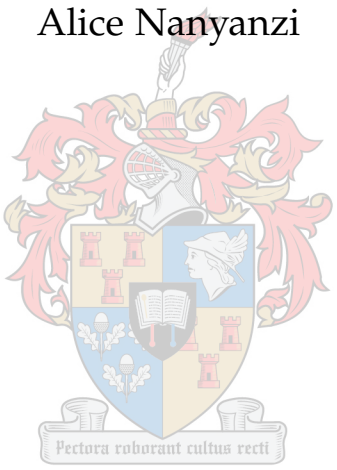
Diffusion on networks

by

Alice Nanyanzi



Thesis presented in partial fulfilment of the requirements for the degree of Master of Science in Mathematics in the Faculty of Science at Stellenbosch University

Supervisors:

Prof. Franck Kalala Mutombo
Dr. Simukai Utete

December 2018

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: 2018/06/30 .....

Abstract

Diffusion on networks

Alice Nanyanzi

Department of Mathematical Sciences,
University of Stellenbosch,
Private Bag X1, Matieland 7602, South Africa.

Thesis: MSc

December 2018

Many real-world processes or simplified models of these processes can be represented as dynamical systems on networks. Diffusion models on networks is a particular case of dynamical systems on networks and often used to develop simple models of many dynamical processes in real-world such as spread of epidemic in the population, dissemination of information or gossip among people, propagation of computer viruses in a computer network, and many others.
In the study of diffusion on networks, various models have been put forward in which the spread of the behaviour of interest is considered to occur only along the edges of the network. It was however observed that apart from the interactions among nearest nodes (short-range interactions), there exists interactions between pairs of non-nearest neighbours. These interactions are referred to as long-range interactions and they contribute significantly to the diffusion on networks. We will therefore explore the diffusion on networks due to both short and long-range interactions. We refer to this model of diffusion as the generalised diffusion model. The long-range interactions are accounted for using the Mellin and Laplace based transforms of the k-path Laplacian matrices.

Moreover, the heat kernel is a very important concept in diffusion as it can be viewed as describing the flow of information across the edges of a graph with time. The heat kernel is related to the Laplacian of the graph. One of the recent developments has been the utility of the heat kernel to graph characterisation for purposes of clustering by extracting useful and stable heat kernel invariants such as the trace of the heat kernel, zeta function, derivative of the zeta function, and heat content invariants from which pattern vectors are constructed. The pattern vectors are then subjected to Principal components analysis resulting into clustering.

Our contribution is to extend the concept of heat kernel by considering the generalised diffusion model which we refer to as the generalised heat kernel. This heat kernel can be viewed as capturing the flow of information not only along the edges but also across hops between nodes separated at shortest distance $d > 1$. We further investigate the use of the generalised heat kernel in clustering graphs following the above procedure and ascertain the impact of long-range interactions based on the resulting clusters. In our experiments, we use the well known COIL database.

Uittreksel

Verspreiding op netwerke

Alice Nanyanzi

Departement Wiskundige Wetenskappe,
Universiteit van Stellenbosch,
Privaatsak X1, Matieland 7602, Suid Afrika.

Tesis: MSc

Desember 2018

Acknowledgements

Thanks to you

Dedications

To all

Contents

List of Figures

Chapter 1

Introduction

Diffusion on networks is one of the most common methods of developing simple models of process in real world for example spread of epidemics, dissemination of information with in social network, among others. To start with, researchers have developed such models in which diffusion is considered to occur through interactions along edges of a graph. In otherwords, there is no interaction between non-nearest neighbours. However, in many real-world situations, it was discovered that there exists some kind of interactions between nodes that are not directly connected to each other. This kind of interactions, known as long-range interactions, were not accounted for in the previously discussed case though they contribute significantly to the diffusion process. There are various methods that have been put forward to account for the long-range range interactions.

In this study, however, we will focus on the method introduced in [40, 41] which is based on Laplace and Mellin transforms of the k-path Laplacian matrices. The summation of the k-path Laplacian matrices for k in the interval $[1, d_{max}]$, (where d_{max} denotes the diameter of the graph) results into a matrix we term as the generalised Laplacian matrix. Through simulations, we will explore the impact of long-range effects on networks of different structures that is the Random graphs particularly Erdos Renyi networks and scale free networks specifically Barabasi Albert networks.

The heat kernel is a very important aspect of diffusion. It is the fundamental solution of the heat equation and it is obtained by exponentiating the eigensystem of the Laplacian matrix over time. Most importantly, the heat kernel describes the flow of information across edges with in a graph. Many researchers have devoted to the study of the heat kernel and a lot of literature has been published explaining various applications of the heat kernel for example as a pagerank of a graph specifically in determination of the ranking of Web pages by Web search engines [26] and local graph partitioning [27], as a community detector in networks [69], as a means of graph embedding [119], among others.
In computer vision and pattern recognition, a number of problems such as graph clustering, pattern matching,image segmentation, routing, sequencing of relational data, among others

have been solved using spectral graph theory. Spectral graph theory is a branch of mathematics that is concerned with characterisation of the structural properties of graphs using the characteristic polynomial, eigenvalues and eigenvectors of the adjacency matrix or Laplacian matrix of a graph. In this study, we focus on graph clustering which is the most important challenge encountered in computer vision and pattern recognition. We approach this problem by use of the heat kernel. According to Xiao [119] different invariants of the heat kernel for example the trace, zeta function, derivative of zeta function, and heat content can be used in structural characterisation of graphs for purposes of clustering. this is achieved by extracting vectors from the invariants and the performing Principal Components Analysis on the vectors to map the corresponding graphs onto low-dimensional spaces.

Our contribution in this area will involve extending this concept in a way that we define the heat kernel based on the generalised Laplacian matrix. We choose to call it the generalised heat kernel. We will then explore the invariants of the new heat kernel and ascertain its utility in graph characterisation.

The thesis is structured as follows: Chapter 2 reviews the literature related to this thesis. This includes an introduction to networks and graphs, terminology and definitions in graph theory, matrix representation of graphs, random models of networks, characterisation and structure of networks. It also consists of background on graph similarity and embedding, Voronoi diagrams and triangulations, corner detection on images and finally Principal Components Analysis.

In chapter 3, we explore diffusion on networks where we illustrate diffusion of heat on network using a model where flow of heat occurs along the edges of a network. We also discuss the equilibrium behaviour. We further ascertain the impact of network structure, choice of initial nodes and network homogeneity on the dynamics of diffusion on a network. We also introduce the idea of diffusion on directed networks in comparison to undirected networks. Moreover, we discuss an extended model where diffusion is not only due to interaction between neighbouring nodes but also through long-range interactions. We refer to this model as the generalised diffusion model. We illustrate this model on networks of different structures that is networks with power-law degree distribution (Barabási-Albert networks) and those with Poisson degree distribution (Erdős-Rényi networks). We further illustrate the impact of the strength of long-rage interactions to the rate of diffusion on both networks.

In chapter 4, we study the heat kernel and its utility in graph characterisation. We then extend the concept of heat kernel by considering the generalised diffusion model. In other words, the heat kernel which we refer to as the generalised heat kernel is the fundamental solution of the generalised diffusion equation. We discuss how stable invariants such as the trace, zeta function, derivative of the zeta function and heat content invariants can be extracted from the generalised heat kernel and how these invariants can be used for purposes of graph cluster-

ing. Experiments are performed on the COIL database.

Finally, we draw conclusions and discuss areas of further research in chapter 5.

Chapter 2

Literature Review

2.1 Graphs and Networks. An introduction

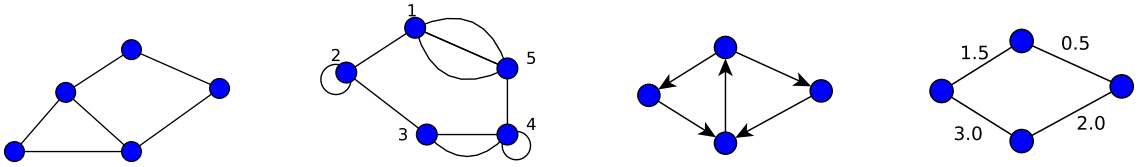
According to [39], in mathematics the study of networks is known as graph theory. In this thesis, we will use the two words: 'graph' and 'network' interchangeably. To begin with, let us discuss some of the terminology that we will use often in this work.

2.2 Graph/Network terminology

Definition 2.2.1 (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 connecting vertices, $E \subseteq \{(u, v)|u, v \in V\}$. A graph may be undirected where edges have no directions or it may be directed (known as digraph) in which each edge has a direction, pointing from one vertex to another. Such edges are called directed edges and are represented by drawing a line with an arrow at one end. The order (or size) of a graph G, denoted as $|G|$, is the number of vertices of a graph. On the other hand, the number of edges of a graph is denoted by $\|G\|$. The size of a graph determines whether it is finite or infinite. A graph with multiple edges is a multi-graph. On the otherhand, a simple graph is a graph with neither multiple edges or self loops. Moreover, 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 transportation costs, distance covered, frequency of information flow and many others. [83].

Remark 2.2.2. In this study, we will be working with simple connected undirected graphs or networks unless stated otherwise.

Definition 2.2.3 (Subgraph)**.** Given a graph $G = (V, E)$, we say $G' = (V', E')$ is a subgraph of G if and only if $V' \subseteq V$ and $E' \subseteq E$. G is called the supergraph [44].



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

Figure 2.1: Classifications of graphs: (a) A simple graph. (b) A graph with multiple edges and loops. (c) A directed graph. (d) A weighted graph.

Definition 2.2.4 (Incidence)**.** Given a graph $G = (V, E)$. 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 $v \in V$ [83].

Definition 2.2.5 (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 $\boldsymbol{A}$ [83].

Definition 2.2.6 (Neighborhood ($N_G(v)$))**.** The neighborhood of a vertex $v \in V$ is a set of all vertices that are adjacent to v [83]. Mathematically, $N_G(v) = \{u \in V | uv \in E\}$

Definition 2.2.7 (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 [39]. The total degree k_v is $k_v = k_v^{in} + k_v^{out}$.

Definition 2.2.8 (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 [44]. 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 $\boldsymbol{A}^k$ where $\boldsymbol{A}$ is the Adjacency matrix of a graph which we discuss in subsequent subsections.

Definition 2.2.9 (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 [39]. 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 2.2.10 (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 [40].

Definition 2.2.11 (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 2.2.12 (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 [83]. A connected component is also referred to as a maximal connected subgraph of a graph.

Some special categories of graphs include:

Definition 2.2.13 (Star graph)**.** A star graph, denoted as S_n, is a complete bipartite network $K_{1,n}$ [117].

Definition 2.2.14 (Complete graph)**.** A graph $G = K_{V(G)}$ is a complete graph on $V(G)$, if every two nodes are adjacent: $E = E(G)$. We denote a complete network of order n by K_n.

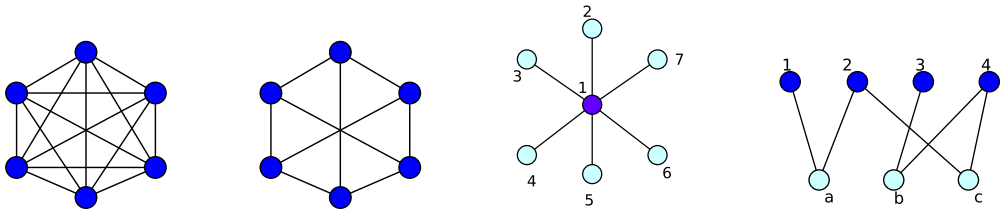
Definition 2.2.15 (Regular graph)**.** A graph network is a graph G in which every node has the same degree. A k-regular graph is one in which every node has degree equal to k.

Definition 2.2.16 (Cycle)**.** A cycle graph is a connected graph 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 [117].

Definition 2.2.17 (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 graph in which each node of V_1 is connected to each node of V_2 is known as a complete

bipartite graph; if $|V_1| = m$ and $|V_2| = n$, such a graph is denoted by $K_{m,n}$ [83].

Definition 2.2.18 (Tree)**.** A tree is a connected undirected graph that contains no closed loops [83]. It is important to note that a tree with n vertices has $(n-1)$ edges. 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.



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

Figure 2.2: (a) A complete graph with 6 nodes. (b) A 3-regular graph. (c) A star graph, S_6. (d) A bipartite graph with $m = 4$ and $n = 3$.

Definition 2.2.19 (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 [39].

However, Definition 2.2.19 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 [55] suggested definitions for a simple network as well as a more general definition of networks. First, let us understand the term 'relation'.

Definition 2.2.20 (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 $[v_i \neq v_j]$ [39].

Definition 2.2.21 (Network: More general definition)**.** A network is a triple $G = (V, E, f)$, where V is a finite set of nodes, $E \subseteq V \otimes V = \{e_1, e_2, \cdots, e_m\}$ is a set of links, and f is a mapping which associates some elements of E to a pair of elements of V, such as that if $v_i \in V$ and $v_j \in V$, then $f : e_1 \to [v_i, v_j]$ and $f : e_2 \to [v_j, v_i]$. A weighted network is created by

replacing the set of links E by the set of link weights $W = \{w_1, w_2, \cdots, w_m\}$, such that $w_i \in \mathcal{R}$. Then, a weighted network is defined by $G = (V, W, f)$ [39].

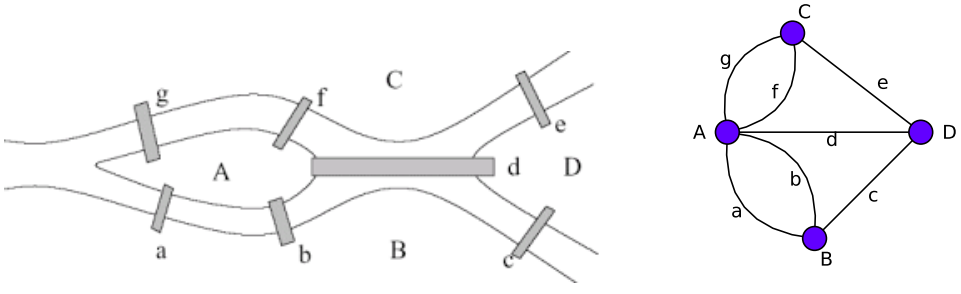
2.3 Complex systems and complex networks

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 on January 23, 2000, Stephen Hawking referred to the 21st century as a century of complexity [59].

Complex systems are composed of interconnected components, however, it has been observed that many complex systems display a behaviour phenomenon (also known as emergent behaviour) that cannot be explained by any conventional analysis of the system's constituent parts [23]. 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 Königsberg bridge problem which is described in [46, 47]. Euler solved the problem by reformulating it in terms of a graph where vertices represent islands while edges represent the seven bridges joining any two islands as shown in Fig 2.3. Work published by Leonhard Euler [46] is considered the genesis of the story of network theory.



(a) (b)

Figure 2.3: The Königsberg bridges: (a) is a schematic diagram of the seven Königsberg bridges. (b) is a graph representing the Königsberg bridges. Source: [15].

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 the shortest path length of the network is, and many others.

2.4 Examples of real-world 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. In his work [84], 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 [61]. 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 [120]. Other networks include the dolphin network [115], terrorist networks [74], 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 [60]. 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.

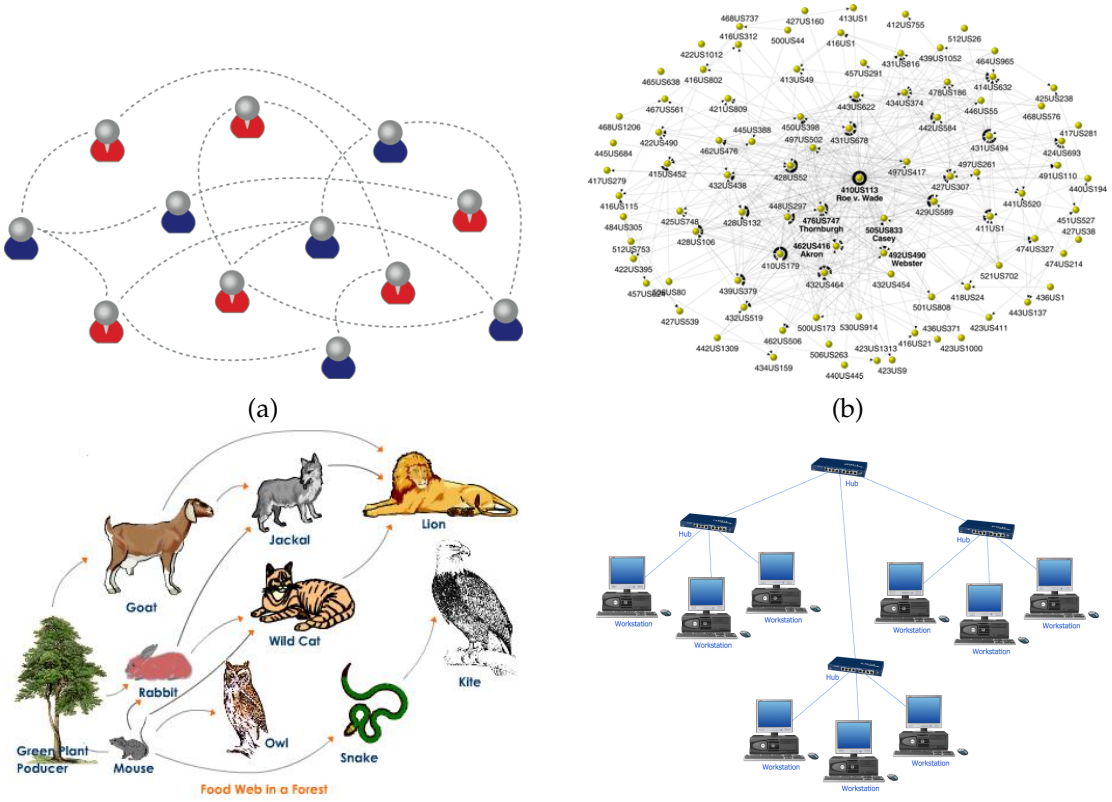
c) Technological networks:

   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 [7, 48, 89].

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 include the human brain network, protein-protein interaction network, network of metabolic path ways, ecological networks [39, 95, 100].



(a) (b)

(c) (d)

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

2.5 Matrix representations of graphs

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 graphs especially for large graphs whose structure cannot be captured by human eye. In addition, representing graphs by matrices enables the application of mathematical and computer tools on networks for purposes of summation, pattern identification and many others [24, 107]. In the subsequent subsections, we discuss the most common matrices used in the field of graph theory as well as their properties.

2.5.1 Adjacency Matrix

The Adjacency matrix (also known as binary adjacency) is a simple but very useful matrix commonly used in graph representation. It captures the connection between nodes in the graph that is to say,the edge structure of a graph. For a graph with n nodes, The adjacency matrix (A) is an $n \times n$ matrix with elements

$$\boldsymbol{A}_{ij} = \begin{cases} 1 & \text{if } i \text{ and } j \text{ are adjacent,} \\ 0 & \text{otherwise.} \end{cases} \tag{2.5.1}$$

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 edge directions have to be considered. For multigraphs, the entries of $\boldsymbol{A}$ 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 [16, 52]. For graphs with weighted edges, the adjacency matrix is given by

$$\boldsymbol{A}_{ij} = \begin{cases} w_{i,j} & \text{if } i \text{ and } j \text{ are adjacent,} \\ 0 & \text{otherwise.} \end{cases} \tag{2.5.2}$$

The spectrum, which is the eigenvalues and their multiplicities, of the Adjacency matrix is such a rich one and is thus used to 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 [31].
The summation of the absolute values of the eigenvalues of the adjacency matrix is referred to as the energy, E, of the graph. It is given by

$$E = \sum_i |\lambda_i|, \tag{2.5.3}$$

where λ_i is the ith eigenvalue of $\boldsymbol{A}$. The concept of energy of a graph has its origin in chemistry (by Gutman in 1978) where it is used in approximating the total π electron energy of molecules [53]. This concept has however been extended to other matrices resulting into different kinds of graph energy as discussed in [54, 56, 92]

2.5.2 Degree matrix

The degree matrix is a diagonal matrix that provides information about the degree of each node in a given network [83]. Given a network $G = (V, E)$ with $n = |V|$, the degree matrix $\boldsymbol{D}(G)$ is defined as

$$D_{i,j} = \begin{cases} k_i & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases} \tag{2.5.4}$$

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

2.5.3 Distance matrix

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

$$S_{i,j} = \begin{cases} l_{i,j} & \text{if } i \neq j \\ 0 & \text{otherwise,} \end{cases} \tag{2.5.5}$$

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

2.5.4 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 $\boldsymbol{B}(G)$ has entries defined as

$$B_{ij} = \begin{cases} +1 & \text{if node } v_i \text{ is the positive end of the edge } e_j \\ -1 & \text{if node } v_i \text{ is the negative end of the edge } e_j \\ 0 & \text{otherwise.} \end{cases} \tag{2.5.6}$$

2.5.5 Laplacian Matrix

The Laplacian matrix or graph Laplacian is another matrix used in representation of graphs. 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.

2.5.5.1 Definitions and properties of the Laplacian matrix

Definition 2.5.6 (Combinatorial Laplacian Matrix)**.** The Combinatorial Laplacian or unnormalised Laplacian matrix of a network is defined as the difference between the Degree matrix $\boldsymbol{D}$ and the Adjacency matrix $\boldsymbol{A}$ of a network. That is,

$$\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}. \tag{2.5.7}$$

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

$$L_{ij} = \begin{cases} k_{v_i} & \text{if } i = j \\ -1 & \text{if } i \neq j \text{ and } v_i \text{ is adjacent to } v_j \\ 0 & \text{otherwise,} \end{cases} \tag{2.5.8}$$

where k_{v_i} denotes the degree of node i [39].
Alternatively, we can define the combinatorial Laplacian matrix of a graph in terms of the vertex-edge incidence matrix $\boldsymbol{B}$. That is,

$$\boldsymbol{L} = \boldsymbol{B}\boldsymbol{B}^T, \tag{2.5.9}$$

where $\boldsymbol{B}^T$ is the transpose of $\boldsymbol{B}$ [39].

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 [32]. 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 [32].

3. Positive semi-definite

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

2.5.6.1 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 [39]. Let $\lambda_1 < \lambda_2 < \ldots < \lambda_n$ be the distinct eigenvalues of $\boldsymbol{L}$ and let $m(\lambda_1), m(\lambda_2), \ldots, m(\lambda_n)$ be their multiplicities. Then, the spectrum of $\boldsymbol{L}$ is written as

$$Sp\boldsymbol{L} = \begin{pmatrix} \lambda_1 & \lambda_2 & \ldots & \lambda_n \\ m(\lambda_1) & m(\lambda_2) & \ldots & m(\lambda_n) \end{pmatrix}. \tag{2.5.10}$$

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

- The eigenvalues of $\boldsymbol{L}$ are bounded as $0 \leq \lambda_j \leq 2k_{max}$ and $\lambda_n \geq k_{max}$ [39].

- The eigenvalue λ_1 is always equal to zero [39]. At least 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 $\boldsymbol{L}$ is equal to the number of connected components in the network [39].

- Every row sum and column sum of $\boldsymbol{L}$ is zero. Thus, the vector $\boldsymbol{v_1}$ of all ones is an eigenvector associated with $\lambda_1 = 0$, since $\boldsymbol{L}\boldsymbol{v}_1 = \boldsymbol{0}$ [32].

- 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 depicts how well connected the overall 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 [44].

- Let G be a graph with n connected components $G_i (1 \leq i \leq n)$. Then the spectrum of G is the union of the spectra of G_i (and multiplicities are added) [21].

- For a graph G, the sum of the eigenvalues, that is, the trace of $\boldsymbol{L}$ is twice the number of edges of G. Mathematically, $\sum_{i=1}^{n} \lambda_i = Trace(\boldsymbol{L}) = 2E$. [21]

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

- Star, $S_n : Sp(\boldsymbol{L}) = \{0\ 1^{n-2}\ n\}$.

- Complete, $K_n : Sp(\boldsymbol{L}) = \{0\ n^{n-1}\}$.

- Complete bipartite, $K_{m,n} : Sp(\boldsymbol{L}) = \{0\ m^{n-1}\ n^{m-1}\}$ [39].

Theorem 2.5.7 (Kirchoff's Matrix-Tree Theorem)**.** *If G is a connected graph with Laplacian matrix $\boldsymbol{L}$, then the number of unique spanning trees of G is equal to the value of any cofactor of the matrix $\boldsymbol{L}$ [58].*

2.5.8 Spectral matrix of the Laplacian

The spectral matrix is obtained by performing the eigenvector expansion for the Laplacian matrix $\boldsymbol{L}$ such that its columns are the scaled eigenvectors, that is

$$\boldsymbol{\Phi} = (\sqrt{\lambda_1}\boldsymbol{e}_1, \sqrt{\lambda_2}\boldsymbol{e}_2, \cdots, \sqrt{\lambda_n}\boldsymbol{e}_n), \tag{2.5.11}$$

λ_i and v_i are the ith eigenvalues and corresponding eigenvectors respectively.
The matrix $\mathbf{\Phi}$ is quite interesting as it is a complete representation of the graph. Moreover, we can reconstruct the original Laplacian matrix from $\mathbf{\Phi}$ using the formula $\boldsymbol{L} = \mathbf{\Phi}\mathbf{\Phi}^T$. It is important to note that the matrix $\mathbf{\Phi}$ is a unique representation of $\boldsymbol{L}$ iff all n eigenvalues are distinct or zero which implies that the eigenvectors are distinct as well. Wilson and Hancock have shown how graph features can be obtained from the coefficients of symmetric polynomials with elements of the spectral matrix constituting the set of variables [116].

2.5.9 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

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

where $\boldsymbol{D}^{-1/2}$ is the diagonal matrix determined by the inverse square root of each diagonal entry of the degree matrix [39]. For a k-regular graph, we have

$$\mathcal{L} = \boldsymbol{I} - \frac{1}{k}\boldsymbol{A}. \tag{2.5.12}$$

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 [28]. 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 equal to 2 only 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.

2.5.10 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

$$\boldsymbol{L}_r = \boldsymbol{L}D^{-1} \tag{2.5.13}$$

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

$$\boldsymbol{L}_r = \boldsymbol{D}^{1/2}\boldsymbol{L}\boldsymbol{D}^{-1/2}. \tag{2.5.14}$$

By diagonalizing $\mathcal{L}$, Eqn.2.5.14 can be rewritten as

$$\boldsymbol{L}_r = (\boldsymbol{D}^{1/2}\boldsymbol{\Phi})\boldsymbol{\Lambda}(\boldsymbol{D}^{1/2}\boldsymbol{\Phi})^{-1}, \tag{2.5.15}$$

where $\boldsymbol{\Lambda}$ is a diagonal matrix of eigenvalues of $\mathcal{L}$ and $\boldsymbol{\Phi}$ is the eigenvector matrix. Thus, $\boldsymbol{L}_r$ 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 $\boldsymbol{L}_r$ in place of $\mathcal{L}$.
A detailed account of other variants of the Laplacian matrix can be found in [106].

2.5.11 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 [18]) are given by

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

The sum of all elements of $\boldsymbol{R}$ is $2R$, where R is known as the Randić index. This index has numerous applications in chemistry with a historical one being its usage as a molecular structure descriptor, introduced by Milan Randić [93].
The Randić matrix is related to the symmetric normalised Laplacian matrix by

$$\mathcal{L} = \boldsymbol{I} - \boldsymbol{R}, \tag{2.5.16}$$

Let p_i denote the ith eigenvalue of $\boldsymbol{R}$, then $\lambda_i(\mathcal{L}) = 1 - p_i$. Moreover, for a graph without isolates, the Randić energy (given by $E_R(G) = \sum_i p_i$) is equal to the combinatorial normalised Laplacian energy (given by $E_{\mathcal{L}}(G) = \sum_i \lambda_i(\mathcal{L})$).

2.5.12 Transition matrix of a random walk on a Graph

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

$$\boldsymbol{P}_{ij} = \frac{1}{k_i}\boldsymbol{A}_{ij}, \tag{2.5.17}$$

In other words, $\boldsymbol{P}_G = \boldsymbol{D}_G^{-1}\boldsymbol{A}$ which is the normalised adjacency matrix.
The entry at $\boldsymbol{P}_{ij}$ indicates the probability for a random walker moving from vertex i to j. Like any other stochastic matrix, the eigenvalues of $\boldsymbol{P}$ are such that $|\lambda_i(\boldsymbol{P})| \leq 1$.
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 [13].

2.6 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 [22], 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 [77, 98, 113, 114].
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, that is where the interconnections between nodes remain constant with time:

2.6.1 Average degree of the nearest neighbors

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

$$k_{nn,i} = \frac{\sum_{j=1}^{n} a_{ij}k_j}{k_i}, \tag{2.6.1}$$

where a_{ij} is the (i,j)th 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 2.6.2 (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 [110]. It is also known as geodesic distance. It is possible to have more than one shortest path between a pair of nodes.

Definition 2.6.3 (Diameter of a network)**.** The diameter of a network is the maximum distance between any two nodes in the network [110]. 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 2.6.4 (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, and n is the size of the network.
The efficiency of a network is a measure of how efficiently 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 to their counterparts with lower values of efficiency.

Definition 2.6.5 (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

$$L_G = \frac{1}{n(n-1)} \sum_{i,j \in V,\ i \neq j} d_{ij}, \tag{2.6.2}$$

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 [110].

2.6.6 Clustering coefficient

The 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 [44]. 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 the clustering coefficient is in the interval $[0, 1]$. There are two types of clustering coefficients namely, the local and the global clustering coefficients.

Definition 2.6.7 (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

$$C_i = \frac{\text{number of pairs of neighbors of } i \text{ that are connected}}{\text{number of pairs of neighbors of } i} = \frac{2t_i}{k_i(k_i - 1)}, \tag{2.6.3}$$

where t_i is the number of triangles attached to node i. For nodes with degree equal to zero or one, we set $C_i = 0$ since there are no triangles attached to such nodes [83]. The average clustering coefficient for the network is given by

$$\bar{C} = \frac{1}{n}\sum_i C_i. \tag{2.6.4}$$

(2.6.5)

Definition 2.6.8 (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 [88]. Mathematically, we define the global clustering coefficient C as

$$C = \frac{3 \times \text{number of triangles}}{\text{number of connected triplets of vertices}} = \frac{\sum t_{\triangle}}{\sum t}, \tag{2.6.6}$$

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

2.6.9 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 [39]. Fig. 2.5 is of plots of some of the common degree distributions in networks namely Gaussian, Poisson, exponential and power-law degree distributions.

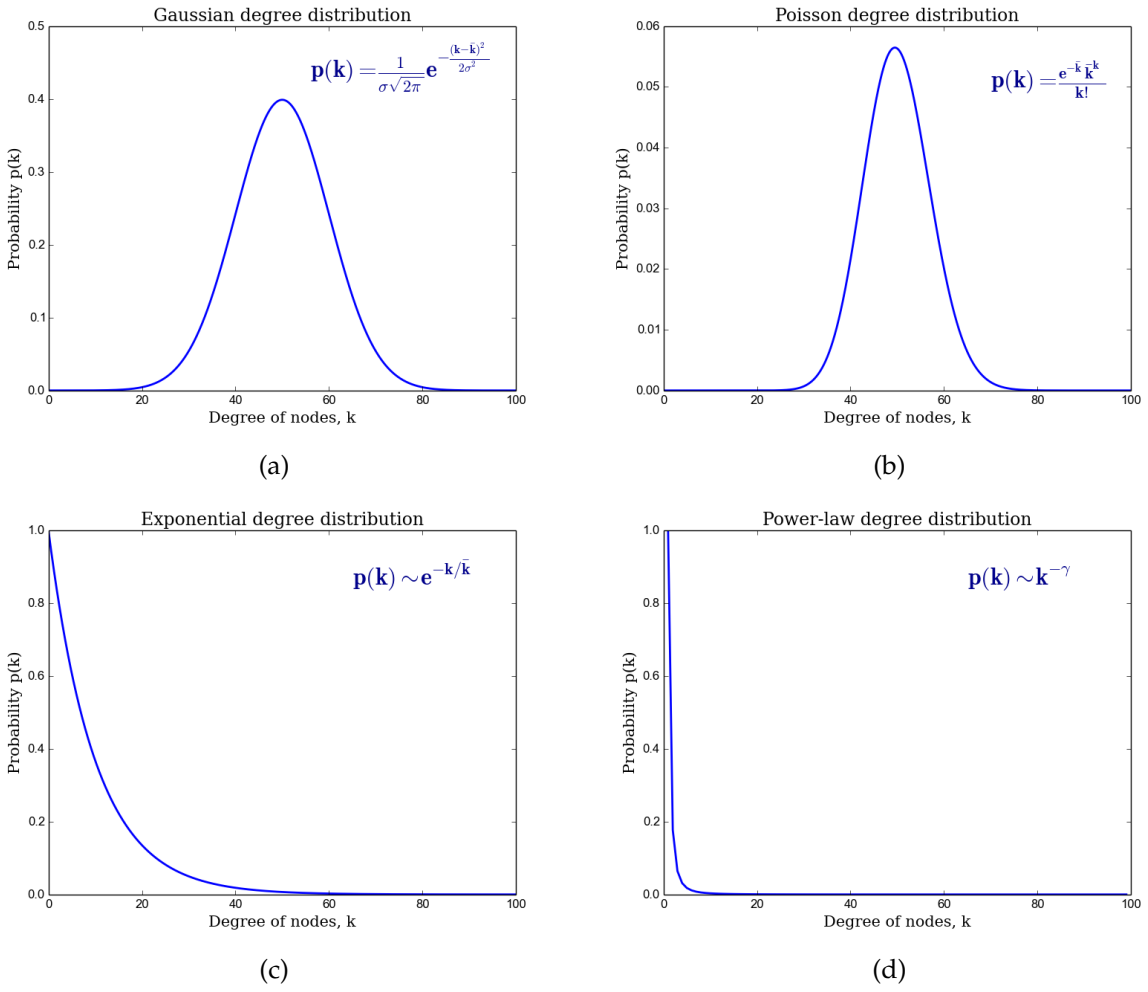


Figure 2.5: Common degree distributions of networks: (a) Gaussian distribution. (b) Poisson distribution. (c) Exponential distribution.(d) Power-law distribution.

2.6.9.1 Power-law degree distribution

In the power-law degree distribution, there exists few nodes with very high degree and many with very low degree. In otherwords, 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 [39]. Formally,

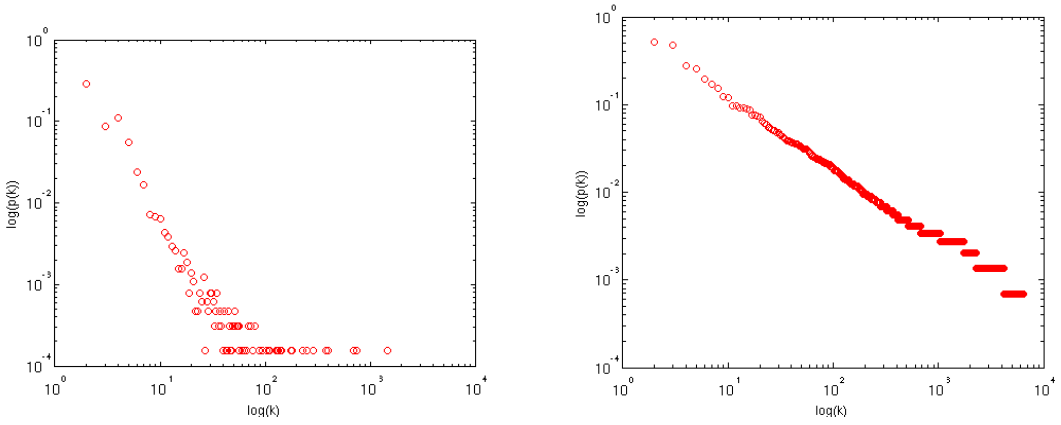
$$p(k) = Ck^{-\gamma}, \text{ for } 2 \leq \gamma \leq 3. \tag{2.6.7}$$

The range of the exponent between 2 and 3 was asserted by Barabási in Linked [8]. However, other literature indicate an exponent of grater than zero (that is, $\gamma > 0$).
Using a logarithmic scale, the plot of Equation (2.6.7) is a straight line, $\ln p(k) = -\gamma \ln k + \ln C$, with a slope equal to $-\gamma$ and an intercept equal to $\ln C$ as illustrated in Fig. 2.6(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 [39]. The plot of cummulative distribution function on a logarithmic plot is a straight line as shown in Fig. 2.6(b).



(a) (b)

Figure 2.6: Probability (a) and Cummulative Distribution Function (b) logarithmic plots for the version of the internet at autonomous system level (where anetwork or a collection of networks that are all managed and supervised by a single entity or organization) following a power-law distribution. Source: [82].

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

$$p(k, a) = C(ak)^{-\gamma} = a^{-\gamma} p(k) \propto p(k). \tag{2.6.8}$$

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.

In network science, there is an ambiguity in language regarding what 'scale free' really means. Some possible suggestions of what a scale free network is include: a network with power degree distribution with exponent in range 2 to 3, a network formed by the process of preferential attachment, one that follows a power law on some scales or one that does so throughout, and many others.

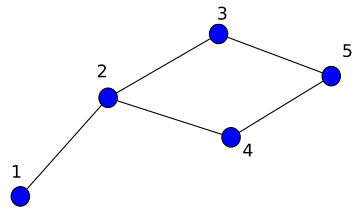
Work published by Barabási and his then graduate student Réka Albert [7] claimed that many real world networks (such as genetic networks and world-wide web) have node connectivities that follow a scale free power law degree distribution. They further explained the observed degree distribution is attributed to the mechanism of formation of these networks known as preferential attachment (also known as rich-get-richer phenomenon) in which a

new node coming into the network would most likely get connected to existing high-degree nodes (known as hubs) other than low-degree nodes. However, some researchers pointed out that preferential attachment is not the only mechanism that can be used in modelling networks that follow power law [30]. They further drew insights to the fact that networks can have the same power laws but different topologies. Moreover, this work triggered a multitude of publications affirming the existence of scale-freeness in biological, social, and many other real world networks [3, 19, 29, 62].

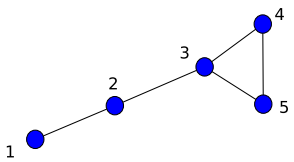
The claim that scale free networks are universal has been criticised by some network scientists and experts. For example, the scale freeness of some networks such as the power networks and metabolic networks has been questioned [4, 103]. The most recent and so far the most debatable criticism was raised by Clauset and Broido in [20] where they asserted that scale free networks are rare in nature not contrary to what was claimed earlier on. They based their argument on statistical analysis of nearly $1,000$ networks from social, technological, biological and many other fields. Results from their result indicate that only about 4% of the networks qualified highly with strongest direct evidence of scale-freeness in their structure and only 33% exhibited weak levels of direct evidence of scale free structures. On considering a relaxed criteria where indirect evidence of scale free structure in networks was checked, only 53% of the networks datasets passed the statistical tests in [20]. A very interesting debate on this issue is still on going as captured in [67], however, this posses an opportunity for research into this area.

2.6.9.2 Structure of network and 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. 2.7a & 2.7b) and whose degree distribution is the same (see table 2.7c).



(a)



(b)

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

(c)

Figure 2.7: Two simple graphs (a and b) of size 5 both have the same degree distribution given in table (c).

We then compute some of the structural properties of the two graphs (as in table 2.1) to ascertain whether their structures are the same.

	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 2.1: Some of the structural properties of the two graphs (Figs. 2.7a & 2.7b) whose degree distribution is the same.

Table 2.1 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.

2.6.10 Random models of networks

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

2.6.10.1 The Erdős-Rényi (ER) model

This type of random model was introduced by Erdős and Rényi in 1959. It is the best known model for random networks. In this model, we take n isolated nodes and then fix a probability p with which we link the nodes. For each pair of nodes, we generate a random number, r, uniformly from $[0, 1]$. If $p > r$, the two nodes are connected forming a network. Networks generated by the ER model have a small average path length and their average clustering coefficient tends to zero as n increases [39].

2.6.10.2 The Barabási-Albert (BA) model

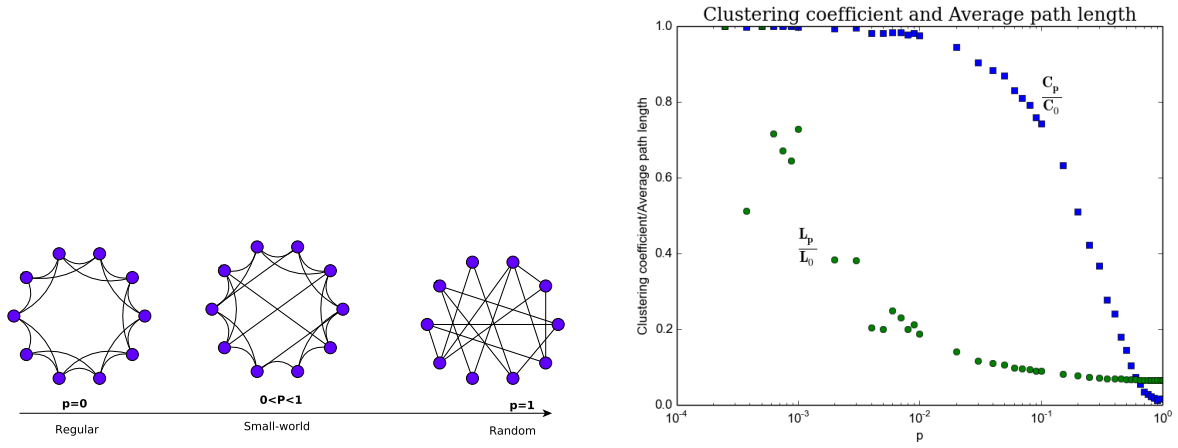
Most of the real-world networks follow a power-law degree distribution. This distribution was not manifested in ER networks and thus, Barabási and Albert introduced a network model that generates networks that mimic the degree distribution observed in the real-world networks. This model uses a preferential attachment, or rich get richer, mechanism to evolve a given initial graph. The preferential attachment technique involves an initial network with n_0 nodes, and at each step, a new node is introduced and connected to the existing nodes with a probability proportional to their degrees (or connectivity). The resulting networks are scale-free and have high clustering coefficients [39].

2.6.10.3 The Watts-Strogatz (WS) model

An experiment carried out by Stanley Milgram in 1967 [78] consisted of randomly selected people in the cities of Omaha and Wichita in the United States of America (USA) who were asked to send letters to target persons living in Boston. The individuals at the starting points were asked to send the letters to persons they knew. Despite the fact that the senders and their respective targets were separated by more than 2000km and that there was a total of 200 million inhabitants in the USA by then, results showed that the letters took an average of about six steps to reach the targets and that there was a large group interconnection where an acquaintance of an individual fed back into his own circle, thus eliminating new contacts. In terms of networks, two properties are observed: the former implies a small average path length of 6 while the latter implies a high clustering coefficient. It is from this experiment that the phrase "six degrees of separation" was born. This phrase explains a small world in which an individual is connected to another by a maximum of six steps. In 1998, [111] proposed a model that reproduces the two properties mentioned previously. This model starts with a circulant (or ring) network with n nodes connected to k nearest neighbors. With a fixed probability p, an end to each original link is rewired to a new randomly selected node. The WS model interpolates between a regular and a random network as shown in Fig. 2.8(a). The intermediate network is a small-world with high clustering coefficient and a very small average path length.
In Fig. 2.8(b), as the probability increases during the rewiring process, the average path length and clustering coefficient vary. The region where the average path length is small and the clustering coefficient is high corresponds to the small world network in Fig. 2.8(a).
In this study, we will perform simulations on networks developed by the ER and BA models using the following functions implemented in NetworkX:

```
nx.gnp_random_graph(n,p,seed=None)
nx.barabasi_albert_graph(n,m,seed=None),
```

where n, p, and m are the number of nodes, the probability for edge creation, and number of edges to attach from a new node to existing nodes respectively. The argument seed (an integer) is the seed for random number generator.



(a) (b)

Figure 2.8: The rewiring process: (a) Interpolation of WS model as probability increases. (b) Illustration of the variation of clustering coefficient and average path length during the rewiring process.

2.6.11 Robustness of complex systems

A complex systems is considered robust if it can withstand 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 can 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 [9] 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.

2.6.12 Robustness measures in networks

According to Ellens [36], 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 [102]. We explore some measures of robustness in networks which are categorised as follows:

2.6.12.1 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.

2.6.12.2 Distance-based measures

1. Diameter

   The diameter of a graph, denoted as D, is the maximum distance between pairs of nodes in the graph[110]. 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

   $$L_G = \frac{1}{n(n-1)} \sum_{i,j \in V,\ i \neq j} d_{ij}, \tag{2.6.9}$$

   where d_{ij} is the shortest path between node i and j and n is the total number of nodes in G [110]. 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

$$E = \frac{1}{n(n-1)} \sum_{i,j \in V, i \neq j} \frac{1}{d_{ij}}. \tag{2.6.10}$$

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.

2.6.12.3 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 disconnected. The one 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 which implies that this measure is insensitive to edge addition in some cases.

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 [58]. It is given by

   $$\xi = \frac{1}{n} \prod_{i=2}^{n} \lambda_i. \tag{2.6.11}$$

   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 [11].

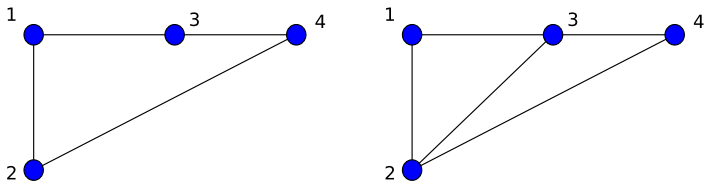
   Let us consider a simple example in which we illustrate how the both the algebraic connectivity behaves on edge addition.

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

   $$R = \sum_{1 \leq i < j \leq n} R_{i,j} = n \sum_{i=2}^{n} \frac{1}{\lambda_i} \tag{2.6.12}$$

   Unlike the algebraic connectivity, the effective resistance involves not only one but all the non-zero eigenvalues of the Laplacian. It is for this reason that any changes due



(a) (b)

Figure 2.9: The simple network in (a) has algebraic connectivity $\lambda_2 = 2$. Adding a new edge,$e_{2,3}$ to form a network in (b) with algebraic connectivity $\lambda_2 = 2$. We observe that on a change in network through edge addition, the algebraic connectivity remains constant.

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.[118]. 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 [39] 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,

$$S = \sum_{k=0}^{\infty} \frac{n_k}{k!}, \tag{2.6.13}$$

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

$$n_k = trace(\boldsymbol{A}^k) = \sum_{i=1}^{N} \lambda_i^k, \tag{2.6.14}$$

   where λ_i is the ith largest eigenvalue of $\boldsymbol{A}(G)$. Substituting for n_k in Eqn.2.6.13 gives

$$S = \sum_{k=0}^{\infty} \sum_{i=1}^{N} \frac{\lambda_i^k}{k!} = \sum_{i=1}^{N} \sum_{k=0}^{\infty} \frac{\lambda_i^k}{k!} = \sum_{i=1}^{N} e^{\lambda_i}. \tag{2.6.15}$$

   From Eqn. 2.6.15, 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

$$\bar{\lambda} = \ln\left(\frac{S}{N}\right) = \ln\left(\frac{\sum_{i=1}^{N} e^{\lambda_i}}{N}\right). \tag{2.6.16}$$

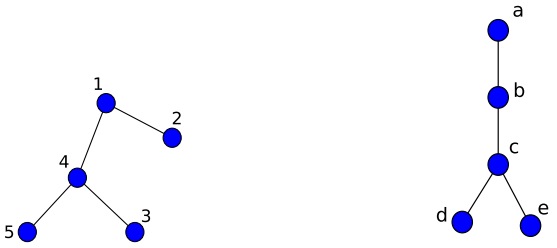
   Unlike the algebraic connectivity, the natural connectivity changes monotonically when edges are added or deleted which is one of the desired properties of a robustness measure.

2.6.13 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 [86, 121]. 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 [121]. 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 [121]. In other words, 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 as illustrate in Fig. 2.10. It is worth recalling that graph isomorphism is an NP-complete problem.



(a) (b)

Figure 2.10: 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 [71, 108, 112]

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 [51].

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 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 [63, 76, 121]

In this work, we use the concept of graph similarity as a basis for performing clustering (also known as graph clustering) on a given set of graphs extracted from objects. 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. In the subsequent subsection, we explore some of the key concepts we will encounter when extracting graphs from objects.

2.6.14 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 [105]. Though various algorithms for corner detection which are now considered state of the art, in this study we are interested in the Harris corner detection for demonstrating a principle of application and also since we are building on the earlier work of [101, 119]. Moreover, the algorithm could also be applied in this and other settings.
The Harris corner point detector was introduced by Harris and Stephens [57] as an improvement to the Moravec's corner detector [79, 80]. In his method, Moravec considered corner points as the points where there is intensity variation in all directions. The Harris corner detector is widely used because it is simple to compute, fast and most importantly, the corner

points obtained based on this method are invariant in position to rotation, scale, illumination, and partially to affine intensity changes as discussed in detail in [101]. On the other hand, the Harris corner detector is not invariant to image scaling, however, there are various ways of going about this problem as explained in [101].
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 in 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 [101]

$$E_{x,y} = \sum_{u,v} w_{u,v} |I_{x+u,y+v} - I_{u,v}|^2, \tag{2.6.17}$$

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 Taylor expansion of E gives:

$$E_{x,y} = \sum_{u,v} \{[I_x(u,v)x]^2 + [I_y(u,v)y]^2 + 2I_x(u,v)I_y(u,v)xy\}, \tag{2.6.18}$$

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

$$E(x,y) = \mathbb{M} \begin{bmatrix} x & y \end{bmatrix} \left(\sum_{u,v} w(u,v) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \right) \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x & y \end{bmatrix} \mathbb{M} \begin{bmatrix} x \\ y \end{bmatrix} \tag{2.6.19}$$

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 [57], the measure of corner and edge quality used for selecting core pixels known as the response function, denoted as R is given by

$$R = det(\mathbb{M}) - kTrace(\mathbb{M})^2, \tag{2.6.20}$$

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 [101].

2.6.15 Voronoi diagrams and Delaunay triangulation

2.6.15.1 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, finding clear routes in autonomous robot navigations, among others [87]. However, in later chapters, we will explore the application of Voronoi diagrams to image segmentation as explained in [101].

Definition 2.6.16. 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 [6]. 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 other hand, is the common boundary of two adjacent cells.

One simple and practical example that also doubles as an application of voronoi diagrams. Let us consider n ambulances placed at different spots of a city. The spots are a subset of points denoted by $S = p_1, p_2, \cdots, p_n$. We assume that the distance between two points is given by the Euclidean distance function:

$$\ell_2 = d[(a_1, a_2), (b_1, b_2)] = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2}. \tag{2.6.21}$$

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 definitions used in the study of Voronoi diagrams as by ref [6]:

Definition 2.6.17 (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 2.6.18 (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 2.6.19 (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 2.6.20 (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 2.6.21 (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.

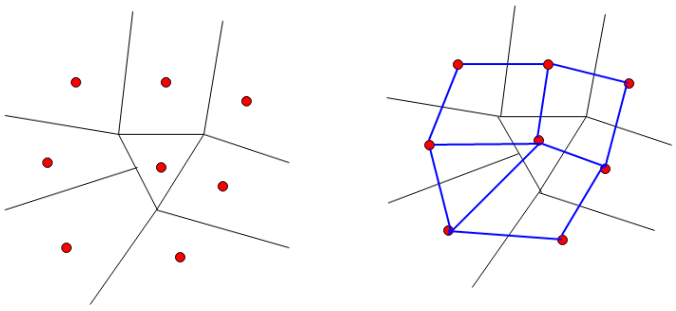
Voronoi diagrams exhibit interesting properties which include the following:

- 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: $v - e + n = 2$ (where v, e, and n are the number of vertices, edges and faces of a planar graph respectively), 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.

2.6.21.1 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 [6]. Delaunay Triangulation have been applied in various fields for example climate and global modeling [64], modeling computer vision problems [34], and many others.
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.



(a) (b)

Figure 2.11: A sample Vorronoi diagram on 8 seeds/points (red) (a) and its corresponding dual graph (Delaunay Triangulation in blue) superimposed (b).

Corner point detection, Voronoi diagrams, and Delaunay triangulation 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 mentioned concepts.

2.6.22 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 represented in graph form 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 efficient and 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,

$$\Phi : G \longrightarrow \mathbb{R}^n, \quad g \longrightarrow \Phi(g) = (f_1, f_2, f_3, \cdots, f_n) \tag{2.6.22}$$

There are various methods studied in literature that are used in creating feature vectors of graphs. These can be categorised as spectral clustering methods, Substructure finding methods, and Dissimilarity representation. For purposes of this work, we explore the spectral clustering methods and related invariants in Chapter.

2.6.23 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 [97]. 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 in [99] :

Definition 2.6.24 (Mean)**.** The mean, $\bar{X}$, of a sample is the average of elements of the sample and is given by

$$\bar{X} = \frac{\sum_{i=1}^{n} X_i}{n} \tag{2.6.23}$$

Definition 2.6.25 (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

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{(n-1)}} \tag{2.6.24}$$

Definition 2.6.26 (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

$$\sigma^2 = \frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{(n-1)} \tag{2.6.25}$$

It is important to note that both the standard deviation and variance measure the spread of a data set that is 1-dimensional.

Definition 2.6.27 (Covariance)**.** For data sets with more than 1 dimension, we use the covariance instead of variance (or standard deviation) to measure the deviation from the mean amongst any pair of dimensions. The covariance of two dimensions x and y is given by

$$cov(X,Y) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{(n-1)} \tag{2.6.26}$$

Definition 2.6.28 (Covariance matrix)**.** For a set of data with n dimensions, the covariance matrix is given by

$$C^{n\times n} = (c_{i,j}, c_{i,j} = cov(Dim_i, Dim_j)), \tag{2.6.27}$$

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

$$C = \begin{pmatrix} cov(x,x) & cov(x,y) & cov(x,z) \\ cov(y,x) & cov(y,y) & cov(y,z) \\ cov(z,x) & cov(z,y) & cov(z,z) \end{pmatrix} \tag{2.6.28}$$

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 corresponding to largest eigenvalues. The principal components are the directions where there is most variance (that is, where data is most spread). 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

$$NewData = FeatureVector^T \times NormalisedData^T \tag{2.6.29}$$

Having obtained the feature vectors as explained in the previous subsection, we create a database whose columns are the feature vectors for each graph. We can then apply principal components analysis to the database so as to embed the respective graphs onto low dimension space as we will elaborate in the following chapters.

Chapter 3

Diffusion on networks

3.1 Introduction

Diffusion is a particular example of a broad concept of dynamical systems on networks. A dynamical system is defined as a system whose state, as represented by a given set of quantitative variables, changes over time according to some given rules or equations [83]. Diffusion on networks aid in developing simplified models of real-world processes that occur in real-world systems for example propagation of infection through a population, transmission of signals in brain networks, dissemination of information in a social network, virus spread in computer network, spread of money within an economy, and many others [66].
According to [83], diffusion is, among others, the movement of substance from a region of high concentration to a region of low concentration. In network theory, diffusion is a process, by which behaviours such as information, heat, viruses and any others spread over networks. In the study of diffusion process in networks, a familiar model discussed in literature is based on the idea that the behaviour under study spreads through a network along paths that connect pairs of nodes. This model has been widely used in modeling the spread of information, ideas, opinions in social networks, social media marketing of commodities and infection spreads among population [12, 43, 66, 72].
Estrada pointed out that in modeling the spread of epidemics, the most important task is the determination of the network of social contacts over which the spread of infections occurs. Though it is an easy task in cases such the spread of sexually transmitted diseases, it is however a challenging task in cases that involve spread of airborne or close contact infections [43].This is so because the airborne or close contact infections can be transmitted in two ways namely through interactions between individuals that connected to each other in the social network known as close contacts or through encounters between two individuals that are not close to each other in the network. Such contacts are known as casual contacts. The causal contacts play a vital role in the spread of infection as evidenced in the case of the severe acute respiratory syndrome (SARS) discussed in [95], spread of obesity [29] and epidermic hysteria [17]. It is thus of utmost importance to account for the casual contacts when modeling the spread of infections. To that cause, various experiments have been carried for purposes

of accounting for both close and casual contacts. An interesting attempt by Estrada et.al [43] account for both the close and casual contacts among individuals by considering the transmission of an infection through paths and by long-range (LR) interactions in a complex network. This method is based on the idea that with the structure of network of close contact to hand then the casual contacts can be deduced for any pair of individuals as long range interactions based on the distance of separation between them. It is worth noting that in this model, the long-range interactions are considered to be nonrandom as opposed to other models in which casual contacts were assumed to occur randomly.
In this work, we discuss the diffusion of heat on a network using the model where diffusion occurs along the edges of a network. We also study the equilibrium behaviour under this model and also investigate the role played by the structure of a network, choice of source nodes, and network homogeneity on the diffusion process on a network. Moreover, we explore diffusion on directed networks and discuss equilibrium behaviour on different types of directed networks.
Furthermore, we consider the diffusion model presented in [41] that involves both direct interactions (among nearest neighbor nodes) as well as long-range interactions between pairs of non-nearest nodes in the network. Our interest in this model is the significant contribution og the long-range interactions observed in many processes in real-world for example epidermic spread in population, synchronisation processes in physics, consensus problems, to mention but a few.

3.2 Heat diffusion model

Recently, various models have been developed to depict the process of heat diffusion on networks. In this study, we consider a simple model described in [83]. Let us consider heat flow over as simple network as described below:
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

$$\frac{d\phi_i(t)}{dt} = \varepsilon \sum_j (\boldsymbol{A}_{ij} - \delta_{ij} k_i)\phi_j(t), \tag{3.2.1}$$

where $\boldsymbol{A}$ 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

$$\frac{d\boldsymbol{\phi}(t)}{dt} = -\varepsilon \boldsymbol{L}\boldsymbol{\phi}(t), \quad \boldsymbol{\phi}(0) = \boldsymbol{\phi}_0, \tag{3.2.2}$$

where $\boldsymbol{L}$ is the Laplacian matrix of a graph. Alternatively, the normalised version $\mathcal{L}$ of the Laplacian is used.
The solution to equation 3.2.2 is

$$\boldsymbol{\phi}(t) = \boldsymbol{\phi}_0 \, e^{-\varepsilon t \boldsymbol{L}}. \tag{3.2.3}$$

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

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

where λ_i, $\boldsymbol{v}_i$ are respectively the eigenvalues and corresponding eigenvectors of the Laplacian matrix and $\langle \boldsymbol{\phi}(0), \boldsymbol{v}_i \rangle$ is simply the projection of $\boldsymbol{\phi}(0)$ onto the set of eigenvectors [5].

3.3 Equilibrium behaviour

For any dynamical system, equilibrium is a very vital concept. The equilibrium is the simplest possible solution to a dynamical system. It is a solution where the state variables do not change with time. Most importantly, the study of equilibrium lays a foundation for analysing the behaviour of a system when time is not a factor.
In this section, we study the equilibrium of diffusion process on a simple undirected network. As t goes to infinity, we have

$$\lim_{t\to\infty} e^{-\varepsilon \lambda_i t} = \begin{cases} 0 & \text{if } \lambda_i > 0 \\ 1 & \text{if } \lambda_i = 0, \end{cases} \tag{3.3.1}$$

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

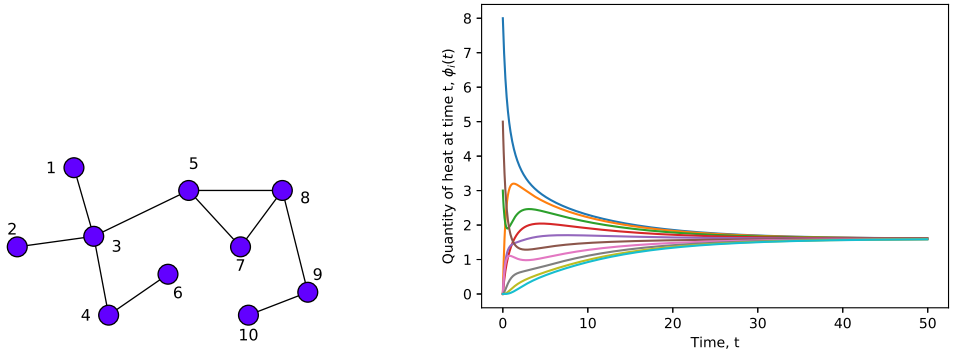
$$\lim_{t\to\infty} \boldsymbol{\phi}(t) = \langle \boldsymbol{\phi}(0), \boldsymbol{v^1} \rangle \boldsymbol{v^1}. \tag{3.3.2}$$

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

$$\lim_{t\to\infty} \phi_j(t) = \frac{1}{n} \sum_{i=1}^{n} \phi_i(0). \tag{3.3.3}$$

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 concept of equilibrium behaviour on networks, let us consider a simple example:

Example 3.3.1. Let us consider diffusion of heat over the network in Fig. 3.1(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. 3.1(b) illustrates how heat spreads over the network in Fig. 3.1(a).



(a) (b)

Figure 3.1: Results from simulation of diffusion process over the network in (b).

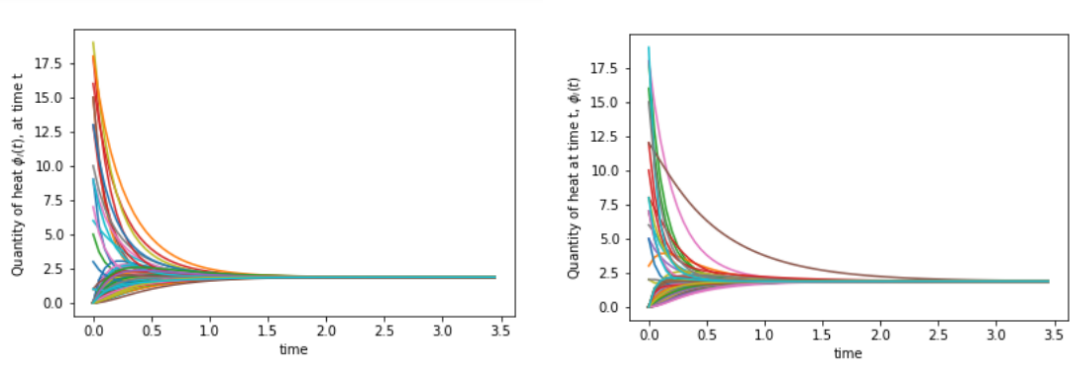
From Fig.3.1, we observe that at each time step t, nodes that initially have high amounts of heat (i.e $1, 3, 5, 6$, and 10) exchange heat with adjacent nodes that initially had none or little amounts of heat. The latter gain heat from the former and eventually all nodes in the network have relatively equal amounts of heat. This explains the fact that as time t increases, the quantity of heat $\boldsymbol{\phi}_j(t)$ at each node tends to the equilibrium value of 0.2 which is attained as t approaches 50.

From Eqn.3.3.3, it is clear that at equilibrium, the state variables at each of the node has the same value which is the average value of the initial values at all nodes in the network. It should however, be noted that it is not always the case for all networks for instance in some directed networks, equilibrium can be attained at different values (not the average of initial values) at each of nodes as we will illustrate in subsequent sections.

3.4 Impact of network structure on the dynamics of diffusion processes

The structure of a network as discussed before is 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 via interactions within neighbourhoods of nodes, it reasonable to investigate the impact of network structure on the diffusion process on that network. In this section therefore, we discuss the role played by the structure of the network in the spreading process.

We consider two networks with different structures: one is an Erdős-Rényi (ER) network that follows a Poisson degree distribution and the other Barabási-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 3.2.



(a) (b)

Figure 3.2: results of the simulations for diffusion on networks following equation 3.2.2. BA network (left) and ER (right). Both networks have 100 nodes and average path length 2.3.

In Fig 3.2 we observe that diffusion occurs much faster in BA network than in ER network based on the time at which equilibrium is attained. In BA, equilibrium is reached after about 1.5 time steps while in ER network equilibrium is reached much later after about 2.5 time steps. This behaviour can be attributed to the difference in structure of the two networks. The BA network is composed of highly connected nodes known as hubs that interact with a big number of nodes at the same time thus fastening the diffusion process. On the other hand, the ER network almost all nodes have a few number of nearest neighbours and thus spread of heat or information is less rapid compared to the previous case. Following these results, we can thus appreciate that the structure of a network plays a significant role in the diffusion process and thus a determinant of how fast equilibrium can be attained.

3.5 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 [1, 38, 84]. The star network is considered the most heterogeneous for which the average degree tends to two as the number of nodes increases. On the other hand, the regular networks is considered the least heterogeneous as all nodes have the same degree, that is to say, regular networks are homogeneous.

From the degree distribution, we can obtain some useful insights about the heterogeneity of a network. However, this technique may not perform well some cases. For example situation where more than one degree distribution fits for a given network, comparison between two networks with totally different degree distributions, among others. To work around such cases, various measures that uniquely quantify network homogeneity have been put forward [2, 14, 38].

For simulations, we consider heterogeneity in scale free networks by varying power exponent, γ. Each of the networks consists of 1000 nodes and of average degree $\bar{k} = 20$. We commence by assigning random initial quantities of heat to each of the 100 nodes with highest degree. Fig. 3.3 illustrates results of simulations of diffusion over the networks with time.

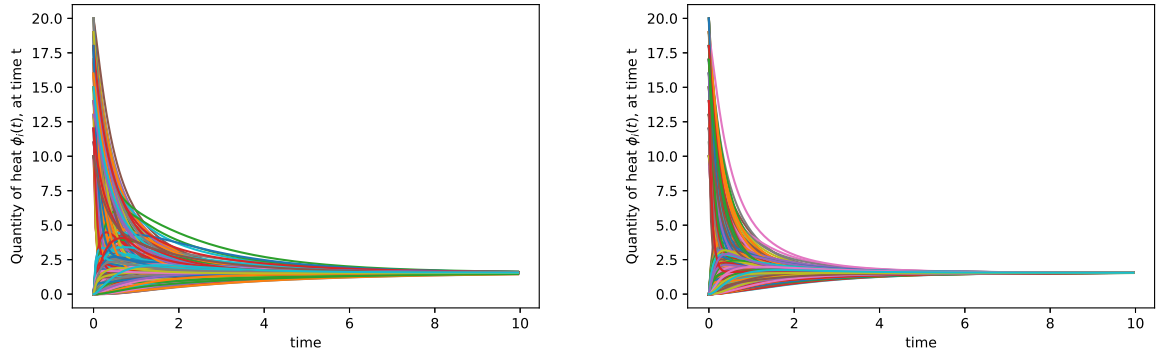
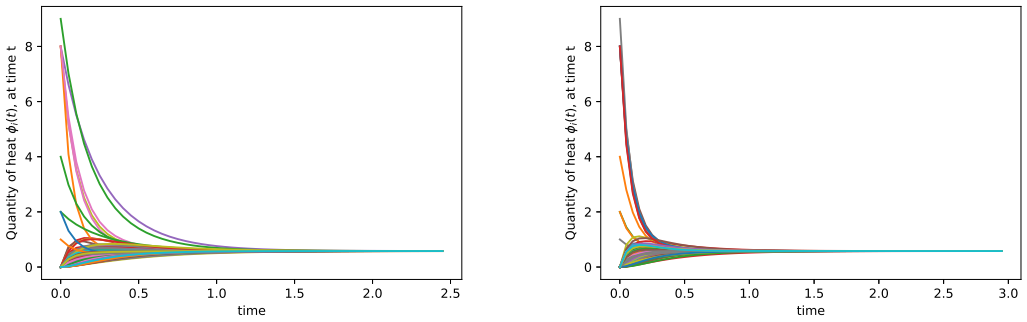


Figure 3.3: Results of simulation on two BA networks with power exponents 2.0 (left) and 2.3 (right).

3.6 Impact of choice of initial diffusion nodes on the diffusion process on networks

We consider an ER network of 100 nodes and average path length of 6. Firstly, we randomly select 5 nodes in both networks. Secondly, we select 10 nodes with the highest degree centrality to which we assign specific amounts of heat. We then consider diffusion on the network for the two options of choice of initial nodes and results are illustrated in Fig. 3.4.



(a) (b)

Figure 3.4: Results of the simulations for diffusion on ER network of size 100. Illustration at the left corresponds to diffusion for which source nodes are selected randomly while the one at the right side corresponds to the case for which initial nodes are ones with highest degree.

From the simulations, we observe that when the source nodes (those from which diffusion commerces) 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 other hand, 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.

3.7 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 structural 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 occur on such networks as this can be used in modeling processes on these systems. There are various categories of directed networks which include:

Definition 3.7.1 (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 [94].

Definition 3.7.2 (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 [94].

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

$$\frac{d\boldsymbol{\phi}}{dt} = -\varepsilon \boldsymbol{L}\boldsymbol{\phi}, \quad \boldsymbol{\phi}(0) = \boldsymbol{\phi}_0, \tag{3.7.1}$$

where $\boldsymbol{L}$ is the graph Laplacian of directed graph. This matrix is normally unsymmetrical which implies that the dynamics on the digraph are possibly different from those of their undirected counterparts.

Definition 3.7.3 (Balanced Graphs)**.** We say the node v_i of a digraph $G = (V, E)$ is balanced if and only if its in-degree (number of edges towards a particular node) and out-degree (number of edges leaving a particular node) 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 $\sum_j a_{ij} = \sum_j a_{ji}, \forall i$ [94].

3.7.4 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 digraph Laplacian. Let $G = (V, E)$ be a digraph with Laplacian $\boldsymbol{L}$ with eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_n$ in non-decreasing order.

3.7.4.1 Estimation of eigenvalues of the Laplacian

From the Gershgorin disk theorem, a strogly connected digraph G with laplacian matrix $\boldsymbol{L}$ has has a zero eigenvalue $\lambda_1 = 0$ and all the other non-trivial eigenvalues have non-negative real parts.
Let $\boldsymbol{\phi}_0$ be the vector of quantities of heat at all nodes at $t = 0$, $\varepsilon = 1$ be the diffusion coefficient. Similar to undirected case, the diffusion process on this network is given by

$$\boldsymbol{\phi}(t) = \boldsymbol{\phi}_0 \, e^{-t\boldsymbol{L}}. \tag{3.7.2}$$

The equilibrium behaviour on strongly connected digraphs can be studied using this theorem:

Theorem 3.7.5 (Limit Theorem for Exponential Matrices)**.** *Assume G is a strongly connected digraph with Laplacian $\boldsymbol{L}$ satisfying $\boldsymbol{L}\boldsymbol{v}_r = \boldsymbol{0}$, $\boldsymbol{v}_l^T\boldsymbol{L} = \boldsymbol{0}$, and $\boldsymbol{v}_l^T\boldsymbol{v}_r = 1$. Then*

$$R = \lim_{t \longrightarrow +\infty} e^{(-t\boldsymbol{L})} = \boldsymbol{v}_r\boldsymbol{v}_l^T \in M_n, \tag{3.7.3}$$

where M_n denotes a set of square $n \times n$ matrices, $\boldsymbol{v}_r$ and $\boldsymbol{v}_l$ denote the right and left eigenvalues of $\boldsymbol{L}$ associated with eigenvalue $\lambda_1 = 0$ [94].

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

$$\lim_{t \longrightarrow \infty} = \boldsymbol{\phi}_0\boldsymbol{v}_r\boldsymbol{v}_l^T \tag{3.7.4}$$

It is important to note that following Equation 3.7.4, 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 3.7.6. Consider a directed network $G = (V, E)$ that is strongly connected. Then the digraph G globally attains average equilibrium if and only if $\boldsymbol{1}^T\boldsymbol{L} = \boldsymbol{0}$ [94].

3.7.7 Equilibrium state for balanced graphs

The proposition in [94] states that

Proposition 3.7.8. 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:

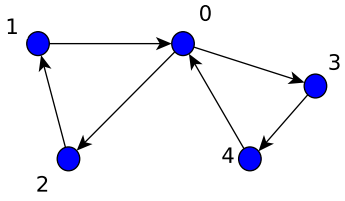
a) G is balanced,

b) $v_l = \boldsymbol{1}$ is the left eigenvector of the Laplacian of G associated with the zero eigenvalue, that is, $\boldsymbol{1}^T\boldsymbol{L} = 0$.

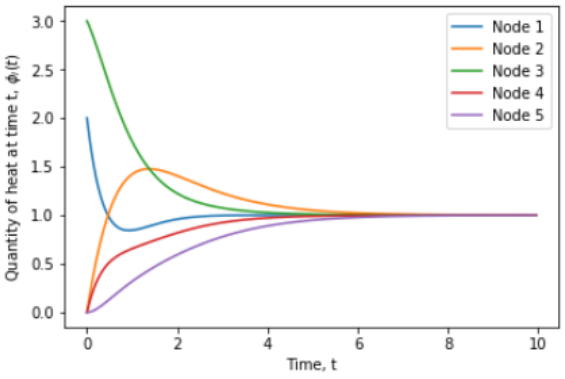
c) $\sum_{i=1}^{n} u_i = 0, \forall x \in \mathbb{R}^n$ with $u_i = \sum_{j \in N_i} a_{i,j}(x_j - x_i)$.

Since for a balanced digraph v_l is an all ones vector, it therefore follows from Proposition 3.7.8 that at equilibrium, the value at all nodes in a balanced graph is the average of the initial values at all nodes.

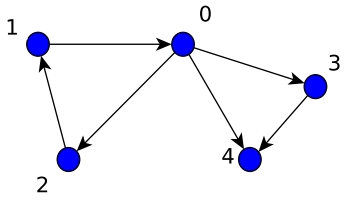
Example 3.7.9. Let us consider two directed graphs, one is a balanced digraph and the other is not. We then assign initial quantities of heat to all nodes in the order 0 to 4 as in the vector $\phi_0 = [2, 0, 3, 0, 0]$ and set the diffusion coefficient, $C = 1$. We then obtain plots for diffusion on both graphs after a specific time t as shown in Figs. 3.5(b) and 3.5(d).



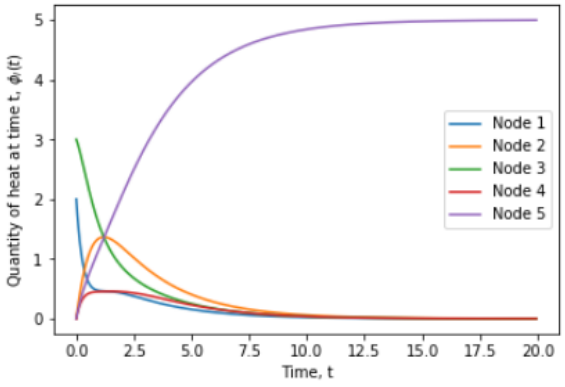
(a)



(b)



(c)



(d)

Figure 3.5: 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. 3.5a, its Laplacian matrix $\boldsymbol{L}$,v_r and v_l are respectively:

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

The values for v_r and v_l satisfy Theorem 3.7.5 as well as Proposition 3.7.8 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. 3.5c, we have the following matrices

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

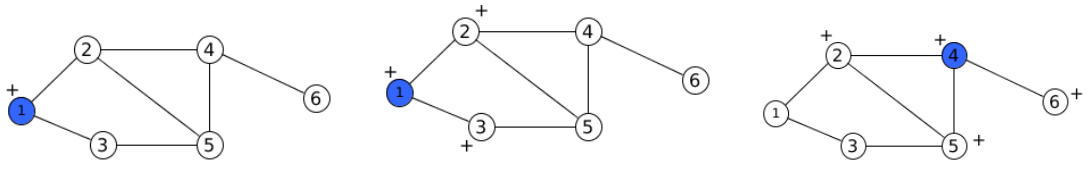
We observe that $\boldsymbol{L}\boldsymbol{v}_r = 0$ and $\boldsymbol{v}_l^T\boldsymbol{L} = 0$. However, the condition $\boldsymbol{v}_l^T\boldsymbol{v}_r = 1$ is not satisfied and therefore equilibrium cannot be attained as shown in Fig. 3.5. In addition, we observe that considering the structure, 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. 3.5.

3.8 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 in various dynamical processes in real-world for example in the spread of infections among people, consensus process in multi-agent systems, synchronisation dynamics, among others. A number of models to capture these interactions have been presented in literature for example Random Walks with Levy Flights (RWLF) [68], Fractional Diffusion Equation (FDE), k-path Laplacian based model[42], and many others. In this work, however, we adopt an elegant model based on k-path Laplacian matrices introduced by Estrada [42]. To start with, let us understand what the k-path Laplacian matrices are.

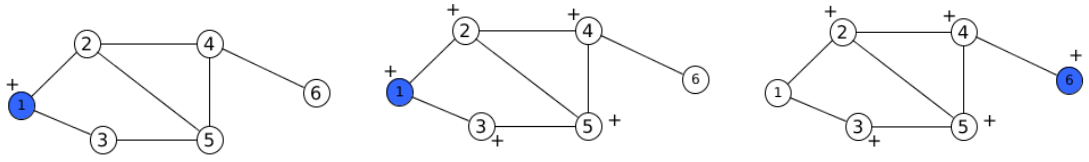
3.8.0.1 k-path Laplacian matrices, $\boldsymbol{L}_k$

The k-path Laplacian matrices are a natural generalisation of the combinatorial Laplacian of a graph [40]. 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 the k-path Laplacian through considering a polarisation process on a network, that is to say, 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.3.6 and Fig.3.7 for $d = 1$ and $d = 2$ respectively.



(a) (b) (c)

Figure 3.6: 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 3.7: 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 3.8.1 (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} [40].

Definition 3.8.2 (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 [40].

For a simple graph in Fig.3.8 with diameter equal to 2, the k-degree ($k \leq 2$) for each vertex is summarised in Table.

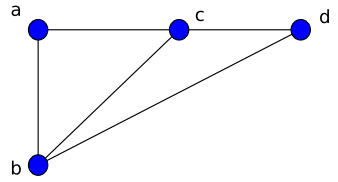


Figure 3.8: A Graph of size 4.

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

Table 3.1: k-path degree for vertices of graph in Fig. 3.8.

Definition 3.8.3 (k-path Laplacian matrix)**.** The k-path Laplacian matrix $\boldsymbol{L}_k(k \leq d_{max})$ of a connected undirected graph $G = (V, E)$ is defined as the square symmetric $n \times n$ matrix whose entries are given by

$$\boldsymbol{L}_k(ij) = \begin{cases} \delta_k(i) & \text{if } i = j, \\ -1 & \text{if } d_{i,j} = k, \\ 0 & \text{otherwise,} \end{cases} \tag{3.8.1}$$

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.3.8. Note that since $d_{max} = 2$ for the graph, then we consider $k = 1$ and $k = 2$.

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

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

$$\boldsymbol{y}^T c_k \boldsymbol{L}_k \boldsymbol{y} \geq 0 \quad \text{for } c_k > 0 \tag{3.8.2}$$

The concept of k-path Laplacians defined in Eqn.3.8.3 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.

$$\delta_k(v) := \#\{w \in V : d(v, w) = k\}. \tag{3.8.3}$$

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

$$\langle f, g\rangle = \sum_{v\in V} f(v)\overline{g(v)}, \quad f, g \in \ell^2(V) \tag{3.8.4}$$

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

$$e_v(w) := \begin{cases} 1 & if w = v, \\ 0 & otherwise. \end{cases} \tag{3.8.5}$$

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

$$(\boldsymbol{L}_k)(f) := \sum_{w\in V: d(v,w)=k} (f(v) - f(w)), \quad f \in C(V). \tag{3.8.6}$$

Definition 3.8.4 (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 $\eta_k(G) = m[\lambda_1(\boldsymbol{L}_k) = 0]$ [40].

Example 3.8.5. Let us consider the graph, G in Fig.3.8, since $d_{max} = 2$ we compute the 1-hopping and 2-hopping connected components of G.

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

Table 3.2: Computation of connected components of the graph Fig.3.8.

We see from Table. 3.2, 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 $0, 0, 0, 2$.

3.8.6 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

$$\boldsymbol{L}_G = \sum_{k=1}^{\Delta} c_k \boldsymbol{L}_k \tag{3.8.7}$$

where $1 \leq \Delta \leq d_{max}$ and c_k are the coefficients [40]. 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 [40] Estrada proposed two approaches categorised as social and physical ways of influence.

3.8.7 Choice of co-efficients, c_k

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.There exists various ways of modeling long-range interaction. In this study, we consider two ways as presented in [42]:

a) **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

$$\boldsymbol{L}_G = \boldsymbol{L} + \sum_{k=2}^{\infty} e^{-\lambda k} \boldsymbol{L}_k, \tag{3.8.8}$$

where $\lambda > 0$. Thus, the coefficients of Eqn. 3.8.15 are $c_1 = 1$ and $c_{k\geq 2} = e^{-\lambda k}$.

b) **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

$$\boldsymbol{L}_G = \sum_{k=1}^{\infty} k^{-s} \boldsymbol{L}_k, \tag{3.8.9}$$

For this case, the coefficients of Eqn. 3.8.15 are $c_k = k^{-s}$. In [42], 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, $\boldsymbol{L}_G$, is derived. It is thus given by

$$\boldsymbol{L_G} = \tilde{\boldsymbol{L}}_\tau = \begin{cases} \boldsymbol{L} + \sum_{k=2}^{\Delta} e^{-\lambda k} \boldsymbol{L}_k, & \text{for } \tau = \text{ Laplace}, \lambda > 0 \\ \sum_{k=1}^{\Delta} k^{-s} \boldsymbol{L}_k, & \text{for } \tau = \text{ Mellin}, s > 0, \end{cases} \tag{3.8.10}$$

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

3.8.8 Properties of the generalised Laplacian matrix

- The generalised matrix $\boldsymbol{L}_G$ is real and symmetric which follows from the fact that $\boldsymbol{L}_G$ 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 $\boldsymbol{y}$

  $$\boldsymbol{y}^T \boldsymbol{L}_G \boldsymbol{y} = \boldsymbol{y}^T (c_1 \boldsymbol{L}_1 + c_2 \boldsymbol{L}_2 + \cdots + c_\Delta \boldsymbol{L}_\Delta) \boldsymbol{y} = \boldsymbol{y}^T c_1 \boldsymbol{L}_1 \boldsymbol{y} + \boldsymbol{y}^T c_2 \boldsymbol{L}_2 \boldsymbol{y} + \cdots + \boldsymbol{y}^T c_\Delta \boldsymbol{L}_\Delta \boldsymbol{y} \tag{3.8.11}$$

  Since $\boldsymbol{y}^T c_k \boldsymbol{L}_k \boldsymbol{y} \geq 0$ for $c_k > 0$ and $1 \leq k \leq \Delta$ as in Eqn. 3.8.2, then

  $$\boldsymbol{y}^T \boldsymbol{L}_G \boldsymbol{y} \geq 0 \tag{3.8.12}$$

  □

- Like for the normal Laplacian matrix, zero is always an eigenvalue of $\boldsymbol{L}_G$ with eigenvector, $\mathbf{1}$, an all ones vector.
- Behaviour of eigenvalues with change in s (or λ). As discussed in [41], for a graph $G = (V, E)$ with generalised Laplacian matrix $\boldsymbol{L}_G$ formed by the Laplace or Mellin transform of path Laplacian, its eigenvalues behave as follows:

  $$\left(\frac{\mu_i}{N}\right) \longrightarrow \begin{cases} \left(\frac{\lambda_i}{N}\right), & \text{if } \lambda, s \longrightarrow \infty \\ 1, & \text{if } \lambda, s \longrightarrow 0, \end{cases} \tag{3.8.13}$$

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

3.8.9 Time complexity of the generalised Laplacian matrix

In computing of the generalised Laplacian matrix $\boldsymbol{L}_G$, we need to compute k-path Laplacian matrices for $1 \leq k \leq d_{max}$ which we obtain from the distance matrix $\boldsymbol{D}$. The matrix $\boldsymbol{D}$ 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 [25]. 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 [33, 49]. Though improvements to Dijkstra's algorithm have been developed [90, 96, 104], 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.

3.8.10 Generalised diffusion model

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 $\boldsymbol{L}_G$ for $\boldsymbol{L}$ in Eqn. 3.2.2. That is

$$\frac{d\boldsymbol{\phi}}{dt} = -\varepsilon \boldsymbol{L}_G \boldsymbol{\phi}, \quad \boldsymbol{\phi}(0) = \boldsymbol{\phi}_0, \tag{3.8.14}$$

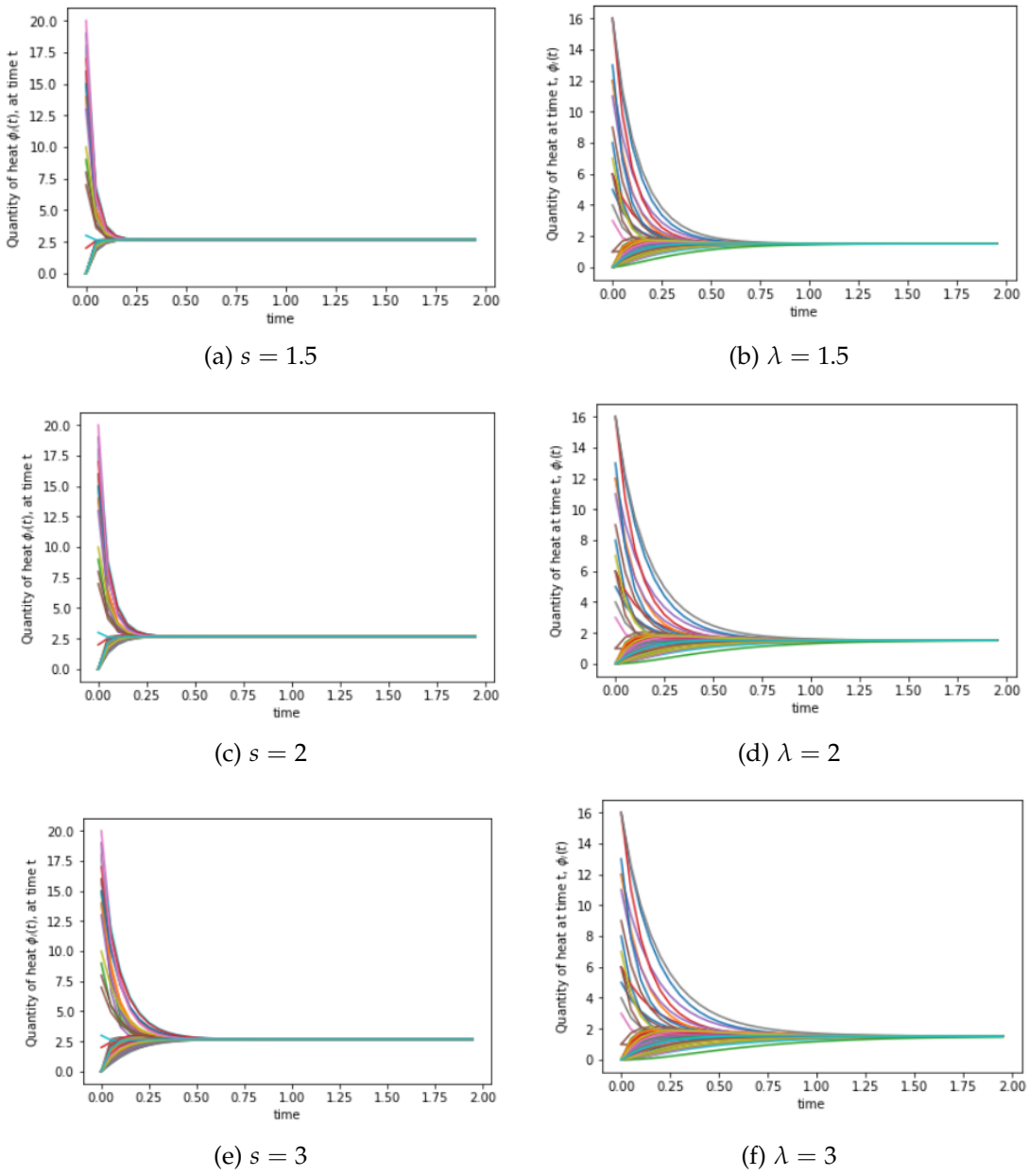
where $\boldsymbol{L}_G$ is the generalised Laplacian matrix.
We then consider generalised diffusion on graph where interactions are both short-range and long-range. The long-range interactions are accounted for by use of k-path Laplacian matrices. Following this generalisation, Eqn. 3.8.14 then becomes

$$\frac{d\boldsymbol{\phi}}{dt} = -\varepsilon \Big(\sum_{k=1}^{\Delta} c_k \boldsymbol{L}_k \Big) \boldsymbol{\phi}, \quad \boldsymbol{\phi}(0) = \boldsymbol{\phi}_0, \tag{3.8.15}$$

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.

3.8.11 Comparison of Mellin and Laplace based generalised diffusion

Here, we consider two networks of different structures that is the Erdős-Rényi[37, 65, 85] and Barabasi networks [10, 85] 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. 3.8.14 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) $\lambda = 1.5$

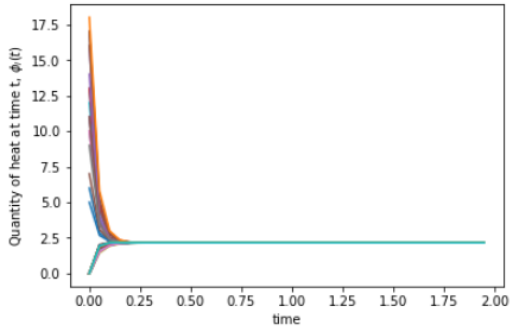
(c) $s = 2$ (d) $\lambda = 2$

(e) $s = 3$ (f) $\lambda = 3$

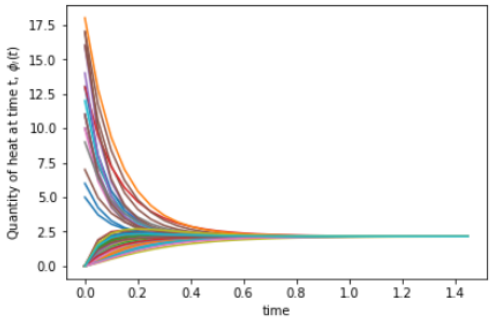
Figure 3.9: Simulations for diffusion on Barabasi network of 100 nodes for which long-range interactions are accounted for using the Mellin and Laplace transforms of the k-path Laplacian matrices using $s = \lambda = 1.5, 2$ and 3. The left column corresponds to the Mellin while the left column corresponds to the Laplace.

We observe from simulations in Fig. 3.9 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 faster in the former

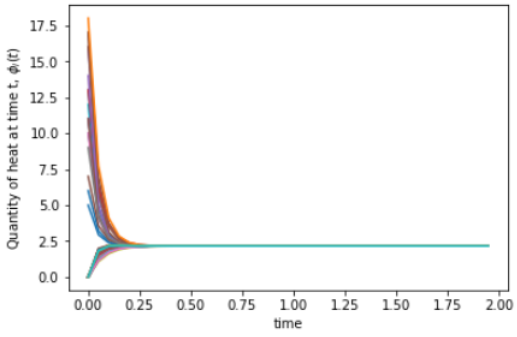
than the later. For instance, we observe that equilibrium is attained at about 0.25 and 1.0 time steps for $s = 2$ and $\lambda = 2$ respectively.



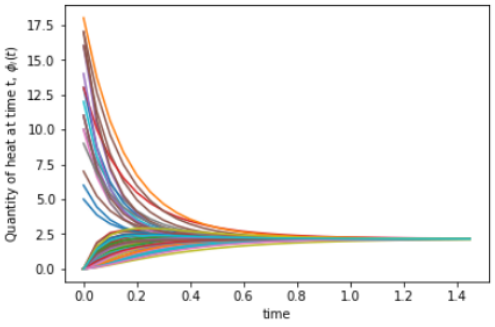
(a) $s = 1.5$



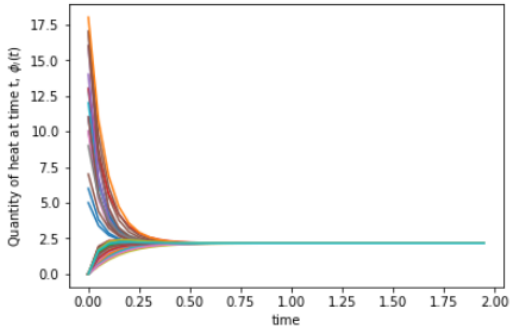
(b) $\lambda = 1.5$



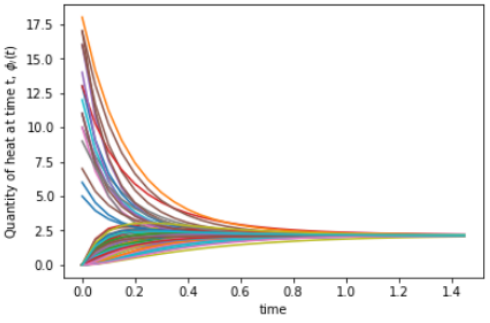
(c) $s = 2$



(d) $\lambda = 2$



(e) $s = 3$



(f) $\lambda = 3$

Figure 3.10: Simulations (performed using Eqn. 3.8.14) 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.

3.8.11.1 Simulations of diffusion on lattice

We consider a 20 by 20 2-dimensional discrete grid in which each point is connected to 8 of its nearest neighbours. Initially, we consider diffusion of heat (represented by a function ϕ) over the graph. We commence by assigning 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 3.8.8 and 3.8.9 respectively.

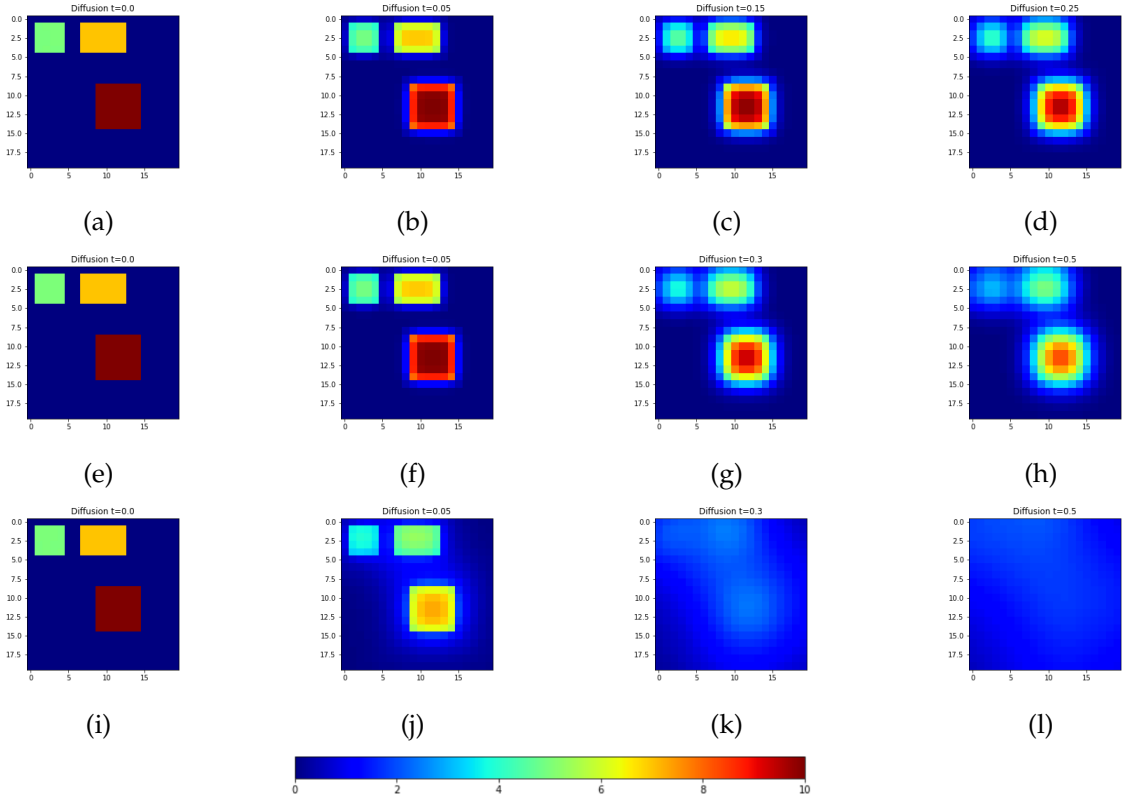


Figure 3.11: Sample illustrations for progression of diffusion over a 20×20 lattice with initial heat quantities indicated by coloured blocks. Diffusion state is captured at different time steps that is, from left to right, $t = 0$, 0.5, 3 and 5 respectively. The top row (a - d) correspond to diffusion 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 for 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 3.11l) yet for the the Laplace based case, (see Fig 3.11h), different blocks contain different amounts of heat that is, equilibrium is not yet attained.

Chapter 4

Heat kernel on graphs

4.1 Introduction

Diffusion on graphs is an important concept in the study and modelling of various systems in real world. One important aspect of diffusion is the fundamental solution of the heat equation known as the heat kernel. The heat kernel has been deeply explored by researchers despite its relevance in studying diffusion on networks. In [119], Xiao extensively studied the heat kernel and its invariants namely the heat kernel trace, zeta function, derivative of the zeta function and the heat content. He further explores the use of these invariants as a means for characterisation of graph structure for purposes of clustering. In this chapter, we will extend this work by exploring the heat kernel (which we will refer to as the generalised heat kernel) obtained from the generalised diffusion model. Moreover, we will ascertain whether these invariants can also be used for graph characterisation. Considering the Mellin and Laplace transform based generalised diffusion models, we will perform experiments on the COIL database to find which range of the Mellin or Laplace exponents do we obtain good clusters.

4.2 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 [26, 27, 69]. As discussed earlier, diffusion of heat on a graph can be modelled by the equation

$$\frac{d\phi}{dt} = -\boldsymbol{L}\phi, \tag{4.2.1}$$

where $\boldsymbol{L}$ is either the Laplacian matrix or its normalised version.
The heat kernel is the fundamental solution to the diffusion equation (4.2.1) and it is obtained by exponentiating the Laplacian eigensystem over time. It is given by

$$\boldsymbol{H}_t = e^{-t\boldsymbol{L}} \tag{4.2.2}$$

It literally describes the flow of substance (heat) across edges (direct interactions) in the graph [119]. On applying spectral decomposition to Equation 4.2.2, we have

$$\boldsymbol{H}_t = \boldsymbol{V}e^{-t\boldsymbol{\Lambda}}\boldsymbol{V}^T = \sum_{i=0}^{n} e^{-\lambda_i t}\boldsymbol{v}_i\boldsymbol{v}_i^T \tag{4.2.3}$$

where λ_i are the eigenvalues of $\boldsymbol{L}$ 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 $\boldsymbol{v}_i$ is the eigenvector corresponding to the eigenvalue λ_i [5].
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

$$\boldsymbol{H}_t(p, q) = \sum_{i=1}^{|V|} e^{-\lambda t}\boldsymbol{v}_i(p)\boldsymbol{v}_i(q) \tag{4.2.4}$$

We should note that for any two vertices p and q, $\boldsymbol{H}_t(p, q) \geq 0$ [28].
When t tends to zero, the kernel behaviour can be obtained from the Taylor's expansion of Equation 4.2.2 which is

$$e^{-t\boldsymbol{L}} = \sum_{k=0}^{\infty} \frac{(-t)^k}{k!}\boldsymbol{L}^k = \boldsymbol{I} - t\boldsymbol{L} + \frac{t^2}{2!}\boldsymbol{L}^2 + \frac{t^3}{3!}\boldsymbol{L}^3 + \cdots \tag{4.2.5}$$

Thus,

$$\lim_{t \longrightarrow 0}\left(e^{-t\boldsymbol{L}}\right) = \boldsymbol{I} - t\boldsymbol{L}. \tag{4.2.6}$$

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 4.2.3

$$\lim_{t \longrightarrow \infty}\left(e^{-t\boldsymbol{L}}\right) = e^{-\lambda_2 t}\boldsymbol{v}_2\boldsymbol{v}_2^T, \tag{4.2.7}$$

where λ_2 is the smallest non-zero eigenvalue and $\boldsymbol{v}_2$ is the corresponding eigenvector which is also known as the Fiedler vector. From Equation 4.2.7, it is evident that for large t, the heat kernel behaviour is determined by the global structure of the graph.

4.3 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

$$Tr(\boldsymbol{H}_t) = Tr(\boldsymbol{V}e^{-t\boldsymbol{\Lambda}}\boldsymbol{V}^T) = Tr(e^{-t\boldsymbol{\Lambda}}(\boldsymbol{V}^T\boldsymbol{V})) = Tr(e^{-t\boldsymbol{\Lambda}}) \tag{4.3.1}$$

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

$$Z(t) = Tr(\boldsymbol{H}_t) = \sum_p \boldsymbol{H}_t(p, p) = \sum_i e^{-\lambda_i t}, \tag{4.3.2}$$

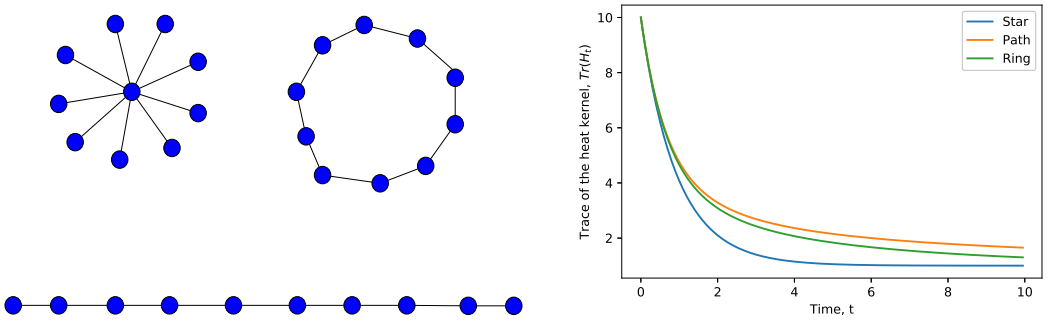
where λ_i is the eigenvalue of the normalised Laplacian matrix [119]. From Equation 4.3.2, it is evident that the trace of the heat kernel is invariant to permutations. For a connected graph, Equation 4.3.2 can be written as

$$Z(t) = 1 + e^{-\lambda_2 t} + e^{-\lambda_3 t} + \cdots + e^{-\lambda_{|V|} t} \tag{4.3.3}$$

Chung [28] 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 4.3.2). 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 $\boldsymbol{H}_t(v, v)$ is the same which makes computation of its trace much easier.

4.3.1 Heat kernel trace as a graph analysis technique

According to Xiao [119], 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 4.1 shows the plot of heat kernel trace against time for the three graphs.

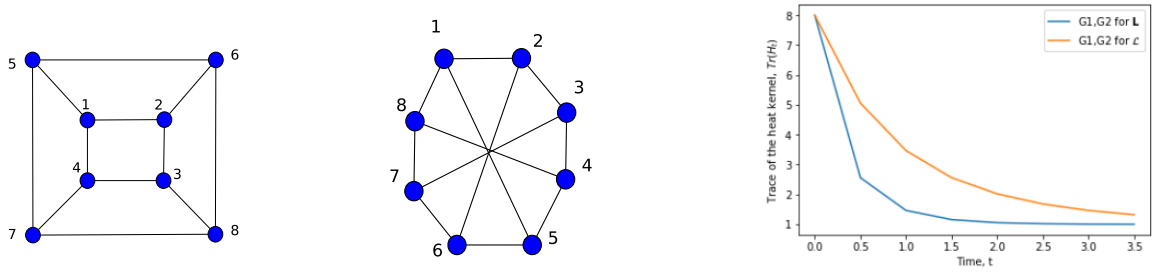


(a) (b)

Figure 4.1: (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 (4.4b), 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.(4.4b).

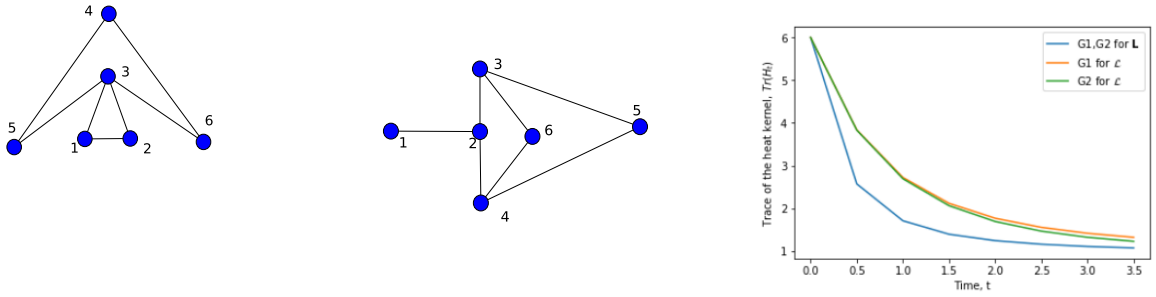
Since the trace function is computed from the eigenvalues of the Laplacian matrix $\boldsymbol{L}$. It implies that for isomorphic graphs, the curves of the trace function coincide following from the coincidence of the eigenvalues of the graphs as illustrated in Fig. 4.2.



(a) (b) (c)

Figure 4.2: (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 other hand, 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.4.3 along with the corresponding plot of the trace function of the heat kernel.



(a) (b) (c)

Figure 4.3: (a) and (b) are two co-spectral graphs with respect to $\boldsymbol{L}$. (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 [119] 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. In addition, we propose that a combination of eigenvalues and their corresponding eigenvectors would possibly give better characterisation of graphs in comparison with using only eigenvalues.

4.4 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 [40] 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 3.8.14 and it is given by

$$\boldsymbol{H}_{G_t} = e^{-t\boldsymbol{L}_G}, \tag{4.4.1}$$

where

$$\boldsymbol{L}_G = \Big(\sum_{k=1}^{\Delta} c_k \boldsymbol{L}_k \Big) = c_1\boldsymbol{L}_1 + c_2\boldsymbol{L}_2 + \cdots + c_\Delta \boldsymbol{L}_\Delta, \tag{4.4.2}$$

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. 4.4.2 can be written as

$$\boldsymbol{H}_{G_t} = e^{-t(c_1\boldsymbol{L}_1 + c_2\boldsymbol{L}_2 + \cdots + c_\Delta \boldsymbol{L}_\Delta)}, \tag{4.4.3}$$

where $\boldsymbol{L}_1, \boldsymbol{L}_2, \cdots, \boldsymbol{L}_\Delta$ 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. 4.2.2.
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.

4.5 Trace of the generalised heat kernel

The trace of the generalised heat kernel is therefore given by

$$Z_G(t) = Tr(\boldsymbol{H}_{G_t}) = \sum_{i=1}^{|V|} e^{-\mu_i t}, \tag{4.5.1}$$

where μ_i is the ith eigenvalue of the generalised Laplacian matrix. Alternatively, Eqn. 4.5.1 can be written as

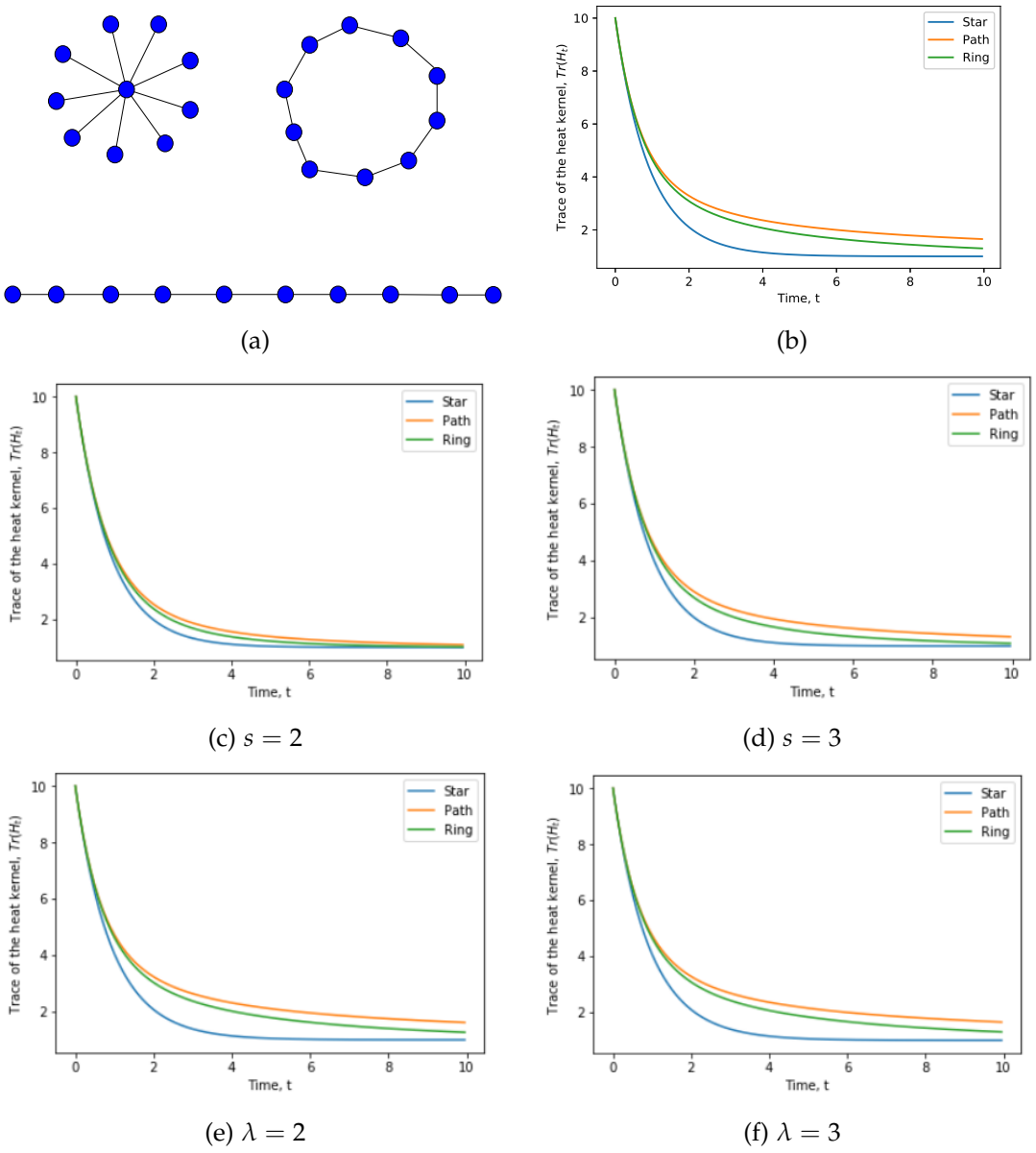
$$Z_G(t) = 1 + e^{-\mu_2 t} + e^{-\mu_3} + \cdots + e^{-\mu_N t}, \tag{4.5.2}$$

which takes on a similar format as eqn. 4.3.3 for the trace of the 'classical' Laplacian matrix. We earlier on pointed out that the multiplicity of zero as an eigenvalue of $\boldsymbol{L}_G$ is equal to the number of connected components in a given graph, it therefore follows from eqn.4.5.2 that the trace of the generalised heat kernel can also be expressed as

$$Tr(\boldsymbol{H}_{G_t}) = C + \sum_{\mu_i \neq 0} e^{-\mu_i t}, \tag{4.5.3}$$

where C is the multiplicity of zero as an eigenvalue of $\boldsymbol{L}_G$ 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. 4.4a. 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 4.4: (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. 4.4, 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. 4.4b 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.

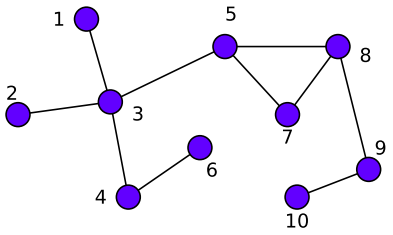
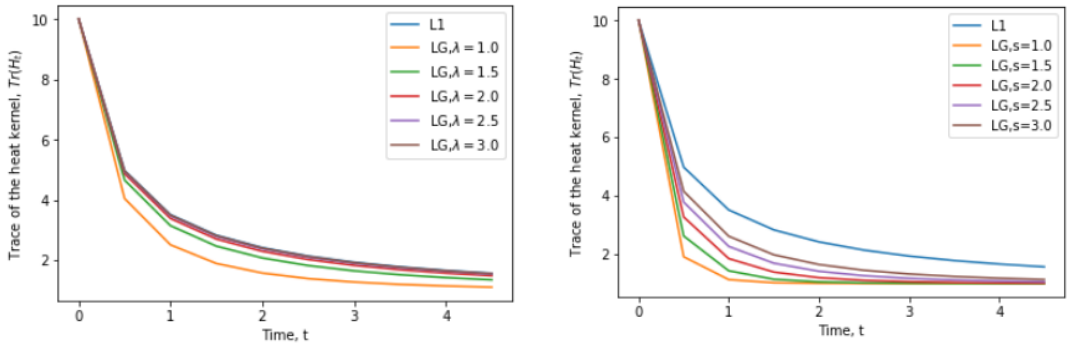


Figure 4.5: A simple network of size 10



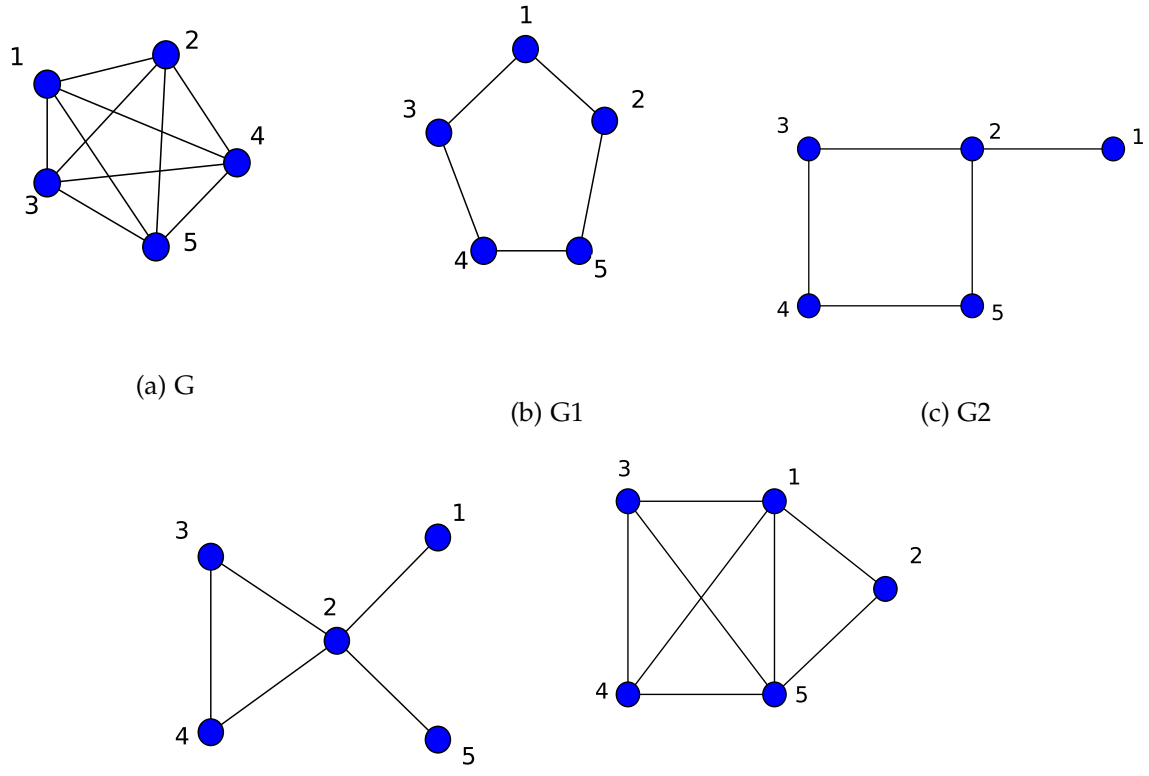
(a) (b)

Figure 4.6: Plots (performed using Eqn. 4.5.1) of the trace of the generalised heat kernel against time for the simple graph in Fig.4.5, 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.

First, we observe from the plots in Fig.4.6 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.4.6 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.

4.5.1 Comparison with complete graph based on trace of the heat kernel



(a) G (b) G1 (c) G2

(d) G3 (e) G4

Figure 4.7: 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.

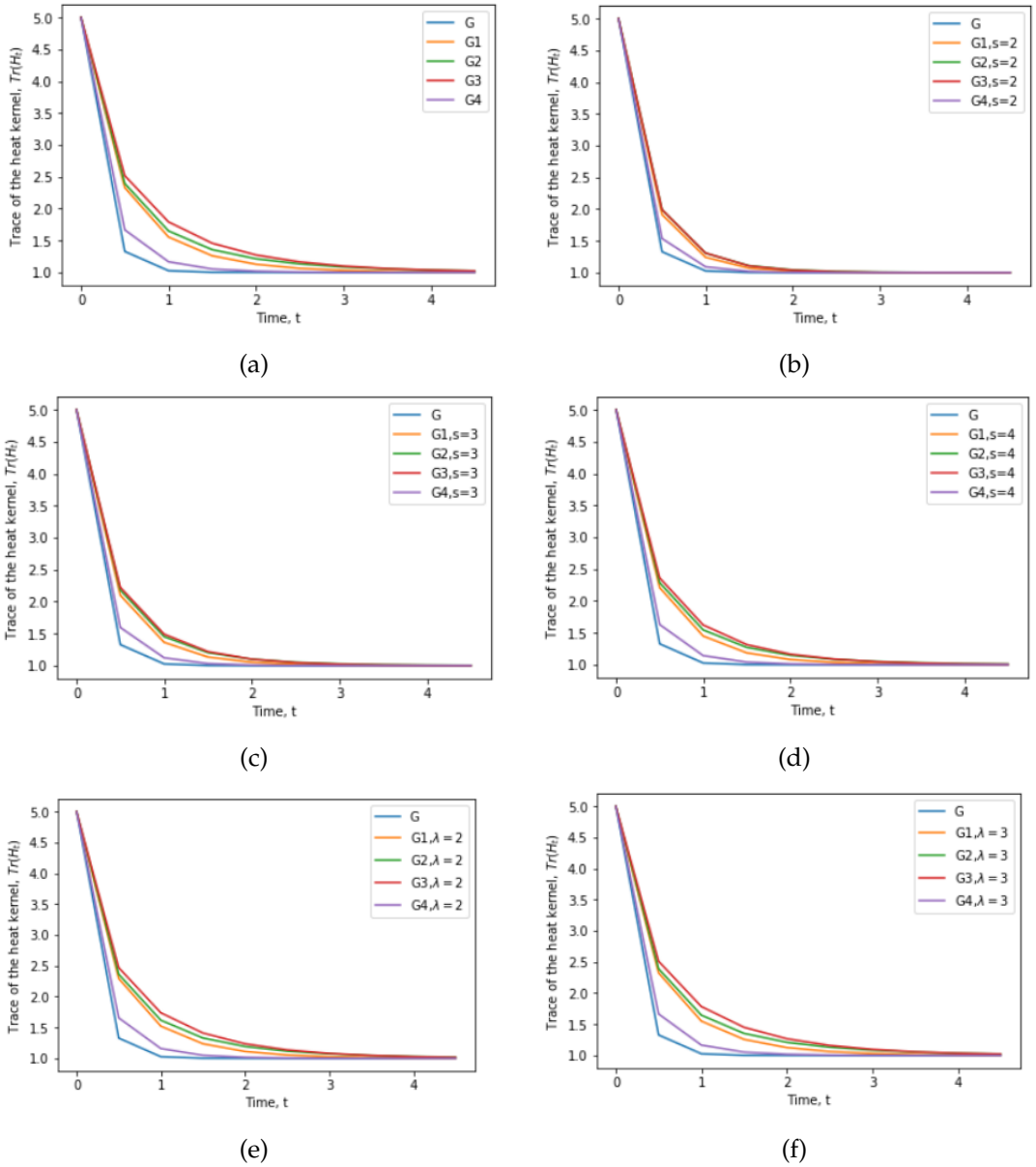


Figure 4.8: Plots of trace function of generalised heat kernel against time for graphs in Fig.4.7. From left to right and top to bottom, the plots are respectively for the normal Laplacian (a), Mellin based generalised Laplacian for $s = 2, 3$ and 4. To the right, is a plot of the trace function of the Mellin transformed Laplacian matrix at $s = 3$ against time for graphs G(blue), $G1$(orange),$G2$(green), $G3$(red), and $G4$(purple).

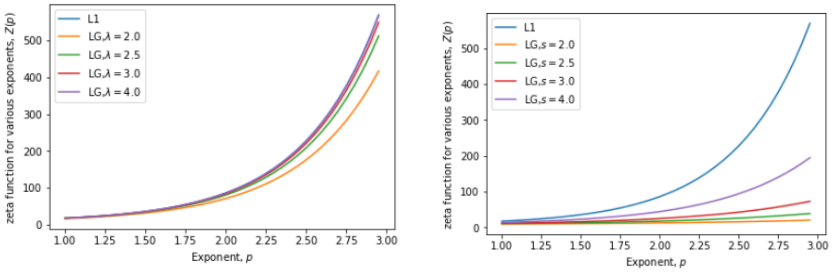
We observe from Fig.4.8a that based on the trace function curves, the process of diffusion over graph G_4 is closer to that occurring on the complete graph G. Graphs G_1, G_2 and G_3 follow by that order. Interestingly, on considering the generalised heat kernel of both the Mellin and Laplace based transforms, 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.

4.6 Zeta function

In the literature [50, 70], 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 [81, 109] as well as applicability in various aspects in differential geometry and theoretical physics. However, in this work we consider the Zeta function associated with the eigenvalues of the generalised Laplacian matrix which is obtained by exponentiating and summing the reciprocal of the non-zero Laplacian eigenvalues [50]. It is thus defined by

$$\zeta_G(p) = \sum_{\mu_i \neq 0} \mu_i^{-p}. \tag{4.6.1}$$



(a) (b)

Figure 4.9: Illustration of the Zeta function of the graph in Fig. 4.5 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.4.9a 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 the contrary, as the Mellin exponent, s, changes, there are observable changes in the corresponding curves for the zeta function against p (see Fig. 4.9b). This can be attributed to the pronounced long-range interactions evident in the Mellin-based transformed Laplacian.

4.6.1 Zeta function and generalised heat kernel trace moments

In his work [119], 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

$$\mu_i^{-p} = \frac{1}{\Gamma(p)} \int_0^\infty t^{p-1} e^{-\mu_i t} dt, \tag{4.6.2}$$

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

$$\Gamma(p) = \int_0^\infty t^{p-1} e^{-t} dt. \tag{4.6.3}$$

On summation for all non-zero eigenvalues of the Laplacian, Eqn.4.6.2 becomes

$$\zeta(p) = \sum_{\mu_i \neq 0} \mu_i^{-p} = \frac{1}{\Gamma(p)} \int_0^\infty t^{p-1} \sum_{\mu_i \neq 0} e^{-\mu_i t} dt \tag{4.6.4}$$

Using the connected component based formula for the trace of the heat kernel,that is, Eqn.4.5.3 in Eqn 4.6.4 gives

$$\zeta(p) = \frac{1}{\Gamma(p)} \int_0^\infty t^{p-1} \left\{ Tr(\boldsymbol{H}_{G_t}) - C \right\} dt. \tag{4.6.5}$$

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.

4.6.2 Derivative of the 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:

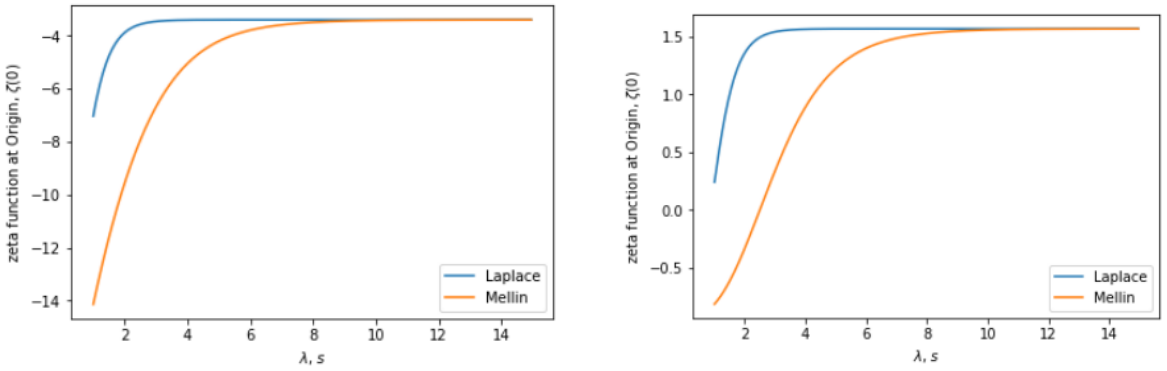
$$\zeta(p) = \sum_{\mu_i \neq 0} \mu_i^{-p} = \sum_{\mu_i \neq 0} e^{-p \ln \mu_i}, \tag{4.6.6}$$

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

$$\zeta'(p) = \sum_{\mu_i \neq 0} \{ - \ln \mu_i \} e^{-p \ln \mu_i} \tag{4.6.7}$$

so, the derivative at the origin is

$$\zeta'(0) = - \sum_{\mu_i \neq 0} \ln \mu_i \tag{4.6.8}$$



(a) (b)

Figure 4.10: Derivative of Zeta function against time for the graph in Fig.4.5. (a) corresponds to the plot for the normal Laplacian $\boldsymbol{L}$ matrix of the graph. (b) corresponds to the plot for which the eigenvalues of the normalised Laplacian $\mathcal{L}$ is used in Fig. 4.6.8.

To start with we compute the values of the zeta function for graph (Fig.4.5) 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 (4.10a) and (4.10b)) that of the normal diffusion with no long range interactions compared to the former case.

4.6.3 Heat content

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

$$Q(t) = \sum_{p \in V} \sum_{q \in V} \boldsymbol{H}_{G_t}(p, q) \tag{4.6.9}$$

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.4.4.3, 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 [119].

On substituting for $\boldsymbol{H}_{G_t}(p, q)$ in Eqn.4.6.9 gives

$$Q(t) = \sum_{p \in V} \sum_{q \in V} \sum_{k=1}^{|V|} e^{(-\mu_k t)} v_k(p) v_k(q), \tag{4.6.10}$$

which can be expanded into a polynomial in time as in [75]

$$Q(t) = \sum_{m=0}^{\infty} q_m t^m, \tag{4.6.11}$$

where q_m is given by

$$q_m = \sum_{k=1}^{|V|} \left\{ \left(\sum_{p \in V} v_k(p) \right)^2 \right\} \frac{(-\lambda_k)^m}{m} \tag{4.6.12}$$

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.

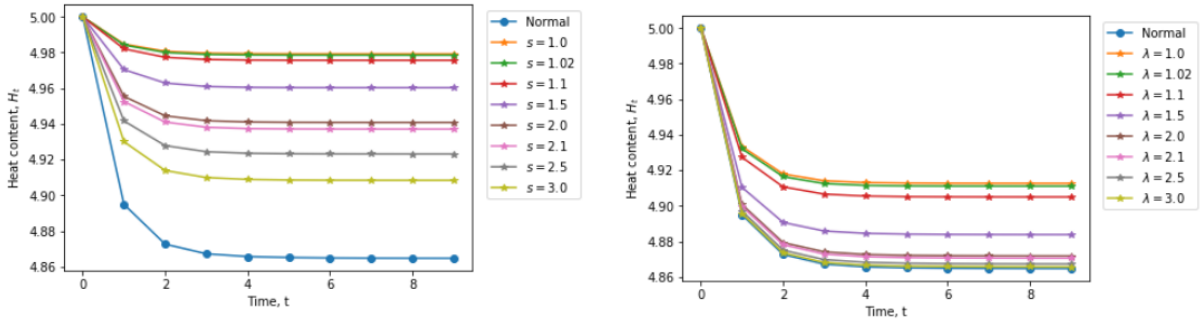
Alternatively, Eqn. 4.6.10 can be written as

$$Q(t) = \sum_{i=1}^{m} \alpha_i e^{-\lambda t}, \tag{4.6.13}$$

where $\alpha_i = \sum_{p \in V} \sum_{q \in V} v_k(p) v_k(q)$. We can see from Eqn 4.6.13 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.

4.6.4 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 4.11: Simulations for heat content against time for the graph in Fig.4.7c. (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 $\boldsymbol{H}_{G_0} = \boldsymbol{I}$ thus the heat content is equal to the trace of $\boldsymbol{I}$ which is in turn equal to the number of vertices, $|V|$, in the graph, thus for graph in Fig.4.7c, $Q(0) = 5$.

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.

4.7 Graph characterisation of COIL database using heat kernel invariants

4.7.1 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 unweighted Delaunay graph.
The Delaunay graph is obtained from Delaunay triangulation of the corner points (using Harris corner detection) of the objects as discussed 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.

We further demonstrate the extraction of Delaunay graphs of objects 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, different images were taken at every 5 degree turn up to 360 making a total of 72 images per object.
In our experiments, we consider 8 objects and their corresponding images at different views. In order to extract the graph corresponding to each image, we implement the algorithm explained in the previous subsection using a code in Matlab onto the selected objects. The extracted Delaunay graphs are then superimposed onto the objects as shown in Fig.4.12.

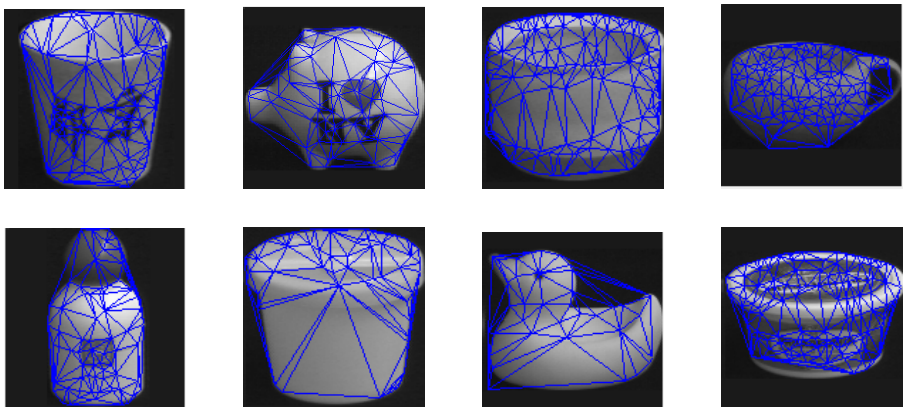


Figure 4.12: Illustration of 8 selected objects (Left to right: top row(1-4) and bottom row (5-8)) from the COIL-100 database with their Delaunay graphs superimposed.

4.7.2 Zeta function Experiments

Here, we explore the use of the zeta function (Eqn. 4.6.1) as a means of characterising the structure of a graph. For the selected 8 objects from the COIL database, we obtain graphs for different images of each object using the method explained in the previous subsection. First, we consider the zeta function associated with the non-zero eigenvalues of the normalised Laplacian matrix for each of the graphs corresponding to different views of each object. Second, we look at experiments that involve the zeta function at integer argument 1 associated with the eigenvalues of generalised normalised Laplacian matrix which is obtained by either the Mellin or Laplace transforms of the path Laplacian matrices. The purpose of these experiments is to ascertain whether the zeta function related to the eigenvalues of the generalised Laplacian can also be used for structural characterisation of graphs and for what range of values of Mellin exponent,s, or Laplace exponent,λ, can the characterisation be possible.

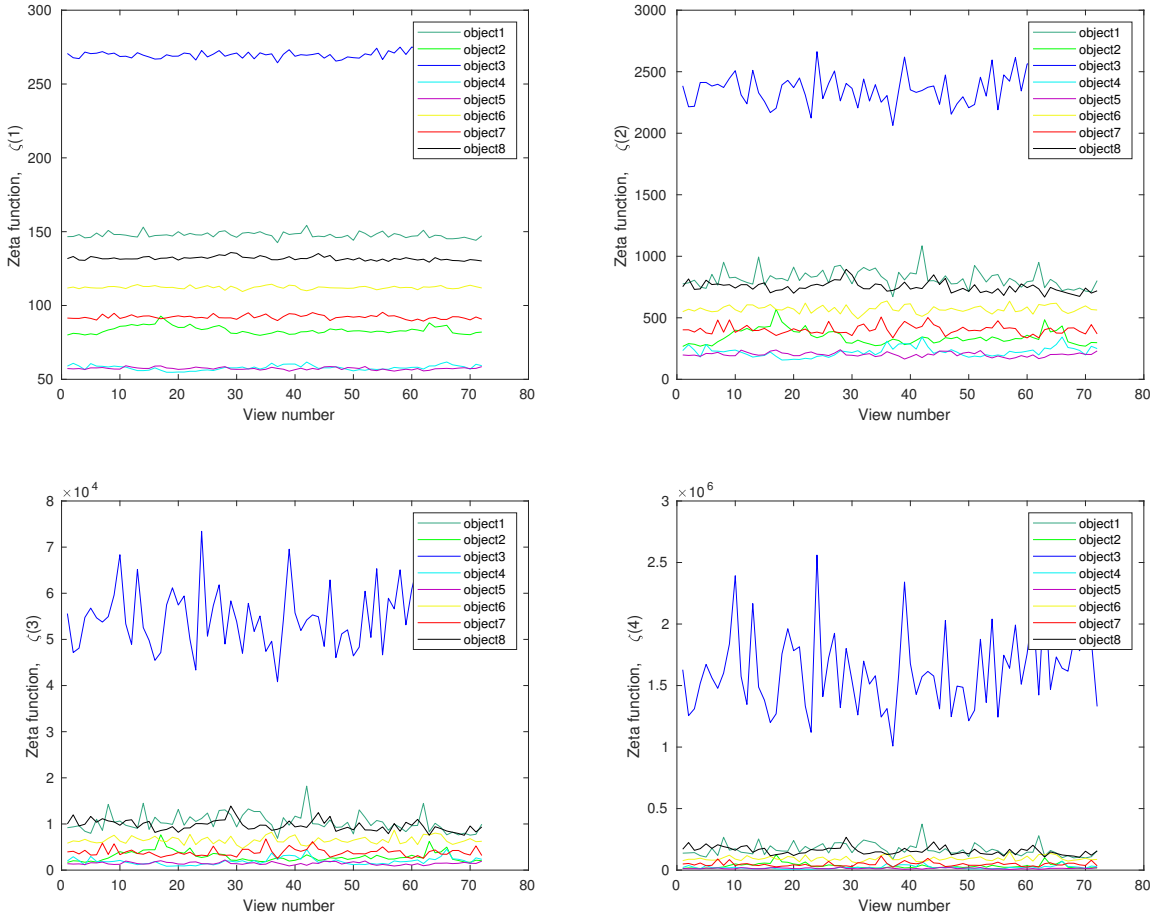


Figure 4.13: Zeta function $\zeta(p)$ associated with the normalised Laplacian eigenvalues with view number. (from left to right, and top to bottom, p = 1,2,3 and 4 respectively

In Fig. 4.13,the different curves correspond to different objects in the four plots. Most importantly, we can observe that the curves for different objects are well separated. Generally, we

observe no significant variation in the zeta function for different views of each of the object. However, to be specific we can tell that the fluctuation in the zeta functions for different objects becomes pronounced as the integer input increases. That is to say, for $p = 1$ and $p = 2$, the fluctuations are smaller. However, for $p = 4$, the fluctuations are larger especially for object 3 whose graph has relatively more number of nodes compared to the other objects. The explanation to the difference in size of the fluctuations is due to the dominant role played by the small eigenvalues when p becomes large.

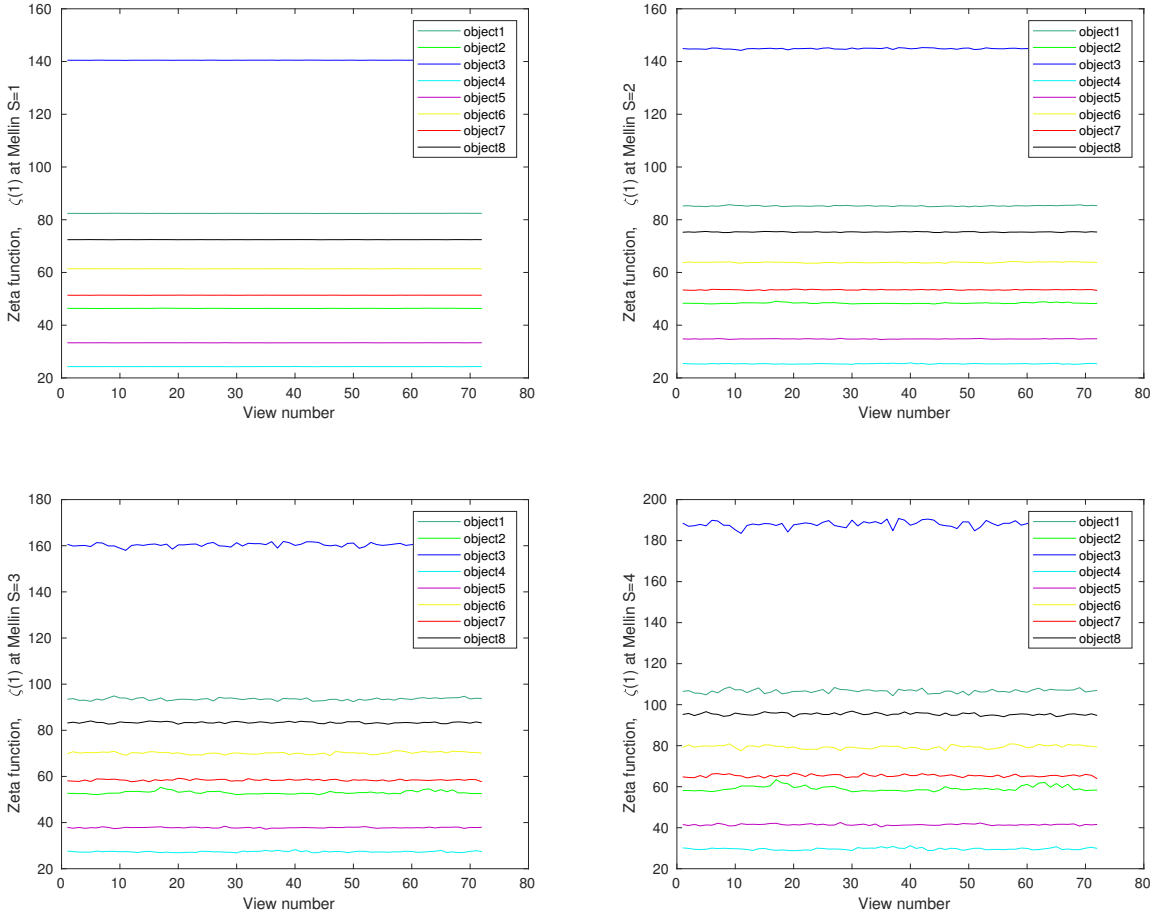


Figure 4.14: Zeta function $\zeta(p)$ (at $p = 1$) associated with the Mellin transform-based generalised normalised Laplacian eigenvalues with view number. (from left to right, and top to bottom, $s = 1$, 2, 3, and 4 respectively).

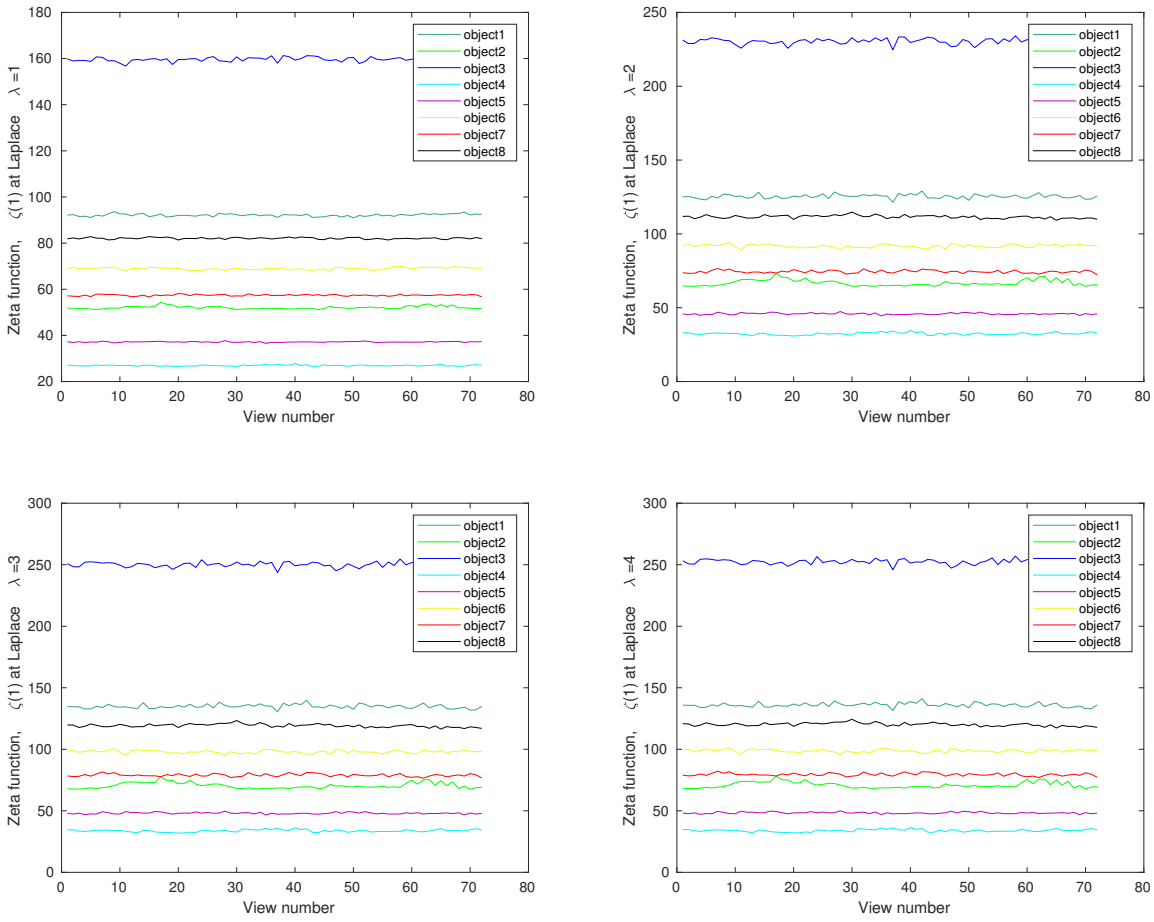
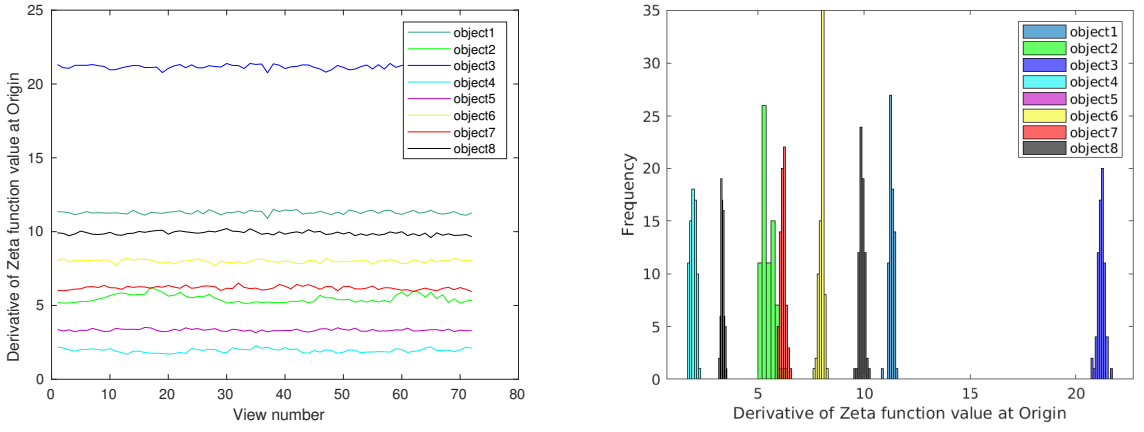


Figure 4.15: Zeta function $\zeta(1)$ (at $p = 1$) associated with the Laplace transform-based generalised normalised Laplacian eigenvalues with view number. (from left to right, and top to bottom, $\lambda = 1$, 2, 3, and 4 respectively).

4.7.3 Zeta function at origin



4.8 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 respectively. The idea behind graph-based techniques is to interpret the concept of interest as a graph theory concept for instance object classification which is an essential aspect in computer vision and pattern recognition can be viewed as a graph clustering concept on using graph representation of the objects. Though great progress has been attained in regards to related areas such as graph similarity measurements and inexact matching of graphs, little has been attained regarding the challenge of graph clustering which is quite an important concept with applications in unveiling the view structure of objects, organisation of large structural databases and many others. One major challenge is that graphs are not vectorial in nature and more so the task of transforming graphs into vectors is not an easy one because of the following reasons: First, there is non natural way of mapping nodes or edges of a graph into components of vectors as there exists no standard ordering of nodes or edges of graphs. Second, suppose we find an ordering of nodes or edges, vectors of different lengths can be obtained for graphs with different number of nodes and edges. This, thus, calls for a method of dealing with pattern-vectors of different lengths. The difficulty associated with vectorising graphs makes it practically impossible to adopt clustering methods which involve pairwise similarity measurements. Luo et. al. [73] attempted to overcome this problem by using the spectral representation of graphs in which the structure of a graph is mapped onto a vector of fixed length. The pattern vector was developed based on the adjacency matrix of a graph where the content of the vectors are unary features such as leading eigenvalues, eigenmode volume, eigenmode perimeter and Cheeger constant. Alternatively the vectors can be composed of binary features which are pairwise attributes of eigenmodes and these include mode association matrix and inter-node distances. Shortly after, Xiao [119] introduced mapping of graphs onto pattern vectors using the spectrum of the Laplacian matrix as well as other invariants of the heat kernel such as the zeta function and heat content co-efficients. We further explore this concept by considering the spectrum of the generalised Laplacian matrix and the invariants associated with the generalised heat kernel.

4.8.1 Principal Component Analysis (PCA) on images

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 [35] 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 $\boldsymbol{B}_k$ 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 $\boldsymbol{B}_k = (q_1, q_2, \cdots, q_n)^T$ or from the n leading Laplacian eigenvalues: $\boldsymbol{B}_k = (l_1, l_2, \cdots, l_n)^T$.

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

v) We then compute the matrix, $\hat{\boldsymbol{S}}$, by subtracting the mean of the feature vectors from each of the column of the matrix $\boldsymbol{S}$.

vi) We follow this with computation of the covariance matrix $\boldsymbol{C}$ by taking the matrix product $\boldsymbol{C} = \hat{\boldsymbol{S}}\hat{\boldsymbol{S}}^T$.

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

$$\boldsymbol{C} = \sum_{i=1}^{m} \lambda_i \boldsymbol{v}_i \boldsymbol{v}_i^T, \tag{4.8.1}$$

where λ_i are the eigenvalues and $\boldsymbol{v}_i$ 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 $\boldsymbol{V} = (\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_s)$.

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

4.8.2 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.

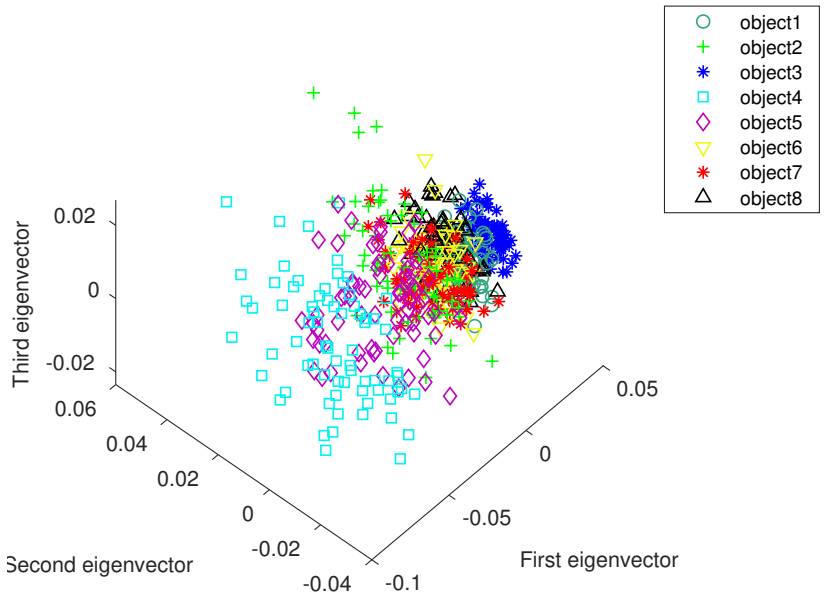


Figure 4.17: Clustering using PCA with feature vector composed of the 6 leading eigenvalues of the graph Laplacian matrix for images of objects. The 3D illustration consists of the 3 principal components as axes.

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.

4.8.3 Pattern vectors from symmetric polynomials

Symmetric polynomials are polynomials on a set of variables that are invariant under permutation of variable indices. In other words, on interchange of one or more variables, one obtains the same polynomial. Here, we discuss two kinds of symmetric polynomials namely the elementary and power symmetric polynomials.
For a given set of n variables $\{v_1, \cdots, v_n\}$, the set of elementary symmetric polynomials are

given by

$$\begin{aligned} S_1(v_1,\ldots,v_n) &= \sum_{i=1}^{n} v_i \\ S_2(v_1,\ldots,v_n) &= \sum_{i=1}^{n}\sum_{j=i+1}^{n} v_i v_j \\ &\vdots \\ S_r(v_1,\cdots,v_n) &= \sum_{i_1<i_2<\cdots i_r} \\ &\vdots \\ S_n(v_1,\ldots,v_n) &= \prod_{i=1}^{n} v_i \end{aligned}$$

On the other hand, for the same set of variables, the set of power symmetric polynomials are defined as

$$\begin{aligned} P_1(v_1,\ldots,v_n) &= \sum_{i=1}^{n} v_i \\ P_2(v_1,\ldots,v_n) &= \sum_{i=1}^{n} v_i^2 \\ &\vdots \\ P_r(v_1,\ldots,v_n) &= \sum_{i=1}^{n} v_i^r \\ &\vdots \\ P_n(v_1,\cdots,v_n) &= \sum_{i=1}^{n} v_i^n \end{aligned}$$

The two polynomials are related to each other by the Newton-Girard formula as

$$S_r = \frac{(-1)^{r+1}}{r}\sum_{k=1}^{r}(-1)^{k+r}P_k S_{r-k}, \tag{4.8.2}$$

where S_r is shortform for $S_r(v_1,\ldots,v_n)$ as P_r is for $P_r(v_1,\ldots,v_n)$. This relation provides an efficient method of computing elementary symmetric polynomials using power symmetric polynomials.

Symmetric polynomials have proved useful in constructing pattern vectors that can be used for graph analysis. This is due to the permutation invariant property of these polynomials. For graph clustering purposes, the set of variables to these polynomials can be the n leading eigenvalues of the normalised Laplacian [119], the zeta function for different non-integer arguments [119], the elements of the spectral matrix [116], and many others. Experiments for graph clustering using pattern vectors obtained from symmetric polynomials demonstrate

results of good clustering as discussed in [116].

From the two definitions of the set of symmetric polynomials , we observe when the polynomials take as its arguments the eigenvalues of the normalised Laplacian. The following relations hold:

1. The lowest order elementary symmetric polynomial is the sum of the eigenvalues that is, the trace of the normalised Laplacian.

2. The highest order elementary symmetric polynomial is the product of non-zero eigenvalues of the normalised Laplacian and thus related to the derivative of the zeta function at the origin by

$$S_{|V|}(\lambda_1, \lambda_2, \ldots, \lambda_{|V|}) = e^{[}(\zeta'(0))^{-1}] \tag{4.8.3}$$

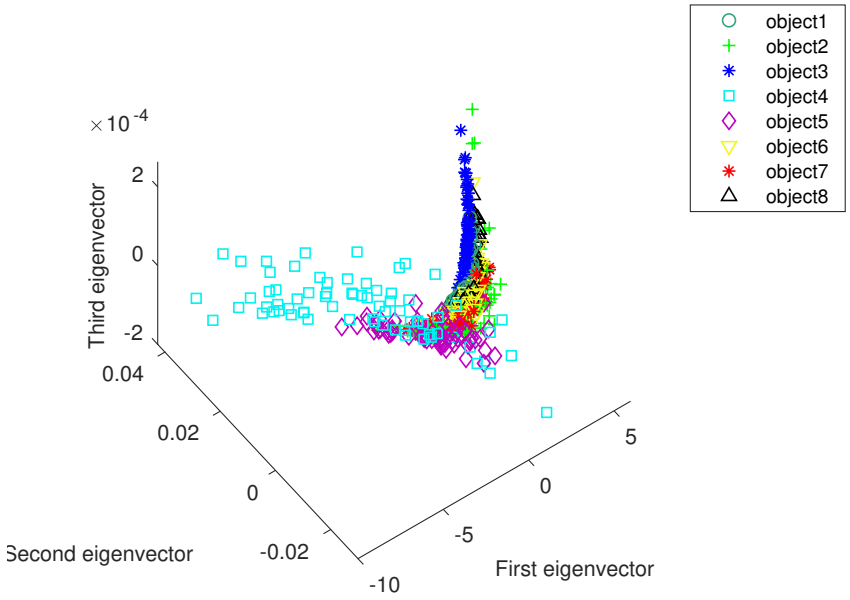


Figure 4.18: Clustering based on elementary symmetrical polynomial with the 6 leading eigenvalues of the normalised Laplacian matrix as variables.

4.8.4 Clustering using Zeta function

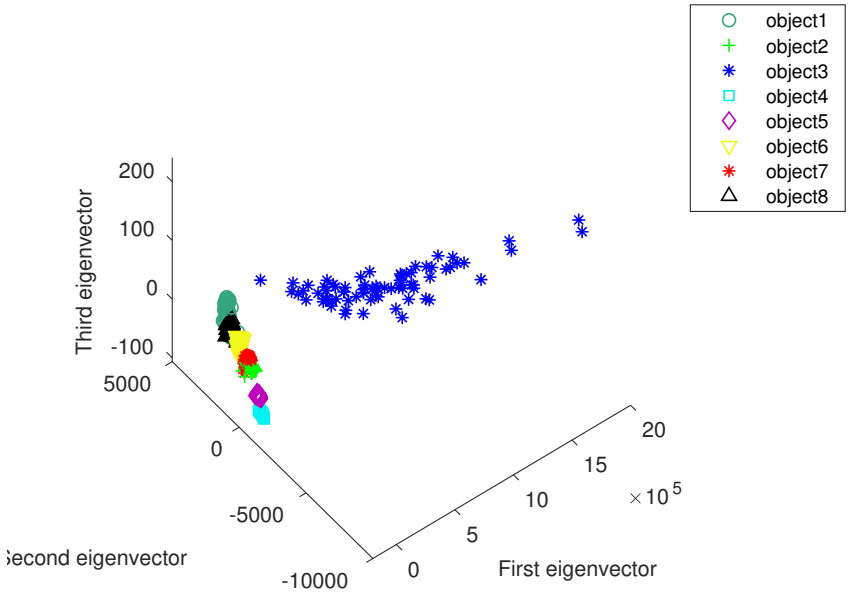
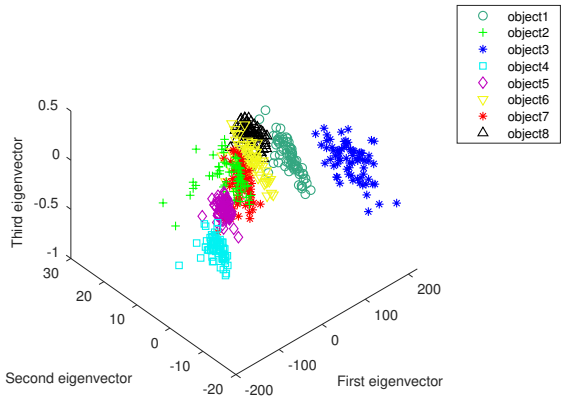
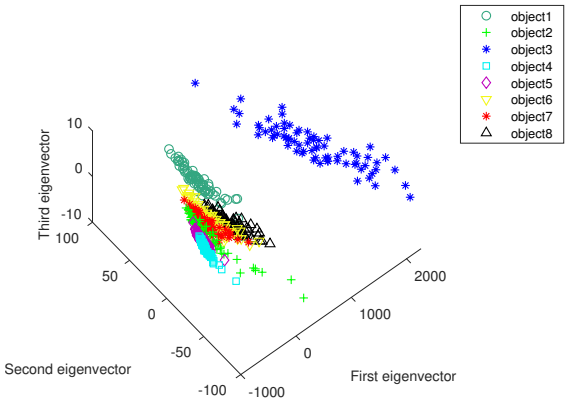


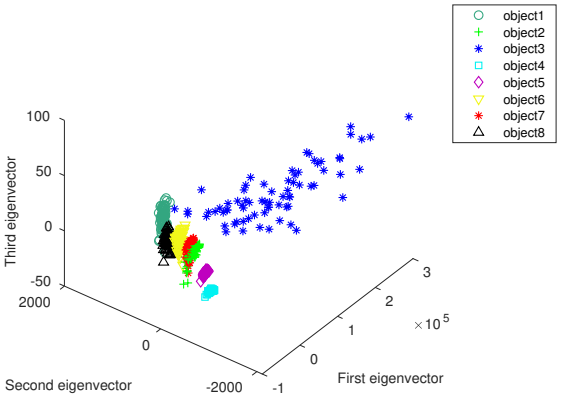
Figure 4.19: Clustering based on zeta function for 4 different integer arguments, that is $1, 2, 3$ and 4.



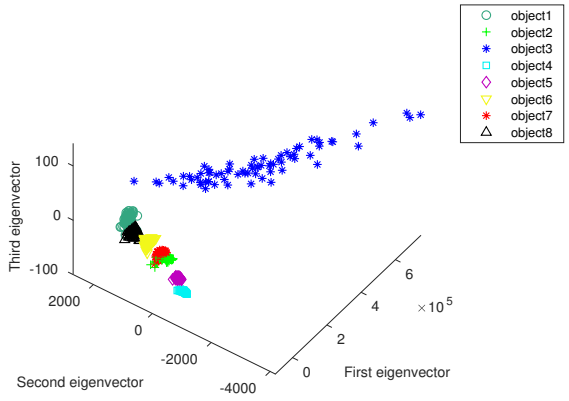
(a) $s = 2$



(b) $s = 3$

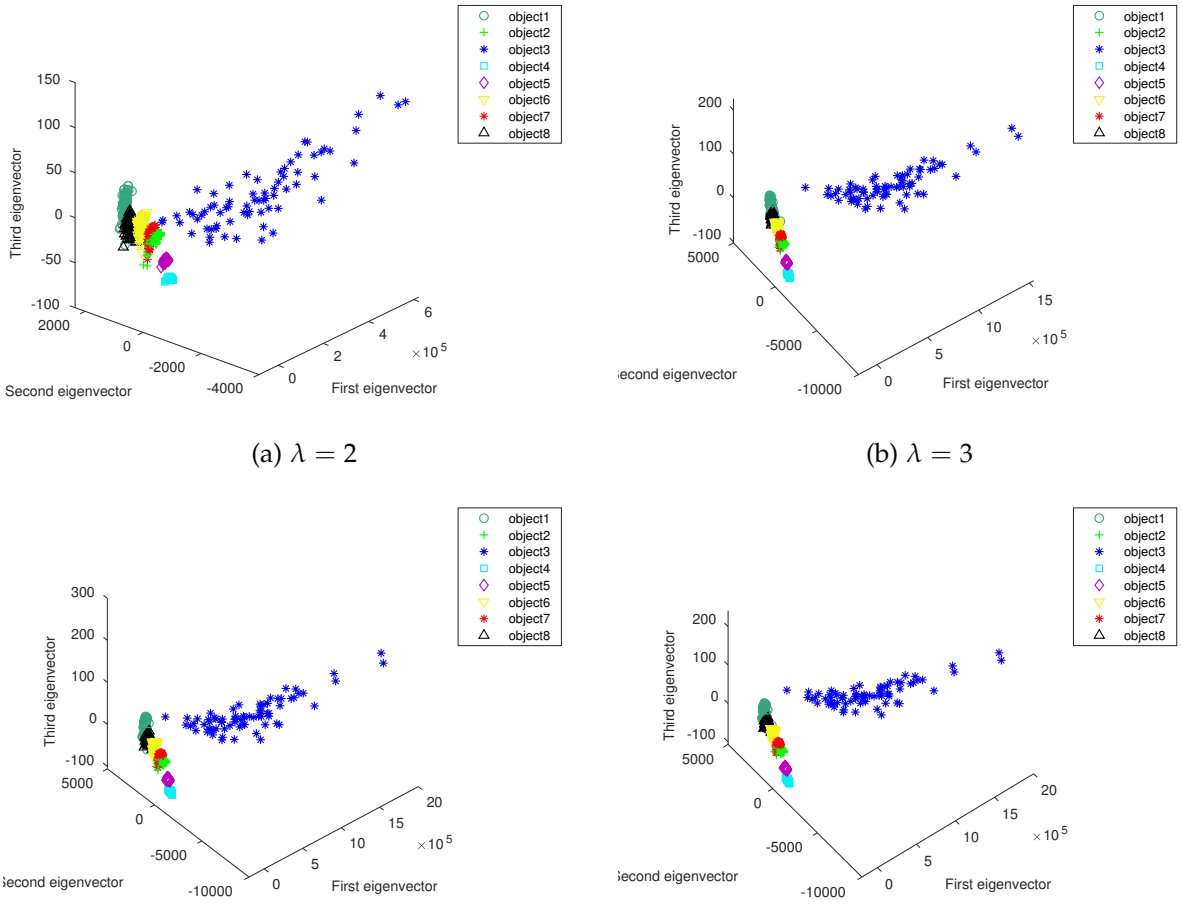


(c) $s = 5$



(d) $s = 6$

Figure 4.20: Illustration of PCA based clustering for 8 selected objects of the COIL-100 database. The feature vector consist of the zeta function on 4 arguments (1 to 4) of the Normalised 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 4.21: Illustration of PCA based clustering for 8 selected objects of the COIL-100 database. The feature vector consist of the zeta function on 4 arguments (1 to 4) of the Normalised 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.

4.8.5 Heat content polynomials

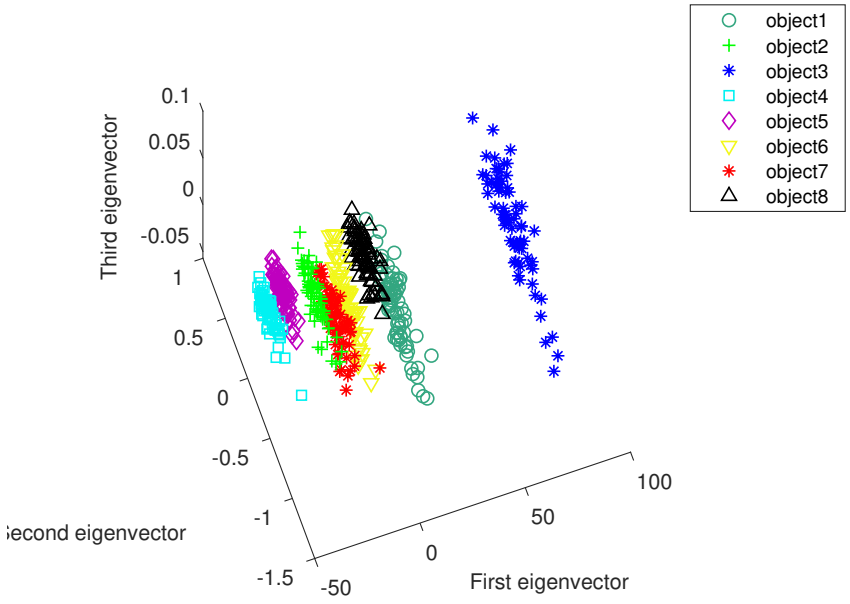


Figure 4.22: Clustering with feature vectors consisting of four leading coefficients (q_1, q_2, q_3, q_4) of the heat content polynomial.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

5.2 Directions for future research

In this section, we discuss some of the topics that we consider open for future research.
Centrality is a measure of importance of a node or an edge in a graph. However, the definition of 'importance' is quite versatile as determined by the application. Some of the most common centralities include degree centrality (importance based on degree of the node), closeness centrality (measure of proximity of a node to other nodes in the network), betweenness centrality (role played by a node as a bridge) and Laplacian centrality (drop in Laplacian energy on node removal) [91]. Other centrality measures based on the number of walks include subgraph and total communicability centralities [45]. Centrality has numerous applications in real-world such as control of spread of diseases with in a social network, prevention of virus spread in a computer network, designing robust technological systems, faster dissemination of information with in a social network, and many others.
In this section, we focus on building onto the existing research about centrality measures by exploring measures based on heat kernel and k-path communicability among nodes of a network.

5.2.1 Heat kernel centrality

As we have discussed in the previous chapters, the heat kernel is of great importance in the study of diffusion on networks as it describes the flow of substance say information, heat, disease, etc. within the network. We find it intriguing to develop a new measure of centrality where the importance of a particular node or edge is measured by the role it plays as indicated by the heat kernel of the network.

5.2.2 k-based communicability centrality

List of References

[1] Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1):47, 2002.

[2] Michael O Albertson. The irregularity of a graph. *Ars Combinatoria*, 46:219–225, 1997.

[3] Uri Alon. Biological networks: the tinkerer as an engineer. *Science*, 301(5641):1866–1867, 2003.

[4] Luıs A Nunes Amaral, Antonio Scala, Marc Barthelemy, and H Eugene Stanley. Classes of small-world networks. *Proceedings of the national academy of sciences*, 97(21):11149–11152, 2000.

[5] Howard Anton and Chris Rorres. Elementary linear algebra, (2000). *Anton Textbook Inc, Ottawa*, 2007.

[6] Franz Aurenhammer and Rolf Klein. Voronoi diagrams. *Handbook of computational geometry*, 5:201–290, 2000.

[7] Jayanth R Banavar, Amos Maritan, and Andrea Rinaldo. Size and form in efficient transportation networks. *Nature*, 399(6732):130, 1999.

[8] Albert-László Barabási. Linked: The new science of networks, 2003.

[9] Albert-László Barabási. *Network science*. Cambridge university press, 2016.

[10] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

[11] John S Baras and Pedram 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.

[12] Alain Barrat, Marc Barthelemy, and Alessandro Vespignani. *Dynamical processes on complex networks*. Cambridge university press, 2008.

[13] Ehrhard Behrends. *Introduction to Markov chains*, volume 228. Springer, 2000.

[14] Francis K Bell. A note on the irregularity of graphs. *Linear Algebra and its Applications*, 161:45–54, 1992.

[15] Azer Bestavros. Discussion on eulerian circuits. [Online; accessed 2017-03-29].

[16] Norman Biggs. *Algebraic graph theory*. Cambridge university press, 1993.

[17] Leslie P Boss. Epidemic hysteria: a review of the published literature. *Epidemiologic Reviews*, 19(2):233–243, 1997.

[18] S Burcu Bozkurt, A Dilek Güngör, Ivan Gutman, and A Sinan Cevik. Randic matrix and Randic energy. *MATCH Commun. Math. Comput. Chem*, 64:239–250, 2010.

[19] Dennis Bray. Molecular networks: the top-down view. *Science*, 301(5641):1864, 2003.

[20] Anna D Broido and Aaron Clauset. Scale-free networks are rare. *arXiv preprint arXiv:1801.03400*, 2018.

[21] Andries E Brouwer and Willem H Haemers. *Spectra of graphs*. Springer Science & Business Media, 2011.

[22] Alexander Crum Brown and Thomas 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.

[23] John L. Casti. MS Windows NT complexity, September 26, 2017.

[24] MB Chandak, S Bhalotia, and SC Agrawal. A novel approach to compute steiner point in graph: Application for network design. 2017.

[25] Boris V Cherkassky, Andrew V Goldberg, and Tomasz Radzik. Shortest paths algorithms: Theory and experimental evaluation. *Mathematical programming*, 73(2):129–174, 1996.

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

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

[28] Fan RK Chung. *Spectral graph theory*. Number 92. American Mathematical Soc., 1997.

[29] David Cohen. All the world's a net. *New Scientist*, 174(2338):24–9, 2002.

[30] Ewan R Colman and Geoff J Rodgers. Complex scale-free networks with tunable power-law exponent and clustering. *Physica A: Statistical Mechanics and its Applications*, 392(21):5501–5510, 2013.

[31] DM Cvetkovic and PETER Rowlinson. Spectral graph theory. *Topics in algebraic graph theory*, pages 88–112, 2004.

[32] K Ch Das. The Laplacian spectrum of a graph. *Computers & Mathematics with Applications*, 48(5):715–724, 2004.

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

[34] SIMENA Dinas and JOSÉ MARÍA Banon. A review on delaunay triangulation with application on computer vision. *Int. J. Comp. Sci. Eng*, 3:9–18, 2014.

[35] Chris Ding and Xiaofeng He. K-means clustering via principal component analysis. In *Proceedings of the twenty-first international conference on Machine learning*, page 29. ACM, 2004.

[36] Wendy Ellens and Robert E Kooij. Graph measures and network robustness. *arXiv preprint arXiv:1311.5064*, 2013.

[37] Paul Erdös and Alfréd Rényi. On random graphs. *Publicationes Mathematicae (Debrecen)*, 6:290–297, 1959.

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

[39] Ernesto Estrada. *The structure of complex networks: theory and applications*. OUP Oxford, 2011.

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

[41] Ernesto Estrada, Lucia Valentina Gambuzza, and Mattia Frasca. Long-range interactions and network synchronization. *arXiv preprint arXiv:1704.01349*, 2017.

[42] Ernesto Estrada, Ehsan Hameed, Naomichi Hatano, and Matthias Langer. Path Laplacian operators and superdiffusive processes ongraphs. i. one-dimensional case. *Linear Algebra and its Applications*, 523:307–334, 2017.

[43] Ernesto Estrada, Franck Kalala-Mutombo, and Alba Valverde-Colmeiro. Epidemic spreading in networks with nonrandom long-range interactions. *Physical Review E*, 84(3):036110, 2011.

[44] Ernesto Estrada, Philip Knight, et al. *A first course in network theory*. Oxford University Press, USA, 2015.

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

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

[47] Leonhard Euler. Leonhard euler and the Königsberg bridges. *Scientific American*, 189(1):66–70, 1953.

[48] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the internet topology. In *ACM SIGCOMM computer communication review*, volume 29, pages 251–262. ACM, 1999.

[49] Michael L Fredman and Robert Endre Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of the ACM (JACM)*, 34(3):596–615, 1987.

[50] Fabien Friedli, Anders 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.

[51] Xinbo Gao, Bing Xiao, Dacheng Tao, and Xuelong Li. A survey of graph edit distance. *Pattern Analysis and applications*, 13(1):113–129, 2010.

[52] Chris Godsil and G Royle. Algebraic graph theory springer. *New York*, 2001.

[53] Ivan Gutman. The energy of a graph: old and new results. In *Algebraic combinatorics and applications*, pages 196–211. Springer, 2001.

[54] Ivan Gutman, Dariush Kiani, Maryam Mirzakhah, and Bo Zhou. On incidence energy of a graph. *Linear Algebra and its Applications*, 431(8):1223–1233, 2009.

[55] Ivan Gutman and Oskar E Polansky. *Mathematical concepts in organic chemistry*. Springer Science & Business Media, 2012.

[56] Ivan Gutman and Bo Zhou. Laplacian energy of a graph. *Linear Algebra and its applications*, 414(1):29–37, 2006.

[57] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, pages 10–5244. Citeseer, 1988.

[58] John Michael Harris, Jeffry L Hirst, and Michael J Mossinghoff. *Combinatorics and graph theory*, volume 2. Springer, 2008.

[59] Stephen Hawking. San Jose Mercury News, 2000. [].

[60] Bernardo A Huberman. The laws of the web, 2001.

[61] Matthew O Jackson. *Social and economic networks*. Princeton university press, 2010.

[62] Barbara R Jasny and L Bryan Ray. Life and the art of networks, 2003.

[63] Glen Jeh and Jennifer 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.

[64] Lili Ju, Todd Ringler, and Max Gunzburger. Voronoi tessellations and their application to climate and global modeling. In *Numerical techniques for global atmospheric models*, pages 313–342. Springer, 2011.

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

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

[67] Erica Klarreich. Scant evidence of power laws found in real- world networks, February 15, 2018.

[68] Jon M Kleinberg. Navigation in a small world. *Nature*, 406(6798):845, 2000.

[69] Kyle Kloster and David 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.

[70] Oliver Knill. The zeta function for circular graphs. *arXiv preprint arXiv:1312.4239*, 2013.

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

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

[73] Bin Luo, Richard C Wilson, and Edwin R Hancock. Spectral clustering of graphs. In *International Workshop on Graph-Based Representations in Pattern Recognition*, pages 190–201. Springer, 2003.

[74] Justin Magouirk, Scott Atran, and Marc Sageman. Connecting terrorist networks. *Studies in Conflict & Terrorism*, 31(1):1–16, 2008.

[75] Patrick McDonald and Robert Meyers. Diffusions on graphs, Poisson problems and spectral geometry. *Transactions of the American Mathematical Society*, 354(12):5111–5136, 2002.

[76] Sergey Melnik, Hector Garcia-Molina, and Erhard 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.

[77] Zlatko Mihalić and Nenad Trinajstić. A graph-theoretical approach to structure-property relationships, 1992.

[78] Stanley Milgram. The small world problem. *Psychology today*, 2(1):60–67, 1967.

[79] Hans 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.

[80] Hans 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.

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

[82] Franck Kalala Mutombo. *Long-range interactions in complex networks*. PhD thesis, University of Strathclyde, 2012.

[83] Mark Newman. *Networks: an introduction*. OUP Oxford, 2010.

[84] Mark EJ Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003.

[85] Mark EJ Newman, Duncan J Watts, and Steven H Strogatz. Random graph models of social networks. *Proceedings of the National Academy of Sciences*, 99(suppl 1):2566–2572, 2002.

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

[87] Atsuyuki Okabe, Barry Boots, Kokichi Sugihara, and Sung Nok Chiu. *Spatial tessellations: concepts and applications of Voronoi diagrams*, volume 501. John Wiley & Sons, 2009.

[88] Tore Opsahl and Pietro Panzarasa. Clustering in weighted networks. *Social networks*, 31(2):155–163, 2009.

[89] Giuliano Andrea Pagani and Marco Aiello. The power grid as a complex network: a survey. *Physica A: Statistical Mechanics and its Applications*, 392(11):2688–2700, 2013.

[90] Seth 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.

[91] Xingqin Qi, Eddie Fuller, Qin Wu, Yezhou Wu, and Cun-Quan Zhang. Laplacian centrality: A new centrality measure for weighted networks. *Information Sciences*, 194:240–253, 2012.

[92] Harishchandra S Ramane, Deepak S Revankar, Ivan Gutman, Siddani Bhaskara Rao, B Devadas Acharya, and Hanumappa B Walikar. Bounds for the distance energy of a graph. *Kragujevac Journal of Mathematics*, 31(31):59–68, 2008.

[93] Milan Randic. Characterization of molecular branching. *Journal of the American Chemical Society*, 97(23):6609–6615, 1975.

[94] R Olfati Saber and Richard 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.

[95] Benno Schwikowski, Peter Uetz, and Stanley Fields. A network of protein–protein interactions in yeast. *Nature biotechnology*, 18(12):1257, 2000.

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

[97] Jonathon Shlens. A tutorial on principal component analysis. *arXiv preprint arXiv:1404.1100*, 2014.

[98] AMA Smith, Jeffrey Grierson, David Wain, Marian Pitts, and Pip 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.

[99] Lindsay I Smith. A tutorial on principal components analysis. Technical report, 2002.

[100] Olaf Sporns, Dante R Chialvo, Marcus Kaiser, and Claus C Hilgetag. Organization, development and function of complex brain networks. *Trends in cognitive sciences*, 8(9):418–425, 2004.

[101] Adina Raluca 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.

[102] Ali Sydney, Caterina Scoglio, Phillip Schumm, and Robert 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.

[103] Reiko Tanaka. Scale-rich metabolic networks. *Physical review letters*, 94(16):168101, 2005.

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

[105] Miroslav Trajković and Mark Hedley. Fast corner detection. *Image and vision computing*, 16(2):75–87, 1998.

[106] Alexander Tsiatas. *Diffusion and clustering on large graphs*. University of California, San Diego, 2012.

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

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

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

[110] Xiao Fan Wang and Guanrong Chen. Complex networks: small-world, scale-free and beyond. *Circuits and Systems Magazine, IEEE*, 3(1):6–20, 2003.

[111] Duncan J Watts and Steven H Strogatz. Collective dynamics of 'small-world' networks. *nature*, 393(6684):440–442, 1998.

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

[113] Barry Wellman and Stephen D Berkowitz. *Social structures: A network approach*, volume 2. CUP Archive, 1988.

[114] Tina Wey, Daniel T Blumstein, Weiwei Shen, and Ferenc Jordan. Social network analysis of animal behaviour: a promising tool for the study of sociality. *Animal behaviour*, 75(2):333–344, 2008.

[115] Jamelia A Williams, Stephen M Dawson, and Elisabeth Slooten. The abundance and distribution of bottlenosed dolphins (tursiops truncatus) in doubtful sound, new zealand. *Canadian journal of zoology*, 71(10):2080–2088, 1993.

[116] Richard C Wilson, Edwin R Hancock, and Bin Luo. Pattern vectors from algebraic graph theory. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(7):1112–1124, 2005.

[117] Robin J Wilson. *An introduction to graph theory*. Pearson Education India, 1970.

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

[119] Bai Xiao, Edwin R Hancock, and Richard C Wilson. Graph characteristics from the heat kernel trace. *Pattern Recognition*, 42(11):2589–2606, 2009.

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

[121] Laura A Zager and George C Verghese. Graph similarity scoring and matching. *Applied mathematics letters*, 21(1):86–94, 2008.