Technical Report

Alice NANYANZI (alicenanyanzi@aims.ac.za)

May 15, 2018

Outline of the Thesis

1. Abstract
2. Introduction
    - Brief intro of networks
    - Say what you are going to do
    - why and how are you going to do the above
    - Context - compare to others
    - Structure of the thesis
3. Chapter 1: Review of networks
    - Similarity in networks
    - Voronnoi Tensselation
    - Delaunay graphs
    - Harris corner detection
    - Image manipulation such as cropping, rotation, etc
    - Data analysis include PCA, etc
4. Chapter 2: Diffusion on networks
5. Chapter 3: Laplacian centrality of an edge
    - Laplacian centrality of a node
    - Motivation for edge centrality
6. Chapter 4: New horizon
    - Heat kernel Centrality
    - Communicability Centrality with k-hop (Do some toy models)
7. Conclusion

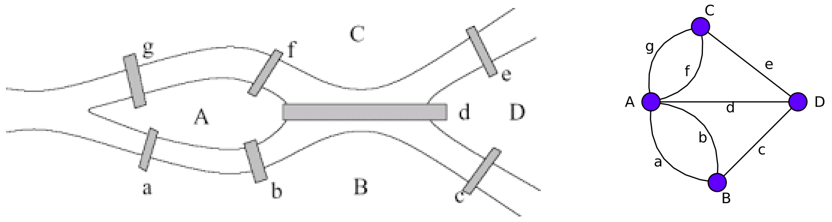
1 Review of Networks

1.1 Graphs and Networks. An Introduction

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.

1.2 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. 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. 1(b). It was then that the story of graph theory begun (Estrada, 2011).



(a) (b)

Figure 1: The Königsberg bridges: (a) A schematic diagram of the seven Königsberg bridges. (b) A graph of the Königsberg bridges. Source: (Internet).

Definition 1.3 (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. 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.
Definition 1.4 (Multi-graph)**.** A multi-graph is a graph with multiple edges (Newman, 2010).
Definition 1.5 (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.
Definition 1.6 (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. **??**(**??**) is a weighted graph of 4 nodes.

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.

Definition 2.1 (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 2.1 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'.

Definition 2.2 (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).

Definition 2.3 (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).

Definition 2.4 (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 $f : e_1 \rightarrow [v_i, v_j]$ and $f : e_2 \rightarrow [v_j, v_i]$. 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).

2.5 Examples of real-world networks

In his work (Newman, 2003), Newman considered a loose categorisation of networks: social networks, communication networks, technological networks, and biological networks.

a. Social Networks

   Networks considered as social networks are ones whose nodes correspond to people or groups of people while the edges represent the interactions or relationship between them (Jackson, 2010). For instance friendship networks such as facebook, twitter in which the interactions represent friendship ties among acquaintances, networks of intermarriages between families, social interaction networks which capture peoples' interactions through social activities or events, employee networks with companies, and many others. Some common networks that researchers have frequently experimented upon include: the Zachary karate network which consists of two communities centred at the administrator and instructor as a result of misunderstanding that prevailed with the karate club earlier on. The nodes in the network are the members of the club as the links represent interactions between members during non-club activities (Zachary, 1977). Other networks include the Dolphine network (Williams et al., 1993), terrorist networks (Magouirk et al., 2008) among others .

b. Information networks:

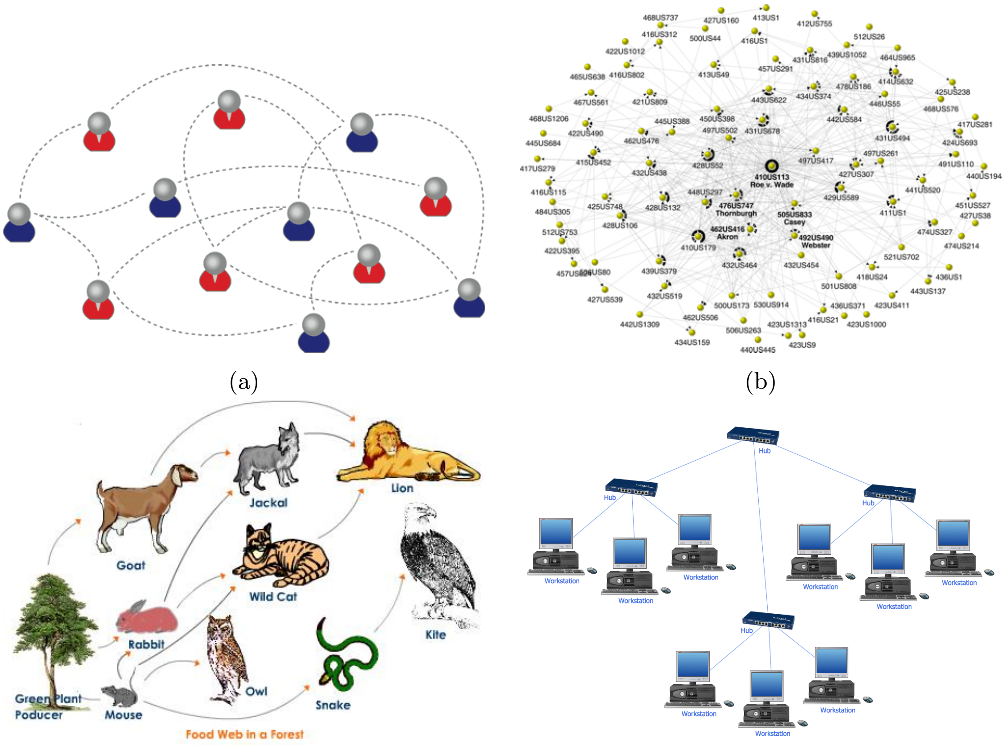
   Information networks are also referred to as knowledge networks. Examples of networks under this category include: The world wide web which consists of billions of web pages as nodes that are linked together through links known as hyperlinks (Huberman, 2001). Another network categorised as information networks are citation networks that are composed of nodes which are articles while directed link between two nodes written as $i \longrightarrow j$ indicate article i cites article j.

c. Technological networks:

   This category consist of networks made by man to aid in distribution or transfer of resources, services or commodities such as electricity, water, transportation services, and many others. Examples of such networks include the internet, transportation networks, power grids, to mention but a few (Faloutsos et al., 1999; Pagani and Aiello, 2013; Banavar et al., 1999).

d. Biological networks:

Biological networks exists in areas related human and processes that take place with in the human body, animals and their ways of survival, chemistry. Such networks are the human brain network, protein-protein interaction network, network of metabolic path ways, ecological networks (Estrada, 2011; Sporns et al., 2004; Schwikowski et al., 2000).



(a)

(b)

(c)

(d)

Figure 2: Networks in real world: (a) A social network. (b) A citation network. (c) A food web. (d) Computer network. Source: (Internet)

2.6 Complex Systems and Complex Networks

Complex systems are very vital in our daily lives. They exist in fields such as social, economic, science, technology among others. During his interview with San Jose Mercury News in January 2000, Stephen Hawking referred to the 21st century as a century of complexity. Complex systems are composed of interconnected components,however, it has been observed that many complex systems display a behaviour phenomena (also known as emergent behaviour) that cannot be explained by any conventional analysis of the system's constituent parts (Casti, September 26, 2017). In otherwords, for one to understand the behaviour of a system, it is necessary for one to consider a holistic system-level view point. There are different approaches to study of complex systems for instance statistical description, empirical data analysis, simulations, analytical approach and network approach. In this work, we focus on the network approach in which we represent complex systems by complex networks whose nodes (vertices) and links(edges) represent the components and the interactions among components respectively. For example, a transportation system can be represented by network where nodes are cities or towns and the links are the roads, railways or flight routes. This is then followed by mathematical formulation of the problem, modelling and validation. An interesting early historical application of the network approach to the study of complex systems is the Käonigsberg bridge problem where Euler (Euler, 1976, 1953) solved the problem by reformulating problem in terms of a graph where vertices represent islands while edges represent the seven bridges joining any two islands. Work published by Leonhard Euler (Euler, 1976) is considered the genesis of the story of network theory. As the size of network increases from just graphs of tens or hundreds of nodes which could easily be analysed by direct use of eye so as to ascertain the structure of the network to complex networks consisting of million or billion of nodes which call for advanced analytic approach that involves development of statistical methods to quantify such large networks. The statistical methods aid in answering questions such as how many nodes or edges should be removed for the network to break down?, what is the shortest path length of the network?, and many others.

Some of the characteristics of complex systems include: emergent behaviour, self organisation, decentralised organisation of components, evolving nature of complex systems, among others.

2.7 Terminology in network theory

Definition 2.8 (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).

Definition 2.9 (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).

Definition 2.10 (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, $N_G(v) = \{u \in V | uv \in E\}$

Definition 2.11 (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 $k_v = k_v^{in} + k_v^{out}$.

Definition 2.12 (Walk)**.** A walk in a network is a series of edges (not necessarily distinct)

$$(u_1, v_1), (u_2, v_2), \ldots, (u_k, v_k), \quad \text{for which } v_i = u_{i+1} \ (i = 1, 2, \ldots, l - 1).$$

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 $\mathbf{A}^k$ where $\mathbf{A}$ is the Adjacency matrix of a graph which we discuss in subsequent subsections.

Definition 2.13 (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). For any pair of nodes v_i,v_j in a connected graph, there exists at least one path connecting v_i to v_j. The paths with minimum length are referred as shortest-paths.

Definition 2.14 (Irreducible set of shortest paths)**.** An irreducible set of shortest paths of length l is the set $P_l = P_l(v_i, v_j), P_l(v_i, v_r), \cdots, P_l(v_s, v_t)$ in which the endpoints of every shortest-path $P_l(v_i, v_j)$ in the set are different. Each path in this set is referred to as an irreducible shortest-path (Estrada, 2012).

Definition 2.15 (Connectivity of a graph)**.** A non-empty graph G is said to be connected if there exists a path between any two pair of vertices (Diestel,2000).

Definition 2.16 (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.

2.17 Special Categories of Networks

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

Definition 2.18 (Star network)**.** A star network S_n is a complete bipartite network $K_{1,n}$ (Wilson, 1970).

Definition 2.19 (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.

Definition 2.20 (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.

Definition 2.21 (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).
Definition 2.22 (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).
Definition 2.23 (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.
Definition 2.24 (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).

2.25 Matrix Representation of Graphs/Networks

Networks/graphs can be represented in a number of ways namely edge lists, matrices, and many others. However, matrices are the most widely used technique for representation of networks especially for large graphs/networks whose structure cannot be captured by human eye. In addition, representing networks by matrices enables the application of mathematical and computer tools on networks for purposes of summation, pattern identification and many others (Chandak et al., 2017; Turán, 1984). In the subsequent subsections, we discuss the most common matrices used in the field of graph theory as well as their properties.

2.26 Adjacency Matrix

The Adjacency matrix is very useful and simple matrix commonly used in graph representation. It captures the connection between nodes in the graph that is to say, which node is connected to which one in the graph. The adjacency matrix (also known as binary adjacency) is square matrix whose entries are given by

$$\mathbf{A}_{ij} = \begin{cases} 1 & \text{if } i \text{ and } j \text{ are adjacent,} \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

The summation of the ith row or column is equivalent to the total number of immediate neighbours, known as degree, of vertex v_i. For simple undirected networks, the matrix is symmetric with zeros entries at the main diagonal. However, for directed networks, the matrix may be asymmetric since direction of edges have to be considered. For multigraphs, the entries are the number of edges between each pair of vertices and for graphs with loops the diagonal entries are non-zero due to self-loops which may be counted once or twice based on whether the network is directed or undirected (Biggs, 1993; Godsil and Royle, 2001). For graphs with weighted edges, the adjacency matrix is given by

$$\mathbf{A}_{ij} = \begin{cases} w_{i,j} & \text{if } i \text{ and } j \text{ are adjacent,} \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

The spectrum, which is the eigenvalues and their multiplicities, of the Adjacency matrix is such a rich one and is thus used in mine interesting information about the graph. For example, the multiplicity of the largest eigenvalues is equal to the number of connected components of the graph (Cvetkovic and Rowlinson, 2004).

2.27 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 $\mathbf{D(G)}$ is defined as

$$D_{i,j} = \begin{cases} k_i & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

In a directed network the degree of node may be the in-degree or the out-degree.

2.28 Distance matrix

The distance matrix also known as the all-pairs shortest path matrix denoted by $\mathbf{S}$ is a symmetric matrix whose elements are defined as

$$S_{i,j} = \begin{cases} l_{i,j} & \text{if } i \neq j \\ 0 & \text{otherwise,} \end{cases} \tag{4}$$

where $l_{i,j}$ is the length of the irreducible shortest path between nodes i and j.

2.29 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. It should be, however, noted that the orientation does not matter. Then the oriented incidence matrix $\mathbf{B(G)}$ has entries defined as

$$B_{ij} = \begin{cases} +1 & \text{if node } v_i \text{ is the positive end of the edge } e_j \\ -1 & \text{if node } v_i \text{ is the negative end of the edge } e_j \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

2.30 Laplacian Matrix

The Laplacian matrix or graph Laplacian is one of matrices used to represent a graph or network. Recently, a number of researchers have been deeply involved in the study of the Laplacian matrix of a graph since this matrix has interesting spectral properties that provide more useful information about the structure of a graph as compared to other matrices such as the adjacency matrix. The Laplacian plays a key role as a natural link between discrete representations like graphs and continuous representations such as vector spaces and manifolds. There are various applications of the graph Laplacian which include spectral clustering, spectral matching, diffusion on networks, centrality measure, among others, which we explore later on.

The Laplacian matrix takes on different versions namely the normalised and unnormalised Laplacian matrices.

2.31 Definitions and Properties of the Laplacian Matrix

Definition 2.32 (Combinatorial Laplacian Matrix)**.** The Combinatorial Laplacian or unnormalised Laplacian matrix of a network is defined as the difference between the Degree matrix $\mathbf{D}$ and the Adjacency matrix $\mathbf{A}$ of a network. That is,

$$\mathbf{L} = \mathbf{D} - \mathbf{A}. \tag{6}$$

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

$$L_{ij} = \begin{cases} k_{v_i} & \text{if } i = j \\ -1 & \text{if } i \neq j \text{ and } v_i \text{ is adjacent to } v_j \\ 0 & \text{otherwise,} \end{cases} \tag{7}$$

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

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

$$\mathbf{L} = \mathbf{B}\mathbf{B}^T, \tag{8}$$

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

Some of the properties of the Combinatorial graph Laplacian include the following:

1. Real and symmetric matrix

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

2. Singular matrix

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

3. Positive semi-definite

   A matrix is positive semi-definite if and only if all its eigenvalues are non-negative. For a given matrix $\mathbf{L}$, this property is denoted by $\mathbf{L} \geq 0$. This property of the Laplacian matrix makes it more suitable for spectral analysis compared to the adjacency and incidence matrices.

2.32.1 Spectrum of the Combinatorial Laplacian Matrix

As mentioned earlier, 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

$$Sp\mathbf{L} = \begin{pmatrix} \lambda_1 & \lambda_2 & \ldots & \lambda_n \\ m(\lambda_1) & m(\lambda_2) & \ldots & m(\lambda_n) \end{pmatrix}. \tag{9}$$

We consider the non increasing order of the eigenvalues of $\mathbf{L}$: $\lambda_n \geq \lambda_{n-1} \geq \cdots \geq \lambda_2 \geq \lambda_1 = 0$. Some of the results associated with the spectrum of the Laplacian matrix include:

- The eigenvalues of $\mathbf{L}$ are bounded as $0 \leq \lambda_j \leq 2k_{max}$ and $\lambda_n \geq k_{max}$ (Estrada, 2011).
- The eigenvalue λ_1 is always equal to zero (Estrada, 2011). Atleast one eigenvalue of the Laplacian is 0.

  Proof. Consider the vector $v = (1/\sqrt{n}, \cdots, 1/\sqrt{n})$. We know that the ith entry of Lv is $\sum_{i\sim j} v_{(i)} - v_{(j)} = \sum_{i\sim j}(1/\sqrt{n} - 1/\sqrt{n}) = 0 = 0 \cdot v_{(i)}$. □

- 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 magnitude of this value depict how well connected the over all graph is. The algebraic connectivity has significant implications for properties such as clustering and synchronizability. The eigenvector corresponding to the eigenvalue λ_2 is called the Fiedler vector (Estrada et al., 2015).
- Let G be a graph with connected components $G_i (1 \leq i \leq s)$. Then the spectrum of G is the union of the spectra of G_i (and multiplicities are added) (Brouwer and Haemers, 2011).

- For a graph G, the sum of the eigenvalues, that is, the trace of $\mathbf{L}$ is twice the number of edges of G. Mathematically, $\sum_{i=1}^{n} \lambda_i = Trace(\mathbf{L}) = 2E$. (Brouwer and Haemers, 2011)

Theorem 2.33 (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*

$$V_1 = \{i \in V | x_i \geq r\}, \;\; V_2 = \{i \in V | x_i < r\}, \tag{10}$$

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.

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, $K_{m,n} : Sp(\mathbf{L}) = \{0 \; m^{n-1} \; n^{m-1}\}$ (Estrada, 2011).

Theorem 2.34 (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).*

2.35 Normalized Laplacian matrix

One format of normalized Laplacian matrix also known as symmetric normalised Laplacian is defined as

$$\mathcal{L}_{ij} = \begin{cases} 1, & \text{if } i = j \text{ and } k_i \neq 0, \\ -\frac{1}{\sqrt{k_i k_j}}, & \text{if } v_i \text{ and } v_j \text{ are adjacent,} \\ 0, & \text{otherwise.} \end{cases}$$

We can also write

$$\mathcal{L} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$$

where $\mathbf{D}^{-1/2}$ is the diagonal matrix determined by the inverse square root of each diagonal entry of the degree matrix (Estrada, 2011). For a k-regular graph, we have

$$\mathcal{L} = \mathbf{I} - \frac{1}{k}\mathbf{A}. \tag{11}$$

The matrix $\mathcal{L}$ is symmetric with real and nonnegative eigenvalues which satisfies:

$$0 = \lambda_1(\mathcal{L}) \leq \lambda_2(\mathcal{L}) \leq \cdots \leq \lambda(\mathcal{L}) \leq 2.$$

It is very convenient to work with the spectrum of the normalised Laplacian because of the small interval ($[0, 2]$) for example it is easier to compare the spectrum of the normalised than the unnormalised Laplacian of two graphs. In addition, the eigenvalues of the normalised Laplacian are consistent with the eigenvalues in spectral geometry and in stochastic processes. Using this spectrum, results that were only known for regular graphs can be extended to general graphs (Chung, 1997). It is for these reasons that we will, in most cases, consider the normalised version of the Laplacian. The largest eigenvalue, $\lambda_n(\mathcal{L})$, is only equal to 2 for a bipartite graph. Like the combinatorial Laplacian, the smallest eigenvalue is zero and its the multiplicity is equal to the number of connected components of the corresponding graph.

For a graph G with n vertices, $\sum_i \lambda_i(\mathcal{L}) \leq n$ and inequality only holds when G consists of isolated vertices.

2.36 Random Walk Normalised Laplacian

This variant of the Laplacian matrix is based on the random walks in the graph. It is an unsymmetric matrix defined as

$$\mathbf{L}_r = \mathbf{L}D^{-1} \tag{12}$$

The relationship between $\mathbf{L}_r$ and $\mathcal{L}$ is :

$$\mathbf{L}_r = \mathbf{D}^{1/2}\mathbb{L}\mathbf{D}^{-1/2}. \tag{13}$$

By diagonalizing ($\mathcal{L}$), Eqn.13 can be rewritten as

$$\mathbf{L}_r = (\mathbf{D}^{1/2}\mathbf{\Phi})\mathbf{\Lambda}(\mathbf{D}^{1/2}\mathbf{\Phi})^{-1}, \tag{14}$$

where $\mathbf{\Lambda}$ is a diagonal matrix of eigenvalues of ($\mathcal{L}$) and $\mathbf{\Phi}$ is the eigenvector matrix. Thus, $\mathbf{L}_r$ can is diagonizable and it shares the same set of eigenvalue as $\mathcal{L}$ though the corresponding eigenvectors are different. From this relationship, some applications would use the random matrix $\mathbf{L}_r$ in place of ($\mathcal{L}$).

A detailed account of other variants of the Laplacian matrix can be found in (Tsiatas, 2012).

2.37 Randić matrix

Let $G = (V, E)$ be a graph with vertex set $v_1, v_2, \cdots, v_n$. Let d_i denote the degree of a vertex v_i. The entries of the Randić matrix (name proposed in (Bozkurt et al., 2010)) are given by

$$\mathbf{R}_{ij} = \begin{cases} 0, & \text{if } i = j \\ \frac{1}{\sqrt{k_i k_j}}, & \text{if } v_i \text{ and } v_j \text{ are adjacent} \\ 0, & \text{otherwise.} \end{cases}$$

The Randić matrix is related to the symmetric normalised Laplacian matrix by

$$\mathcal{L} = \mathbf{I} - \mathbf{R} \tag{15}$$

2.38 Transition matrix of a random walk on a Graph

It is an $n \times n$ matrix $\mathbf{P}_G$ whose entries are given by

$$\mathbf{P}_{ij} = \frac{1}{k_i}\mathbf{A}_{ij}, \tag{16}$$

In other words, $\mathbf{P}_G = \mathbf{D}_G^{-1}\mathbf{A}$ which is the normalised adjacency matrix.

The entry at $\mathbf{P}_{ij}$ indicates the probability for a random walker moving from vertex i to j. Like any other stochastic matrix, the eigenvalues of $\mathbf{P}$ are such that $|\lambda_i(\mathbf{P})| \leq 1$.

The transition matrix is normally encountered in the study of Markov chains where its spectrum is used to compute the time it takes for the chain to reach its stationary distribution (mixing time) as discussed in (Behrends, 2000).

3 Structure of a Network

The structure of a network is a description of how nodes are connected to each other in a network. The study of network structure for purposes of understanding and predicting properties of networks was first and much embraced by chemists who used graphs to represent chemical molecules where vertices are used to represent atoms and edges to represent chemical bonds. In their work (Brown and Fraser, 1868), Crum Brown and Fraser were among the earliest researchers who put forward the idea that the properties of a chemical molecule are greatly dependent on its structure. This structure-property relationship has further been explored using graph-theoretical approach not only in chemistry but in other fields such as social networks, biological networks and many others (Mihalić and Trinajstić, 1992; Smith et al., 2004; Wellman and Berkowitz, 1988; Wey et al., 2008).

As pointed out in previous chapters, complex networks are used to represent complex systems. The behaviour exhibited by complex systems is due to the interconnectedness between system components not individual components. In order to understand behaviour and properties of these system, we need to study the structure of the corresponding networks from which we deduce properties of the networks from which the behaviour of the systems can be drawn. Let us discuss some of the properties that characterise the static structure of a network:

3.1 Average Degree of the Nearest Neighbors

The average degree of the nearest neighbor of a vertex i is given by

$$k_{nn,i} = \frac{\sum_{j=1}^{n} a_{ij} k_j}{k_i}, \tag{17}$$

where a_{ij} is the i, jth element of the adjacency matrix and n is the number of nodes in the network. This measure checks the correlation between the degrees of different vertices.

Definition 3.2 (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.

Definition 3.3 (Diameter of a network)**.** The diameter 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.

Definition 3.4 (Efficiency)**.** The efficiency, $\bar{e}$, of a network is a measure defined as

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

where $d_{i,j}$ is the shortest path between vertices i and j.

The efficiency of a network is a measure of how efficient information spreads between vertices thus networks with high values of efficiency are considered fast spreaders of information and a large percentage of vertices are reached compared their counter parts with lower values of efficiency.

Definition 3.5 (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

$$L_G = \frac{1}{n(n-1)} \sum_{i,j \in V,\ i \neq j} d_{ij}, \tag{18}$$

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).

3.6 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.

Definition 3.7 (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

$$C_i = \frac{\text{number of pairs of neighbors of } i \text{ that are connected}}{\text{number of pairs of neighbors of } i} = \frac{2t_i}{k_i(k_i-1)}, \tag{19}$$

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

$$\bar{C} = \frac{1}{n}\sum_i C_i. \tag{20}$$

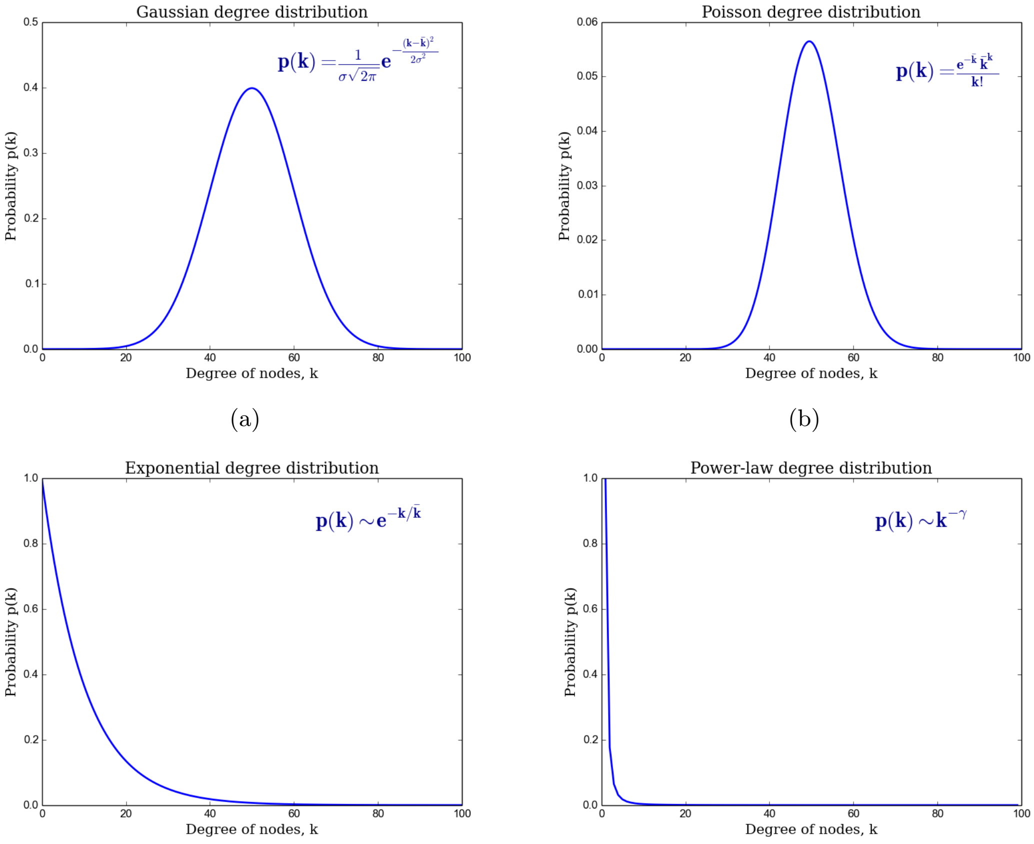
Definition 3.8 (Global clustering coefficient)**.** The global clustering coefficient is concerned with the density of triplets of nodes in a network. A triplet is defined as three nodes that are connected by either two (open triplet) or three (closed triplet) ties. Global clustering coefficient determines the overall level of clustering in a network (Opsahl and Panzarasa, 2009). Mathematically, we define the global clustering coefficient C as

$$C = \frac{3 \times \text{number of triangles}}{\text{number of connected triplets of vertices}} = \frac{\sum t_\Delta}{\sum t}, \tag{21}$$

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

3.9 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 k (Estrada, 2011). Fig. 3 is of plots of some of the common degree distributions in networks namely Gaussian, Poisson, exponential and power-law degree distributions.



(a)

(b)

(c)

(d)

Figure 3: Common degree distributions of networks: (a) Gaussian distribution. (b) Poisson distribution. (c) Exponential distribution.(d) Power-law distribution.

3.9.1 Power-law degree distribution

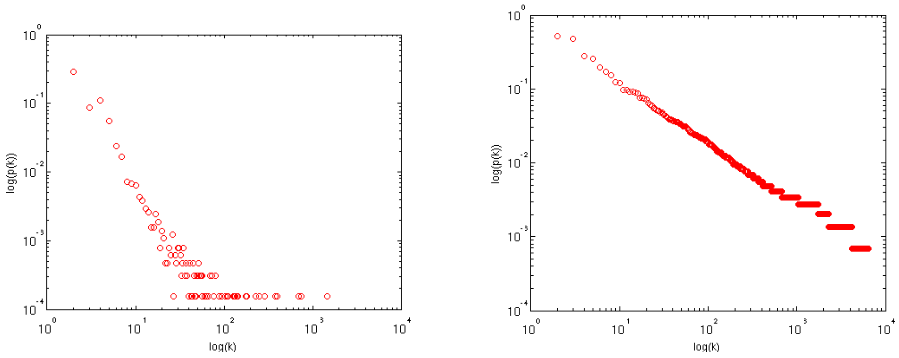
Research shows that most of the real-world networks roughly follow a power-law degree distribution. In this distribution, the probability of finding a node with degree k decreases as a negative power of degree k. This implies that in such networks, it is less likely to find a node with high degree (Estrada,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

$$p(k) = Ck^{-\gamma}, \text{ for } 2 \leq \gamma \leq 3. \tag{22}$$

Using a logarithmic scale, the plot of Equation (22) 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. 4(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. 4(b).



(a)

(b)

Figure 4: 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).

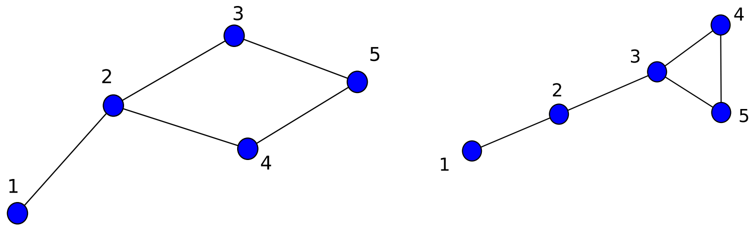
When we scale the degree by a constant factor a, we obtain

$$p(k, a) = C(ak)^{-\gamma} = a^{-\gamma} p(k) \propto p(k). \tag{23}$$

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.

3.9.2 Deduction of Structure of network using its degree distribution

The degree distribution of a graph provides useful insights about its structure though it is not conclusive as graphs 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 graph. Thus, in most cases, we cannot deduce a complete structure of a network based on knowledge of its degree distribution. To further backup this argument, we consider a simple example of two graphs both of size 5 (Figs. 5a & 5b) and whose degree distribution is the same (see table 5c). We then compute some of the structural properties of the two graphs (as in table 1) to ascertain whether their structures are the same.



(a)

(b)

Degree, k	$p(k)$
1	1/5
2	3/5
3	1/7

(c)

Figure 5: Two simple graphs (a and b) of size 5 both have the same degree distribution given in table (c).

	G1	G2
Average neighbour	1 : 3.0 2 : 1.7 3 : 2.5 4 : 2.5 5 : 2.0	1 : 2.0 2 : 2.0 3 : 2.0 4 : 2.5 5 : 2.5
Diameter	3	3
Average Degree	1.6	1.7
Clustering Coefficient	0	0.4667
Global efficiency	0.7333	0.7167

Table 1: Some of the structural properties of the two graphs (Figs. 5a & 5b) whose degree distribution is the same.

Table 1 contains some of the selected structural properties computed for the two graphs. First, the average neighbor degree for the 5 nodes for both graphs is different for example the average neighbor degree for node labelled 1 in graphs $G1$ and $G2$ is 1.20 and 1.30 respectively. The diameter, however, of both graphs is the same. On the other hand, other properties such as average degree, clustering coefficient and global efficiency have different values for both graphs. For example graph $G1$ has clustering coefficient of 0 yet graph $G2$ has a value of 0.4667. From the two values, we can tell that the structure for the two graphs is different. We can thus conclude that we cannot totally rely on the similarity in degree distribution for both graphs to deduce that the graphs under study are structurally similar since other structural properties indicate otherwise.

3.10 Robustness of Complex Systems

A complex systems is considered robust if it can with stand failures or perturbations that is to say a system can still perform as expected even in circumstances of failure of one or more components in the system. In other words, robustness intuitively deals with the existence of back-up possibilities. In a network, this cab be captured in the existence of alternative paths with in the network. Robustness of systems plays an important role in a number of fields for instance in Engineering, understanding robustness acts as a basis for designing communication, transportation systems, power grids that can perform basic operation despite failure of some system components. In biology, robustness explains why some mutations lead to diseases while others do not. For ecologists and environmental experts, robustness helps in predicting the failure of an ecosystem when faced with disruptive human behaviours. In general, study of system robustness aids in understanding system operation, improving system performance and designing of new robust systems. As mentioned earlier, the study of a network or graph underlying a complex system provides insights about the properties and characteristics of that systems. Thus, Barabási (Barabási, 2016) highlighted the fact that networks play a vital role in robustness of complex systems which implies that exploring robustness of the network reflects that of the system.

3.11 Robustness measures in networks

According to Ellens (Ellens and Kooij, 2013), robustness of a network is its ability to perform well when subject to failures or attacks. The attacks take on two forms namely: random attacks and targeted attacks. However, in order to tell whether a particular network is robust, there is need for a measure that quantifies the robustness. In the past, various robustness measures have been put forward by researchers (Sydney et al., 2008). We explore some of the common measures of robustness in networks. We categorise the measures as follows:

3.11.1 Connectivity-based measures

Here we consider robustness measures that are based on the connectivity of the network. These include the classical connectivity κ, edge connectivity κ_e, and vertex connectivity κ_v. Firstly, the classical connectivity κ is a measure whose value $\kappa = 1$ for graphs in which there is a path between any pair of vertices that is connected graphs and $\kappa = 0$ for unconnected graphs that is graphs in which atleast one pair of vertices for which no path exists between them. Secondly, the edge connectivity κ_e and vertex connectivity κ_v are respectively the minimum number edges and vertices that need to be removed to disconnect the graph. The inequality $\kappa_v \leq \kappa_e \leq \delta_{min}$ holds for non-complete graphs, where δ_{min} is the minimum degree of vertices in a graph.

3.11.2 Distance-based measures

1. Diameter

   The diameter of a graph, denoted as D, is the maximum distance between pairs of nodes in the graph(Wang and Chen, 2003). The diameter of a graph $G = (V, E)$ is defined as

   $$D = \max_{i,j \in V}\{d_{ij}\},$$

   where d_{ij} is the shortest path between node i and j. Based on the diameter, a graph is considered more robust if it's diameter is shorter.

2. 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

   $$L_G = \frac{1}{n(n-1)} \sum_{i,j \in V,\ i \neq j} d_{ij}, \tag{24}$$

   where d_{ij} is the shortest path between node i and j and n is the total number of nodes in G (Wang and Chen, 2003). On comparing the average path length and diameter as measures of robustness, the former is considered a better measure as it strictly decreases on addition of edges which is not necessarily the case with the latter.

3. Efficiency

   We can observe that we cannot compute the robustness of disconnected graph based on the two distance-based measures discussed previously. However, this is a possibility when we adopt the efficiency measure. The efficiency of a graph, E is defined as

   $$E = \frac{1}{n(n-1)} \sum_{i,j \in V, i \neq j} \frac{1}{d_{ij}}. \tag{25}$$

   It is important to note that these measures based on distance consider only shortest path distances which implies that other alternative paths are not put into consideration which is a disadvantage for that matter.

3.11.3 Spectral Graph measures

1. Algebraic connectivity

   Given the spectrum of the Laplacian matrix of a graph G in which the eignvalues are arranged in non-decreasing order: $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$. The algebraic connectivity is the second smallest eigenvalue λ_2 of the Laplacian. It is the most common measure of robustness.The algebraic connectivity is equal to zero if and only if the graph is unconnected. The disadvantage of this measure is the fact that it does not necessarily capture the addition of edges to a graph that is to say the value of λ_2 does not strictly increase on edge addition.

2. Number of spanning trees

   A spanning tree is a subgraph containing $n-1$ edges and no cycles. According to the Kirchoff's Matrix-Tree Theorem, the number of unique spanning trees,ξ of graph G is equal to the value of any cofactor of the Laplacian (Harris et al., 2008). It is given by

$$\xi = \frac{1}{n}\prod_{i=2}^{n}\lambda_i. \tag{26}$$

   Baras and Hovareshti suggest the number of spanning trees as a global indicator of network robustness to edge removal. It has been proved that for p close to zero, the number of spanning tree gives similar results for robustness as the reliability polynomial (Baras and Hovareshti, 2009).

3. Effective resistance

   The effective graph resistance R, also called total effective resistance or Kirchhoff index, is defined as the sum of the effective resistances over all pairs of vertices. It can be expressed in terms of the non-zero eigenvalues of Laplacian as

$$R = \sum_{1\leq i<j\leq n} R_{i,j} = n\sum_{i=2}^{n}\frac{1}{\lambda_i} \tag{27}$$

   Unlike the algebraic connectivity, the effective resistance involves not only one but all the non-zero eigenvalues of the Laplacian. It is for this result that any changes due to edge addition or removal are captured which makes the latter a better measure of robustness.

4. Natural Connectivity

   This spectral measure of robustness was put forward by Wu et al.(Wu et al.). It captures the core of robustness that is the capturing of redundancy of alternative paths. This is achieved by quantifying the weighted number of walks of all lengths in the graph. Closed walks are related to subgraphs of a graph for instance a closed walk of length $k=3$ represents a triangle. The number of closed walks of all lengths is obtained following the principle used in the computing the subgraph centrality as in (Estrada, 2011) in which the shorter closed walks have more influence than their longer counterparts. The penalisation entails dividing the sum of closed walks of length k by the factorial of k. That is,

$$S = \sum_{k=0}^{\infty}\frac{n_k}{k!}, \tag{28}$$

   where n_k is the number of closed walks of length k. We also know that,

$$n_k = trace(\mathbf{A}^k) = \sum_{i=1}^{N}\lambda_i^k, \tag{29}$$

   where λ_i is the ith largest eigenvalue of $\mathbf{A}(G)$. Substituting for n_k in Eqn.28 gives

$$S = \sum_{k=0}^{\infty}\sum_{i=1}^{N}\frac{\lambda_i^k}{k!} = \sum_{i=1}^{N}\sum_{k=0}^{\infty}\frac{\lambda_i^k}{k!} = \sum_{i=1}^{N}e^{\lambda_i}. \tag{30}$$

   From Eqn. 30, we observe two facts. First, the weighted sum of closed walks can be obtained from the spectrum of the Adjacency matrix of a graph. second, the sum S will be a large number for large N and thus the need to scale S. The scaled version of S, termed as the 'average eigenvalue' and denoted by $\bar{\lambda}$ is given by

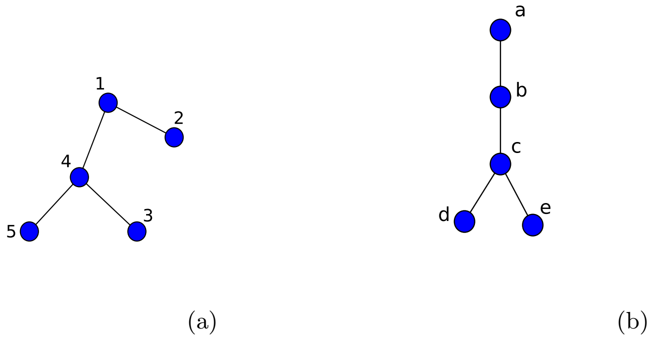
$$\bar{\lambda} = \ln\Big(\frac{S}{N}\Big) = \ln\Big(\frac{\sum_{i=1}^{N}e^{\lambda_i}}{N}\Big). \tag{31}$$

   Unlike the algebraic connectivity, the natural connectivity changes monotonically when edges are added or deleted which is one of the desired properties of a robustness measure.

3.12 Graph Similarity

Graph similarity has a wide range of applications such as web searching, chemical structure matching, comparing biological networks, synonym extraction, image clustering, social network mapping among others (Zager and Verghese, 2008; Nikolić, 2012). The notion of graph similarity can be defined in many different ways based on the application for instance similarity of graphs can be based on whether graphs are identical copies of each other (isomorphism), how much the neighbourhood of a given node in one graph is similar to neighbourhood of a given node in the other graph, how much changes (node or edge deletion, redirection or addition) can be performed to one graph to obtain the other graph, among others (Zager and Verghese, 2008). Based on different definitions of similarities, measures of similarity can be categorised as follows.

1. **Isomorphism based techniques** Two graphs are isomorphic if there exists a bijective (one-to-one and onto) function between the sets of nodes such that two nodes are connected in one graph if and only if their images under the bijection are connected (Zager and Verghese, 2008). In otherwords, two graphs are isomorphic if they are structurally identical. Graph isomorphism from G to itself is known as graph automorphism. Formally, an automorphism of a graph $G = (V, E)$ is a permutation σ of the vertex set V, such that the pair of vertices (u, v) form an edge if and only if the pair $(\sigma(u), \sigma(v))$ also form an edge for example, vertex-transitive graphs are graphs for which any pair of vertices u and v, there is an edge-preserving isomorphism mapping u to v. Fig. 6 is an example of two isomorphic graphs.



(a)

(b)

Figure 6: Two simple isomorphic graphs

Two graphs are similar if they are isomorphic, or they have isomorphic subgraphs (minimum or maximum common subgraphs) for which the larger the subgraph then the greater the similarity between the two graphs, or if one graph is isomorphic to a subgraph of another graph (subgraph isomorphism). Subgraph isomorphism can be used in image analysis to ascertain whether a given object is part of another object or a group of objects for instance in scenery study. Detailed algorithms for the three similarity measures can be found in (Weinberg, 1966; Levi, 1973; Ullmann, 1976)

Alternatively, another known technique is graph edit-distance (considered a generalisation of graph isomorphism) which involves transforming one graph into another by performing edit operations such as edge or node deletions, additions or substitutions, among others. Each operation is associated with a cost and the sequence of operations with the minimum cost is attained which amounts to a measure of similarity between the two graphs (Gao et al., 2010).

2. **Feature extraction** These are statistical methods that measure graph similarity based on properties of graph structure such as degree distribution, betweeness measures, diameter, eigenvalues and many others. Graphs are counted similar if their respective aggregated properties are assessed based on certain similarity measures and similarity is established. This method is very powerful as it maps graphs to various statistics that are quite smaller and easier to work with compared to original graphs. However, one draw-back of this method is the choice of the statistics which at

times may not give intuitive results. Take an example of graphs with different number of nodes but with the same diameter, these graphs may be considered similar which is not actually true.

3. **Iterative methods** These methods are based on the idea that two nodes are similar if their neighbourhoods are similar. The methods involve iterative processes in which nodes share similarity scores at each iteration until convergence is attained. A detailed review of some of the algorithms can be obtained in (Jeh and Widom, 2002; Melnik et al., 2002; Zager and Verghese, 2008)

In this work, we use the concept of graph similarity as a basis for performing clustering on a given set of graphs, also known as graph clustering. Here, we aim at grouping graphs into classes such that graphs belonging to the same class are similar and those belonging to different classes are dissimilar.

3.13 Voronoi Diagrams and Delaunay Triangulation

3.13.1 Voronoi Diagrams

Voronoi diagrams (also called Voronoi tessellations, Voronoi decompositions, or Dirichlet tessellations) are important geometrical structures that are found almost everywhere in the world. They have a wide range of applications such as modeling of biological structures such as cells, study of growth patterns of forests and forest canopies in ecology, tracing sources of infections in epidemics, in finding clear routes in autonomous robot navigations, among others. However, in later chapters, we will explore the application of Voronoi diagrams to image segmentation as explained in (Stoica, 2011).

Definition 3.14. A Voronoi diagram is a special kind of decomposition of a metric space (or plane) into regions based on distances to a specified discrete set of objects in the space (usually denoted by a set of points normally referred to as seeds, sites or generators), according to the nearest-neighbor rule, such that each point is associated with the region of the plane closest to it. The regions are referred to as Voronoi cells. The Voronoi vertices are the vertices of a complex formed from a set of all Voronoi cells and their faces.In other words, a Voronoi vertex is the common boundary of 3 adjacent cells. The Voronoi edge, on the otherhand, is the common boundary of two adjacent cells.

Let us consider a simple example of n ambulances placed at different spots of a city. The spots are considered as a subset of points denoted by $S = p_1, p_2, \cdots, p_n$. Let us assume that the distance between two points is given by the Euclidean distance function:

$$\ell_2 = d[(a_1, a_2), (b_1, b_2)] = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2}. \tag{32}$$

Suppose other factors remain constant and that the nearest ambulance is contacted in cases of emergency, then the area to be served by an ambulance located at spot p_k is given by the region R_k around point p_k.

Before we go any further, we first discuss some of the common terminology used in Voronoi diagrams:

Definition 3.15 (Convex Set)**.** In a Euclidean space, a convex region (or set) is a region where, for every pair of points within the region, every point on the straight line segment that joins the pair of points is also within the region.
Definition 3.16 (Convex Hull)**.** The convex hull (or convex envelope) of a set P of points in the Euclidean plane or in a Euclidean space is the smallest convex set that contains P.
Definition 3.17 (Simplex)**.** A k-simplex is a k-dimensional polytope (a geometric object with flat sides) which is the convex hull of its $k + 1$ vertices.
Definition 3.18 (Planar graph)**.** A planar graph is a graph that can be drawn in the plane without any edges crossing. A planar graph drawn in this way divides the plane into regions bounded by the edges of the graph. These regions are called faces.
Definition 3.19 (Dual Graph)**.** The dual graph G^* of a plane graph G is a plane graph whose vertices correspond to the faces of G and whose edges correspond to the edges of G.

Voronoi diagrams exhibit interesting properties which include:

- The closest pair of points corresponds to two adjacent cells in the Voronoi diagram.
- Assume the setting is the Euclidean plane and a group of different points are given. Then two points are adjacent on the convex hull if and only if their Voronoi cells share an infinitely long side.
- The Voronoi diagram on n points (or sites) is a planar graph with n faces and by Euler's formula for planar graphs, the number of Voronoi vertices and edges are atmost $2n-5$ and $3n-6$ respectively. The time complexity is $O(n \log n)$.
- Each point on an edge of the Voronoi diagram is equidistant from its two nearest neighbors p_i and p_j. Thus, there is a circle centered at such a point such that p_i and p_j lie on this circle, and no other site is interior to the circle.
- It follows that vertex at which three Voronoi cells $V(p_i)$, $V(p_j)$, and $V(p_k)$ intersect is equidistant from all sites. Thus it is the center of the circle passing through these sites, and this circle contains no other sites in its interior.
- If we assume that no four points are co-circular, then the vertices of the Voronoi diagram all have degree three.

3.19.1 Delaunay Triangulation

Let $P = p_1, p_2, \ldots, p_n \subset \mathbb{R}^2$ be a point set. A triangulation $\mathcal{T}$ of P is a maximal planar subdivision with vertex set P. Following the Empty circle property, $\mathcal{T}$ is a Delaunay triangulation of P if and only if the circumcircle of any triangle in $\mathcal{T}$ does not contain a point of P in its interior. Delaunay Triangulation is used in constructing Euclidean minimum Spanning Trees (EMST) used in solving the famous traveling salesman problem.

There are various methods of computing the Delaunay Triangulations namely: plane sweeping, iterative flipping from any other triangulation, randomized incremental construction, and conversion from Voronoi diagram. For this work, however, we will consider the Voronoi diagram based method. For a Euclidean space with point sites, the dual graph of the Voronoi diagram corresponds to the Delaunay Triangulation whose vertices are the point sites. Delaunay triangulations have interesting properties which include the following:

- Circum-circle property: The circum-circle of any triangle in the Delaunay triangulation is empty that is it contains no sites of P.
- Empty circle property: Two sites p_i and p_j are connected by an edge in the Delaunay triangulation, if and only if there is an empty circle passing through p_i and p_j.
- Closest pair property: The closest pair of sites in P are neighbors in the Delaunay triangulation.
- Given a point set P with n sites where there are h sites on the convex hull, in the plane, the Delaunay triangulation has $2n$–2–h triangles, and $3n$–3–h edges.
- The exterior face of the Delaunay triangulation is the convex hull of the point set.

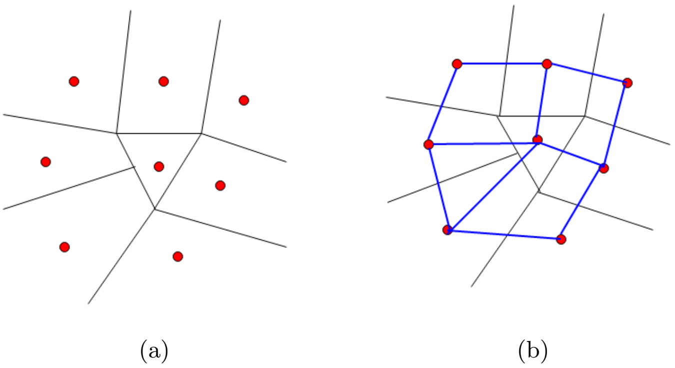


Figure 7: A sample Vorronoi diagram on 8 seeds/points (red) (a) and its corresponding dual graph (Delaunay Triangulation in blue) superimposed (b).

3.20 The Harris Corner Point Detector

The detection of feature points in an image is vital in a number of tasks such as object tracking, 3D scene reconstruction from stereo image pairs and many other tasks in machine vision (Trajković and Hedley, 1998). The Harris corner point detector is one of the famous methods used in corner points detection. This method was introduced by Harris and Stephens (Harris and Stephens, 1988) as an improvement to the Moravec's corner detector (Moravec, 1979, 1980). In his method, Moravec considered corner points as the points where there is intensity variation in all directions. The Harris corner Detector in widely used because it is simple to compute, fast and most importantly, the corner points obtained based n this method are invariant in position to rotation, scale, illumination, and partially to affine intensity changes. On the other hand, Harris corner detector method is not invariant to image scaling, however, there are various ways of going about this problem as explained in (Stoica, 2011).

The Harris Corner Detector method determines the nature of the point by computing the average change of intensity in the image when shifting a small local window in the image by small amount in any direction. For instance for a flat region, all shifts of the window result into very small changes in intensity, for an edge, a shift in the perpendicular results into a large change while for corner points, all shifts result into a large change in intensity. For a given image with intensities, I, a change due to a shift of a window w of size u, v by (x, y) is given by

$$E_{x,y} = \sum_{u,v} w_{u,v} |I_{x+u,y+v} - I_{u,v}|^2, \tag{33}$$

where E is the change due to the shift, x and y are the window's displacements in the x and y directions respectively, I is the intensity of image at a position (u, v), and w is the Gaussian window function $e^{-\frac{u^2+v^2}{2\sigma^2}}$.

The Taylors expansion of E gives:

$$E_{x,y} = \sum_{u,v} \{[I_x(u,v)x]^2 + [I_y(u,v)y]^2 + 2I_x(u,v)I_y(u,v)xy\}, \tag{34}$$

where $I_x = \partial I/\partial x$ and $I_y = \partial I/\partial y$. E is a close approximation of the local autocorrelation function given by

$$E(x,y) = \mathbb{M} \begin{bmatrix} x & y \end{bmatrix} \left(\sum_{u,v} w(u,v) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \right) \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x & y \end{bmatrix} \mathbb{M} \begin{bmatrix} x \\ y \end{bmatrix} \tag{35}$$

Matrix $\mathbb{M}$ describes the shape of the local autocorrelation function E at the origin. Let λ_1 and λ_2 be the eigenvalues of $\mathbb{M}$, according to (Harris and Stephens, 1988), the measure of corner and edge quality used for selecting core pixels known as the response function, denoted as R is given by

$$R = det(\mathbb{M}) - kTrace(\mathbb{M})^2, \tag{36}$$

where $det(\mathbb{M}) = \lambda_1\lambda_2$, $Trace(\mathbb{M}) = \lambda_1 + \lambda_2$ and k is an empirical constant such that $0.04 < k < 0.06$. A corner region with $R > 0$ is selected as a nominated corner pixel only if its response is an 8-way local maximum.

A clear step-by-step algorithm and Matlab code for the Harris corner Detector can be reviewed (Stoica, 2011).

Voronoi diagrams, Delaunay triangulation and corner point detection are very useful methods in the extraction of graphs from objects. In later chapters of this work, we demonstrate the process of image extraction from objects of selected database using a combination of the 3 concepts.

3.21 Principal Component Analysis (PCA)

Principal Component Analysis is a very important and widely used statistical technique in various applications such as image compression, face recognition, pattern identification in high dimensional data, among others. It is such a powerful tool for data analysis especially for dimension reduction of data. Before we dive into details of PCA, let us first define some of the statistical concepts that we will come across most often. Given a sample of size n of a population. We have the following definitions:
Definition 3.22 (Mean)**.** The mean, $\bar{X}$, of a sample is the average of elements of the sample and is given by

$$\bar{X} = \frac{\sum_{i=1}^{n} X_i}{n} \tag{37}$$

Definition 3.23 (Standard Deviation)**.** The standard deviation of a data set is a measure of how spread out the data is. It is average distance of a given point from the mean of the data set. The standard deviation, denoted by σ is given by

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{(n-1)}} \tag{38}$$

Definition 3.24 (Variance)**.** The variance, σ^2, is the square of the standard deviation. It is also a measure of spread of a data set. It is given by

$$\sigma^2 = \frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{(n-1)} \tag{39}$$

It is important to note that both the standard deviation and variance measure the spread of data set that is 1-dimensional.
Definition 3.25 (Covariance)**.** For data sets with more than 1 dimension, we use the covariance instead of variance (or standard deviation) measure the deviation from the mean amongst any pair of dimensions. The covariance of two dimensions x and y is given by

$$cov(X,Y) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{(n-1)} \tag{40}$$

Definition 3.26 (Covariance Matrix)**.** For a set of data with n dimensions, the covariance matrix is given by

$$C^{n\times n} = (c_{i,j}, c_{i,j} = cov(Dim_i, Dim_j)), \tag{41}$$

where $C^{n\times n}$ is a matrix with n rows and n columns, Dim_i is the ith dimension.

For example, for a data set with 3 dimensions x,y, and z, we write the covariance matrix as

$$C = \begin{pmatrix} cov(x,x) & cov(x,y) & cov(x,z) \\ cov(y,x) & cov(y,y) & cov(y,z) \\ cov(z,x) & cov(z,y) & cov(z,z) \end{pmatrix} \tag{42}$$

Given a set of data with n dimensions, we perform PCA on the data set following the steps below:

- Compute the mean for each dimension and then subtract the mean from the respective dimension for instance for dimensions x and y, subtract each x value, compute $x - \bar{x}$ and for each y value, compute $y - \bar{y}$. The data obtained after subtracting the mean is referred to as Normalised data.
- Calculate the covariance matrix as explained previously.
- Compute the eigenvalues and eigenvectors of the covariance matrix.
- For dimensional reduction and data compression, choose principal components which are the eigenvectors with large eigenvalues. The number of principal components corresponds to the new dimensionality.
- Feature vector formation. Having chosen which eigenvectors to keep, we construct the feature vector which is a matrix with the chosen eigenvectors as columns.
- New data set. We obtain the new data set (with reduced dimensions) by

$$NewData = FeatureVector^T \times NormalisedData^T \tag{43}$$

3.27 Graph Embedding

In this era where graphs manifest in a number of fields ranging from chemistry, infrastructure, social and biology, there is quite large amounts of data consisting of graphs from which information needs to be extracted. Graph analysis for purposes of obtaining useful information has been greatly explored by researchers recently and various techniques have been put forward to attaining this task. It is worth noting that effective graph analysis can aid in classification of nodes/graphs, link predictions, graph visualisations, graph clustering among others. One effective yet effective way of handling the task of graph analysis is graph embedding. Graph embedding is a way of mapping a graph into a low-dimensional vector space in which all graph information is preserved. Such information includes structural information, node label or attribute information, among others. The task of graph embedding involves an input as well as an output. The embedding input is normally a graph which could be homogeneous, heterogeneous, attributed graph or graph constructed from relational data. Given an input graph, different embedding outputs can be obtained depending on the task at hand. The outputs include node embedding, edge embedding, substructure embedding and whole-graph embedding. In this work, however, we will consider whole-graph embedding where a whole-graph is represented as a vector. Formally, we consider mapping of a set of graphs to vectorial space, that is,

$$\Phi : G \longrightarrow \mathbb{R}^n, \quad g \longrightarrow \Phi(g) = (f_1, f_2, f_3, \cdots, f_n) \tag{44}$$

4 Diffusion on networks

Dynamical processes on networks aid in the modelling of processes that occur in real-world systems for example spread of diseases in a social group, transmission of signals in brain networks, spread of information in a social network, and many others. In this work, we explore the diffusion process is one of the most popular dynamic processes studied in literature.

According to (Newman, 2010), diffusion is, among others, the movement of substance from a region of high concentration to a region of low concentration. Such substance include heat, gas, and many more.

The diffusion process on networks is used in developing models that depict processes that occur in real-world systems. The modeling scheme used in this case involves: First, identifying each node of a network with a particular component or part of the system. Secondly, for each node i, a variable σ_i is introduced that characterises its dynamical state (Barrat et al., 2008). Examples of diffusion models that adopt the mentioned modeling scheme can be found in (Estrada et al., 2011; Kasprzak, 2012; López-Pintado, 2008).

In this work, we discuss the diffusion of heat on a network in which we explore the existing diffusion models, study the equilibrium behaviour, ascertain the impact of structure on rate of diffusion, and also look at the diffusion with long range interactions. In addition, we also discuss the heat kernel of a graph, its invariants such as the trace, zeta function, heat content for both the normal diffusion processes as well as diffusion with long-range interactions.

4.1 Heat Diffusion Model

Recently, various models have been developed to depict the process of heat diffusion on networks. Here, we consider a simple model on a simple graph as explained:

Let $G = (V, E)$ be a simple connected undirected graph with vertex set V and edge set E. Suppose we randomly select a few nodes (that is, sources) to which we assign specific amounts of heat as in vector ϕ_0. Let $\varepsilon \in [0, 1]$ be the heat diffusion coefficient that controls the rate of diffusion. When ε tends to 0, heat transfer among nodes becomes difficult and as a result, heat does not spread to each of the nodes with in the network. However, as ε tends to 1, heat spreads rapidly among nodes and thus, with out loss, heat is distributed to all nodes in the network.

At each time t, we obtain the quantities, $\phi_i(t)$, of heat at each node, i. The spread of heat is considered to occur along the edges connecting nodes, that is to say, through direct interactions.

The process of heat spread through out the network can therefore be modelled by

$$\frac{d\phi_i(t)}{dt} = \varepsilon \sum_j (\mathbf{A}_{ij} - \delta_{ij} k_i)\phi_j(t), \tag{45}$$

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

$$\frac{d\boldsymbol{\phi}(t)}{dt} = -\varepsilon \mathbf{L} \boldsymbol{\phi}(t), \quad \boldsymbol{\phi}(0) = \boldsymbol{\phi}_0, \tag{46}$$

where $\mathbf{L}$ is the Laplacian matrix of a graph. Alternatively, the normalised version $\mathcal{L}$ of the Laplacian is used.

Remark 4.2. It is worth noting that for diffusion processes, the normal Laplacian $\mathbf{L}$ performs well as is its normalised counterpart. Thus, for most simulations of diffusion in this work, we use the normal Laplacian matrix.

The solution to equation 46 is

$$\boldsymbol{\phi}(t) = \boldsymbol{\phi}_0 \; e^{-\varepsilon t \mathbf{L}}. \tag{47}$$

Alternatively, the solution can be expressed as a linear combination of eigenvectors of the Laplacian matrix. That is

$$\boldsymbol{\phi}(t) = \sum_i \langle \boldsymbol{\phi}(0), \mathbf{v}_i \rangle \quad e^{-\varepsilon \lambda_i t} \mathbf{v}_i,$$

where λ_i, $\mathbf{v}_i$ are respectively the eigenvalues and corresponding eigenvectors of the Laplacian matrix and $\langle \boldsymbol{\phi}(0), \mathbf{v}_i \rangle$ is simply the projection of $\boldsymbol{\phi}(0)$ onto the set of eigenvectors (Anton and Rorres, 2007).

4.3 Equilibrium behaviour

We study the behaviour of the diffusion process and the quantities of heat at each of the nodes after an infinite period of time. For a simple undirected network, as t goes to infinity, we have

$$\lim_{t \to \infty} e^{-\varepsilon \lambda_i t} = \begin{cases} 0 & \text{if } \lambda_i > 0 \\ 1 & \text{if } \lambda_i = 0, \end{cases} \tag{48}$$

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

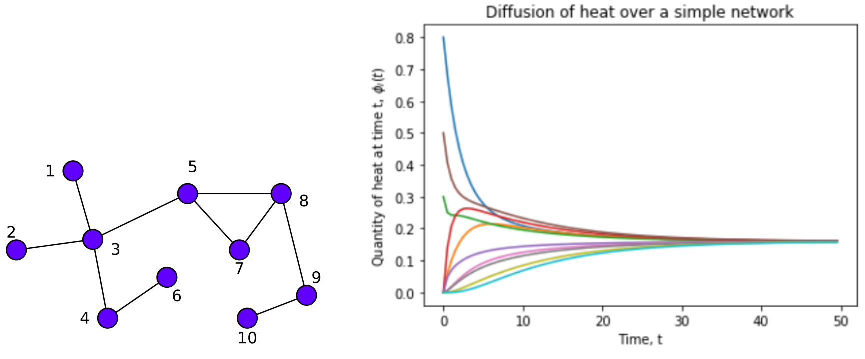
$$\lim_{t \to \infty} \boldsymbol{\phi}(t) = \langle \boldsymbol{\phi}(0), \mathbf{v^1} \rangle \mathbf{v^1}. \tag{49}$$

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

$$\lim_{t \to \infty} \phi_j(t) = \frac{1}{n} \sum_{i=1}^{n} \phi_i(0). \tag{50}$$

At steady state, the quantity $\phi_i(t)$ at each of the nodes is the same, which is the average of the initial values at all of nodes. This is because, as expected, neighboring nodes in the network will exchange heat amongst each other until all nodes attain equal amounts of heat (i.e no difference in amount for any given pair of nodes).

For better understanding of the heat diffusion model, let us consider the following simple example.
Example 4.4. Let us consider diffusion of heat over the network in Fig. 8(a). Suppose the quantity of heat at each node at time $t = 0$ is given by the vector $\boldsymbol{\phi}(0) = [0.3, 0.0, 0.8, 0.0, 0.5, 0.2, 0.0, 0.0, 0.0, 0.2]$, random values between 0 and 1. Let $\varepsilon = 0.05$. Fig. 8(b) illustrates how heat spreads over the network in Fig. 8(a).



(a) (b)

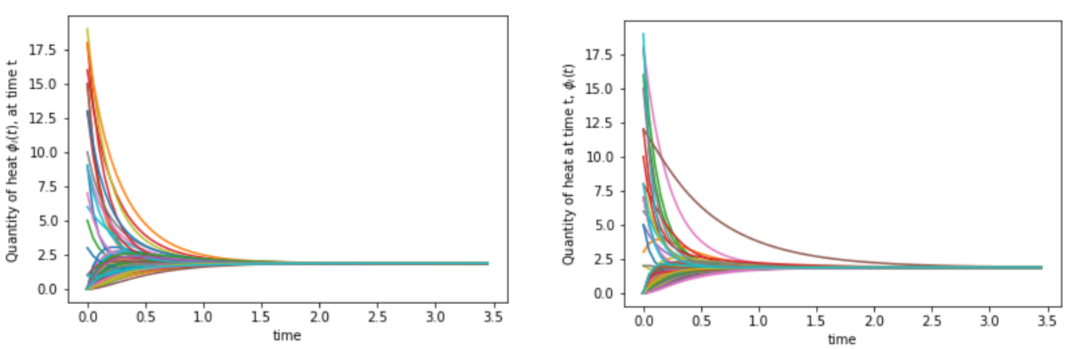
Figure 8: (b) is an illustration of the diffusion process over the network in (a).

From Fig.8,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 initially had none or little amounts of heat. The latter gain heat from the former and eventually all nodes in the network have relatively equal amounts of heat. This explains the fact that as time t increases, the quantity of heat $\phi_j(t)$ at each node tends to the equilibrium value of 0.2 which is attained as t approaches 50.

4.5 Impact of Network structure on the rate of diffusion

The structure of a network basically means the way in which nodes are connected in the network. For instance, in a regular network each node is connected to equal number of nodes, for a star network one node is positioned in a way that all other nodes are connected to it. Since diffusion on networks occurs due to the interactions within neighbourhoods of nodes, it therefore implies that the topology of a network has a strong influence on the diffusion process.

We consider two networks with different structures: one is an Erdos-Renyi (ER) network that follows a Poisson degree distribution and the other Barabasi-Albert (BA) network in which connection of nodes follows scale free power-law distribution, that is to say, the probability of finding a node with degree k decreases as the negative power of k. We perform simulations of diffusion on these networks and the results are explained: Consider ER and BA networks with $n = 100$ and average degree $\bar{k} = 2.3$, we randomly select 20 nodes to which we assign random quantities (range of 0 to 20) of heat and then allow diffusion to occur. After every time step t, we compute the quantities at each node as depicted in Figure 9.



(a)

(b)

Figure 9: Simulations for diffusion on networks following equation 46. To the left is the BA network and to the right is ER. Both networks have 100 nodes and average path length 2.3.

From Fig 9 we observe that diffusion occurs much faster in BA network than in ER network. In BA, equilibrium is reached after 1.5 time steps while in ER network equilibrium is reached much later after 2.5 time steps. This behaviour is attributed to the different structures of the two networks. The BA network is composed of highly connected nodes known as hubs that interact with a number of nodes at the same time thus fastening the diffusion process. On the other hand, the ER network is homogeneous which means there are no hubs and for that reason, diffusion occurs quite slower than in BA networks as evident in Fig. 9. Following these results, we can see that the structure of a network plays a key role in influencing the rate of diffusion and thus a determinant of how fast equilibrium is attained.

4.6 Influence of Heterogeneity on Diffusion over network

The heterogeneity of a network is the irregularity characterised by the existence of a nodes with degree significantly larger than the average degree of the network Estrada (2010); Albert and Barabási (2002); Newman (2003). The quantification of heterogeneity is one the areas where tremendous research has been on going and various measures have been introduced Estrada (2010). Here, we consider heterogeneity in scale free networks with $n = 1000$ and average degree $\bar{k} = 20$ by varying power exponent, γ. For different conductances x, we assign initial quantities of heat to each of the 200 nodes with highest degree. Figure 10 illustrates how the average quantities of heat of the selected initial diffusion nodes varies with time.

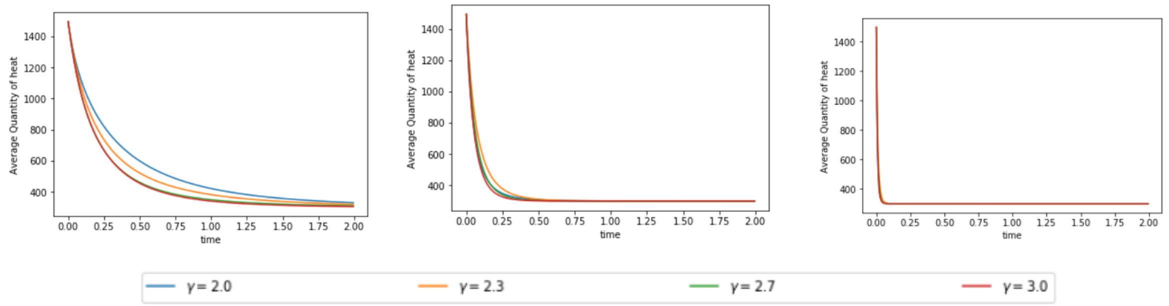
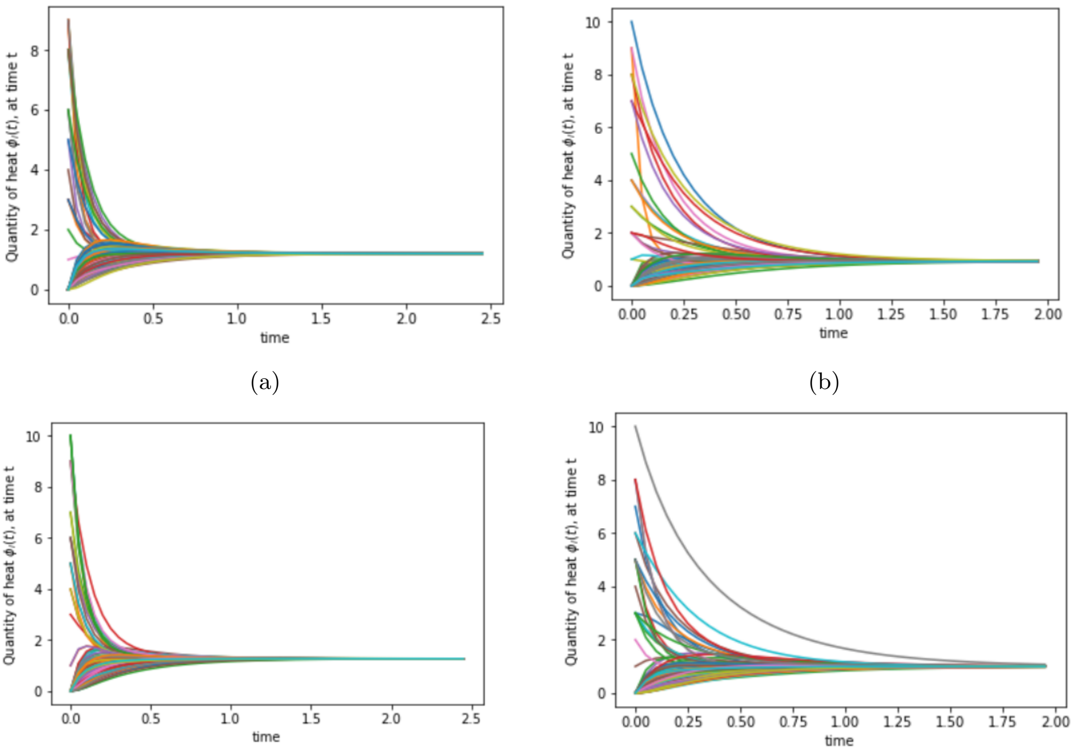


Figure 10: Plots of the average quantity of heat for 200 nodes with the highest degree centrality against time for 3 scale free networks having different values of the power exponent(2.0,2.3,2.7, and 3.0), n=1000 and average degree=6. The figures to the left, centre and right correspond to x values 0,0.1, and 0.3 respectively.

4.7 Impact of choice of Initial diffusion nodes on the diffusion process on networks

We consider a BA network and ER network, both of 100 nodes and average path length of 6. First case, we select 20 nodes in both networks with the highest degree centrality to which we assign specific amounts of heat. Second case, we randomly select 5 nodes in both networks. At each time t, we measure the average quantity of heat at the 20 nodes and the results of the simulations are illustrated by Fig. 11.



(a)

(b)

(c)

(d)

Figure 11: Results of the simulations for two networks. The top row corresponds to the BA network, (left) is illustration for which the 20 source nodes are ones with the highest degree and (right) is illustration for randomly selected source nodes. On the other hand, the bottom row is ER network for which (left) corresponds to diffusion for which the source nodes are the highest degree nodes and to figure to the right is one for which the source nodes are selected randomly.

From the simulations, we observe that when the source nodes (those from which diffusion kicks off) are chosen based on the degree, diffusion occurs much faster and equilibrium is attained quickly compared to when the source nodes are chosen randomly. We explain this observation based on the fact when the highest degree nodes initiate the diffusion process, they quickly interact with their neigbouring nodes at the same time and since their neighbourhood is big in size, this results into fast spread of heat among the nodes in the network. On the otherhand, when source nodes are randomly selected, their is a possibility of including nodes with low degree among the selection, these low degree nodes are not fast agents of heat transfer due to their small neighbourhood which results into a relatively slow diffusion process.

4.8 Diffusion on Directed Networks

A directed graph, also known as a digraph or directed network, is one in which all the edges are directed from one vertex to another.

There are various complex systems whose skeleton can be captured by directed networks. Examples include ecological networks, power grids, transportation networks, communication networks, metabolic networks, gene regulatory networks, citation networks among others. It is therefore paramount to study how dynamic processes such as diffusion, consensus, occur on such networks. There are various categories of directed networks which include:

Definition 4.9 (Weakly connected digraph)**.** A directed graph is called weakly connected if replacing all of its directed edges with undirected edges produces a connected (undirected) graph.

Definition 4.10 (Strongly connected digraph)**.** A digraph is called strongly connected if and only if any two distinct nodes of the graph can be connected via a path that respects the orientation of the edges of the digraph (Saber and Murray, 2003).

For a strongly connected digraph with atleast two distinct nodes and with no self loops, the diffusion process on this network can be modelled in a similar manner as its undirected counterpart by

$$\frac{d\boldsymbol{\phi}}{dt} = -C\mathbf{L}\boldsymbol{\phi}, \quad \boldsymbol{\phi}(0) = \boldsymbol{\phi}_0. \tag{51}$$

For undirected graph G, the graph Laplacian $\mathbf{L}$, is symmetric positive semi-definite. However, for directed graphs $\mathbf{L}$ is non-symmetric which implies that the diffusion on the former and latter graphs is not necessarily the same.

Definition 4.11 (Balanced Graphs)**.** We say the node v_i of a digraph $G = (V, E)$ is balanced if and only if its in-degree and out-degree are equal, that is, $d_{out}(v_i) = d_{in}(v_i)$. A graph G is balanced if and only if all its nodes are balanced, i.e $\sum_j a_{ij} = \sum_j a_{ji}, \forall i$.

4.12 Diffusion and Equilibrium behaviour in Directed Network

In order to understand the process of attainment of steady state in networks, we need to study the spectral properties of graph Laplacian. Let $G = (V, E)$ be a digraph with Laplacian $\mathbf{L}$ with eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_n$ in non-decreasing order.

4.12.1 Estimation of Eigenvalues of the Laplacian

Let d_{max} be the maximum node out-degree of G, then following from Gershgorin disk theorem, then all the eigenvalues of $\mathbf{L}$ are located in the following disk

$$\mathbf{D} = \{z \in \mathbb{C} : |z - d_{max}| \leq d_{max}\} \tag{52}$$

with centre at $z = d_{max} + 0j$ in the complex plane (Saber and Murray, 2003). Thus, for a strongly connected digraph G, $\mathbf{L}$ has a zero eigenvalue $\lambda_1 = 0$ and all the other non-trivial eigenvalues have non-negative real parts. Let us consider a strongly connected digraph $G = (V, E)$. Let $\boldsymbol{\phi}_0$ be the vector of quantities of heat at all nodes at $t = 0$, $C = 1$ be the diffusion coefficient. Similar to undirected case,the quantities of heat, $\boldsymbol{\phi}(t)$ at each node at a given time t is given by

$$\boldsymbol{\phi}(t) = \boldsymbol{\phi}_0 \, e^{-\mathbf{L}t}. \tag{53}$$

Theorem 4.13 (Limit Theorem for Exponential Matrices)**.** *Assume G is a strongly connected digraph with Laplacian $\mathbf{L}$ satisfying $\mathbf{L}\mathbf{v}_r = \mathbf{0}$, $\mathbf{v}_l^T\mathbf{L} = \mathbf{0}$, and $\mathbf{v}_l^T\mathbf{v}_r = 1$. Then*

$$R = \lim_{t \longrightarrow +\infty} e^{(-\mathbf{L}t)} = \mathbf{v}_r\mathbf{v}_l^T \in M_n, \tag{54}$$

where M_n denotes a set of square $n \times n$ matrices, $\mathbf{v}_r$,and $\mathbf{v}_l^T$ denote the right and left eigenvalues of $\mathbf{L}$ *associated with eigenvalue $\lambda_1 = 0$ (Saber and Murray, 2003).*

From the theorem, we deduce that for a strongly connected digraph, equilibrium state can be attained and the quantity of heat at the nodes is given by

$$\lim_{t \longrightarrow \infty} = \boldsymbol{\phi}_0 \mathbf{v}_r \mathbf{v}_l^T \tag{55}$$

It is important to note that following Equation 55, any equilibrium value x^* can be attained such that $x_i^* = x_j^*$ for all i, j. This therefore motivates the search for which classes of digraphs attain equilibrium similar to that of undirected graphs where the value at each node is the average of the initial values at all nodes in the network.

Proposition 4.14. Consider a directed network $G = (V, E)$ that is strongly connected. Then the digraph G globally attains average equilibrium if and only if $\mathbf{1}^T\mathbf{L} = \mathbf{0}$.

4.14.1 Equilibrium state for Balanced Graphs

The proposition in (Saber and Murray, 2003) states that

Proposition 4.15. Let $G = (V, E)$ be a digraph with an adjacency matrix $A = [a_{ij}]$ satisfying $a_{ii} = 0, \forall i$. Then, all the following statements are equivalent:

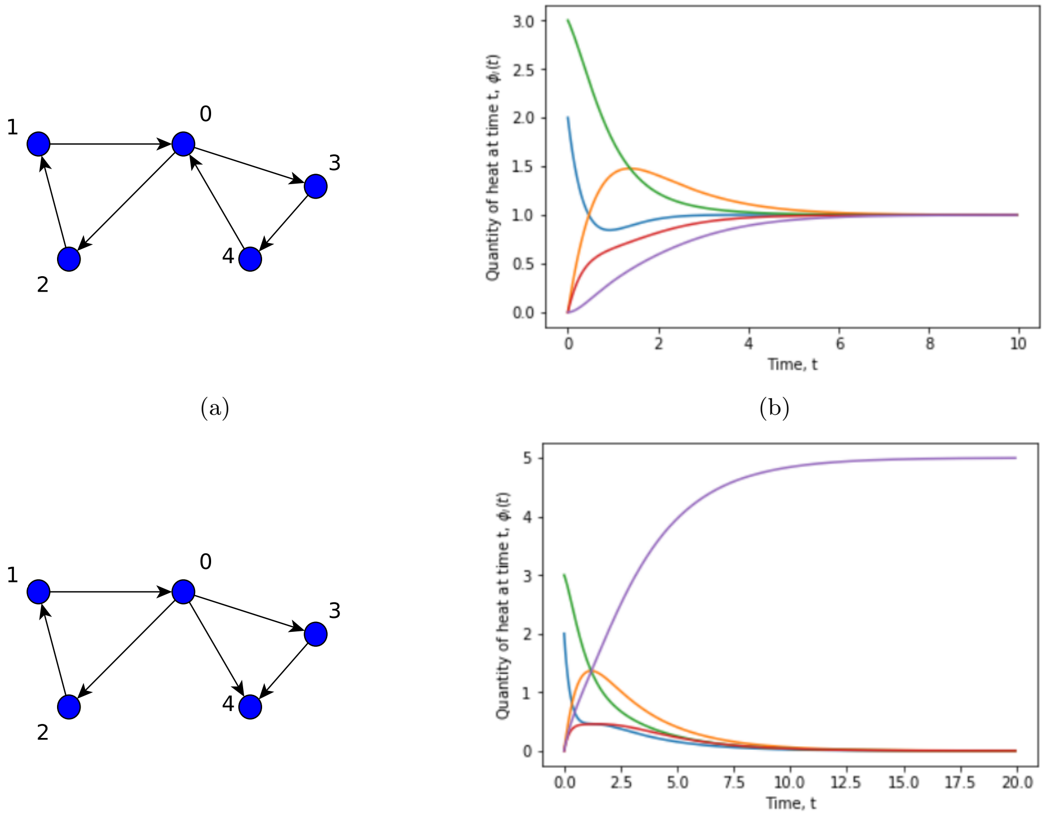
i) G is balanced,

ii) $\mathbf{v}_l = \mathbf{1}$ is the left eigenvector of the Laplacian of G associated with the zero eigenvalue, that is, $\mathbf{1}^T\mathbf{L} = 0$.

iii) $\sum_{i=1}^{n} u_i = 0, \forall x \in \mathbb{R}^n$ with $u_i = \sum_{j \in N_i} a_{i,j}(x_j - x_i)$.

Since for a balanced digraph $\mathbf{v_l}$ is an all ones vector, it therefore follows from Proposition 4.15 that at equilibrium, the value at all nodes in a balanced graph is the average of the initial values at all nodes.

Example 4.16. Let us consider two directed graphs, one is a balanced digraph and the other is not. We then assign initial quantities of heat to all nodes in the order 0 to 4 as in the vector $\phi_0 = [2, 0, 3, 0, 0]$ and set the diffusion coefficient, $C = 1$. We then obtain plots for diffusion on both graphs after a specific time t as shown in



(a) (b)

(c) (d)

Figure 12: Diffusion over different categories of directed networks. (c) is an illustration of diffusion over weakly connected and unbalanced digraph in (a). (c) is an illustration of diffusion over strongly connected and balanced digraph (c).

For the balanced graph in Fig. 12a, its Laplacian matrix $\mathbf{L}$,v_r and v_l are respectively:

$$\mathbf{L} = \begin{pmatrix} 2 & 0 & -1 & -1 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 \\ -1 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad v_r = v_l = \begin{pmatrix} 0.4472136 \\ 0.4472136 \\ 0.4472136 \\ 0.4472136 \\ 0.4472136 \end{pmatrix}$$

The values for v_r and v_l satisfy Theorem 4.13 as well as Proposition 4.15 and thus, equilibrium is attained at $x^* = 1.0$ which is the average of initial values x_0.
On the other hand, for the unbalanced graph in Fig. 12c, we have the following matrices

$$\mathbf{L} = \begin{pmatrix} 3 & 0 & -1 & -1 & -1 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad \mathbf{v}_r = \begin{pmatrix} 0.4472136 \\ 0.4472136 \\ 0.4472136 \\ 0.4472136 \\ 0.4472136 \end{pmatrix}, \text{ and } \mathbf{v}_l = \begin{pmatrix} 0.0 \\ 0.0 \\ 0.0 \\ 0.0 \\ 1.0 \end{pmatrix}$$

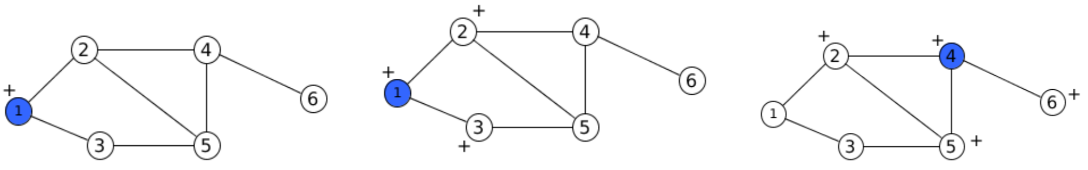
We observe that $\mathbf{L}\mathbf{v}_r = 0$ and $\mathbf{v}_l^T\mathbf{L} = 0$. However, the condition $\mathbf{v}_l^T\mathbf{v}_r = 1$ is not satisfied and therefore equilibrium cannot be attained as shown in Fig. 12. In addition, we observe that considering the structure, vertex 4 has only in coming edges which signifies that during the diffusion process, vertex 4 only receives heat from the immediate neighbours vertices 0 and 3 without giving out any due to lack of out going links. As a result, quantity of heat at vertex 4 keeps on increasing as shown in Fig. 12.

4.17 Diffusion on network with long-range interactions

The 'classical' case considers diffusion over a network where a substance under consideration, say heat, flows along the edges of the network. However, long range interactions (which are interactions between non neighbouring nodes) are evident during diffusion processes on networks in real-world. Various models to capture these interactions have beeen documented in literature for example Random Walks with Levy Flights (RWLF), Fractional Diffusion Equation (FDE), and many others. In this work, however, we account for longrange interactions using an elegant model based on k-path Laplacian matrices which was introduced by Estrada (Estrada et al., 2017b). To start with, let us understand what the k-path Laplacian matrices are.

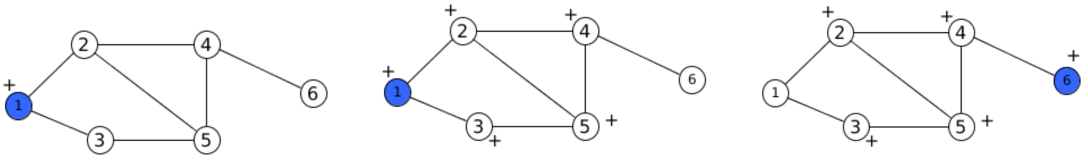
4.17.1 k-path Laplacian matrices, $\mathbf{L}_k$

The k-path Laplacian matrices are a natural generalisation of the combinatorial Laplacian of a graph (Estrada, 2012). The motivation behind this generalisation is the idea of determining whether every node in a graph can be visited by means of a process that involves hopping from one node to another separated at a distance k. We can better understand the concept of k-path Laplacian through considering a polarisation process on a network, that is to say as, suppose a particle with a positive charge resides at a given node of simple graph $G = (V, E)$ and while at that node, it polarises all nodes at a distance d from it. Consequently, the particle's movement to another is such that it hops to any nearest non-positively charged node. While at the new node, the particle polarises neighbouring nodes in the same manner as before. As a result, the particle either hops to the nearest non-positive nodes or returns to the origin node as illustrated in Fig.13 and Fig.14 for $d = 1$ and $d = 2$ respectively.



(a) (b) (c)

Figure 13: Illustration of how the polarisation analogy used as a motivation for the k-path Laplacian concept for networks. Starting with a positively charged particle at node 1 as shown in (a), taking $d = 1$, the particle polarises all its nearest neighbours at a distance d from it (that is nodes 2 and 3) as depicted in (b). The particle can therefore jump to the non-polarised nearest neighbours namely nodes 4 and 5 and 6 (though node 6 is further compared to other two alternatives). Suppose the particle jumps to node 4, similar polarisation process as the particle polarises the new nearest neighbours. The particle either jumps to node 3 or returns to node 1 as shown in (c).



(a) (b) (c)

Figure 14: Illustration of how the a charged particle navigates the network taking jumps of length $d = 2$. As discussed before, a particle starting off at node 1 will polarise neighbouring nodes separated at not more than distance 2 from it (that is nodes 2,3, 5, and 4) as shown in (b). The particle then has only an option of jumping to the non-polarised node 6 after which a similar process occurs again as in (c).

As for the 'classical' case in which traversing the graph involves subsequent hops of length 1 at a time, terminology such as walk, path, and many more are defined. In the same way, for the generalised case in which hops of various length not exceeding the diameter of a graph are taken into account, we need to define terminology as well:

Definition 4.18 (k-hopping walk)**.** A k-hopping walk of length l is any sequence of (not necessarily different) nodes $v_1, v_2, \cdots, v_l, v_{l+1}$ such that $d_{i,i+1} = k$ for each $i = 1, 2, \cdots, l$. In otherwords, this walk is referred to as a k-hopping walk from v_1 to v_{l+1} (Estrada, 2012).

Definition 4.19 (k-path degree)**.** The k-path degree $\delta_k(v_i)(k \leq d_{max})$ of a node v_i is the number of irreducible shortest-paths of length k having $v_i$as an endpoint (Estrada, 2012).

For a simple graph in Fig.15 with diameter equal to 2, the k-degree ($k \leq 2$) for each vertex is summarised in Table.

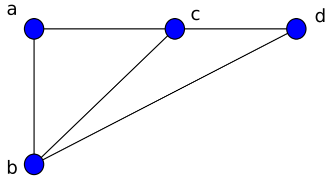


Figure 15: A Graph of size 4.

Vertex	σ_1	σ_2
a	2	1
b	3	0
c	3	0
d	2	1

Table 2: k-path degree for vertices of graph in Fig. 15.

Definition 4.20 (k-path Laplacian matrix)**.** The k-path Laplacian matrix $\mathbf{L}_k(k \leq d_{max})$ of a connected undirected graph $G = (V, E)$ is defined as the square symmetric $n \times n$ matrix whose entries are given by

$$\mathbf{L}_k(ij) = \begin{cases} \delta_k(i) & \text{if } i = j, \\ -1 & \text{if } d_{i,j} = k, \\ 0 & \text{otherwise}, \end{cases} \tag{56}$$

where $d_{i,j}$ is the shortest path distance between nodes i and j, $\delta_k(i)$ known as the k-path degree is the number of irreducible shortest paths of length k having node i as an endpoint.

For clarity, let us compute the k-path Laplacian matrices for the simple graph in Fig.15. Note that since $d_{max} = 2$ for the graph, then we consider $k = 1$ and $k = 2$.

$$\mathbf{L}_1 = \begin{pmatrix} 2 & -1 & 0 & -1 \\ -1 & 3 & -1 & -1 \\ 0 & -1 & 2 & -1 \\ -1 & -1 & -1 & 3 \end{pmatrix}, \quad \mathbf{L}_2 = \begin{pmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

In his work, Estrada proved that the k-path Laplacian matrices are positive semi-definite and they satisfy the condition:

$$\mathbf{y}^T c_k \mathbf{L}_k \mathbf{y} \geq 0 \quad \text{for } c_k > 0 \tag{57}$$

The concept of k-path Laplacians defined in Eqn.4.20 for finite undirected graphs was extended for connected and locally finite infinite graphs as follows: Consider $\Gamma = (V, E)$ to be an indirected finite or infinite graph with vertices V and edges E. We assume that Γ is connected and locally finite that is to say each vertex has only finitely many edges emanating from it. Let d be the distance metric on Γ, i.e. $d(v, w)$ is the length of the shortest path from v to w, and let $\delta_k(v)$ be the k-path degree of the vertex v, i.e.

$$\delta_k(v) := \#\{w \in V : d(v, w) = k\}. \tag{58}$$

Since γ is locally finite, $\delta_k(v)$ is finite for every $v \in V$. Denote by $C(V)$ the set of all complex-valued functions on V and by $C_0(V)$ the set of the complex-valued functions on V with finite support. Moreover, let $\ell^2(V)$ be the Hilbert space of square-summable functions on V with inner product

$$\langle f, g \rangle = \sum_{v \in V} f(v)\overline{g(v)}, \quad f, g \in \ell^2(V) \tag{59}$$

In $\ell^2(V)$ there is a standard orthonormal basis consisting of the vectors $e_v, v \in V$, where

$$e_v(w) := \begin{cases} 1 & \text{if w = v,} \\ 0 & \text{otherwise.} \end{cases} \tag{60}$$

Let $\mathbf{L}_k$ be the following mapping from $C(V)$ into itself:

$$(\mathbf{L}_k)(f) := \sum_{w \in V : d(v,w) = k} (f(v) - f(w)), \quad f \in C(V). \tag{61}$$

Definition 4.21 (k-hopping connected component)**.** A k-hopping connected component in a graph $G = (V, E)$ is a subgraph $G' = (V', E')$, $V' \subset V$, $E' \subset E$, such that there is at least one k-hopping walk that visit every node $v_i \in V'$. As mentioned earlier, the motivation of the generalisation of graph Laplacian to find the solution of the problem of whether a given graph can be k-hopped. If not, how many k-hopping connected components exist?

Remark: The number of k-hopping connected components in a connected undirected graph $G = (V, E)$ is given by $\eta_k(G) = m[\lambda_1(\mathbf{L}_k) = 0]$ (Estrada, 2012).

Example 4.22. Let us consider the graph, G in Fig.15, since $d_{max} = 2$ we compute the 1-hopping and 2-hopping connected components of G.

		no. of components	components
$\lambda_i(\mathbf{L}_1)$	**0.000** 2.000 4.000 4.000	1	1- 2- 3- 4
$\lambda_i(\mathbf{L}_2)$	**0.000** **0.000** **0.000** 2.000	3	1-3, 2, 4

Table 3: Computation of connected components of the graph Fig.15.

We see from Table. 3, that when we consider hops of length $k = 1$, there is only one connected component which is equal to the multiplicity of 0 as an eigenvalue in the spectrum given as $0, 2, 2, 4$. On the other hand, however, considering hops of length 2 ($k = 2$), we obtain 3 components which are reflective of the multiplicity of 0 eigenvalue in the respective spectrum $0, 0, 0, 2$.

4.22.1 The Generalised Laplacian Matrix

The generalised Laplacian matrix is obtained as a linear combination of the k-path Laplacian matrices and it is given by

$$\mathbf{L}_G = \sum_{k=1}^{\Delta} c_k \mathbf{L}_k \tag{62}$$

where $1 \leq \Delta \leq d_{max}$ and c_k are the coefficients (Estrada, 2012). The coefficients c_k play a crucial role in the generalisation of diffusion process on network and so determining the values of these coefficients is an important task. The values of c_k are expected to give more weight to shorter than to the longer range interactions. In (Estrada, 2012) Estrada proposed two approaches categorised as social and physical ways of influence.

4.22.2 Choice of Co-efficients, c_k

1) **Social Influence**

Here, we consider a social network where nodes represent the people with in a society and the links are the social relationship or ties among the people for instance friendship, family relations, collaboration, among others. In such networks, influence between two people connected to each other in the network can be accounted for. However, it is quite challenging to account for the indirect influence between two people that are not directly connected in the network. An approach introduced in (Estrada et al., 2011)considers that which on empirical evidence, the indirect influence or long range interactions among people can be thought of as a pre-conditioner for establishment of a new social tie. In otherwords, new social ties among humans are created as an investment in the future as justified by empirical evidence in (Estrada et al., 2011). it is quite obvious that two individuals that influence each other, probability is high that the two become friends compared to those that have no mutual influence. This process can be considered as an analogy in which the future value of money, in particular the future value of a growing annuity, is determined in quantitative finance but for this case we consider a transaction involving information instead of money. Suppose an individual A lends information to individual B whose present value is PVI at an interest rate r and for a time period t. The future value of the information FVI is given by

$$FVI = PVI(1+r)^t \tag{63}$$

Let us consider the process on a network where node v_1 lends information to node $v_(l+1)$. On assumption that information flows through the shortest path and considering a discrete time at every step, the information is transferred from v_1 to nearest neighbour v_2 at a value A and rate r. At v_2, the present value $PVI = A/(1+r)$. The information is enriched at v_2 at a growing rate of g and then transferred to v_3. The process is repeated as before and at each node information is enriched before transfer to the next node. Finally, the information at borrower node $v_(l+1)$ is $A(1+g)^{l-1}/(1+r)^l$. Thus, The cumulative present value of the information in this process is given by the sum of all the values at the nodes of the chain, that is,

$$PVI = A/(1+r) + A(1+g)/(1+r)^2 + \cdots + A(1+g)^(l-1)/(1+r)^l. \tag{64}$$

Suppose $g = r$ and $A = 1$ for the sake of simplicity, for a connected network with shortest distance between any pair of nodes denoted by $d_{i,j}$, the future value of information transmitted from i to j is:

$$FVI_{i,j} = d_{i,j}x^{di,j-1}, \tag{65}$$

where $x = 1 + r = 1 + g$. Thus, from the analogy, we can consider that the mutual influence between two nodes separated at distance k is given by the future value of the investment that the creation of a new link will represent to them.

For the social influence, we can define the coefficients in Eqn. 74 as $c_1 = 1$ and $c_{k\geq 2} = kx^{k-1}$, where $0 < x < 1/2$. The empirical parameter x also known as the conductance in (Estrada et al., 2011) controls the strength of interaction between nodes i and j separated at distance k. This implies that the strength of the casual contact between two nodes reduces with increase in social distance between them.

When we account the long-range interactions by the social influence based co-efficients, the generalised Laplacian is thus given by

$$\mathbf{L}_{G,x} = \begin{cases} \delta_{Gv_i} & \text{if } i = j \\ -1 & \text{if } i \neq j \text{ and } v_i \text{ is adjacent to } v_j \\ -kx^{k-1} & \text{otherwise,} \end{cases} \tag{66}$$

where δ_{Gv_i} denotes the generalised degree.

In modelling the spread of epidermic, Estrada in (Estrada et al., 2011) considered two types of contacts that is close contacts which are frequent interactions among individuals and casual or long range interactions are the non frequent encounters among individuals which facilitate the spread of infections.The latter which were considered as non-random where accounted by use of the social influence approach.

2) **Physical Influence**

It is observed in many man-made and naturally evolving systems that communication among the agents of the system follows a spatial decay as illustrated in sensor systems where sensors far away from the target display low noise-signal ratio as a result of attenuation (spatial decay) of signal energy, in earthquake incidences where the aftershocks follow a spatial decay, that is, areas further from the main shock are less affected compared to nearer areas. This spatial decay takes on the form $r^{-\alpha}$, where r is the distance from the main shock. Other examples of similar physical scenarios include the brain in which the interconnectivity certain neurons in mammalian neo-cortex decays exponentially with the intersomatic distance, and many others. Following from the physical scenarios, a similar idea is used in accounting for long-range interactions between a given pair of nodes separated at a distance k ($k$is the length of the shortest path between them) where by weights are assigned based on the fact that the longer the distance of separation, the weaker the strength of long-range influence. As Estrada (Estrada, 2012) suggested, we can model the long-range interactions using two laws namely:

i) **Laplace transform**

Here, the rate at which long-range influence weakens with increase in distance d (i.e decay rate) is exponential. It is given by $e^{-\lambda k}$. We thus obtain the generalised Laplacian from the Laplacian transformed Laplacian matrices of a graph by

$$\mathbf{L}_G = \mathbf{L} + \sum_{k=2}^{\infty} e^{-\lambda k}\mathbf{L}_k, \tag{67}$$

where $\lambda > 0$. Thus, the coefficients of Eqn. 74 are $c_1 = 1$ and $c_{k\geq 2} = e^{-\lambda k}$.

ii) **Mellin transform**

The decay rate with distance k in this case follows a power law, that is, k^{-s} where $s > 0$. The generalised Laplacian matrix can be computed from the Mellin transformed path Laplacians by

$$\mathbf{L}_G = \sum_{k=1}^{\infty} k^{-s}\mathbf{L}_k, \tag{68}$$

For this case, the coefficients of Eqn. 74 are $c_k = k^{-s}$. In (Estrada et al., 2017b), it is shown that normal diffusion occurs only when $s > 3$. On the other hand, superdiffusion occurs when $1 < s < 3$ with superdiffusive exponent being $\kappa = \frac{2}{s-1}$, which leads to arbitrary values for $\kappa \in (1, \infty)$.

Given a finite graph $G = (V, E)$ for which both direct interactions and long-range interactions are taken into account. we account for the latter using two models namely the Laplace and Mellin transforms of the path Laplacian matrix from which the generalised Laplacian matrix, $\mathbf{L}_G$, is derived. It is thus given by

$$\mathbf{L_G} = \tilde{\mathbf{L}}_\tau = \begin{cases} \mathbf{L} + \sum_{k=2}^{\Delta} e^{-\lambda k}\mathbf{L}_k, & \text{for } \tau = \text{ Laplace}, \lambda > 0 \\ \sum_{k=1}^{\Delta} k^{-s}\mathbf{L}_k, & \text{for } \tau = \text{ Mellin}, s > 0, \end{cases} \tag{69}$$

where λ and s are positive constant parameters for the Laplace and Mellin transforms respectively and Δ is the diameter of G.

4.22.3 Properties of the Generalised Laplacian Matrix

- The generalised matrix $\mathbf{L}_G$ is real and symmetric which follows from the fact that $\mathbf{L}_G$ is a linear combination of real and symmetric k-path matrices.
- The generalised matrix is also a positive semi-definite matrix.

  Proof. For any column vector $\mathbf{y}$

  $$\mathbf{y}^T\mathbf{L}_G\mathbf{y} = \mathbf{y}^T(c_1\mathbf{L}_1 + c_2\mathbf{L}_2 + \cdots + c_\Delta\mathbf{L}_\Delta)\mathbf{y} = \mathbf{y}^T c_1\mathbf{L}_1\mathbf{y} + \mathbf{y}^T c_2\mathbf{L}_2\mathbf{y} + \cdots + \mathbf{y}^T c_\Delta\mathbf{L}_\Delta\mathbf{y} \tag{70}$$

  Since $\mathbf{y}^T c_k\mathbf{L}_k\mathbf{y} \geq 0$ for $c_k > 0$ and $1 \leq k \leq \Delta$ as in Eqn. 57, then

  $$\mathbf{y}^T\mathbf{L}_G\mathbf{y} \geq 0 \tag{71}$$

  □

- Like for the normal Laplacian matrix, zero is always an eigenvalue of $\mathbf{L}_G$ with eigenvector, $\mathbf{1}$, an all ones vector.
- Behaviour of eigenvalues with change in s (or λ). As discussed in (Estrada et al., 2017a), for a graph $G = (V, E)$ with generalised Laplacian matrix $\mathbf{L}_G$ formed by the Laplace or Mellin transform of path Laplacian, its eigenvalues behave as follows:

  $$\left(\frac{\mu_i}{N}\right) \longrightarrow \begin{cases} \left(\frac{\lambda_i}{N}\right), & \text{if } \lambda, s \longrightarrow \infty \\ 1, & \text{if } \lambda, s \longrightarrow 0, \end{cases} \tag{72}$$

  where μ_i and λ_i are the eigenvalues of $\mathbf{L}_G$ and $\mathbf{L}$ respectively and N is the size of the spectrum.

4.22.4 Time Complexity of the Generalised Laplacian matrix

In computing of the generalised Laplacian matrix $\mathbf{L}_G$, we need to compute k-path Laplacian matrices for $1 \leq k \leq d_{max}$ which we obtain from the distance matrix $\mathbf{D}$. The matrix $\mathbf{D}$ is obtained by computing all pairs of shortest paths between nodes in the network. There are various algorithms used in computing the all-pairs shortest paths in a given network as discussed in (Cherkassky et al., 1996). The most common one of these algorithms is the Dijkstra's with time complexity $O(mn + n^2 \log n)$ where n and m are the number of nodes and links respectively (Dijkstra, 1959; Fredman and Tarjan, 1987). Though improvements to Dijkstra's algorithm have been developed (Pettie, 2002; Thorup, 1999; Seidel, 1995), for very large networks, however, computation of the generalised Laplacian matrix still remains a computationally cumbersome task especially as the size of the network increases. For this reason, research directed towards developing more efficient algorithms for solving the all-pairs shortest path problem for very large networks would be of utmost importance.

4.22.5 Generalised Diffusion Model

In this case, we consider diffusion on a graph where by both direct and long-range interactions are involved. One interesting study of long-range interactions is by Estrada on modelling epidermic spread in networks (Estrada et al., 2011). Here, the long range interactions are considered to be nonrandom and depend on the social distances between individuals in the social network. In this section, we consider diffusion on network which involves not only direct interactions but also interactions between non-neighboring nodes which are referred to as indirect interactions. This type of diffusion which we call the generalised diffusion is obtained by substituting $\mathbf{L}_G$ for $\mathbf{L}$ in Eqn. 46. That is

$$\frac{d\boldsymbol{\phi}}{dt} = -\varepsilon\mathbf{L}_G\boldsymbol{\phi}, \quad \boldsymbol{\phi}(0) = \boldsymbol{\phi}_0, \tag{73}$$

where $\mathbf{L}_G$ is the generalised Laplacian matrix.

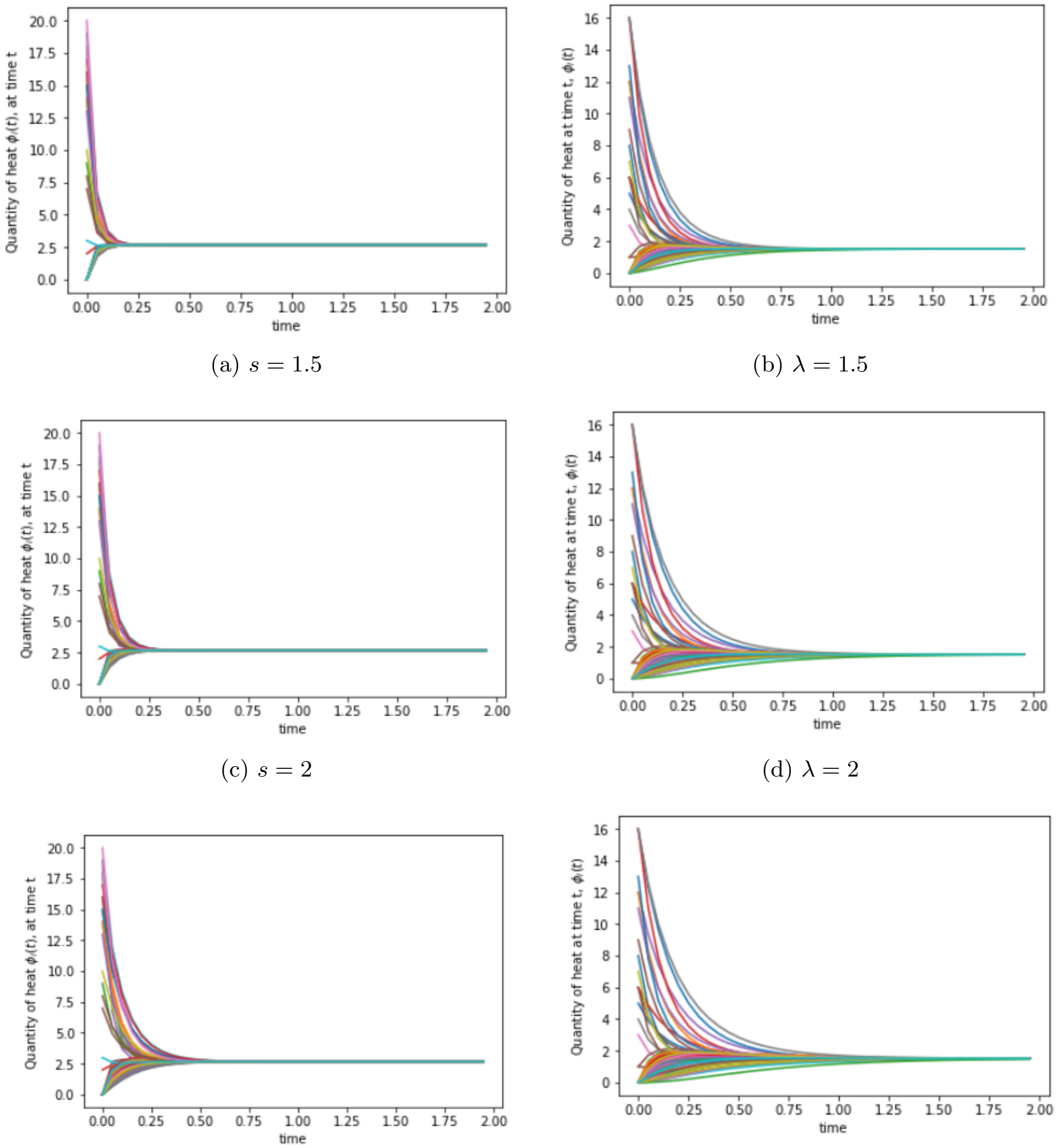
We then consider generalised diffusion on graph where interactions are both short-range and long-range. The long-range interactions are accounted for by use of k-path Laplacian matrices. Following this generalisation, Eqn. 73 then becomes

$$\frac{d\boldsymbol{\phi}}{dt} = -\varepsilon\Big(\sum_{k=1}^{\Delta} c_k \mathbf{L}_k\Big)\boldsymbol{\phi}, \quad \boldsymbol{\phi}(0) = \boldsymbol{\phi}_0, \tag{74}$$

It is necessary to consider long-range interactions in studying diffusion on networks due to the fact that many real-world networks consist of highly connected clusters which are poorly linked amongst themselves. In network theory, such clusters are referred to as communities. Within individual communities, connection between nodes is very high but the interconnection between communities is very poor. Therefore, diffusion with in a particular community occurs faster and as result equilibrium can be attained easily for different communities. On the otherhand, diffusion between nodes belonging to different communities is quite slower when we consider only direct interactions among nodes. However, it is observed that in many real-networks made up of such communities, equilibrium is attained faster despite the limited direct interactions between nodes in different communities. This behaviour could possibly be justified by the long-range influences between non-nearest nodes in the network.

4.23 Comparison of Mellin and Laplace based Generalised Diffusion

Here, we consider two networks of different structures that is the Erdos-Renyi(Erdös and Rényi, 1959; Karoński, 1982; Newman et al., 2002) and Barabasi networks (Barabási and Albert, 1999; Newman et al., 2002) of size 100 and average path length approximately equal to 2.3. We consider a random selection of 20 nodes from each of the network and randomly generated amounts of heat are assigned to the selected nodes. The rest of the nodes have $Q_i = 0$ at $t = 0$. We then perform simulations using Eqn. 73 for both networks for different values of s and λ following the Mellin and Laplace transform based diffusion with long-range interactions. Equilibrium is attained when $|Q(i)(t) - Q_j(t) \leq 10^{-4}|$ for each pair of nodes in the graph.



(a) $s = 1.5$

(b) $\lambda = 1.5$

(c) $s = 2$

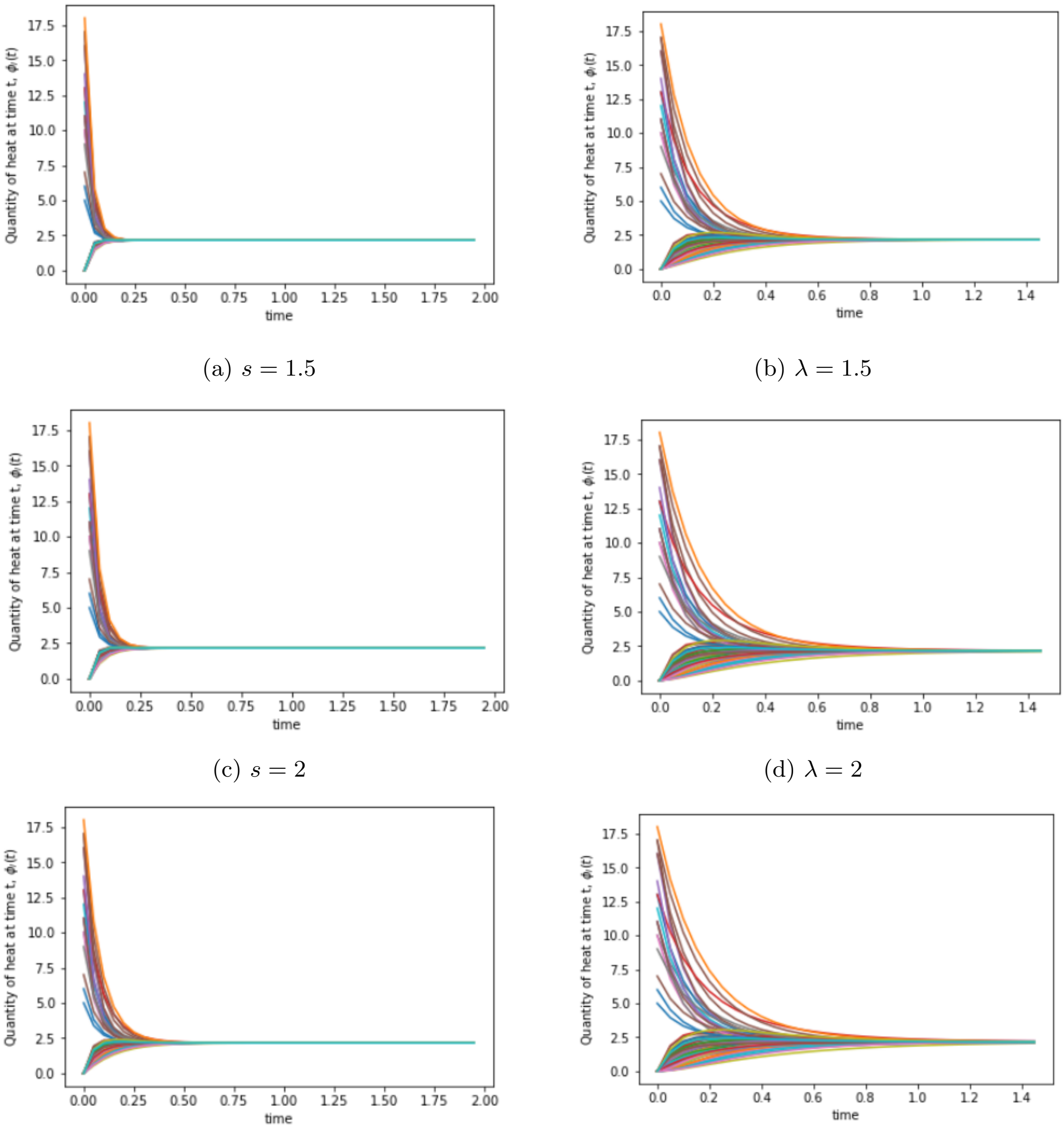
(d) $\lambda = 2$

(e) $s = 3$

(f) $\lambda = 3$

Figure 16: Simulations for diffusion on Barabasi network of 100 nodes for which long-range interactions are accounted for using the Mellin and Laplace transforms of the k-path Laplacian matrices using $s = \lambda = 1.5, 2$ and 3. The left column corresponds to the Mellin while the left column corresponds to the Laplace.

We observe from simulations in Fig. 16 for both the Mellin and Laplace transforms, diffusion due to both direct and long-range interactions becomes less faster as the values of s and λ respectively are increased. For instance at $s = 1.5$, equilibrium is attained at about 0.25 time steps compared to $s = 3$ where equilibrium is attained at about 0.50 time steps which is double the time for the former. It is however important to note that though diffusion occurs faster in both the Mellin and Laplace cases, it is evident that it is much more faster in the former than the later. For instance,we observe that equilibrium is attained at about 0.25 and 1.0 time steps for $s = 2$ and $\lambda = 2$ respectively.



(a) $s = 1.5$

(b) $\lambda = 1.5$

(c) $s = 2$

(d) $\lambda = 2$

(e) $s = 3$

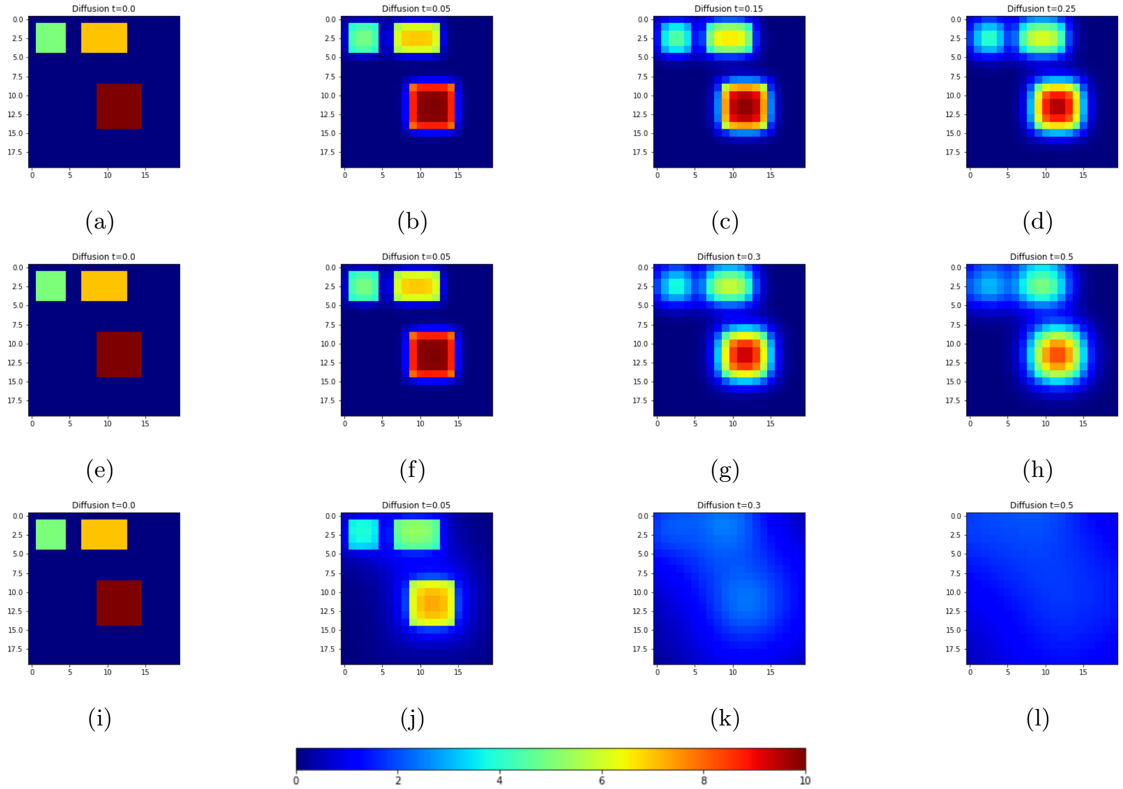
(f) $\lambda = 3$

Figure 17: Simulations (performed using Eqn. 73) for diffusion on Erdos-Renyi network of 100 nodes for which long-range interactions are accounted for using the Mellin and Laplace transforms of the k-path Laplacian matrices using $s = \lambda = 1.5, 2,$ and 3. The left column corresponds to the Mellin while the left column corresponds to the Laplace.

4.23.1 Simulations of diffusion on lattice

We consider a 2-dimensional discrete grid in which each point is connected to 8 of its nearest neighbours. Initially, we assign heat quantities to all the points on the grid and then we investigate how the diffusion process occurs and at each time t.

Let us take a 20 by 20 grid on which we assigned heat quantities of amounts 5(green), 7(yellow) and 10(red) to selected blocks of the lattice and the rest are assigned zeroes (dark blue). We consider two cases of diffusion over the grid namely one through direct interactions of nearest neighbours and the other through both direct and long-range interactions. The latter are accounted for by the Mellin and Laplace transforms of the path Laplacian matrices following equations 67 and 68 respectively.



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

Figure 18: Sample illustrations for progression of diffusion over a 20×20 lattice with initial heat quantities indicated by coloured blocks. Diffusion state is captured at different time steps that is, from left to right, $t = 0, 0.5, 3$ and 5 respectively. The top row (a - d) correspond to diffusion through direct interactions only. The middle row (e - h) and bottom row (i - l) correspond to diffusion with long-range interactions accounted for by the Laplace and Mellin transforms of path Laplacians at $\lambda = s = 3$ respectively.

At the start, $t = 0$. Both cases (direct and long-range cases) are in the same state. As time progresses, we observe that diffusion occurs much faster in the cases where long-range interactions are involved (middle and bottom rows) compared to the case where diffusion occurs through direct interactions only (top row). However, it is evident from the illustrations that the case where long-range interactions are accounted for by Mellin transform, diffusion occurs much more faster than the Laplace based counterparts. For example, at $t = 5$ almost the whole lattice is at equilibrium (light blue coloured blocks) for the Mellin based case (see Fig 18l) yet for the the Laplace based case, (see Fig 18h), different blocks contain different amounts of heat that is, equilibrium is not yet attained.

5 The Heat Kernel

The heat kernel associated with the diffusion equation has been proved very useful in a number of applications for instance identification of communities in graph, partitioning of graphs, as a pagerank of a graph, as a means of embedding a graph into a pattern space, among others (Chung, 2007, 2009; Kloster and Gleich, 2014). As discussed earlier, diffusion of heat on a graph can be modelled by the equation

$$\frac{d\phi}{dt} = -\mathbf{L}\phi, \tag{75}$$

where $\mathbf{L}$ is either the Laplacian matrix or its normalised version.

The heat kernel is the fundamental solution to the diffusion equation (75) and it is obtained by exponentiating the Laplacian eigensystem over time. It is given by

$$\mathbf{H}_t = e^{-t\mathbf{L}} \tag{76}$$

It literally describes the flow of substance (heat) across edges (direct interactions) in the graph (Xiao et al., 2009). On applying spectral decomposition to Equation 76, we have

$$\mathbf{H}_t = \mathbf{V}e^{-t\mathbf{\Lambda}}\mathbf{V}^T = \sum_{i=0}^{n} e^{-\lambda_i t}\mathbf{v}_i\mathbf{v}_i^T \tag{77}$$

where λ_i are the eigenvalues of $\mathbf{L}$ in a non-decreasing order $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ (or $0 \leq \lambda_i \leq 2$ for normalised Laplacian) and $\mathbf{v}_i$ is the eigenvector corresponding to the eigenvalue λ_i (Anton and Rorres, 2007).

For a graph $G = (V, E)$, the heat kernel matrix of the graph is an $|V| \times |V|$ matrix whose entry for a pair of node p, q is given by

$$\mathbf{H}_t(p,q) = \sum_{i=1}^{|V|} e^{-\lambda t}\mathbf{v}_i(p)\mathbf{v}_i(q) \tag{78}$$

We should note that for any two vertices p and q, $\mathbf{H}_t(p,q) \geq 0$ (Chung, 1997).

When t tends to zero, the kernel behaviour can be obtained from the Taylor's expansion of Equation 76 which is

$$e^{-\mathbf{L}t} = \sum_{k=0}^{\infty} \frac{(-t)^k}{k!}\mathbf{L}^k = \mathbf{I} - t\mathbf{L} + \frac{t^2}{2!}\mathbf{L}^2 + \frac{t^3}{3!}\mathbf{L}^3 + \cdots \tag{79}$$

Thus,

$$\lim_{t \longrightarrow 0}\left(e^{-t\mathbf{L}}\right) = \mathbf{I} - t\mathbf{L}. \tag{80}$$

It therefore implies that for t tending to zero, the heat kernel depends on the local connectivity structure of the graph. On the other hand, as t tends to infinity, following from Equation 77

$$\lim_{t \longrightarrow \infty}\left(e^{-t\mathbf{L}}\right) = \mathbf{I} - e^{-\lambda_2 t}\mathbf{v}_2\mathbf{v}_2^T. \tag{81}$$

From Equation 81, it is evident that for large t, the heat kernel behaviour is determined by the global structure of the graph.

5.1 Heat Kernel Trace

The combinatorial trace of the heat kernel, denoted as $Tr(H_t)$, is the sum of the entries at the main diagonal of the heat kernel matrix. It is given by

$$Tr(\mathbf{H}_t) = Tr(\mathbf{V}e^{-t\mathbf{\Lambda}}\mathbf{V}^T) = Tr(e^{-t\mathbf{\Lambda}}(\mathbf{V}^T\mathbf{V})) = Tr(e^{-t\mathbf{\Lambda}}) \tag{82}$$

Thus, the trace of the heat kernel, $Z(t)$, is a function whose parameter are the eigenvalues of the Laplacian matrix and whose argument is time. It is given by

$$Z(t) = Tr(\mathbf{H}_t) = \sum_p \mathbf{H}_t(p,p) = \sum_i e^{-\lambda_i t}, \tag{83}$$

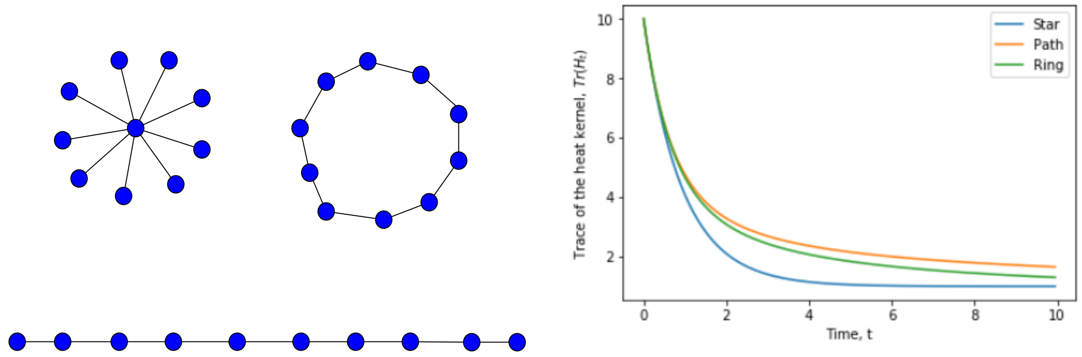
where λ_i is the eigenvalue of the normalised Laplacian matrix (Xiao et al., 2009). From Equation 83, it is evident that the trace of the heat kernel is invariant to permutations. For a connected graph, Equation 83 can be written as

$$Z(t) = 1 + e^{-\lambda_2 t} + e^{-\lambda_3 t} + \cdots + e^{-\lambda_{|V|} t} \tag{84}$$

Chung (Chung, 1997) pointed out that in spectral geometry, various invariants of the Riemannian manifold can be extracted by estimating the heat kernel. Using the trace of the heat kernel instead of the heat kernel helps overcome the cumbersome task associated with computing the heat kernel as the trace can capture the essential part of the heat kernel and can be computed in polynomial time (following from Equation 83). The trace is thus an effective tool in capturing graph properties as well as major invariants. For example, for a vertex-transitive graph such as complete graph, k_4, for all vertices, v, the entry $\mathbf{H}_t(v,v)$ is the same which makes computation of its trace much easier.

5.1.1 Heat kernel trace as a Graph Analysis Technique

According to Xiao (Xiao et al., 2009), the trace of the heat kernel has a potential applicability to distinguishing graphs with different topologies based on the shape of the curves obtained by plots of the trace of the heat kernel as a function of time. Let us consider 3 simple graphs namely a star, path and 2-regular graph of size 10. Fig 19 shows the plot of heat kernel trace against time for the three graphs.



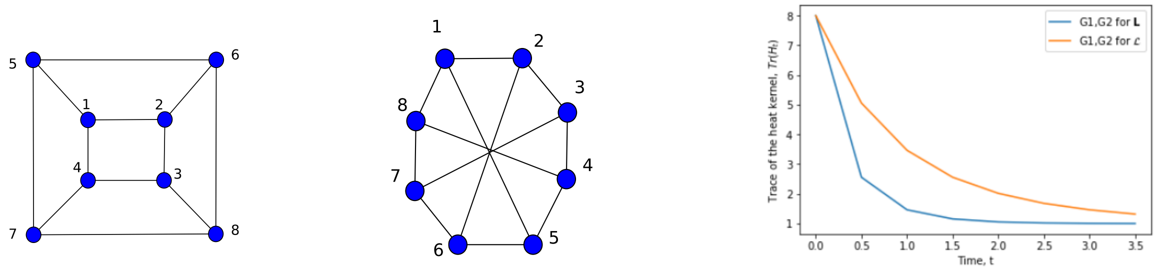
(a)

(b)

Figure 19: (a) are the three graphs used for which analysis is performed. (b) plot of the heat kernel trace against time for star (blue), path (orange) and regular (green) graphs.

From (22b), we observe that since the 3 graphs have different topologies, the corresponding curves take on different shapes as well , that is to say, the curves are distinct. It is evident that since path and 2-regular graphs have almost similar topologies, the two curve corresponding the graphs are close to each other unlike for the star graph whose curve is quite separate and has a different (deeper trough). On comparing the plots for $s = 2$ and $s = 3$, we observe drastic shift in the curves for the former than the latter in comparison with the plot for the normal graph Laplacian in Fig.(22b).

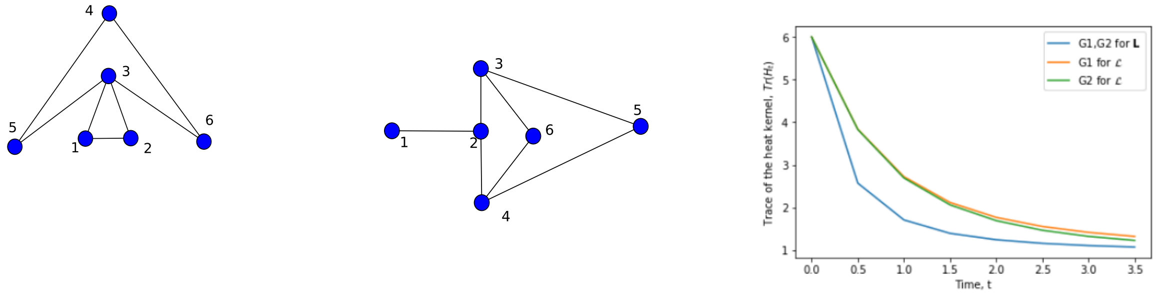
Since the trace function is computed from the eigenvalues of the Laplacian matrix $\mathbf{L}$. It implies that for isomorphic graphs, the curves of the trace function coincide following from the coincidence of the eigenvalues of the graphs as illustrated in Fig. 20.



(a) (b) (c)

Figure 20: (a) and (b) are two isomorphic graphs of size 8. (c) is the plot of the trace function of the normal Laplacian against time for both graphs. Only one curve is visible since both graphs have same values for the trace function due to the same eigenvalues for both of them.

On the otherhand, there exists graphs that are not isomorphic but have the same multi-set of eigenvalues. Such graphs are known as co-spectral graphs. They too show similar behaviour of the trace function as isomorphic graphs due to the similarity of eigenvalues. An example of co-spectral graphs is shown in Fig.21 along with the corresponding plot of the trace function of the heat kernel.



(a) (b) (c)

Figure 21: (a) and (b) are two co-spectral graphs with respect to **L**. (c) is the plot of the trace function of the normal Laplacian matrix against time for both graphs. It evident that the two graphs have similar multi-set of eigenvalues of the heat kernel matrix as only one curve is visible because of coincidence between the two curves for different plots.

At this point, it is worth noting that, the use of the trace formula for characterisation of graphs is limited. One limitation is based on analysis of co-spectral graphs, as they display similarity in structure following from the plots of the trace function and yet their structures are quite different. In addition, Xiao (Xiao et al., 2009) highlighted another limitation that is attributed to the fact that for each value of time, t, only a single scalar attribute is provided which implies that the heat kernel trace function must be either sampled with time or a fixed time value must be selected.

5.2 Generalised Heat kernel

In the previous section, we have discussed the heat kernel for diffusion process that occurs through direct interactions between nearest neighbours in the network. In this section however, we consider the fact that in many observed real-world process, it is observed that interactions do not only occur among nearest neighbours but also among non-nearest which we term as the long range interactions. In this work, we consider the method introduced by Estrada (Estrada, 2012) which accounts for long-range interactions using the concept of k-path Laplacian matrices. Consequently, The generalised heat kernel is the fundamental solution of the generalised diffusion equation 73 and it is given by

$$\mathbf{H}_{G_t} = e^{-t\mathbf{L}_G}, \tag{85}$$

where

$$\mathbf{L}_G = \Big(\sum_{k=1}^{\Delta} c_k \mathbf{L}_k\Big) = c_1\mathbf{L}_1 + c_2\mathbf{L}_2 + \cdots + c_\Delta \mathbf{L}_\Delta, \tag{86}$$

where $k \in \mathbb{N}$, $1 \leq \Delta \leq d_{max} \in \mathbb{N}$ and c_k are the coefficients. As discussed earlier, the coefficients c_k are chosen in a way that as distance k increases, the long range effect is weakened. Some of the common expressions for coefficients c_k are $c_1 = 1, c_{k\geq 2} = e^{-\lambda k}$, $c_{k\geq 2} = k^{-s}$ which depict physical influence while $c_k = kx^{k-1}$ for the social influence as discussed before in Subsection 4.17. On expansion of eqn. 86 can be written as

$$\mathbf{H}_{G_t} = e^{-t(c_1\mathbf{L}_1 + c_2\mathbf{L}_2 + \cdots + c_\Delta \mathbf{L}_\Delta)}, \tag{87}$$

where $\mathbf{L}_1, \mathbf{L}_2, \cdots, \mathbf{L}_\Delta$ are k-path Laplacian matrices for hops of length $k = 1, 2, \cdots, \Delta$ respectively.

At $k = 1$, we recover the normal heat kernel in eqn. 76.

On performing permutations of node labels of a graph, the spectrum of Laplacian matrix of the graph remains unchanged thus functions whose arguments are the spectrum of the Laplacian are considered invariants under node label permutations. In subsequent subsections, we explore some of the invariants of the generalised heat kernel which include trace, zeta function, derivative of the zeta function at the origin and heat content.

5.3 Trace of the Generalised Heat Kernel

The trace of the generalised heat kernel is therefore given by

$$Z_G(t) = Tr(\mathbf{H}_{G_t}) = \sum_{i=1}^{|V|} e^{-\mu_i t}, \tag{88}$$

where μ_i is the ith eigenvalue of the generalised Laplacian matrix. Alternatively, Eqn. 88 can be written as

$$Z_G(t) = 1 + e^{-\mu_2 t} + e^{-\mu_3} + \cdots + e^{-\mu_N t,} \tag{89}$$

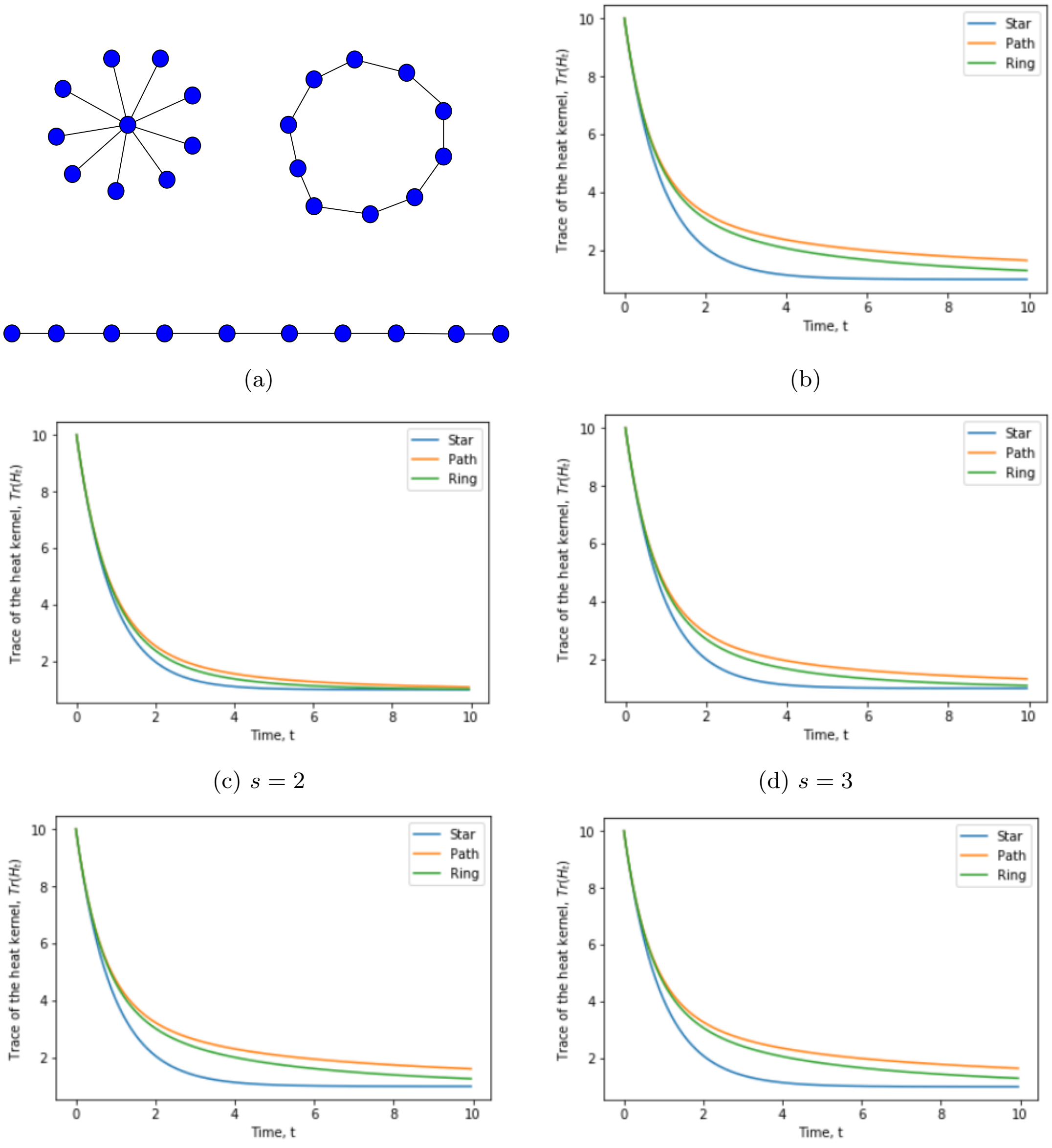
which takes on a similar format as eqn. 84 for the trace of the 'classical' Laplacian matrix.

We earlier on pointed out that the multiplicity of zero as an eigenvalue of $\mathbf{L}_G$ is equal to the number of connected components in a given graph, it therefore follows from eqn.89 that the trace of the generalised heat kernel can also be expressed as

$$Tr(\mathbf{H}_{G_t}) = C + \sum_{\mu_i \neq 0} e^{-\mu_i t}, \tag{90}$$

where C is the multiplicity of zero as an eigenvalue of $\mathbf{L}_G$ that is the number of connected components of a graph.

It is quite interesting to ascertain whether the trace of the generalised heat kernel can be used as a basis for analysing graphs as is the case with the trace of the 'classical' heat kernel discussed earlier. We consider an example of the three graphs namely the star, ring and path graphs of size 10 each in Fig. 22a. We use the Mellin and Laplace transforms for the Generalised Laplacian matrix.



(a)

(b)

(c) $s = 2$

(d) $s = 3$

(e) $\lambda = 2$

(f) $\lambda = 3$

Figure 22: (a) are the three graphs namely star, circular and path for which analysis is performed. (b) plot of the trace of the heat kernel (based on normal Laplacian) against time for star (blue), path (orange) and regular (green) graphs. (c) and (d) in the middle row correspond to plots of the trace function for the generalised heat kernel with Mellin transform for $s = 2$ and $s = 3$ respectively. The bottom row, that is, (e) and (f) are plots of trace function of the generalised heat kernel with Laplace transform for $\lambda = 2$ and $\lambda = 3$ respectively.

From Fig. 22, we observe distinct curves with distinct shapes for the 3 graphs for the heat kernel of both the normal and generalised Laplacian matrix. Focusing at the Mellin transformation is in the middle row, we observe that for $s = 2$, the curves get much closer to each other, nevertheless, we can still observe their distinctiveness, albeit their slopes are only slightly different from each other. On the other hand, the curves corresponding to the 3 graphs are much more distinct and also much closer in shape to those depicting trace plots of the normal Laplacian in Fig. 22b due to minimal influence of longrange interactions in the Laplace based case than the Mellin based scenario. From the plots, we can thus conclude that the trace function of the generalised heat kernel can as well be used as a tool for analysing graphs with different topologies.

Let us consider a simple toy example to illustrate the variation of the trace of the generalised heat kernel with time.

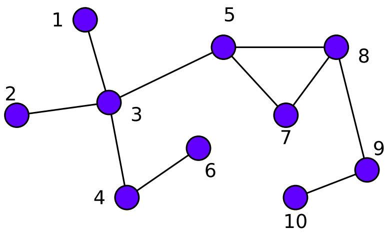
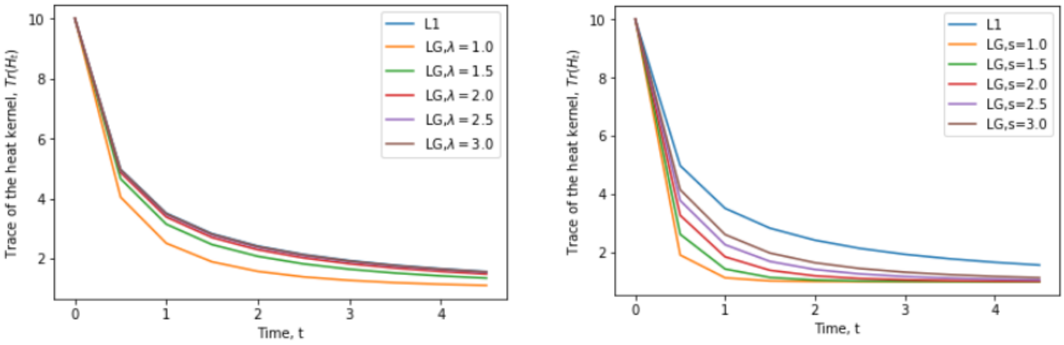


Figure 23: A simple network of size 10

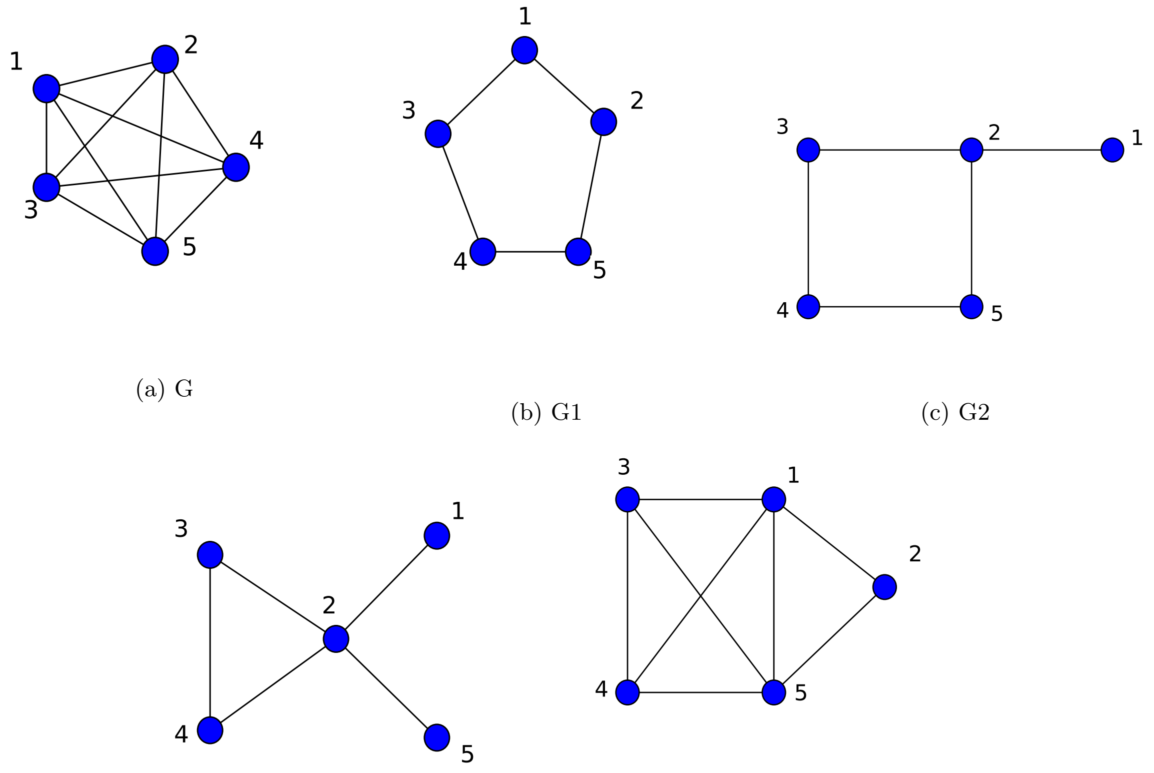


(a) (b)

Figure 24: Plots (performed using Eqn. 88) of the trace of the generalised heat kernel against time for the simple graph in Fig.23, for which the long-range influence is accounted for by the Laplace (left) and Mellin (right) transforms of the Laplacian matrix of the graph for different values of λ and s respectively.

Firstly, we observe from the plots in Fig.24 that the curve for the trace function against time for diffusion along edges (also known as direct interactions) of the graph (in blue) is the top most and it decreases gradually with time. When we consider long-range influence, we can see that as vary the values of the parameters λ and s for the Laplace and Mellin based transforms respectively, we deduce that as the parameter values are increases the corresponding curves approach that of the normal diffusion (in blue). However, we note that the approach to the normal curve occurs much faster in the Laplace transform than the Mellin counterpart. This is explained by the fact the longrange influence is more pronounced in the Mellin case than for the Laplace for the same value of respective parameters. For clarity, let us take $\lambda = s = 3.0$ (brown), we observe from Fig.24 that the curve corresponding to Laplace is already coinciding with that of the normal Laplacian while for the Mellin is a quite further from the normal Laplacian curve.

5.4 Comparison with Complete Graph based on Trace of the diffusion kernel



(a) G

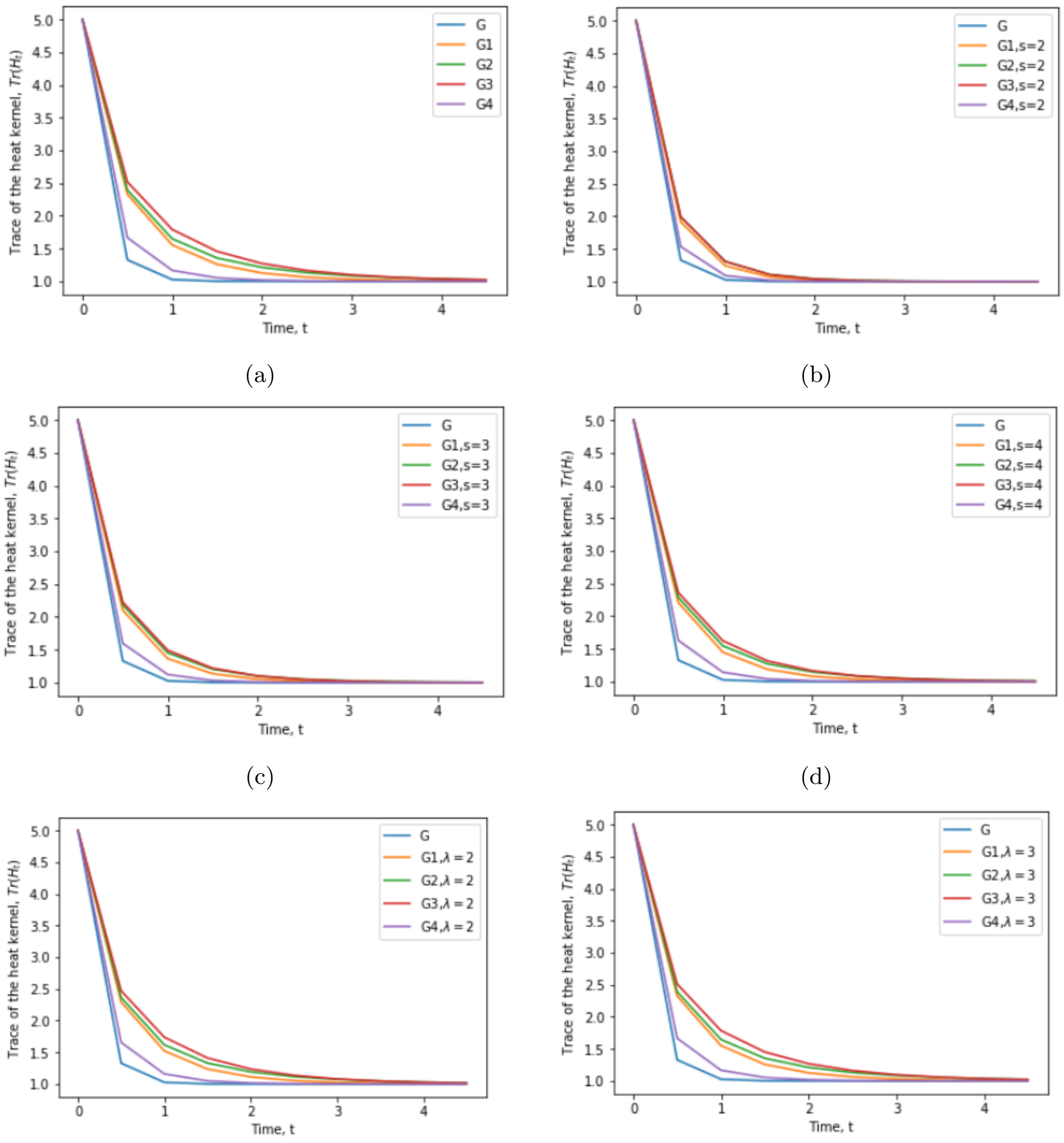
(b) G1

(c) G2

(d) G3

(e) G4

Figure 25: The five graphs of size 5. (c) is the complete graph,K_5 whose trace function is to be compared with that of the other 4 graphs.



(a)

(b)

(c)

(d)

(e)

(f)

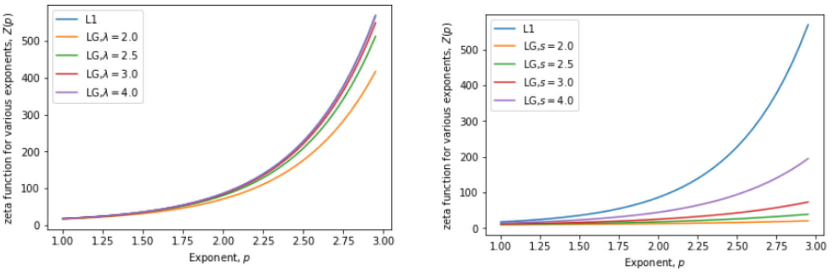
Figure 26: Plots of trace function of generalised heat kernel against time for graphs in Fig.25. From left to right and top to bottom, the plots are respectively for the normal Laplacian (a), Mellin based generalised Laplacian for $s = 2, 3$ and 4. To the right, is a plot of the trace function of the Mellin transformed Laplacian matrix at $s = 3$ against time for graphs G(blue), $G1$(orange),$G2$(green), $G3$(red), and $G4$(purple).

We observe from Fig.26a that based on the trace function curves, the process of diffusion over graph G_4 is closer to that occurring on the complete graph G. Graphs G_1, G_2 and G_3 follow by that order. Interestingly, on considering the generalised heat kernel of both the Mellin and Laplace based transforms, the order is maintained. However, we can see that for the former case, as s reduces, the curves corresponding to the four incomplete graphs get closer to that of the complete graph G which implies that as the long-range influence increases (which happens with decrease in s), the diffusion process tends closer to that on a complete graph. However, for the case where long-range influence is accounted for by the Laplace transformed Laplacian matrix, we can see that as the value of λ decreases (or increases), the diffusion process on the four graphs $G1$, $G2$, $G3$, and $G4$ gets slightly closer (or further) from that of the complete graph.

5.5 Zeta Function

In literature (Knill, 2013; Friedli et al., 2017), there exists various definition for the zeta function for finite simple graphs. Zeta functions play a vital role in definition of determinants of Laplacians and analytic torsion (Voros, 1987; Moscovici and Stanton, 1991) as well as applicability in various aspects in differential geometry and theoretical physics. However, in this work we consider the Zeta function (introduced by Carleman) associated with the eigenvalues of the generalised Laplacian matrix which is obtained by exponentiating and summing the reciprocal of the non-zero Laplacian eigenvalues (Friedli et al., 2017). It is thus defined by

$$\zeta_G(p) = \sum_{\mu_i \neq 0} \mu_i^{-p}. \tag{91}$$



(a) (b)

Figure 27: Illustration of the Zeta function of the graph in Fig. 23 against exponent δ. (a) corresponds to the Laplace transform of the graph Laplacian with $\lambda = 2, 2.5, 3$ and 4. (b) corresponds to the Mellin transform of the graph Laplacian with $s = 2, 2.5, 3$, and 4.

For the normal Laplacian, Mellin and Laplace transformed generalised Laplacian, we observe that the zeta function increases with increase in the exponent p. For different values of Laplace exponent λ, we can tell from the illustration in Fig.27a that the variation of the zeta function with p follows a similar trend as that in the normal Laplacian curve (in blue) due to a moderate long-range influence. On contrary, as the Mellin exponent, s changes, there is observable changes observed in the corresponding curves for the zeta function against p (see Fig. 27b). This can be attributed to the pronounced long-range interactions evident in the Mellin-based transformed Laplacian.

5.6 Zeta Function and Generalised Heat Kernel Trace Moments

In his work (Xiao et al., 2009), Xiao showed that the zeta function and the heat kernel trace are related in some way using the Mellin transform. In a similar way, we explore this relationship for the case of the generalised heat kernel and the zeta function of the eigenvalues of the generalised Laplacian matrix.

We consider a function $f(t) = e^{-\mu_i t}$, its Mellin transform is given by

$$\mu_i^{-s} = \frac{1}{\Gamma(p)} \int_0^\infty t^{p-1} e^{-\mu_i t} dt, \tag{92}$$

where μ_i is the i-th eigenvalue of $\mathbf{L}_G$ and $\Gamma(p)$ is the gamma function defined as

$$\Gamma(p) = \int_0^\infty t^{p-1} e^{-t} dt. \tag{93}$$

On summation for all non-zero eigenvalues of the Laplacian, Eqn.92 becomes

$$\zeta(p) = \sum_{\mu_i \neq 0} \mu_i^{-p} = \frac{1}{\Gamma(p)} \int_0^\infty t^{p-1} \sum_{\mu_i \neq 0} e^{-\mu_i t} dt \tag{94}$$

Using the connected component based formula for the trace of the heat kernel,that is, Eqn.90 in Eqn 94 gives

$$\zeta(p) = \frac{1}{\Gamma(p)} \int_0^\infty t^{p-1} \{Tr(\mathbf{H}_{G_t}) - C\} \, dt. \tag{95}$$

Thus the zeta function is related to the moments of the heat kernel trace. It is the moment generating function and thus a way of characterising the shape of the heat kernel trace.

5.7 Derivative of Zeta Function at the Origin

The derivative or slope of the zeta function at the origin is another characterisation of the heat kernel trace second to the zeta function which measures its shape. It is obtained as follows:

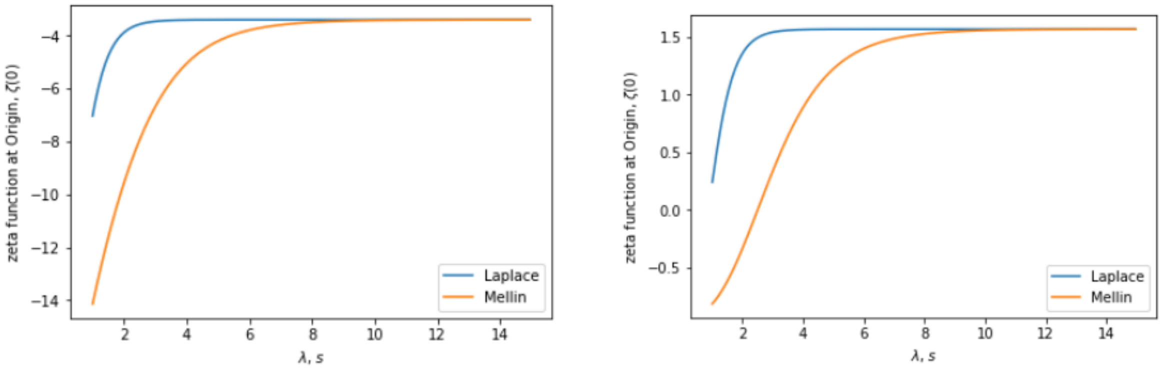
$$\zeta(p) = \sum_{\mu_i \neq 0} \mu_i^{-p} = \sum_{\mu_i \neq 0} e^{-p \ln \mu_i}, \tag{96}$$

where C is the number of connected components of the graph. Thus, the derivative is given by

$$\zeta'(p) = \sum_{\mu_i \neq 0} \{-\ln \mu_i\} e^{-p \ln \mu_i} \tag{97}$$

so, the derivative at the origin is

$$\zeta'(0) = -\sum_{\mu_i \neq 0} \ln \mu_i \tag{98}$$



(a)

(b)

Figure 28: Derivative of Zeta function against time for the graph in Fig.23. (a) corresponds to the plot for the normal Laplacian **L** matrix of the graph. (b) corresponds to the plot for which the eigenvalues of the normalised Laplacian $\mathcal{L}$ is used in Fig. 98.

To start with we compute the values of the zeta function for graph (Fig.23) which are 1.5686 and −3.4012 for the normalised and unnormalised Laplacian matrices respectively. As discussed earlier on, increase in values of s and λ results into lesser influence due to long-range interaction in the Mellin and Laplace transform cases respectively. We, however, observe that for the latter case, the value for the derivative of zeta function reaches faster (in both (28a) and (28b)) that of the normal diffusion with no long range interactions compared to the former case.

5.8 Heat Content of the Generalised Laplacian matrix

Heat content of a graph is intuitively the total amount of heat preserved in the graph. The heat content is defined as the sum of entries of the generalised heat kernel matrix of a graph. Its given by

$$Q(t) = \sum_{p \in V} \sum_{q \in V} \mathbf{H}_{G_t}(p, q) \tag{99}$$

The structural properties of a graph play a role in determining the quantity of heat preserved within a graph over time.

We note that from Eqn.87, when we consider a particular case of hops of length $1 (k = 1)$, we recover the heat content based on the normal Laplacian matrix which is extensively presented in (Xiao et al., 2009).

On substituting for $\mathbf{H}_{G_t}(p, q)$ in Eqn.99 gives

$$Q(t) = \sum_{p \in V} \sum_{q \in V} \sum_{k=1}^{|V|} e^{(-\mu_k t)} v_k(p) v_k(q), \tag{100}$$

which can be expanded into a polynomial in time as in (McDonald and Meyers, 2002)

$$Q(t) = \sum_{m=0}^{\infty} q_m t^m, \tag{101}$$

where q_m is given by

$$q_m = \sum_{k=1}^{|V|} \left\{ \left(\sum_{p \in V} v_k(p) \right)^2 \right\} \frac{(-\lambda_k)^m}{m} \tag{102}$$

The set of polynomial co-efficients, q_m is unique for a given graph and thus, can be used for graph characterisation. For purposes of graph clustering, we can construct feature vector using the k leading co-efficients, that is $B_k = (q_1, q_2, \cdots, q_k)^T$. We will further explore this idea in Section 6.

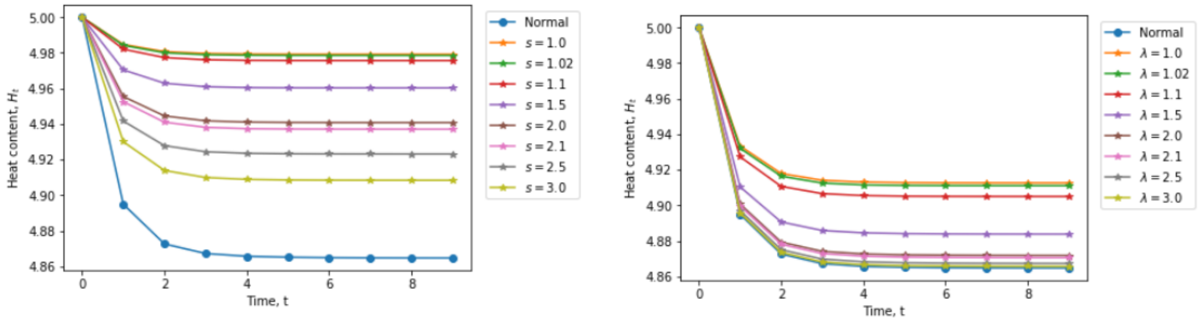
Alternatively, Eqn. 100 can be written as

$$Q(t) = \sum_{i=1}^{m} \alpha_i e^{-\lambda t}, \tag{103}$$

where $\alpha_i = \sum_{p \in V} \sum_{q \in V} v_k(p) v_k(q)$. We can see from Eqn 103 that the heat content can be treated as a summation of exponential functions with different decay rates determined by Laplacian eigenvalues and different weights (α_i) determined by the eigenvectors of the Laplacian.

5.9 Heat content Simulations

As pointed out before, the normalised Laplacian matrix performs better than the normal Laplacian in some scenarios. For computation of heat content, we opt for the normalised Laplacian since for the unnormalised case, the heat content remains constant over time.



(a)

(b)

Figure 29: Simulations for heat content against time for the graph in Fig.25c. (a) shows the simulations for different values of the parameter s of the Mellin transform based generalised normalised Laplacian matrix while (b) corresponds to that of the Laplace based generalised normalised Laplacian for different values of λ.

At $t = 0$, the heat kernel matrix $\mathbf{H}_{G_0} = \mathbf{I}$ thus the heat content is equal to the trace of $\mathbf{I}$ which is in turn equal to the number of vertices, $|V|$, in the graph, thus for graph in Fig.25c, $Q(0) = 5$.

Considering diffusion via interactions over the edges of the graph, we observe for the corresponding plot (in blue) that with time, the heat content decreases till when its constant (at approximately $t = 4$). On comparing generalised diffusion based on both Laplacian and Mellin based transforms, we can see that as the respective values of parameters λ and s increases, the faster the drop in heat content in both cases. It is however evident that the drop rate is higher in the former than the latter case.

6 Graph Clustering

Graph-based techniques are widely used in a number of applications such as computer vision, image processing and analysis, pattern recognition, object clustering among others. These techniques involve graph representation where nodes represent the objects or parts of objects, while the edges (or links) describe relations between the objects or parts of the objects. The idea behind graph-based techniques is to interpret the concept of interest as a graph theory concept for instance object similarity which is an essential aspect in computer vision and pattern recognition can be viewed as a graph similarity concept on using graph representation of the objects under study.

6.1 Image representation using Delaunay graphs

Different applications call for different graph representation of images or objects. In this work, we consider representation of objects by the Delaunay graph. The Delaunay graph is a graph obtained from Delaunay triangulation of the corner points of the objects as introduced in the previous chapters. The process of graph representation of Image using Delaunay triangulation follows the following steps:

i) First, we obtain feature or corner points which are the nodes of the graph. Here we use the Harris corner detection method discussed in Chapter 1.

ii) We then compute the Voronoi tessellations on the feature points (nodes). For each feature point, there is a corresponding region consisting of all points that are closer to that feature point than any other feature points. This results into a Voronoi diagram.

iii) We obtain the edges of the Delaunay graph by drawing an edge whenever two faces of the Voronoi diagram are separated from each other by an edge. This thus forms a graph known as the Delaunay graph.

6.2 Delaunay Graph Superimposition on Objects

Here, we use some objects of the well known Columbia Object Image Library (COIL-100) database. This database consists of color images (on dark background) of 100 objects. For each object, images were taken at every 5 degree turn up to 360 making a total of 72 images per object. We select 5 objects along with one of its object view. We then implement the Delaunay triangulation process in Matlab. This is followed by applying the code on to the selected images and the outcome are illustrated below.

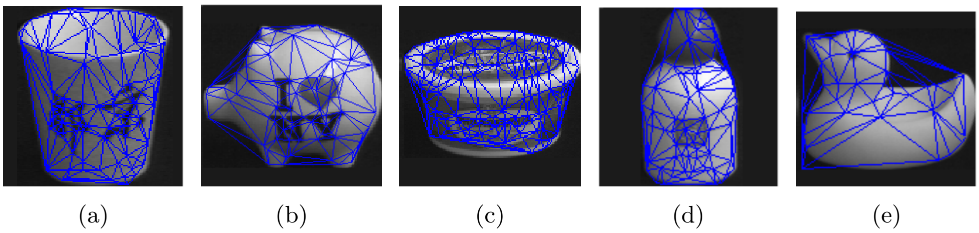
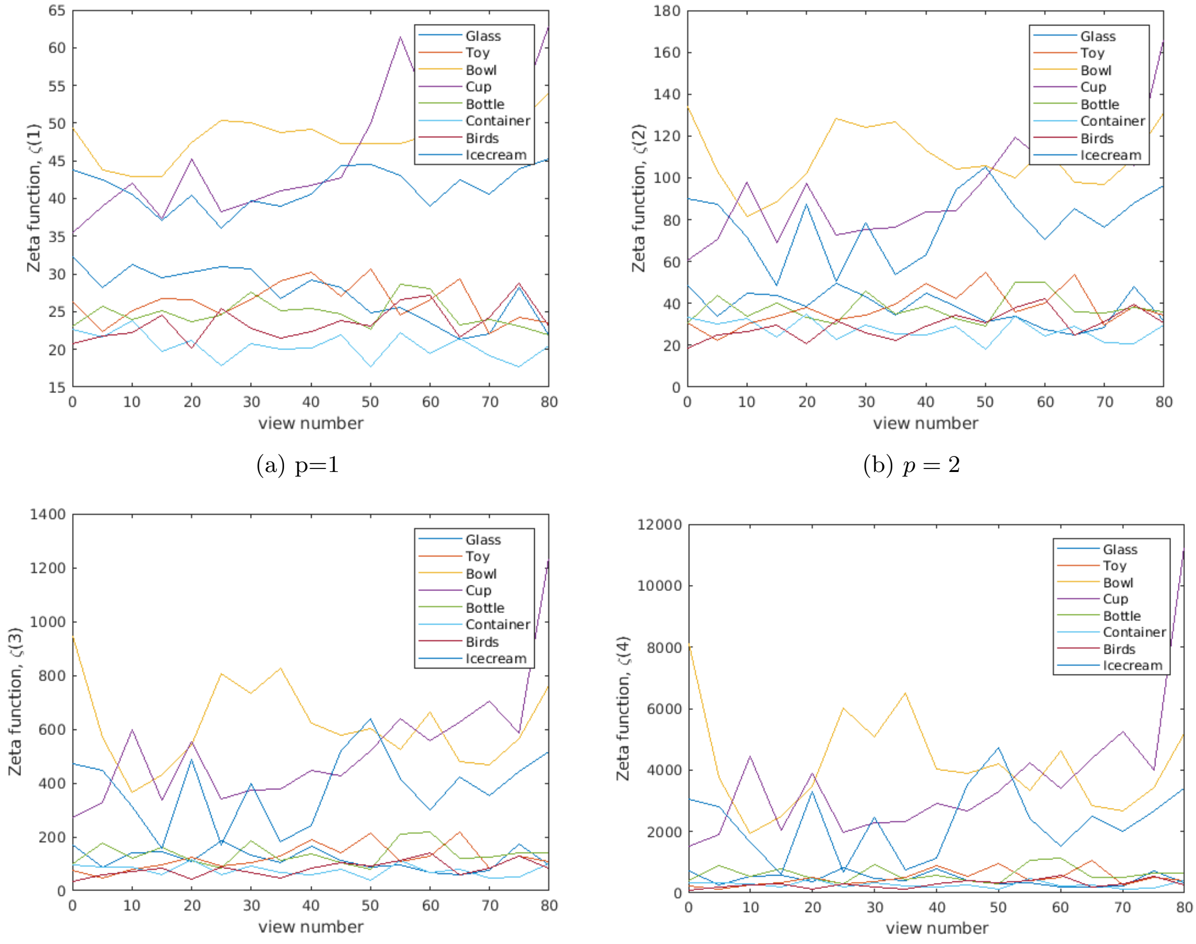

(a) (b) (c) (d) (e)

Figure 30: Illustration of selected objects from the COIL-100 database with their Delaunay graphs superimposed.

6.3 Zeta Function against View Number



(a) p=1

(b) $p = 2$

(c) $p = 3$

(d) $p = 4$

Figure 31

6.4 Principal Component Analysis (PCA) on objects

As mentioned earlier, PCA is a widely used statistical tool for dimension reduction. The PCA-based dimension reduction relies on selection of dimensions with the largest variance. Ding and He (?) showed that PCA dimension reduction indirectly performs clustering according to the K-means objective function by proving that the principal components are the continuous solution of the cluster membership indicators in the K-means clustering method. In this subsection, we discuss the steps followed in performing PCA-based dimension reduction on a given number m of objects.

i) We commence by selecting the objects from a database to which PCA is to be applied say m objects.

ii) Next, we extract graphs from each object using Delaunay triangulation with the Harris corner detection technique. This results into graphs $G_1, G_2, \cdots, G_m$.

iii) We construct the feature vector $\mathbf{B}_k$ for each graph G_k. For instance the vector can be obtained from the n leading co-efficients of the heat content polynomial that is $\mathbf{B}_k = (q_1, q_2, \cdots, q_n)^T$ or from the n leading Laplacian eigenvalues: $\mathbf{B}_k = (l_1, l_2, \cdots, l_n)^T$.

iv) We compute the matrix $\mathbf{S} = [\mathbf{B}_1 | \mathbf{B}_2 | \cdots | \mathbf{B}_m]$. The feature vectors form the columns of $\mathbf{S}$.

v) We then compute the matrix,$\hat{\mathbf{S}}$, of normalised data by subtracting the mean feature vector of the data set, $\bar{\mathbf{B}}$, from each of the column vectors as $\hat{\mathbf{S}} = [\mathbf{B}_1 - \bar{\mathbf{B}}|\mathbf{B}_2 - \bar{\mathbf{B}}|\cdots|\mathbf{B}_m - \bar{\mathbf{B}}]$.

vi) We compute the covariance matrix $\mathbf{C}$ by taking the matrix product $\mathbf{C} = \hat{\mathbf{S}}\hat{\mathbf{S}}^T$.

vii) We extract the principal components directions by performing eigendecomposition on the covariance matrix $\mathbf{C}$ that is

$$\mathbf{C} = \sum_{i=1}^{m} \lambda_i \mathbf{v}_i \mathbf{v}_i^T, \tag{104}$$

where λ_i are the eigenvalues and $\mathbf{v}_i$ are the eigenvectors. This is followed by selection of the first s leading eigenvectors (normally 3 for purposes of visualisation) to represent the graphs that we obtained from the images of the objects. By selecting the principal components, we reduce the dimension of the data. The coordinate system of the eigenspace is spanned by the s orthogonal vectors $\mathbf{V} = (\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_s)$.

viii) Finally, we project individual graphs (represented by $\mathbf{B}_k$ for $1 \leq k \leq m$) onto this eigenspace using $\mathbf{B}'_k = \mathbf{V}^T(\mathbf{B}_k - \bar{\mathbf{B}})$. Therefore, each graph G_k is represented by an s-component vector $\mathbf{B}'_k$ in the eigenspace.

6.5 Clustering using Spectrum of the Laplacian matrix

As discussed earlier on, one of the crucial steps involved in performing PCA is to create a feature vector of the images. One way is by using the leading eigenvalues of the Laplacian matrix. In this case, we take 6 of them. We then develop data whose columns correspond to the 6 eigenvalues labelled $l_1, l_2, \cdots, l_6$ while rows correspond to individual graphs of different images of the 8 objects. For visualisation purposes, we perform dimensionality reduction to only 3 dimensions as shown.

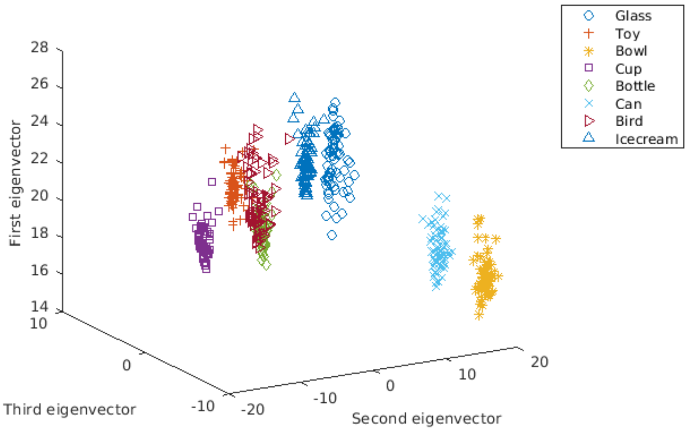
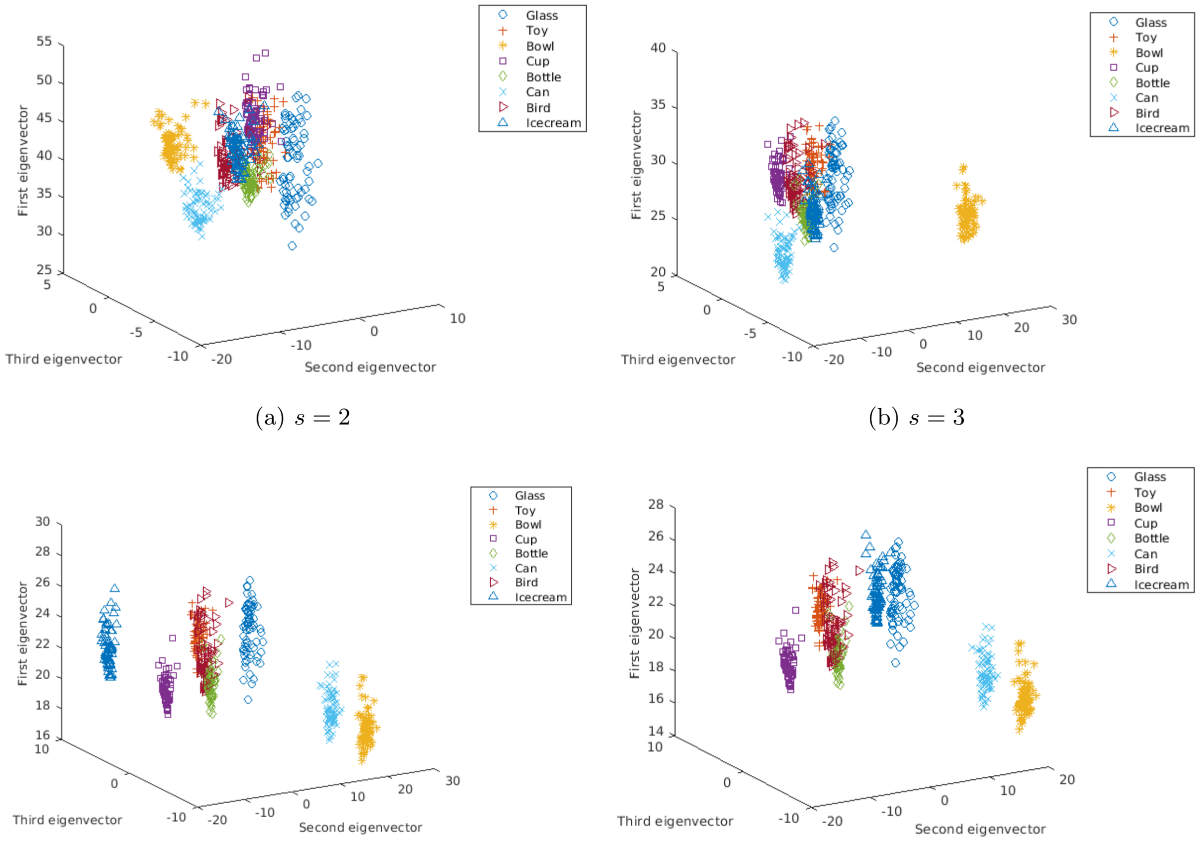


Figure 32: Clustering using PCA with feature vector composed of the 6 leading eigenvalues of the graph Laplacian matrix for images of objects. The 3D illustration consists of the 3 principal components as axes.

We dive deeper in performing clustering using PCA for which the feature vector consists of eigenvalues of the generalised Laplacian matrix whose long-range interactions are accounted for by the Mellin and the Laplace transforms of the k-Laplacian matrices.



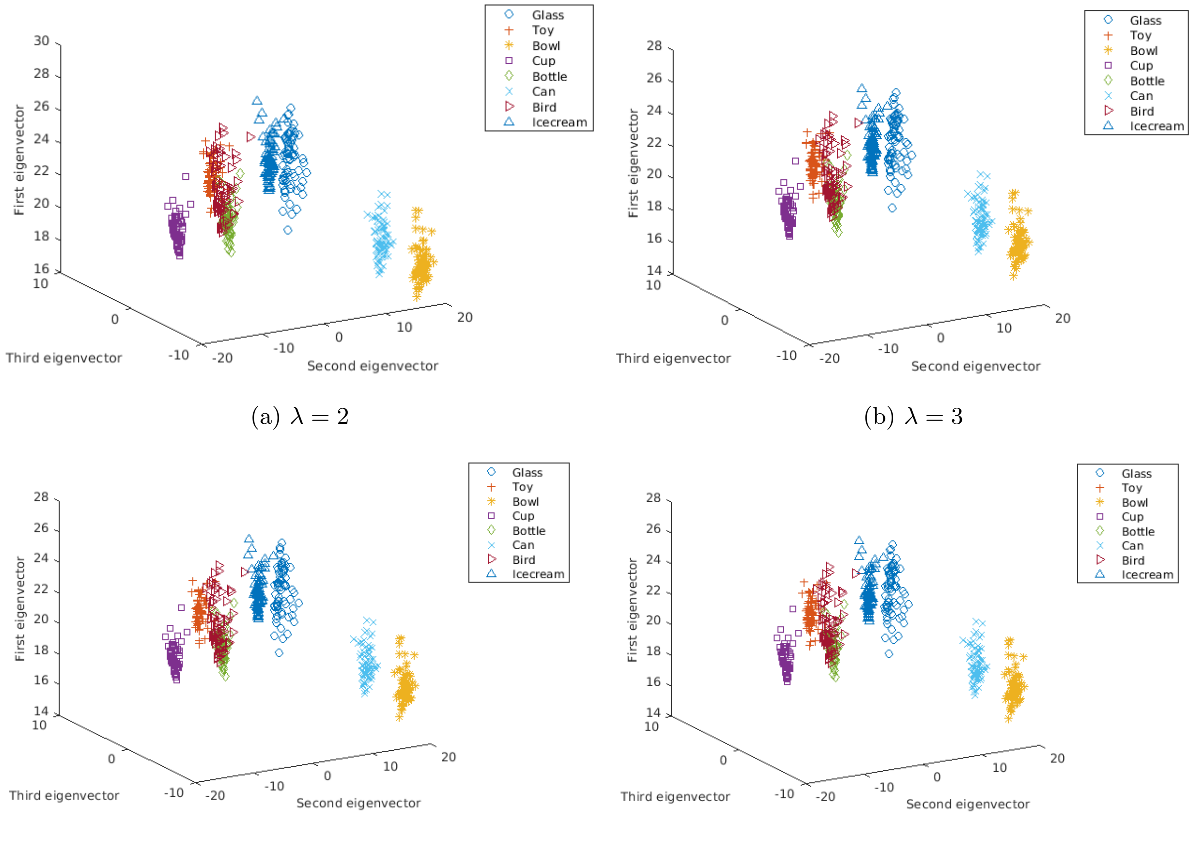
(a) $s = 2$

(b) $s = 3$

(c) $s = 5$

(d) $s = 6$

Figure 33: Illustration of PCA based clustering for 8 selected objects of the COIL-100 database. The feature vector consist of the largest 6 eigenvalues of the Laplacian matrix of the respective graphs. From left to right and top to bottom, we start off with the normal Laplacian followed by generalised Laplacian based on Mellin transform at $s = 2, 3, 5$, and 6.



(a) $\lambda = 2$

(b) $\lambda = 3$

(c) $\lambda = 4$

(d) $\lambda = 6$

Figure 34: Illustration of PCA based clustering for 8 selected objects of the COIL-100 database. The feature vector consist of the largest 6 eigenvalues of the Laplacian matrix of the respective graphs. From left to right and top to bottom, we start off with the normal Laplacian followed by generalised Laplacian based on Laplace transform at $\lambda = 2, 3, 4$, and 6.

6.6 Clustering using Zeta function

Here, we perform clustering by executing PCA based on the feature vector of the zeta function for different values of p for each graph representing images of selected objects in the COIL-100 database. For these simulation, we take p equal to $1, 2, 3$, and 4. First we consider the zeta function of the normal Laplacian followed by that of the Mellin and Laplace based generalised Laplacian matrices.

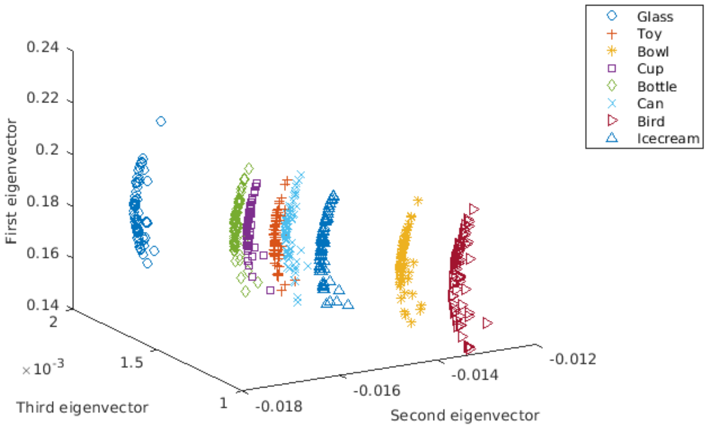
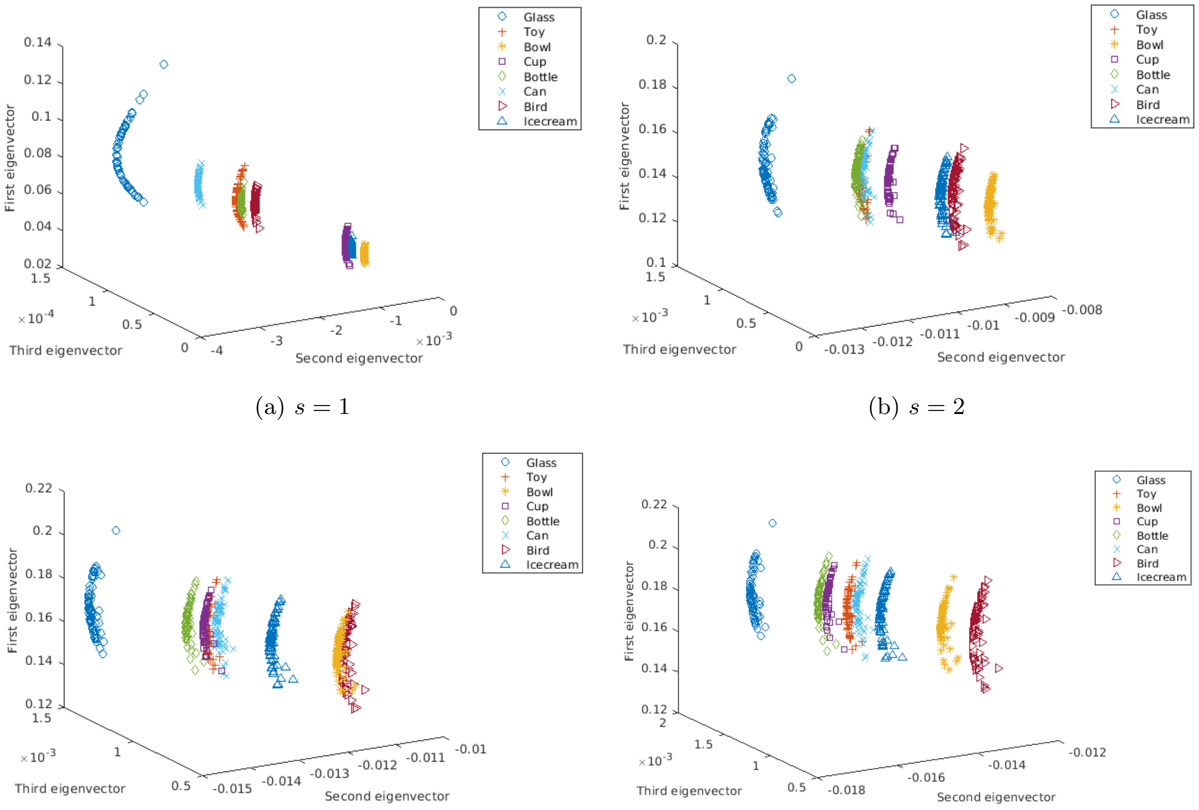


Figure 35



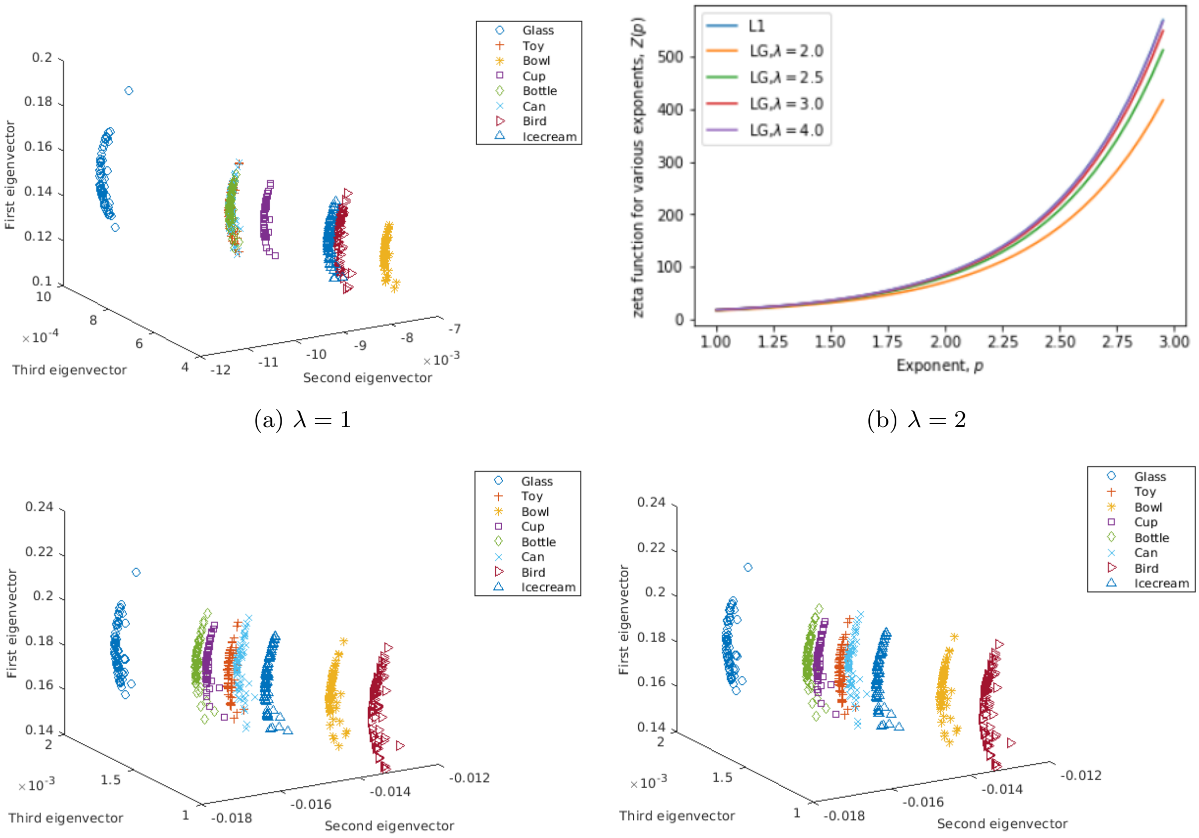
(a) $s = 1$

(b) $s = 2$

(c) $s = 4$

(d) $s = 6$

Figure 36



(a) $\lambda = 1$

(b) $\lambda = 2$

(c) $\lambda = 4$

(d) $\lambda = 6$

Figure 37

References

C. Adiga and M. Smitha. On the skew laplacian energy of a digraph. In *Int. Math. Forum*, volume 4, pages 1907–1914, 2009.

T. Alahakoon, R. Tripathi, N. Kourtellis, R. Simha, and A. Iamnitchi. K-path centrality: A new centrality measure in social networks. In *Proceedings of the 4th workshop on social network systems*, page 1. ACM, 2011.

R. Albert and A.-L. Barabási. Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1):47, 2002.

J. M. Anthonisse. The rush in a directed graph. *Stichting Mathematisch Centrum. Mathematische Besliskunde*, (BN 9/71):1–10, 1971.

H. Anton and C. Rorres. Elementary linear algebra, (2000). *Anton Textbook Inc, Ottawa*, 2007.

F. Aurenhammer and R. Klein. Voronoi diagrams. *Handbook of computational geometry*, 5:201–290, 2000.

F. Aurenhammer, R. Klein, and D.-T. Lee. *Voronoi diagrams and Delaunay triangulations.* World Scientific Publishing Company, 2013.

J. R. Banavar, A. Maritan, and A. Rinaldo. Size and form in efficient transportation networks. *Nature*, 399(6732):130, 1999.

A.-L. Barabási. *Network science.* Cambridge university press, 2016.

A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

J. S. Baras and P. Hovareshti. Efficient and robust communication topologies for distributed decision making in networked systems. In *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pages 3751–3756. IEEE, 2009.

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.

A. Barrat, M. Barthelemy, and A. Vespignani. *Dynamical processes on complex networks.* Cambridge university press, 2008.

E. Behrends. *Introduction to Markov chains*, volume 228. Springer, 2000.

M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.

N. Biggs. *Algebraic graph theory.* Cambridge university press, 1993.

S. B. Bozkurt, A. D. Güngör, I. Gutman, and A. S. Cevik. Randic matrix and randic energy. *MATCH Commun. Math. Comput. Chem*, 64:239–250, 2010.

A. E. Brouwer and W. H. Haemers. *Spectra of graphs.* Springer Science & Business Media, 2011.

A. C. Brown and T. R. Fraser. On the connection between chemical constitution and physiological action; with special reference to the physiological action of the salts of the ammonium bases derived from strychnia, brucia, thebaia, codeia, morphia, and nicotia. *Journal of anatomy and physiology*, 2 (2):224, 1868.

R. Byrne, J. Feddema, and C. Abdallah. Algebraic connectivity and graph robustness. *Sandia National Laboratories, Albuquerque, New Mexico*, 87185, 2005.

H. Cai, V. W. Zheng, and K. Chang. A comprehensive survey of graph embedding: problems, techniques and applications. *IEEE Transactions on Knowledge and Data Engineering*, 2018.

J. L. Casti. MS Windows NT complexity, September 26, 2017. URL https://www.britannica.com/science/complexity-scientific-theory.

M. Chandak, S. Bhalotia, and S. Agrawal. A novel approach to compute steiner point in graph: Application for network design. 2017.

B. V. Cherkassky, A. V. Goldberg, and T. Radzik. Shortest paths algorithms: Theory and experimental evaluation. *Mathematical programming*, 73(2):129–174, 1996.

F. Chung. The heat kernel as the pagerank of a graph. *Proceedings of the National Academy of Sciences*, 104(50):19735–19740, 2007.

F. Chung. A local graph partitioning algorithm using heat kernel pagerank. *Internet Mathematics*, 6(3): 315–330, 2009.

F. R. Chung. *Spectral graph theory*. Number 92. American Mathematical Soc., 1997.

D. Cvetkovic and P. Rowlinson. Spectral graph theory. *Topics in algebraic graph theory*, pages 88–112, 2004.

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

P. De Meo, E. Ferrara, G. Fiumara, and A. Ricciardello. A novel measure of edge centrality in social networks. *Knowledge-based systems*, 30:136–150, 2012.

E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1): 269–271, 1959.

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

W. Ellens and R. E. Kooij. Graph measures and network robustness. *arXiv preprint arXiv:1311.5064*, 2013.

P. Erdös and A. Rényi. On random graphs, i. *Publicationes Mathematicae (Debrecen)*, 6:290–297, 1959.

P. Erdos and A. Rényi. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*, 5(1): 17–60, 1960.

E. Estrada. Quantifying network heterogeneity. *Physical Review E*, 82(6):066102, 2010.

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 N. Hatano. Communicability in complex networks. *Physical Review E*, 77(3):036111, 2008.

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.

E. Estrada, L. V. Gambuzza, and M. Frasca. Long-range interactions and network synchronization. *arXiv preprint arXiv:1704.01349*, 2017a.

E. Estrada, E. Hameed, N. Hatano, and M. Langer. Path laplacian operators and superdiffusive processes on graphs. i. one-dimensional case. *Linear Algebra and its Applications*, 523:307–334, 2017b.

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

L. Euler. The solution of a problem relating to the geometry of position. 1976.

M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the internet topology. In *ACM SIGCOMM computer communication review*, volume 29, pages 251–262. ACM, 1999.

M. L. Fredman and R. E. Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of the ACM (JACM)*, 34(3):596–615, 1987.

L. C. Freeman. Centrality in social networks conceptual clarification. *Social networks*, 1(3):215–239, 1978.

F. Friedli, A. Karlsson, et al. Spectral zeta functions of graphs and the riemann zeta function in the critical strip. *Tohoku Mathematical Journal*, 69(4):585–610, 2017.

X. Gao, B. Xiao, D. Tao, and X. Li. A survey of graph edit distance. *Pattern Analysis and applications*, 13(1):113–129, 2010.

C. Godsil and G. Royle. Algebraic graph theory springer. *New York*, 2001.

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

L. Grady. Random walks for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 28(11):1768–1783, 2006.

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

J. H. Grisi-Filho, R. Ossada, F. Ferreira, and M. Amaku. Scale-free networks with the same degree distribution: Different structural properties. *Physics Research International*, 2013, 2013.

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

C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, pages 10–5244. Citeseer, 1988.

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

B. A. Huberman. The laws of the web, 2001.

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].

M. O. Jackson. *Social and economic networks*. Princeton university press, 2010.

A. Jamakovic and P. Van Mieghem. On the robustness of complex networks by using the algebraic connectivity. *NETWORKING 2008 Ad Hoc and Sensor Networks, Wireless Networks, Next Generation Internet*, pages 183–194, 2008.

G. Jeh and J. Widom. Simrank: a measure of structural-context similarity. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 538–543. ACM, 2002.

M. Karoński. A review of random graphs. *Journal of Graph Theory*, 6(4):349–389, 1982.

R. Kasprzak. Diffusion in networks. *Journal of Telecommunications and Information Technology*, pages 99–106, 2012.

P. Kissani and Y. Mizoguchi. Laplacian energy of directed graphs and minimizing maximum outdegree algorithms. *Kyushu University Institutional Repository*, 2010.

K. Kloster and D. F. Gleich. Heat kernel based community detection. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1386–1395. ACM, 2014.

O. Knill. The zeta function for circular graphs. *arXiv preprint arXiv:1312.4239*, 2013.

R. Kondor and J.-P. Vert. Diffusion kernels. *kernel methods in computational biology*, pages 171–192, 2004.

J. Lafferty and G. Lebanon. Diffusion kernels on statistical manifolds. *Journal of Machine Learning Research*, 6(Jan):129–163, 2005.

G. Levi. A note on the derivation of maximal common subgraphs of two directed or undirected graphs. *Calcolo*, 9(4):341, 1973.

D. López-Pintado. Diffusion in complex social networks. *Games and Economic Behavior*, 62(2):573–590, 2008.

H. Ma, H. Yang, M. R. Lyu, and I. King. Mining social networks using heat diffusion processes for marketing candidates selection. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 233–242. ACM, 2008.

J. Magouirk, S. Atran, and M. Sageman. Connecting terrorist networks. *Studies in Conflict & Terrorism*, 31(1):1–16, 2008.

P. McDonald and R. Meyers. Diffusions on graphs, poisson problems and spectral geometry. *Transactions of the American Mathematical Society*, 354(12):5111–5136, 2002.

S. Melnik, H. Garcia-Molina, and E. Rahm. Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In *Data Engineering, 2002. Proceedings. 18th International Conference on*, pages 117–128. IEEE, 2002.

Z. Mihalić and N. Trinajstić. A graph-theoretical approach to structure-property relationships, 1992.

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

R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network motifs: simple building blocks of complex networks. *Science*, 298(5594):824–827, 2002.

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

H. P. Moravec. Visual mapping by a robot rover. In *Proceedings of the 6th international joint conference on Artificial intelligence-Volume 1*, pages 598–600. Morgan Kaufmann Publishers Inc., 1979.

H. P. Moravec. Obstacle avoidance and navigation in the real world by a seeing robot rover. Technical report, STANFORD UNIV CA DEPT OF COMPUTER SCIENCE, 1980.

H. Moscovici and R. J. Stanton. R-torsion and zeta functions for locally symmetric manifolds. *Inventiones mathematicae*, 105(1):185–216, 1991.

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.

M. E. Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003.

M. E. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004.

M. E. Newman, D. J. Watts, and S. H. Strogatz. Random graph models of social networks. *Proceedings of the National Academy of Sciences*, 99(suppl 1):2566–2572, 2002.

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

M. Nikolić. Measuring similarity of graph nodes by neighbor matching. *Intelligent Data Analysis*, 16(6): 865–878, 2012.

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.

R. Ortiz Gaona, M. Postigo Boix, and J. L. Melus Moreno. Centrality metrics and line-graph to measure the importance of links in online social networks. *International Journal of New Technology and Research*, 2(12):20–26, 2016.

G. A. Pagani and M. Aiello. The power grid as a complex network: a survey. *Physica A: Statistical Mechanics and its Applications*, 392(11):2688–2700, 2013.

S. Pettie. A faster all-pairs shortest path algorithm for real-weighted sparse graphs. In *International Colloquium on Automata, Languages, and Programming*, pages 85–97. Springer, 2002.

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.

R. O. Saber and R. M. Murray. Agreement problems in networks with directed graphs and switching topology. In *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, volume 4, pages 4126–4132. IEEE, 2003.

B. Schwikowski, P. Uetz, and S. Fields. A network of protein–protein interactions in yeast. *Nature biotechnology*, 18(12):1257, 2000.

R. Seidel. On the all-pairs-shortest-path problem in unweighted undirected graphs. *Journal of computer and system sciences*, 51(3):400–403, 1995.

J. Shlens. A tutorial on principal component analysis. *arXiv preprint arXiv:1404.1100*, 2014.

A. Smith, J. Grierson, D. Wain, M. Pitts, and P. Pattison. Associations between the sexual behaviour of men who have sex with men and the structure and composition of their social networks. *Sexually transmitted infections*, 80(6):455–458, 2004.

L. I. Smith. A tutorial on principal components analysis. Technical report, 2002.

O. Sporns, D. R. Chialvo, M. Kaiser, and C. C. Hilgetag. Organization, development and function of complex brain networks. *Trends in cognitive sciences*, 8(9):418–425, 2004.

A. R. Stoica. Delaunay diagram representations for use in image near-duplicate detection. *Senior project submittd tot he division of science, mathematics and computing of Bard College. New York*, 2011.

A. Sydney, C. Scoglio, P. Schumm, and R. E. Kooij. Elasticity: topological characterization of robustness in complex networks. In *Proceedings of the 3rd International Conference on Bio-Inspired Models of Network, Information and Computing Sytems*, page 19. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008.

D. Thanou, X. Dong, D. Kressner, and P. Frossard. Learning heat diffusion graphs. *IEEE Transactions on Signal and Information Processing over Networks*, 3(3):484–499, 2017.

M. Thorup. Undirected single-source shortest paths with positive integer weights in linear time. *Journal of the ACM (JACM)*, 46(3):362–394, 1999.

M. Trajković and M. Hedley. Fast corner detection. *Image and vision computing*, 16(2):75–87, 1998.

A. Tsiatas. *Diffusion and clustering on large graphs.* University of California, San Diego, 2012.

G. Turán. On the succinct representation of graphs. *Discrete Applied Mathematics*, 8(3):289–294, 1984.

J. R. Ullmann. An algorithm for subgraph isomorphism. *Journal of the ACM (JACM)*, 23(1):31–42, 1976.

A. Voros. Spectral functions, special functions and the selberg zeta function. *Communications in Mathematical Physics*, 110(3):439–465, 1987.

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.

L. Weinberg. A simple and efficient algorithm for determining isomorphism of planar triply connected graphs. *IEEE Transactions on Circuit Theory*, 13(2):142–148, 1966.

B. Wellman and S. D. Berkowitz. *Social structures: A network approach*, volume 2. CUP Archive, 1988.

T. Wey, D. T. Blumstein, W. Shen, and F. Jordan. Social network analysis of animal behaviour: a promising tool for the study of sociality. *Animal behaviour*, 75(2):333–344, 2008.

J. A. Williams, S. M. Dawson, and E. Slooten. The abundance and distribution of bottlenosed dolphins (tursiops truncatus) in doubtful sound, new zealand. *Canadian journal of zoology*, 71(10):2080–2088, 1993.

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

J. Wu, Y. Tan, H. Deng, Y. Li, B. Liu, and X. Lv. Spectral measure of robustness in complex networks, 2008. *arXiv preprint arXiv:0802.2564.*

B. Xiao, E. R. Hancock, and R. C. Wilson. Graph characteristics from the heat kernel trace. *Pattern Recognition*, 42(11):2589–2606, 2009.

C. Yang, J. Mao, and P. Wei. Air traffic network optimization via laplacian energy maximization. *Aerospace Science and Technology*, 49:26–33, 2016.

W. W. Zachary. An information flow model for conflict and fission in small groups. *Journal of anthropological research*, 33(4):452–473, 1977.

L. A. Zager and G. C. Verghese. Graph similarity scoring and matching. *Applied mathematics letters*, 21(1):86–94, 2008.