2.1.1 Königsberg bridge problem. In Prussia, there was a city called Königsberg (now Kaliningrad, Russia) which was set on both sides of the branched Pregel river forming two islands. These islands were connected to each other and the mainland by seven bridges (see Fig. 2.1(a)). The challenge to the citizens was to find a walk around the city that crosses each bridge exactly once (known as the Königsberg bridge problem). In the attempt to solve the problem, Euler reformulated and represented the Königsberg bridge problem in a way that is similar to what is referred to as a graph as shown in Fig. 2.1(b). It was then that the story of graph theory begun (Estrada, 2011).



(a) (b)

Figure 2.1: The Königsberg bridges: (a) A schematic diagram of the seven Königsberg bridges. (b) A graph of the Königsberg bridges. Source: (Internet).

2.1.2 Definition (Graph)**.** A graph is a pair $G = (V, E)$, where V is a set of vertices or nodes, and E is a set of edges between the vertices, $E \subseteq \{(u, v)|u, v \in V\}$. A graph may be undirected, that is edges have no orientation or it may be directed, that is edges have direction. Fig. 2.1(b) is an example of a graph. The order of a graph G, denoted as $|G|$, is the number of vertices of that graph. On the other hand, the number of edges of a graph is denoted by $||G||$. The order (or size) of a graph determines whether it is finite or infinite. For example, the graph in Fig. 2.2(a) has order 7. In this essay, we will work with finite graphs.

2.1.3 Definition (Subgraph)**.** Given a graph $G = (V, E)$, we say $G' = (V', E')$ is a subgraph of G if and only if $V' \subseteq V$ and $E' \subseteq E$. G is called the supergraph (Estrada et al., 2015). For example, Fig. 2.2(b) and Fig. 2.2(c) are subgraphs of the supergraph in Fig. 2.2(a).

Graphs are categorized based on different aspects, the most common being the nature and orientation of their edges. We therefore, discuss some of the popular categories of graphs.

2.1.4 Definition (Simple graph)**.** A simple graph is a graph that has neither loops nor multiple edges. A loop is an edge that connects a vertex to its self. Multiple edges are two or more edges that are

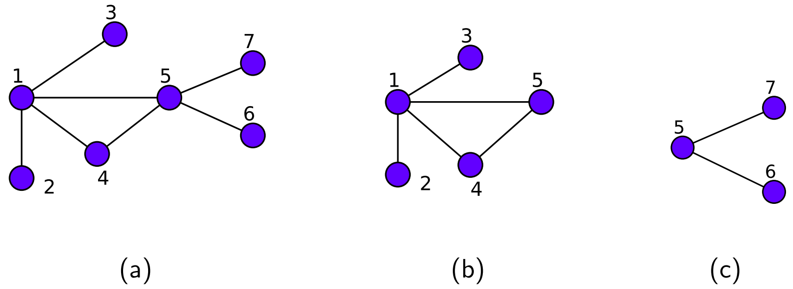


Figure 2.2: A graph of size 7 and its subgraphs: (a) A graph G. (b) A subgraph of G with $|V'| = 5$ and (c) is another subgraph of G with $|V'| = 3$.

incident to the same two vertices (Newman, 2010). See Fig. 2.3(a).

2.1.5 Definition (Multi-graph)**.** A multi-graph is a graph with multiple edges (Newman, 2010). Fig. 2.3(b) illustrates a graph in which the three edges connecting node 1 to node 5 and the two edges connecting node 3 to node 4 are multiple edges.

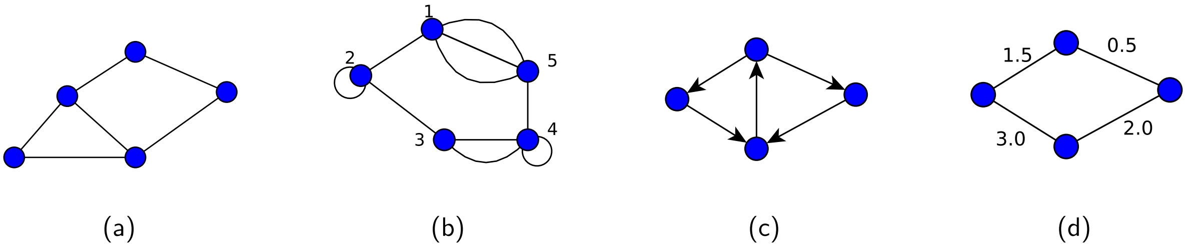


Figure 2.3: Classifications of graphs: (a) A simple graph. (b) A graph with multiple edges and loops. (c) A directed graph. (d) A weighted graph.

2.1.6 Definition (Directed graph)**.** A directed graph (or digraph) is a graph in which each edge has a direction, pointing from one vertex to another (Newman, 2010). Such edges are called directed edges and are represented by drawing a line with an arrow at one end as shown in Fig. 2.3(c).

2.1.7 Definition (Weighted graph)**.** A weighted graph is a graph in which each edge $e = \{i, j\}$ is associated with a value or weight $w_{i,j}$ which is usually a real number. The weights take on different interpretations depending on what the graph represents. For example, a graph depicting a transportation system in a city, the routes have weights that represent the cost of fuel incurred by using those routes while for a social network, the weights on the connections represent the frequency of communication between the two people (Newman, 2010). Fig. 2.3(d) is a weighted graph of 4 nodes.

2.2 Networks

Networks are used in many fields such as in biology, chemistry, computer science, transport, psychology, social sciences among others. For instance, in computer science, a network can be a representation of computers, routers, or any other electronic devices that are connected together by wires or wireless connections.

2.2.1 Definition (Network)**.** A network is a diagrammatic representation of a system. It consists of nodes (vertices), which represent the entities of the system. Pairs of nodes are are joined by links (edges), which represent a particular kind of interconnection between those entities (Estrada, 2011).

However, Definition 2.2.1 does not exploit the different ways in which the nodes are connected and their directions. For instance, directed edges, self-loops and multiple edges. It is because of such issues that Gutman and Polansky (2012) suggested definitions for a simple network as well as a more general definition of networks. First, let us understand the term 'relation'.

2.2.2 Definition (Relation)**.** Consider a finite set $V = \{v_1, v_2, \cdots, v_n\}$ of unspecified elements, and let $V \otimes V$ be the set of all ordered pairs $[v_i, v_j]$ of the elements of V. A relation on the set V is any subset $E \subseteq V \otimes V$. The relation E is symmetric if $[v_i, v_j] \in E$ implies $[v_j, v_i] \in E$, and it is reflexive if $\forall\ v \in V$, $[v, v] \in E$. The relation E is antireflexive if $[v_i, v_j] \in E$ implies $[v_i \neq v_j]$ (Estrada, 2011).

2.2.3 Definition (Simple network)**.** A simple network is the pair $G = (V, E)$, where V is a finite set of nodes and E is a symmetric and antireflexive relation on V. In a directed network the relation E is non-symmetric (Estrada, 2011).

2.2.4 Definition (Network: More general definition)**.** A network is a triple $G = (V, E, f)$, where V is a finite set of nodes, $E \subseteq V \otimes V = \{e_1, e_2, \cdots, e_m\}$ is a set of links, and f is a mapping which associates some elements of E to a pair of elements of V, such as that if $v_i \in V$ and $v_j \in V$, then $f : e_1 \to [v_i, v_j]$ and $f : e_2 \to [v_j, v_i]$. A weighted network is created by replacing the set of links E by the set of link weights $W = \{w_1, w_2, \cdots, w_m\}$, such that $w_i \in \mathcal{R}$. Then, a weighted network is defined by $G = (V, W, f)$ (Estrada, 2011).

2.2.5 Remark. In this essay, we will consider simple undirected networks unless specified otherwise.

2.2.6 Examples of real-world networks.

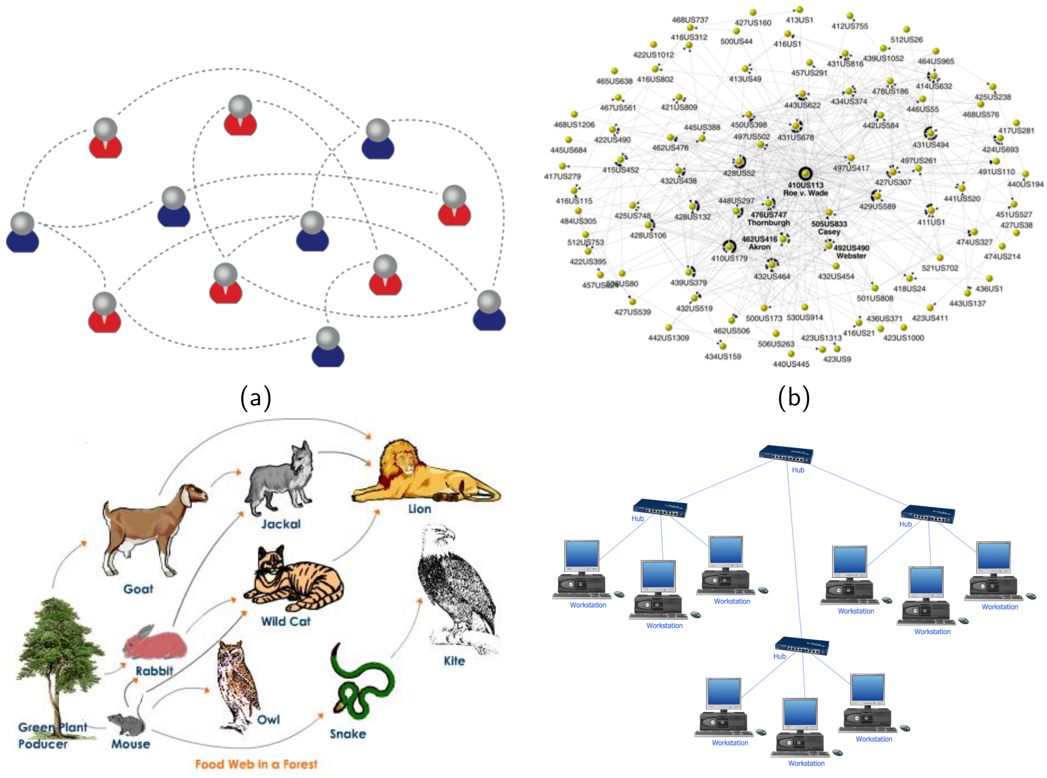
- Social Networks: In social networks, the nodes are the individuals (or people) and the links represent the friendship between them. Examples of social networks are facebook, twitter, etc.
- Citation network: The nodes of the network may be articles, patents, or legal case while the links are the citations.
- Food web: In a food web, the species are the nodes as the links represent the predation.
- Computer network: The nodes are computers or routers while the links are cables or wireless data connections.

2.2.7 Terminology in network theory.

2.2.8 Definition (Incidence)**.** Given a network $G = (V, E)$. We say that node v and edge e are incident if v is one of the nodes to which edge e connects. Two edges e_1 and e_2 are said to be incident if they share a vertex $v \in V$ (Newman, 2010).

2.2.9 Definition (Incidence matrix)**.** Consider a network with vertex set $V = \{v_1, v_2, \cdots, v_n\}$ and edge set $E = \{e_1, e_2, \cdots, e_m\}$. Let us consider an arbitrary orientation of every edge in the network, say, we label each edge $\{v_i, v_j\}$ in a way that v_i is the positive end and v_j is the negative end. Then the oriented incidence matrix $\mathbf{B}(\mathbf{G})$ has entries defined as

$$B_{ij} = \begin{cases} +1 & \text{if node } v_i \text{ is the positive end of the edge } e_j \\ -1 & \text{if node } v_i \text{ is the negative end of the edge } e_j \\ 0 & \text{otherwise.} \end{cases} \tag{2.2.1}$$



(a) (b)

(c) (d)

Figure 2.4: Networks in real world: (a) A social network. (b) A citation network. (c) A food web. (d) Computer network. Source: (Internet)

2.2.10 Definition (Vertex adjacency)**.** For a given network, two vertices v_i and v_j are adjacent if there exists an edge, e, connecting the two vertices, that is, $e = \{v_i, v_j\}$. With the understanding of adjacency, we can represent a network using a matrix known as adjacency matrix $\mathbf{A}$ (Newman, 2010).

2.2.11 Adjacency matrix. For a simple finite network $G = (V, E)$, the adjacency matrix $\mathbf{A(G)}$ is an $n \times n$ matrix with entries such that

$$A_{ij} = \begin{cases} 1 & \text{if there is an edge between vertices } v_i \text{ and } v_j, \\ 0 & \text{otherwise.} \end{cases} \tag{2.2.2}$$

The diagonal entries of $\mathbf{A(G)}$ are zeroes and if the network is undirected, $\mathbf{A(G)}$ is a symmetric matrix (Newman, 2010).

2.2.12 Definition (Neighborhood ($N_G(v)$))**.** The neighborhood of a vertex $v \in V$ is a set of all vertices that are adjacent to v (Newman, 2010). Mathematically, $N_G(v) = \{u \in V | uv \in E\}$

2.2.13 Definition (Degree of a node (k_v))**.** The degree of a vertex v is the number of edges incident to it. A self-edge is counted as two edges. The degree of a node v is the number of nearest neighbors of v, that is, $k_v = |N_G(v)|$. If $k_v = 0$, then node v is said to be isolated in G, and if $k_v = 1$, then v is a leaf of the graph. The minimum degree $k_{min}(G) = min\{k_v | v \in G\}$ and the maximum degree $k_{max}(G) = max\{k_v | v \in G\}$. For a directed network, we consider two types of degrees, namely in-degree (k_v^{in}) and the out-degree (k_v^{out}), which are the number of edges pointing towards or departing from a node v respectively (Estrada, 2011). The total degree k_v is $k_v = k_v^{in} + k_v^{out}$.

2.2.14 Definition (Degree matrix)**.** The degree matrix is a diagonal matrix that provides information about the degree of each node in a given network (Newman, 2010). Given a network $G = (V, E)$ with $n = |V|$, the degree matrix $\mathbf{D}(\mathbf{G})$ is defined as

$$D_{i,j} = \begin{cases} k_i & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases} \tag{2.2.3}$$

2.2.15 Lemma (Handshaking)**.** For any given undirected network $G = (V, E)$, the sum of all vertex degrees is equal to twice the number of edges (Newman, 2010).

$$\sum_{v \in V} k_v = 2\ |E|. \tag{2.2.4}$$

2.2.16 Example. In this example, we compute the incidence, adjacency and degree matrices of the simple graph in Fig. 2.5.

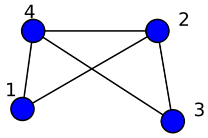


Figure 2.5: A simple graph G.

$$\mathbf{B}(\mathbf{G}) = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & -1 \\ -1 & 0 & -1 & -1 & 0 \end{pmatrix}, \quad \mathbf{A}(\mathbf{G}) = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}, \text{ and } \mathbf{D}(\mathbf{G}) = \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 3 \end{pmatrix}.$$

2.2.17 Definition (Walk)**.** A walk in a network is a series of edges (not necessarily distinct)

$$(u_1, v_1), (u_2, v_2), \dots, (u_k, v_k),$$

for which $v_i = u_{i+1}$ $(i = 1, 2, \dots, k-1)$. A trail is a walk in which all the edges are distinct (Estrada et al., 2015). A walk of length k is referred to as a k-walk. We can compute the number of k-walks between any pair of nodes in a network using the entries of $\mathbf{A}^k$.

2.2.18 Example. Let us compute the number of 2-walks and 3-walks between pairs of nodes in network in Fig. 2.5.

$$\mathbf{A}^2 = \begin{pmatrix} 2 & 1 & 2 & 1 \\ 1 & 3 & 2 & 2 \\ 2 & 1 & 2 & 1 \\ 1 & 2 & 1 & 3 \end{pmatrix}, \quad \mathbf{A}^3 = \begin{pmatrix} 2 & 5 & 2 & 5 \\ 5 & 4 & 5 & 5 \\ 2 & 5 & 2 & 5 \\ 5 & 5 & 5 & 4 \end{pmatrix}$$

From the matrix $\mathbf{A}^2$, we can tell that there are 2 walks of length 2 between nodes 1 and 3, 1 walk of length 2 between nodes 3 and 4, 2 walks of length 2 between nodes 2 and 4, etc. From matrix $\mathbf{A}^3$, there are 2 walks of length 3 between nodes 1 and 3, node 3 to it self, and node 1 to it self. There are 5 walks of length 3 between nodes 2 and 3, nodes 2 and 4, etc.

2.2.19 Definition (Path)**.** A path of length l is a walk of length l in which all the nodes and edges are distinct. A closed path is called a cycle (Estrada, 2011).

2.2.20 Definition (Connected component of a graph)**.** A component of an undirected graph is a subgraph in which any two vertices are connected to each other by paths, and which is connected to no additional vertices in the supergraph (Newman, 2010). A connected component is also referred to as a maximal connected subgraph of a graph.

2.2.21 Definition (Tree)**.** A tree is a connected undirected graph that contains no closed loops (Newman, 2010). It is important to note that a tree with n vertices has $(n-1)$ edges. Fig. 2.6(b) illustrates a tree with 6 vertices and 5 edges.

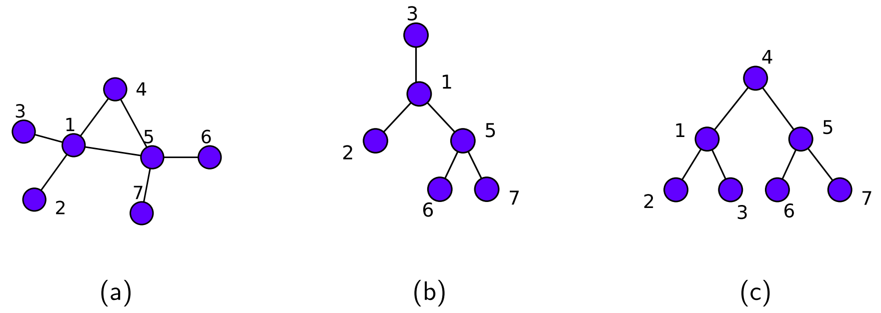


Figure 2.6: (a) A graph (b) along with its tree (c) and spanning tree (c).

2.2.22 Definition (Spanning tree)**.** A spanning tree of a graph $G = (V, E)$ is a subgraph of G with vertex set V, which is a tree. G has a spanning tree if and only if G is connected (Newman, 2010). Fig. 2.6(c) is a spanning tree of the network in Fig 2.6(a).

There are various categories of networks some of which we discuss below.

2.2.23 Definition (Complete network)**.** A network $G = K_{V(G)}$ is a complete network on $V(G)$, if every two nodes are adjacent: $E = E(G)$. We denote a complete network of order n by K_n. For instance, Fig. 2.7(a) is a complete network of size 6, K_6.

2.2.24 Definition (Regular network)**.** A regular network is a network G in which every node has the same degree. A k-regular network is one in which every node has degree equal to k. Fig. 2.7(b) illustrates a 3-regular network.

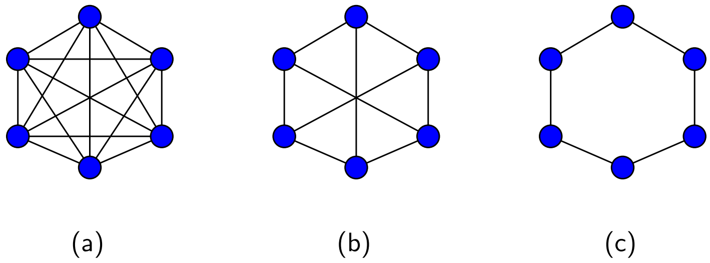


Figure 2.7: (a) A complete network with 6 nodes. (b) A 3-regular network. (c) A cycle network C_6.

2.2.25 Definition (Cycle)**.** A cycle network is a connected network in which there exists an edge connecting one node to another and each node has degree 2. A cycle with n nodes is denoted as C_n (Wilson, 1970). Fig. 2.7(c) is a cycle of size 6, denoted as C_6.

2.2.26 Definition (Bipartite)**.** A network $G = (V, E)$ is bipartite if the nodes can be divided into disjoint sets V_1 and V_2 such that $(u, v) \in E$ implies that $u \in V_i$, $v \in V_j$, $i \neq j$. A bipartite network in which each node of V_1 is connected to each node of V_2 is known as a complete bipartite network; if $|V_1| = m$ and $|V_2| = n$, such a network is denoted by $K_{m,n}$ (Newman, 2010).

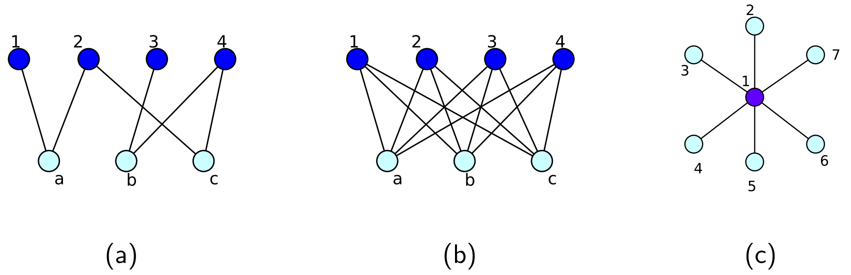


Figure 2.8: (a) A bipartitte network with $m = 4$ and $n = 3$. (b) A complete bipartite network, $K_{4,3}$. (c) A star network, S_6.

2.2.27 Definition (Star network)**.** A star network S_n is a complete bipartite network $K_{1,n}$ (Wilson, 1970). Fig. 2.8(c) illustrates a star network with $n = 6$.

2.3 Characteristics of Networks

2.3.1 Degree distributions. The scattering of node degrees over a network is characterised by the distribution function, $p(k)$, which is the probability that a node chosen uniformly at random has degree k. We define $p(k)$ to be the fraction of nodes in a network that have degree k. That is, $p(k) = n(k)/n$, where $n(k)$ is the number of nodes with degree k in a network of size n. The degree distribution of a network is referred to as the probability distribution of node degrees over that network. It is represented by plot of $p(k)$ against n (Estrada, 2011).

2.3.2 Example. We compute the degree distribution for the network in Fig. 2.9.

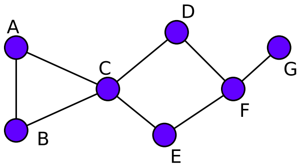
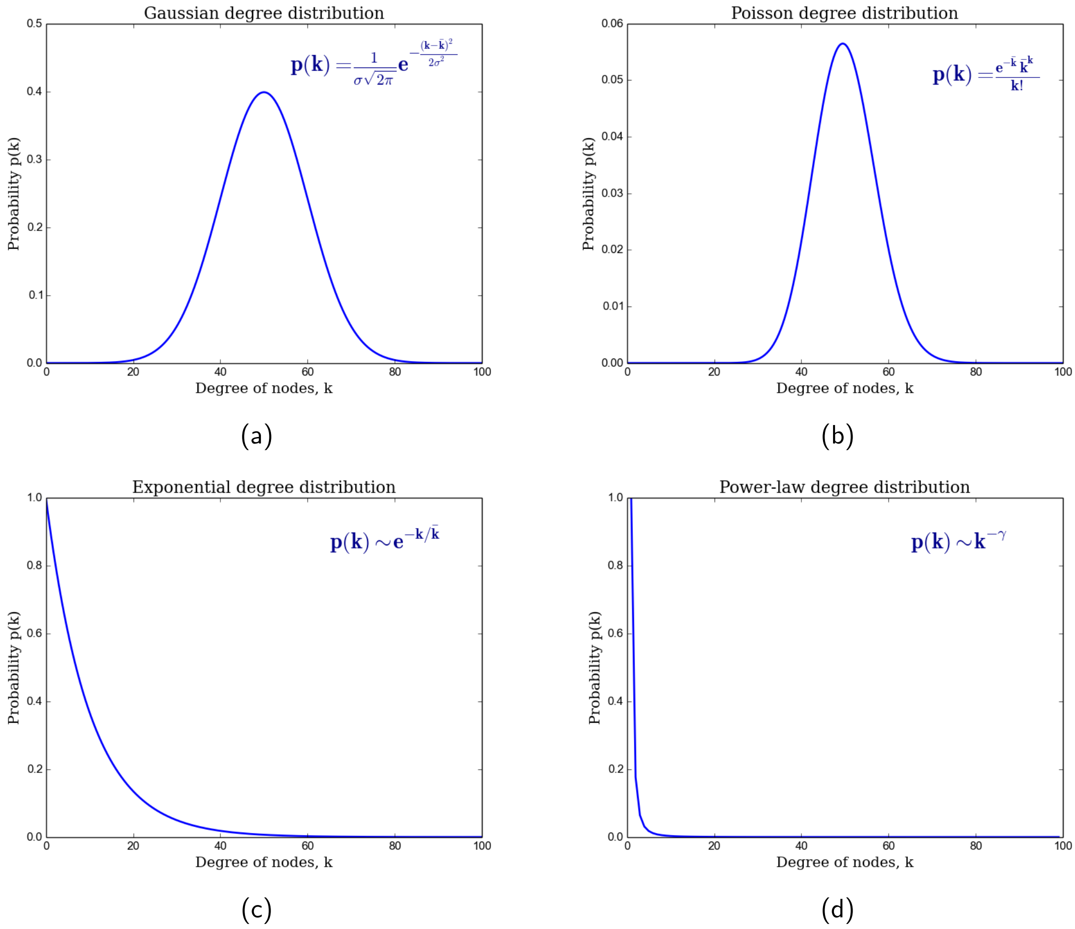


Figure 2.9: A network of size 7.

Table 2.1: Degree distribution of nodes in network in Fig. 2.9.

Node	Degree, k	$p(k)$
G	1	$1/7$
A,B,D,E	2	$4/7$
F	3	$1/7$
C	4	$1/7$

We use the degree distribution of a network to obtain useful insights about the structure of a network. The structure of a network is a description of how nodes are linked to each other to form a network. It is important to note that networks with different structures can have the same degree distribution which implies that degree distribution gives us some but not all the information regarding the structure of a network. Thus, we cannot deduce a complete structure of a network based on information about its degree distribution alone. Fig. 2.10 shows some of the common degree distributions in networks.



(a)

(b)

(c)

(d)

Figure 2.10: Common degree distributions of networks: (a) Gaussian distribution. (b) Poisson distribution. (c) Exponential distribution.(d) Power-law distribution.

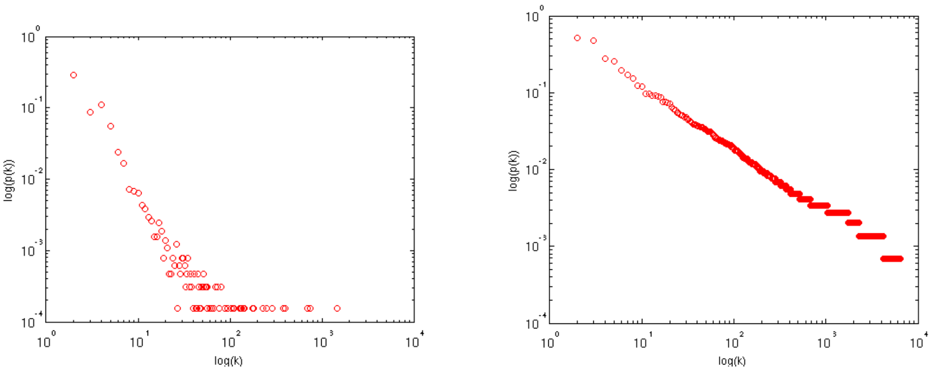
2.3.3 Power-law degree distribution. Research shows that most of the real-world networks roughly follow a power-law degree distribution. In this distribution, the probability of finding a node with degree k decreases as a negative power of degree k. This implies that in such networks, it is less likely to find a node with high degree (Estrada, 2011). Formally,

$$p(k) = Ck^{-\gamma}, \text{ for } 2 \leq \gamma \leq 3. \tag{2.3.1}$$

Using a logarithmic scale, the plot of Equation (2.3.1) is a straight line, $\ln p(k) = -\gamma \ln k + \ln C$, with a slope equal to $-\gamma$ and an intercept equal to $\ln C$ as illustrated in Fig. 2.11(a). However, we observe that the part that corresponds to high degrees (tail of the distribution) is very noisy. In order to overcome this problem, one of the solutions is to consider the cumulative distribution function, which is defined as

$$P(k) = \sum_{k'=k}^{\infty} p(k'),$$

which represents the probability of randomly choosing a node with degree k or greater (Estrada, 2011). The plot of cummulative distribution function on a logarithmic plot is a straight line as shown in Fig. 2.11(b).



(a) (b)

Figure 2.11: Probability (a) and Cummulative Distribution Function (b) logarithmic plots for the version of the internet at autonomous system (AS) level following a power-law distribution. Source: Mutombo (2012).

When we scale the degree by a constant factor a, we obtain

$$p(k,a) = C(ak)^{-\gamma} = a^{-\gamma}p(k) \propto p(k). \tag{2.3.2}$$

Thus, scaling by a constant a multiplies the original power-law relation by the constant $a^{-\gamma}$ which implies that all power laws with a particular scaling factor are scaled versions of each other. Thus, networks that follow a power-law distribution are referred to as scale-free networks.

2.3.4 Clustering coefficient. Clustering coefficient is a measure of the degree to which nodes tend to cluster together. Such behaviour is more evident in real-world networks, especially social networks where nodes tend to form tightly knit groups that have relatively high density of ties among them (Estrada et al., 2015). Consider three nodes in a network, say, i, j and k. Suppose i is connected to both j and k (two neighbors of a node will be neighbors themselves), then the likelihood that j and k are also connected is what is known as the clustering coefficient. In other words, clustering coefficient measures the density of triangles in a network. The value of clustering coefficient is in the interval $[0, 1]$. There are two types of clustering coefficients namely, the local and the global clustering coefficients.

2.3.5 Definition (Local clustering coefficient)**.** The local clustering coefficient is a measure of the clustering tendency in a node's immediate network. The local clustering coefficient for a node i with degree k_i is formally defined as

$$C_i = \frac{\text{number of pairs of neighbors of } i \text{ that are connected}}{\text{number of pairs of neighbors of } i} = \frac{2t_i}{k_i(k_i-1)}, \tag{2.3.3}$$

where t_i is the number of triangles attached to node i. For nodes with degree equal to zero or one, we set $C_i = 0$ since there are no triangles attached to such nodes (Newman, 2010). The average clustering coefficient for the network is given by

$$\bar{C} = \frac{1}{n}\sum_i C_i. \tag{2.3.4}$$

2.3.6 Definition (Global clustering coefficient)**.** The global clustering coefficient is concerned with the density of triplets of nodes in a network. A triplet is defined as three nodes that are connected by either two (open triplet) or three (closed triplet) ties. Global clustering coefficient determines the overall level of clustering in a network (Opsahl and Panzarasa, 2009). Mathematically, we define the global clustering coefficient C as

$$C = \frac{3 \times \text{number of triangles}}{\text{number of connected triplets of vertices}} = \frac{\sum t_{\triangle}}{\sum t}, \tag{2.3.5}$$

where $\sum t_{\triangle}$ is the total number of closed triplets and $\sum t$ is the total number of connected triplets of vertices in the network.

2.3.7 Example. Let us compute the local, global and average clustering coefficients for the network in Fig. 2.12.

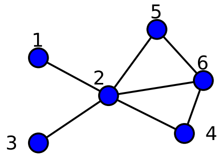


Figure 2.12: A network of size 6.

i) Local clustering coefficients C_i:

For nodes 1 and 3: $C_1 = C_3 = 0$, nodes 4 and 5: $C_4 = C_5 = 1$, node 2: $C_2 = 4/20 = 0.2$, and node 6: $C_6 = 4/6 = 0.6667$.

ii) Average clustering coefficient, $\bar{C}$:

$$\bar{C} = \frac{1}{6}\left(\frac{1}{5} + \frac{2}{3} + 1 + 1\right) = 0.4778.$$

iii) Global clustering coefficient, C:

Number of triangles $= 2$, number of connected triplets $= 9 + (2 \times 3) = 15$. Thus,

$$C = \frac{3 \times 2}{15} = 0.4.$$

2.3.8 Definition (Distance between a pair of nodes)**.** In a network, the distance d_{ij} between two nodes, labelled i and j respectively, is defined as the length of the shortest path (or geodesic path) connecting them (Wang and Chen, 2003). It is also known as geodesic distance. It is possible to have more than one shortest paths between a pair of nodes.

2.3.9 Definition (Diameter of a network)**.** The diameter D of a network is the maximum distance between any two nodes in the network (Wang and Chen, 2003). The diameter of a graph $G = (V, E)$ is defined as

$$\text{diam}(G) = \max_{i,j \in V} d_{ij}.$$

For a disconnected network, the diameter is undefined and therefore, for such a case, we take the efficiency of a network which is a finite value given by

$$\bar{e} = \frac{1}{n(n-1)} \sum_{i,j \in V, i \neq j} \frac{1}{d_{ij}}.$$

2.3.10 Definition (Average path length)**.** The average path length of a network is the average number of steps along the shortest paths for all possible pairs of network nodes. Let $G = (V, E)$ be a graph the average path length L_G is defined by

$$L_G = \frac{1}{n(n-1)} \sum_{i,j \in V,\ i \neq j} d_{ij}, \tag{2.3.6}$$

where d_{ij} is the shortest path between node i and j and n is the total number of nodes in G. The value of L determines the size of a network and helps to determine the efficiency of information flow or disease spread over a network (Wang and Chen, 2003).

2.3.11 Centrality. In network analysis, it is vital to identify the most central or 'important' node in a network (Qi et al., 2012). However, the basis for determining the importance of the node depends on a number of reasons. The centrality of a node has a wide range of applications in the real-world such as, in prevention of a computer network from breaking down, control of the spread of a disease, faster dissemination of information over a social network, etc. We use various methods to compute the centrality of a given node. Some of the most common centralities include:

2.3.12 Definition (Degree centrality)**.** Degree centrality is the number of edges incident to a node. It is the simplest centrality measure and it is highly effective in determining the most important node especially in social networks (Qi et al., 2012). However, a weakness of the degree centrality measure is that it only captures information about the local structure around a node leaving out the global structure. Mathematically, the degree centrality $C_D(i)$ of a node i is given by

$$C_D(i) = \sum_j A_{i,j}.$$

For comparison purposes, we normalise the degree centrality by dividing it by the maximum centrality value possible; $n - 1$.

2.3.13 Definition (Closeness centrality)**.** Closeness centrality is a centrality measure in which the importance of a node is determined by how close the given node is to its neighbors. In other words, how easily a node is reachable. This centrality helps in determining how fast it is to get to other nodes from a particular node and so closeness centrality can provide information regarding how fast an epidermic or information can spread over a given network. Closeness centrality captures information regarding the global network structure. It is defined as the inverse sum of shortest distances from the focal node i to all other nodes in a network, that is,

$$C_C(i) = \frac{1}{\sum_j d_{ij}}, \tag{2.3.7}$$

where d_{ij} is the shortest distance between nodes i and j (Qi et al., 2012). The normalised closeness centrality is given by $C'_C(i) = (n-1)C_C(i)$.

2.3.14 Definition (Betweenness centrality)**.** This type of centrality determines the node that acts as a bridge to a pair of nodes. In other words, it determines which node lies at the shortest path between pairs of nodes in a network. In social networks, betweenness centrality measures the control of an actor in communication between two other friends. We define the betweenness centrality as

$$C_B(i) = \sum_{h \neq i} \sum_{j \neq i, j \neq h} \frac{L_{h,j}(i)}{L_{h,j}}, \tag{2.3.8}$$

where $L_{h,j}$ is the total number of shortest paths between nodes h and j, $L_{h,j}(i)$ is the number of shortest paths that pass through node i (Qi et al., 2012). The normalised betweenness centrality is given as

$$C'_B(i) = \frac{2}{(n-1)(n-2)} C_B(i). \tag{2.3.9}$$

The betweenness centrality of a node gives information about the global position of a node in a network and it can be applied to disconnected components. However, one of the limitations of betweenness centrality is that all nodes that do not lie on the shortest path between any other two vertices are assigned the same score of zero.

2.3.15 Definition (Eigenvector centrality)**.** This type of centrality is based on the quality of connections to a particular node. Eigenvector centrality computes the centrality of a node as a function of the centralities of its neighbors. It assigns relative scores to all nodes in the network based on the principle that connections to high scoring nodes contribute more to the score of the node than equal nodes to low-scoring node. The eigenvector centrality of a node i, denoted as $C_E(i)$, is defined as

$$C_E(i) \propto \sum_{j \neq i} A_{ij} C_E(j), \tag{2.3.10}$$

where A_{ij} is an element of the adjacency matrix of the network (Qi et al., 2012).

Suppose that we have an initial value for all $x_i(0)$. Then, we compute next iteration of values using the formula

$$\mathbf{x}(t+1) = \mathbf{A}\mathbf{x}(t) = \mathbf{A}^t \mathbf{x}(0).$$

Let us express $\mathbf{x}(0)$ in terms of the eigenvectors $\mathbf{v}_i$ of $\mathbf{A}$,

$$\mathbf{x}(0) = \sum_i c_i \mathbf{v}_i.$$

Let λ_i be the eigenvalues of $\mathbf{A}$ and λ_1 be the spectral radius of $\mathbf{A}$,

$$\mathbf{x}(t) = \mathbf{A}^t \mathbf{x}(0) = \sum_i c_i \lambda_i^t \mathbf{v}_i = \lambda_1^t \sum_i c_i \left(\frac{\lambda_i}{\lambda_1} \right)^t \mathbf{v}_i, \tag{2.3.11}$$

since $\frac{\lambda_i}{\lambda_1} < 1$ for all $i > 1$, all terms (other than the first) decay exponentially as t grows. Therefore,

$$\mathbf{x}(t) \to c_1 \lambda_1^t \mathbf{v_1} \text{ as } t \to \infty.$$

Eigenvector centrality is proportional to the leading eigenvector of $\mathbf{A}$. Equivalently, define centrality vector x satisfying: $\mathbf{A}\mathbf{x} = \lambda_1 \mathbf{x}$, then $C_E(i) = \mathbf{x}(i)$ (Newman, 2010).

2.3.16 Definition (Subgraph centrality)**.** This type of centrality was presented by Ernesto and Rodriguez-Velazquez. With this centrality measure, the importance of a node is characterized by its participation in all subgraphs in a network. We define the subgraph centrality $C_S(i)$ of a node i as

$$C_S(i) = \sum_{k=0}^{\infty} \frac{u_k(i,i)}{k!} = \sum_{k=0}^{\infty} \frac{\mathbf{A}^k(i,i)}{k!} = (e^{\mathbf{A}})_{ii}, \tag{2.3.12}$$

where $u_k(i,i)$ is the number of closed k-walks that vertex i participates in, in the network, and $\mathbf{A}$ is the adjacency matrix of the network (Estrada and Rodriguez-Velazquez, 2005). It is sometimes desirable to normalize the subgraph centrality of a node by the sum

$$EE(G) = \sum_{i=1}^{N} C_S(i) = \sum_{i=1}^{N} (e^{\mathbf{A}})_{ii} = Tr(e^{\mathbf{A}}) = \sum_{i=1}^{N} e^{\lambda_i}$$

of all the subgraph centralities. The quantity $EE(G)$ is known as the Estrada index of the graph G.

2.3.17 Example. We compute the degree, closeness, betweenness, eigenvector and subgraph centralities for nodes of the network in Fig. 2.13.

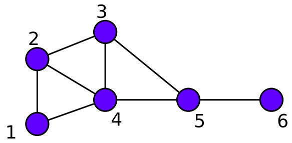


Figure 2.13: A network G.

Table 2.2: Centralities for nodes of the network in Fig. 2.13.

Node	Degree	Closeness	Betweenness	Eigenvector	Subgraph
1	0.4	0.5556	0.00	0.3363	0.1302
2	0.6	0.6250	0.05	0.4531	0.1911
3	0.6	0.7143	0.10	0.4649	0.1942
4	0.8	0.8333	0.35	0.5494	0.2528
5	0.6	0.7143	0.40	0.3834	0.1632
6	0.2	0.4545	0.00	0.1286	0.0685

The results in Table 2.2 show that node 4 has the highest scores based on all centrality measures except the betweenness centrality as expected. Based on betweenness, we observe that node 5 has the highest score, this is because node 5 acts as a bridge between node 6 and all other nodes. Nodes 1 and 6 have a score of zero based on betweenness centrality since they do not lie on any shortest path between any pair of nodes in the network.

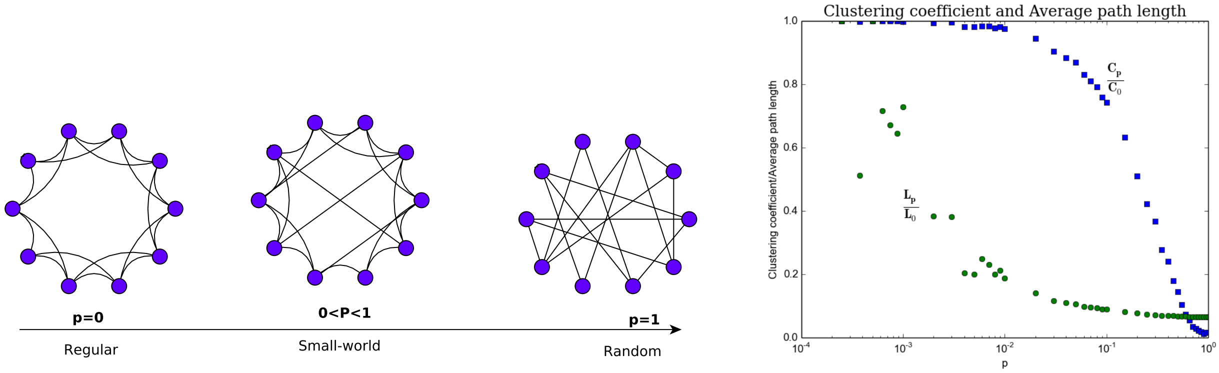
2.4 Network Models

2.4.1 Random models of networks. In the study of real-world networks (such as protein-protein interaction, transport network, food web, etc.), we observe different topological structures of these networks and it is of great importance to understand the mechanisms that are responsible for such structures. We therefore discuss some of the models that mimic real-world networks.

2.4.2 The Erdös-Rényi (ER) model. This type of random model was introduced by Erdös and Rényi in 1959. It is the best known model for random networks. In this model, we take n isolated nodes and then fix a probability p with which we link the nodes. For each pair of nodes, we generate a random number, r, uniformly from $[0,1]$. If $p > r$, the two nodes are connected forming a network. Networks generated by the ER model have a small average path length and their average clustering coefficient tends to zero as n increases (Estrada, 2011).

2.4.3 The Barabási-Albert (BA) model. Most of the real-world networks follow a power-law degree distribution. This distribution was not manifested in ER networks and thus, Barabási and Albert introduced a network model that generates networks that mimic the degree distribution observed in the real-world networks. This model uses a preferential attachment, or rich get richer, mechanism to evolve a given initial graph. The preferential attachment technique involves an initial network with n_0 nodes, and at each step, a new node is introduced and connected to the existing nodes with a probability proportional to their degrees (or connectivity). The resulting networks are scale-free and have high clustering coefficients (Estrada, 2011).

2.4.4 The Watts-Strogatz (WS) model. An experiment carried out by Stanley Milgram in 1967 (Milgram, 1967) consisted of randomly selected people in the cities of Omaha and Wichita in the United States of America (USA) who were asked to send letters to target persons living in Boston. The individuals at the starting points were asked to send the letters to persons they knew. Despite the fact that the senders and their respective targets were separated by more than 2000km and that there was a total of 200 million inhabitants in the USA by then, results showed that the letters took an average of about six steps to reach the targets and that there was a large group interconnection where an acquaintance of an individual fed back into his own circle, thus eliminating new contacts. In terms of networks, two properties are observed: the former implies a small average path length of 6 while the latter implies a high clustering coefficient. It is from this experiment that the phrase "six degrees of separation" was born. This phrase explains a small world in which an individual is connected to another by a maximum of six steps. In 1998, Watts and Strogatz (1998) proposed a model that reproduces the two properties mentioned previously. This model starts with a circulant (or ring) network with n nodes connected to k nearest neighbors. With a fixed probability p, an end to each original link is rewired to a new randomly selected node. The WS model interpolates between a regular and a random network as shown in Fig. 2.14(a). The intermediate network is a small-world with high clustering coefficient and a very small average path length.



(a) (b)

Figure 2.14: The rewiring process: (a) Interpolation of WS model as probability increases. (b) Illustration of the variation of clustering coefficient and average path length during the rewiring process.

In Fig. 2.14(b), as the probability increases during the rewiring process, the average path length and clustering coefficient vary. The region where the average path length is small and the clustering coefficient is high corresponds to the small world network in Fig. 2.14(a).

3. Laplacian Matrix of a Network

The Laplacian matrix, sometimes called admittance matrix or Kirchhoff matrix, is a matrix representation of a network. The Laplacian matrix provides useful information about the properties of a network.

3.1 Definitions and Properties of the Laplacian Matrix

3.1.1 Definition (Laplacian Matrix)**.** The Laplacian matrix of a network is the difference between the degree matrix $\mathbf{D}$ and the adjacency matrix $\mathbf{A}$ of a network. That is,

$$\mathbf{L} = \mathbf{D} - \mathbf{A}. \tag{3.1.1}$$

Given a simple network $G = (V, E)$, the entries of the Laplacian matrix $\mathbf{L(G)}$ are defined as

$$L_{ij} = \begin{cases} k_{v_i} & \text{if } i = j \\ -1 & \text{if } i \neq j \text{ and } v_i \text{ is adjacent to } v_j \\ 0 & \text{otherwise,} \end{cases} \tag{3.1.2}$$

where k_{v_i} denotes the degree of node i (Estrada, 2011).

Alternatively, we can define the Laplacian matrix of a graph in terms of the vertex-edge incidence matrix $\mathbf{B}$. That is,

$$\mathbf{L} = \mathbf{B}\mathbf{B}^T, \tag{3.1.3}$$

where $\mathbf{B}^T$ is the transpose of $\mathbf{B}$ (Estrada, 2011).

3.1.2 Example. We compute the Laplacian matrix of the network in Fig. 3.1.

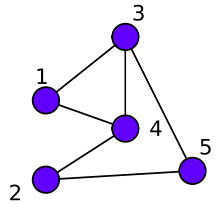


Figure 3.1: A Network G.

$$\mathbf{L} = \begin{pmatrix} 2 & 0 & -1 & -1 & 0 \\ 0 & 2 & 0 & -1 & -1 \\ -1 & 0 & 3 & -1 & -1 \\ -1 & -1 & -1 & 3 & 0 \\ 0 & -1 & -1 & 0 & 2 \end{pmatrix}.$$

The properties of the Laplacian matrix of a network are very useful in understanding the structure of a network. We therefore discuss some of these properties.

3.1.3 Real and symmetric matrix. The entries of the Laplacian matrix are real numbers and are symmetric with respect to the main diagonal (Das, 2004).

3.1.4 Singular matrix. The Laplacian matrix is a square matrix that is not invertible. Its determinant is equal to zero (Das, 2004).

3.1.5 Positive semi-definite. A matrix is positive semi-definite if and only if all its eigenvalues are non-negative. Suppose the eigenvalues of a Laplacian matrix are: $\lambda_1, \lambda_2, \ldots, \lambda_n$, we show that $\lambda_i \geq 0 \; \forall \; i$.

Proof. Let $\mathbf{v}_i$ be a normalised eigenvector of $\mathbf{L}$ with eigenvalue λ_i. Then, using Equation (3.1.3)

$$\mathbf{v}_i^T \mathbf{B}\mathbf{B}^T \mathbf{v}_i \quad = \quad \mathbf{v}_i^T \mathbf{L} \mathbf{v}_i = \lambda_i \mathbf{v}_i^T \mathbf{v}_i = \lambda_i.$$

Thus, any eigenvalue λ_i of the Laplacian matrix is equal to $(\mathbf{v}_i^T \mathbf{B}\mathbf{B}^T \mathbf{v}_i)$ which can be written as $(\mathbf{v}_i \mathbf{B}^T)^T (\mathbf{B}^T \mathbf{v}_i)$. The quantity $(\mathbf{v}_i \mathbf{B}^T)^T (\mathbf{B}^T \mathbf{v}_i)$ is the inner product of a real vector $(\mathbf{B}^T \mathbf{v}_i)$ with itself, that is, the sum of the squares of the real elements of the vector $(\mathbf{B}^T \mathbf{v}_i)$ and hence it is non-negative (Estrada et al., 2015). □

3.2 Spectrum of the Laplacian Matrix

The spectrum of the Laplacian provides useful information about the structure of a network. The spectrum of the Laplacian matrix is the set of all its eigenvalues and their multiplicities (Estrada, 2011). Let $\lambda_1 < \lambda_2 < \ldots < \lambda_n$ be the distinct eigenvalues of $\mathbf{L}$ and let $m(\lambda_1), m(\lambda_2), \ldots, m(\lambda_n)$ be their multiplicities. Then, the spectrum of $\mathbf{L}$ is written as

$$Sp\mathbf{L} = \begin{pmatrix} \lambda_1 & \lambda_2 & \ldots & \lambda_n \\ m(\lambda_1) & m(\lambda_2) & \ldots & m(\lambda_n). \end{pmatrix} \tag{3.2.1}$$

Let us write the eigenvalues of $\mathbf{L}$ in decreasing order: $\lambda_n \geq \lambda_{n-1} \geq \cdots \geq \lambda_2 \geq \lambda_1 = 0$. Some of the results associated with the spectrum of the Laplacian matrix include:

- The eigenvalues of $\mathbf{L}$ are bounded as $0 \leq \lambda_j \leq 2k_{max} \quad \text{and} \quad \lambda_n \geq k_{max}$.
- The eigenvalue λ_1 is always equal to zero (Estrada, 2011).
- The multiplicity of 0 as an eigenvalue of $\mathbf{L}$ is equal to the number of connected components in the network (Estrada, 2011).
- Every row sum and column sum of $\mathbf{L}$ is zero. Thus, the vector $\mathbf{v_1}$ of all ones is an eigenvector associated with $\lambda_1 = 0$, since $\mathbf{L}\mathbf{v_1} = \mathbf{0}$ (Das, 2004).
- A network is connected if its second smallest eigenvalue is nonzero. That is, $\lambda_2 > 0$ if and only if G is connected. The eigenvalue λ_2 is thus called the algebraic connectivity of a network, $a(G)$. The eigenvector corresponding to the eigenvalue λ_2 is called the Fiedler vector (Estrada et al., 2015).

3.2.1 Theorem (Fiedler, 1975)**.** *Suppose $G = (V, E)$ is a connected network with graph Laplacian $\mathbf{L}$ whose second smallest eigenvalue is $\lambda_2 > 0$. Let x be the eigenvector associated with λ_2. Let $r \in \mathbb{R}$ and partition the nodes in V into two sets*

$$V_1 = \{i \in V | x_i \geq r\}, \; V_2 = \{i \in V | x_i < r\}, \tag{3.2.2}$$

then the subgraphs of G induced by the sets V_1 and V_2 are connected (Estrada et al., 2015).

This result is useful for partitioning a network while ensuring that all the parts remain connected. This method of partitioning using eigenvalues is known as spectral clustering. For clusters of equal size, we choose r such that it is the median value of x.

3.2.2 Example. Using Theorem 3.2.1, let us partition the network in Fig. 3.2(a) into two clusters. First, we compute the eigenvalues and corresponding eigenvectors of the Laplacian matrix $\mathbf{L}$ of the network which are: $\lambda_1 = 0, \lambda_2 = 2, \lambda_3 = 4, \lambda_4 = 4$. The second smallest non-zero eigenvalue $\lambda_2 = 2$ is associated with eigenvector $\mathbf{v_2} = (1, 0, 0, -1)^T$ which implies that $x_1 = 1, x_2 = 0, x_3 = 0$ and $x_4 = -1$. Taking $r = 0$, we obtain the two vertex sets V_1 and V_2 as $V_1 = \{1, 2, 3\}$ and $V_2 = \{4\}$. The two clusters are shown in Fig. 3.2(b).

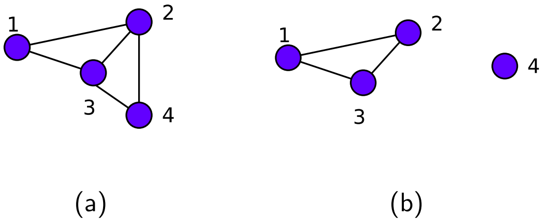


Figure 3.2: (a) A network (b) and its two clusters of size 3 and 1.

Some analytic expressions for the spectra of different kinds of simple networks are:

- Star, S_n : $Sp(\mathbf{L}) = \{0 \; 1^{n-2} \; n\}$.
- Complete, K_n : $Sp(\mathbf{L}) = \{0 \; n^{n-1}\}$.
- Complete bipartite, $K_{m,n}$: $Sp(\mathbf{L}) = \{0 \; m^{n-1} \; n^{m-1} \; m + n\}$(Estrada et al., 2015) .

3.2.3 Theorem (Kirchoff's Matrix-Tree Theorem)**.** *If G is a connected graph with Laplacian matrix $\mathbf{L}$, then the number of unique spanning trees of G is equal to the value of any cofactor of the matrix $\mathbf{L}$ (Harris et al., 2008).*

3.2.4 Example. Let us compute the number of spanning trees of network G in Fig. 3.2(a) using Theorem 3.2.3. The (i, j) cofactor is obtained as $(-1)^{i+j} \times |\mathbf{L}(i|j)|$ where $\mathbf{L}(i|j)$ is the $(n-1)\times(n-1)$ submatrix of $\mathbf{L(G)}$ obtained by deleting its ith row and jth column. The Laplace matrix is given by

$$\mathbf{L(G)} = \begin{pmatrix} 2 & -1 & -1 & 0 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 3 & -1 \\ 0 & -1 & -1 & 2 \end{pmatrix}, \text{ deleting row 1 and column 1 gives } \mathbf{L}(1|1) = \begin{pmatrix} 3 & -1 & -1 \\ -1 & 3 & -1 \\ -1 & -1 & 2 \end{pmatrix}.$$

The $(1, 1)$ cofactor of $\mathbf{L(G)}$ is $(-1)^2 \times |\mathbf{L}(1|1)| = 8$. Thus, there are 8 spanning trees of graph G.

4. Applications of Laplacian Matrix of a Network

4.1 Laplacian Centrality

In chapter 2, we discussed various centrality measurements for unweighted networks. These include: degree, closeness, betweenness, eigenvector and subgraph centralities. These centrality measures were extended to weighted networks (Newman, 2001); (Barrat et al., 2004); (Opsahl, 2009); (Opsahl et al., 2010). These standard centrality measures give information on either the local environment of a node (i.e degree centrality) or the global position of the node in the network (i.e closeness, betweenness and subgraph centralities). This implies that information about the intermediate (between local and global) environment of a node cannot be captured by any of the standard centralities, yet, such information is very useful in the study of real-world networks. For instance, quantifying the relative importance of a particular actor in a social network. It is for this reason that a new type of centrality known as the Laplacian centrality was introduced (Qi et al., 2012).

With Laplacian centrality measure, the importance of a node is determined by the ability of the network to respond to the deactivation of the node from the network. In other words, it is a measure of the relative drop of Laplacian energy in the network due to the removal (or deactivation) of the node from the network. The drop of Laplacian energy with respect to node v is determined by the number of 2-walks that v participates in in the network.

4.1.1 Laplacian energy of a network. Let $G = (V, E, W)$ be a simple undirected weighted network with the vertex set $V(G) = \{v_1, v_2, \cdots, v_n\}$, edge set E, where each edge $e = (v_i, v_j)$ is attached with a weight w_{ij}. If there is no edge between v_i and v_j, $w_{i,j} = 0$. In addition, $w_{i,i} = 0$ and $w_{i,j} = w_{j,i}$. We define

$$\mathbf{W(G)} = \begin{pmatrix} 0 & w_{1,2} & \ldots & w_{1,n} \\ w_{2,1} & 0 & \ldots & w_{2,n} \\ \vdots & \vdots & \vdots & \vdots \\ w_{n,1} & w_{n,2} & \ldots & 0 \end{pmatrix} \text{ and } \mathbf{X(G)} = \begin{pmatrix} x_1 & 0 & \ldots & 0 \\ 0 & x_2 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \ldots & x_n \end{pmatrix},$$

where x_i is the sum-weight of vertex v_i given by $x_i = \sum_{j=1}^{n} w_{i,j} = \sum_{u \in N(v_i)} w_{v_i,u}$, where $N(v_i)$ is the neighborhood of v_i.

4.1.2 Definition (Weighted Laplacian matrix)**.** The Laplacian matrix of a weighted network G is the matrix $\mathbf{L(G)} = \mathbf{X(G)} - \mathbf{W(G)}$.

4.1.3 Definition (Laplacian Energy of a network)**.** Let $G = (V, E, W)$ be a weighted network on n vertices and $\lambda_1, \lambda_2, \ldots, \lambda_n$ be the eigenvalues of its Laplacian matrix. The Laplacian energy of G is defined as

$$E_L(G) = \sum_{i=1}^{n} \lambda_i^2.$$

As networks become larger, computing eigenvalues of the Laplacian matrix becomes very hard. We therefore, use the entries of the Laplacian matrix rather than its eigenvalues to compute the Laplacian energy of a network as given by Theorem 4.1.4.

4.1.4 Theorem. *For any network $G = (V, E, W)$ on n vertices whose vertex sum-weights are $x_1, x_2, \ldots, x_n$ respectively, we have*

$$E_L(G) = \sum_{i=1}^{n} x_i^2 + 2\sum_{i<j} w_{i,j}^2. \tag{4.1.1}$$

Proof. Let

$$\mathbf{L}(\mathbf{G}) = \begin{pmatrix} x_1 & -w_{1,2} & \ldots & -w_{1,n} \\ -w_{2,1} & x_2 & \ldots & -w_{2,n} \\ \vdots & \vdots & \vdots & \vdots \\ -w_{n,1} & -w_{n,2} & \ldots & x_n \end{pmatrix}.$$

The characteristic polynomial of $\mathbf{L}(\mathbf{G})$ is given by

$$p(\lambda) = \lambda^n + a_{n-1}\lambda^{n-1} + a_{n-2}\lambda^{n-2} + \ldots + a_1\lambda + a_0, \tag{4.1.2}$$

whose roots are $\lambda_1, \lambda_2, \ldots, \lambda_n$. Using Leverrier's method and Newton's formulae (Gower, 1980), the coefficients a_{n-1} and a_{n-2} are given by

$$a_{n-1} = -Tr(\mathbf{L}(\mathbf{G})) = \sum_{i=1}^{n} x_i \tag{4.1.3}$$

and

$$\begin{aligned} a_{n-2} &= \frac{1}{2}[(Tr(\mathbf{L}(\mathbf{G})))^2 - Tr(\mathbf{L}(\mathbf{G})^2)] = \frac{1}{2}\left[\left(\sum_{i=1}^{n} x_i\right)^2 - \left(\sum_{i=1}^{n} x_i^2 + 2\sum_{i<j} w_{ij}^2\right)\right] && (4.1.4) \\ a_{n-2} &= \sum_{i<j} x_i x_j - \sum_{i<j} w_{i,j}^2 && (4.1.5) \end{aligned}$$

where $Tr(\mathbf{L}(\mathbf{G}))$, $Tr(\mathbf{L}(\mathbf{G})^2)$ are the traces of $\mathbf{L}(\mathbf{G})$ and $\mathbf{L}(\mathbf{G})^2$ respectively. Using the Viète rules, we have

$$\sum_{i=1}^{n} \lambda_i = -a_{n-1} \quad \text{and} \quad \sum_{i<j} \lambda_i \lambda_j = a_{n-2},$$

substituting for a_{n-1} and a_{n-2} in Equations (4.1.3) and (4.1.5) respectively, gives

$$\sum_{i=1}^{n} \lambda_i = \sum_{i=1}^{n} x_i \quad \text{and} \quad \sum_{i<j} \lambda_i \lambda_j = \sum_{i<j} x_i x_j - \sum_{i<j} w_{i,j}^2. \tag{4.1.6}$$

Thus,

$$\begin{aligned} E_L(G) &= \sum_{i=1}^{n} \lambda_i^2 = \left(\sum_{i=1}^{n} \lambda_i\right)^2 - \sum_{i\neq j} \lambda_i \lambda_j = \left(\sum_{i=1}^{n} x_i\right)^2 - \sum_{i\neq j} x_i x_j + \sum_{i\neq j} w_{i,j}^2 && (4.1.7) \\ E_L(G) &= \sum_{i=1}^{n} x_i^2 + 2\sum_{i<j} w_{i,j}^2. && (4.1.8) \end{aligned}$$

□

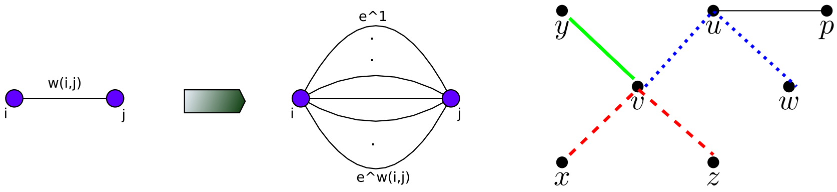
4.1.5 Corollary. If H is an arbitrary subgraph of a network G, then $E_L(H) \leq E_L(G)$. Equality holds if and only if $V(G) - V(H)$ is a set of isolated vertices.

From Theorem 4.1.4, the Laplacian energy is obtained from the edge weights of the network. Since for a subgraph of G, the edge and vertex sets are improper subsets of the edge and vertex sets of network G respectively, then the Laplacian energy of the subgraph is always less or equal to Laplacian energy of G. When $V(G) - V(H)$ is a set of isolated vertices, it implies $E(G) = E(H)$ and so both G and its subgraph have equal edge weights. In addition, the sum of node degrees for G is equal to that of its subgraph. Thus, $E_L(H) = E_L(G)$.

4.1.6 Definition (Laplacian centrality)**.** If $G = (V, E, W)$ is a network with n nodes $V = \{v_1, v_2, \ldots, v_n\}$. Let G_i be the network obtained by deleting v_i from G. The Laplacian centrality is given by

$$C_L(v_i, G) = \frac{(\Delta E)_i}{E_L(G)} = \frac{E_L(G) - E_L(G_i)}{E_L(G)} \tag{4.1.9}$$

For any vertex v, the denominator remains unchanged and from Corollary 4.1.5, we can tell that $E_L(G) - E_L(G_i)$ is non-negative. We then focus on obtaining the expression for $(\Delta E)_i$. In order to obtain the graph theoretical descriptions of Laplacian centrality, we will study the k-walks (discussed in Chapter 2) for the weighted graph, specifically, for $k = 2$. For better understanding of the weighted network concept, we represent a weighted network as an unweighted multigraph network by replacing each edge $e = (v_i, v_j)$ with w_ij copies of multiedges as shown in Fig. 4.1(a). For instance, for a 2-walk $v_1v_2v_3$ in a weighted network, the number of 2-walks in its corresponding unweighted network is $w_{v_1,v_2}w_{v_2,v_3}$.



(a) (b)

Figure 4.1: (a) Representation of weighted edge as a copies of multiedges. (b) 2-walks in weighted network.

The number of 2-walks that any vertex v participates in a weighted network $G = (V, E, W)$ are described below.

4.1.7 Lemma. Let $G = (V, E, W)$ be a weighted network and v be an arbitrary vertex of G. Then, there are three types of 2-walks containing v with the following observation:

1. Closed 2-walks containing the vertex v (such as walk (vyv) in Fig. 4.1(b)): The number of such 2-walks is given by

$$NW_2^C(v) = \sum_{y_i \in N(v)} w_{v,y_i}^2. \tag{4.1.10}$$

2. Non-closed 2-walks containing the vertex v as one of the end-points (such as walk (vuw) in Fig. 4.1(b)): The number of such 2-walks, denoted as $NW_2^E(v)$ is given by

$$NW_2^E(v) = \sum_{y_i \in N(v)} \left(\sum_{z_j \in \{N(y_i) - v\}} w_{v,y_i} w_{y_i,z_j} \right). \tag{4.1.11}$$

3. Non-closed 2-walks containing the vertex v as the middle point (such as walk xvz in Fig. 4.1(b)): The number of such 2-walks, denoted as $NW_2^M(v)$ is given by
$$NW_2^M(v) = \sum_{y_i,y_j \in N(v), y_i \neq y_j} w_{y_i,v} w_{v,y_j}. \tag{4.1.12}$$

4.1.8 Theorem. *Let $G = (V, E, W)$ be a weighted network of n vertices $v_1, v_2, \ldots, v_n$. Let H be the network obtained by deleting vertex v from G, then the drop of Laplacian energy with respect to v_i is*
$$(\Delta E)_i = E_L(G) - E_L(H) = 4 \cdot NW_2^C(v_i) + 2 \cdot NW_2^E(v_i) + 2 \cdot NW_2^M(v_i). \tag{4.1.13}$$

Proof. Without loss of generality, assume $H = G - v_1$. Let $N(v_1)$ be the neighborhood of vertex v_1 in G and $x'(v_i)$ be the corresponding sum-weight of the vertex v_i in H. We have:
$$x'(v_i) = \begin{cases} 0 & \text{if } i = 1, \\ x(v_i) - w_{v_1,v_i} & \text{if } v_i \in N(v_1), \\ x(v_i) & \text{otherwise.} \end{cases} \tag{4.1.14}$$

By Theorem 4.1.4 and Equation (4.1.14), we have
$$E_L(H) = \sum_{v_i \in N(v_1)} (x(v_i) - w_{v_1,v_i})^2 + \sum_{v_i \notin N(v_1)} x^2(v_i) + 2 \sum_{1<i<j} w_{i,j}^2. \tag{4.1.15}$$

But, we know that
$$E_L(G) = \sum_{i=1}^{n} x^2(v_i) + 2 \sum_{i<j} w_{i,j}^2. \tag{4.1.16}$$

We compute the drop of Laplacian energy with respect to vertex v_i by subtracting Equation (4.1.15) from Equation (4.1.16) which gives
$$\begin{aligned}
(\Delta E)_i &= E_L(G) - E_L(H) \\
&= x^2(v_1) + \sum_{v_i \in N(v_1)} (2w_{v_1,v_i} \cdot x(v_i) - w_{v_1,v_i}^2) + 2 \sum_{j=2}^{n} w_{1,j}^2 \\
&= x^2(v_1) + \sum_{v_i \in N(v_1)} 2w_{v_1,v_i} \cdot x(v_i) - \sum_{v_i \in N(v_1)} w_{v_1,v_i}^2 + 2 \sum_{j=2}^{n} w_{1,j}^2 \\
&= \left(\sum_{v_i \in N(v_1)} w_{v_1,v_i} \right)^2 - \sum_{v_i \in N(v_1)} w_{v_1,v_i}^2 + \sum_{v_i \in N(v_1)} 2w_{v_1,v_i} \cdot x(v_i) + 2 \sum_{j=2}^{n} w_{1,j}^2 \\
&= 2 \sum_{v_i,v_j \in N(v_1), v_i \neq v_j} w_{v_1,v_i} \cdot w_{v_j,v_1} + 2 \sum_{v_i \in N(v_1)} w_{v_1,v_i} \cdot \sum_{u \in N(v_i)} w_{v_i,u} + 2 \sum_{j=2}^{n} w_{1,j}^2 \\
&= 2 \sum_{v_i,v_j \in N(v_1), v_i \neq v_j} w_{v_1,v_i} \cdot w_{v_j,v_1} + 2 \sum_{v_i \in N(v_1)} \sum_{u \in N(v_i)} w_{v_1,v_i} \cdot w_{v_i,u} + 2 \sum_{y \in N(v_1)} w_{v_1,y}^2 \\
&= 2 \sum_{v_i,v_j \in N(v_1), v_i \neq v_j} w_{v_1,v_i} \cdot w_{v_j,v_1} + 2 \sum_{v_i \in N(v_1)} \left(\sum_{u \in \{N(v_i) - v_1\}} w_{v_1,v_i} \cdot w_{v_i,u} \right) + 4 \sum_{y \in N(v_1)} w_{v_1,y}^2 \\
&= 2 \cdot NW_2^M(v_1) + 2 \cdot NW_2^E(v_1) + 4 \cdot NW_2^C(v_1)
\end{aligned}$$

□

4.2 Simulations and Discussions

Here we discuss two examples that show the difference between the Laplacian centrality and the known centrality measures (i.e degree, betweenness, and closeness) for weighted networks.

4.2.1 Example. We consider a simple weighted network with 6 nodes and 6 weighted edges (see Fig. 4.2). We use the Igraph package in Python to compute the degree, betweenness, and closeness centralities. We develop code in Python to calculate the Laplacian centralities (See Appendix).

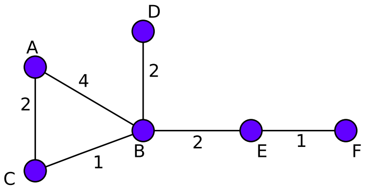


Figure 4.2: A weighted network with 6 nodes and 6 weighted edges.

Table 4.1: Centralities for nodes of the network in Fig. 4.2.

Node	Degree	Betweenness	Closeness	Laplacian
A	6	0	0.2381	0.70
B	**9**	**8**	**0.4545**	**0.90**
C	**3**	**4**	**0.3846**	**0.28**
D	2	0	0.2632	0.22
E	**3**	**4**	**0.3333**	**0.26**
F	1	0	0.2632	0.04

From Fig. 4.2, we observe that the position of node B in the network is in such a way that it has the highest number of neighbors, it is easily accessible from all other nodes, and participates in many shortest paths and 2-walks in the network. For these reasons, node B has the highest scores based on various centrality methods as listed in Table. 4.1. In addition, we observe that nodes C and E have the same scores for the degree and betweenness centralities. We can see that node C is more important than node E based on closeness centrality since node C is easily reachable to other nodes than node E. Despite the fact that nodes C and E have the same score in degree centrality, node C has a higher score than node E based on Laplacian method because the number of 2-walks that node C participates in are more than those that node E participates in. We can therefore see that degree centrality captures the local environment around a node, the closeness and betweenness centralities capture the global environment, and the Laplacian centrality captures the intermediate environment around a node.

4.2.2 Zachary's karate club network. Zachary's karate club network is a weighted social network. Zachary created the dataset after observing 34 members of a karate club over two years (Nexus, 2012). As a result of misunderstandings between the club's administrator and the club's instructor, the club split into two groups forming two karate clubs. Zachary observed the interaction of members of both clubs outside normal activities of the club (karate classes and club meetings), thus, creating a dataset where the nodes represent club members, the edges represent the friendship between members outside normal club activities and the edge weights represent the frequency of interactions between each pair of friends as shown in Fig. 4.3. The administrator and instructor are represented by the nodes n_0 and n_{33} respectively.

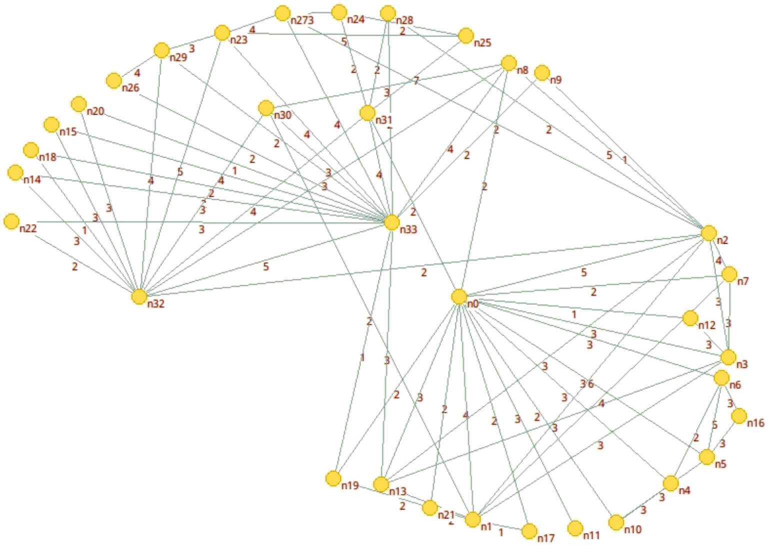


Figure 4.3: Zachary's karate club network.

Table 4.2: The scores and ranks based on four centrality methods for the 34 members of the Zachary's karate club network.

	Scores				Ranks			
Node	Degree	Betweenness	Closeness	Laplacian	Degree	Betweenness	Closeness	Laplacian
$\mathbf{n_0}$	**42**	**250.15**	**0.2538**	**0.2544**	**2**	**1**	**1**	**2**
n_1	29	33.80	0.2000	0.1725	5	8	8	5
n_2	33	36.65	0.1964	0.2166	4	6	11	4
n_3	18	1.33	0.1765	0.0965	8	18	17	10
n_4	8	0.50	0.1528	0.0350	18	21	28	22
n_5	14	15.50	0.1521	0.0571	11	10	29	16
n_6	13	15.50	0.1535	0.0541	13	10	27	18
n_7	13	0.00	0.1823	0.0789	13	24	16	11
n_8	17	13.10	0.1988	0.1222	9	12	10	8
n_9	3	7.28	0.1908	0.0218	31	14	13	30
n_{10}	8	0.50	0.1755	0.0309	18	21	19	24
n_{11}	3	0.00	0.1460	0.0216	31	24	30	31
n_{12}	4	0.00	0.2050	0.0174	28	24	5	33
n_{13}	17	1.20	0.1908	0.1189	9	19	13	9
n_{14}	5	0.00	0.1710	0.0366	25	24	22	20
n_{15}	7	0.00	0.1375	0.0549	20	24	32	17
n_{16}	6	0.00	0.1086	0.0173	22	24	34	34
n_{17}	3	16.10	0.1930	0.0192	31	9	12	32
n_{18}	3	3.00	0.1875	0.0226	31	16	15	29
$\mathbf{n_{19}}$	**5**	**127.07**	**0.2481**	**0.0331**	**25**	**3**	**3**	**23**
n_{20}	4	0.00	0.2037	0.0280	28	24	6	26
n_{21}	4	0.00	0.1765	0.0246	28	24	17	27
n_{22}	5	0.00	0.1587	0.0382	25	24	24	19
n_{23}	21	1.00	0.1387	0.1294	6	20	31	7
n_{24}	7	33.83	0.1579	0.0227	20	7	25	28
n_{25}	14	0.50	0.1236	0.0645	11	21	33	15
n_{26}	6	0.00	0.1692	0.0282	22	24	23	25
n_{27}	13	6.50	0.1564	0.0752	13	15	26	12
n_{28}	6	10.10	0.2025	0.0365	22	13	7	21
n_{29}	13	0.00	0.1746	0.0707	13	24	20	14
n_{30}	11	3.00	0.1737	0.0709	17	16	21	13
n_{31}	21	66.33	0.2089	0.1310	6	4	4	6
n_{32}	38	38.13	0.2000	0.2371	3	5	8	3
$\mathbf{n_{33}}$	**48**	**209.50**	**0.2519**	**0.3067**	**1**	**2**	**2**	**1**

From the results in Table. 4.2, we observe that the administrator (n_0) and the instructor (n_{33}) have the highest scores based on the four centrality measures as we expected since the two members are connected to many other members, they are close to many members, act as bridges between the two groups and participate in a large number of two walks in the network. Another observation is that node n_{19} has the third highest score based on betweenness and closeness centrality measures. This is because from Fig. 4.3, we observe that node n_{19} connects to the two most important nodes n_0 and n_{33} and thus acts as a bridge to members of both groups and also node n_{19} is in a position of close proximity to most of the nodes belonging to two groups.

In order to understand the relationship of the Laplacian centrality to the other standard centrality measures, we compute the Pearson's correlation coefficients which we obtain as: 0.9880, 0.6770, and 0.5136 for degree, closeness, and betweenness centralities respectively. We observe a good correlation coefficient between the degree and Laplacian centrality measures which implies that nodes with high degree centrality tend to have high Laplacian centrality while nodes with low degree centrality also tend to have low Laplacian centrality except for a few cases.

4.3 Physical Analogy of Laplacian Matrix

4.3.1 Diffusion. Diffusion is, among others, the movement of substance from a region of high concentration to a region of low concentration. Such substances include heat, gas, etc. We can consider the diffusion process over a network to build simple models of spread of a disease, information, etc. across a network (Newman, 2010).

Consider a simple undirected network. Suppose we have a quantity ϕ_i of a substance on a vertex i at time t. Suppose the substance moves along edges from node j to an adjacent node i at a rate $C(\phi_j - \phi_i)$, where C is the diffusion constant ($C > 0$). The amount of substance moving from j to i in a small time interval dt is $C(\phi_j - \phi_i)\ dt$. Then the rate at which substance ϕ_i changes is given by

$$\frac{d\phi_i}{dt} = C\sum_j A_{ij}(\phi_j - \phi_i) = C\sum_j A_{ij}\phi_j - C\phi_i\sum_j A_{ij} = C\sum_j A_{ij}\phi_j - C\phi_i k_i.$$

Thus,

$$\frac{d\phi_i}{dt} = C\sum_j (A_{ij} - \delta_{ij}k_i)\phi_j, \tag{4.3.1}$$

where $\mathbf{A}$ is the adjacency matrix, k_i is the degree of node i, and δ_{ij} is the Kronecker delta whose value is 1 if $i = j$ and 0 otherwise. In matrix-vector notation, we have

$$\frac{d\boldsymbol{\phi}}{dt} + C\mathbf{L}\boldsymbol{\phi} = 0, \quad \boldsymbol{\phi}(0) = \boldsymbol{\phi}_0. \tag{4.3.2}$$

We observe that Equation (4.3.2) is similar to that of the ordinary heat equation $\left(\frac{du}{dt} - \alpha\nabla^2 u = 0\right)$ where the Laplace operator ∇^2 is replaced by the Laplacian matrix $\mathbf{L}$. Thus, the Laplacian matrix is called the Graph Laplacian. The matrix solution to Equation (4.3.2) is given as

$$\boldsymbol{\phi}(t) = \boldsymbol{\phi}_0\ e^{-C\mathbf{L}t}. \tag{4.3.3}$$

The solution vector $\boldsymbol{\phi}(t)$ can also be found as a linear combination of eigenvector $\mathbf{v}_i$ of the Laplacian matrix $\mathbf{L}$ (Edwards and Penney, 2004). That is,

$$\boldsymbol{\phi}(t) = \sum_i a_i(t)\mathbf{v}_i. \tag{4.3.4}$$

Introducing Equation (4.3.4) in the left hand side of Equation (4.3.2) and after algebraic manipulation we get

$$\frac{da_i}{dt} + C\lambda_i a_i = 0, i = 1, \cdots, n, \tag{4.3.5}$$

whose solution is

$$a_i(t) = a_i(0)e^{-C\lambda_i t}. \tag{4.3.6}$$

Thus,

$$\boldsymbol{\phi}(t) = \sum_i a_i(0)e^{-C\lambda_i t}\mathbf{v}_i,$$

where $a_i(0), \lambda_i$, and $\mathbf{v}_i$ are the initial condition, eigenvalues and eigenvectors of the Laplacian matrix respectively. As the Laplacian matrix is symmetric, its set of eigenvectors is orthonormal and $a_i(0)$ in terms of $\boldsymbol{\phi}(0)$ can be found simply by projection of $\boldsymbol{\phi}(0)$ onto the set of eigenvectors, i.e.

$$a_i(0) = \langle \boldsymbol{\phi}(0), \mathbf{v}_i \rangle \;\; i = 1, \cdots, n. \tag{4.3.7}$$

As discussed earlier, the eigenvalues λ_i of $\mathbf{L}$ are non-negative which implies that the solution to the diffusion equation either decays exponentially or remains constant which implies that equilibrium can be attained. Thus, given λ_i and the initial condition $a_i(0)$, we be find the solution at any time t.

4.3.2 Equilibrium Behaviour. As t goes to infinity, we have

$$\lim_{t\to\infty} e^{-C\lambda_i t} = \begin{cases} 0 & \text{if } \lambda_i > 0 \\ 1 & \text{if } \lambda_i = 0, \end{cases} \tag{4.3.8}$$

Asymptotically, the equilibrium state is completely determined by the kernel of $\mathbf{L}$. Since $\sum_j \mathbf{L}_{ij} = 0$, it is easy to see that $\mathbf{v^1} = \frac{1}{\sqrt{n}}[1, \cdots, 1]$, the eigenvector associated with $\lambda_i = 0$, is in the kernel of $\mathbf{L}$. We then have

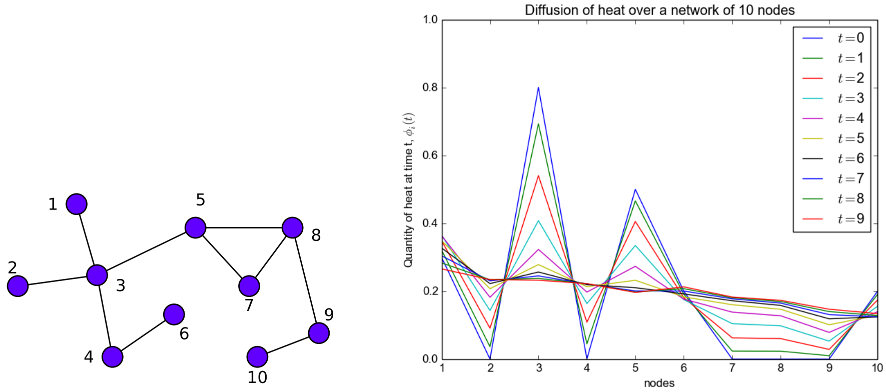
$$\lim_{t\to\infty} \boldsymbol{\phi}(t) = \langle \boldsymbol{\phi}(0), \mathbf{v^1} \rangle \mathbf{v^1}. \tag{4.3.9}$$

The quantity of heat $\phi_j(t)$ at any node j at time t is given by

$$\lim_{t\to\infty} \phi_j(t) = \frac{1}{n}\sum_{i=1}^{n} \phi_i(0). \tag{4.3.10}$$

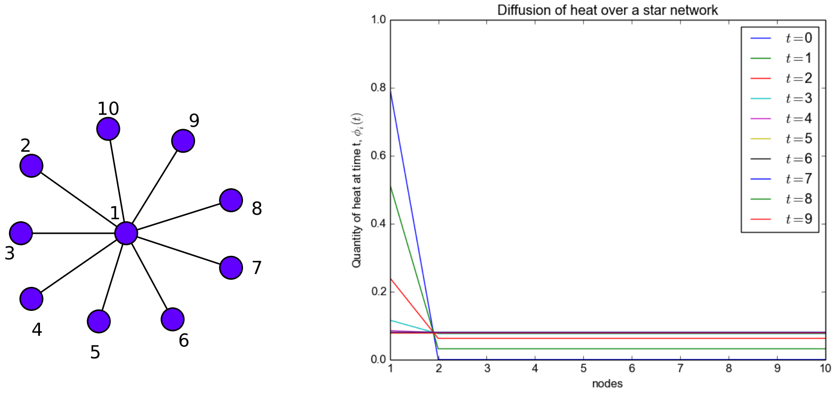
At steady state, the value of $\boldsymbol{\phi}$ converges to the same value at each of the nodes in the network, which is the average of the initial values at all of the nodes. This is because, as expected, neighboring nodes in the network will exchange heat until that heat is spread out evenly throughout all of the nodes that are connected to each other.

4.3.3 Example. Let us consider diffusion of heat over the network in Fig. 4.4(a). Suppose the quantity of heat at each node at time $t = 0$ is given by the vector $\boldsymbol{\phi}(0) = [0.3, 0.0, 0.8, 0.0, 0.5, 0.2, 0.0, 0.0, 0.0, 0.2]$, values between 0 and 1. Let $C = 0.05$. Fig. 4.4(b) illustrates how heat spreads over the network in Fig. 4.4(a). We observe that at each time step t, nodes that initially have high amounts of heat (i.e $1, 3, 5, 6$, and 10) exchange heat with adjacent nodes that were initially cool. The latter gain heat from the hot neighbors and eventually all nodes in the network have relatively equal amounts of heat. At this point, all nodes are approaching the equilibrium point. At $t = 9$, the quantity of heat at each node is $\boldsymbol{\phi}(9) = [0.20038, 0.20038, 0.20031, 0.20054, 0.19987, 0.20067, 0.19972, 0.19962, 0.19934, 0.19918]$. This explains the fact that as time t increases, the quantity of heat $\boldsymbol{\phi}_j(t)$ at each node tends to the equilibrium value of 0.2.



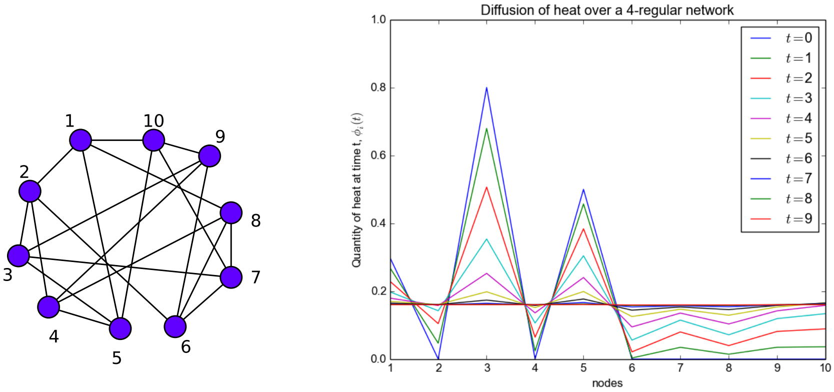
(a) (b)

Figure 4.4: (b) is an illustration of the diffusion process over the network in (a).



(a) (b)

Figure 4.5: (b) is an illustration of the diffusion process over a star network in (a).



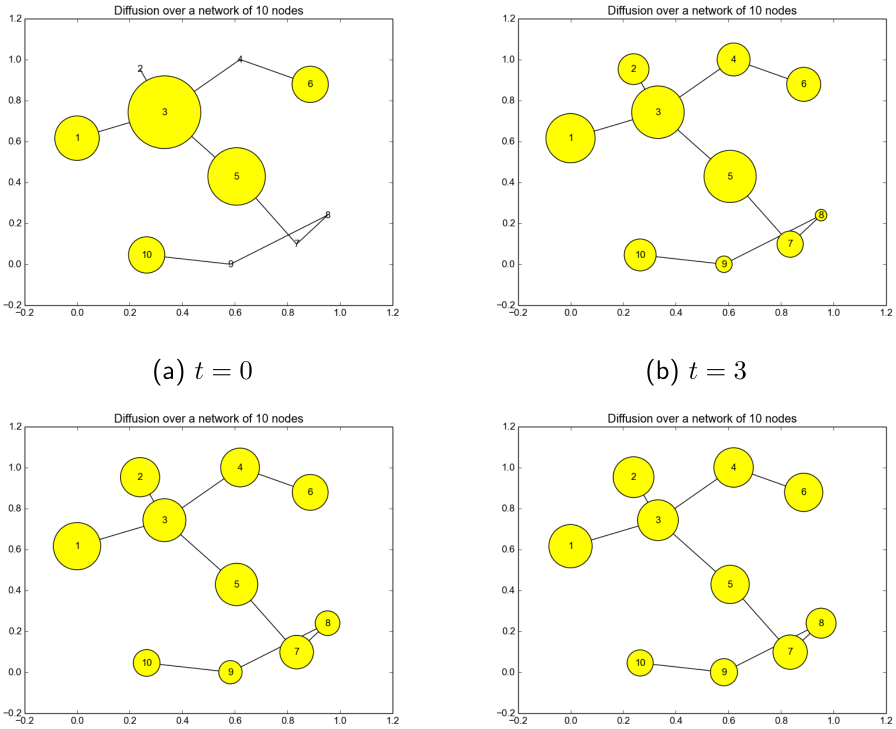
(a) (b)

Figure 4.6: (b) is an illustration of the diffusion process over a 4-regular network in (a).

Consider a star network in Fig. 4.5(a). Suppose the initial quantity of heat at the nodes is $\phi(0) = [0.8, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$ and let $C = 0.5$. From Fig. 4.5(b), we observe that the quantity of heat at node 1 drops so fast. This is because node 1, which initially has the highest amount of heat, is directly connected to many other nodes whose heat contents are very low. Thus, node 1 plays a central role in the exchange of heat within its neighborhood. The neighboring nodes to node 1 which had no heat initially, steadily gain equal amounts of heat from node 1 and as such their heat content raises as time goes by until the equilibrium point is reached. At $t = 9$, the quantity of heat at all nodes is the same, that is, $\phi_j(9) = 0.08$ which implies equilibrium.

For the 4-regular network in Fig. 4.6(a), we set $\phi(0) = [0.3, 0.0, 0.8, 0, 0.5, 0, 0, 0, 0, 0, 0]$ and $C = 0.5$. From Fig. 4.6(b), we observe that the nodes with high mounts of heat slowly loose their heat to their neighbors that have low quantities of heat and with time all nodes tend to have equal amounts of heat. At $t = 9$, all nodes in the network have relatively the same amount of heat which is 0.16. This implies that nodes are almost at the equilibrium point.

A better visualisation of the diffusion process over the network in Fig. 4.4(a) at different time intervals is illustrated in Fig. 4.7 where the quantity of heat at any node at time t is proportional to the size of the node.



(a) $t = 0$

(b) $t = 3$

(c) $t = 6$

(d) $t = 9$

Figure 4.7: Illustration of the diffusion process at different time intervals over the network in Fig. 4.4(a).

In conclusion, we can tell that the structure of a network, the centralities of nodes, and the initial distribution of heat over a network play a key role in the diffusion process and hence have an impact on when diffusion is reached. For instance, equilibrium is reached faster in the star network as compared to the other two networks because the most central node of the star network (node 1) exchanges heat with its neighbors at a fast rate until equilibrium is reached.

5. Conclusion

The Laplacian matrix of a network is a matrix representation of a network whose properties provide useful information about the structure of a network. The spectrum of the Laplacian matrix is vital in describing the connectivity of a network. For instance, the number of connected components in a network, the number of spanning trees of a network, etc. The Laplacian matrix has various applications in the study of networks. In this essay, we discussed two of the applications namely Laplacian centrality and the diffusion process. The Laplacian centrality is a centrality measure based on the Laplacian matrix of a weighted undirected network. The Laplacian method of estimating the centrality (or importance) of a node involves not only the direct connections to its immediate neighborhood but also the importance of its neighbors. The intermediate characterisation of the node centrality based on the Laplacian method plays a vital role in the study of social networks such as Zachary's karate club network. One limitation of the Laplacian centrality is that it is applied to undirected weighted networks only. However, some real-world networks that we need to study are directed networks, for instance, food webs. The Laplacian matrix of a directed network is not symmetric which means that we need to define the Laplacian centrality for directed networks. To start with, we have to redefine the 'concept of 2-walks' in directed networks.

Further more, we explored another application of the Laplacian matrix called the diffusion process over a network. The proces by which a substance spreads over a network is similar to the diffusion process of heat in physics and thus the Laplacian matrix is referred to as the graph Laplacian. We observed that the structure of a network and the centralities of the nodes in a network play a key role in determining how the diffusion process occurs on a network. We can use the concept of diffusion to create simple models for spread of epidemics, information, rumour, belief, etc. over a network. We consider extending the study of diffusion to real-world networks and to introduce the concept of long-range interaction in the diffusion process over networks. The concept of long-range diffusion has been used successfully in the analysis of consensus in networks (Estrada, 2012) and in modelling the spread of epidemic in networks (Estrada et al., 2011).

6. Appendix

6.1 Python code that computes Laplacian Centrality

```
import networkx as nx              #import the package for graphs
import numpy as np
import pylab as plt


# function that computes the laplacian centrality for all nodes in G
def laplacian_centrality(G):

 # function that computes the Laplacian energy of the whole graph G.
 # First,compute weighted degrees for the nodes,
 #followed by the sum of the square of the weight-sums and
 # then laplacian energy of the graph.

    def laplacian_energy(G):
        d = G.degree(weight='weight')
        x_i= d.values()
        sum_x = np.sum( np.square(x_i))

        s_w = 0
        edges = G.edges()
        for edge in edges:
            s_w = s_w + np.square(G[edge[0]][edge[1]]['weight'])

        E = sum_x + 2*s_w
        return(E)
# function that the computes the chhange in laplacian energy.
# First, it computes the number of open 2-walks in which node is a middle point to zero,
# number of closed 2-walks to zero and
# number of open 2-walks in which node is a end point to zero

    def change_in_energy(node):
        NWM=0
        NWC=0
        NWE=0
        neigb = G.neighbors(node)

        for i in range(0,len(neigb)-1):
            for j in range(i+1,len(neigb)):
                NWM = NWM + (G[node][neigb[i]]['weight'] * G[node][neigb[j]]['weight'])

        for n in neigb:
            NWC = NWC + np.square(G[node][n]['weight'])
            for m in list(set(G.neighbors(n))^ set(node)):
                NWE = NWE + (G[node][n][’weight’]*G[n][m][’weight’])

        return(4*NWC + 2*(NWE + NWM))

# computes the Laplacian centralities for all nodes
    lap_dict ={}
    for nd in G.nodes():
        lap_centrality = change_in_energy(nd)/laplacian_energy(G)
        lap_dict[nd]=lap_centrality
    return(lap_dict)

################example 4.2.1 ############################################
G = nx.Graph()
edge_list = [(’A’,’B’,4.0),(’A’,’C’,2.0),(’B’,’C’,1.0),(’B’,’D’,2.0),
(’B’,’E’,2.0),(’E’,’F’,1.0)]
G.add_weighted_edges_from(edge_list)
laplacian_centrality(G)
```

Acknowledgements

First and foremost, I am grateful to the Almighty God for the gift of life and for seeing me through up to this far. I would like to express my sincere gratitude to my research supervisor, **Dr Franck Kalala Mutombo** for his generous support, guidance, and willingness to impart knowledge to me. My special thanks go to my tutor, **Martha Kamkuemah** for the academic support. I am very grateful to **Dr Simukai Utete** for her valuable advice, assistance and motivation. I would also like to extend my appreciation to the entire AIMS family for providing a healthy and conducive environment, technical advice, and moral support. Finally, I wish to thank my parents, relatives, and friends for their support and encouragement throughout my study.

References

A. Barrat, M. Barthelemy, R. Pastor-Satorras, and A. Vespignani. The architecture of complex weighted networks. *Proceedings of the National Academy of Sciences of the United States of America*, 101 (11):3747–3752, 2004.

K. C. Das. The laplacian spectrum of a graph. *Computers & Mathematics with Applications*, 48(5): 715–724, 2004.

C. H. Edwards and D. E. Penney. *Differential equations and boundary value problems*, volume 2. Prentice Hall, 2004.

E. Estrada. *The structure of complex networks: theory and applications*. OUP Oxford, 2011.

E. Estrada. Path laplacian matrices: introduction and application to the analysis of consensus in networks. *Linear Algebra and its Applications*, 436(9):3373–3391, 2012.

E. Estrada. Introduction to complex networks: structure and dynamics. In *Evolutionary Equations with Applications in Natural Sciences*, pages 93–131. Springer, 2015.

E. Estrada and J. A. Rodriguez-Velazquez. Subgraph centrality in complex networks. *Physical Review E*, 71(5):056103, 2005.

E. Estrada, F. Kalala-Mutombo, and A. Valverde-Colmeiro. Epidemic spreading in networks with non-random long-range interactions. *Physical Review E*, 84(3):036110, 2011.

E. Estrada, P. Knight, et al. *A first course in network theory*. Oxford University Press, USA, 2015.

L. Euler. Leonhard euler and the königsberg bridges. *Scientific American*, 189(1):66–70, 1953.

J. Gower. A modified leverrier-faddeev algorithm for matrices with multiple eigenvalues. *Linear Algebra and its Applications*, 31:61–70, 1980.

I. Gribkovskaia, Ø. Halskau, and G. Laporte. The bridges of königsberg—a historical perspective. *Networks*, 49(3):199–203, 2007.

I. Gutman and O. E. Polansky. *Mathematical concepts in organic chemistry*. Springer Science & Business Media, 2012.

J. M. Harris, J. L. Hirst, and M. J. Mossinghoff. *Combinatorics and graph theory*, volume 2. Springer, 2008.

Internet. Network. URL https://www.google.com/search?q=internet+network&client=ubuntu&hs=xZp&channel=fs&gl=za&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiZsrLOueTMAhVMCcAKHS_8AysQ_AUIBygB&biw=1215&bih=900#channel=fs&gl=za&tbm=isch&q=computer+network&imgrc=_. [Online; accessed 2016-03-29].

S. Milgram. The small world problem. *Psychology today*, 2(1):60–67, 1967.

J. J. Molitierno. *Applications of combinatorial matrix theory to Laplacian matrices of graphs*. CRC Press, 2012.

F. K. Mutombo. *Long-range interactions in complex networks*. PhD thesis, University of Strathclyde, 2012.

M. Newman. *Networks: an introduction*. OUP Oxford, 2010.

M. E. Newman. Scientific collaboration networks. ii. shortest paths, weighted networks, and centrality. *Physical review E*, 64(1):016132, 2001.

Nexus. Zachary's karate club [karate], 2012. URL http://nexus.igraph.org/api/dataset_info?id=1&format=html. [Online; accessed 2016-04-25].

T. Opsahl. *Structure and evolution of weighted networks*. PhD thesis, Queen Mary, University of London, 2009.

T. Opsahl and P. Panzarasa. Clustering in weighted networks. *Social networks*, 31(2):155–163, 2009.

T. Opsahl, F. Agneessens, and J. Skvoretz. Node centrality in weighted networks: Generalizing degree and shortest paths. *Social Networks*, 32(3):245–251, 2010.

X. Qi, E. Fuller, Q. Wu, Y. Wu, and C.-Q. Zhang. Laplacian centrality: A new centrality measure for weighted networks. *Information Sciences*, 194:240–253, 2012.

X. Qi, R. D. Duval, K. Christensen, E. Fuller, A. Spahiu, Q. Wu, Y. Wu, W. Tang, C. Zhang, et al. Terrorist networks, network energy and node removal: a new measure of centrality based on laplacian energy. *Social Networking*, 2(01):19, 2013.

X. F. Wang and G. Chen. Complex networks: small-world, scale-free and beyond. *Circuits and Systems Magazine, IEEE*, 3(1):6–20, 2003.

D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world'networks. *nature*, 393(6684): 440–442, 1998.

R. J. Wilson. *An introduction to graph theory*. Pearson Education India, 1970.