Methods for analyzing complex networks using random walker approaches

Thesis
submitted in fulfillment of the requirements for the degree of
Doktor der Naturwissenschaften

by

Nataša Djurdjevac (Đurđevac), M.Sc.

Supervisor: Professor Dr. Christof Schütte

Fachbereich Mathematik und Informatik
Freie Universität Berlin
Berlin, April 2012

Betreuer: Prof. Dr. Christof Schütte
Freie Universität Berlin
Fachbereich Mathematik und Informatik
Arnimallee 2-6
14195 Berlin

Gutachter: Prof. Dr. Christof Schütte
Prof. Dr. Wilhelm Huisinga (Universität Potsdam)

Tag der Disputation: 10. August 2012

Contents

Introduction **1**

1 Analyzing complex networks **7**
1.1 Basic properties of networks 7
1.1.1 Adjacency matrix 9
1.1.2 Degree and degree distribution 10
1.1.3 Shortest path and betweenness centrality 13
1.1.4 Clustering coefficient 15
1.1.5 Laplacian matrix 16
1.2 Modeling complex networks 18
1.2.1 Random graphs 18
1.2.2 Small-world networks 18
1.2.3 Scale-free networks 20
1.3 Analyzing real-world networks 20
1.3.1 Graph clustering 20
1.3.2 Hubs in networks 23
1.3.3 Real-world networks 24

2 Random walks on undirected networks **27**
2.1 Standard random walk 28
2.1.1 Properties of random walks on networks 28
2.1.2 Invariant measure and reversibility 30
2.1.3 Transfer operator 34
2.1.4 Random walks on weighted networks 36
2.2 Modules and metastable sets 37
2.3 Time-continuous random walks 39
2.3.1 Time-continuous Markov processes 39
2.3.2 The new random walker approach 42

3 Modules and metastability **47**
3.1 Analyzing Markov processes 48
3.1.1 Hitting times 48
3.1.2 Committor functions 49
3.2 Metastable sets of random walks 52

3.3 Spectral characterization of metastability 53
3.3.1 Complete modular partition 54
3.3.2 Metastable hitting times 55

4 Identification of modules **57**
4.1 Fuzzy affiliation functions 58
4.2 Fuzzy modular partition 59
4.2.1 Milestoning process 60
4.2.2 Jump statistics of milestoning process 60
4.3 Reversible Markov processes 65
4.3.1 Generalized eigenvalue problem 65
4.3.2 Approximation of Dominant Eigenvalues 67
4.4 Algorithm for identification of modules 71
4.4.1 New heuristic for finding modules 71
4.4.2 Finding optimal modules 75
4.5 Related work 77
4.5.1 Contrasting different graph clustering approaches 79

5 Identification of hubs **83**
5.1 Transition Path Theory for Markov Jump Processes 84
5.1.1 Global transition behavior: Reactive flows 84
5.1.2 Reaction pathways 87
5.1.3 Local transition behavior 88
5.2 Hubs in undirected networks 89
5.3 Inter-module hubs 91
5.3.1 Inter-module hubs and reactive flows 92
5.3.2 Inter-module hubs and important paths 94
5.4 Module hubs 99
5.4.1 Module bottleneck hubs 100
5.4.2 Module central hubs 101
5.5 Example: Finding hubs 103
5.6 Related work 104

6 Analyzing real-world networks **107**
6.1 Analyzing US political books network 108
6.2 Analyzing yeast PPI network 111

7 Analyzing directed networks **117**
7.1 Analyzing non-reversible Markov processes 118
7.1.1 Transfer operators of non-reversible processes 118
7.1.2 Extended detailed balance condition 120
7.1.3 Singular value decomposition of transfer operators 122
7.1.4 Coarse graining of non-reversible processes 124

7.1.5 Approximation quality of Markov state models for non-reversible processes 127
7.2 Random walks on directed networks 128
7.2.1 Identification of modules and hubs using random walks on directed networks 132
7.2.2 Open questions and perspectives 133

Summary **135**

Zusammenfassung **137**

Introduction

With the increasing power of high-throughput technologies and storage capacities, more and more data-sets from real-world systems become available [12, 79, 166, 124, 105]. These data-sets represent a rich source of information that can be used for modeling various real-world systems, such as social, technological and biological systems. Systematical analysis and understanding of this data is of crucial importance for uncovering the basic principles of organization and function of the underlying system.

A common way to mathematically represent this data is by **complex networks**, where nodes represent entities, for example web-pages, proteins, people; and edges represent relations between the system entities, such as links, interactions and friendships. However, there are different types of modeling approaches that consider also attributes and features of nodes and edges when constructing networks [96], such as functional properties of proteins in biological networks or age and gender of individuals in friendship networks. Construction and analysis of complex networks have become a topic of major research interest in many fields of science, since the abstract representation of networks can be used to describe a wide range of complex systems that span biological, physical and social sciences [6, 121, 125]. Therefore, understanding the behavior of complex systems starts with understanding the topology and dynamics of the associated networks.

The beginnings of network study originate from graph theory, a field of mathematics that has its roots in the 17^{th} century and is based on the famous work of Leonhard Euler. Over the last years the area of network analysis has expanded rapidly and attracted a lot of attention, creating the so called "new science of networks" [14, 125]. Networks are now widely recognized not only as outcomes of complex interactions, but as key determinants of structure, function and dynamics in many real-world systems. In particular, over the last years identification of the following network structures (elements) has received a considerable amount of attention:

1. **Modules (also clusters, communities)** are connected subgraphs that are highly interconnected, but have relatively few connections to nodes in other modules.

2. **Hubs** are nodes that are important for inter- and intra-connection of modules.

When analyzing real-world networks these elements are of special importance for understanding and modeling of the underlying systems, as their identification could yield valuable insights about the organization and functional mechanisms of these systems. Therefore, developing methods for identification of modules and hubs will be the main topics of interest in this thesis.

Identifying modules

Networks that describe real-world systems are often very large (several thousands of nodes) and complex. Finding modules is one of the well-established approaches for reducing the complexity of such networks, as it helps decomposing the complex network structure into smaller sub-units that can be analyzed in more detail in subsequent stages [70, 130, 140]. Furthermore, modules often correspond to functional units of the underlying system, such as protein complexes in biological networks, pages with similar topic in WWW or people with same interests in social networks. For these reasons, many new algorithms for identification of modules in networks have been developed [122].

Particularly, the method of **random walks**, as a fundamental dynamic process [85] has been well-established for structural analysis of networks, as it can fully account for local as well as global topological structure within the network [158, 128]. The random walk process defines a Markov chain on the state space that is given by the network's nodes. Analyzing this Markov chain can provide valuable information about the topological and dynamical properties of the network. Some of the random-walk-based methods use properties of random walks to introduce a metric on a network that would quantify the structural similarities of these nodes [56, 132, 120, 102]. Then, modules are found in such a way that the sum of the distances between each node and its module should be small.

Other random-walk-based methods are using the idea that in terms of the random walk process, a module represents a set of nodes where the random walker is trapped for a long period of time. More precisely, modules represent **metastable sets** of the random walk process. There exists rich literature addressing different variants of the problem of identifying dominant metastable sets of Markov processes [114, 102, 22, 55, 112, 111, 40, 118, 110, 50] that can be used as possible approaches for identifying modules. In particular, the so-called **Markov State Models** (MSM) have been successfully used as low-dimensional models for metastable Markov processes [148, 49, 149, 150, 28]. The goal of MSM is to approximate the original Markov process by a Markov chain on a small state space, such that the longest timescales of the dynamics of the original process are reproduced well. In most MSM-based approaches, the state space of the obtained Markov chain consists of sets that cover the whole original state space. In terms of Markov processes on networks this means that we consider a **full decomposition** of a given network into modules. However, in many real-world networks the assignment of nodes to only one of the modules is not always straightforward, i.e. a node can be assigned

to more than one module with some probability. In such cases we don't consider full, but **fuzzy decomposition** of the network into modules, also called fuzzy clustering.

Different random-walk-based fuzzy clustering methods have recently been discussed in the literature [50, 145, 106]. Many of these methods are based on spectral decomposition ideas, where the number and choice of modules is governed by spectral properties of the transition matrix P of the random walk process. Especially, the number of **dominant eigenvalues** of P, i.e. the eigenvalues that are significantly closer to 1 in modulus than all other eigenvalues in the spectrum, indicates the number of modules in the network. However, it is often difficult to determine the number of modules, since the spectrum does not always show a clear gap. In this thesis we will addressed this problem in detail and point out usual causes of this phenomena. Furthermore, we will introduce a new type of random walk processes, namely **time-continuous** random walks, which can be used to overcome this obstacle. We will use the stochastic properties of this new process in order to define the **fuzzy assignment functions**, which will provide a probability of a given node to belong to a certain module.

Identifying hubs

This problem of identifying hubs is closely related to **network robustness** [43, 44, 155], which measures resilience of a network to the removal of its nodes and edges. When analyzing real-world networks, of particular interest is identification of such "weak points" in a network, whose removal can significantly perturb and sometimes even cause a breakdown of the entire network. Such nodes and edges are often shown to correspond to elements and interactions that are crucial for the proper functioning of the underlying system [160, 169, 75, 73]. As we mentioned above, in this thesis we will focus on characterization and identification of **hubs**, which represent nodes that are **important** for inter- and intra-connection of modules. Hubs can correspond, for example, to computers that connect many other computers in the Internet network or to lethal proteins in biological networks. Many different measures of importance of nodes [89, 70, 92, 61] have been introduced aiming at identifying essential functional elements of real-world systems. However, recent results coming from various applications revealed that the currently available methods are not always sufficient to identify all important hubs [169, 75, 73].

Most of the available approaches for module and hub identification are considering undirected networks, that is, networks where connection between nodes don't have a direction. Many real-world networks are of this type, such as transportation networks, where if there is a connection from a city x to a city y, then there is a connection from y to x. However, interactions often do have a direction, for example in citation networks where the citation between two scientific papers can be done only in one direction. Generalizing the definition of modules and hubs to the case of directed networks is not an easy task, especially concerning the not

so clear interpretation of these structures in the underlying systems. From the mathematical point of view, introducing directed edges produces asymmetries in various matrix representations of the networks, which makes most of the currently available methods inapplicable to directed networks. Especially, the problem of applying random-walk-based methods on directed networks is that the random walk processes are non-reversible, that is we have to deal with non-reversible Markov chains.

Outline of the thesis

The main goal of this thesis is to provide new random-walk-based methods for analyzing complex networks. Networks of our interest are coming from various applications and they can describe a wide range of real-world complex systems. The analysis of these networks is focused on finding network structures and elements that are of special importance in the underlying system. Especially, the main focus of this thesis is using random walker approach for developing new methods for identification of modules and hubs in a given network. Most parts of the thesis consider undirected networks, whereas in Chapter 7 introduced approaches are generalized to the case of directed networks.

The outline of the thesis is as follows:

In Chapter 1 we will introduce the basic graph theoretical concepts that will be crucial for the analysis of complex networks. Then we will present the relevant network models (Section 1.2) that describe properties of real-world systems. Finally, in Section 1.3 we will highlight the importance of finding modules and hubs when analyzing different real-world networks.
Focusing on analyzing undirected networks, in Section 2.1 we will refer to standard random-walk-based method as a well-established approach for structural analysis of networks. However, this method often fails to correctly detect modules that are metastable sets of this random walk process (Section 2.2). This is because also many other structures, such as long chains, represent metastable sets of this process. In order to overcome this problem we will introduce a new time-continuous random walk process (Section 2.3), defined in such a way that its only metastable sets are the dense modular structures of the network.
Chapter 3 deals with characterizing metastable processes and metastable sets in terms of the basic objects from the theory of Markov processes. In Section 3.3 we will present two spectral characterizations of metastable processes, introduced in [86, 88] and [23, 28]. We will show how spectral properties of random walk processes are connected to the properties of the metastable processes and in particular, that the behavior of the random walk process on its longest time-scales is encoded in the dominant eigenvalues of P.
In Chapter 4 we will present our new approach for finding fuzzy decomposition of a network into modules. In Section 4.1 we will introduce fuzzy assignment functions, the so called **committors** [57, 58], which will take the form of a probability that is

essentially dynamics-based. More precisely, the assignment probability of a given node x to a certain module M is the probability that the random walk process, if started in x, enters M before it reaches any other module. We will use the committor functions to generalize the above mentioned MSM approach and develop a **fuzzy Markov State Models** approach by defining small disjoint sets in the most dominant metastable regions (see Sections 4.2 and 4.3). In order to ensure good approximation quality of the resulting MSM we will consider error between the original and reproduced dominant eigenvalues (see Theorem 9), since they capture the long-term behavior of the two metastable processes. Based on this idea, in Section 4.4 we will present our new algorithm for identifying modules and in Section 4.5 we will compare it with several state-of-the-art algorithms from the literature.
In Chapter 5 we will present our new method for hub finding, where as mentioned above, we define hubs as nodes that are important for the communication in the network. The notion of "communication" will be specified using the basic objects from Transition Path Theory [58, 117], that will be presented in Section 5.1. Sections 5.2, 5.3 and 5.4 will introduce new types of hub nodes and algorithms for their identification. We will end this chapter by presenting state-of-the-art approaches for hub finding.
In Chapter 6 we will demonstrate our methods on two real-world examples: a social and a biological undirected network. We will indicate the main problems in analyzing real-world networks, that are often caused by unreliable initial data-sets on one hand and no general strategy for evaluating the obtained results on the other hand. In general, these problems are not connected to the theoretical considerations of the applied algorithms, but they certainly have a huge impact when analyzing their quality. Therefore, when analyzing real-world networks one should keep these issues in mind.
Chapter 7 deals with analyzing directed networks using random-walk-based approaches. In Section 7.1 we will develop a new approach for analyzing different non-reversible processes. Then, we will adopt this approach specifically to the case of directed networks and using the idea from [26] we will define two random walk processes for analyzing directed networks, namely the forward and backward random walk process. Finally, we will propose how to generalize our new methods for module and hub identification from Chapters 4 and 5 using the same topological definitions of these objects as in the case of undirected networks. The ideas presented in this chapter can be used as a starting point for developing methods that could enable more detailed analysis of directed networks. Therefore, we will end this thesis by listing some relevant open problems that will be left for the future work.

Acknowledgements

It is my pleasure to thank all those people who supported me during the time I did my Ph.D. First of all, my advisor, Christof Schütte, for introducing me to this field of mathematics, guiding my research and giving me constant encouragement and support. I was fortunate to have an advisor from whom I could not only learn a lot of new mathematics, but also many new ways to think about mathematics.

I would like to express my gratitude to the Berlin Mathematical School (BMS) for supporting my studies and providing a very inspiring mathematical environment in Berlin. I thank Carsten Hartman and Illia Horenko for valuable scientific and very useful non-scientific advice; Marco Sarich for walking all the way together and making research become much more fun; Stefanie Winkelmann, Tim Conrad and Sharon Hüffner for giving useful suggestions and advice to improve this thesis at its various stages. Especially, I would like to express my thanks to the whole Biocomputing group for a very motivating research atmosphere, that I have appreciated a lot.

In particular, I would like to thank my family and friends, who accompanied me during the last few years. I am deeply grateful to my parents, not only for being there, but also for making my first steps in mathematics fun and interesting. I thank Tim for his great support, patience and advice. Finally, I would like to thank my sister, Ana, for her love and support I could always rely on.

This work was funded by the Berlin Mathematical School (BMS).

Chapter 1

Analyzing complex networks

Many real-world systems such as the Internet, social groups, air transportation systems, protein-protein interactions and metabolic pathways can be modeled by networks. Discovering the relations between structural elements of networks and their functions in the underlying system could yield valuable information about the basic principles of this system. In this sense, analyzing complex networks can help us to understand and predict the behavior of the underlying system.

This chapter deals with introducing graph theoretical concepts that are used to describe a wide range of complex networks from real-world systems. This will provide us with specific characterizations of different networks that will turn out to be crucial for their complete analysis. We will start with Section 1.1, where we will define some of the basic network topological properties; for more details on this topic see [32, 26, 54, 20]. Then, in Section 1.2 we will introduce three different network models that were proposed to describe properties of real-world networks, namely random graphs, small-world networks and scale-free networks. Finally in Section 1.3, we will highlight two challenging problems in analyzing complex networks and explain their interpretation and importance on real-world networks.

1.1 Basic properties of networks

The description of complex systems by means of complex networks is based on the natural properties of the underlying system, as explained in the introduction. More precisely, the elements of the system are identified as the *nodes* of the network and interactions between these elements are represented by the *edges* between the network nodes. For example, if we want to describe friendships in a social system, we can form a so-called network of friendships. Nodes in this network represent individuals of the system and an edge between two individuals exists if they are friends. In a similar way, we can also create a network for an air-transportation system, where nodes represent cities and edges direct flights between two cities.
Mathematically, networks are known as graphs, the elementary objects of graph theory. We will now introduce some basic concepts from graph theory that are used to topologically characterize networks and provide their interpretation for the underlying complex system.

Definition 1 *A **network** or **finite graph** $G(V, E)$ is defined by*

- *a set of **nodes** (states, vertices) V, and*
- *a set of **edges** (links, connections) $E \subseteq V \times V$ between the nodes.*

If for two nodes $x, y \in V$ there exists an edge that connects them, we say that x and y are *neighbors*. In graph theory the number of nodes $|V| = n$ is called the **order** of a network, whereas for the underlying system n is the **size of the system**, as it represents the number of elements of the system. Since the network is a description of the system, in the following we will refer to n as the size of the network and not to the number of edges of the network $|E|$, as it is usually done in graph theory.

Depending on the nature of interactions in the system, edges in the network can be **undirected** or **directed**. Undirected edges can be used, for example to describe social relations or interactions of proteins, as the relation between these elements is mutual [163, 164, 75]. On the other hand in citation networks only one of the authors can cite the other one, forming an directed edge [125, 137]. The orientation of edges results in the following differentiation of graphs:

1. **undirected graphs**: all edges are undirected,
2. **directed graphs** (digraphs): all edges are directed, and
3. **mixed graphs**: edges can be directed or undirected.

Since mixed graphs are encountered rarely in the following, we will consider only undirected and directed graphs, as special classes of the family of mixed graphs. From the definition of graphs, both directed and undirected edges can link two nodes, but can also connect a node to itself, forming a **loop**. In general, there can also exist multiple edges connecting two nodes. Graphs that do not have loops or multiple edges are called **simple** graphs. Complex networks are often represented as simple graphs, since in many applications elements of a system can not interact with themselves and multiple edges do not exist. For this reason, in the following, we will refer to simple graphs.

Many real-world networks are characterized by one more parameter, namely the **weight** of interactions. Weights can correspond for example to distances between cities in transportation networks or the amount of information flow in communication networks. More formally, relations between elements of an underlying system can be characterized by a specific function that quantifies the property of interest. Obviously, this information is fundamental for a complete description of this system.

Definition 2 *An edge-weighted, in the following **weighted graph**, $G(V, E, W)$ is a graph in which each edge (x, y) is assigned a non-negative number, called **edge weight** $w(x, y)$, that satisfies $w(x, y) = 0$ if $(x, y) \notin E$ and $w(x, y) > 0$ if $(x, y) \in E$. Matrix $W = (w(x, y))_{x,y \in V}$ is the **weight matrix** of the graph G.*

As weights can reflect some of the crucial systems properties, their appropriate choice has been a topic of various studies [19, 71, 166]. However, there are many examples of real-world networks, in which edge weights are not considered to be of crucial importance for describing a system because it is often very hard, if not impossible, to quantify relations in a system. For example, in networks of friendships although there have been attempts to quantify friendships as a function of their duration, emotional intensity and intimacy [71], these attempts have been argued, as it is very hard to numerically compare two friendships. In such cases, the weights have a simple, binary form, meaning that edges are either present or absent $w(x, y) = 1$, $\forall(x, y) \in E$. Graphs for which w has a simple form are called **unweighted graphs**.

Example 1 *Figures 1.1 and 1.2 show examples of an unweighted, undirected network and an unweighted, directed network. Figure 1.3 shows an example of a weighted, undirected network and its weight matrix W. In this plot, width of every edge (x, y) is proportional to its weight $w(x, y)$ and the green label of edges represents their weights.*

1.1.1 Adjacency matrix

A common way to represent graphs is by means of matrices, in particular

Definition 3 ***The adjacency matrix*** $A = (a(x, y))_{x,y\in V}$ *of a graph $G(V, E, w)$ is a $n \times n$ matrix with entries*

$$A(x, y) = \begin{cases} 1, & \text{if } (x, y) \in E \\ 0, & \text{if } (x, y) \notin E. \end{cases}$$

Defined in this way, the adjacency matrix is unique for every graph, up to permutation of its rows and columns.

From the structure of adjacency matrix we can conclude the main topological characteristics of the network. For instance, null diagonal elements of A imply the absence of loops. Furthermore, if A is symmetric, i.e. $a(x, y) = a(y, x), \forall x, y \in V$, then its graph is undirected. In particular, the sum of off-diagonal elements of A equals to $2N$, where N is the number of non-loop edges in the graph. On the other hand, asymmetric adjacency matrix implies that its graph is directed and the sum of its off-diagonal elements is N.

The eigenvalues of the adjacency matrix, representing the **spectrum** of A, can discover different structural properties of the network. Obviously, since the adjacency matrix of every undirected graph is symmetric, its spectrum is characterized by real eigenvalues and an orthogonal eigenvector basis, whereas the spectrum of A for directed graphs can have complex eigenvalues. Furthermore, the adjacency matrix spectrum enables detecting if the graph is bipartite.

Definition 4 *A **bipartite** graph is a graph $G = (V, E)$, such that V can be represented as an union of two disjoint sets V_1 and V_2 and there are no edges connecting nodes within each of the two sets.*

For example, bipartite graphs can be used to model a special type of social networks, the *affiliation networks*. These networks are characterized by two types of entities, the individuals and the groups to which these individuals are affiliated. For example, the social network of film actors consists of group of actors and the group of movies where these actors played [166]. The adjacency matrix of bipartite graph is of the form

$$A = \begin{pmatrix} 0 & B \\ B^T & 0 \end{pmatrix},$$

where B is a $p \times r$ matrix and $|V_1| = p$, $|V_2| = r$. Furthermore, if a graph is bipartite, then the spectrum of its adjacency matrix has specific properties, such as

Proposition 1
A graph is bipartite if and only if the spectrum of its adjacency matrix is symmetric around zero.

This means that if λ is an eigenvalue of A, then $-\lambda$ is also an eigenvalue of A. For the proof of this proposition and more details about the spectral properties of adjacency matrix of bipartite graphs, see [37].

1.1.2 Degree and degree distribution

We will now use properties of adjacency matrices to study the connectivity of a particular node in a given network. Later we will generalize this approach in order to define the global connectivity of a network.
One of the basic properties of a node is its degree. To this end, we define

Definition 5 ***The out-degree*** $d_{out}(x)$ *of a node* x *in a directed graph is the number of edges that exit from node* x

$$d_{out}(x) = \sum_{y \in V} a(x, y). \tag{1.1}$$

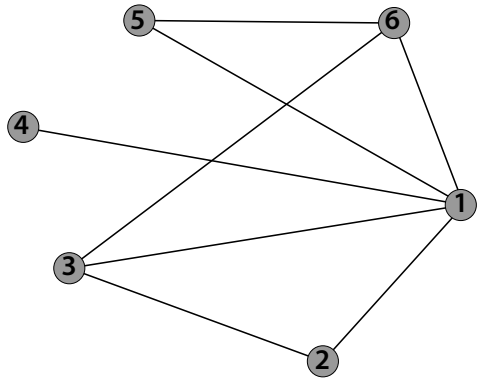
The in-degree $d_{in}(x)$ *of a node* x *in a directed graph is the number of edges that enter node* x

$$d_{in}(x) = \sum_{y \in V} a(y, x). \tag{1.2}$$

The degree $d(x)$ *of a node* x *in a directed graph is the total number of edges that enter and exit node* x

$$d(x) = d_{in}(x) + d_{out}(x). \tag{1.3}$$

Therefore, the degree of a node quantifies the connectivity of a node in the network. More precisely, if $d_{out}(x) = 0$ then node x doesn't have any outgoing edges and is considered to be a **sink** of the network. If $d_{in}(x) = 0$, then node x doesn't have any incoming edges and is called a **source** of the network. Especially, a node x for which $d(x) = 0$ is **disconnected** from the network.
For undirected graphs, the node degree is defined in the following way:



$$A = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \end{pmatrix} \begin{matrix} \boldsymbol{d} \\ \mathbf{5} \\ \mathbf{2} \\ \mathbf{3} \\ \mathbf{1} \\ \mathbf{2} \\ \mathbf{3} \end{matrix}$$

Figure 1.1: The labeled undirected network and its adjacency matrix. The degrees of network nodes are shown next to the adjacency matrix.

Definition 6 ***The degree*** $d(x)$ *of a node* x *in an undirected graph is the number of connections that* x *has with other nodes of the network*

$$d(x) = \sum_{y \in V} a(x, y). \tag{1.4}$$

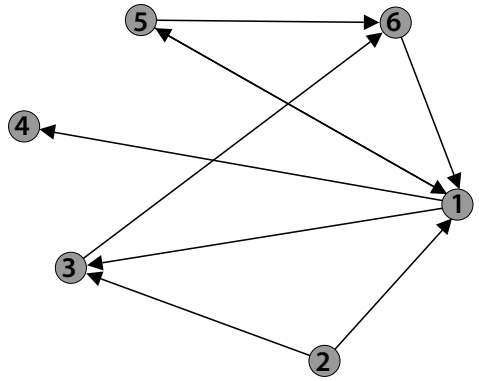
Since the adjacency matrix of undirected graphs is symmetric, it holds that $d(x) = d_{in}(x) = d_{out}(x), \forall x \in V$. Then, we say that a node x is disconnected from a network if $d(x) = 0$.
In undirected, weighted networks with a given weighting function w, apart from the network degree, another special characterization of nodes can be given

Definition 7 ***The weighted degree*** *or* ***strength*** *of a node* x *in an undirected, weighted networks is*

$$s(x) = \sum_{y \in V} w(x, y) \tag{1.5}$$

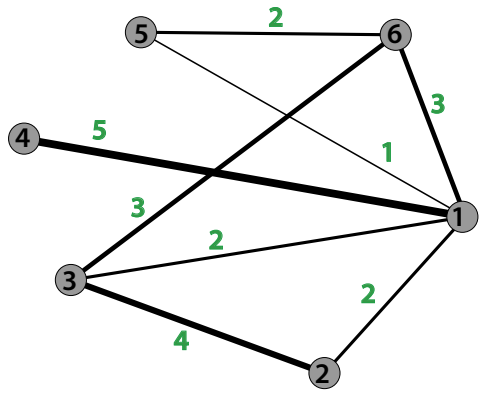
The degree of a node as a measure of node connectivity has found an important place in complex network analysis. In particular, in certain network-types nodes that have a high degree are shown to represent vulnerable points of these networks, usually corresponding to essential functional points of the underlying system. Such nodes are usually referred to as **hub nodes**. More details about hubs and their identification will be given in Chapter 5.

Example 2 *Let us now examine some properties of networks introduced in Example 1. The graph shown in Figure 1.1 consists of six nodes, labeled with numbers from* 1 *to* 6. *This graph is undirected, which can be observed also from the symmetry of*



$$A = \begin{array}{c} \\ \\ \\ \\ \\ \\ \boldsymbol{d_{in}} \end{array} \begin{array}{c} \begin{pmatrix} 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \\ \mathbf{3\ 0\ 2\ 1\ 0\ 2} \end{array} \begin{array}{c} \boldsymbol{d_{out}} \\ \mathbf{2} \\ \mathbf{2} \\ \mathbf{1} \\ \mathbf{0} \\ \mathbf{2} \\ \mathbf{1} \end{array}$$

Figure 1.2: The labeled directed network and its adjacency matrix. The in- and out-degrees of all 6 network nodes are shown next to the adjacency matrix.



$$W = \begin{pmatrix} 0 & 2 & 2 & 5 & 1 & 3 \\ 2 & 0 & 4 & 0 & 0 & 0 \\ 2 & 4 & 0 & 0 & 0 & 3 \\ 5 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 2 \\ 3 & 0 & 3 & 0 & 2 & 0 \end{pmatrix} \begin{array}{c} \boldsymbol{s} \\ \mathbf{13} \\ \mathbf{6} \\ \mathbf{9} \\ \mathbf{5} \\ \mathbf{3} \\ \mathbf{8} \end{array}$$

Figure 1.3: The labeled weighted network and its weight matrix. Widths of network edges corresponds to their weights, marked in green. The strengths of nodes are shown next to the weight matrix of the network W.

its adjacency matrix A. Furthermore, zeros on the diagonal of A indicate that there are no loops in this network. As explained earlier, the degree of every node x can be calculated by summing up the elements of the x-th row of the matrix A. In Figure 1.1, the degree of every node is written next to the adjacency matrix. We see that node 1 *has the highest possible degree, because it is connected to every other node in this graph.*

The second example graph is directed and it is shown together with its adjacency matrix A in Figure 1.2. This graph has the same number of nodes and edges as the graph from the previous example. Moreover, the only difference to the previous graph is that its edges are directed. From this graph we notice that the node labeled as 4 is a sink of the network, because there aren't any outgoing edges from this node. There is also one source node - node 2, as no edges enter this node. These properties of the network can also be derived from the adjacency matrix A and degrees of nodes, marked next to A. Especially, we distinguish between in- and out-degrees of nodes (see Definition 5) and see that $d_{out}(4) = 0$ and $d_{in}(4) = 0$.
Figure 1.3 shows an example of a weighted, undirected network and its weight matrix W. Furthermore, for every node x we calculated its strength $s(x)$ according to equation (1.5). Notice that this graph is just a weighted version of the unweighted graph shown in Figure 1.1. Thus, the adjacency matrices and the degrees of nodes of these two graphs are the same. Now, we can compare the degrees and node strengths in the weighted graph. We see that the ordering of nodes according to their degrees does not coincide with the ordering according to their strength. This is because degree is a measure that is dependent only on the position of edges in the network. On the other hand, the strength of a node depends additionally on the newly introduced parameter of the network - the weight.

The node degree represents a local property of a node. In contrast to that, the **degree distribution** of a graph provides a measure for global characterization of complex networks. The degree distribution is one of the most studied statistical properties of networks [6, 122]. For a given network, we define

$$P(k) = \frac{n(k)}{n}, \tag{1.6}$$

where $n(k)$ is the number of nodes that have a degree k. More formally, $P(k)$ represents the probability that a node chosen uniformly at random will have a degree k. Many real-world networks such as the world wide web and some social networks are shown to have similar degree distribution of a special type. These networks, are also found to share some similar properties. In this sense, network classification according to the degree distribution can reflect many important functional and structural similarities between the networks in these groups. We will address this point in Section 1.2.

1.1.3 Shortest path and betweenness centrality

As stated above, a node x is disconnected from the network if $d(x) = 0$. The next natural question to ask is: How can we check if two nodes are connected?
If two nodes x and y are connected, then a sequence of nodes that are visited when going from a node x to a node y is called a **path**. In particular, in directed graphs we consider a **directed path** from x to y. A special type of paths are **cycles**, i.e. paths that start and end in the same node. We say that two nodes are connected

if there is a path between them. Now, we can generalize the notion of connectivity of nodes to connectivity of graphs.

Definition 8 *An undirected graph is **connected** if every node can be reached from every other node, whereas a directed graph is **connected** if for all pairs of nodes $x, y \in V$ there exists either a directed path from x to y or a directed path from y to x. If for all $x, y \in V$ in a directed graph both directed paths from x to y and from y to x exist, then this is a **strongly-connected** graph. A **weakly-connected** graph is a directed graph whose underlying undirected graph is connected.*

A graph $G = (V, E)$ that is not connected is called a disconnected graph. When dealing with disconnected graphs, the usual approach is to decompose this graph into (strongly) **connected components**, that is maximal, (strongly) connected subgraphs and analyze them separately. Identification of all connected components of a graph can be done by using breadth-first search (BFS) or depth-first search (DFS).

Paths between two nodes in a graph can serve for calculating the distance between these nodes. More precisely, the number of edges of a path that connects two nodes can be used as a measure of their distance. Since there can be many paths connecting two nodes, of particular interest is the identification of

Definition 9 *A **shortest path** between any two nodes $x, y \in V$ is a path connecting x and y that has the minimal number of edges. The number of edges of a shortest path between two nodes is the **distance** between these two nodes. Especially, the longest of all shortest paths in a network is called a **diameter** of a network.*

Shortest paths are used to define a measure for topological importance of a node or an edge, namely the **betweenness centrality**.

Definition 10 *For any given node (or edge) of a network, its **betweenness centrality** is calculated as a fraction of all shortest paths that go through this node (edge).*

Nodes and edges with high betweenness centrality, often called (topological) bottlenecks, are shown to be essential connectors in many real-world networks, usually representing the elements that link together different disconnected components of a network. Because of this, betweenness centrality is often used for identifying important structural components of a network, such as communities [126] and hubs [169]. State of the art algorithms for finding shortest paths in both undirected and directed graphs are Dijkstra's algorithm, the Bellman-Ford algorithm and the Floyd-Warshall algorithm (for more details about these algorithms see [46]).

Example 3 *Figure 1.4 shows how an undirected graph with five nodes and four edges changes when we add new edges. We will observe how these changes influence the betweenness centrality of graph nodes. The node with the highest betweenness*

centrality is node 1*, as it belongs to all paths between any two nodes in this graph. When we add three new edges to this graph (colored in blue in the middle plot in Figure 1.4), the betweenness centrality of node* 1 *decreases. This is due to the appearance of new edges that connect some of the other nodes. Now node* 1 *takes part only in the following shortest paths:* $2 \to 4 : 2 - 1 - 4$*,* $2 \to 5 : 2 - 1 - 5$ *and* $3 \to 5 : 3 - 1 - 5$*. Adding three new edges to this graph, this becomes a regular graph, i.e. a graph where all nodes have the same degree. Here,* $d(x) = 4, \forall x \in V$*, so this is an* 4*-regular graph, implying that every node is a neighbor of every other node. Therefore, in this graph node* 1 *doesn't belong to any of the shortest paths in the network and has betweenness centrality* 0*.*

1.1.4 Clustering coefficient

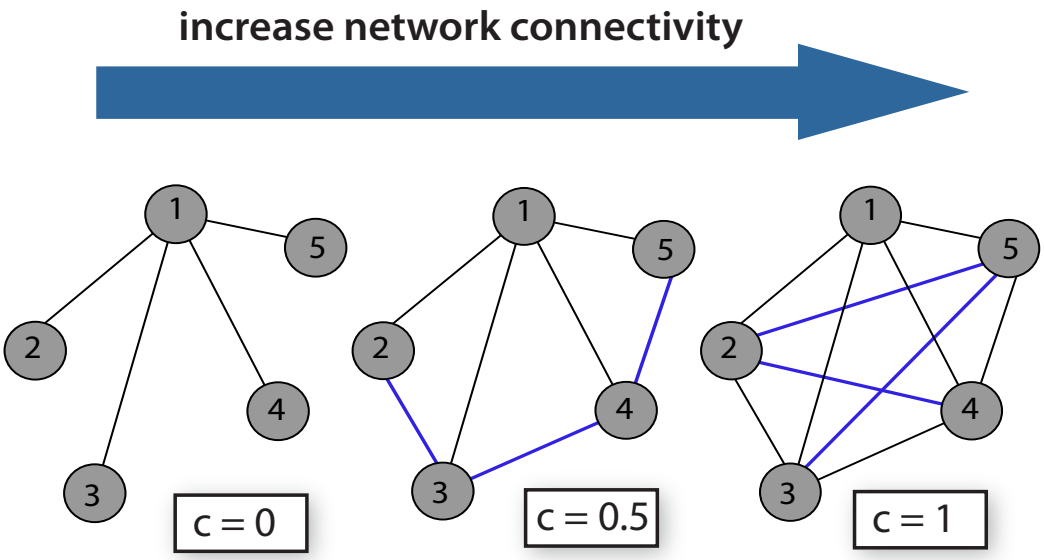


Figure 1.4: Changes of the clustering coefficient $c = c(1)$ of node 1 when adding new edges in the network. Left: The undirected network with five nodes. Middle: Adding three edges (colored in blue) between the neighbors of node 1. Right: By adding three new edges (marked in blue) every node becomes a neighbor of every other node in the network.

Another basic characterization of a node in a network can be obtained by describing the structure of its local neighborhood. In the context of social networks, it has been observed that it is very likely that friends of an individual x are also friends themselves. Many other real-world networks have the same property [84]. The following measure quantifies how densely connected the neighbors of a particular node are [166].

Definition 11 *The* ***local clustering coefficient*** $c(x)$ *of a node* x *is the ratio of the number of edges between its neighbors* n_x *and the number of possible edges*

between these neighbors. For undirected graphs this is

$$c(x) = \frac{2n_x}{d(x)(d(x)-1)}, \tag{1.7}$$

whereas for directed graphs $c(x)$ *has the form*

$$c(x) = \frac{n_x}{d(x)(d(x)-1)}.$$

In particular, $c(x) = 0$ *for all nodes that have only one neighbor, i.e.* $d(x) = 1$.

The distinction between calculating clustering coefficient in undirected and directed networks comes from the fact that in undirected graphs edges (x, y) and (y, x) are considered to be identical. This implies that the number of possible edges between the neighbors of a node x in directed networks is $d(x)(d(x)-1)$ and in undirected networks $d(x)(d(x)-1)/2$. Obviously, in both undirected and directed graphs, the number of edges between neighbors of x can be calculated from the adjacency matrix A as

$$n_x = \sum_{y,z \in V} a(x,y)a(x,z)a(y,z).$$

Now, the clustering coefficient for the whole network can be obtained as the average of the local clustering coefficients of all nodes in the network [166],

$$C = \frac{1}{n} \sum_{x \in V} c(x).$$

Defined in this way, the clustering coefficient is a measure for how nodes in a graph tend to cluster together. In Section 1.2.2, the clustering coefficient will be of great importance for characterization of a special class of networks called small-world networks.

Example 4 *Let us demonstrate how the clustering coefficient changes as the number of edges in a network change. We will show this on the example network from Figure 1.4. We have seen in Example 3 that in the network on the left hand side node* 1 *has the highest possible betweenness centrality, as it is part of all shortest paths in the network. However, its clustering coefficient is* $c(1) = 0$*, since none of its neighbors are connected to each other. Adding three edges between some neighbors of* 1 *produces the network in the middle of the figure, where the clustering coefficient changes to* $c(1) = \frac{3}{6}$*. Finally, in the plot on the right hand side, the neighborhood of* 1 *is fully connected and therefore* $c(1) = 1$.

1.1.5 Laplacian matrix

Another matrix representation of a graph can be given using

Definition 12 ***The Laplacian matrix*** $\mathfrak{L}$ *of an undirected graph is defined as*

$$\mathfrak{L} := D - A, \tag{1.8}$$

where $D = diag(d(1), \ldots, d(n))$ *is a diagonal matrix of node degrees.*

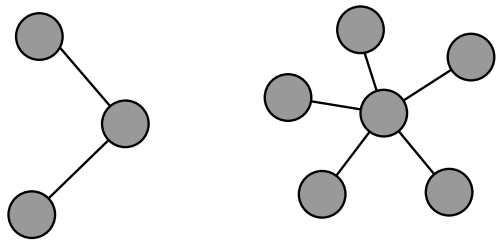
In the case of directed graphs, either in- or out-degree of nodes can be used to create the matrix D. From this definition it follows that the Laplacian of a graph has the following form

$$\mathfrak{L}(x,y) = \begin{cases} -1, & \text{if } (x,y) \in E \\ d(x), & \text{if } x = y \\ 0, & otherwise. \end{cases}$$

The Laplacian matrix can be used to find many properties of the graph, such as the number of spanning trees. Of special importance is its spectrum, as it can indicate the number of connected components of the graph. However, unlike the spectrum of the adjacency matrix, from the spectrum of $\mathfrak{L}$ it can not be seen whether the graph is bipartite. In order to improve this, **the normalized Laplacian** is often used [41]

$$\bar{\mathfrak{L}} = D^{-\frac{1}{2}} \mathfrak{L} D^{-\frac{1}{2}},$$

as its spectrum can indicate both the number of connected components and the existence of bipartite structures in the graph. In particular, the number of eigenvalues equal to zero represents the number of connected components, whereas it holds that the graph is bipartite iff for each eigenvalue λ, $2 - \lambda$ is also an eigenvalue.



$\lambda(A)$:	0	0	0	0	0	−1.41	1.41	−2.23	2.23
$\lambda(\mathfrak{L})$:	0	0	1	1	1	1	1	3	6
$\lambda(\bar{\mathfrak{L}})$:	0	0	1	1	1	1	1	2	2

Figure 1.5: Undirected, bipartite graph with two connected components and the spectrum of its adjacency matrix A, Laplacian matrix $\mathfrak{L}$ and normalized Laplacian matrix $\bar{\mathfrak{L}}$.

Example 5 *Figure 1.5 shows an example of an undirected graph that has* 9 *nodes organized in two connected components, which can be seen from the fact that both spectra of Laplacian matrix and normalized Laplacian matrix have two eigenvalues that are equal to zero. This graph is also bipartite, which can be seen from the symmetric spectrum of its adjacency matrix A and also from the spectrum of its normalized Laplacian matrix $\bar{\mathfrak{L}}$.*

1.2 Modeling complex networks

The so called "new science of networks" [14] has introduced novel paradigms of basic system properties, such as scale-free networks [15] and small-world structure [166]. These are shown to be correlated with specific structural properties of networks, for example organization into modules [80] or existence of highly connected nodes [15, 18]. In this Section we will present three main classes of modeling paradigms: random graphs, small-world networks and scale-free networks.

1.2.1 Random graphs

Introduced by Erdös and Rènyi in 1959, this is one of the most used network models [60]. This model is based on studying the probability space of undirected graphs. More precisely, given a fixed number of nodes n, all possible edges between these nodes appear with the same probability p. The degree distribution of random graphs is binomial or in the limit for large n a Poisson distribution

$$P(k) = \binom{n-1}{k} p^k (1-p)^{n-1-k} \approx \frac{(np)^k e^{-np}}{k!},$$

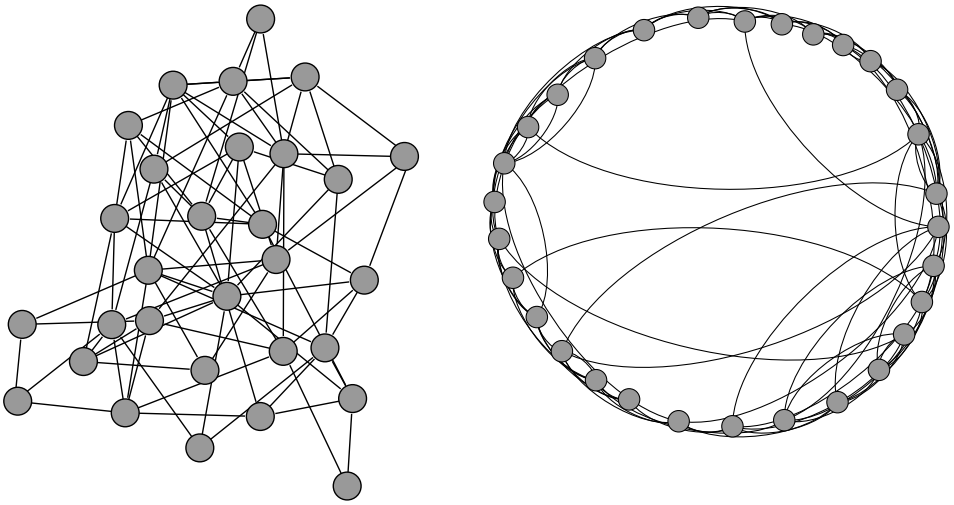
where k represents a degree of a node and $P(k)$ the probability that a node chosen uniformly at random will have a degree k. However, many of the real-world networks are shown to have a different degree distribution than random graphs. This makes them usually an inadequate model for describing structural features of complex networks. Despite this, random graphs had a great influence on modeling complex networks and are still widely used in many fields. Figure 1.6a shows an example of a random network with $n = 30$ nodes, where edges are generated with $p = 0.2$.

1.2.2 Small-world networks

This model was motivated by the observation that nodes in many real-world networks are connected, on average by very short paths. For example, "a path of just three reactions will connect almost any pair of chemicals in a cell" [17], most species in food webs are at most three links apart from each other [167] and "in certain portions of the Internet" Web pages are on average 19 clicks away from each other [7].

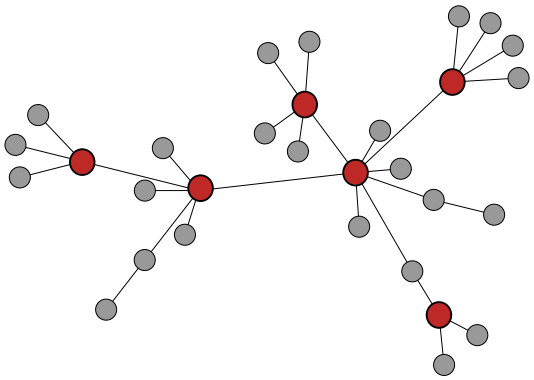
Small-world networks are networks that have a small average shortest paths length. The name "small-world" comes from the social science systems, where empirical experiments by the psychologist Stanley Milgram in 1967 implied the existence of the so called small-world phenomenon in human society. After these experiments many other studies have indicated that the average path length between any two people in the United States is six (also known as "six degrees of separation").

However, small average shortest path length is the property that can appear in many types of networks, such as random graphs. In order to resolve this issue, a particular category of small-world networks was introduced by Watts and Strogatz



(a) Random graph.

(b) Small world network.



(c) Scale-free network.

Figure 1.6: Three different network types, with 30 nodes each.

in 1998 [166]. Networks according to the **Watts and Strogatz model** have a small average shortest path length and a large clustering coefficient. Random graphs have a small clustering coefficient, because all edges are equally likely and there is no tendency towards clustering. Because of this, a "large clustering coefficient" in the Watts and Strogatz model in practice translates into "significantly higher" clustering coefficient compared to the one for random graphs. A wide range of real-world networks have shown to posses properties of the Watts and Strogatz model such as food webs, road networks, electric power grids, metabolic networks, networks of brain neurons and social networks. For an extensive review of small-world networks, we refer the reader to [165]. Figure 1.6b shows an example of a small-world network of the Watts and Strogatz model.

1.2.3 Scale-free networks

Many real-world networks are found to have a so called power law degree distribution [15, 16, 5, 90]

$$P(k) \sim k^{-\gamma},$$

where $\gamma > 0$ is a constant. Intuitively, this means that there exist many nodes with small degree and only a few nodes with very high degree. For example, a degree distribution of a directed network that describes the Word Wide Web is found to follow a power law with $\gamma_{in} \approx 2.1$ and $\gamma_{out} \approx 2.45$ [7]. Empirical studies have shown that the power law degree distribution characterizes many other networks such as the actor collaboration network $\gamma \approx 2.3$, the metabolic network of E.coli $\gamma_{in}, \gamma_{out} \approx 2.2$ [90], the Internet $\gamma \approx 2.2$ [62] etc..
Networks that have a degree distribution which approximately follows a power law are commonly called **scale-free networks**. This name was motivated by the fact that power law degree distribution is scale invariant, that is the law of the distribution doesn't change when being multiplied by a constant. Many scale-free networks are shown to have certain similar topological properties such as the existence of highly connected node-hubs [15, 18]. These nodes are of special interest, as they can have a great impact on the network's robustness. Furthermore, scale-free networks are shown to be resistant to random network failures. However, it has been observed [43, 44] that removal of highly connected nodes can significantly perturb and sometimes even cause a breakdown of the entire network. We will address this matter in Chapter 5.
Figure 1.6c shows an example of a scale-free network, where nodes with degree higher than average are marked in red color.

1.3 Analyzing real-world networks

Describing complex systems as abstract networks is a powerful tool [125], but even as abstractions the constructed networks remain highly complex. Reducing the complexity of networks and understanding their structure is based on the observation that most elements of the real-world systems are naturally grouped into categories. Books can be grouped according to their content, people can be grouped according to their occupation, living beings are grouped into species etc.. It would be of great importance to determine the grouping mechanisms in different complex systems, as they could enable discovering new elements of these systems, their unknown functions and relations between different elements.

1.3.1 Graph clustering

Natural system decomposition groups together elements with similar characteristics, so that the relationships between them is strong, whereas their connection to the elements from different groups is usually weak. This very intuitive definition has been formulated more precisely as a **clustering problem** in the field of data

mining. Clustering is the process of finding groups of data, called **clusters**, based on some similarity measure between the data elements [147]. In the context of networks, we consider a **graph clustering** problem where networks are coarse-grained into clusters of nodes, such that nodes belonging to one cluster are highly interconnected, but have relatively few connections to nodes in other clusters.

The problem of graph clustering is closely related to the problem of graph partitioning, where the task is to partition the network into smaller components given certain constraints [156, 38, 65, 59]. More precisely, given a number of partitions and their size, by minimizing a particular cost function standard graph partitioning (also know as cut-based) algorithms find disjoint sets of approximately the same size that form a full partitioning of the network. However, in most of the real-world examples the number of partitions is typically unknown in advance. Furthermore, the natural grouping of elements of the underlying system produces sets that are often of unequal sizes.
One of the oldest graph partitioning algorithms is **minimum-cut** algorithm. This algorithm is widely used in computer science in parallel and distributed computing for solving a problem of workload distribution [38]. A fixed amount of tasks (representing nodes of a workload graph) needs to be done on a certain number of processors (each corresponding to one graph partition), typically minimizing the overall runtime. As mentioned above, in many real-world networks we don't have these parameters in advance. The problem when setting the size of partitions to be free is that the trivial partitioning into one group provides the optimal solution of the cost function. In order to overcome this obstacle, several new approaches have been proposed, where the two most common are **RatioCut** [76] and **Ncut** [154]. However, both methods and their modifications need at least the approximate size of partition in advance.

In contrast to graph partitioning algorithms, the number of clusters and their sizes are not needed in advance when solving graph clustering problems. More precisely, the aim of graph clustering algorithms is to find

Definition 13 ***Modules*** *(also called clusters, communities) are connected subgraphs of the network where all the nodes belonging to a single module are highly interconnected while having relatively sparse connections to the remaining nodes in the network.*

It has been observed that dense connections between nodes of one module imply functional relationships between elements of the underlying system [70, 130, 140]. This discovery introduces a new way for analyzing and understanding the organization of many real-world systems. For example, modules may correspond to protein complexes with the same function, group of Web pages with the same subject or ecological stepsisters etc. In this sense, finding modules can help decomposing the complex network structure into functional sub-units that can be analyzed in more detail in subsequent stages. Some of the common methods for module identification will be presented in more detail in Section 4.5.

Standard graph clustering approaches aim at full decompositions of the network into clusters, such that every node belongs to exactly one cluster. This means that values of assignment functions, representing the probability of a node to belong to a cluster, can be either 0 or 1. However, many real-world systems are characterized by existence of elements that do not clearly belong to any particular group. This situation can be illustrated for example in a social network. In particular, let us consider a political affiliation network, where two individuals are linked if they have the same political opinion and groups of individuals forming clusters have the same political affiliation. In this sense, almost everyone in one group is linked to almost everyone in the same group, but to no one of the individuals from the other groups, with the exception of several persons that connect groups by being linked to some members of both groups. These persons interconnect the clusters, but belong to neither, as they have no clear political affiliation.
Motivated by this observation, we introduce

Definition 14 ***A hard*** *clustering of a graph is a partitioning in which every node has to be assigned to exactly* ***one*** *cluster.* ***Soft*** *or* ***fuzzy*** *clustering of a graph allows that nodes can be assigned to more than one cluster with some probability.*

In this thesis, we will describe a new algorithm that identifies modules by means of soft clustering. Its **fuzzy** or **soft** assignment functions represent the probability of a node to belong to a certain cluster. Thus, they can take values between 0 and 1. Setting a certain threshold value θ, e.g. $\theta = 0.9$, we can distinguish between nodes that belong to a certain cluster with a high probability, meaning higher than θ. In order to avoid confusion between the approach to be presented and standard clustering approaches, we will here use the term ”modules” instead of ”clusters”, referring to nodes that belong to a particular set with a probability higher than θ.

Example 6 *Figure 1.7 shows an example network with* 50 *nodes. Using our fuzzy clustering method (see Chapter 4),* 45 *nodes were divided into five modules, that are represented in five different colors. The five nodes that do not "strongly" ($\theta = 0.9$) belong to any module are marked as black nodes. These nodes form the inter-modular region.*

Different random-walker-based soft assignments have recently been discussed in the literature [50, 145, 106]. However, these approaches suffer from two essential drawbacks:

1. it is often difficult to determine the number of modules in the network;
2. the underlying approaches do not lead to efficient algorithms and therefore can not be applied to very large networks.

We will address these two problems in more detail later and propose a new approach that resolves them.

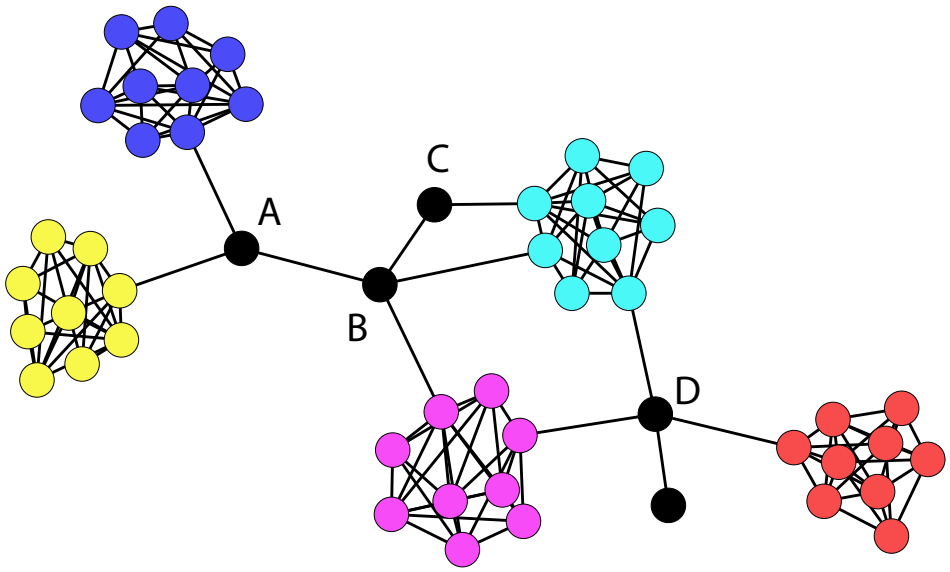


Figure 1.7: Network with 50 modules, 45 of them are arranged in five modules that are marked in different colors. Nodes marked with letters A, B, C and D are the nodes that do not "strongly" belong to any of the five modules.

1.3.2 Hubs in networks

A soft clustering approach will allow us to go beyond the above introduced problem of identifying modules. In particular, it introduces the **inter-modular** structure, consisting of nodes that do not belong to any of the network modules. Some of these nodes will play an important role for the network connectivity and as such will often correspond to the essential elements of the underlying system. We will refer to these elements as **inter-module hub** nodes. In the same way we will define **modular hubs** as nodes that belong to modules and are important for the network connectivity. The concept of hub nodes will be introduced in more detail in Chapter 5.
Figure 1.7 shows an example network with 50 nodes organized in five modules. Here we highlighted four inter-module hubs of the network and labeled them with A, B, C and D. For example, node A is important for the communication between the blue and the yellow module, but also for their communication with the rest of the network.

Identification of modular and inter-modular structures of the network could yield valuable information about the global network mechanisms. Furthermore, determining these structures could be used to understand dynamical processes taking place on the underlying system. For example, if we consider an air-transportation network, we can use its modular structure (where modules correspond to different continents for example) and existence of hubs (big airports having many intercontinental flights) to understand and control the spread of human diseases. Similar

processes are of importance in many other systems, such as spread of opinions and ideas in social systems. In this sense, discovering modules in the network and locating their essential connectors could provide crucial insights for controlling these processes.

1.3.3 Real-world networks

Many network representations of real-world systems are useful for understanding the main mechanisms and functions of the underlying system. The question is: how to find an appropriate network representation that would contain the essential characterization of the original system in its structure? In order to tackle this problem, we will now review network representations of three different real-world systems: social, biological and World Wide Web. Many review articles provide a more detailed overview of various real-world networks, their features and common approaches for their analysis [122, 6].

Social Networks

Understanding social interactions was always a task of great interest, as it mirrors the structure of a society and in particular, a position of its individuals in this society. Later, this became a topic of research in many fields like sociology, psychology, anthropology, philosophy etc.. Representing social interactions in the form of networks allowed introducing new tools for analyzing social systems, by methods coming from mathematics, physics and computer science [152, 163, 164, 105].
Different social networks are of interest, such as friendship networks, actor collaboration networks, political networks etc.. However, the strategy for creating these networks is always the same: given a social system of interest, nodes of the network correspond to individuals of this system and edges to the relationships between these individuals. Depending on the system, individuals can be for example, high school students, movie actors, researches, but also animals or fictional characters. An example of social networks are the so called friendship networks, where nodes of this network represent high school students and edges between them exist if they are friends. A movie actors network is formed in such a way that the nodes correspond to movie actors and the edges between two actors exist if they have been acting in the same movie.
Understanding social networks and their participants can enable extracting important information about the organization of the system. A natural phenomenon in social systems is the existence of community structures, each representing a group of individuals that are closely related in some way. This could be for instance a group of people with the same interests or people coming from the same country or people speaking the same language... In this sense, identification of modules in a network can discover special roles of individuals or specific subject of groups. Another important element of social systems are hub nodes. Hubs can correspond

to individuals who have leader roles in the system or in a particular module, like a person with many friends or an actor who collaborated with many of his colleagues. Hubs can also have a role of a connector, for example a person who speaks many languages and can therefore establish a communication between groups of people speaking different languages.
However, the main difficulties for analyzing social networks occur due to the quality of the initial data. Namely, the tests for collecting the relevant data are often subjective, incomplete and inaccurate. Furthermore, validating the obtained results is another critical issue, as the "correct" results are usually not known. In some cases, like Facebook, organization of the friendship network is known although not complete, but it is not available due to privacy policies.

Biological Networks

The analysis of biological networks is a very challenging and fascinating task, due to the complex organization and the diverse dynamic behaviors of the underlying systems. Of specific interest are biochemical networks, such as protein-protein interaction networks and metabolic networks, which can be used to understand basic cellular organization principals and molecular mechanisms of living organisms.
Protein-protein interaction networks (PPI) are one of the most studied types of biological networks [89, 75, 78]. They are built according to experiments that tested physical interactions between different proteins of an organism. The nodes of PPI networks are proteins and an edge between proteins implies that the two proteins can interact with each other. Protein-protein networks are commonly represented as undirected networks.
Metabolic networks reflect molecular metabolic mechanisms of a particular organism [136, 89]. The nodes represent substrates that participate in bio-chemical reactions which are presented by directed edges oriented towards a product of the reaction.
Another type of well studied biological networks are ecological networks [131] and in particular **food webs** [119]. Food webs are commonly represented as directed networks, in which nodes correspond to species of an ecosystem and edges represent predator-prey relationships between them.
Biological systems are often organized in network modules [70, 126, 135, 72], which represent a specific function, for example a specific synthesis pathway in a metabolic reaction network [136, 80]. A large number of biological networks, such as protein-protein interaction networks or metabolic networks are shown to be scale-free. As we mentioned before, this implies the existence of highly connected network hubs [15, 18], which may serve as central distributing elements or linkage points for many regions of a network [18, 89]. For example, in an early study by Fell and Wagner the authors found the metabolites with the highest degree to be the core of *E. coli* metabolism [64]. Another study found that the ranking of the most connected metabolites is largely identical for all organisms [89, 90].

World Wide Web (WWW)

The World Wide Web is the largest currently available network, having already at the end of 1999 close to one billion nodes. The nodes of WWW are Web pages and directed edges are hyperlinks that point from one document to another. Complex organization of this network, no natural ordering and its constant growth makes the WWW be one of the most studied networks [7, 16, 35]. Its topological properties, such as the small-world property and power-law in- and out-degree distributions, suggested the existence of certain structural network elements that can be of interest for the underlying system. For example, it has been shown that hub nodes correspond to the most vulnerable points of the WWW, as their removal can destroy the links towards many other Web pages and therefore, can cause the breakdown of some parts of the network [43, 44].

Chapter 2

Random walks on undirected networks

During the last years, there has been increasing interest in studying random walks, as they can be used to model processes coming from various fields. The path of a dust particle in a room, stock market fluctuations and surfing on the Internet are only some of the examples that can be described using random walks.

The pioneering steps in studying the basis of random walks go back to the research of the Scottish botanist Robert Brown in 1828. He observed the motion of the pollen particles in water and discovered that this motion is governed by irregular drift. However, the first formal definition of the random walk problem was published only in 1905 by Pearson, as a part of his discussion with Lord Rayleigh:

> *"A man starts from a point* 0 *and walks* l *yards in a straight line; he then turns through any angle whatever and walks another* l *yards in a second straight line. He repeats this process* n *times. I require the probability that after* n *of these stretches he is at a distance between* r *and* $r + \delta r$ *from his starting point."*

Following this discussion, many scientists such as Einstein, Schmoluckovski, Markoff and others contributed to setting up what is now considered to be the basis of stochastic processes. Since then, random-walk-based methods have been playing an important role not only in probability theory, but also in physics, economics, chemistry, biology, computer science, etc..

In this chapter we will refer to the method of random walks on networks, as the method that has been well-established for structural analysis of complex networks [128, 69]. We will first introduce the theoretical background of the standard random walker based approach to networks in Section 2.1. Then in Section 2.2 we will establish the fundamental connections between the modular structure of the network and kinetic properties of the random walk process. We will see that this specific behavior of standard random walk process in modules does not always correspond only to modular network structures in the sense of their topological definition (See Section 1.3). This will motivate our new time-continuous random walk process, that will be introduced in Section 2.3.

2.1 Standard random walk

In Section 1.1, a simple, connected, undirected, unweighted graph $G = (V, E, w)$, is defined with: the set V of n nodes, the set of edges between the nodes E and the non-negative edge weights $w(x, y)$

$$w(x,y) = \begin{cases} 1, & (x,y) \in E \\ 0, & (x,y) \notin E. \end{cases} \tag{2.1}$$

Now we can define the standard **time-discrete random walk** process on a graph G, with discrete time steps $n = 1, 2, ...$ at which the random walker moves. Initially the random walk process starts in a certain node $u_0 \in V$. Then, at each time step he moves from the current node to one of its neighbors that has been chosen uniformly at random. More precisely, being in node x the walker will jump next to one of his neighbors y, with **transition probability**

$$p(x,y) = \frac{w(x,y)}{d(x)}, \tag{2.2}$$

where $d(x)$ is the degree of a node x (1.4).
The sequence of nodes visited by the random walk process defines a Markov chain $\{X_n \in V, n \in \mathbb{N}\}$, i.e. a time-discrete stochastic process on discrete state space, that satisfies the **Markov property**

$$\mathbb{P}[X_{n+1} = x | X_n = x_n, \ldots, X_2 = x_2, X_1 = x_1] = \mathbb{P}[X_{n+1} = x | X_n = x_n]. \tag{2.3}$$

That is, being at the present node, the future and past jumps are independent from each other. Therefore, the choice of the next state depends entirely on the current state. If the transition probability does not depend on the actual time step n, that is

$$\mathbb{P}[X_{n+1} = y | X_n = x] = \mathbb{P}[X_n = y | X_{n-1} = x] = \ldots = \mathbb{P}[X_2 = y | X_1 = x],$$

then the associated Markov chain is called **time-homogeneous** and is characterized by the **transition function** $P : V \times V \to [0, 1]$

$$P(x,y) = \mathbb{P}[X_{n+1} = y | X_n = x], \forall x, y \in V, \tag{2.4}$$

that governs the transitions from one state to another. In the following, we will consider only time-homogeneous Markov chains.

2.1.1 Properties of random walks on networks

Every Markov chain defines via its transition function P (2.4), the **one-step transition matrix**

$$P = (p_{xy})_{x,y \in V}, \quad \text{with entries} \quad p_{xy} = p(x,y). \tag{2.5}$$

This matrix is a stochastic matrix, since

$$p(x,y) \geq 0, \forall x,y \in V \quad \text{and} \quad \sum_{y \in V} p(x,y) = 1.$$

One important task in the theory of Markov chains is to determine the probability that after k steps a Markov chain is in state y. First, let us calculate the k-step transition probability from x to y for $k \geq 0$, that is the probability to be in state y after k steps conditional on initially starting in state x

$$p_{xy}^{(k)} = \mathbb{P}[X_{k+1} = y | X_1 = x], \quad \text{if} \quad \mathbb{P}[X_1 = x] > 0.$$

Now, using the basic properties of Markov chains, we can associate the k-step transition probabilities to the matrix of k-step transition probabilities as

$$P^{(k)} = P^k, k \geq 0, \quad \text{where} \quad P^{(0)} = I. \tag{2.6}$$

This implies that in order to calculate the k-step transition matrix, it is sufficient to the have the one-step transition matrix P and raise it to the power of k. Furthermore, another important relation is given by Chapman-Kolmogorov equation (or semigroup property)

$$P^{k+n}(x,y) = \sum_{z \in V} P^k(x,z) P^n(z,y), \ \ \forall k,n \geq 0.$$

A probability distribution π that controls the choice of an initial state

$$\pi(x) = \mathbb{P}[X_1 = x],$$

is called the **initial distribution**. Now, we can determine the distribution of the Markov chain while it evolves over time. More precisely, if a Markov chain is initially distributed according to π, we can calculate the probability to be in node y after one step

$$\begin{aligned} \mathbb{P}[X_2 = y] &= \sum_{x \in V} \mathbb{P}[X_1 = x] \mathbb{P}[X_2 = y | X_1 = x] \\ &= \sum_{x \in V} \pi(x) P(x,y), \end{aligned} \tag{2.7}$$

and more general, the probability to be in node y in k-th step

$$\begin{aligned} \mathbb{P}[X_k = y] &= \sum_{x \in V} \mathbb{P}[X_{k-1} = x] \mathbb{P}[X_k = y | X_{k-1} = x] \\ &= \sum_{x \in V} \pi(x) P^k(x,y). \end{aligned}$$

Many structural properties of networks influence a nature of random walks on these networks and vice versa. In the following, we will point out some fundamental properties. For a complete overview we refer to [107] and [129].

- A state y is **accessible** from a state x via a random walk process, if

$$\exists n \geq 1, \quad P^n(x, y) > 0, \tag{2.8}$$

A Markov chain is **irreducible** if all of its states are accessible from all other states. In terms of networks, node y is accessible from node x if there exist a path from node x to node y. If there exists a path from every node to every other node, then the network is **connected**. The following result connects these two properties.

Proposition 2
A Markov chain associated to a random walk (2.2) is irreducible, if and only if, the underlying undirected network is connected.

- For a state x, we define its **period** as

$$\text{period}(x) = \gcd\{n \geq 1, P^n(x, x) > 0\},$$

where *gcd* is the greatest common divisor. If $\text{period}(x) = 1$, then the state x is said to be **aperiodic**. A Markov chain is aperiodic if all its states are aperiodic. In particular, a network is said to be aperiodic if the gcd of the lengths of all cycles in the networks is equal to 1. Undirected, aperiodic graphs are called **non-bipartite**. Especially, the following holds

Proposition 3
A Markov chain associated to a random walk (2.2) is aperiodic, if and only if the underlying graph is non-bipartite.

- A special class of problems are induced by the above introduced type of networks. Namely, a random walk on an undirected, connected, non-bipartite network defines an **ergodic Markov chain**. Ergodic Markov chains are of special interest in the theory of Markov processes, as they are useful from an algorithmic perspective. This issue will be addressed in more details in the next section.

2.1.2 Invariant measure and reversibility

For a given Markov chain $(X_n)_{n\in\mathbb{N}}$ with transition matrix P, a non-negative vector μ with $\sum_x \mu(x) = 1$ and

$$\mu P = \mu, \tag{2.9}$$

is called the **invariant measure** or the **stationary distribution** of that Markov chain. From (2.9) it follows that μ is the left eigenvector of P with respect to the eigenvalue $\lambda_1 = 1$. It is easy to check that the invariant measure of a random walk process defined as (2.2) is given by

$$\mu(x) = \frac{d(x)}{\sum_{y\in V} d(y)}. \tag{2.10}$$

If the walker starts μ-distributed, then it will again be μ-distributed after one step, that is

$$\mathbb{P}[X_2 = y] = \sum_{x \in V} \mathbb{P}[X_1 = x] p(x, y) = \sum_{x \in V} \mu(x) p(x, y) = \mu(y).$$

This means that the probability to be in a certain node is proportional to the degree of that node.

The following result states that regardless of the initial distribution, the ergodic random walk always converges to one distribution: the stationary distribution [34].

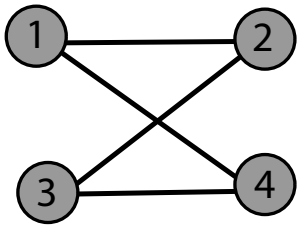
Theorem 1 (The fundamental theorem of Markov chains)
For a finite ergodic Markov chain with transition matrix P, there exists an unique stationary distribution μ such that

$$\lim_{n \to \infty} p^n(x, y) = \mu(y), \quad \forall x, y \in V.$$

Note that this limit does not depend on a starting state x. That is, no matter where the random walker starts on an undirected, connected, non-bipartite network; after long enough time n it will end up in a state y with probability $\mu(y)$. Hence, we will assume in the following that the Markov chain is initially distributed by the stationary distribution μ. The proof of this theorem is provided in many textbooks, for example in [34]. It is clear that for networks that are not connected this theorem doesn't hold, since there can exist several stationary states or even none. Let us show on the following example, why aperiodicity is a necessary condition for this theorem.

Example 7 *Let us consider a special class of graphs, namely **bipartite graphs**, introduced in Definition 4. Bipartite graphs are an example of graphs that are periodic, since their cycles have a length that is divisible by two. An example of a bipartite graph is shown in Figure 2.1, together with the transition matrix of the random walk defined on this graph. In every step the random walker will go from one side of the network to another side. That is, if the walk starts on one side its limiting distribution at time t will depend on the parity of t, so the random walk oscillates all the time and never converges. It is easy to check that a stationary distribution of this process is $\mu = (0.25, 0.25, 0.25, 0.25)$, which is not obtainable if we start for example with an initial distribution $\pi = (1, 0, 0, 0)$.*

Spectral properties of an ergodic Markov chain are described by the following theorem [153]:



(a) Bipartite graph

$$P = \begin{pmatrix} 0 & 0.5 & 0 & 0.5 \\ 0.5 & 0 & 0.5 & 0 \\ 0 & 0.5 & 0 & 0.5 \\ 0.5 & 0 & 0.5 & 0 \end{pmatrix}$$

(b) The transition matrix

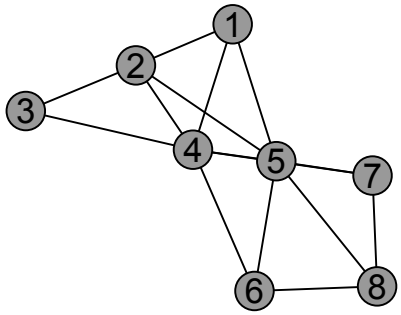
Figure 2.1: An example of a bipartite network with four nodes and its transition matrix P.

Theorem 2 (Perron-Frobenius theorem for ergodic Markov chains) *If $(X_t)_{t\in\mathbb{N}}$ is an irreducible, aperiodic Markov chain with $n \times n$ transition matrix P, then*

1. *P has a unique eigenvalue $\lambda_1 = 1$, with the corresponding right eigenvector $P\mathbb{1} = \mathbb{1}$, $\mathbb{1} = (1, \ldots, 1)$ and the left eigenvector $\mu P = \mu$, that has all positive entries.*
2. *All other eigenvalues of P are also real valued and strictly smaller (in modulus) than λ_1, i.e. $1 = \lambda_1 > |\lambda_2| \geq |\lambda_3| \geq \ldots \geq |\lambda_n|$.*

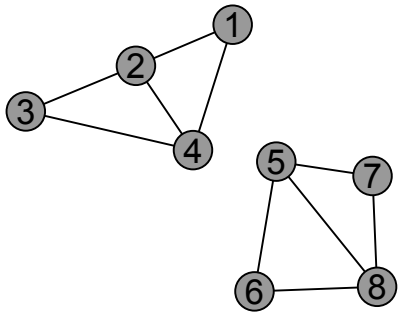
Example 8 *At this point we will show three examples in order to illustrate spectral properties of random walks on different types of undirected networks. Let us first consider an undirected-unweighted network with eight nodes, shown in Figure 2.2. This network is connected and aperiodic, so that the random walk process defined on this network is ergodic and Theorem 2 holds. Eigenvalues of its transition matrix are also shown in Figure 2.2.*
Our second example refers to a disconnected network shown in Figure 2.3. As discussed in Proposition 2, the random walk process defined on a disconnected network is not irreducible, so Theorem 2 doesn't hold. Therefore, the spectrum of the transition matrix P is characterized by the multiple maximal eigenvalue $\lambda = 1$, as it can be seen in Figure 2.3.
The third example studies the spectrum of an already introduced bipartite, undirected network from Example 7. From Proposition 3, it follows that the random walk process on this network is periodic and since the network is undirected, period$(x) = 2$, $\forall x \in V$. Moreover, the spectrum of the transition matrix of a random walk process on a bipartite graph is always characterized by its symmetry with respect to 0 and in particular by the eigenvalue $\lambda = -1$ [107]. In this particular example, the spectrum is

$$\lambda_1 = 1 \quad \lambda_2 = \lambda_3 = 0 \quad \lambda_4 = -1.$$



Eigenvalues of P	
λ_1	1
λ_2	0.5071
λ_3	0.0754
λ_4	0
λ_5	-0.1319
λ_6	-0.3333
λ_7	-0.4802
λ_8	-0.6371

Figure 2.2: An example of an unweighted, undirected network with eight nodes and the eigenvalues of its transition matrix P.



Eigenvalues of P	
λ_1	1
λ_2	1
λ_3	0
λ_4	0
λ_5	-0.3333
λ_6	-0.3333
λ_7	-0.6667
λ_8	-0.6667

Figure 2.3: An example of a disconnected unweighted, undirected network with eight nodes and the eigenvalues of its transition matrix P.

It is important to notice that for an undirected network, the symmetry of its adjacency matrix doesn't imply the symmetry of the transition matrix P. In particular, P is symmetric if and only if the original network is regular, i.e. if degrees of all nodes are the same. The symmetry of the adjacency matrix of the network implies that the random walk process is **reversible** in time

$$\mathbb{P}[X_n = x, X_{n+1} = y] = \mathbb{P}[X_n = y, X_{n+1} = x],$$

i.e. the random walk process running forward in time is equivalent to the process running backward in time. This easily follows from (2.1), (2.2) and (2.10)

$$\begin{aligned} \mathbb{P}[X_n = x, X_{n+1} = y] = \mu(x)p(x,y) &= \frac{d(x)}{\sum_{z\in V} d(z)} \frac{w(x,y)}{d(x)} \\ &= \mu(y)p(y,x) = \mathbb{P}[X_n = y, X_{n+1} = x]. \end{aligned} \tag{2.11}$$

This means that the random walk (2.2) is time-reversible with respect to its stationary distribution μ. Also, if a Markov chain is reversible with respect to π, then

π is a stationary distribution

$$(\pi P)(y) = \sum_{x \in V} \pi(x)P(x,y) = \sum_{x \in V} \pi(y)P(y,x) = \pi(y).$$

Reversibility of the Markov chain implies that the **detailed balance condition** holds

$$\mu(x)p(x,y) = \mu(y)p(y,x). \tag{2.12}$$

Remark 1 *In general we can define an irreducible Markov chain with transition matrix P and stationary distribution μ and the time-reversed Markov chain with transition matrix P^b. The connection between these two processes is*

$$\mu(y)P^b(y,x) = \mu(x)P(x,y).$$

Especially, if the Markov process is time-reversible we have that $P = P^b$. We will refer to this point again later in this Chapter.

An interesting interpretation of equation (2.12) can be given based on the notion of probability flux [39]. The quantity $\mu(x)P(x,y)$ is called the **probability flux** from x to y, that is the amount of probability mass flowing from x to y. Then, equation (2.12) states that for every pair of states (x,y) the flux from x to y is equal to the flux from y to x, that is a flux is locally balanced between each pair of states. We can also define a probability flux between two sets $A, B \subset V$, as

$$\text{flux}(A,B) = \sum_{x \in A} \mu(x)P(x,B) = \sum_{x \in A} \sum_{y \in B} \mu(x)P(x,y),$$

and due to reversibility prove the global conservation of flux between two sets. Therefore, when $B = A^c$ it always holds that

$$\text{flux}(A, A^c) = \text{flux}(A^c, A), \quad A \subset V, \tag{2.13}$$

hence the probability flux from A to its complement A^c and the flux from A^c to A are equal. Especially, when we observe only one state $x \in V$, i.e. $A = \{x\}$, this means that the total flux leaving a state x is the same as the total flux that goes into state x.

2.1.3 Transfer operator

As stated above, we assume that a given Markov chain is ergodic, so that μ is its unique invariant measure. Let us introduce the μ-weighted Hilbert space

$$L^2_\mu = \{f : V \to \mathbb{R} | \sum_{x \in V} f(x)^2 \mu(x) < \infty\},$$

where the scalar product, the induced 2-norm and the 1-norm are defined as follows:

$$\langle f, g \rangle_\mu = \sum_{x \in V} f(x)g(x)\mu(x), \qquad \|f\|^2_\mu = \langle f, f \rangle, \qquad \|f\|_{1,\mu} = \sum_{x \in V} |f(x)|\mu(x). \tag{2.14}$$

We define the **transfer operator** P on L^2_μ that describes a propagation of densities in L^2_μ

$$(Pf)(y)\mu(y) = \sum_{x\in V} \mu(x)p(x,y)f(x) \tag{2.15}$$

The reason to switch to the space L^2_μ is that since the detailed balance condition (2.12) holds, we have that

$$\langle u, Pv\rangle_\mu = \sum_{x,y\in V} u(x)p(x,y)v(y)\mu(x) = \sum_{x,y} p(y,x)u(x)v(y)\mu(y) = \langle Pu, v\rangle_\mu.$$

i.e. P is **self-adjoint** in L^2_μ. Therefore, the spectrum of P is real-valued and can be ordered as follows:

$$1 = \lambda_1 > \lambda_2 \geq \lambda_3 \geq \ldots \geq \lambda_n$$

where λ_j are the eigenvalues,

$$Pu_j = \lambda_j u_j, \quad j = 1, \ldots, n.$$

The associated eigenvectors u_j form an orthonormal basis of L^2_μ, i.e. $\langle u_j, u_k\rangle_\mu = \delta_{jk}$. Notice that the largest eigenvalue $\lambda_1 = 1$ corresponds to the eigenvector $u_1 = \mathbb{1}$ and is unique. Especially, $u_1 = \mathbb{1}$ is the invariant measure in L^2_μ.

Remark 2 *Note that up to now P has the following meanings:*

1. *the transition function corresponding to the given Markov chain (2.4),*
2. *the stochastic matrix (2.5),*
3. *the transfer operator (2.15).*

These three objects provide the equivalent expressions of the same fact. To this end, let us observe the spectral properties of the operator P and the transition matrix P. From 2.15, we have that

$$(Pu)(x) = \sum_{y\in V} \frac{\mu(y)}{\mu(x)} p(y,x)u(y) = \sum_{y\in V} p(x,y)u(y) = (P\cdot u)[x],$$

where for the second equality we used the detailed balance condition. Again, the expression on the left hand side represents the action of the operator on the function and the last expression is a matrix multiplication, where $[x]$ refers to the x^{th} element of the resulting vector. Now, it is easy to see that both objects have the same eigenvalues and eigenvectors.

Since the eigenvectors of P form an orthonormal basis in L^2_μ, we can write for $v \in L^2_\mu$

$$v = \sum_{j=1}^{n} \langle v, u_j\rangle_\mu u_j.$$

Thus, we can write the kth-power of P in the form

$$P^k = \sum_{j=1}^{n} \lambda_j^k \langle u_j, \cdot \rangle_\mu \, u_j. \tag{2.16}$$

Since P^k describes the k-step transition probabilities of the random walk process,

$$\mathbb{P}[X_k = y | X_0 = x] = P^k(x, y),$$

equation (2.16) means that the eigenvalues λ_j imply the timescales $T_j = 1/|\log \lambda_j|$ of all relaxation processes of the random walker on the network, starting with the trivial timescale $T_1 = \infty$ on which the random walker relaxes to its invariant measure, via the slowest non-trivial scale T_2 to shorter and shorter relaxation timescales.
If some eigenvalues, say $\lambda_1, \ldots, \lambda_m$, are particularly close to 1 (i.e. significantly closer to 1 in modulus than all others), then the associated timescales are very long and significantly longer than all other relaxation timescales. These eigenvalues are called leading or **dominant eigenvalues**. Moreover, right eigenvectors corresponding to the dominant eigenvalues capture the large-scale behavior of the random walk, whereas eigenvectors with smaller eigenvalues contain the small-scale behavior. We see that the basic properties of the random walk process are determined by the spectrum of the transition matrix.

2.1.4 Random walks on weighted networks

Let us now extend the definition of random walk processes to weighted, undirected networks. As explained above, in the case of undirected, unweighted networks at each time step the random walker moves from some node x to a node chosen uniformly at random among neighbors of x. However, if a given network is weighted this choice is not uniform, but proportional to the weight of the edge connecting the two nodes. Then, the transition probability is defined as in (2.2)

$$p(x, y) = \frac{w(x, y)}{d(x)}, \quad d(x) = \sum_{y \in V} w(x, y)$$

where now $d(x)$ is the *weighted degree* of node x (see Definition 7). The properties of random walk processes on unweighted networks from the beginning of the Section 2.1 can be extended and applied also to undirected, weighted networks. Especially important is that Perron-Frobenius theorem (see Theorem 2) holds for weighted networks, so that for ergodic Markov chains the eigenvalues of P are real valued and $1 = \lambda_1 > \lambda_2 \geq \lambda_3 \geq \ldots \geq \lambda_n$. Furthermore, it can be easily shown that random walk processes on weighted, undirected networks are also time-reversible.

In the following we will consider unweighed, undirected networks in order to operate only with $0 - 1$ edge weights, which provides a much clearer calculation than when dealing with arbitrary edge weights. However, one should have in mind that our results can be easily generalized to the case of weighted, undirected networks.

2.2 Modules and metastable sets

In Section 1.3 we have defined modules as specific structural elements of networks. More precisely, modules are subgraphs in which nodes are densely connected to each other, but have only a few edges to nodes outside of that module. Identification of modules in networks has many important advantages, especially when applying to real-world systems. We have already mentioned their potential use in analyzing social and biological networks (see Section 1.3.3). After defining a random walk process on a network, the obvious questions to ask are:

- What are modules in terms of the random walk process?
- How does the existence of these structures influence the behavior of the random walk on the network?

The answers to these questions will connect dynamical properties of the random walk to the structural properties of the network and justify the usage of the random walker approach in the structural analysis of complex networks.

Let us introduce the notion of the transition probability between two sets A and B

$$p(A,B) = \mathbb{P}(X_2 \in B | X_1 \in A) = \frac{1}{\mu(A)} \sum_{x \in A, y \in B} \mu(x) p(x,y),$$

i.e. the probability that the walker, after having started in the set $A \subset V$ distributed according to the invariant measure μ, will be found in the set $B \subset V$ after one step. Modules, being defined by the property of being connected internally more densely than externally, can thus be described as subsets $M_1, \ldots, M_m \subset V$ of the nodes for which:

(1) the transition probability $p(M_i, M_j)$ from the module M_i to some other module $M_j, j \neq i$ is significantly small, i.e. $p(M_i, M_j) \approx 0$;

(2) the residence probability in every module M_i, $p(M_i, M_i)$ is close to 1, that is $p(M_i, M_i) \approx 1$.

The first condition describes the communication between modules, namely that modules of a network are well separated in the sense that jumps between them are rare. On the other hand, the second condition is connected to the behavior of the random walk process inside a module: a random walk tends to get trapped in a module for a very long time. In Markov chain theory these sets are called **metastable sets** [49, 50]. Our main idea is that modules of a network correspond to metastable sets of a random walk process on that network [150, 88, 53, 144]. Hence, identification of modules in a network is directly connected to the behavior of the random walk process on its longest time scales. In Section 2.1 we have shown that the longest time scales are encoded in the dominant eigenvalues of the transition matrix P. Because of this, the number of metastable sets (modules) can be determined from the number of dominant eigenvalues of P. Furthermore, the

eigenvectors corresponding to the dominant eigenvalues of P can indicate how to decompose the network into modules, according to the change of sign of eigenvectors. This strategy is a crucial idea of **spectral clustering** methods. We will explain this idea in Chapter 4, together with the strategies for identification of modules. Properties of metastable sets will be studied in more detail in Chapter 3.

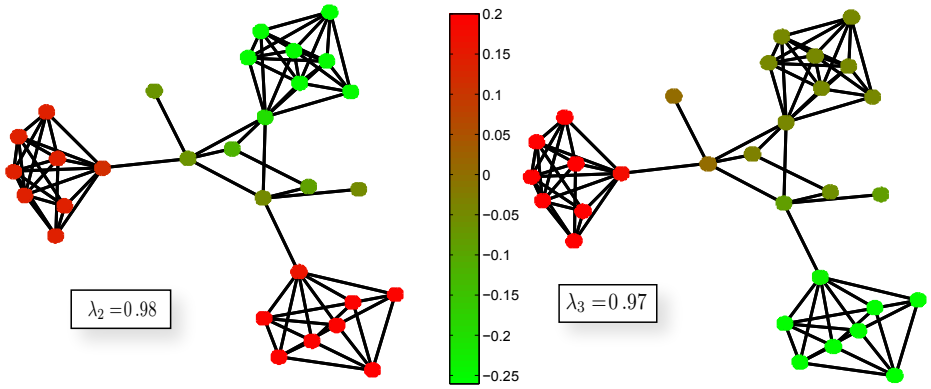


Figure 2.4: Network with 30 nodes, where 24 nodes belong to three modules. The transition matrix P has three dominant eigenvalues: $\lambda_1 = 1$, $\lambda_2 = 0.98$ and $\lambda_3 = 0.97$. This plot shows the two eigenvectors corresponding to the dominant eigenvalues λ_2 and λ_3, respectively.

Example 9 *Let us now illustrate how to determine the number of modules in given networks, as metastable sets of the random walk process. In Figure 2.4 we show an example network with* 30 *nodes. The transition matrix P has the following eigenvalues, sorted in the descending order*

$$\lambda_1 = 1 \quad \lambda_2 = 0.98 \quad \lambda_3 = 0.97 \quad \lambda_4 = 0.76 \quad \lambda_5 = 0.41 \ldots$$

We see a clear spectral gap after the third eigenvalue, which implies that there are three dominant eigenvalues, i.e. three modules. Since the eigenvector corresponding to the first eigenvalue λ_1 is constant, the eigenvectors corresponding to λ_2 and λ_3 indicate which nodes belong to which network module.

However, in many cases the spectrum of the transition matrix P does not provide a clear answer about the number of modules and their structure. The next example will demonstrate this problem and explain the reasons why this happens.

Example 10 *Figure 2.5 shows a network that has* 200 *nodes, of which only* 80 *are in one of the two modules. This network consists of the large loosely connected area, taking place on the left part of the network. Due to its structure, this area is characterized by the appearance of metastability, since the random walker spends a long time in it. As a consequence, the spectrum of P has no clear spectral gap, which can be seen in the plot of the first* 20 *eigenvalues of P. This doesn't necessarily*

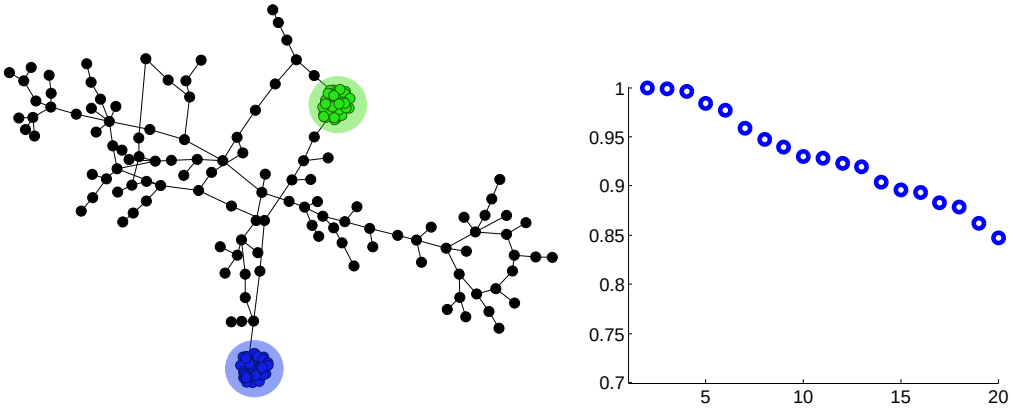


Figure 2.5: Left: Network with 200 nodes, where 80 nodes belong to the two modules colored in blue and green. Right: The first 20 eigenvalues of the transition matrix P.

exclude the usage of spectral methods for decomposing networks into modules, but in many cases it does decrease the quality of the obtained solutions.
Furthermore, the spectrum of P is characterized by negative eigenvalues close to -1. This effect is caused by the existence of alternating (bipartite) structure in the network, such as the long chain on the right part of the network, see Definition 4. When the standard random walker enters a structure like this, it gets trapped in it for a long time. Therefore, these type of structures can be very metastable, although they do not correspond to modules in the topological sense, as in Definition 13.

Regarding the outlined drawbacks of the standard random walker approach, we will introduce in the next section a new approach that overcomes these problems.

2.3 Time-continuous random walks

Let us introduce a new class of random walks on networks, as processes that will overcome the problem of identifying non-modular substructures of networks as metastable sets of the process.

2.3.1 Time-continuous Markov processes

Our new approach is based on employing time-continuous random walks on networks, i.e. time-continuous Markov processes $(X_t)_{t\in\mathbb{R}^+}$ on the finite, discrete state space V, that satisfy the Markov property

$$\mathbb{P}[X_{t_{n+1}} = x | X_{t_n} = x_n, \ldots, X_{t_2} = x_2, X_{t_1} = x_1] = \mathbb{P}[X_{t_{n+1}} = x | X_{t_n} = x_n],$$

where $0 \leq t_1 \leq t_2 \leq \ldots \leq t_{n+1}$ and $x_1, \ldots, x_n, x \in V$.
We will consider a Markov process $(X_t)_{t\in\mathbb{R}^+}$ and its associated family of transition

matrices $\{P(t) = (p_t(x,y))_{x,y}, t \geq 0\}$, with entries

$$p_t(x,y) = \mathbb{P}[X(t) = y | X(0) = x],$$

such that there are no transitions happening at time $t = 0$

$$p_0(x,y) = \delta_{xy} = \begin{cases} 1 & \text{if } x = y; \\ 0 & \text{if } x \neq y. \end{cases}$$

As a necessary regularity condition, we assume that transition probabilities are continuous at time $t = 0$

$$\lim_{t \to 0^+} p_t(x,y) = \delta_{xy}, \ \forall x, y \in V.$$

In other words, realizations of the process are right-continuous functions of time, i.e. step-functions. Because of this, time-continuous Markov processes are also called **Markov jump processes**, where transitions from one state to another correspond to jumps in step functions.
Like the transition matrix of a Markov chain, also the family of transition matrices $\{P(t), t \geq 0\}$ of a Markov process satisfies the Chapman-Kolmogorov equation

$$P(s)P(t) = P(s+t), \ s, t \geq 0,$$

with $P(0) = \mathrm{Id}$. The main difference between this setting and the one for Markov chains is that the time-discrete process $(X_n)_{n \in \mathbb{N}}$ was fully characterized by one transition matrix P, whereas the Markov jump process is described by a family of transition matrices $\{P(t), t \geq 0\}$. The reason for this lies in the fact that Markov chain can be seen as a sequence of snapshots of the time-continuous process, for a certain fixed time step t.

Regarding this, the interesting questions that arise are the following

- What are the advantages of using Markov jump processes?
- How can we describe the behavior of the process at every point in time?
- How quickly does the process jump from one state to another?

Addressing these questions is connected to observing the infinitesimal changes of transition probabilities, that can be expressed by

$$L = \lim_{t \to 0^+} \frac{P(t) - \mathrm{Id}}{t}, \tag{2.17}$$

where L is the **rate matrix** of the Markov jump process $(X_t)_{t \in \mathbb{R}}$ with entries

$$l(x,y) \geq 0 \quad \forall x \neq y \quad \text{and} \quad l(x,x) = -\sum_{y \in V \setminus x} l(x,y). \tag{2.18}$$

A rate matrix is the basic object in the theory of Markov jump processes, since it entirely characterizes the dynamics of a given process.

More precisely,

- $l(x,y), x \neq y$ represents the **transition rate**, i.e. the average number of transitions from x to y per time unit;
- $-l(x,x)$ corresponds to the rate of leaving node x, called **escape rate**.

Escape rates are one of the crucial novelties of Markov jump processes compared to Markov chains, since they introduce the concept of waiting times of a process in states [129]. This means that the time-continuous Markov process jumps between states, but also spends some time "waiting" in states. The expected time the process spends in a state x is $\frac{1}{|l(x,x)|}$. This relation motivates the introduction of our new random walker approach and will be explored in more details in 2.3.2.
The rate matrix L is also called the **infinitesimal generator** of a Markov process, since it can generate the whole family of transition matrices

$$P_t = \exp(tL), t \geq 0. \tag{2.19}$$

For a fixed timescale t this equation establishes the connection between the generator of a time-continuous process and the transition matrix of its embedded Markov chain. Therefore, it enables us to obtain properties of a Markov jump process observed at a certain time step t. In the case of Markov chains, we will often use the so called **discrete generator** L_d

$$L_d = P - \mathrm{Id}, \tag{2.20}$$

as an object that mimics some of the properties of the infinitesimal generator L, namely (2.18). However, it is important to notice that the discrete generator clearly does not have the same meaning as the infinitesimal generator, in the sense of representing the underlying dynamics of a process.

In analogy to the transfer operator P defined by equation (2.15), we introduce on L^2_μ

$$(\mathcal{L}f)(y)\mu(y) = \sum_x l(x,y)f(x)\mu(x) \tag{2.21}$$

and for the discrete case $\mathcal{L}_d = P - Id$, in terms of the operator P.

Remark 3 *We can observe the dynamics of the process (X_t) running backward in time. Then, $\mathcal{L}$ refers to the generator of the **time-reversed process** and especially $\mathcal{L}_d$ is the discrete generator of the time-reversed process. As explained above, $\mathcal{L}$ or in time-discrete setting $\mathcal{L}_d$, generate the family of transition matrices (P^b_t) with entries*

$$p^b_t(y,x) = \frac{\mu(x)}{\mu(y)} p_t(x,y).$$

Obviously, when the process is time-reversible (2.12), we have

$$p^b_t(x,y) = p_t(x,y), \tag{2.22}$$

so the process running backward in time is equivalent to the process running forward in time. Then, from the detailed balance condition (2.12) it follows that $L = \mathcal{L}$, that is in time-discrete case $L_d = \mathcal{L}_d$.

2.3.2 The new random walker approach

Here, we will introduce the new random walker approach as a family of time-continuous Markov jump processes, defined by its generator

$$L_p(x,y) = \begin{cases} -\frac{1}{d(x)^p}, & x = y \\ \frac{k(x,y)}{k(x)d(x)^p}, & x \neq y, (x,y) \in E \\ 0, & \text{else} \end{cases} \tag{2.23}$$

where $p \in \mathbb{R}$ and $k(x,y) \geq 0$ are weights such that $k(x,y) = 0$ if $(x,y) \notin E$, $k(x,y) = k(y,x)$ and $k(x) = \sum_y k(x,y)$.
From Section 2.3.1, it follows that a time-continuous random walk is characterized by the waiting time in nodes of the network. In terms of a random walk process this means that if the random walker is in node x, the expected waiting time in this node is proportional to its degree $d(x)^p$. That is, the more neighbors a node has, the longer it takes the random walker on average to decide where to go next. Defined in this way, the process becomes faster in simple regions, which are loosely connected, and in more complicated, densely interconnected structures the process becomes slower. The parameter p obviously refers to the level of metastability we want to take into account, namely increasing the parameter p increases $d(x)^p$, i.e. the expected waiting time. For example, let us observe two random walk processes for $p = 1$ and $p = 2$ on a given modular network. In the process with the generator L_2 network modules are much more metastable than for the process with the generator L_1. The choice of p, as well as the other parameters that determine the random walk, will depend also on the topological properties of the given network. This issue will be discussed at the end of this section.

Following the theory of Markov processes, let us now state some of the main properties of time-continuous random walk processes defined above.

Proposition 4

The invariant measure of a Markov jump process defined with (2.23) is

$$\mu(x) = \frac{1}{Z} d(x)^p k(x), \tag{2.24}$$

where Z is a normalization constant.

Proof. We have to show that $L_p^T \mu(y) = 0, \forall y \in V$, so

$$\begin{aligned} L_p^T \mu(y) &= \sum_{x \in V} L_p(x,y)\mu(x) = \frac{1}{Z} \sum_{x \in V} L_p(x,y) d(x)^p k(x) \\ &= \frac{1}{Z} \Big(\sum_{\substack{x \neq y \\ (x,y) \in E}} \frac{k(x,y)}{k(x)d(x)^p} d(x)^p k(x) + L_p(y,y) d(y)^p k(y) \Big) \\ &= \frac{1}{Z} \Big(\sum_{\substack{x \neq y \\ (x,y) \in E}} k(x,y) - k(y) \Big) = 0. \end{aligned}$$

□

Proposition 5
A Markov jump process defined with a generator L_p from (2.23) is reversible.

Proof. We have to prove that the detail balance condition holds

$$\mu(x)L_p(x,y) = \mu(y)L_p(y,x), \forall x, y \in V.$$

This follows trivially when $x = y$. The same applies for $x, y \in V$, such that $(x,y) \notin E$ since then $L_p(x,y) = 0 = L_p(y,x)$. Now, the only possibility is that $x \neq y$ and $(x,y) \in E$. We now apply (2.24) and obtain

$$\mu(x)L_p(x,y) = \frac{1}{Z}d(x)^p k(x)\frac{k(x,y)}{k(x)d(x)^p} = \frac{k(x,y)}{Z} = \frac{k(y,x)}{Z} = \mu(y)L_p(y,x).$$

□

Time reversibility of the process X_t implies that the generator L_p is self-adjoint in L^2_μ. Therefore, the spectrum of the generator is real-valued and its eigenvectors are orthonormal. If the network is connected and non-bipartite, the largest eigenvalue $\Lambda_1 = 0$ has multiplicity one and the whole spectrum can be ordered as

$$0 = \Lambda_1 > \Lambda_2 \geq \Lambda_3 \geq \ldots \geq \Lambda_n.$$

For fixed timescale t, we will consider the transition matrix $P = P_t$ of the Markov chain, where $P = \exp(tL)$ according to (2.19) and $L = L_p$, for a particular p. The eigenvalues of the transition matrix P are given with

$$\lambda_i = \exp(\Lambda_i t), \tag{2.25}$$

and can be ordered as as follows:

$$1 = \lambda_1 > \lambda_2 \geq \lambda_3 \geq \ldots \geq \lambda_n \tag{2.26}$$

with an orthonormal system of eigenvectors $\{u_1, \ldots, u_n\}$ in L^2_μ:

$$Pu_i = \lambda_i u_i \qquad \langle u_i, u_j \rangle = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \tag{2.27}$$

The properties of the family of random walks introduced in (2.23) depend on the given parameters and the given network. To illustrate this, let us consider the following cases:

(1) In the simplest case we introduce

$$L(x,y) = \begin{cases} -\frac{1}{d(x)}, & x = y \\ \frac{1}{d(x)^2}, & x \neq y, (x,y) \in E \\ 0, & \text{else.} \end{cases} \tag{2.28}$$

i.e. $k(x,y) = a(x,y)$, $p = 1$ and $k(x) = d(x)$. For this process the expected waiting time in a node equals the degree of that node and the invariant measure is given as $\mu(x) = \frac{1}{Z}d(x)$. Another important property is that the choice of the node to jump to is made uniformly at random, from the set of neighbors of a node. The embedded Markov chain of this time-continuous random walk is exactly the standard random walk, defined in (2.2).
However, in some cases from the properties of the process defined as in (2.28) we will not be able to discover all modular structures in the network. For example, let us consider the network shown in Figure 2.6. Here, we highlighted the 8 modules in different colors. We see that between the red and the blue module no inter-module nodes exist. Especially, since the nodes in both modules have almost the same degree, the random walker defined as in (2.28) doesn't distinguish between them, i.e. "sees" them as one module. The next choice of the random walk parameters will overcome this problem.

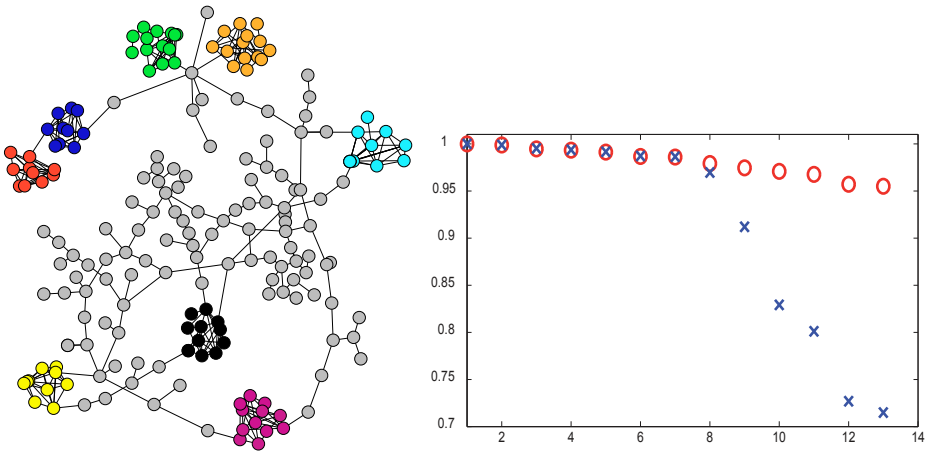


Figure 2.6: Example network with 200 nodes and the first 13 eigenvalues of standard random walk (red circles) and time-continuous random walk (blue crosses), based on (2.29).

(2) Another choice of the random walk is for $p = 1$ and

$$k(x,y) = a(x,y) \cdot \Big(1 + \langle a_x, a_y \rangle\Big), \tag{2.29}$$

where a_z is the zth row of the adjacency matrix and $\langle \cdot, \cdot \rangle$ is the usual Euclidean scalar product. Here, a jump from x to y is not random, but depends on the similarity of the neighborhood of two nodes. This dependence can be seen better when using a local clustering coefficient c (see Definition 11), that is defined by

$$c(x) = \sum_{y,z \in V} \frac{a(x,y)a(x,z)a(y,z)}{d(x)(d(x)-1)}.$$

As explained in Section 1.1, the clustering coefficient measures the connectedness of the neighborhood of x. By choice of (2.29), we obtain

$$\begin{aligned} k(x) &= \sum_{y \in V} a(x,y) \cdot (1 + \langle a_x, a_y \rangle) \\ &= \sum_{y \in V} a(x,y) + \sum_{y \in V} a(x,y) \cdot \langle a_x, a_y \rangle \\ &= d(x) + c(x)(d(x) - 1) \end{aligned}$$

that enters the invariant measure of the random walk such that

$$\mu(x) = \frac{1}{Z} d(x)^2 \Big(1 + (d(x) - 1) \cdot c(x) \Big). \tag{2.30}$$

This shows that the nodes with high degree and high clustering coefficient become very attractive for the random walker. Let us demonstrate the advantage of this process compared to the previous one by comparing the eigenvalues corresponding to the transition matrix of the time-continuous random walk (2.29) with the one from the standard random walker approach (2.2). The first 13 eigenvalues of both matrices are shown in Figure 2.6. The spectrum of the standard transition matrix does not offer a clear gap to give an idea about the number of modules being present. On the other hand, the time-continuous random walk shows a strong gap after 8 eigenvalues, indicating the existence of 8 modules.

(3) In the case of weighted networks, we can generalize the previous approaches by choosing $k(x,y) = w(x,y)$, where w denotes the weights of edges.

In the following chapters we will use time-continuous random walk processes for analyzing complex networks and identifying important structural components of these networks, such as modules and hubs, that can correspond to important elements of the underlying system.

Chapter 3

Modules and metastability

In this chapter we will investigate metastability, an important phenomenon that occurs in different complex systems coming from various applications [23, 101] for example, chemical reactions, thermodynamical systems, climate systems, financial systems or ecological systems. Nevertheless, for all these examples metastability has the same interpretation, namely it refers to the property of the process to stay in a certain part of the state space for a long period of time, until it goes to some other part of state space, where it will stay again for a long period of time.

It has been observed that large bio-molecules such as proteins and enzymes exhibit different behavior on different time scales. The short time scales, from femtoseconds to picoseconds are characterized by fast oscillations and fluctuations of the amino acid side chains around some metastable state. On the other hand, transitions between different metastable states occur on longer time scales, that are in the range of microseconds to milliseconds. In the context of bio-molecules, metastable states are called *conformations* and correspond to spatial structures of bio-molecules. It is of great importance to identify conformations of a certain bio-molecule and the transitions between these conformations, so-called conformational changes, since they determine functions of such molecules. For example, protein folding can be seen through different conformations from unfolded to folded state, where the transitions from one state to another are of crucial importance for the functioning of the protein. These transitions occur on long time scales and are rare events, compared to the fluctuations within each conformation. Therefore, the main goal is to extract properties of the process on metastable sets, at long time scales.

In order to achieve this, we have to give rigorous answers to the following questions

- What is the formal definition of metastability?
- What are metastable sets of a given process?
- How can we identify metastable sets of a process?

To this end, in Section 3.1 we will introduce the basic objects from the theory of Markov processes [129], which will play the fundamental role in characterizing metastable processes and metastable sets. Based on these objects, in Section 3.2, we

will give an interpretation of metastable sets in the context of random walker process on modular networks. In Section 3.3 we will consider another characterization of metastability that is oriented towards the connection between the long time behavior of the observed Markov process, which can be described using the spectral properties of the generator (transition matrix) of the considered process. This approach will be used in the next chapter to provide effective means for identifying modules of a network, as metastable sets of the random walk process defined on a network.

3.1 Analyzing Markov processes

Until now we have discussed the importance of metastability of stochastic processes in various applications, as well as its intuitive definition in the case of Markov processes on networks. In this section we will provide mathematical characterization of metastability for Markov process on discrete state space V, that will hold for both time-continuous and time-discrete setting. In the following, we will refer to this by (X_t) for both cases.

3.1.1 Hitting times

For a given Markov process (X_t) and some set B, we define

Definition 15 (First hitting time) *The first hitting time τ_B of the process (X_t) into the set $B \subset V$, is defined as*

$$\tau_B = \inf\{t > 0 : X_t \in B\},$$

that is, the first time at which the process (X_t) enters the set B.
In addition, $\tau_B(x)$ is the first hitting time of the process into set B conditional on starting in state $x \in V$

$$\tau_B(x) = \inf\{t > 0 : X_t \in B, X_0 = x\}.$$

Since we consider only irreducible processes, the first hitting time is finite for any choice of set B. Note that τ_B is a random variable and particulary, τ_B is a stopping time, since it depends only on the behavior of the process (X_t) up until a time τ_B. The first hitting time describes dynamical properties of the process (X_t) and as such is used as one of the crucial elements in studying metastable processes. In this context, of practical interest is the expected value of the first hitting time, that is defined as

Definition 16 (Mean first hitting time) *For a fixed set $B \subset V$, the mean first hitting time $T_B(x)$ of a process is defined as the expected entry time of the process (X_t) into the set B, conditional on starting in node x*

$$T_B(x) = \mathbb{E}\tau_B(x).$$

The mean first hitting time can be computed from the generator of the process, namely from the generator L of the time continuous Markov process $(X_t)_{t\in\mathbb{R}}$ or from the discrete generator L_d of the underlying Markov chain $(X_n)_{n\in\mathbb{N}}$. For a fixed set B, T_B defines a function on the state space which is the minimal, non-negative solution of the discrete Dirichlet problem [129]

$$\begin{aligned} LT_B(x) &= -1, \quad x \notin B \\ T_B(x) &= 0, \quad x \in B. \end{aligned} \tag{3.1}$$

Since V corresponds to the discrete state space, T_B solves the system of linear equations (3.1).

Metastability of a process can also be seen via the fast jumps of a process between metastable sets. Therefore, it is important to study the dynamical properties of a process between given sets. To this end let us introduce **the mean first passage time** between two states and more general, between two sets of the state space.
For any pair of two different states (x, y), the mean first passage time $T(x, y)$ represents the expected time that the process (X_t) reaches a state y for the first time, conditional on starting in a state x. This definition can be generalized to the case of sets.

Definition 17 (Mean first passage time) *For two disjoint sets $A, B \subset V$, the mean first passage time $T(A, B)$ is the expected time at which the process started in a set A reaches a set B, that is*

$$T(A, B) = \frac{1}{\mu(A)} \sum_{x\in A} \mu(x) T_B(x), \tag{3.2}$$

where $\mu(A) = \sum_{y\in A} \mu(y)$.

Together with (3.1), this definition represents a functional method for describing the behavior of a process with respect to the given sets. Clearly, this will be of great importance for characterizing metastable processes.

3.1.2 Committor functions

From a probabilistic point of view an important question is: What is the probability that given two disjoint sets A and B, the process will enter first set A? In order to answer this question, we will use the fundamental object of the Transition Path Theory (TPT) [58, 117], i.e. the committor functions.
Let $C_1, ..., C_m \subset V$ be nonempty, disjoint sets. We assume that these sets don't form a full partition of V and call the region that is not assigned to any set **transition region** and denote it by

$$T = V \setminus \bigcup_{k=1}^{m} C_k.$$

The **forward committor** function $q_i^+ : V \to [0,1]$ is defined as the probability that a process (X_t) starting in x will visit the set C_i first, rather than any other set

$$q_i^+(x) = \mathbb{P}[\tau_{C_i}(x) < \tau_{M_i}(x)], \quad M_i = \cup_{j=1,j\neq i}^{m} C_j. \tag{3.3}$$

One can derive that q_i^+ is the solution of a linear system with boundary conditions [117]

$$\begin{aligned} (Lq_i^+)(x) &= 0, \quad \forall x \in T, \\ q_i^+(x) &= 1, \quad \forall x \in C_i, \\ q_i^+(x) &= 0, \quad \forall x \in C_j, j \neq i. \end{aligned} \tag{3.4}$$

Here L is the generator of the observed Markov process with entries $l(x,y)$ and $(Lq_i^+)(x) = \sum_y l(x,y) q_i^+(y)$. In particular, when (X_t) is a Markov chain, one has to replace L by the discrete generator L_d (2.20).
In a similar way, we define the **backward committor** as the probability that the process (X_t) came last from the set C_i, conditional on being in state x. The backward committor solves

$$\begin{aligned} (\mathcal{L}q_i^-)(x) &= 0, \quad \forall x \in T, \\ q_i^-(x) &= 1, \quad \forall x \in C_i, \\ q_i^-(x) &= 0, \quad \forall x \in C_j, j \neq i, \end{aligned} \tag{3.5}$$

where $\mathcal{L}$ refers to the generator of the time-reversed process with entries $l^b(x,y)$, see equation (2.21). As discussed in Remark 3, for the time-discrete case one has to replace $\mathcal{L}$ by $\mathcal{L}_d$. The two systems of equations (3.4) and (3.5) have a unique solution under the assumption that the invariant measure is unique and not vanishing on all of the sets [115].
We have shown in the previous chapter, that random walk processes defined on undirected networks are time-reversible. Therefore, (3.4) and (3.5) yield the identity of forward and backward committors, i.e.

$$q_i^- = q_i^+ \ \forall i = 1, ..., m. \tag{3.6}$$

Hence, in the following if the process is time-reversible we will use the shorthand notation $q_i := q_i^- = q_i^+$ and we will call this object the committor. Figure 3.1 shows the values of the committor function q_1, when going from module C_1 to C_2. The color of nodes indicate the value of the committor function, such that red colors corresponds to the values close to 1 and green for the values close to 0.

Remark 4 *In this Section, we have defined two different types of committors: forward and backward committors. Also, we distinguished whether the observed process is time-discrete or time-continuous Markov process. The natural question to ask is: What is the difference between the committor of the time-discrete process and the committor of the time-continuous process?*

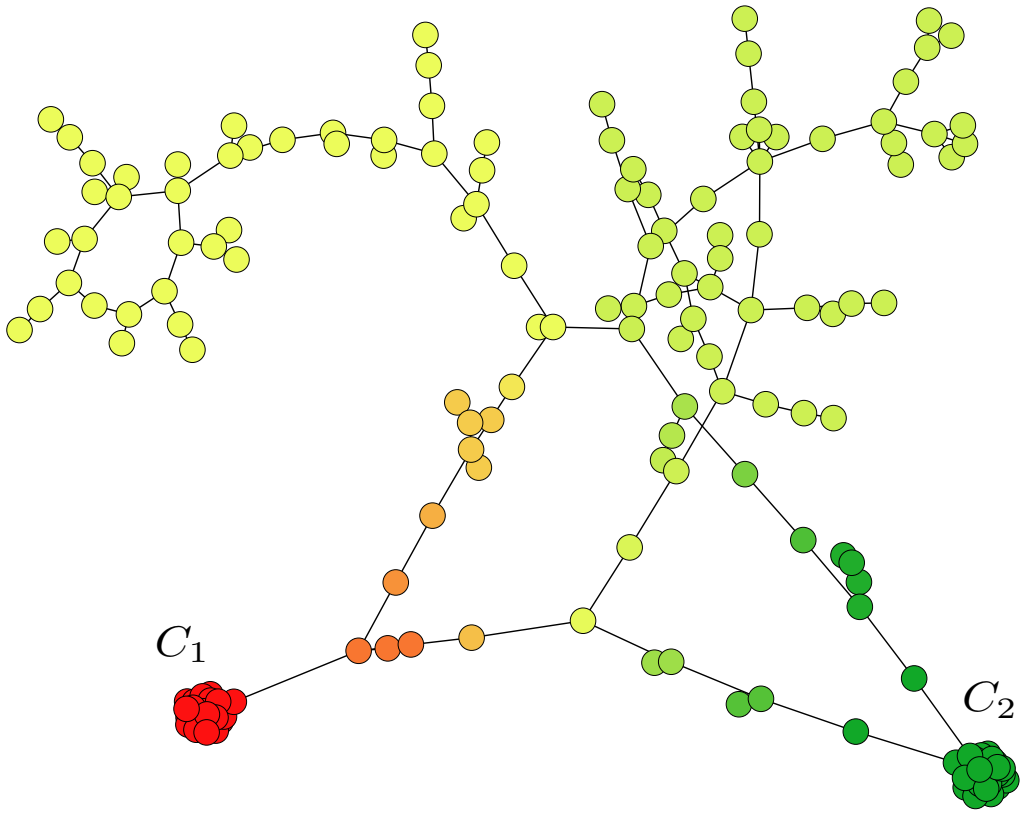


Figure 3.1: The values of the committor function q_1 on the undirected network with two modules, red color indicates values close to 1, whereas green color indicates values close to 0.

The main difference between these two objects lies in the nature of the underlying processes. Let us observe the time-continuous process $(X_t)_{t\in\mathbb{R}^+}$ and the time-discrete process $(X_{n\tau})_{n\in\mathbb{N}}$, obtained from the continuous process for the fixed lag time $\tau > 0$. The difference between these two processes is that they have different time-resolution. More precisely, in the values of the time-continuous committor functions are encoded properties of the time-continuous process at every point in time. On the contrary, in the values of the time-discrete committor functions are encoded only certain, discrete snapshots of the continuous trajectory. Therefore, these can not always provide the exact point when the process entered a set, but only recognize points inside of a set at time steps $n\tau$.

The interpretation of committors in the sense of fuzzy network partitioning will be given in Chapter 4. For more details on the properties of committors we refer to Chapter 5 and [57, 117, 116, 115].

3.2 Metastable sets of random walks

In Section 2.2 we established the connection between modules, as topological structures of a network and a random walk as a dynamical process defined on a network. In particular, modules correspond to metastable sets of the random walk process defined on the network, i.e. modules are sub-graphs where the random walker stays for a long period of time and, rarely, the random walker quickly switches to another set. Therefore, the problem of identifying modules amounts to the identification of metastable sets of the random walk process.
Let us observe the dynamical properties of the process (X_t) on the given collection of disjoint sets C_i, $i = 1, \ldots, m$, from the point of view of metastability. Metastable processes are processes that exhibit specific behavior on different timescales with respect to the specific sets. More precisely, metastable processes are characterized by the long residence time in sets C_i on one hand, and by quick jumps from one set to another, on the other hand. That is, a metastable process spends long time inside each of the sets and, compared to that, very short time outside of these sets. Let us now express these two types of behavior in terms of the mean hitting times and the mean first passage times:

- **The return time R** is the quantity that will be useful for characterizing the existence of quick jumps between the given sets. More precisely, if the process starts outside of sets $C_i, i = 1, \ldots, m$, then the return time represents the longest time of return to one of the sets. That is,

$$R = \max_{x \notin C} T_C(x), \quad C = \bigcup_{i=1}^{m} C_i.$$

- **The waiting time $\mathbf{W_i}$** of the process in the set C_i, $i = 1, \ldots, m$ denotes the time the process spends within the set C_i, i.e. the time between a transition from the set C_i to another set $C_j \in C \setminus C_i$

$$W_i = T(C_i, C \setminus C_i).$$

  The waiting time W of the process denotes the shortest time the process spends within each of the sets C_i namely, the minimal residence time in the given sets.

$$W = \min_{i=1,\ldots,m} T(C_i, C \setminus C_i).$$

Now using these quantities, we provide a first formal definition of metastable processes ([28])

Definition 18 *Markov process (X_t) is **metastable**, if there exist disjoint sets $C_i \subset V$, $i = 1, \ldots, m$, such that*

$$\frac{R}{W} = \frac{\max_{x \notin C} T_C(x)}{\min_{i=1,\ldots,m} T(C_i, C \setminus C_i)} \leq \rho \ll 1, \tag{3.7}$$

where $C = \bigcup_{i=1}^{m} C_i$ and ρ is the parameter that characterizes the degree of metastability of the process (X_t), with respect to the sets $C_i, i = 1, \ldots, m$. Sets C_i are called ***metastable sets*** *of the process (X_t).*

The interpretation of this definition agrees with the intuitive definition of metastability. Condition (3.7) considers the processes for which the return time R is small and the waiting time W is large, that is, the process spends short time outside the sets and long time inside of the sets.
It is important to notice that in general, Definition 18 has a disadvantage that it involves quantities that are not always easy to compute, namely the mean first passage times. Therefore, in the next section we will present analogous formulations of metastability, that employ quantities that are easier to compute. However, in the case of networks the generator is explicitly given and as outlined above, the calculation of the mean hitting times for given sets is easy and straightforward.

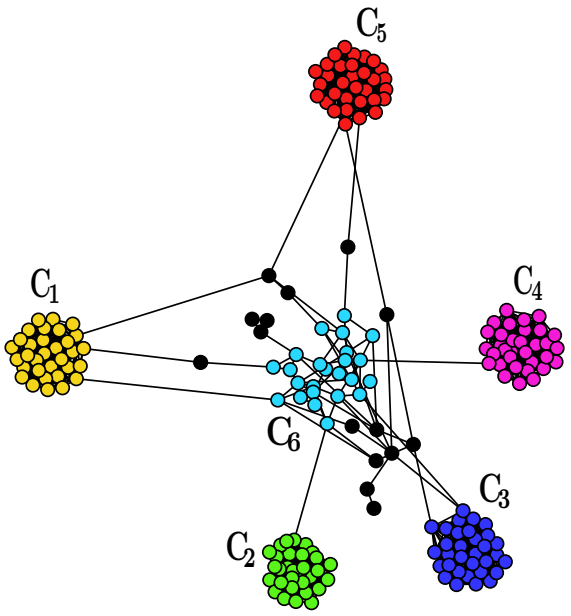
Let us now show in the following example the link between the mean first passage time and metastability of random walk process on modular networks and how these quantities change for different choice of modules in a network.

Example 11 *We consider the example network shown in Figure 3.2a with five and six modules. The choice of $m = 5$ modules consists of very densely connected modules $C_1, \ldots, C_5$, whereas in the case of $m = 6$ modules we include also the light blue module, whose nodes are not as densely connected as the nodes from the other modules. Table 3.2b shows the values of R/W_i for each of the modules C_i when $m = 5$ and $m = 6$. The small values of R/W_i for $m = 5$ indicate high metastability of the sets $C_1, \ldots, C_5$. In the case of $m = 6$ the metastability quotient takes a much higher value for the newly added module C_6, expressing that this module is not as metastable as $C_1, \ldots, C_5$.*

However, condition (3.7) is lacking the necessary precision, since "significantly small" will not be sufficient for an algorithmic identification of modules. In order to overcome this problem, in the next chapter we will develop a new approach for low-dimensional approximation of the long-term behavior of the random walk process on the network. This will result in an effective method for identification of modules, since as we have already discussed (see Section 2.2), the behavior of the random walker process on its longest time scales is directly connected to identification of modules in the network.

3.3 Spectral characterization of metastability

Metastability has been studied as a phenomenon in many different fields in various contexts. From the mathematical point of view several different methodologies have been developed [23], such as: large deviation method [67], asymptotic perturbation theory [82] and spectral methods [87]. Here we will present two approaches for



(a)

Module C_i	C_1	C_2	C_3	C_4	C_5	C_6
R/W_i	0.0015	0.0006	0.0017	0.0006	0.0012	/
	0.0003	0.0001	0.0004	0.0001	0.0002	0.18

(b)

Figure 3.2: Example network, where we marked in different colors the six modules C_1,..., C_6. Each row in the table shows how R/W_i changes for every module C_i depending on the choice of number of modules, five or six.

studying metastability that are both oriented towards the spectral properties of the generator (transition matrix) of the considered process.

3.3.1 Complete modular partition

Here we will present a relation between spectral properties of the transition matrix of the random walk process and full partitioning of a network into **metastable sets** of this random walk process [86]. That is, we will consider m sets, such that

$$C_i \cap C_j = \emptyset,\ i \neq j \quad \text{and} \quad \cup_{i=1}^{m} C_j = V. \tag{3.8}$$

First, let us fix a timescale t and consider the transition matrix $P = P(t)$ of the random walk X_t. Furthermore, let $\{\lambda_1, \ldots, \lambda_m\}$ be the m dominant eigenvalues of P and $\{u_1, \ldots, u_m\}$ the corresponding eigenvectors of P, for which (2.26) and (2.27) hold. For the rest of the spectrum, from [88] it follows that

$$\{\lambda_{m+1}, ..., \lambda_n\} \subset B_r(0) \subset \mathbb{C}, \quad r < \lambda_m.$$

The best possible partition of the network into modules has to maximize the joint metastability of the sets

$$\mathcal{D}(C_1, \ldots, C_m) = \sum_{i=1}^{m} p(C_i, C_i),$$

where $p(C_i, C_i)$ is the residence probability in set C_i. Markov chain theory [150], [88], [86] provides us with the following result for the lower and upper bound of the functional $\mathcal{D}$ for arbitrary partitions $C_1, \ldots, C_m$:

Theorem 3
The joint metastability of an arbitrary decomposition $C_1, \ldots, C_m$ of the state space is bounded from below and above by

$$\lambda_1 + \delta_2^2\lambda_2 + \ldots + \delta_m^2\lambda_m + c \leq \mathcal{D}(C_1, \ldots, C_m) \leq \lambda_1 + \lambda_2 + \ldots + \lambda_m, \qquad (3.9)$$

where $c = \lambda_{m+1}\left(1 - \delta_2^2 + \ldots + 1 - \delta_m^2\right)$ and δ_j is the error of the orthogonal projection Q with regards to the μ-weighted scalar product of the eigenvector u_j onto the space spanned by the characteristic functions $D = \mathrm{span}\{\mathbb{1}_{C_1}, \ldots, \mathbb{1}_{C_m}\}$ of the sets

$$\delta_j = \|Q^{\perp}u_j\|, \quad j = 2, \ldots, m. \qquad (3.10)$$

This theorem establishes the relation between a given decomposition of a state space into metastable sets and spectral properties of the transition matrix P. As outlined above, the best decomposition is the one that maximizes the joint metastability of the sets. According to this theorem, the metastability of an arbitrary decomposition into m sets cannot be larger than the sum of the m dominant eigenvalues of the transfer operator. Therefore, the maximal metastability is achieved when the lower bound is very close to the upper bound, that is when $\delta_j \approx 1$. This is achieved whenever the dominant eigenvectors are almost constant on the metastable sets. Using this, we can find the optimal or at least an almost optimal partition by minimizing the projection error δ_j. Many spectral clustering methods like PCCA+ [49], [50] exploit this idea to identify the optimal clustering.

3.3.2 Metastable hitting times

In this section we will present results that further relate metastability with spectral properties of the observed process [23, 28]. Especially, we will consider fuzzy partitioning of a network into metastable sets of a random walk process X_t that is given by a generator L, defined in equation (2.23). If $C_i, i = 1, \ldots, m$ are m metastable sets of this process, then when starting outside of the metastable sets C_i, the time the random walk process needs to enter some metastable set is very short compared

to the time the random walker spends in metastable states (see Section 3.2). The hitting time of metastable sets is often referred to as **metastable hitting time**. In the previous section, we introduced maximal metastable hitting time, the so-called return time R, that was used to characterize quick jumps between the given metastable sets. The following theorem [23] can be used to relate the return time to $C = \bigcup_{i=1}^{m} C_i$ to the spectral properties of the process:

Theorem 4
Let L be a generator of a random walk process (X_t) and $C_1, \ldots, C_m$ an arbitrary fuzzy decomposition of a given network. If S is an orthogonal projection on the space $D = \{v | \, v = 0 \text{ on } C\}$ and $\tilde{\Lambda}_1$ the smallest eigenvalue of the projected generator SLS, then

$$|\tilde{\Lambda_1}| \geq \frac{1}{\max_{x \notin C} \mathbb{E}\tau_C(x)}, \quad \textit{where} \quad C = \bigcup_{i=1}^{m} C_i.$$

Interpretation of this theorem is the following: if C_i, $i = 1, \ldots, m$ are chosen in such a way that they represent metastable sets of the random walk process, then the maximal mean hitting time to some of the sets C_i is small. This is reflected in the spectrum of SLS through a large absolute value of the smallest eigenvalue of SLS.
Note that the dominant eigenvalues of the projected generator SLS endorse the metastabilities of the process outside the set C, unlike the eigenvalues from the approach in the previous section. Using this theorem, we can check if a given decomposition of the network correspond to metastable sets of the random walk process defined on this network.

Chapter 4

Identification of modules

Discovering network modules, as its essential structural components is a challenging task when analyzing real-world networks, as it can provide valuable information about the underlying system. Dense connections between nodes within the same module could reflect their functional similarities. For example in protein-protein interaction networks modules can correspond to groups of proteins having the same biological function [80, 136]. Therefore, extracting information about the modular organization of the network provides valuable insights for decomposing the system into functional units. Furthermore, similarities between nodes that belong to the same module could reveal unknown properties and functions of the system elements corresponding to these nodes. In this sense, module identification could deepen our understanding of the structural organization and complex mechanisms of the underlying system.

When looking into the relevant literature, most articles are concerned with complete partitioning of networks, that is, hard clusterings in which the modules form a full partition of the network and every node belongs to exactly one module. However, in many real-world networks there are nodes which cannot be assigned only to a particular module, but rather have an affiliation to several modules. In Chapter 1, we referred to this type of clustering as fuzzy or soft clustering and introduced its characterization by the affiliation functions that take the form of a probability that a node belongs to one of the modules.

In this chapter we will present our novel techniques for **fuzzy decomposition** of a network into modules, where modules represent **metastable sets** of the random walk process defined on the network (see Chapter 3). We start by introducing fuzzy affiliation functions that will provide a probability to assign a node to some module (see Section 4.1). In order to find these modules in Section 4.2 we will adopt our recent results regarding coarse graining of Markov processes [53, 141]. In Section 4.3 we will develop these ideas further for the case of reversible Markov processes, such as random walks on undirected networks. Based on these results, in Section 4.4 we will present our new algorithm for identifying modules, discuss its computational aspects and suggest an extension of this method that can be applied also to very large real-world networks. Finally, in Section 4.5 we will present several state-of-the-art algorithms for identifying modules.

4.1 Fuzzy affiliation functions

As outlined above, we are aiming at finding a soft decomposition of a network in m modules $C_1, \ldots, C_m$. Moreover, we are a interested in obtaining a fuzzy affiliation of the nodes to these modules, that is for every node x we specify its affiliation $f_i(x)$ to module i such that

$$f_i(x) \in [0,1] \quad \text{and} \quad \sum_i^m f_i(x) = 1. \tag{4.1}$$

If we assume that we have already identified the modules $C_i, i = 1, \ldots, m$, then there is a natural way to define this affiliation by learning from the random walk. To do this, we simply start the random walk in node x and see which module it will enter next. Then, we set the affiliation $f_i(x)$ to be the probability that the next module to be entered is C_i, i.e.

$$f_i(x) = \mathbb{P}[\tau_{C_i}(x) < \tau_{M_i}(x)], \qquad M_i = \cup_{j=1, j\neq i}^m C_j, \tag{4.2}$$

where $\tau_A(x)$ is the first hitting time of the set $A \subset V$ by the process (X_t), if started in x. These functions were introduced in Section 3.1.2 as the forward committor functions q_i^+. The committor functions $q_1^+, \ldots, q_m^+$ satisfy $q_i^+(x) \in [0,1]$ and form the partition of unity,

$$\sum_{i=1}^m q_i^+(x) = 1, \quad \forall x \in V.$$

We can interpret $q_i^+(x)$ as the natural walker-based probability of assignment of a node x to a module C_i. Especially, when the process is time-reversible, as the random walk process on undirected networks, we will use $q_i := q_i^+$. One can compute these affiliation functions very efficiently by solving sparse, symmetric and positive definite linear system (3.4).

Example 12 *Figure 4.1 shows a network with four modules C_1, C_2, C_3 and C_4. For every choice of C_i, $i = 1, \ldots, 4$ and set $M_i = C \setminus C_i$, we can compute the associated forward committor function q_i^+ with respect to the random walk process defined on this network. Nodes are colored according to the values of the committor functions, where red color represents values close to one and green color values close to zero. These values provide at the same time the affiliation of nodes to the appropriate module.*

Special case: Complete partition

When the modules are chosen such that they form a complete partition of the network (3.8), the definition of the committors directly yields

$$q_i(x) = \mathbb{1}_{C_i}(x). \tag{4.3}$$

That is, in the case of complete partition the committors are given by the characteristic functions on the modules. In the following, we will always refer to committors

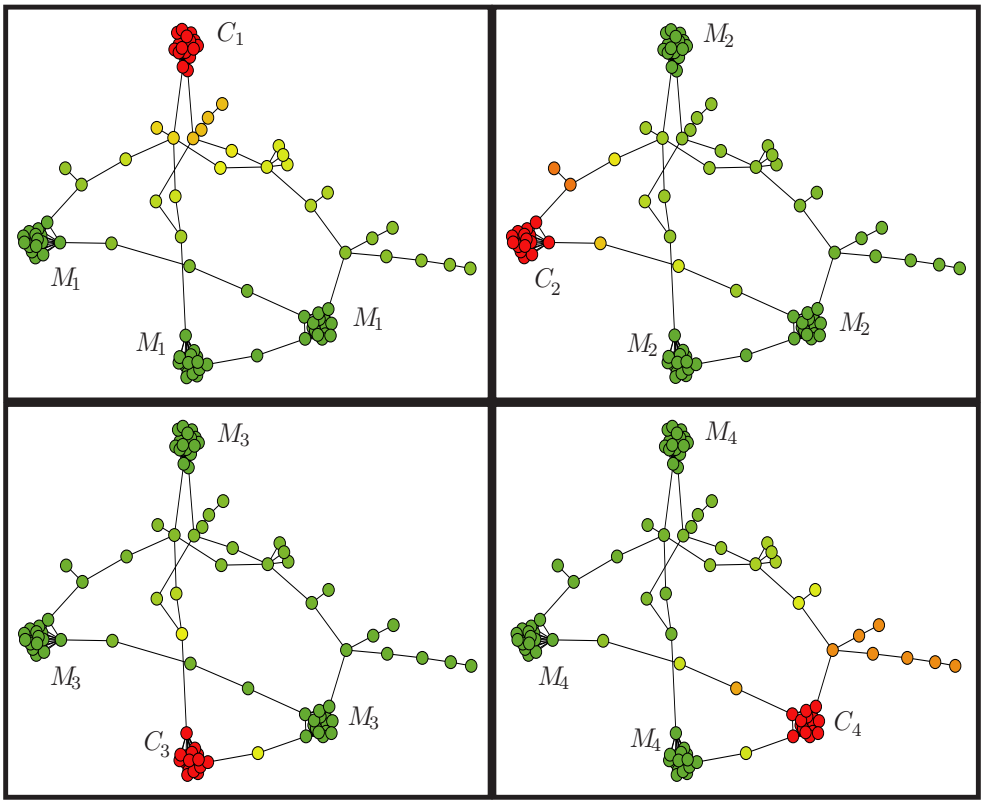


Figure 4.1: Example network with 100 nodes that are grouped in four modules. For every choice of C_i, $i = 1, \ldots, 4$ and M_i, colors of nodes correspond to the values of the associated committor function q_i^+.

as affiliation functions and consider complete partition a special case of fuzzy partition.

Fuzzy decomposition can be well interpreted in the sense of a coarse graining of our random walk by Markov State Modeling [51, 151, 53, 52, 49, 141]. This approach will be presented in more details in the following sections.

4.2 Fuzzy modular partition

In this section we will present a novel approach for identifying metastable sets of a random walk process defined on a network. We will follow the approach from [151] that uses Markov State Models (MSM) for finding low-dimensional structure-preserving approximations of metastable processes in the sense that the longest timescales of the dynamics of the this process are reproduced well [148, 49, 149, 150, 28]. While most standard MSMs approaches are based on a full decomposition of a state space and restricted to reversible processes, we will present here a general framework that considers fuzzy decomposition of state space and furthermore,

can be applied also to non-reversible processes, such as random walks on directed networks (see Chapter 7 for more details).

4.2.1 Milestoning process

For a given network, we can define a random walk as described in Chapter 2. We will differentiate later between time-continuous and time-discrete random walks. Let us now assume that the network is decomposed into m modules $C_1, \ldots, C_m \subset V$, that are disjoint and do not form a full partition

$$\cup_{i=1}^{m} C_i \neq V \quad \Rightarrow \quad T = V \setminus \cup_{i=1}^{m} C_i \neq \emptyset. \tag{4.4}$$

Then, this partition induces a coarse-grained random walk on the state space $\{1, \ldots, m\}$ that jumps from module to module. For analyzing the jump dynamics of the coarse-grained random walk process, we introduce the **milestoning process** $(\hat{X}_t)$ [63]

$$\hat{X}_t = i \Leftrightarrow X_{\sigma(t)} \in C_i, \text{with } \sigma(t) = \sup_{s \leq t} \left\{ X_s \in \bigcup_{k=1}^{m} C_k \right\}. \tag{4.5}$$

Equation (4.5) states that the milestoning process is in state i, if the original process came last from module C_i. That is, if the last module visited was C_i, we assign the walker to a module C_i as long as it has not entered another module. The transition behavior of the milestoning process is illustrated in Figure 4.2.

Remark 5 *Note that the **backward committor** defined in Section 3.1.2, is the probability that the process (X_t) came last from set C_i, conditional on being in state x. In terms of milestoning process that means*

$$q_i^-(x) = \mathbb{P}[\hat{X}_t = i | X_t = x],$$

i.e. the probability that the milestoning process is in state i, conditional that the original process started in state x.

4.2.2 Jump statistics of milestoning process

We will now study the transition behavior of the milestoning process for time-discrete and time-continuous random walk processes, respectively. Again, we will not restrict the underlying process to be time-reversible.

Time-discrete case: Transition probabilities

When observing a time-discrete process $(X_n)_{n \in \mathbb{N}}$ on the state space $\{1, \ldots, m\}$, we can define the transition matrix $\hat{P}$ of the milestoning process $(\hat{X}_n)_{n \in \mathbb{N}}$, with entries

$$\hat{p}(i, j) = \mathbb{P}_\mu(\hat{X}_{n+1} = j | \hat{X}_n = i).$$

In general, the milestoning process will not be a Markov process. Therefore, we cannot assume that it is essentially characterized by its transition matrix $\hat{P}$. This

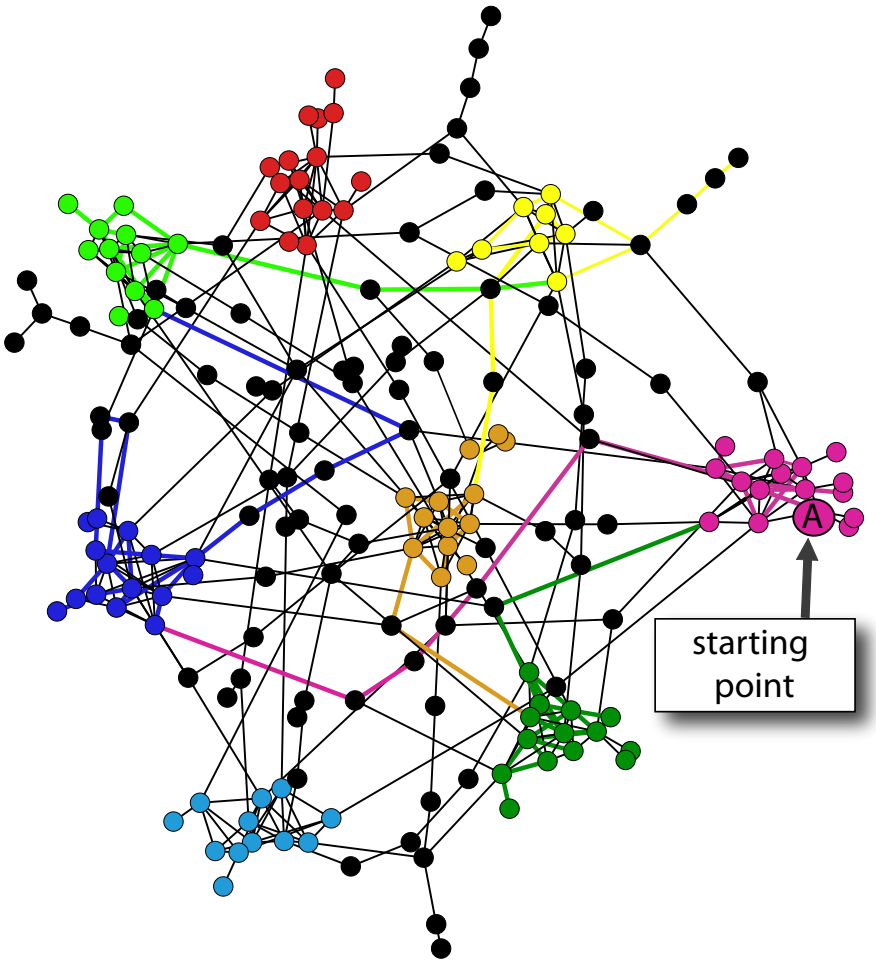


Figure 4.2: Illustration of a milestoning process on a network with 200 nodes, where 106 nodes are grouped into eight modules marked in different colors.

also holds for the discrete generator $\hat{L}_d$ (2.20). We will see that it is not crucial whether the dynamics of the milestoning process is governed by $\hat{P}$ or not.

Based on the introduced quantities we have

$$\mathbb{P}_\mu(\hat{X}_n = i, X_n = x) = \mathbb{P}_\mu(\hat{X}_n = i | X_n = x)\mathbb{P}_\mu(X_n = x) = q_i^-(x)\mu(x).$$

Therefore, the total probability that the milestoning process is assigned to state i, i.e. the invariant measure of the milestoning process, is

$$\hat{\mu}(i) = \mathbb{P}_\mu(\hat{X}_n = i) = \sum_{x \in V} \mathbb{P}_\mu(\hat{X}_n = i, X_n = x) = \sum_{x \in V} q_i^-(x)\mu(x) = \|q_i^-\|_1.$$

The following theorem gives us the entries of the discrete generator $\hat{L}_d$ (2.20).

Theorem 5
For a time-discrete process (X_n), the entries of the discrete generator $\hat{L}_d$ of the milestoning process $(\hat{X}_n)$ are given by

$$\hat{l}_d(i,j) = \frac{1}{\hat{\mu}(i)}\langle q_j^+, \mathcal{L}_d q_i^- \rangle. \tag{4.6}$$

Proof. Using that

$$\begin{aligned}\mathbb{P}_\mu(X_{n+1}=y, \hat{X}_n=i, X_n=x) &= \mathbb{P}_\mu(X_{n+1}=y|\hat{X}_n=i, X_n=x)\mathbb{P}_\mu(\hat{X}_n=i, X_n=x)\\ &= p(x,y)q_i^-(x)\mu(x),\end{aligned}$$

we can calculate

$$\begin{aligned}&\mathbb{P}_\mu(\hat{X}_{n+1}=j, X_{n+1}=y, \hat{X}_n=i, X_n=x) =\\ &= \mathbb{P}_\mu(\hat{X}_{n+1}=j|X_{n+1}=y, \hat{X}_n=i, X_n=x)\mathbb{P}_\mu(X_{n+1}=y, \hat{X}_n=i, X_n=x)\\ &= \begin{cases} \mathbb{1}_{C_j}(y)p(x,y)q_i^-(x)\mu(x), & \text{if } i\neq j\\ \mathbb{1}_{C_i\cup T}(y)p(x,y)q_i^-(x)\mu(x), & \text{if } i=j.\end{cases}\end{aligned}$$

Therefore, the one-step transition probability $\hat{p}(i,j)$ from i to $j\neq i$ is given by

$$\begin{aligned}\hat{p}(i,j) &= \mathbb{P}_\mu(\hat{X}_{n+1}=j|\hat{X}_n=i) = \frac{\mathbb{P}_\mu(\hat{X}_{n+1}=j, \hat{X}_n=i)}{\mathbb{P}_\mu(\hat{X}_n=i)}\\ &= \frac{1}{\mathbb{P}_\mu(\hat{X}_n=i)}\sum_{x,y\in V}\mathbb{P}_\mu(\hat{X}_{n+1}=j, X_{n+1}=y, \hat{X}_n=i, X_n=x)\\ &= \frac{1}{\|q_i^-\|_1}\sum_{x,y\in V}\mathbb{1}_{C_j}(y)p(x,y)q_i^-(x)\mu(x) = \frac{1}{\|q_i^-\|_1}\langle Pq_i^-, \mathbb{1}_{C_j}\rangle.\end{aligned}$$

In addition, when $i=j$

$$\begin{aligned}\hat{p}(i,i) &= \mathbb{P}_\mu(\hat{X}_{n+1}=i|\hat{X}_n=i) = \frac{\mathbb{P}_\mu(\hat{X}_{n+1}=i, \hat{X}_n=i)}{\mathbb{P}_\mu(\hat{X}_n=i)}\\ &= \frac{1}{\mathbb{P}_\mu(\hat{X}_n=i)}\sum_{x,y\in V}\mathbb{P}_\mu(\hat{X}_{n+1}=i, X_{n+1}=y, \hat{X}_n=i, X_n=x)\\ &= \frac{1}{\|q_i^-\|_1}\sum_{x,y\in V}\mathbb{1}_{C_i\cup T}(y)p(x,y)q_i^-(x)\mu(x)\\ &= \frac{1}{\|q_i^-\|_1}\langle Pq_i^-, \mathbb{1}_{C_i\cup T}\rangle.\end{aligned}$$

Using the properties of committors on sets for $i\neq j$, we get that

$$\begin{aligned}\langle Pq_i^-, \mathbb{1}_{C_j}\rangle &= \langle Pq_i^-, q_j^+\rangle - \langle Pq_i^-, q_j^+\mathbb{1}_T\rangle = \langle Pq_i^-, q_j^+\rangle - \langle q_i^-, q_j^+\mathbb{1}_T\rangle\\ &= \langle (P-Id)q_i^-, q_j^+\rangle = \langle \mathcal{L}_d q_i^-, q_j^+\rangle,\end{aligned}$$

which yields

$$\hat{l}_d(i,j) = \hat{p}(i,j) = \frac{1}{\|q_i^-\|_1}\langle q_j^+, \mathcal{L}_d q_i^-\rangle,\ i \neq j.$$

Similarly, for $i = j$ we get

$$\begin{aligned}\langle Pq_i^-, \mathbb{1}_{C_i \cup T}\rangle &= \langle Pq_i^-, \mathbb{1}_{C_i}\rangle + \langle Pq_i^-, \mathbb{1}_T\rangle \\ &= \langle Pq_i^-, q_i^+\rangle - \langle q_i^-, q_i^+ \mathbb{1}_T\rangle + \langle q_i^-, \mathbb{1}_T\rangle \\ &= \langle (P - Id)q_i^-, q_i^+\rangle + \|q_i^-\|_1 = \langle \mathcal{L}_d q_i^-, q_i^+\rangle + \|q_i^-\|_1,\end{aligned}$$

and

$$\hat{l}_d(i,i) = \hat{p}(i,j) - 1 = \frac{1}{\|q_i^-\|_1}(\langle q_i^+, \mathcal{L}_d q_i^-\rangle + \|q_i^-\|_1) - 1 = \frac{1}{\|q_i^-\|_1}\langle q_i^+, \mathcal{L}_d q_i^-\rangle.$$

□

Time-continuous case: Transition rates

We will now show that all the above identities are still valid in a time-continuous case.

Theorem 6
For a time-continuous process (X_t), the entries of a generator of the milestoning process $(\hat{X}_t)$ are given by

$$\hat{l}(i,j) = \frac{1}{\|q_i^-\|_1}\langle \mathcal{L} q_i^-, q_j^+\rangle. \tag{4.7}$$

Proof. In order to prove this theorem, we will use objects of the Transition Path Theory, that will be introduced in more details in Chapter 5.
For a given infinitely long trajectory and $i \neq j$, we define a (i,j)-reactive trajectory as a piece of this infinite long trajectory in a time interval R_{ij}^m such that for any $t \in R_{ij}^m$ we have that the next first entry into a set is in C_j, while the last first entry into a set happened in C_i. Then, at a certain time t we are on a (i,j)-reactive trajectory if

$$t \in R_{ij} = \cup_{m=-\infty}^{\infty} R_{ij}^m.$$

The probability current from x to y generated by (i,j)-reactive trajectories is then given by

$$f_{ij}(x,y) = \lim_{s \to 0+} \frac{1}{s}\mathbb{P}_\mu\Big(X_t = x, X_{t+s} = y, t \in R_{ij}, t+s \in R_{ij}\Big).$$

In order to compute this quantity we use that for $M_j = \bigcup_{k\neq j} C_k$, $\mathbb{P}_y[\tau_{C_j} < \tau_{M_j}]$ is the probability to start at y and enter the set C_j next rather than any other set. Therefore

$$\begin{aligned}
&\mathbb{P}_\mu\Big(X_t = x, X_{t+s} = y, t \in R_{ij}, t+s \in R_{ij}\Big) = \\
&= \mathbb{P}_\mu\Big(X_{t+s} = y, t+s \in R_{ij} | X_t = x, t \in R_{ij}\Big)\mathbb{P}_\mu\Big(X_t = x, t \in R_{ij}\Big) \\
&= \mathbb{P}_\mu\Big(X_{t+s} = y | X_t = x\Big)\mathbb{P}_y(\tau_{C_j} < \tau_{M_j})\mathbb{P}_\mu\Big(X_t = x, \hat{X}_t = i\Big) \\
&= p_s(x,y)q_j^+(y)q_i^-(x)\mu(x).
\end{aligned}$$

Since $i \neq j$ we have $l(x,y) = \lim_{s\to 0+} \frac{1}{s}p_s(x,y)$ and thus

$$f_{ij}(x,y) = l(x,y)q_j^+(y)q_i^-(x)\mu(x).$$

Now we can compute the rate k_{ij} of transitions from i to j, which is defined as the average number of (i,j)-reactive trajectories per unit time. This quantity is given by the total probability current through a dividing surface between C_i and C_j, i.e. by the total probability current generated by (i,j)-reactive trajectories through the boundary of C_i:

$$\begin{aligned}
k_{ij} &= \sum_{x\in C_i, y\in V\setminus C_i} f_{ij}(x,y) \\
&= \sum_{x\in C_i, y\in V\setminus C_i} q_j^+(y)l(x,y)q_j^-(x)\mu(x) \\
&= \sum_{x\in C_i, y\in V} q_j^+(y)l(x,y)q_i^-(x)\mu(x),
\end{aligned}$$

where the last identity results from $q_j^+(y) = 0$ for all $y \in C_i$. Since additionally $q_i^-(x) = 1$ for $x \in C_i$ we find

$$k_{ij} = \langle \mathcal{L}\mathbb{1}_{C_i}, q_j^+\rangle.$$

Therefore, the off-diagonal entries $\hat{l}(i,j)$ of the generator for the milestoning process $\hat{X}_t$ are

$$\hat{l}(i,j) = \frac{1}{\|q_i^-\|_1}\langle \mathcal{L}\mathbb{1}_{C_i}, q_j^+\rangle, \tag{4.8}$$

such that the diagonal entries have to be

$$\begin{aligned}
\hat{l}(i,i) &= -\sum_{j\neq i}\frac{1}{\|q_i^-\|_1}\langle \mathcal{L}\mathbb{1}_{C_i}, q_j^+\rangle = -\frac{1}{\|q_i^-\|_1}\langle \mathcal{L}\mathbb{1}_{C_i}, \sum_{j\neq i} q_j^+\rangle \\
&= -\frac{1}{\|q_i^-\|_1}\langle \mathcal{L}\mathbb{1}_{C_i}, \mathbb{1} - q_i^+)\rangle = \frac{1}{\|q_i^-\|_1}\langle \mathcal{L}\mathbb{1}_{C_i}, q_i^+\rangle.
\end{aligned}$$

Since $\langle \mathcal{L}\mathbb{1}_{C_i}, q_i^+\rangle = \langle \mathcal{L}q_i^-, q_i^+\rangle$, we can use the same arguments as above to end up with

$$\langle \mathcal{L}\mathbb{1}_{C_i}, q_j^+\rangle = \langle \mathcal{L}q_i^-, q_j^+\rangle,$$

which proves the theorem. □

4.3 Reversible Markov processes

Since random walk processes on undirected networks are reversible, in this section we will focus on reversible Markov processes and milestoning processes that are induced by reversible Markov processes. First we note that the milestoning process defined in 4.5 and associated to the time-reversible Markov process is also time-reversible.

Proposition 6
Let (X_t) be a reversible Markov process with unique invariant measure μ. Then the milestoning generator $\hat{L}$ has the invariant measure

$$\hat{\mu}(i) = \sum_{x \in V} q_i(x)\mu(x)$$

and the according operator in $L^2(\hat{\mu})$

$$(\hat{\mathcal{L}}v)(j)\hat{\mu}(j) = \sum_{i=1}^{n} \hat{l}(i,j)v(i)\hat{\mu}(i)$$

is self-adjoint. Therefore it also defines a reversible jump process.

Proof. We have

$$\sum_{i=1}^{m} \hat{l}(i,j)\hat{\mu}(i) = \sum_{i=1}^{m} \langle q_i, \mathcal{L}q_j \rangle = \langle \mathbb{1}, \mathcal{L}q_j \rangle = 0.$$

Moreover,

$$\begin{aligned} \hat{l}(i,j)\hat{\mu}(i) &= \langle q_i, \mathcal{L}q_j \rangle \\ &= \langle \mathcal{L}q_i, q_j \rangle = \hat{l}(j,i)\hat{\mu}(j), \end{aligned}$$

which implies reversibility and self-adjointness. □

4.3.1 Generalized eigenvalue problem

Following the idea of MSM, we want to approximate the dynamics of (X_t) by its projection to some low-dimensional subspace $D \subset L^2_\mu$. Here we consider subspace $D = span\{q_1, ..., q_m\}$ with $\mathbb{1} \in D$, i.e. the invariant measure with density $\mathbb{1}$ in L^2_μ is still contained in D. The basis functions q_i are assumed to be linearly independent, non-negative functions, which need not to be orthogonal w.r.t. $\langle \cdot, \cdot \rangle_\mu$ and are not necessarily identical with the committor functions discussed in Section 4.1. The orthogonal projection Q onto D can be written as

$$Qv = \sum_{i,j=1}^{m} S^{-1}_{ij} \langle v, q_i \rangle q_j, \tag{4.9}$$

with $S_{ij} = \langle q_i, q_j \rangle$. Now, we want to compare the operator P and its projection on D, namely QPQ. The following theorem tells us more about the structure of the operator QPQ [53, 141]:

Theorem 7
Let P be the transfer operator of the random walk process and Q the orthogonal projection onto the space spanned by committors $D = \{q_1, ..., q_m\}$ with respect to m modules. Then, $\hat{P}M^{-1}$ is a matrix representation of QPQ, where

$$\hat{P}_{ij} = \frac{\langle Pq_i, q_j\rangle}{\hat{\mu}(i)}, \quad M_{ij} = \frac{\langle q_i, q_j\rangle}{\hat{\mu}(i)} \text{ and } \hat{\mu}(i) = \sum_{x\in V} q_i(x)\mu(x). \tag{4.10}$$

Proof. Since the vectors q_i are linearly independent the symmetric matrix

$$S_{ij} = \langle q_i, q_j\rangle \tag{4.11}$$

is invertible and we can write the orthogonal projection Q onto subspace D as

$$Qv = \sum_{i,j=1}^{m} S_{ij}^{-1}\langle v, q_i\rangle q_j. \tag{4.12}$$

For the matrix M from (4.10) we have

$$M_{ij} = \frac{1}{\hat{\mu}(i)}S_{ij} \Rightarrow M_{ij}^{-1} = \hat{\mu}(j)S_{ij}^{-1}. \tag{4.13}$$

Now, take the basis $\{\psi_1, ..., \psi_m\}$ of D, $\psi_i = \frac{1}{\hat{\mu}(i)}q_i$. Then,

$$Qv = \sum_{i,j=1}^{m} M_{ij}^{-1}\langle v, q_i\rangle\psi_j. \tag{4.14}$$

This implies

$$\begin{aligned} QPQ\psi_k = QP\psi_k &= \sum_{i,j=1}^{m} M_{ij}^{-1}\langle P\psi_k, q_i\rangle\psi_j \\ &= \sum_{i,j=1}^{m} M_{ij}^{-1}\frac{\langle Pq_k, q_i\rangle}{\hat{\mu}(k)}\psi_j = \sum_{i,j=1}^{m} M_{ij}^{-1}\hat{P}_{ki}\psi_j \\ &= \sum_{j=1}^{m}(\hat{P}M^{-1})_{kj}\psi_j. \end{aligned} \tag{4.15}$$

That is, $\hat{P}M^{-1}$ is a matrix representation of QPQ with respect to the basis $\{\psi_1, ..., \psi_m\}$. □

From this theorem it immediately follows that we can compute the eigenvalues of the projected transfer operator QPQ by solving the generalized eigenvalue problem

$$\hat{P}r = \hat{\lambda}Mr. \tag{4.16}$$

Note that in the proof of this theorem we did not use the fact that D is spanned by committors, so this result is also valid for any other subspace D which is spanned by a basis $\{q_1, ..., q_m\}$. Especially, if we have the fuzzy decomposition into m modules $C_1, \ldots, C_m$ and choose the basis functions q_i to be the committor functions, then the entries of M_{ij} have a stochastic interpretation [52, 141], given by the following Theorem.

Theorem 8
Let Q be the orthogonal projection onto the space spanned by the committor functions $D = \{q_1, \ldots, q_m\}$ and as before, let $M_j = \bigcup_{k \neq j} C_k$. Then,

$$M_{ij} = \mathbb{P}[X_n \in T, \tau_{C_j} < \tau_{M_j} | \hat{X}_n = i],$$

that is, the probability to be outside of the modules and enter the module C_j next rather than any other module, under the condition, that the last visited module was C_i.

The proof of this theorem can be found in [52] and [141].

Special case: Full partition

When the basis functions are chosen such that $q_i(x) = \mathbb{1}_{C_i}(x)$, the sets C_i have to form a full subdivision of state space. In particular, the transition matrix of the coarse-grained process can be written in the form QPQ, where Q is the orthogonal projection onto the finite-dimensional space D of all step-functions that are constant on the sets C_i. Moreover, because of orthogonality of the step-functions we then have

$$M_{ij} = \frac{\langle q_i, q_j \rangle}{\hat{\mu}(i)} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}. \tag{4.17}$$

4.3.2 Approximation of Dominant Eigenvalues

Now we want to find sets $C_1, \ldots, C_m$ such that the longest relaxation timescales of the random walk, being encoded by the m dominant eigenvalues $\lambda_1, ..., \lambda_m$ of P, are optimally reproduced by the timescales of the coarse-grained random walk, encoded by the eigenvalues $\hat{\lambda}_1, ..., \hat{\lambda}_m$ of $\hat{P}$. Therefore, our question is:

How well do the eigenvalues of the projected transfer operator approximate the original eigenvalues of P?

Because of self-adjointness of the transfer operator we can use the results from [100](Theorem 2.2) to prove the following theorem [52, 141].

Theorem 9
Let $1 = \lambda_1 > \lambda_2 > \ldots > \lambda_m$ be the m dominant eigenvalues of P, $u_1, u_2, \ldots, u_m$ the corresponding normalized eigenvectors and $D \subset L^2_\mu$ a linear subspace with

$$\mathbb{1} \in D \qquad \dim(D) =: m \tag{4.18}$$

and Q the orthogonal projection onto D.
Moreover, let $1 = \hat{\lambda}_1 > \hat{\lambda}_2 > \ldots > \hat{\lambda}_m$ be the dominant eigenvalues of the projected operator QPQ. Then,

$$E(C_1, \ldots, C_m) = \max_{i=1,\ldots,m} |\lambda_i - \hat{\lambda}_i| \leq \lambda_2 (m-1)\, \delta^2, \tag{4.19}$$

where

$$\delta = \max_{i=1,\ldots,m} \|Q^\perp u_i\|$$

is the maximal projection error of the eigenvectors onto the space D.

Proof. The eigenvector of P w.r.t. the trivial eigenvalue $\lambda_1 = 1$ is known: $u_1 = \mathbb{1}$. Therefore

$$u_1 \in D \Rightarrow Qu_1 = u_1. \tag{4.20}$$

This implies that u_1 is also eigenvector of QPQ w.r.t. its largest eigenvalue $\hat{\lambda}_1 = 1$. Now define

$$\Pi_0 v = \langle v, u_1 \rangle u_1, \tag{4.21}$$

set again $\Pi_0^\perp = Id - \Pi_0$, and consider the operator $P\Pi_0^\perp = P - \Pi_0$. Since P is self-adjoint, its eigenvectors $u_1, u_2, ...$ are orthogonal, which implies that

$$P\Pi_0^\perp u_j = Pu_j - \Pi_0 u_j = Pu_j = \lambda_j u_j \qquad \forall j > 1$$

and $P\Pi_0^\perp u_1 = 0$. That is, the operator $P\Pi_0^\perp$ has the same eigenvalues with the same corresponding eigenvectors as P, just the eigenvalue $\lambda_1 = 1$ changed to a zero eigenvalue. Moreover,

$$\Pi_0 P \Pi_0^\perp = 0, \quad \text{and therefore} \quad P\Pi_0^\perp = \Pi_0^\perp P \Pi_0^\perp,$$

which implies self-adjointness of the operator $P\Pi_0^\perp$.
Now set $U = \text{span}\{u_1, ..., u_m\}$ and let Π be the orthogonal projection onto U. Then, the operator $\Pi P \Pi_0^\perp \Pi$ has exactly the eigenvalues $\lambda_2, ..., \lambda_m$ and an additional eigenvalue zero, that corresponds to the eigenvector $u_1 = \mathbb{1}$.
From (4.20) it follows that $Q\Pi_0 Q = \Pi_0$ and hence

$$QP\Pi_0^\perp Q = QPQ - \Pi_0.$$

The same argument as above shows that the operator $QP\Pi_0^{\perp}Q$ has the same spectrum as QPQ, just the corresponding eigenvalue of u_1 changed from $\hat{\lambda}_1 = 1$ to zero. Using the results from [100](Theorem 2.2), we find for the error (4.19)

$$E(\delta) = \max_{i=2,\ldots,m} |\lambda_i - \hat{\lambda}_i| \leq (\lambda_2 - \lambda_{\min(U+D)}) \max_i \sin^2(\theta_i(U,D)), \tag{4.22}$$

with $\Theta = \Theta(U,D) = \{\theta_1, ..., \theta_m\}$, a vector of principal angles between the subspaces U and D. $\lambda_{\min(U+D)}$ is the smallest eigenvalue of the operator ZPZ, where Z is an orthogonal projection on the space $U + D$. In our case this means $\lambda_{\min(U+D)} = 0$. Let $\sigma_i(A)$ and $\Lambda_i(B)$ denote the i-th singular value of the operator A and the i-th eigenvalue of operator B, respectively. The principal angles are defined as $\cos(\theta_i) = \sigma_i(Q\Pi)$. Moreover, the definition of leading singular values yields

$$\sigma_i^2(Q\Pi) = \Lambda_i((Q\Pi)^* Q\Pi) = \Lambda_i(\Pi Q \Pi), \quad i = 2, \ldots, m, \tag{4.23}$$

where $(Q\Pi)^*$ denotes the adjoint of $(Q\Pi)$ in L^2_μ, in which sense also $Q^*Q = Q$ holds. We get

$$\sin^2(\theta_i) = 1 - \cos^2(\theta_i) = 1 - \Lambda_i(\Pi Q \Pi) = \Lambda_i(\Pi - \Pi Q \Pi) = \Lambda_i(\Pi Q^{\perp} \Pi). \tag{4.24}$$

As in (4.23),

$$\Lambda_i(\Pi Q^{\perp} \Pi) = \sigma_i^2(Q^{\perp}\Pi) \leq \|Q^{\perp}\Pi\|^2. \tag{4.25}$$

Now let us choose an arbitrary v such that $\|v\| = 1$. If we define $\hat{v} \in \mathbb{R}^{m-1}$ as

$$\hat{v}_j = \langle v, u_j \rangle, j = 2, ..., m,$$

and denote the usual 1-, and 2-norms on $\mathbb{R}^{m-1}$ by $\|\cdot\|_1$ and $\|\cdot\|_2$, respectively, we find immediately that

$$\sum_{j=2}^{m} |\langle v, u_j \rangle| = \|\hat{v}\|_1 \leq \sqrt{m-1}\|\hat{v}\|_2 = \sqrt{m-1} \left(\sum_{j=2}^{m} \langle v, u_j \rangle^2 \right)^{1/2} \leq \sqrt{m-1}. \tag{4.26}$$

Since $Q^{\perp} u_0 = 0$,

$$\begin{aligned} \|Q^{\perp}\Pi v\| = \left\| \sum_{j=2}^{m} \langle v, u_j \rangle Q^{\perp} u_j \right\| &\leq \sum_{j=2}^{m} |\langle v, u_j \rangle| \|Q^{\perp} u_j\| \\ &\leq \sum_{j=2}^{m} |\langle v, u_j \rangle| \delta \leq \sqrt{m-1} \cdot \delta. \end{aligned} \tag{4.27}$$

Combining (4.24), (4.25) and (4.27) yields

$$\sin^2(\theta_i) \leq \|Q^{\perp}\Pi\|^2 \leq (m-1)\,\delta^2. \tag{4.28}$$

Putting everything together gives (4.19). □

This theorem tells that the dominant eigenvalues of P are well approximated by the eigenvalues of the projected matrix $\hat{P}$, if the projection error of the corresponding eigenvectors is small enough.

Inserting (2.25) into (4.19), we get the lag time dependent eigenvalue estimate

$$E(\tau, \delta) = \max_{i=1,\ldots,m-1} |\lambda_i - \hat{\lambda}_i| \leq e^{\Lambda_1 \tau}(m-1)\,\delta^2, \tag{4.29}$$

where (λ_i) are the dominant eigenvalues of the transfer operator P_τ and $(\hat{\lambda}_i)$ the dominant eigenvalues of the projection $QP_\tau Q$.
Since $\Lambda_1 < 0$,

$$E(\tau, \delta) \to 0, \text{ for } \tau \to \infty, \tag{4.30}$$

which results from the asymptotic convergence to the invariant measure. Furthermore, for the **relative eigenvalue error** of the first non-trivial eigenvalue we have

$$\frac{|\lambda_1 - \hat{\lambda}_1|}{|\lambda_1|} \leq (m-1)\,\delta^2, \tag{4.31}$$

from which we see that by decreasing the maximal projection error we will have control even over the relative eigenvalue error.

The result of Theorem 9 does not require any specific assumptions about spectral gaps or comparable quantities. Let us explain this aspect in more details. First, there is a variety of results for metastable processes that show that the existence of a spectral gap leads to small projection error δ, for example for diffusion processes in multi-well potentials, see [67, 29, 30]. Second, there are also cases with small δ for original dynamics with wide spectrum without any significant spectral gaps [127]. Let us illustrate this point by the following example.

Example 13 *Let us consider a* $3-$*state Markov chain with the transition matrix*

$$P = \begin{pmatrix} 1-\alpha & \alpha & 0 \\ 1/2 & 0 & 1/2 \\ 0 & \alpha & 1-\alpha \end{pmatrix},$$

where $\alpha \in (0,1)$. *The eigenvalues of* P *are* $\lambda = 1, 1-\alpha, -\alpha$ *with a clear spectral gap for* α *close to* 0 *and no gap for* α *closer to* 1. *We have* $\mu = 1/3 \cdot (1,1,1)$ *and* $u_1 = \sqrt{3/2} \cdot (1,0,-1)$. *With sets* $C_1 = \{1\}$ *and* $C_2 = \{3\}$ *we can easily compute that* $Qu_1 = u_1$ *such that* $\delta = 0$ *for* $m = 2$ *independent of* α. *In fact,* QPQ *has a matrix representation*

$$\begin{pmatrix} 1-\alpha/2 & \alpha/2 \\ \alpha/2 & 1-\alpha/2 \end{pmatrix},$$

with eigenvalues $\hat{\lambda} = 1, 1-\alpha$.

4.4 Algorithm for identification of modules

The main question of this chapter is how to identify modules $C_1, \ldots, C_m$ in a network. Here we adopt the perspective that we have introduced above: The optimal modules $C_1, \ldots, C_m$ are such sets of nodes that minimize the eigenvalue error (4.19) i.e.,

$$C_1, \ldots, C_m = \text{argmin}_{[C_1^*, \ldots, C_m^*]} E(C_1^*, \ldots, C_m^*). \tag{4.32}$$

Identification of such sets in network is based on optimizing the eigenvalue error function over all possible partitions. Therefore, this is an NP-hard combinatorial optimization problem which is furthermore, characterized by the existence of many local minima. In order to obtain a global minima of 4.32, we will use the standard **Simulated Annealing** (SA) algorithm [98]. The SA is a probabilistic heuristic that has a goal to find the global minima of an objective function, starting from an arbitrary initial solution. Motivation for this approach came from the annealing process in metallurgy, which involves the process of heating of a solid in order to move the atoms from the local minima, followed by a controlled cooling that is done in such a way, that the "freezing" point happens at the minimum energy configuration of the solid. The "heating process" in the SA algorithm can be seen in occasional moves that usually lead to an increase of the cost function, but they also help escape the local minima. This is the main difference between the simulated annealing and local search algorithms, that tend to get stuck in local minima. The cooling process on the other hand decreases the number of hops as we approach the global minimum value. Here we will use the standard SA algorithm and for more details, we refer to [98].

There are two important issues that have to be addressed when using the simulated annealing algorithm. First, the SA requires an initial solution of the minimization problem, where it should be noted that as for every other heuristic, the convergence time of SA is highly dependant on the choice of this initial solution. Second, the convergence time increases with the size of the network, which makes this approach usually very costly for large real-world networks. In order to provide one possible solution to these problems, we will now introduce an heuristic approach [142] that will help us to efficiently find a "good" initial solution and provide valuable insights for finding optimal modules that can decrease the computational effort.

4.4.1 New heuristic for finding modules

Let us return to the approximation error of the dominant eigenvalues, given in (4.19) as

$$E(C_1, \ldots, C_m) = \max_{i=1,\ldots,m} |\lambda_i - \hat{\lambda}_i| \leq \lambda_2 (m-1)\, \delta^2.$$

An upper bound on δ for Markov processes on networks, but also for more general situations was given in [144]. More precisely, for any eigenvalue λ_i of P and the

corresponding normalized eigenvector u_i it holds

$$\delta_i \leq p(u_i) + 2\mu(T)p_{max}(u_i) + r(T)(1-\lambda_i)\left(\sum_{x\in T} u_i(x)^2\mu(x)\right)^{\frac{1}{2}} \tag{4.33}$$

where

$$\begin{aligned} r(T) &= \sup_{\substack{\|v\|=1,\\ v=0 \text{ on } C}} \left(\frac{1}{\sum_{x\in T}(v(x)-(Pv)(x))^2\mu(x)}\right)^{1/2} \\ p(u_i) &= \|e_i\| \quad p_{max}(u_i) = \|e_i\|_\infty \\ e_i(x) &= \begin{cases} 0, & \text{if } x \in T, \\ \frac{1}{\mu(C_j)} \sum\limits_{y\in C_j} u_i(x) - u_i(y)\mu(y), & \text{if } x \in C_j. \end{cases} \end{aligned} \tag{4.34}$$

From this inequality we deduce that in order to ensure small projection errors $\|Q^\perp u_i\|$ for the dominant eigenvectors, modules should be chosen in such a way that the following two conditions are satisfied. First, from the transition region T the random walker should always enter some module quickly enough such that $r(T)(1-\lambda_i)$ is "small enough". More precisely, the more eigenvalues of P we want to approximate, the faster the random walker should leave the transition region T. Second, the dominant eigenvectors should be almost constant on modules in order to guarantee small values of $p(u_i)$ and $p_{max}(u_i)$. It will be particularly useful that the error bound decomposes into these two parts. Namely, the factor $r(T)(1-\lambda_i)$ depends only on the choice of the transition region T, whereas the errors $p(u_i)$ and $p_{max}(u_i)$ depend only on the partitioning of $C = V \setminus T$ into modules. Following this idea, we will divide our approach for finding modules into two separate steps [142]:

1. identifying the transition region T;
2. partitioning the remaining nodes of the network $C = V \setminus T$ into modules $C_1, \ldots, C_m$.

Step 1

In this step we will choose such T that will ensure a small factor $r(T)(1-\lambda_i)$ in the error bound (4.33) for the dominant eigenvalues. If we fix a specific choice of T and let an ensemble of infinitely many random walk processes to start only in this region, then $r(T)$ measures how many of these processes will leave the transition region. That is, the higher the probability that the random walker will leave T quickly, the smaller the factor $r(T)$ will be. Next, this factor is compared to $(1-\lambda_i)$ for the dominant eigenvalues. This yields the following interdependency: For eigenvalues close to one, $(1-\lambda_i)$ will be rather small, which gives us more flexibility in choosing T that influences $r(T)$. Remember that the closer to one the eigenvalues are, the stronger they indicate the presence of metastability in the

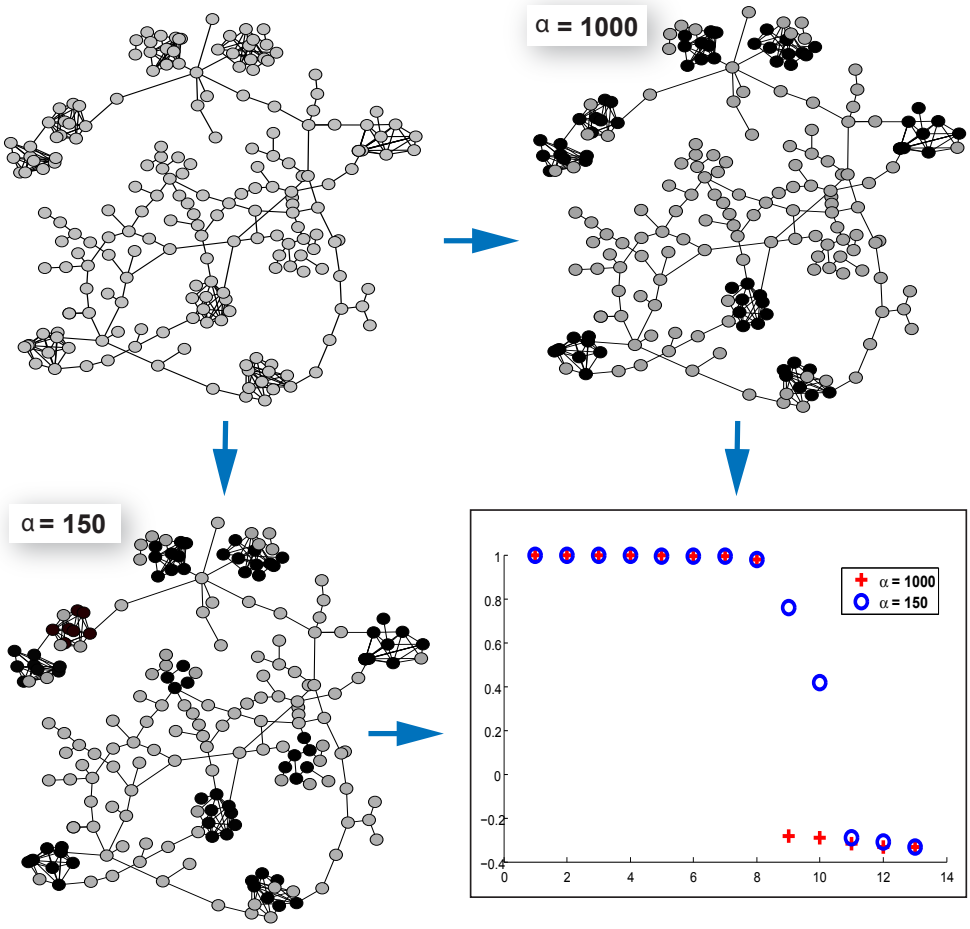


Figure 4.3: Example network with 200 nodes. After applying step 1 of our heuristic the resulting transition region T (gray) and union of modules $\mathcal{M}^\alpha$ (black) is shown for $\alpha = 1000$ (top right) and $\alpha = 150$ (bottom left). Bottom right: the first 13 eigenvalues of $\hat{P}_\alpha$, for $\alpha = 1000$ (red crosses) and $\alpha = 150$ (blue circles).

system. Therefore, if we also want to consider modules, which are less metastable, we will have to approximate eigenvalues, which are less close to one and therefore, the region T has to be left more quickly.

Algorithmically this leads to the following idea: We take the invariant measure μ^* of the random walk process using the generator L, as defined in (2.23) and parameter $p = 0$. That is, we turn off the effect of waiting times, which made the modules in the network more metastable. Then, we consider the random walk for $p = 1$ and choose a lag time $\alpha > 0$ at which we want the random walker to leave the transition region. We will choose a rather large α if we are interested in finding only the most metastable set of modules and decrease α if we also want to identify modules with less metastability. Because of this, we refer to α as the **metastability parameter**.

Now, we choose

$$\mathcal{M}^\alpha = \{x \in V | (P_\alpha^T \mu^*)(x) > \mu^*(x)\} \tag{4.35}$$

to be the set containing modules with respect to α. Connecting to the ensemble point of view, the set $\mathcal{M}^\alpha$ is exactly the region, which rather attracts random walkers in the ensemble than let random walkers leave within the time step α.

At this point, the natural question to ask is: how can we choose the appropriate value for the metastability parameter α? Since α is connected to the timescale at which the random walk leaves the transition region, it is possible to get an idea about reasonable values for α from the spectrum of the generator L. That is, if the dominant eigenvalues of L are denoted by $0 = \Lambda_0 > \Lambda_1 \geq \Lambda_2 \geq ...$ the implied timescales of the random walk, which are given by $1/|\Lambda_1| \geq 1/|\Lambda_2| \geq \ldots$, provide estimates for possible choices of α. If there is a cluster of eigenvalues $0 = \Lambda_0 > \Lambda_1 \geq \ldots >\geq \Lambda_k$ around 0 separated by a spectral gap from the smaller eigenvalues, then a good choice of α would be $1/|\Lambda_k| > \alpha > 1/|\Lambda_{k+1}|$. Several spectral gaps therefore would give us a list of proposals for good values of α. Let us now illustrate this effect on the following example.

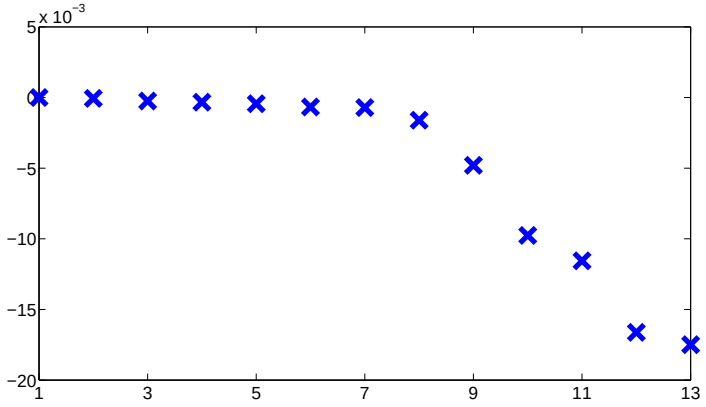


Figure 4.4: This plot shows the 13 largest eigenvalues of the generator L for the example network.

Example 14 *Figure 4.3 shows an example network with* 200 *nodes and the resulting transition region T (marked in gray) together with the region $\mathcal{M}^\alpha$ (marked in black). In order to demonstrate the effect of choosing a parameter α, the first step of our heuristic is done for two different choices of α, namely $\alpha = 1000$ (top right) and $\alpha = 150$ (bottom left). These values of α have been selected according to the spectrum of L shown in Figure 4.4, which exhibits gaps after the eighth and the tenth eigenvalue, such that $1/|\Lambda_8| = 1254$ and $1/|\Lambda_{10}| = 104$. Therefore, for $\alpha = 1000$ the set $\mathcal{M}^\alpha$ contains only the most metastable parts of the network. When decreasing α to* 150*, sets of nodes that correspond to less metastable parts of networks are also added to $\mathcal{M}^\alpha$.*

Step 2

Having identified the set $\mathcal{M}^\alpha$ we now have to find its full partition into modules. For this purpose, we consider the random walk only on the nodes belonging to $\mathcal{M}^\alpha$ with the transition matrix

$$\hat{P}_\alpha(x,y) = \sum_{z\in V} P(x,z)q_y(z), \quad x,y \in \mathcal{M}^\alpha, \tag{4.36}$$

where $q_y(z)$ is the probability that y will be the next node from $\mathcal{M}^\alpha$ that is hit by the random walk starting in z. That is, $\hat{P}_\alpha(x,y)$ describes the transition probabilities between the nodes of $\mathcal{M}^\alpha$, ignoring the waiting times and the transition region. Note that $\hat{P}_\alpha$ describes the dynamics only between the nodes of modules. Now we can use some hard spectral clustering method to split $\mathcal{M}^\alpha$ into the modules $C_1, ..., C_m$.

Let us now apply this step onto the result from the previous example. Figure 4.3 shows the first 13 eigenvalues of $\hat{P}_\alpha$, for $\alpha = 1000$ (marked as red crosses) and $\alpha = 150$ (marked as blue circles). Note that now that we have erased the transition region the spectrum of the random walk clearly indicates the number of modules corresponding to the chosen level of metastability. For $\alpha = 1000$ there is a clear gap after 8 eigenvalues, whereas for $\alpha = 150$ there are two additional eigenvalues indicating less pronounced metastability.

Computational effort

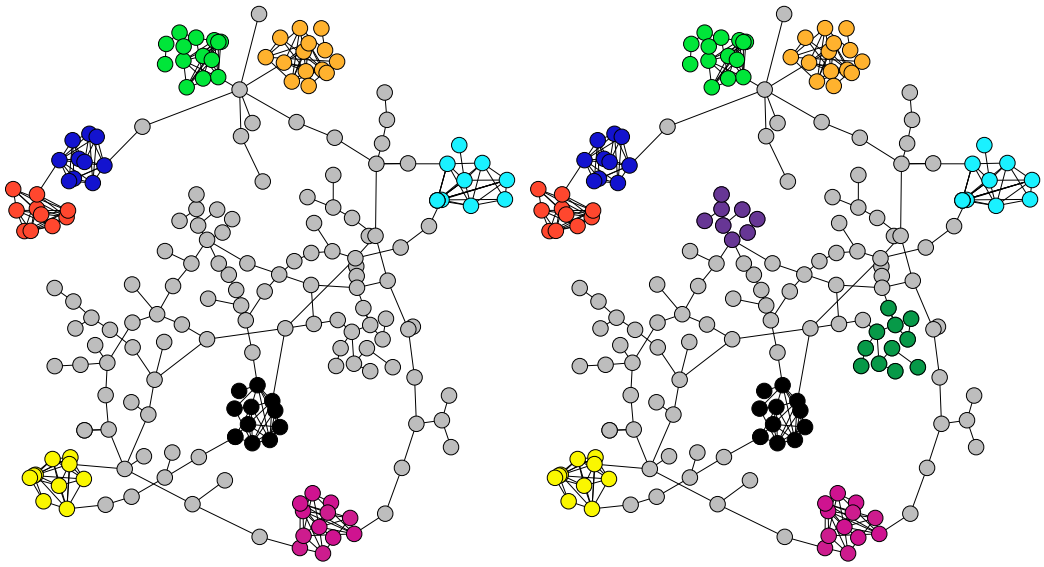
Step 1: For example in [4] it is shown that the computational effort is dominated by matrix multiplications. For a large, sparse matrix L this effort is $\mathcal{O}(n)$.
Step 2: First, we have to compute the committor functions, i.e. solve a symmetric, positive definite linear system for $k = |\mathcal{M}^\alpha|$ right hand sides. Since the matrix $\hat{L}$ is large and sparse, conjugate gradient methods allow to compute the solution in $\mathcal{O}(kn)$ point operations. Then, we have to compute a hard clustering with respect to the coarse grained random walk with $k \times k$ transition matrix $\hat{P}_\alpha$. For this task, a lot of algorithms exist, for example, [49, 56]. The fastest combinatorial methods perform in $\mathcal{O}(k^2 \log k)$.
The whole algorithm: This shows that the overall effort is dominated by step 2, where we have to compute a hard clustering for the k nodes belonging to $\mathcal{M}^\alpha$. The total effort scales like $\mathcal{O}(k^2 \log k) + \mathcal{O}(kn)$. If $k \ll n$, that is, if the number of nodes in modules is much smaller than the number of nodes not assigned to modules, then the effort scales linearly with the total number of nodes. In general, this algorithmic strategy reduces the computational effort to calculate a fuzzy decomposition of a network with n nodes to the effort of computing a hard clustering for k nodes.

4.4.2 Finding optimal modules

The method presented in Section 4.4.1 is an heuristic, that is there is no proof that it finds modules that minimize the upper bound given in 4.33. Because of its

computational efficiency and "good", near optimal solutions, we will use it to obtain an initial solution of our problem. Then, we can apply the simulated annealing algorithm on the whole network in order to find modules which minimize 4.32. However, for very large networks performing the SA algorithm can still be costly.

We propose now a way to reduce this problem using the fuzzy assignment functions



(a) 8 modules (b) 10 modules

Figure 4.5: This figure shows the optimal modules for $\alpha = 1000$ (left) and $\alpha = 150$ (right).

obtained in Section 4.4.1. Namely, if node $x \in T$ is such that $q_i(x) > \theta > 0.5$ for some threshold value θ that is close to 1, e.g. $\theta = 0.9$, then this node is **committed to the module** C_i with a high probability. We will consider that all nodes $x \in V$ for which $q_i(x) > \theta$, i.e. elements of C_i together with nodes committed to C_i belong to this module

$$C_i \cup \{x \in V : \quad q_i(x) \geq \theta\}.$$

This means that the transition region T is dependant on the parameter θ in the following sense

$$T(\theta) = \{x \in V \setminus \bigcup_{i=1}^{m} C_i \,|\, q_i(x) < \theta, \ \forall i = 1, \ldots, m\}.$$

Obviously, decreasing a value of θ, increases the number of elements in modules, i.e. decreases the number of elements in the transition region T. Therefore, for some fixed value of θ we can search for the optimal solution in the space $V \setminus T(\theta)$. Depending on the choice of θ, the state space can be made considerably small in order to make the simulated annealing feasible for huge networks. This approach

is of course also a heuristic solution, but with an advantage that as θ decreases it converges to the state space of original minimization problem.

Example 15 *Let us consider again the network shown in Figure 4.3. In order to identify initial solution for the SA algorithm, we use the heuristic introduced in Section 4.4.1. As we have discussed above, for different choices of parameter α, namely $\alpha = 1000$ and $\alpha = 150$, we have 8 and 10 initial sets. Optimal modules for both values of α are shown in Figure 4.5.*

4.5 Related work

In the last years, finding modules in complex networks has attracted a lot of attention of diverse group of scientists. Most of the state-of-the-art approaches are inspired by concepts from graph theory, statistical physics, dynamical systems theory and computer science. Since the results of these methods are supposed to have an adequate interpretation in the underlying complex system, many of these ideas are affected by the concepts coming from the particular application area, such as social and biological sciences.

Two current challenges regarding modules in network are

- the ability to quantify to what extent a given network is "modular" [121, 126, 56];
- the ability to identify the modules of a given network.

A range of techniques for identifying modules are summarized in a review article [121]. Here, we present four approaches addressing the two challenges:

Girvan-Newman algorithm is a widely used method [70] based on iteratively removing edges from the network that have a high betweenness centrality, see Definition 10. The edges with high betweenness centrality are interpreted as the edges between modules and therefore their removal from the network leaves just the modules themselves. Identified modules form a full partition of a given network. This method has been shown to produce good results and is often used as a benchmark for for newly developed algorithms. However, the method is very slow, namely $O(M^2n)$ where M is the number of edges in the network. For this reason Girvan-Newman algorithm is unpractical for applying to large networks.

Modularity maximization approach is of a particular interest, as it optimizes **modularity** of the network, a widely used measure for the quality of proposed modules [126]. Modularity measure is based on the intuitive concept of "good modules", that are modules with large number of edges between nodes in the same module and only a few edges with nodes outside of module. Here "large number of edges" refers to a larger number than it would be expected purely by chance.

Following this idea, modularity of a network with m modules $C_1, \ldots, C_m$ represents the difference between the number of edges within modules and the expected number of such edges in a random network with the same degree distribution. It can be expressed as

$$Q = \sum_{k=1}^{m} \left(\frac{l_k}{M} - \left(\frac{d_k}{2M} \right)^2 \right), \tag{4.37}$$

where l_k is the number of edges in module C_k, M is the total number of edges in the network and $d_k = \sum_{i \in C_k} d(i)$.
Larger values of modularity indicate existence of strongly connected modules and vice versa. Also this approach identifies modules that form a full partitioning of a given network. However, this method is based on optimizing a modularity function over all possible partitions of the network, which has a high computational cost. Finding optimal modules that maximize modularity is shown to be a NP-hard problem [31]. In order to improve this, different heuristic algorithms have been used such as greedy algorithm [123] and simulated annealing [74]. Recently, the effectiveness of this method has been put to the question, as it has been observed that modularity optimization can fail to identify modules smaller than a certain scale [66].

Markov clustering algorithm (MCL) is a random walk based algorithm, developed in [161]. Given a transition matrix of a random walk process P, this algorithm iteratively finds the modules of the given network. Each iteration consists of two steps: **expansion** and **inflation**. First, expanding the random walk on the network by one steps produces the matrix P^2. Inflation consists of raising every entry of P^2 to the power of α, $\alpha \in \mathbb{R}$, where typically $\alpha \approx 1.2 - 2$ and renormalizing these values, such that the resulting matrix is again stochastic. In this step high values in each row of P^2 are increased, which results in increasing the difference between the intra-modular and inter-modular jump probabilities.
Eventually this method should converge to a $0-1$ matrix that is invariant to the two operations. However, the convergence of MCL has not been proved. Number of non-zero rows of the obtained matrix represent the number of clusters and non-zero entries of these rows indicate which nodes belong to the same cluster. MCL produces clusters that form a full partitioning of a given network, but there can exist clusters with only one node. The choice of the parameter α influences the number, size and elements of the resulting modules. Namely, for smaller values of α MCL produces a few big modules, whereas for larger values of α MCL results in identifying smaller modules. The complexity of this method is $O(n^3)$, for a network with n nodes.

Robust Perron Cluster Analysis (PCCA+) is a spectral clustering algorithm that results in a fuzzy clustering of data, i.e., all states are assigned to clusters within certain assignment probabilities [49, 50]. Assuming that we want to identify m clusters, then for every state $x \in V$ and every cluster $i \in \{1, \ldots, m\}$ PCCA+ calculates

the probability $\chi_i(x)$ that state x belongs to cluster i. Functions $\chi_i, i = 1, \ldots, m$, called membership functions, give the clustering information of the network in the sense that they decompose the complete state space into m metastable sets. Therefore, they are assumed to form a non-negative partition of unity $\sum_{i=1}^{m} \chi_i(x) = 1$, $x \in V$, and to be almost invariant under P. The goal of PCCA+ is to find a linear transformation matrix X that transforms $U = [u_1, \ldots, u_m]$, the first m dominant eigenvectors of P, into the membership functions $\chi = [\chi_1, \ldots, \chi_m]$, i.e., $\chi = XU$. In order to get the optimal clustering, X is chosen such that it maximizes the metastability functional

$$I(X; U, \mu) = \sum_{i=1}^{m} \frac{\langle \chi_i, P\chi_i \rangle_\mu}{\langle \chi_i, \mathbb{1} \rangle_\mu},$$

under the constraint that $\chi = XU$ forms a non-negative partition of unity.
Notice that PCCA+ does not automatically provide us with an estimate for the number of clusters. There are several techniques that can suggest the optimal choice m_{PCCA}, e.g., by running the algorithm for different cluster numbers m of clusters and determine

$$m_{PCCA} = \text{argmax}_m \frac{1}{m} I_m,$$

where I_m the maximum of the functional for a given m; or by determining m_{PCCA} via the minimal overlap between the assignment functions χ_i. For more details about PCCA+ algorithm see [50].

4.5.1 Contrasting different graph clustering approaches

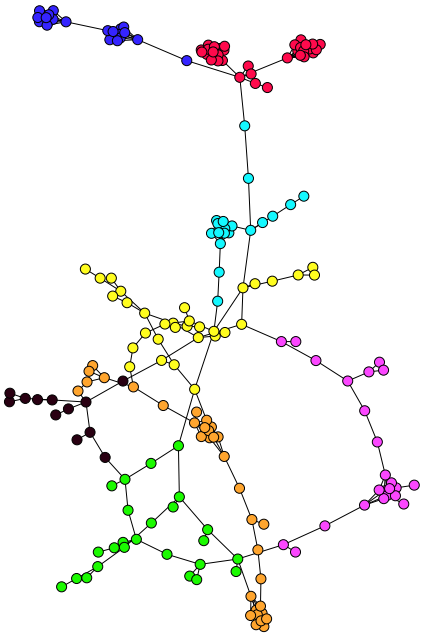
Having mentioned above some of the methods for module identification, the natural questions to ask are:

- What is a good graph clustering method?
- How can we measure the quality of resulting modules in a network?
- What is the measure of quality of some clustering method?

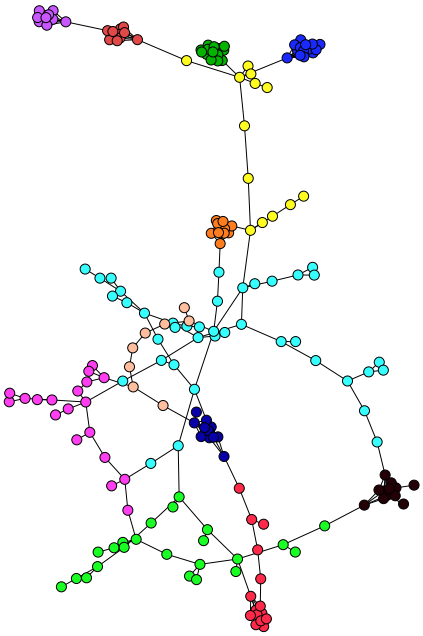
Finding answers to these questions has been a widely studied task [33, 93, 147]. Up until now, several clustering quality measures have been introduced, such as (already introduced) modularity [126], conductance and expansion [93]. However, these measures are usually biased towards methods that optimize their functional. Up to now, there has been no generally established measure for clustering quality. One reason is that it is not always clear what are the "right" modules in a given network. Furthermore, often the quality of chosen modules is dependant on the application, i.e. their interpretation in the underlying system. Therefore, we will not discuss the quality of the above mentioned approaches and our approach based on some established measure, but present the resulting clusterings of these approaches on one example network. The example network is chosen in such a way that the resulting modules do not coincide for all methods.

Example 16 *In this example we will apply five clustering algorithms introduced above, on an undirected, unweighted network with* 200 *nodes. Figure 4.6 shows the resulting optimal modules found by: Girvan-Newman algorithm in Figure 4.6a, modularity maximization algorithm in Figure 4.6b and MCL algorithm in Figure 4.6c. The results for the MCL algorithm are obtained for the choice of* $\alpha = 1.2$*. All three algorithms find modules that form full partitioning of the network. We find that in all three methods almost all densely connected regions are clustered in different modules. The except is that Girvan-Newman algorithm and MCL algorithm put together the two modules (dark blue in Figure 4.6a) in the upper left corner, whereas the modularity maximization algorithm identifies them as two different modules.*

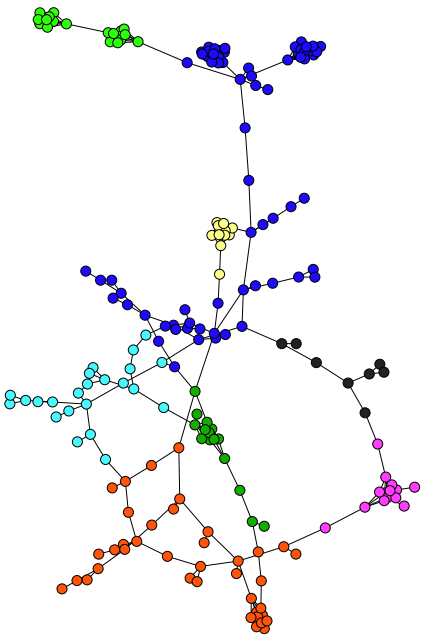
Figure 4.7 shows optimal modules of two fuzzy clustering algorithms, namely PCCA+ in Figure 4.7a and our approach (in Figure 4.7b) that uses the time-continuous random walk process defined in (2.29). Fuzzy clustering results obtained by both approaches are presented in the plot in such a way that nodes that are affiliated to some module with the probability higher than 0.85 *belong to that module. Both methods find* 8 *modules, where again the "problematic modules" from above are clustered together via PCCA+ approach (orange module in Figure 4.7a) and by our approach they represent two different modules (orange and pink modules in Figure 4.7b). We see that most of the transition region coincide in results of both fuzzy clustering methods. Remark that when using the standard random walk process for analyzing this network we encounter the problems stated in Section 2.2, i.e. the spectrum of the transition matrix does not have a clear gap to indicate the number of modules.*



(a) 8 modules found by Girvan-Newman algorithm

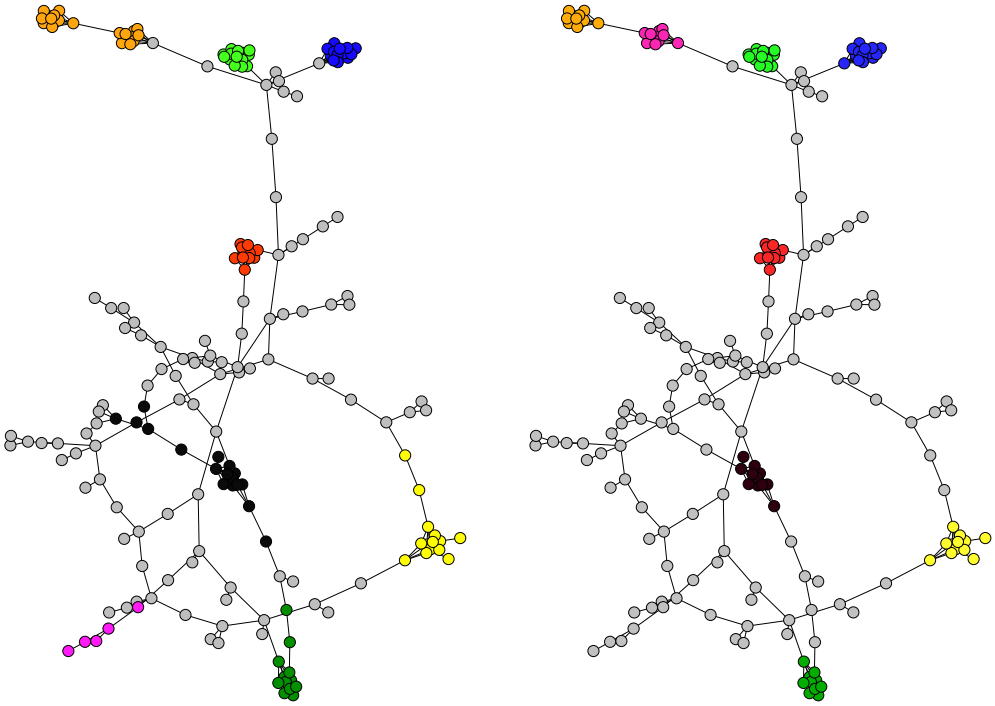


(b) 13 modules modularity maximization approach



(c) 8 modules found by MCL algorithm

Figure 4.6: This figure shows the optimal modules found by different clustering methods. These methods find full partitioning of the network.



(a) 8 modules found by PCCA+

(b) 8 modules found by our approach

Figure 4.7: This figure shows the 8 optimal modules found by PCCA+ algorithm and our algorithm using time-continuous random walk process. Both algorithms find fuzzy clustering of the network.

Chapter 5

Identification of hubs

When analyzing complex, modular networks, one of the main issues is how to identify nodes that have a special importance in the underlying system, such as leading scientists who collaborate with many scientists, airports which are essential connectors of some regions, proteins that are responsible for a certain function etc.

It has been observed that many of the real world networks exhibit scale-free behavior (Section 1.2.3) [6], for example World-Wide Web [7], social networks, biological networks [5, 90], transportation networks [73] etc. Such networks are shown to be resistant to random node attacks and failures [43, 8, 6]. This means, if randomly chosen nodes are removed from the network, that doesn't lead to the global loss of connectivity in the network. On the other hand, scale-free networks are shown to be vulnerable to the systematic removal of particular nodes [44]. More precisely, removal of certain nodes can cause disconnection of some modules from the rest of the network. In that sense these nodes represent the critical points of the network and their identification is of special importance.

Many different measures of importance of nodes have been introduced and tested on various real world examples aiming at identifying nodes of special importance [160, 169, 75, 73], such as: essential proteins, major airports, disease-related genes etc. Most of these methods are based on the topological properties of networks, such as degree distributions [89], betweenness centrality [70, 92] and network motifs [61]. However, recent results coming from various applications revealed that these methods are not always sufficient to identify special functional elements of the underlying system [169, 75, 73].

In this chapter, we will define new measures of node importance based on the idea that such nodes are essential for the communication in the network. Using the framework of Transition Path Theory that will be introduced in Section 5.1, we will specify the notion of *communication* in the network taking into account dynamical properties of the random walk process taking place on the network. Sections 5.2, 5.3 and 5.4 will provide precise definitions for different types of hub nodes and algorithms for their identification, that will be demonstrated on an example network in Section 5.5. We will end this chapter by presenting state of the art approaches for defining hubs and compare them to our approach.

5.1 Transition Path Theory for Markov Jump Processes

In this section, we will study the transition behavior of the random walk process, using the framework of **Transition Path Theory** (TPT) that has been introduced in [58] for specific continuous state spaces and in [117] for discrete setting needed here. More specifically, we will use TPT objects in order to describe communication between modules and to identify hubs, which are nodes that are essential for the communication between modules.

We start by observing transitions from a module C_i to the union of all other modules $M_i = C \setminus C_i$. In particular, we take into account only these parts of trajectories (realizations of the random walk), where the random walker transits directly from C_i to M_i. These are defined as

Definition 19 *The n^{th} **reactive trajectory** from a module C_i to the union of all other modules $M_i = C \setminus C_i$ is the sequence of states*

$$P_n = [x_n^{C_i}, x_n^1, \ldots, x_n^k, x_n^{M_i}], \quad x_n^{C_i} \in C_i, \quad x_n^i \in T, \quad x_n^{M_i} \in M_i, \tag{5.1}$$

*that is the n^{th} transition path starting in C_i and ending in M_i. The union of all such trajectories is called **the set of reactive trajectories**.*

Statistical properties of these trajectories will provide us information about global, as well as local transition behavior of the system.

5.1.1 Global transition behavior: Reactive flows

Let us observe the set of reactive trajectories from a module C_i to the union of all other modules M_i. We will introduce the first object of interest in order to calculate the rate at which the flow goes from one state to the next one. To this end, let us consider

Definition 20 *The **discrete probability current** of reactive trajectories $f^{C_iM_i} = (f_{xy}^{C_iM_i})_{x,y\in V}$ is defined as the average flow of reactive trajectories when going from state x to y, per time unit, i.e.*

$$f_{xy}^{C_iM_i} = \begin{cases} \mu(x)\, q_i^-(x)\, l(x,y)\, (1 - q_i^+(y)), & \text{if } x \neq y \\ 0, \quad \text{otherwise} \end{cases} \tag{5.2}$$

The following proposition will provide an important property of the discrete probability current [115].

Proposition 7
The discrete probability current is conserved in every node outside the two sets

$$\sum_{y\in V} f_{xy}^{C_iM_i} = \sum_{y\in V} f_{yx}^{C_iM_i}, \quad x \in T. \tag{5.3}$$

Proof. Using the properties of committor functions (3.4) and (3.5), namely that $\sum_{y\in V} l(x,y)q_i^+(y) = 0$ and $\sum_{y\in V} l^b(x,y)q_i^-(y) = 0$ for $x \in T$, we have that

$$\begin{aligned}
&\sum_{y\in V} f_{xy}^{C_iM_i} - f_{yx}^{C_iM_i} \\
&\quad = \mu(x)q_i^-(x)\sum_{y\neq x} l(x,y)(1-q_i^+(y)) - (1-q_i^+(x))\sum_{y\neq x}\mu(y)q_i^-(y)l(y,x) \\
&\quad = -\mu(x)q_i^-(x)l(x,x)(1-q_i^+(x)) - \mu(x)(1-q_i^+(x))\sum_{y\neq x} q_i^-(y)l^b(x,y) \\
&\quad = -\mu(x)q_i^-(x)l(x,x)(1-q_i^+(x))(l(x,x)-l^b(x,x)) \\
&\quad = 0.
\end{aligned}$$

□

Additional to the local conservation of the probability current in every node $x \in T$, the more general result holds. More precisely, using that $f_{xy}^{C_iM_i} = 0$ for $x \in V, y \in A$ and $x \in B, y \in V$ one can easily prove that the total conservation of the probability current also holds

$$\sum_{x\in C_i, y\in V} f_{xy}^{C_iM_i} = \sum_{y\in V, x\in M_i} f_{yx}^{C_iM_i}. \tag{5.4}$$

Our main goal is to study the intensity of communication between the two sets. Let us start by introducing a measure of communication between two states along the reactive trajectories, i.e. the net amount of probability current between two states. For this purpose, we define

Definition 21 *The **effective current** f_{xy}^+ is defined by*

$$f_{xy}^+ = \max\left(f_{xy}^{C_iM_i} - f_{yx}^{C_iM_i}, 0\right). \tag{5.5}$$

In other words, the effective current calculates the net average number of reactive trajectories per time unit, that make transitions from x to y when going from C_i to M_i. Again, one can show that the effective current is conserved in each node outside the two sets.

Proposition 8
For every $x \in T$ it holds

$$\sum_{y\in V} f_{xy}^+ = \sum_{y\in V} f_{yx}^+. \tag{5.6}$$

Proof. This is a consequence of Proposition 7. Let us fix $y \in T$ and define

$$I_y = \{x \in V : \quad f_{xy}^{C_iM_i} > f_{yx}^{C_iM_i}\},$$

and its complement $I_y^c = V \setminus I_y$. Then,

$$\begin{aligned}\sum_{x\in I_y} f_{xy}^{C_iM_i} + \sum_{x\in I_y^c} f_{xy}^{C_iM_i} &= \sum_{x\in V} f_{xy}^{C_iM_i} \\ &= \sum_{x\in V} f_{yx}^{C_iM_i} \\ &= \sum_{x\in I_y} f_{yx}^{C_iM_i} + \sum_{x\in I_y^c} f_{yx}^{C_iM_i}.\end{aligned}$$

This can be re-arranged into

$$\sum_{x\in I_y} (f_{xy}^{C_iM_i} - f_{yx}^{C_iM_i}) = \sum_{x\in I_y^c} (f_{yx}^{C_iM_i} - f_{xy}^{C_iM_i}),$$

where the summands are all non-negative and identical to the respective effective flow since $I_y = \{x \in V : \quad f_{xy}^+ > 0\}$. Therefore

$$\sum_{x\in I_y} f_{xy}^+ = \sum_{x\in I_y^c} f_{yx}^+.$$

Filling these sums up with zeros yields the assertion. □

Furthermore, the net amount of reactive trajectories that flow out of C_i is the same as the amount that flows into M_i

$$\sum_{x\in C_i, y\in V} f_{xy}^+ = \sum_{x\in V, y\in M_i} f_{xy}^+. \tag{5.7}$$

Using this we can describe the global transition behavior between two sets and measure how good the communication between them is. More formally, we consider

Definition 22 *The **transition rate** $k_{C_iM_i}$ between sets C_i and M_i is defined as*

$$k_{C_iM_i} = \sum_{x\in C_i, y\in V} f_{xy}^+, \tag{5.8}$$

that is the average number of transitions from C_i to M_i per time unit.

At this point a natural question to ask is:

How many of these transitions are going through a specific node $y \in T$?

The answer to this question will provide us with the importance of a particular node for the global communication between modules in the network. However, in order to answer this question we will introduce new objects that will characterize local transition behavior in the network.

5.1.2 Reaction pathways

Every single transition from set C_i to M_i can be characterized by the path the random walker takes from C_i to M_i. This path can be represented in a weighted, directed graph $G^* = (V, E, f^+)$, where the effective current $f^+ = (f^+_{xy}), \forall x, y \in V$ is the weight function of the edges. Defined in this way, the weights of directed edges determine how much flow can go through a particular edge in a particular direction. Now, we can represent a transition from C_i to M_i by a directed path in the graph G^* in the following way

Definition 23 *A **reaction path (pathway)** is a sequence $w = (i_0, \ldots, i_n)$ with $n > 0$ of states such that $i_0 \in C_i$, $i_n \in M_i$,*

$$i_k \in T, \quad \forall k = 1, \ldots, n-1$$

and

$$f^+_{i_k, i_{k+1}} > 0, \quad \forall k = 0, \ldots, n-1.$$

Obviously, the total flow of a specific reaction path is bounded by the minimal effective current of all edges involved in that path. In particular, the effective current that confines the flow is of special importance.

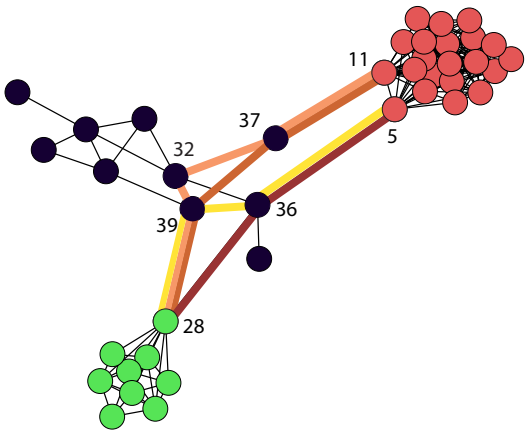


Figure 5.1: The four most important reaction pathways from the red set to the green set. The pathways are colored according to the values of the minimal current that they transport, from dark brown for the most important reaction pathway to yellow for the fourth most important pathway.

Definition 24 *The **capacity** of the reaction path $w = (i_0, \ldots, i_n)$ is defined as*

$$c(w) = \min_{k=0,\ldots,n-1} \{f^+_{x_k, x_{k+1}}\}. \tag{5.9}$$

*The edge with the minimal effective current is called the **dynamical bottleneck** of the path.*

Now, we can characterize every reaction path by its dynamical bottleneck and using this we can distinguish between different transition paths. In practical applications, reaction paths that have the maximal minimal current are of particular interest, since they can transport the most flow. We will refer to these as **the most important reaction pathways**. We can also determine the second most important paths and so on.

The problem of finding these paths is known as the *maximum capacity augmenting path* problem in the context of solving the maximal flow problem in a network [3]. For an algorithmic realization of how to find the important reaction paths see [117]. Figure 5.1 shows the top four reaction pathways from the red set to the green set. These paths are colored according to their importance, i.e. the darker the color of a pathway is, the more current it conducts. In particular, the most important path transports 53% of the total current, the second most important path 25%, the third one 9%, the fourth one 5% and so on.

5.1.3 Local transition behavior

Here we will address the question of how many of the transitions from set C_i to set M_i, are going through a specific node $y \in T$. For each state $y \in T$ outside the given sets let us define the **predecessor** and **successor** sets that contain the states directly before and after y on a transition path

$$P_y = \{x \in V : \quad f^+_{xy} > 0\}, \quad S_y = \{x \in V : \quad f^+_{yx} > 0\}$$

and in particular

Definition 25 *The **reactive flow** through a node $y \in T$ is given with*

$$k_y^{C_i M_i} = \sum_{x \in P_y} f^+_{xy} = \sum_{x \in S_y} f^+_{yx}, \tag{5.10}$$

as the average number of reactive trajectories going through a node y when going from C_i to M_i.

An important property of this quantity is

Proposition 9
For every node $y \in T$ it holds that

$$k_y^{C_i M_i} \leq k_{C_i M_i}. \tag{5.11}$$

Proof. In order to show this, let us fix $y \in T$ and consider the set W_y of all reaction paths that go through node y. Let $w_1, w_2, \ldots, w_h$ be a complete enumeration of W_j. These paths contain no cycles, so there have to be finitely many of them. Let us define r_l to be part of the reactive path w_l that starts with y and ends in set M_i. Now let $G1$ be the sub-graph of the entire network that contains only edges

and nodes that are contained in at least one of the r_l, $l = 1, \ldots, h$. Specifically, $G1$ is a tree with root y and leaves b in M_i, for which we define

$$k_b = \sum_{x \in b} \sum_{z \in G1 \setminus b} f^+_{zx}.$$

Since $b \subset M_i$ and $G1 \subset V$ we have $k_b \leq k_{C_i M_i}$. Because of the local conservation of the flow (5.6) and $G1 \setminus b \subset C_i$ we additionally have that $k_y^{C_i M_i} = k_b \leq k_{C_i M_i}$. □

5.2 Hubs in undirected networks

In the previous section, using TPT we have introduced several quantities for measuring communication in the network on different levels. In particular, the transition rate (5.8) describes the global communication between modules in the network. On the other hand, the effective current (5.5) can be used to measure the local communication between nodes of the network, whereas the reactive flow (5.10) calculates the amount of global flow through a particular node.

Now, we can define

Definition 26 *A **hub** is a node that is important for the communication in the network.*

Here "important for the communication" should be understood in the following sense: if an important hub is removed, communication between modules or between a module and the rest of the network will be considerably perturbed or broken. It has been observed that nodes with this property can have a special meaning for the underlying system, such as being essential proteins that are directly correlated with the viability of cells [89] or cities that connect different communities representing the source for spreading of infections [73]. Therefore, hubs as the nodes that "control" most of the communication in the network, represent at the same time the most vulnerable points of the network. For this reason, identification of such nodes is crucial for the understanding the underlying system.

Example 17 *Figure 5.2 shows an example network with* 100 *nodes, where* 70 *nodes are clustered in four modules. This figure also shows examples of hubs in this network. For the sake of simplicity, we highlighted only some of hub nodes, namely the ten labeled nodes for which we will discuss their importance for the communication in this network. Node M is the only node of the pink module that connects this module with the rest of the network. Therefore, it is a node that is a key connector of the pink module. From the nodes in the transition region, node X is the only node that connects pink and green module. The green module is connected with the rest of the network by nodes Q and R that belong to this module and by X and Z that belong to the transition region. On the other hand, blue module is connected with the rest of the network by nodes S and T that belong to this module. The connection of the pink, green and blue modules with the rest of the network, namely the red module is established through nodes X and Y.*

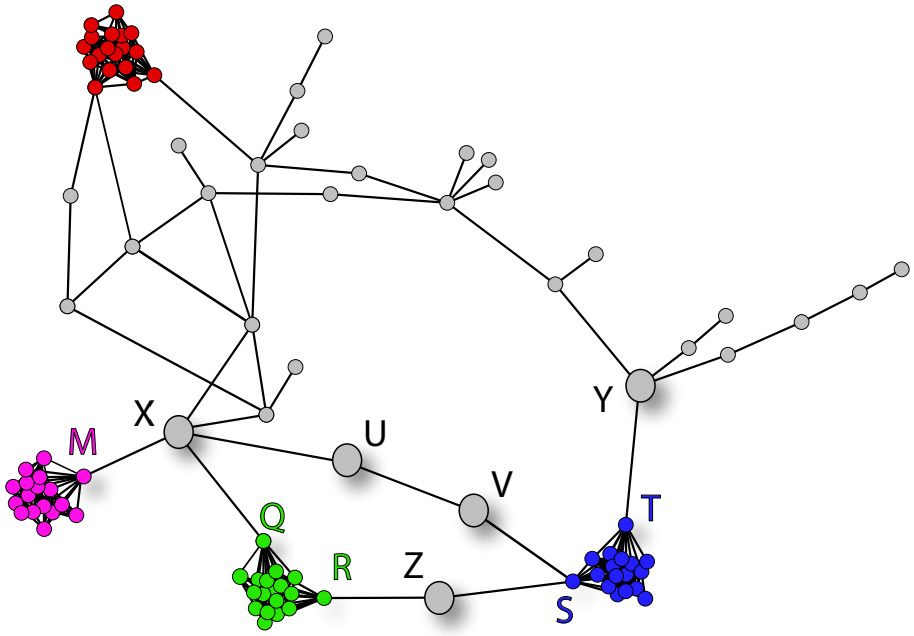


Figure 5.2: Example network with 100 nodes and four modules that contain 70 nodes in total. This figure shows ten network hubs, five of which are module hubs (colored according to the module they belong to) and five are inter-module hubs X, Y, Z, U and V.

However, not all nodes are of same importance for communication in the network. For example, suppose that if node A is deleted from an unweighted, undirected network, it causes disconnection of five modules from the rest of the network. On the other hand, removing node B causes disconnection of three modules from the rest of the network. In this case, we could say that a node A is more important than a node B. However, if in the first case the network breaks into two disconnected subgraphs while deletion of a node B causes the separation of three modules, making the whole network fall apart into four disconnected groups, then the question about which node is more important becomes much harder to answer.

At this point, the natural questions to ask are:

- Can we differentiate between vulnerable nodes in the network, in the sense of their influence on the network robustness?
- Can we develop an importance measure for nodes in the network?
- What should be the main parameter on which such a measure should depend?
- How important should a node be to become a hub?

Determining the importance of a node for the communication between modules is not an easy task and a solution to this problem depends on different factors. These

can be divided into three groups: (1) the topological properties of the network, (2) the dynamical properties of the process defined on the network, and naturally (3) characteristics of the particular real-world system. Considering these factors, in the following we will distinguish between the two main classes of hubs: **inter-module hubs** and **module hubs**. This classification is made depending on whether the node belongs to some module or not. In Figure 5.2 five modular nodes are shown, that are colored according to the module to which they belong to. Inter-module hubs in this figure are X, Y, Z, U and V.

Following the ideas presented above, we will introduce different definitions of hubs, point out the differences between them, present algorithmic approaches for their identification and demonstrate their validity on several examples.

5.3 Inter-module hubs

Here we will study two different natural concepts for declaring a node to be a hub, developed in [51]. Both concepts are based on the same idea

Definition 27 ***Inter-module hubs*** *are nodes $x \in T$ that are essential for the communication between modules of the network.*

Obviously, the first step in identifying inter-module hubs is to determine modules in the network $C_1, \ldots, C_m$. Then, as discussed in Section 4.4, for a certain threshold θ the transition region is given with

$$T = \{x \in V \setminus \bigcup_{i=1}^{m} C_i \,|\, q_i(x) < \theta,\ \forall i = 1, \ldots, m\}.$$

Now, we consider the candidates for the inter-module hubs to be all $x \in T$. From these nodes we will declare as hubs only the ones that are important for the communication between the modules. This communication is established by the random walker making transitions between the modules. However, as we discussed earlier, not all nodes $x \in T$ are of the same importance for the communication between the modules in the same way. To explore this concept further, we will examine two different measures for declaring a node to be an inter-module hub, by presenting two different characterizations of the communication between the modules.

In both approaches, we will consider all m possible choices for the set A from the m given modules $C_1, \ldots, C_m$. For a particular choice of $A = C_i$, we will observe the communication between A and the union of all other modules $B = M_i$. Specifically, we will introduce two measures for the amount of the important communication between A and B, that goes through a particular node $x \in T$. Using this we can discover which nodes are the most important for the communication between module A and the rest of the network, i.e. these nodes that form the essential connections of this module. The removal of such nodes could considerably perturb or even destroy the function of the particularly chosen module in some real-world

networks. In this way, we would be able to control specific functions of many complex systems, such as biological systems, which may lead for example to discovering new drug targets for treating human diseases. Merging the results for all possible choices of the set A, we will be able to obtain a more general quantity that will reflect the importance of a node for the global communication in the network.

5.3.1 Inter-module hubs and reactive flows

First, let us concentrate on transitions from module C_i to the union of all other modules M_i. The global communication properties between these sets are described by the transition rate $k_{C_iM_i}$ (5.8), representing the average number of transitions from C_i to M_i per time unit. In particular, the number of these transitions that are going through a node $x \in T$ is given by the reactive flow $k_x^{C_iM_i}$ (5.10). Now, we can calculate the amount of flow that is going through this particular node x compared to the global flow from C_i to M_i. This will provide us with the measure of importance of a node for the communication in the network. More precisely,

Definition 28 *The importance of the node $x \in T$ when going from C_i to M_i is defined as*

$$p_x^{C_iM_i} = \frac{k_x^{C_iM_i}}{k_{C_iM_i}}, \tag{5.12}$$

that is, the percentage of reactive trajectories going through x, out of all reactive trajectories going from C_i to M_i. The importance of the node $x \in T$ in the network communication is given with

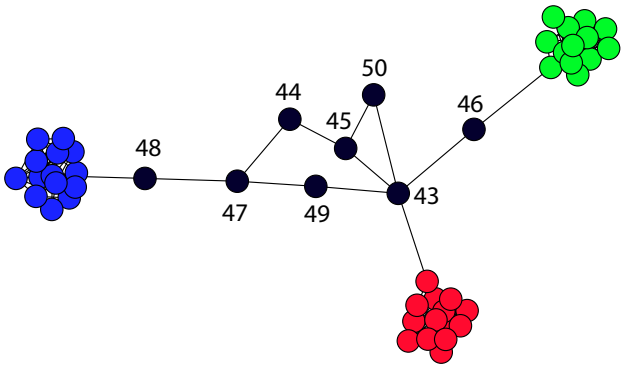
$$p_x = \sum_{i=1,\dots,m} p_x^{C_iM_i}. \tag{5.13}$$

In this sense, a node is said to be an inter-module hub if most of the flow between modules goes through this node. We can now differentiate between potential inter-module hub nodes $x \in T$ in the network according to their importance rate p_x. Moreover, defining a threshold β we can consider all such potential hubs $x \in T$ that have importance rate p_x higher than this threshold to be very important. We will call such nodes the **inter-module hub nodes**.

Remark 6 *In this thesis, we will not develop new strategies for choosing the threshold β. There are two main reasons for this. First, our aim is to determine the connection between the topological and dynamical properties of a network on the one hand and essential elements and functions of the underlying system on the other hand. In particular, we are focused on understanding the link between the network hubs and important elements of the system. Therefore, observing the whole spectrum of potential hub nodes is helpful in order to point out their similarities and differences that can be of importance for further method development. Second, the choice of the threshold β is dependent on the properties of the underlying system. Namely, the importance of hub nodes in real-world networks is that they are supposed to correspond to the functional elements of the system. These elements differ*

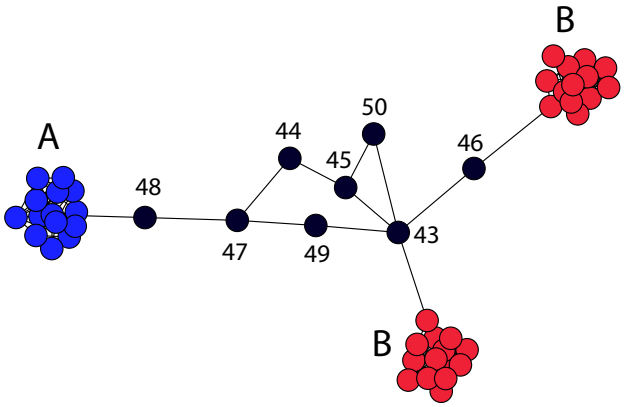
from system to system, hence their properties differ also. Therefore, we propose here a general methodology for identifying hubs as nodes that are important for the communication in the network. For more examples of different criteria and thresholds used to define hubs, we refer to the following references coming from biological systems [78, 21, 10, 91].

Algorithmic realization for identifying inter-module hubs consists of calculating TPT objects, namely committor functions, which satisfy a linear system of equations (3.4); then, computing the transition rate and the reaction flow through a specific node that can be done following Section 5.1.



Node x	p_x
43	3.00
44	0.6987
45	0.6987
46	1.9444
47	1.6303
48	1.6303
49	0.9316
50	0.2329

Figure 5.3: The network shows the modules along with the hub candidates marked with their labels. The table shows the importance rates p_x for all hub candidate nodes.



Node x	p_x^{AB}
43	1.00
44	0.4286
45	0.4286
46	0.3333
47	1.00
48	1.00
49	0.5714
50	0.1429

Figure 5.4: The table shows the importance rates p_x^{AB} when going from A to B, for all inter-module nodes.

Example 18 *Let us now apply our method for identifying inter-module hubs using reactive flows on the example network shown in Figure 5.3. This network consists*

of 50 *nodes,* 42 *of them are arranged in three modules, while being committed to some of the modules with probability of more than* $\theta = 0.9$*. The remaining* 8 *nodes belonging to the transition region serve as the possible inter-module hubs. For these nodes, we can calculate importance rates given by equation (5.13).*
Let us first observe the transition behavior of the random walk process when going from the module A (blue module) to B (red), Figure 5.4a. The importance rates p_x^{AB} *of inter-module nodes, calculated as in (5.12), are shown in Table 5.4b. Nodes* 43, 47 *and* 48 *have the importance rate* 1*, since when going from A to B all communication goes through these nodes. Then from node* 47*,* 42.86% *of reactive trajectories go through node* 44 *and then node* 45*, whereas the rest, that is* 57.14%*, goes through node* 49*.*
In the same way, we can calculate the importance rates for other choice of C_i*. Since the network consists of three modules, the highest possible importance rate is* $p_x = \sum_{i=1,\dots,3} p_x^{C_i M_i} = 3$*. The results are shown in the table of Figure 5.3. Node* 43 *is the only node that connects all three modules. More precisely, if we would remove this node, all three modules would be disconnected from each other and the network would break into three disjoint subgraphs. Therefore, node* 43 *is crucial for the communication in the network and this is reflected by the highest possible importance rate* $p_{43} = 3$*. Node* 46 *is also shown to be important, since it is the only node that connects the green module with the two other modules. However, regarding the global communication in the network, node* 46 *is less important than node* 43 *that connects all three modules. This effect is reflected in their importance rates. Similarly, nodes* 47 *and* 48 *are the only connectors of the blue module to the rest of the network organized as a short chain. Due to the nature of their mutual connection, these nodes transport the same amount of reactive flow and have therefore exactly the same importance rate. Their slightly smaller rate compared to the rate of node* 46 *suggests that the communication between the green and red module is fairly preferred. This can be justified by the fact that the blue module is separated from the other two by several nodes organized as a chain, whereas the red and blue module are divided by only two nodes. This means that the reactive trajectories by which the random walk process transits between the red and blue module need to include just this shorter chain and are therefore preferred.*

5.3.2 Inter-module hubs and important paths

We will now present another novel approach for defining hub nodes. The main idea of this approach is to characterize the communication flow between modules in such a way that we can distinguish between transitions according to their importance. For this, we will use the effective current (5.5) and follow the strategy introduced in Section 5.1.2.

Let us again observe transitions from module C_i to the union of all other modules M_i. In particular, every single transition from C_i to M_i can be characterized by the path the random walker takes from C_i to M_i and we can assign weights to these paths in a sense of how "important" they are. Now, an obvious question to ask is:

what are important paths?
We define important paths as the paths that have the maximal minimal effective current. The reason for this is that the paths with the maximal minimal current can transport the most flow between the modules, representing the crucial connectors of modules. Following this strategy, we can distinguish between important and less important transition paths. Then, we say that a node $x \in T$ is an **inter-module hub** if most of the important transition paths between C_i and M_i go through this node.

Definition 29 *For every node $x \in T$, we introduce its C_iM_i path importance as*

$$s_x^{C_iM_i} = \frac{N_x^{C_iM_i}}{N_{C_iM_i}}, \tag{5.14}$$

where $N_{C_iM_i}$ is the number of most important reaction paths that go from C_i to M_i, and $N_x^{C_iM_i}$ is the number of these paths passing through node x. The importance of a node $x \in T$ in the network is

$$s_x = \sum_{i=1,\dots,m} s_x^{C_iM_i}. \tag{5.15}$$

The quantity s_x represents the global importance of the node, namely the percentage of the total network communication going through node x. Again, defining a threshold β yields highlighting the most important inter-module hub nodes, i.e. the nodes that are taking part in the most intensive communications in the network. For the same reasons that we discussed in Remark 6, we will not discuss strategies for choosing the threshold β.

Remark 7 *It is important to point out that in the following we will not distinguish between the most important reaction paths once they have been obtained. More precisely, after we identify the top N most important reactive paths, we will consider them to be of equal importance. Clearly, there are numerous ways to associate weights to these paths according to their importance. The main difficulty in doing so is that we would have to develop a new strategy for sorting the paths that have been obtained from the communication between all modules in the network. That is, we would have to compare paths going from C_i to M_i for all different choices of the modules $C_i, i = 1, \dots, m$. Furthermore, this distinction should correspond to the appropriate role of modules and their importance coming from a particular application. Because of these reason, we will not develop this point further, but refer the interested reader to the literature about ranking algorithms [45, 68, 47, 2].*

Algorithmic realization of this approach consist of calculating committor functions and the effective flux as in the first approach, and additionally, of identifying the transition pathways and their dynamical bottlenecks [117]. This can be done efficiently by using graph algorithms [3]. In particular, the computational cost for calculating the bottlenecks is given by [115]

Proposition 10
The computational cost for calculating dynamical bottlenecks is in the worst case $O(a \log a)$ *where* a *denotes the number of edges of the directed, weighted graph* $G^* = (V, E, f^+)$.

Another important point to note is that the importance s_x of a node x highly depends on a particular choice of the number of most important reaction paths $N_{C_i M_i}$. This could be seen as an obvious disadvantage of the approach, especially for the algorithmic realization. However, we will see that this dependance allows introducing specific node rankings that correspond to our definition of hubs. We will demonstrate this property in the following example.

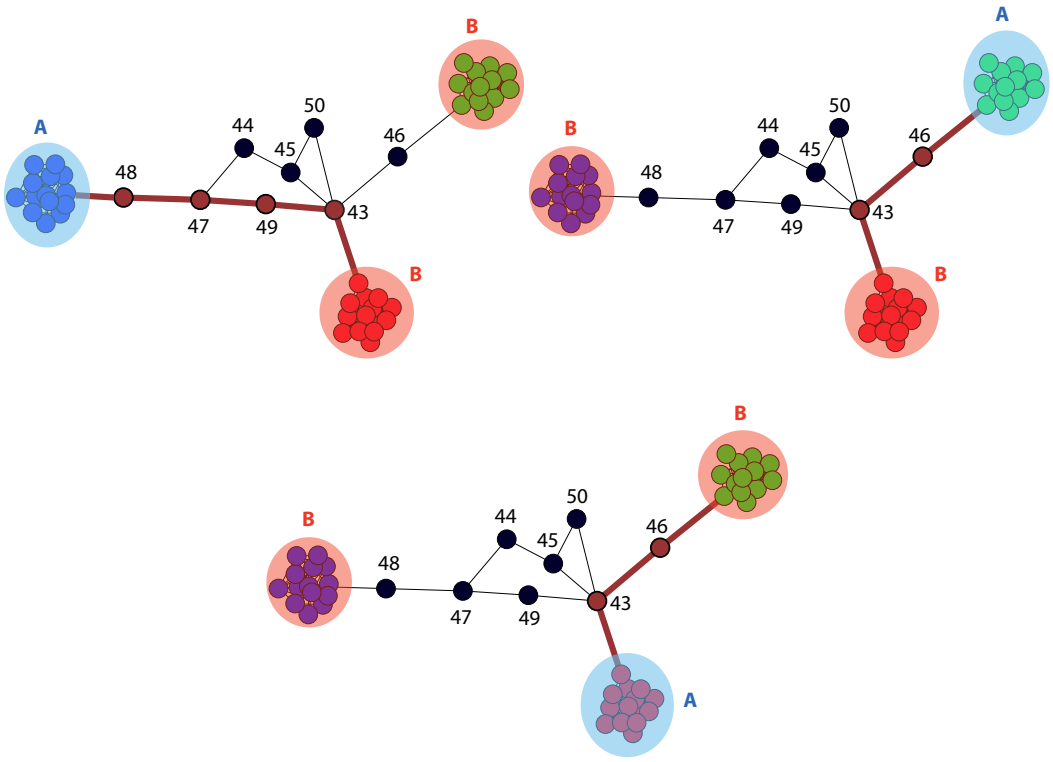


Figure 5.5: This figure shows the first most reactive path when going from A to B, where A corresponds to the blue, green and red module respectively.

Example 19 *Let us now apply our method for identifying inter-module hubs using important paths to the network that was discussed in Example 18. We will show how importance rates change for different choices of the number of important paths, namely for* $N := N_{C_i M_i} = 1, 2, 3$ *and compare the obtained results with the ones from the previous approach. The network consists of three modules that are shown in Figure 5.3 as red, blue and green modules. There are* 8 *nodes that belong to the transition region and are candidates for being inter-module hub nodes. For these nodes we will calculate the importance rates* s_x *given by the equation (5.15). The*

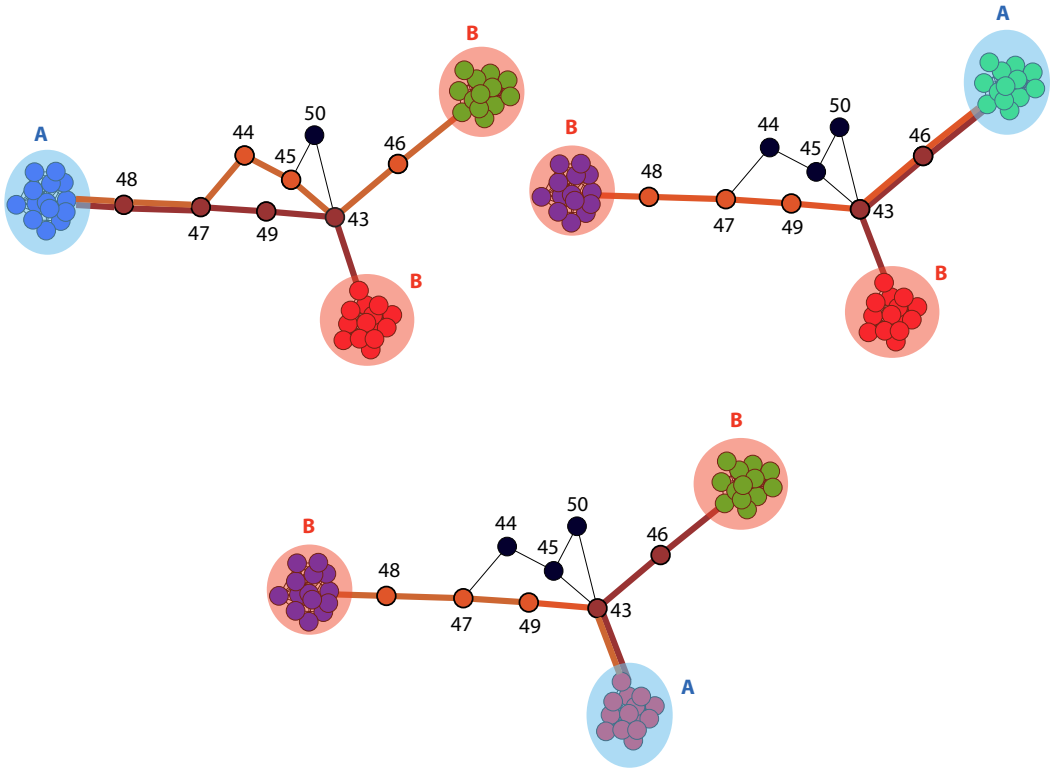


Figure 5.6: This figure shows the $N_{AB} = 2$ most reactive paths when going from A to B, where A corresponds to the blue, green and red module, respectively.

results are shown in Figure (5.8).
In this example we can make three different choices for the set $A = C_i$: the blue, green and red modules. For fixed A, a set B will denote the union of the other two modules. As mentioned above, we will distinguish between different settings in which we consider one, two or three most important reactive paths going from A to B. The paths corresponding to these settings are shown in Figures 5.5, 5.6 and 5.7 respectively. They are colored in the following way: dark brown represents the most important path conducting the most current from A to B, light brown the second most important path and yellow the third most important path.
The node 43 is a node that takes part in all reaction paths, since it connects all modules. Therefore, it has the highest possible importance rate $s_{43} = 3$ for all choices of N. Let us observe how the importance rates of nodes 46, 47 and 48 change for different choices of N. The most important reactive paths $N = 1$ for different choices of A are shown in Figure 5.5, where we see that

$$s_{46} = \frac{0}{1} + \frac{1}{1} + \frac{1}{1} = 2$$

and

$$s_{47} = s_{48} = \frac{1}{1} + \frac{0}{1} + \frac{0}{1} = 1.$$

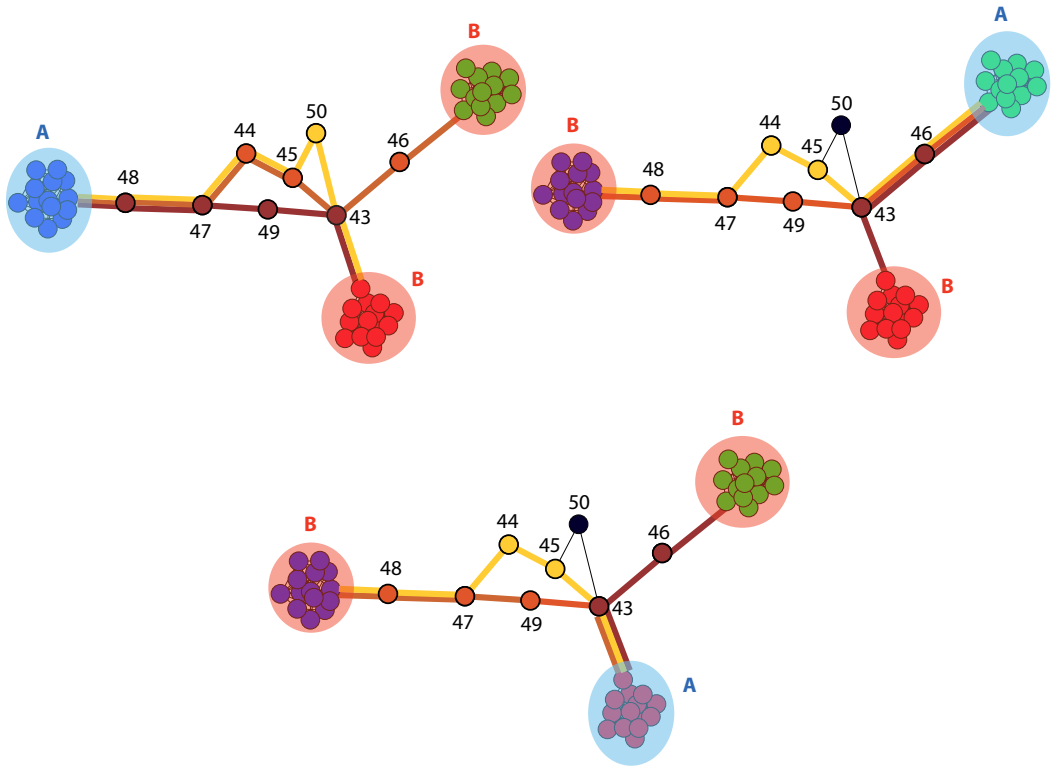


Figure 5.7: This figure shows $N_{AB} = 3$ most reactive paths, respectively when going from A to B, where A corresponds to the blue, green and red module respectively.

	s_x		
Node x	$N = 1$	$N = 2$	$N = 3$
43	3	3	3
44	0	0.5	1.33
45	0	0.5	1.33
46	2	2	1.67
47	1	2	2.33
48	1	2	2.33
49	1	1.5	1
50	0	0	0.33

Figure 5.8: This table shows the importance rates s_x for all hub candidate nodes and different choices of numbers of important reactive paths.

Thus, node 46 *has a higher importance rate than nodes* 47 *and* 48 *since most of the first most important paths go through this node. However, if we observe now the*

two most important reactive paths from Figure 5.6 we see that

$$s_{46} = \frac{1}{2} + \frac{2}{2} + \frac{1}{2} = 2$$

and

$$s_{47} = s_{48} = \frac{2}{2} + \frac{1}{2} + \frac{1}{2} = 2,$$

i.e. nodes 46*,* 47 *and* 48 *are of the same importance. Furthermore, considering the top three most important paths shown in Figure 5.7, it follows that*

$$s_{46} = \frac{1}{3} + \frac{3}{3} + \frac{1}{3} = 1.67$$

and

$$s_{47} = s_{48} = \frac{3}{3} + \frac{2}{3} + \frac{2}{3} = 2.33.$$

That is, if we observe the first three most important paths then nodes 47 *and* 48 *have higher importance rates than node* 46*.*

In the previous example we have demonstrated that for different choice of the parameter N we can obtain different node rankings. Exactly this is the advantage of the approach in applications, because it allows pinpointing the paths of importance and obtaining only their essential elements. This is why our definition of hub nodes states: a node $x \in T$ is an inter-module hub if most of the important transition paths go through this node.
Unlike the first approach, here the most important nodes are not always the ones that are crucial for the connection of a certain module to the rest of the network, unless they are involved in the most important transitions in the network. Obviously, if we consider only one most important path, i.e. $N = 1$, the two approaches are equivalent. This is because the most important path is the one that transports most of the current. In our example, the most important paths shown in Figure 5.5 conduct 60%, 57% and 75% of the transition rate, respectively. Because of this, node 46 has a higher importance rate than nodes 47 and 48 like in the first approach.

5.4 Module hubs

A closer look at the network structure discovers that not only the inter-module nodes play an essential role in the global network communication. Nodes inside modules can also be crucial for the global network communication. To this end, we define

Definition 30 ***Module hubs*** *are nodes $x \in C$ that are essential for the communication in the network.*

Different experimental studies have indicated that in many real-world networks that exhibit a scale-free structure, module hubs are usually of great importance for proper functioning of a particular module, but also of the whole system. This is because module hubs often correspond to the structural key elements of modules that are connecting them to the rest of the network. This role has for example a major bridge or a tunnel on a highway located on the entrance of some city. Although belonging to the elements of the city, these objects are the crucial connectors of this city to other cities. Additionally module nodes that are highly connected to other module nodes are important for the communication within the module itself. They correspond for example to the main crossroad in some city. If this crossroad is closed, this will at least cause substantial disturbance in the transportation system of the city, such as huge traffic jams, changed bus routes etc..

Now, we can express Definition 30 more precisely in the following way:

A module hub is a node $x \in C_i \subset C$ that is either essential for the communication of module C_i with the rest of the network or a node that is important for communication among the nodes of the module C_i.

In order to explore this categorization further, we will distinguish between two types of module hubs: **bottleneck hubs** and **central hubs**. More precisely, bottleneck hubs will denote nodes that are the essential connectors of the module to the rest of the network, whereas central hubs will be considered important for the internal module communication, corresponding to nodes that are well connected to other nodes in that module.

In the following sections, we will develop precise definitions for these objects and explain their importance for the global functioning of the underlying system.

5.4.1 Module bottleneck hubs

Motivated by the topological structure of modular networks and in particular of their modules, we will introduce a new class of module hubs. Let us remember that modules are defined as densely interconnected subgraphs of a network that are characterized by having only a few connecting edges to the rest of the network. Thus, these edges and especially the nodes on these edges control the communication of the module with the rest of the network. We will call these nodes **the module bottlenecks** of the network.

Identification of such modules can be easily done by observing the dominant reaction paths in the network, as described in Section 5.3.2. Recall the definition of reaction paths, presented in Section 5.1.2, where a reaction path from a set A to set B is defined as a sequence of states $w = (i_0, \ldots, i_n), n > 0$ such that $i_0 \in A$, $i_n \in B$ and $i_k \in T$, $\forall k = 1, \ldots, n-1$. In particular, paths of our interest that conduct the most of the current, i.e. dominant reaction paths, are of the same form. This is why module nodes taking part in reaction paths are the nodes that are crucial for connection to other modules in the network.

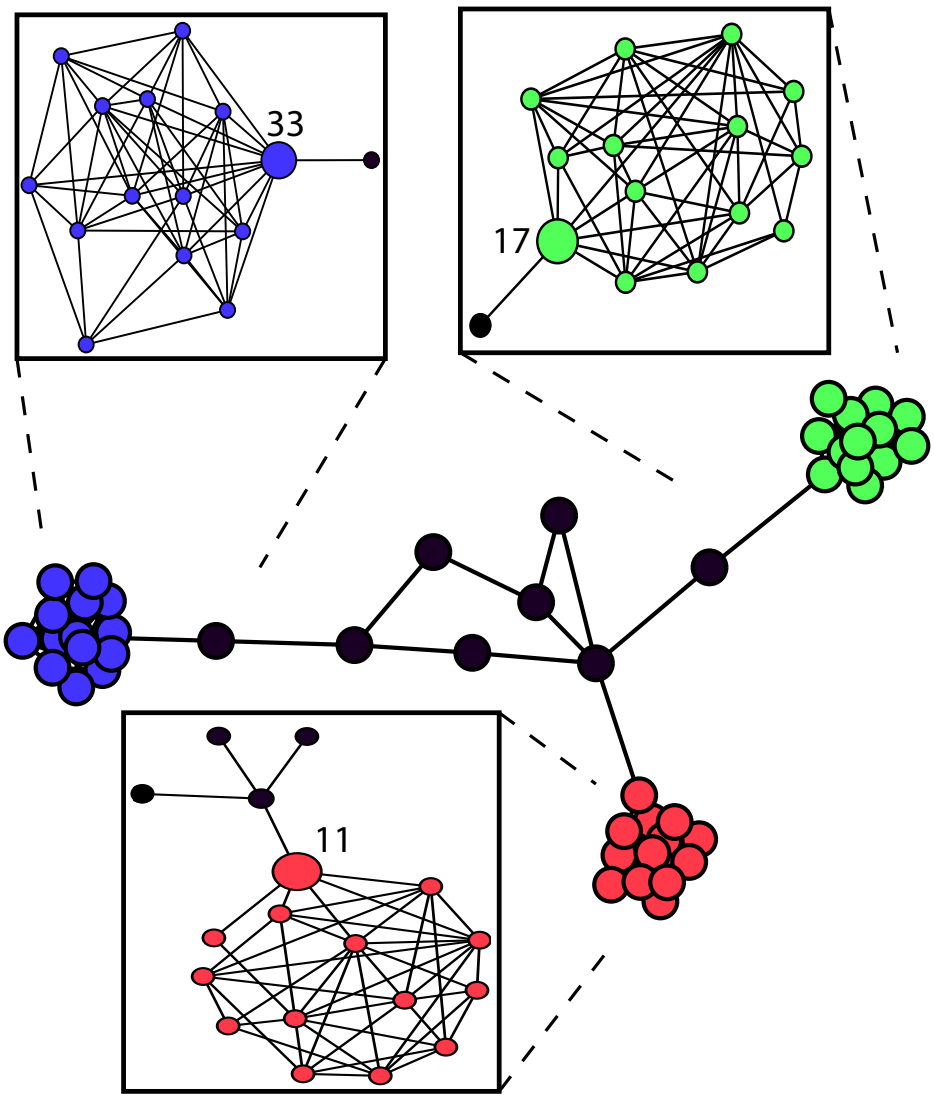


Figure 5.9: This figure shows module bottleneck hubs for each of the three modules in the network.

Figure 5.9 shows the module bottleneck hubs of the network from Example 18, namely node 11 that belongs to the red module, node 33 from the blue module and node 17 from the green module. These nodes belong to the dominant reaction paths illustrated in Figure 5.7. As such, they are the key connectors between the modules they belong to and the rest of the network.

5.4.2 Module central hubs

Using random walk process defined on network, we have already defined several classes of hub nodes that are essential for the communication between modules

in the network. Now we will introduce another class of hub nodes, namely the module nodes that are important for the communication in that module. We will call these nodes **module central hubs**, since they will represent the central points of communication in a particular module.
To this end, let us consider the invariant measure μ of the random walk process, see Section 2.1.2. For a node $x \in V$, the value $\mu(x)$ represents the probability that the random walk process is in a node x. From the definition of μ, we see that $\mu(x)$ is proportional to the degree of a node x. This means that the more neighbors a node x has, the higher the value $\mu(x)$ is. Modular nodes in which the invariant measure has high values will be called **module central hubs**. Here "high values" refers to values that are high compared to the values of μ that have other nodes belonging to the same module. In terms of the random walk process, this means that the probability that a random walker is in a module central hub x is high, what can be interpreted as a high probability that while being in this module the random walker often returns to x. For this reason, we say that module central hubs are important for the communication within the module they belong to.

Figure 5.10 shows the values of invariant measure μ in each of the nodes of the

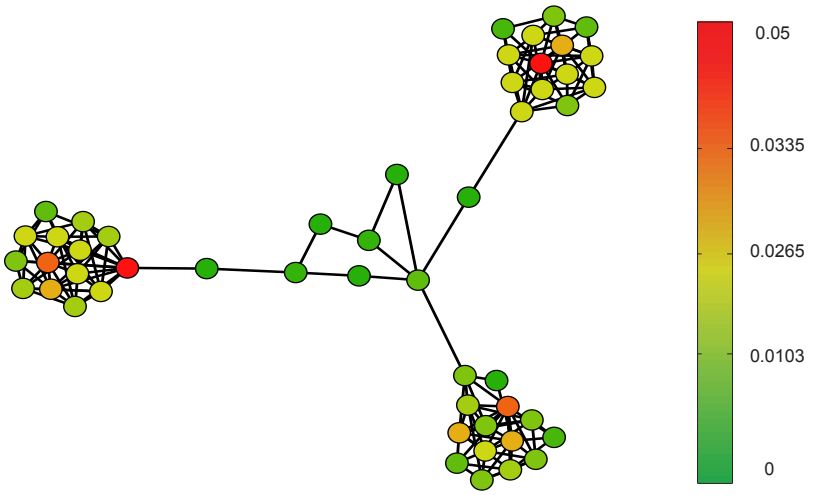


Figure 5.10: This figure shows the values of μ in each node of the example network, where red color indicates high values μ, that is module central hubs.

network from Example 18. Red color indicates higher values of μ, that is module central hubs, whereas green color corresponds to the nodes for which the probability that they are visited by the random walker is low. At this point we should note that sometimes module central hubs may coincide with module bottleneck hubs. The reason for this lies in the fact that the two measures reflect two different properties of the network, as introduced above. For example, in Figure 5.9 node 33 is at the same time the bottleneck hub and the central hub of the blue module. This is because it takes part in all most important reactive pathways of the blue module and has high value of μ (indicated with red color in Figure 5.10), which is influenced

by the fact that it has 10 connections with the other 14 nodes in this module. In the next section we will present an already established and accepted measure of node importance based od node degree. We will see that this measure is closely related to our definition of module central nodes.

5.5 Example: Finding hubs

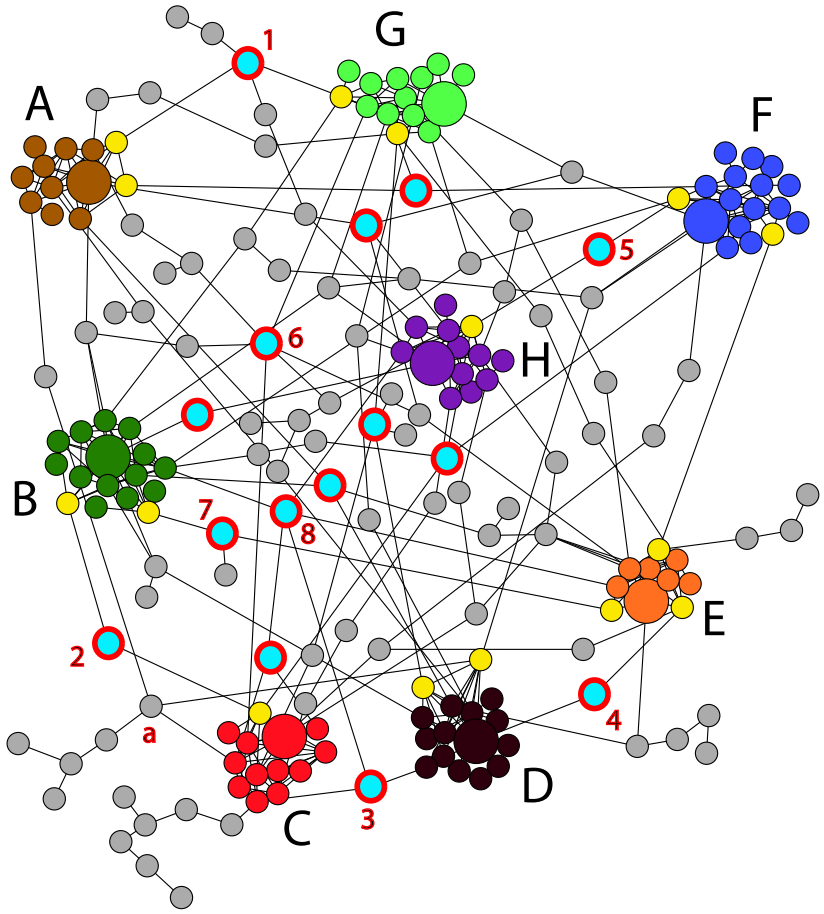


Figure 5.11: This figure shows the result of our hub analysis for a network with 200 nodes. Using our module identification algorithm (see Chapter 4) we find 8 modules marked with letters A, ..., H and colored with different colors. We find: inter-module hubs (marked with light blue color and red border), module bottleneck hubs (marked in yellow) and module central hubs (marked as the largest nodes of each module).

In this section we will conduct complete hub analysis using our approach presented in this chapter on example network that has 200 nodes and shown in Figure 5.11. Using our algorithm for identification of modules (for $\theta = 0.8$, see Chapter 4) we find that 111 nodes belong to 8 modules shown in Figure 5.11, where they are

marked with letters A, ..., H. Furthermore, every module is colored differently.

Let us start by finding module hubs for each of the eight modules. Using the approach from Section 5.4.1 we find 18 module bottleneck hubs, marked in yellow in the figure. Three of these bottleneck hubs are at the same time module central hubs of modules B, D and H (see Section 5.4.2). Module central hubs are colored according to the module they belong to and are marked as the largest nodes of these modules.

Now we will analyze the transition region in order to find inter-module hubs. First, for all nodes that belong to the transition region we calculate importance rates given by equations (5.13) and (5.15), using $N_{C_iM_i} = 3$. For the sake of simplicity, we highlight only some of these nodes, namely the top 15 with the highest importance rates. These nodes correspond to inter-module hubs and are marked with light blue color having red border. Out of these nodes we picked eight nodes (marked with numbers in Figure 5.11) to provide the interpretation of their importance in this network. For example, nodes 1, 2, 3, 4, 5 and 7 are nodes that form a direct connection between two modules, i.e. node 1 connects modules A and G, 2 connects B and C, 3 connects C and D etc.. At this point we should remark that there are also other nodes that take part in the shortest paths between these modules, but these are not marked as hubs. For example, node a directly connects modules C and D, but is not a hub node unlike node 3. This is because most of the communication flow between C and D goes through node 3. Compared to node 1, node 6 has higher importance rate, since it is a node that connects module G to module C and furthermore, transports large amount of communication flow between modules H, E , A and B. Node 8 has the highest importance rate, as the most of the flow between three modules B, H and D go through 8.

5.6 Related work

There have been many attempts to characterize "special" nodes in graphs that have important functional features in the underlying system. Most of these consider different topological properties of graphs in order to provide a quantitative measure of node importance. Here we will present some of the state-of-the-art approaches for identifying such nodes that are "important" in the sense of their interpretation in the underlying system.

A large number of real-world networks have been shown to be scale-free (see Section 1.2.2), i.e. to have a power-law degree distribution. Many types of social networks, biological networks, transportation networks and computer networks are only some examples of scale-free networks. This class of networks is characterized by the existence of **highly connected nodes**, called **hubs**. Most graph theoretic approaches focus on identifying such nodes, since they may serve as central structural elements of modules, as well as key connectors of different modules and therefore, represent the most vulnerable points in a network [15, 18, 18, 89, 8, 43, 44]. Different stud-

ies have confirmed this assumption, for example, it has been observed that highly connected nodes are related to essential genes in regulatory networks [168].

However, node degree is not always a sufficient measure for characterizing the essential points of scale-free networks [73, 81, 169]. Recent studies have shown that **topological bottleneck** nodes, defined as nodes that have high betweenness centrality, can correspond to important functional elements of the underlying system. More precisely, bottleneck nodes are the nodes that take part in a large number of shortest paths between two nodes of a graph. Since one considers all pairs of nodes in the network, bottlenecks can be understood as those nodes that take part in most of communication in a network. Obviously, bottleneck nodes do not need to be highly connected. Different experimental results have shown that bottleneck hubs often have different interpretation than highly connected nodes. For example, in protein-protein interaction networks, bottlenecks are more likely to correspond to essential proteins [92]. Therefore, bottleneck centrality is often used as a measure for identifying potential hub nodes, especially in biological networks.

A new way of classifying hub nodes in protein-protein interaction networks resulted from including temporal information of a system [78]. In particular, proteins that interact with other proteins at the same time are called **party hubs**, whereas proteins that interact with other proteins at different times are called **date hubs**. Party hubs are assumed to have a central modular role in the network, as being connected to many other proteins with the same function. Although the removal of party hubs results "only" in a local decomposition of a network, namely the decomposition of some functional module, party hubs are observed to be essential proteins. On the other hand, date hubs have more global role, since they are shown to be the nodes that connect different functional modules. Date hubs are also observed to be essential, since their removal can have a very harmful effect on the network such as breaking the network into disconnected subunits. Despite many recent debates about the evaluation of experimental results and their interpretation [1], this approach has introduced an important novelty in studying biological networks that includes merging statical and dynamical components of the observed system.

Many studies consider combining different topological properties of a network in order to find essential network elements that would unite many important properties of hubs. For example, a combination of node degree, motif distribution, betweenness centrality and closeness centrality is shown to be useful for reliable identification of hub regions in macaque and cat brain networks [157].

Apart from the state-of-the-art paradigms presented here, many other approaches have been developed with the same purpose. These can be seen in different review papers. An extensive report about the most relevant paradigms used for identifying hubs in protein-protein interaction networks has been presented in [75].

Compared to other paradigms, the main novelty of our approach is that it is based on defining a dynamical process on a network that reflects the main properties of

that network. Our algorithm for identification of hubs is based on features of this process and is crucially dependent on network modules, as they represent the essential functional elements of the network. One more advantage of our approach lies in the fact that it is in general application-independent, but at the same time flexible for adjusting to a particular application. In Chapter 6 we will verify our algorithm on some real-world examples.

Chapter 6

Analyzing real-world networks

Analyzing networks that describe real-world complex systems is an important task, as it could could yield valuable information about the basic principles of the underlying system. Identifying network modules could lead to revealing previously unknown functional similarities between the elements of the system [80, 136]. On the other hand, finding hubs could help uncovering system elements that have a special importance for the functioning of the whole system. All this information resulting from analysis of real-world networks could deepen our understanding of the structural organization and complex mechanisms of different real-world systems.

Analyzing real-world networks is a very challenging task, since interpretation of the obtained results in terms of the real-world system is not always clear. In particular, drawing conclusions that relate structures in complex networks, such as modules and hubs, to functional elements of the underlying system is not explicitly theoretically justified. Furthermore, when dealing with real-world data one often encounters a few typical problems, such as:

- Due to often incomplete and biased experimental strategies, resulting real-world data-sets are usually noisy and not reliable. This of course, strongly influences inferring the meaningful conclusions from networks [9, 133].
- Determining the quality of obtained results is often a very difficult task, since complete and correct relations between the elements of the underlying system are not always known [77].
- Evaluation of obtained results should be done in comparison with putative information about the system. However, it is important to note that these sources should consider the same characteristics of the system as the information the network is based on (see Section 6.1 for more details).

In this chapter we will analyze two examples of real-world networks that were already considered in different studies. In Section 6.1 we will show an example of a social network and in Section 6.2 an example of a biological network. We will apply our algorithms for finding modules and hubs using the random walk process and compare the obtained results to the results that are considered to be well-accepted according to different studies. In the evaluation of our results we will see

that network modules do not necessarily correspond to functional modules of the underlying system that were determined by specific experiments. It is a topic for future research whether dealing with reliable data would provide results that are functionally relevant in terms of the underlying system.

6.1 Analyzing US political books network

In this section we will demonstrate our new method for finding modules by analyzing a network of US political books, which was introduced in [124] (see Figure 6.1). This network contains 105 nodes, each representing a book about US politics sold by the online retailer Amazon. An edge between nodes (books) A and B exists if customers frequently bought books A and B together[1].

In [124], these books have been classified into three categories of political alignment:

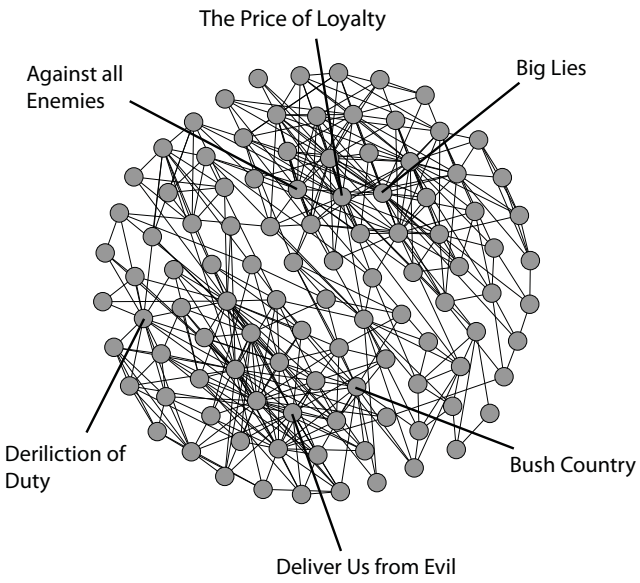


Figure 6.1: The political books network. (Taken from [124])

conservative, liberal, and neutral. This classification has been done according to the author's personal judgment that was based on the descriptions of the books on Amazon. This manual, hard clustering is illustrated in Figure 6.2.

Now we will apply our approach to identify modules in this network (see Chapter 4) and compare the obtained results with the one presented in [124]. In order to do this, we will first apply our heuristic algorithm to obtain the initial guess for the modules (see Section 4.4.1). After performing step 1 for $\alpha = 100$, we identify the

[1] According to the Amazon's "Customers who bought this book also bought..." feature

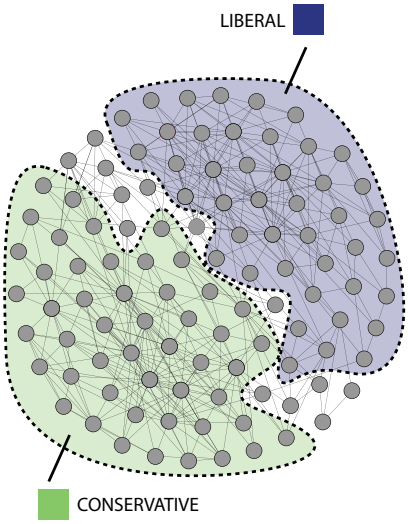


Figure 6.2: The assignment of the books to three categories: conservative (green), liberal (blue) and neutral (without coloring), according to Newman's personal judgment as in [124].

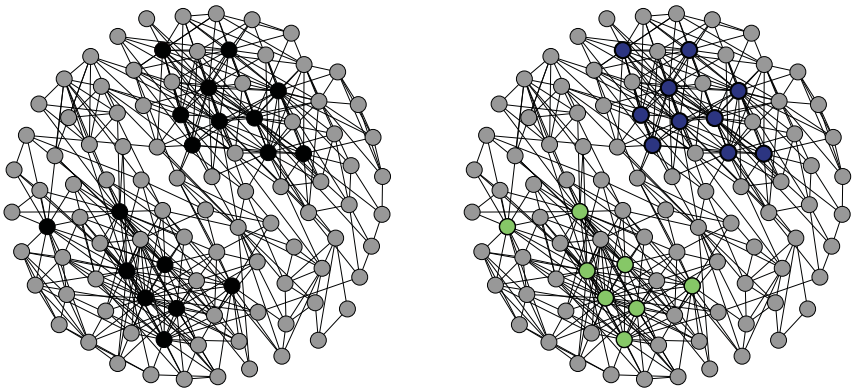


Figure 6.3: Left: Black nodes have been identified by the algorithm to belong to modules. Right: Clustering the nodes within modules using PCCA+.

region of modules $\mathcal{M}^\alpha$, which is colored black in left plot of Figure 6.3. Setting aside the transition region we can then compute eigenvalues of $\hat{P}_\alpha(x, y)$. The five largest eigenvalues are: $1.00, 0.97, 0.31, 0.25, 0.18$. As we noted previously, the resulting spectrum is convenient to interpret: It is easy to see that there is a clear gap after the first two eigenvalues, indicating that clustering into two modules is the natural choice for this example. Next, we perform hard clustering (step 2) to assign nodes

to these two modules. Right plot of Figure 6.3 shows the final modules computed in this step.

Finally, by minimizing the eigenvalue error (4.19), we find the optimal modules, shown in Figure 6.4. For two different choices of a threshold θ, namely $\theta = 0.9$ and $\theta = 0.8$, this figure shows the resulting modules, each consisting of nodes that have the affiliation to one of the modules higher than θ.

We find that most books that have been classified in [124] as belonging to the

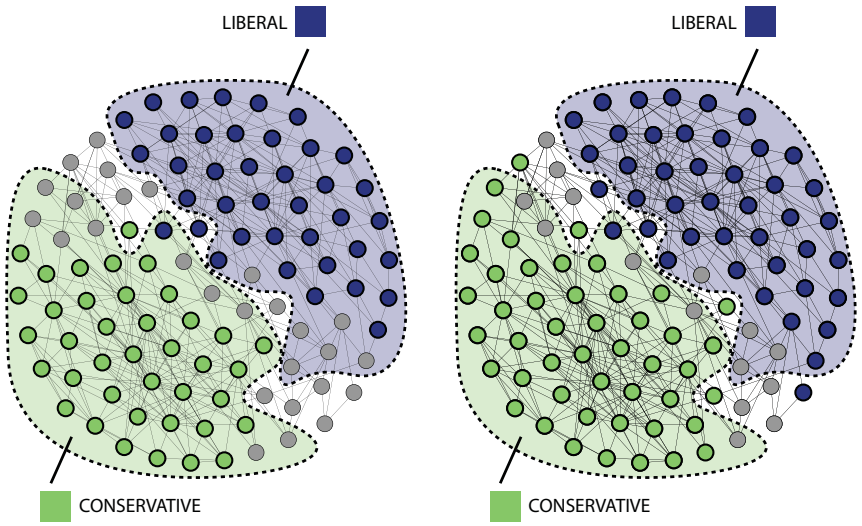


Figure 6.4: Left: Nodes with affiliation higher than $\theta = 0.9$. Right: Nodes with affiliation higher than $\theta = 0.8$

conservative or liberal group, also have a rather high affiliation to one of these modules found by our algorithm. On the other hand, for most of the neutral books we find an affiliation which is less specific. Moreover, if we form hard clustering by assigning every book to a module to which it has the highest affiliation, then all liberal books will be in the same cluster and only two books that have been classified as conservative will be merged into the liberal cluster.

It is important to note that interpretation and comparison of these results has to be done very carefully. First, we cannot expect any clustering algorithm applied to this network to uncover the hard assignment that was based on much more background information than what is used for making this network. Furthermore, the information this network is based on is very different from what the modules in [124] represent. Namely, for constructing the network we have only used the selling statistics of Amazon. In this sense, the result of our method is that there are two strongly interconnected groups of books that have been purchased frequently by the same customers. Now, one could formulate the hypothesis that people having a particular political disposition would rather buy corresponding books. The results above would support this idea, but such a simplification cannot hold for every single book.

Furthermore, a hard assignment to the three categories (conservative, liberal and neutral) that has been done in [124] may not completely correspond to the natural clustering, since not all books have a clear affiliation to one category, but rather a stronger affiliation to one class than others. Therefore, one should not always expect that a full partitioning of a network like this will match the background information fully. Our approach for identifying modules in the network has the advantage that we cluster only these books for which an interpretation really seems to exist, in terms of strong affiliation of books to some group. For the remaining books we then compute their tendencies to belong to some group. We have seen that using this algorithmic strategy we are able to uncover very accurate and interesting connections, which are also interpretable.

6.2 Analyzing yeast PPI network

In this section we will demonstrate the performance of our methods introduced in previous chapters, on a real biological network, the Filtered Yeast Interactome (FYI) network from [78]. This is a protein-protein interaction (PPI) network of Saccharomyces cerevisiae that was created manualy by intersecting data from several large-scale experiments. The resulting network consists of 1379 nodes, representing proteins and 2493 edges, where an edge between two nodes exists if the interaction between the corresponding proteins has been verified by multiple experiments. Here, we will analyze the largest connected component of the FYI network that contains 778 nodes. This network is shown in Figure 6.6.

Let us first apply our algorithm for module finding (see Chapter 4), using the

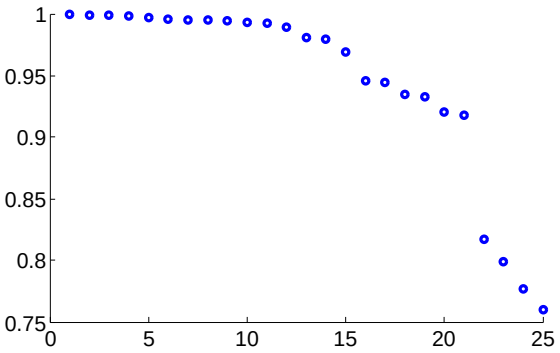


Figure 6.5: The first 25 eigenvalues of $\hat{P}_\alpha$, for $\alpha = 1000$.

heuristic approach introduced in Section 4.4.1, to obtain the initial solution of the optimization problem given by (4.32). For $\alpha = 1000$ the eigenvalues of $\hat{P}_\alpha$ are shown in Figure 6.5. A clear gap after the 21^{st} eigenvalue indicates that there exist 21 modules in this network. For $\theta = 0.9$, the resulting 21 modules found by our approach consist of 498 nodes. These modules are shown in Figure 6.6, where each modules is colored with a different color. The transition region consists of 280 nodes

that are colored in light gray.

In the study of PPI networks a common approach for evaluating the obtained network modules is to compare them with conjectured proteins complexes [36]. The main idea is that proteins that are grouped together share similar properties or functions. This means that the modules we have identified should display functional coherence. In order to check whether our results confirm this, we will compare the identified modules to the protein complexes listed in the CYC2008 [134] data-set. For this comparison we projected 139 complexes on the network, including very small complexes with only two proteins.

Figure 6.6 shows the CYC2008 protein complexes together with the optimal modules obtained by our algorithm. We find that most of the larger complexes coincide almost completely. For example:

- *the cytoplasmic ribosomal small subunit complex* consisting of 23 proteins (module A in Figure 6.6) is completely identified;
- *the* $19/22S$ *regulator complex* (module E) having 17 proteins is completely identified;
- for *the nuclear exasome complex* (module B) having 9 proetins, our method correctly identified 8 proteins;
- for *the Arp 2/3 protein complex* (module D), our method correctly found 6 out of 7 proteins;
- for *the Cytoplasmic ribosomal large subunit complex* (module C), we have identified 37 out of 41 proteins;
- for *the 20S proteasome subunit* (module F), with 14 proteins our method correctly identified 13 proteins.

However, our method was not able to identify very small complexes of size $2-3$, as they do not form network modules in the sense of their topological definition (see Section 13).
A more detailed biological evaluation of our results could be done using annotations from the Gene Ontology (GO) database [12]. This database provides attributes for describing genes and gene product (RNA and protein) across all species using a defined set of annotation terms. Using this, functional coherence of modules can be interpreted such that proteins that belong to the same module are annotated with similar GO annotation terms. GO Term analysis of the modules obtained by our approach can be found in [95].

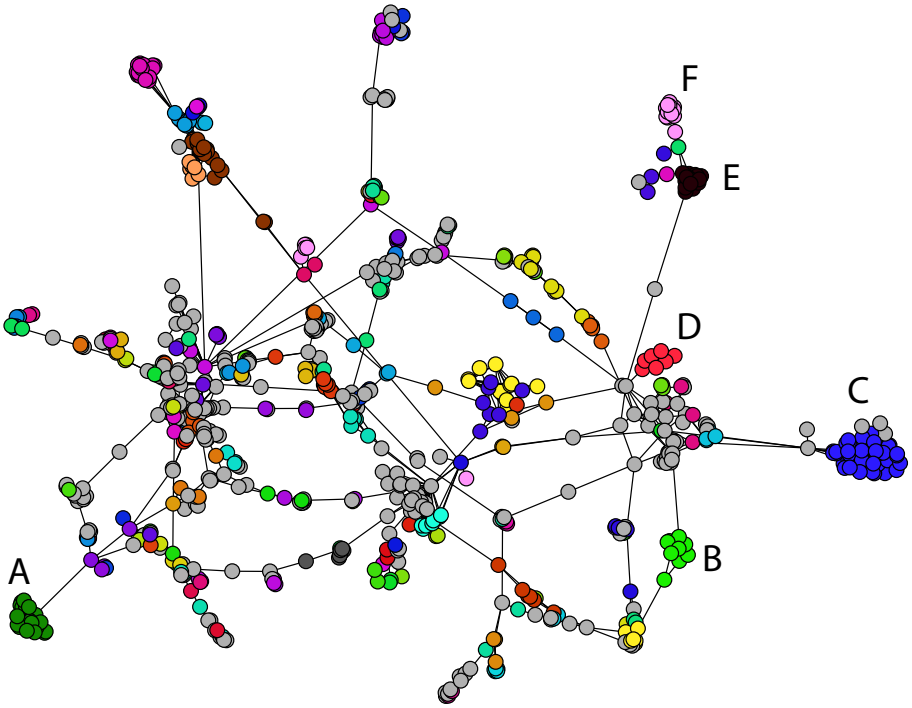
The second part of our network analysis focuses on finding hub nodes using the methods from Chapter 5. For the sake of simplicity, we highlight only some of these nodes, namely one central hub node per module with the highest importance rate and for non-central hubs we highlight nodes with importance rates higher than

0.4. Figure 6.7a shows inter-module hubs marked as light blue nodes, module bottleneck hubs as yellow nodes and module central hubs as pink nodes. All other module nodes are colored in black and nodes belonging to transition region in light grey.
Many of the identified hub nodes show high correspondence with the essential proteins identified in [78]. In Figure 6.7a we have labeled some of these nodes with the names of the underlying proteins. In order to further support the biological importance of identified hubs, we will use the GO Term analysis. Summarized, we find that many of the identified inter-module hubs share the same annotation terms, representing large percentage of proteins which are important for some specific process, such as mRNA processing. We have highlighted the following three groups of hubs according to their GO annotation terms that they share:

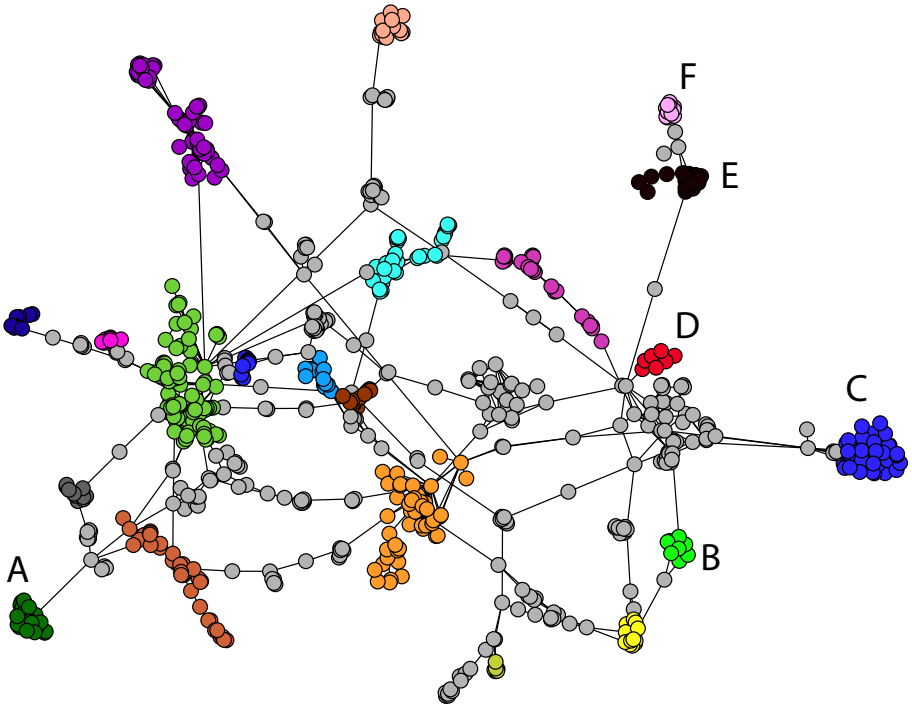
- group A - red: consists of five hub nodes that all take part in polarized growth and cell division process;
- group B - blue: consists of three hubs nodes which are all involved in mRNA processing;
- group C - green: consists of five hub nodes which are all involved in nuclear transport process.

For more detailed analysis of biological interpretation of these nodes see [95].

Finally, let us refer to the importance rates of identified inter-module hubs. Figure 6.7b shows six inter-module hub nodes four of which, i.e. nodes 1, 2, 3 and 4 have the highest importance rates, with values between 2 and 2.3 calculated according to the formula (5.15). Nodes 5 and 6 have both the importance rate 1, coming from the fact that they are the key connectors for each of modules D and E to the rest of the network. From Figure 6.7b we see that nodes 1, 2, 3 and 4 are important for the communication between several modules. For example, node 2 with importance rate 2, is a key connector of module A with the rest of the network. On the other hand, from its importance rate we conclude that this node is also important for the communication between modules A, B and C, i.e. most of the communication between these modules goes through node 2. As we have seen from the functional evaluation of identified modules, module A corresponds to the cytoplasmic ribosomal small subunit complex. This implies that inhibiting the protein corresponding to node 2 could have a huge impact on the functioning of the cytoplasmic ribosomal small subunit complex. However, only reliable experimental techniques could reveal whether importance of such nodes in terms of networks can mirror the appropriate functional importance of associated proteins.

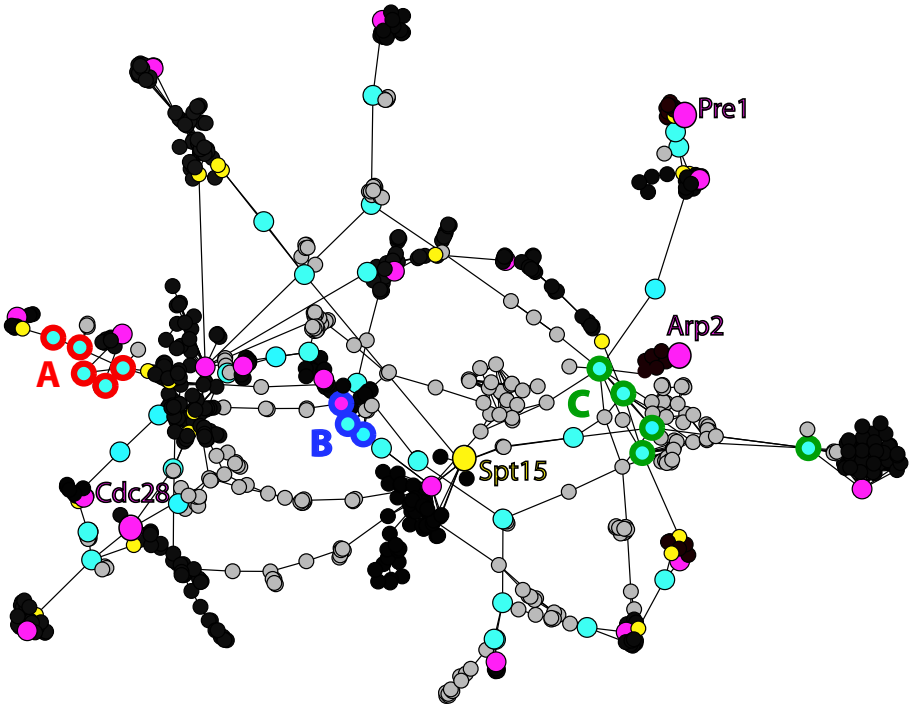


(a) The CYC2008 protein complexes.

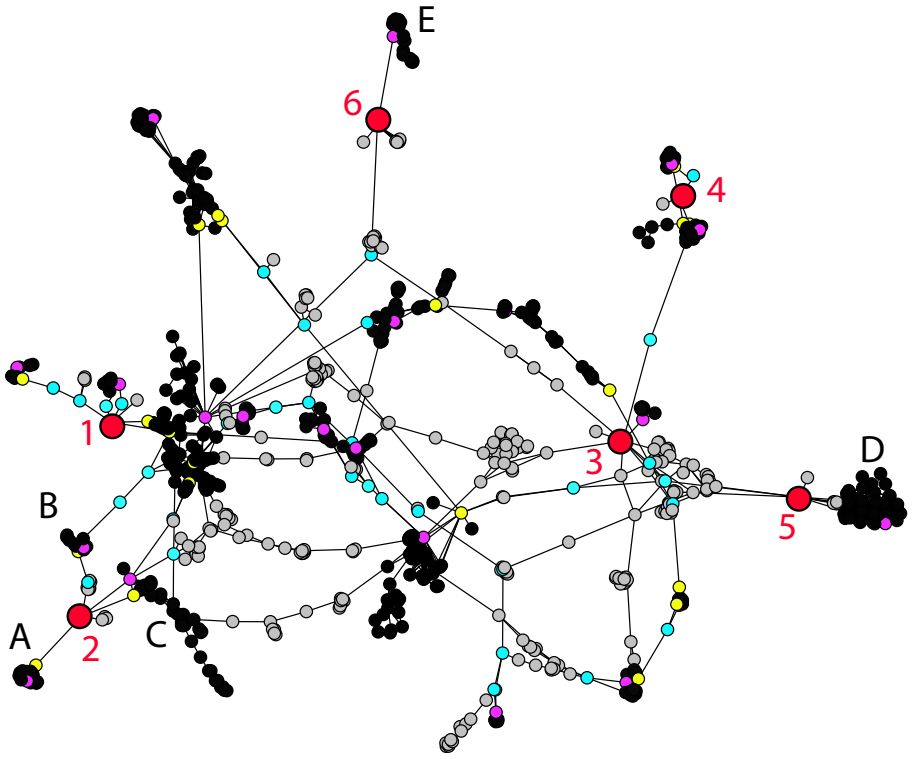


(b) 21 modules found by our algorithm.

Figure 6.6: The largest connected component of the yeast protein-protein interaction network containing 778 nodes. Top figure shows the protein complexes listed in the CYC2008 data-set. Bottom figure shows the 21 modules that were identified by our algorithm. Each module is colored with a different color. Large complexes that were identified as modules in both figures are marked with letters and colored with the same color in both figures.



(a) GO Term analysis of identified hub nodes.



(b) This figure highlights in red color six inter-module hub nodes of FYI network that have high importance rate, i.e. higher than 1.

Figure 6.7: Hub nodes of FYI network, where light blue colored nodes are inter-module hubs, yellow nodes are module bottleneck hubs and pink nodes are module central hubs.

Chapter 7

Analyzing directed networks

In this chapter we will present a novel method for analyzing directed networks using the random walker approach. Properties of random walks on directed networks are significantly different than in the case of undirected networks due to the fact that adjacency matrices of directed networks are in general **not symmetric**. As a result, the existing algorithms for analyzing undirected networks are inapplicable for directed networks.

There have been many attempts to generalize methods for analyzing undirected networks to the case of directed networks. The most common approaches consider forming undirected networks from the directed networks by ignoring directions of edges [42] or perform different types of symmetrization of relevant matrices [113, 146]. However, these assumptions can produce very different results from the original information coming from the underlying system.

Only a few methods have been extended to the case of directed networks [65, 104, 139, 97]. One example is the modularity maximization method (Section 4.5), for which several definitions of directed modularity have been proposed over the last years [11, 104, 97]. The main difference between these methods lies in the fact that they use different definitions for modules in directed networks, as there is no generally accepted definition yet.

In terms of random-walk-based approaches, the asymmetry of the adjacency matrix causes the non-reversibility of the random walk process. Therefore, the algorithms that we have developed in the previous chapters of this thesis have to be adopted to the case of directed networks. In Section 7.1 we will present a new general approach for analyzing non-reversible processes, that can be defined also on continuous state spaces. In particular, we will propose a generalization of our spectral-based approach for finding metastable sets to the case of non-reversible processes. Then, specifically for the case of directed networks we will define two non-reversible random walk processes [26], namely the forward and the backward random walk process (see Section 7.2). In Section 7.2.1 we will address the problem of finding modules and hubs in directed networks using two random walk processes. However, since it is not yet clear how metastable sets of random walk processes are related to modules in directed networks, we will not present here an algorithmic approach for module

identification in directed networks, as we did for the case of undirected networks. But we will show how the ideas presented in this chapter could be used as a starting point for developing methods that could enable more detailed analysis of directed networks. We will end this chapter by pointing out some relevant open problems connected to defining modules and hubs in directed networks (Section 7.2.2).

7.1 Analyzing non-reversible Markov processes

In Section 2.1.2 we pointed out fundamental properties of **time-reversible** Markov processes. These processes are characterized by the property that their behavior is invariant under the reversal of time. Furthermore, the following proposition holds [129]

Proposition 11
A stationary Markov process (X_t) *is time-reversible if and only if the* ***detailed balance condition*** *is fulfilled*

$$\mu(x)p(x,y) = \mu(y)p(y,x), \tag{7.1}$$

where μ *is a stationary distribution of the process and* $p(x,y)$ *is a transition probability from a state* x *to a state* y*.*

However, many observed processes are *not* time-reversible. Such processes will be studied in this section.

7.1.1 Transfer operators of non-reversible processes

We consider two non-reversible Markov processes (X_t) and (Y_t) on a continuous state space S, its two associated transition functions $p_i : V \times V \to [0,1]$, $i = 1,2$ and two *positive* measures μ_i, $i = 1,2$ such that

$$\mu_2(y) = \int p_1(x,y)\mu_1(x)dx \tag{7.2}$$

$$\mu_1(y) = \int p_2(x,y)\mu_2(x)dx. \tag{7.3}$$

Let us introduce two scalar products

$$\langle f,g\rangle_i = \int f(x)g(x)\mu_i(x)dx,\ i = 1,2$$

in Hilbert spaces $H_i = L^2_{\mu_i}$ and two associated **transfer operators**

$$(T_{12}f)(y)\mu_2(y) = \int p_1(x,y)f(x)\mu_1(x)dx \tag{7.4}$$

$$(T_{21}f)(y)\mu_1(y) = \int p_2(x,y)f(x)\mu_2(x)dx. \tag{7.5}$$

We can connect the two weighted spaces H_i with the unweighted space L^2 by means of the linear transformations (multiplication operators)

$$J_i : L^2 \to H_i, \quad g_i = J_i g, \quad (J_i g)(x) = \frac{1}{\sqrt{\mu_i(x)}} g(x). \tag{7.6}$$

These two transformations have the nice property that the respective norms are preserved

$$\begin{aligned} \|g\|^2 = (g,g) &= (J_i^{-1} g_i, J_i^{-1} g_i) \\ &= \int g_i(x)^2 \mu_i(x) dx = \langle g_i, g_i \rangle_i = \|g_i\|_i^2, \end{aligned}$$

where $(\cdot,\cdot)$ denotes the standard scalar product in L^2.

Operators $T_{12} : H_1 \to H_2$ and $T_{21} : H_2 \to H_1$ have the property $T_{12}\mathbb{1} = \mathbb{1}$ and $T_{21}\mathbb{1} = \mathbb{1}$. Furthermore, for arbitrary $f, g \in L^2$ with $f_i = J_i f \in H_i$ and $g_i = J_i g \in H_i$, it holds that

$$\begin{aligned} \langle g_1, T_{21} f_2 \rangle_1 &= \int\int g(y) \frac{1}{\sqrt{\mu_1(y)}} p_2(x,y) f(x) \sqrt{\mu_2(x)} dx dy \\ &= (g, P_{21} f), \end{aligned} \tag{7.7}$$

where P_{21} is the representation of T_{21} in L^2 and

$$(P_{21} f)(y) \sqrt{\mu_1(y)} = \int p_2(x,y) f(x) \sqrt{\mu_2(x)} dx.$$

Then, the typical kernel form of the operator P_{21} is

$$(P_{21} f)(y) = \int \pi_{21}(x,y) f(x) dx, \tag{7.8}$$

with the kernel

$$\pi_{21}(x,y) = \frac{1}{\sqrt{\mu_1(y)}} p_2(x,y) \sqrt{\mu_2(x)}.$$

From (7.7), using linear transformations J_1 and J_2 we calculate that

$$\begin{aligned} (g, P_{21} f) &= \langle g_1, T_{21} f_2 \rangle_1 = (J_1^{-1} g_1, J_1^{-1} T_{21} f_2) \\ &= (g, J_1^{-1} T_{21} J_2 f), \end{aligned}$$

so that

$$P_{21} = J_1^{-1} T_{21} J_2. \tag{7.9}$$

In analogy we have

$$\langle T_{12} g_1, f_2 \rangle_2 = (P_{12} g, f),$$

with

$$(P_{12} f)(y) = \int \pi_{12}(x,y) f(x) dx, \qquad \pi_{12}(x,y) = \frac{1}{\sqrt{\mu_2(y)}} p_1(x,y) \sqrt{\mu_1(x)} \tag{7.10}$$

and

$$P_{12} = J_2^{-1} T_{12} J_1. \tag{7.11}$$

7.1.2 Extended detailed balance condition

Having defined two transfer operators T_{12} and T_{21} and their representations in L^2 given in (7.8), (7.10), we can prove the following theorem:

Theorem 10
*Operators P_{12} and P_{21} are adjoint in L^2 with standard scalar product, i.e. $P_{21}^* = P_{12}$ if and only if the **extended detailed balance condition***

$$\mu_1(x)p_1(x,y) = \mu_2(y)p_2(y,x) \tag{7.12}$$

is satisfied.

Proof. We have $P_{21}^* = P_{12}$ in L^2 if and only if $\pi_{12}(x,y) = \pi_{21}(y,x)$ which is equivalent to the extended detailed balance condition. □

In the following examples, we will show that the extended detailed balance condition (7.12) is satisfied for time-reversible processes and Langevin dynamics.

Example 20 ***(Time-reversible Markov process)***
In this example, we will consider a time-reversible Markov process (X_t), such as standard random walk process on undirected networks. If $p_1 = p_2 = p$ and $\mu_1 = \mu_2 = \mu$, then the extended detailed balance condition (7.12) is identical to the standard detailed balance condition (2.12). In this case Theorem 10 is the typical statement about reversibility of the process and self-adjointness of the associated transfer operator.

Example 21 ***(Langevin dynamics)***
Let us consider non-reversible Langevin dynamics [27, 138] given with the following equation

$$M\dot{q} = p, \quad \dot{p} = -D_q V(q) - \gamma p + \sigma \dot{W},$$

where $\gamma > 0$ is the friction coefficient, V is a potential energy function and W is a Wiener process or Brownian motion. For the state space S, with states $x = (q,p)$, where $q, p \in \mathbb{R}^n$, the Hamiltonian is

$$H(x) = \frac{1}{2} p^T M p + V(q).$$

Under certain regularity conditions on U [27] the process is geometrically ergodic with respect to an unique invariant measure $\mu(x) = e^{-\beta H(x)}/Z$, where parameter $\beta > 0$ refers to the inverse temperature. For a regular linear transformation $A : S \to S$

$$A = \begin{pmatrix} Id & 0 \\ 0 & -Id \end{pmatrix},$$

with $A^{-1} = A$, we find that $H(Ax) = H(x), \forall x \in S$ and

$$\mu(x)p(x,y) = \mu(Ay)p(Ay,Ax),\ \forall x,y \in S. \tag{7.13}$$

If $p_1(x,y) = p(x,y)$, $p_2(x,y) = p(Ax,Ay)$, $\mu_1(x) = \mu(x)$ and $\mu_2(x) = \mu(Ax)$, then we see that the extended detailed balance condition (7.12) is satisfied.

In the following, we will suppose that the extended detailed balance condition holds. Then, it is easy to check that

$$\langle T_{12}f_1, g_2\rangle_2 = \langle f_1, T_{21}g_2\rangle_1 \tag{7.14}$$

and

$$\langle T_{21}f_2, g_1\rangle_1 = \langle f_2, T_{12}g_1\rangle_2, \tag{7.15}$$

that is T_{12} and T_{21} are **adjoint operators** in the sense of operators acting between different Hilbert spaces H_1 and H_2 [159].

Let us now consider $\mathcal{T} = T_{21}T_{12} : H_1 \to H_1$, where

$$\begin{aligned}(T_{21}T_{12}f)(y)\mu_1(y) &= \int\int p_1(x,z)p_2(z,y)f(x)\mu_1(x)\,dx\,dz \\ &= \int p(x,y)f(x)\mu_1(x)dx\end{aligned}$$

with

$$p(x,y) = \int p_1(x,z)p_2(z,y)dz,$$

such that $\mathcal{T}$ has the kernel $p(x,y)$. Then,

Theorem 11
The operator $\mathcal{T} = T_{21}T_{12} : H_1 \to H_1$ is self-adjoint on H_1.

Proof. For $f, g \in L^2$ with $f_1 = J_1 f \in H_1$ and $g_1 = J_1 g \in H_1$, we have that $T_{12}f_1 \in H_2$ and $T_{12}g_1 \in H_2$. Using 7.14 and 7.15, we can prove that

$$\langle g_1, T_{21}T_{12}f_1\rangle_1 = \langle T_{12}g_1, T_{12}f_1\rangle_2 = \langle T_{21}T_{12}g_1, f_1\rangle_1, \tag{7.16}$$

that is $\mathcal{T} = T_{21}T_{12}$ is a self-adjoint operator. □

In analogy $\mathcal{B} = T_{12}T_{21} : H_2 \to H_2$ is a self-adjoint operator on H_2.

7.1.3 Singular value decomposition of transfer operators

In this section we will restrict our considerations to discrete state spaces for simplicity of presentation; all statements made in the following can be generalized to continuous state spaces as well. The self-adjoint operator $\mathcal{T} = T_{21}T_{12}$ on H_1 can be written [94] as

$$\mathcal{T} = T_{21}T_{12} = \sum_k \lambda_k^2 \langle \cdot, \varphi_k \rangle_1 \varphi_k,$$

where $\{\varphi_k\}$ are orthonormal eigenvectors of $\mathcal{T}$, i.e. $\langle \varphi_k, \varphi_l \rangle_1 = \delta_{kl}$, corresponding to eigenvalues λ_k^2

$$\mathcal{T}\varphi_k = T_{21}T_{12}\varphi_k = \lambda_k^2 \varphi_k.$$

Let us now introduce $\psi_k = \lambda_k^{-1} T_{12}\varphi_k$, such that

$$\begin{aligned} \langle \psi_k, \psi_l \rangle_2 &= \lambda_k^{-1}\lambda_l^{-1} \langle T_{12}\varphi_k, T_{12}\varphi_l \rangle_2 \\ &= \lambda_k^{-1}\lambda_l^{-1} \langle \varphi_k, T_{21}T_{12}\varphi_l \rangle_1 \\ &= \lambda_k^{-1}\lambda_l \langle \varphi_k, \varphi_l \rangle_1 = \delta_{kl}, \end{aligned}$$

i.e. vectors $\{\psi_k\}$ are orthonormal wrt. $\langle , \rangle_2$. Furthermore, vectors $\{\psi_k\}$ are eigenvectors of $\mathcal{B} = T_{12}T_{21}$ that correspond to eigenvalues λ_k^2

$$\begin{aligned} T_{12}T_{21}\psi_k &= \lambda_k^{-1} T_{12}T_{21}T_{12}\varphi_k = \lambda_k^{-1}\lambda_k^2 T_{12}\varphi_k \\ &= \lambda_k T_{12}\varphi_k = \lambda_k \lambda_k \psi_k = \lambda_k^2 \psi_k. \end{aligned}$$

Now, we can write the so called **dispersion relations** for transfer operators T_{12} and T_{21}

$$\begin{aligned} T_{12}\varphi_k &= \lambda_k \psi_k \\ T_{21}\psi_k &= \lambda_k \varphi_k, \end{aligned} \tag{7.17}$$

where as pointed out above, $\{\psi_k\}$ and $\{\varphi_k\}$ solve the following **eigenvalue problem** for self-adjoint operators $\mathcal{T}$ and $\mathcal{B}$

$$\begin{aligned} T_{21}T_{12}\varphi_k &= \lambda_k^2 \varphi_k \\ T_{12}T_{21}\psi_k &= \lambda_k^2 \psi_k. \end{aligned} \tag{7.18}$$

The **singular value decomposition** of the two transfer operators is given with [13, 94, 162]

$$T_{12} = \sum_k \lambda_k \langle \cdot, \varphi_k \rangle_1 \psi_k$$

and

$$T_{21} = \sum_k \lambda_k \langle \cdot, \psi_k \rangle_2 \varphi_k.$$

Let us introduce the following vectors $\{\hat{\varphi}_k\}$ and $\{\hat{\psi}_k\}$ from L^2, such that

$$\hat{\varphi}_k = J_1^{-1}\varphi_k \quad \text{and} \quad \hat{\psi}_k = J_2^{-1}\psi_k. \tag{7.19}$$

Now using (7.11) and (7.17)

$$T_{12}\varphi_k = J_2 P_{12} J_1^{-1} J_1 \hat{\varphi}_k = J_2 P_{12} \hat{\varphi}_k = \lambda_k J_2 \hat{\psi}_k, \tag{7.20}$$

that is $P_{12}\hat{\varphi}_k = \lambda_k \hat{\psi}_k$. Similarly, it holds that

$$T_{21}\psi_k = J_1 P_{21} J_2^{-1} J_2 \hat{\psi}_k = J_1 P_{21} \hat{\psi}_k = \lambda_k J_1 \hat{\varphi}_k.$$

Therefore, $\{\hat{\varphi}_k\}$ and $\{\hat{\psi}_k\}$ satisfy the dispersion relations for P_{12} and P_{21}

$$\begin{aligned} P_{12}\hat{\varphi}_k &= \lambda_k \hat{\psi}_k \\ P_{21}\hat{\psi}_k &= \lambda_k \hat{\varphi}_k, \end{aligned} \tag{7.21}$$

where $\{\hat{\psi}_k\}$ and $\{\hat{\varphi}_k\}$ solve the following **eigenvalue problem**

$$\begin{aligned} P_{21}P_{12}\hat{\varphi}_k &= \lambda_k^2 \hat{\varphi}_k \\ P_{12}P_{21}\hat{\psi}_k &= \lambda_k^2 \hat{\psi}_k. \end{aligned}$$

Furthermore, $\{\hat{\varphi}_k\}$ and $\{\hat{\psi}_k\}$ are sets of orthonormal vectors in L^2, since

$$\begin{aligned} \delta_{kl} &= \langle \varphi_k, \varphi_l \rangle_1 = (J_1^{-1}\varphi_k, J_1^{-1}\varphi_l) = (\hat{\varphi}_k, \hat{\varphi}_l) \\ \delta_{kl} &= \langle \psi_k, \psi_l \rangle_2 = (J_2^{-1}\psi_k, J_2^{-1}\psi_l) = (\hat{\psi}_k, \hat{\psi}_l). \end{aligned}$$

For arbitrary $f, g \in L^2$ with $f_i = J_i f \in H_i$ and $g_i = J_i g \in H_i$, we can provide a biorthogonal decomposition of the operator P_{12} in L^2

$$\begin{aligned} P_{12}f &= J_2^{-1} T_{12} J_1 f = \sum_k \lambda_k \langle J_1 f, \varphi_k \rangle_1 J_2^{-1}\psi_k \\ &= \sum_k \lambda_k (f, J_1^{-1}\varphi_k)\hat{\psi}_k = \sum_k \lambda_k (f, \hat{\varphi}_k)\hat{\psi}_k, \end{aligned}$$

and analogously $P_{21}g = \sum_k \lambda_k (g, \hat{\psi}_k)\hat{\varphi}_k$.
Using the spectral properties of P_{12} and P_{21}, we can write the matrix representation of these operators in the following way. From (7.21) and $P_{12} = P_{21}^T$, it follows that $P_{21}^T \hat{\varphi}_k = \lambda_k \hat{\psi}_k, k = 1, \ldots, n$, such that

$$\hat{\psi}_k^T P_{21}^T \hat{\varphi}_k = \lambda_k.$$

Thus, for $\hat{\varphi} = [\hat{\varphi}_1 \ldots \hat{\varphi}_n]$ and $\hat{\psi} = [\hat{\psi}_1 \ldots \hat{\psi}_n]$, it holds that

$$\hat{\psi}^T P_{12} \hat{\varphi} = D, \quad D = \mathrm{diag}(\lambda_i),$$

where $\hat{\varphi}$ and $\hat{\psi}$ are invertible matrices, due to the orthogonality of their columns. Using that $(\hat{\psi}_k, \hat{\psi}_l) = \delta_{kl} = (\hat{\varphi}_k, \hat{\varphi}_l)$ it follows that the matrix representation of P_{12} is

$$P_{12} = (\hat{\psi}^T)^{-1} D \hat{\varphi}^{-1}. \tag{7.22}$$

Similarly, we can calculate the matrix representation of P_{21}.

7.1.4 Coarse graining of non-reversible processes

Let us now focus on one of the two the non-reversible process we have discussed above, for example (X_t) and assume that this process has an unique invariant measure μ. Then, we introduce the transfer operator $T : L^2_\mu \to L^2_\mu$, with

$$(Tf)(y)\mu(y) = \sum_{x\in S} p(x,y)f(x)\mu(x),$$

For a linear transformation

$$J : L^2 \to L^2_\mu, \quad (J\tilde{f})(x) = \frac{1}{\sqrt{\mu(x)}}\tilde{f}(x) = f(x),$$

arbitrary $\tilde{f}, \tilde{g} \in L^2$ and $f = J\tilde{f}, g = J\tilde{g} \in L^2_\mu$, it follows that

$$\begin{aligned}\langle f, Tg\rangle_\mu &= \sum_{y\in S} f(y)(Tg)(y)\mu(y) = \sum_{x,y\in S} \frac{1}{\sqrt{\mu(y)}}\tilde{f}(y)p(x,y)\tilde{g}(x)\sqrt{\mu(x)} \\ &= (\tilde{f}, P\tilde{g}),\end{aligned}$$

where P is the representation of T in L^2

$$(Pf)(y)\sqrt{\mu(y)} = \sum_{x\in S} p(x,y)f(x)\sqrt{\mu(x)},$$

which can be written in the typical kernel form

$$(Pf)(y) = \sum_{x\in S} \pi(x,y)f(x), \quad \pi(x,y) = \frac{1}{\sqrt{\mu(y)}}p(x,y)\sqrt{\mu(x)} \tag{7.23}$$

Furthermore, from

$$\begin{aligned}\langle f, Tg\rangle_\mu &= (J^{-1}f, J^{-1}Tg) \\ &= (\tilde{f}, J^{-1}TJ\tilde{g}),\end{aligned} \tag{7.24}$$

we see that

$$P = J^{-1}TJ. \tag{7.25}$$

We will now follow the approach from Section 4.2 based on MSM and use it in order to find dominant metastable sets of the observed process (X_t). Let us assume that the state space S is decomposed into m disjoint sets $C_1, \ldots, C_m \subset S$

$$\cup_{i=1}^m C_i \neq S \quad \Rightarrow \quad S \setminus \cup_{i=1}^m C_i \neq \emptyset,$$

Then, for the process (X_t) with the transfer operator T, we can define the milestoning process $(\hat{X}_t)$ by equation (4.5) [63]

$$\hat{X}_t = i \Leftrightarrow X_{\sigma(t)} \in C_i, \text{with } \sigma(t) = \sup_{s\leq t}\left\{X_s \in \bigcup_{k=1}^m C_k\right\}.$$

Let q_i^+ and q_i^-, $i = 1, \ldots, m$ denote forward and backward committors related to T (see Section 3.1.2), defined as in equations (3.4) and (3.5).

For backward committors q_i^-, let us introduce $\tilde{q}_i^- \in L^2$ with $\tilde{q}_i^- = J^{-1}q_i^-$, such that for $x \notin \cup_{i=1}^m C_i$

$$(Tq_i^-)(x) = (JPJ^{-1}q_i^-)(x) = (JP\tilde{q}_i^-)(x) = q_i^-(x) = (J\tilde{q}_i^-)(x),$$

where we used that $(Tq_i^-)(x) = q_i^-(x)$, for all $x \notin \cup_{j=1}^m C_j$ (see (3.5)). Therefore, functions $\tilde{q}_i^-$ are representation of q_i^- in L^2, for $i = 1, \ldots, m$

$$(P\tilde{q}_i^-)(x) = \tilde{q}_i^-(x), \quad \forall x \notin \cup_{i=1}^m C_i.$$

Similarly, we define the set of $\tilde{q}_i^+ \in L^2$ with $\tilde{q}_i^+ = J^{-1}q_i^+$ that are representation of q_i^+ in L^2.

The transition behavior of the milestoning process can be expressed using the results from Section 4.2.2, that hold also for non-reversible processes. To this end, let us restrict our attention to time-discrete Markov processes for simplicity. However, all statements made in the following can be generalized to time-continuous Markov processes. Then, the equivalent of Theorem 5 in L^2 space is the following

Theorem 12
For a time-discrete process (X_n), the entries of the discrete generator $\hat{L}_d$ of the milestoning process $(\hat{X}_n)$ are given by

$$\hat{l}_d(i,j) = \frac{1}{\hat{\mu}(i)}(\tilde{q}_j^+, L_d\tilde{q}_i^-),$$

where $\hat{\mu}(i) = \sum_{x\in V} \tilde{q}_i^-(x)\sqrt{\mu(x)}$, $L_d = P - Id$ and P is the representation of T in L^2.

Proof. The proof follows directly from the Theorem 5. The entries $\hat{l}_d(i,j)$ from equation (4.6) are

$$\hat{l}_d(i,j) = \frac{1}{\hat{\mu}(i)}\langle q_j^+, \mathcal{L}_d q_i^-\rangle_\mu,$$

where using (7.24) it follows that

$$\begin{aligned}\langle q_j^+, \mathcal{L}_d q_i^-\rangle_\mu &= \langle q_j^+, (T - Id)q_i^-\rangle_\mu \\ &= (\tilde{q}_j^+, P\tilde{q}_i^-) - (\tilde{q}_j^+, \tilde{q}_i^-) = (\tilde{q}_j^+, (P - Id)\tilde{q}_i^-).\end{aligned}$$

The invariant measure of the milestoning process $\hat{\mu}$ is

$$\hat{\mu}(i) = \mathbb{P}_\mu(\hat{X}_n = i) = \sum_{x\in S} \mathbb{P}_\mu(\hat{X}_n = i, X_n = x) = \sum_{x\in S} q_i^-(x)\mu(x) = \sum_{x\in S} \tilde{q}_i^-(x)\sqrt{\mu(x)}.$$

□

Let us introduce the following subspaces $D_1, D_2 \subset L^2$, where $D_1 = span\{\tilde{q}_1^+, \ldots, \tilde{q}_m^+\}$ and $D_2 = span\{\tilde{q}_1^-, \ldots, \tilde{q}_m^-\}$ with $\mathbb{1} \in D_1, D_2$. Then the orthogonal projection Q_1 onto D_1 and the orthogonal projection Q_2 onto D_2 can be written as

$$Q_1 v = \sum_{i,j=1}^{m} S_{ij}^{-1}(v, \tilde{q}_j^+)\tilde{q}_i^+$$
$$Q_2 v = \sum_{i,j=1}^{m} R_{ij}^{-1}(v, \tilde{q}_j^-)\tilde{q}_i^-,$$

with $S_{ij} = (\tilde{q}_i^+, \tilde{q}_j^+)$ and $R_{ij} = (\tilde{q}_i^-, \tilde{q}_j^-)$. Now, we can generalize Theorem 7 for non-reversible process (X_t) in order to compare the operator P and its projection $Q_1 P Q_2$.

Theorem 13
Let P be the L^2 representation of the transfer operator T of the non-reversible process (X_t), Q_1 the orthogonal projection onto the space spanned by forward committors $D_1 = span\{\tilde{q}_1^+, ..., \tilde{q}_m^+\}$ and Q_2 the orthogonal projection onto the space spanned by backward committors $D_2 = span\{\tilde{q}_1^-, ..., \tilde{q}_m^-\}$ with respect to m disjoint sets $C_1, \ldots, C_m \subset S$. Then, $\hat{P}M^{-1}$ is a matrix representation of $Q_1 P Q_2$, where

$$\hat{P}_{ij} = \frac{(\tilde{q}_j^+, P\tilde{q}_i^-)}{\hat{\mu}(i)}, \quad M_{ij} = \frac{(\tilde{q}_i^+, \tilde{q}_j^+)}{\hat{\mu}(i)}. \tag{7.26}$$

Proof. For the matrix M from (7.26) we have

$$M_{ij} = \frac{1}{\hat{\mu}(i)} S_{ij} \Rightarrow M_{ij}^{-1} = \hat{\mu}(j) S_{ij}^{-1}. \tag{7.27}$$

Now, take the basis $\{\psi_1, ..., \psi_m\}$ of D_1, $\psi_i = \frac{1}{\hat{\mu}(i)}\tilde{q}_i^+$ and $\{\varphi_1, ..., \varphi_m\}$ of D_2, $\varphi_i = \frac{1}{\hat{\mu}(i)}\tilde{q}_i^-$. Then,

$$Q_1 v = \sum_{i,j=1}^{m} M_{ij}^{-1}(v, \tilde{q}_i^+)\psi_j. \tag{7.28}$$

This implies

$$\begin{aligned} Q_1 P Q_2 \varphi_k = Q_1 P \varphi_k &= \sum_{i,j=1}^{m} M_{ij}^{-1}(P\varphi_k, \tilde{q}_i^+)\psi_j \\ &= \sum_{i,j=1}^{m} M_{ij}^{-1} \frac{(P\tilde{q}_k^-, \tilde{q}_i^+)}{\hat{\mu}(k)}\psi_j = \sum_{i,j=1}^{m} M_{ij}^{-1}\hat{P}_{ki}\psi_j \\ &= \sum_{j=1}^{m} (\hat{P}M^{-1})_{kj}\psi_j. \end{aligned} \tag{7.29}$$

That is, $\hat{P}M^{-1}$ is a matrix representation of $Q_1 P Q_2$ with respect to the basis $\{\varphi_1, ..., \varphi_m\}$. □

7.1.5 Approximation quality of Markov state models for non-reversible processes

In this section we will consider Markov state models (MSM) for non-reversible processes, following the ideas from above. Especially, we will present some fundamental issues about generalizing one approach for determining the approximation quality of MSM for reversible processes presented in [143]. This approach deals with providing a quality measure of MSM, in terms of the error bound for the difference in propagation of probability densities between the original and the coarse grained process on long time scales [143].
For a given choice of sets $C_1, \dots, C_m$ Theorem 13 provided a representation of coarse grained process in terms of the projected transfer operator Q_1PQ_2. Following this idea, we conclude that if X_0 and $\hat{X}_0$ are initially equally distributed, then the maximal possible error between X_k and $\hat{X}_k$ distributions after k steps is given by

$$E(k) = \|Q_1P^kQ_2 - (Q_1PQ_2)^k\|. \tag{7.30}$$

Thus, a natural question to ask is: Under which assumptions is the error $E(k)$ small? In [143] this question was answered for reversible Markov processes. Here we show how this result could be generalized for non-reversible processes. However, providing strict theoretical statements for general non-reversible processes exceeds the scope of this thesis. This will be the topic of future interest.
Following the result of Theorem 13 that provides a matrix representation of the projected transfer operator Q_1PQ_2, we will present now a matrix representation of $P^k, k \geq 1$ that is needed for estimating the error $E(k)$.

Proposition 12
For $J_2^{-1}J_1 \approx Id$, $A = J^{-1}J_1$ and $B = J_2^{-1}J$ it holds that

$$P^k \approx AP_{12}^kB, \quad k \geq 1.$$

Proof. From (7.10) and (7.23), using the matrix representation of J_1, J_2 and J, we can write the matrix form of the transfer operator P in the following way

$$P = J^{-1}J_1P_{12}J_2^{-1}J, \tag{7.31}$$

where the matrix representation of P_{12} is given in (7.22). From $J_2^{-1}J_1 \approx Id$, it easily follows that $P^k \approx J^{-1}J_1P_{12}^kJ_2^{-1}J, k \geq 1$. □

From Proposition 12 we can deduce the form of $E(k)$. The assumption $J_2^{-1}J_1 \approx Id$ is connected to the two measures μ_1 and μ_2 in the sense that $\mu_1(x) \approx \mu_2(x), \forall x \in V$. In particular, when dealing with reversible processes (see Example 20), such as random walk processes on undirected networks, we have that $\mu_1(x) = \mu_2(x), \forall x \in V$ and therefore $J_2^{-1}J_1 = Id$. Similarly, we can see that $J_2^{-1}J_1 = Id$ holds also in the case of Langevin dynamics (see Example 21). However, it is important to notice that this assumption doesn't hold in general.

Remark 8 *We remark that in the above computation it is assumed that* $\mu_i(x) \neq 0, \forall x \in V$*, since the entries of* J_i^{-1}*,* $i = 1, 2$ *are* $\frac{1}{\sqrt{\mu_i(x)}}, \forall x \in S$*. In terms of the observed process this means that we assume that none of the states is neither absorbing nor transient. We will see later that in the case of directed networks this assumption means that we do not consider networks that have sinks and sources.*

Remark 9 *In [143] the upper bound for the error* $E(k)$ *is provided. This bound resulted from certain spectral properties of ergodic, reversible processes. These properties hold also for some non-reversible processes that have special properties [86], for example, for processes that are sufficiently ergodic [86, 150] and have a dominant part of* T *that is nearly self-adjoint. This is the case for second-order Langevin dynamics with not too large friction [83] or for thermostated Hamiltonian molecular dynamics or stochastically perturbed Hamiltonian systems [150, 48]. However, in general little is known about which processes satisfy the above mentioned conditions. Numerical results for some processes, such as some cases of Langevin dynamics, imply the existence of wanted spectral properties of* T *[86], but the theoretical justification of these results is the topic of future research.*

7.2 Random walks on directed networks

We will apply now the strategy introduced in the previous section in order to analyze directed networks using random walk processes. For the simplicity, we will focus on unweighted, directed networks, but we can generalize this approach in a fairly straightforward way to the case of random walks on weighted, directed networks. Furthermore, here we will follow the ideas from standard random-walk-based approach (Section 2.1), that can be easily extended to the case of time-continuous random walk processes (see Section 2.3).

Let us define the **time-forward random walk** [42] as a time-discrete process, where at each time step the random walker jumps from a node x to a node y that has been chosen uniformly at random from all the nodes, for which there is a directed edge from x to y, i.e. $(x, y) \in E$. A sequence of visited nodes form a Markov chain, with transition probabilities

$$p^+(x, y) = \begin{cases} \frac{1}{d_{out}(x)}, & (x, y) \in E \\ 0, & (x, y) \notin E \end{cases} \tag{7.32}$$

where $d_{out}(x)$ is the out-degree of the node x, as defined in (1.1). The transition matrix $P^+ = (p^+(x, y))_{x,y \in V}$ is a well defined stochastic matrix only if there are no sinks in the network, i.e. $d_{out}(x) \geq 1, \forall x \in V$.

In the similar way, we can define the **backward-time random walk**, that describes random walk going backward in time and it's associated Markov chain is given by

the transition matrix

$$p^-(x,y) = \begin{cases} \frac{1}{d_{in}(x)}, & (y,x) \in E \\ 0, & (y,x) \notin E \end{cases} \tag{7.33}$$

where $d_{in}(x)$ is the in-degree of the node x, as in (1.2). Again, the transition matrix $P^- = (p^-(x,y))_{x,y \in V}$ is a well defined stochastic matrix if there are no sources in the network, i.e. $d_{in}(x) \geq 1, \forall x \in V$.
Note that for undirected networks, (7.32) and (7.33) correspond to the forward and backward random walks, where due to $d_{out}(x) = d_{in}(x)$ and $w(x,y) = w(y,x)$ these processes are identical. However, some properties of random walk processes on directed networks are similar to the ones mentioned in Section 2.1.1. As introduced in Definition 8, network is strongly connected if for every pair of nodes (x,y) there exist a directed path from x to y and a directed path from y to x.

Proposition 13
Markov chains associated to a random walks (7.32) and (7.33) are irreducible, if and only if, the underlying directed network is strongly connected.

Another important property is given by the following property [108, 42, 24, 25]

Proposition 14
If G is a strongly connected and aperiodic network, the random walk converges to a unique stationary distribution.

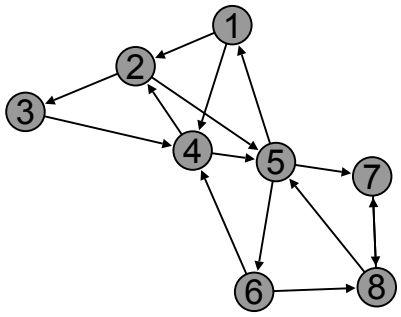
Spectral properties are described in the following extended version of Perron-Frobenius theorem:

Theorem 14 (Perron-Frobenius theorem for irreducible Markov chains) *If $(X_t)_{t \in \mathbb{N}}$ is an irreducible Markov chain with period $k > 1$ and $n \times n$ the transition matrix P, then*

1. *P has an eigenvalue $\lambda_1 = 1$, with the corresponding right eigenvector $P\mathbb{1} = \mathbb{1}$, $\mathbb{1} = (1, \ldots, 1)$ and the left eigenvector $\pi P = \pi$, that has all positive entries $\pi > 0$.*
2. *For all other eigenvalues of P it holds $1 = \lambda_1 \geq |\lambda_2| \geq |\lambda_3| \geq \ldots \geq |\lambda_n|$.*
3. *There are exactly k eigenvalues of modulus 1. These eigenvalues are of the form $e^{\frac{2i\pi}{k}}$ and they are invariant under the rotation by $\frac{2\pi}{k}$ around the origin.*

The Theorem 14 reveals that structural properties of directed networks are reflected in the spectrum of the transition matrix of the associated Markov chain. For example, the existence of cycles in the network results in appearance of complex conjugated pairs of eigenvalues of P^+ and P^-, with a module equal to one as it will be shown in the following example.

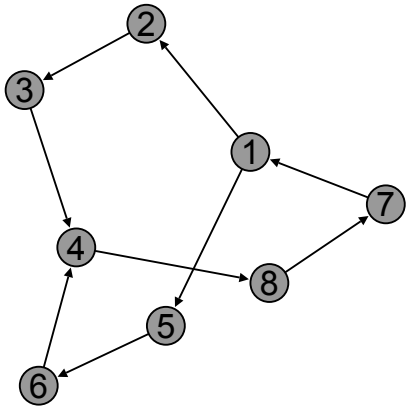
Example 22 *First, let us define the forward and backward random walk processes on a directed, unweighted network shown in Figure 7.1 and observe the spectral properties of the associated Markov chains. Since this network is strongly connected, it follows from Proposition 13 that the associated Markov chains are irreducible and we can apply the Perron-Frobenius theorem. The spectrum of both transition matrices are shown in Figure 7.1.*
The second example considers an unweighted, bipartite directed network shown in



	Eigenvalues of P^+	Eigenvalues of P^-
λ_1	1	1
λ_2	$-0.3923 + 0.6782i$	$-0.4153 + 0.7303i$
λ_3	$-0.3923 - 0.6782i$	$-0.4153 - 0.7303i$
λ_4	0.6721	0.5164
λ_5	-0.4323	$-0.2133 + 0.3186i$
λ_6	$-0.2276 + 0.2550i$	$-0.2133 - 0.3186i$
λ_7	$-0.2276 - 0.2550i$	-0.2592
λ_8	0	0

Figure 7.1: A Figure shows an example of an unweighted-directed network with 8 nodes and eigenvalues of transition matrices P^+ and P^-.

Figure 7.2. The period of this network is $k = 6$. From Theorem 14 we know that the spectrum of transition matrices consists of exactly 6 eigenvalues with module 1. Since the graph is bipartite, the eigenvalues of the two transition matrices P^+ and P^- are the same, they are symmetric with respect to 0 and they have an eigenvalue $\lambda = -1$.



Eigenvalues of P^+, P^-	
λ_1	1
λ_2	$0.5000 + 0.8660i$
λ_3	$0.5000 - 0.8660i$
λ_4	0
λ_5	0
λ_6	$-0.5000 + 0.8660i$
λ_7	$-0.5000 - 0.8660i$
λ_8	-1

Figure 7.2: A Figure shows an example of an unweighted, bipartite, periodic directed network that consists of 8 nodes. A table shows the eigenvalues of transition matrix P.

Since the adjacency matrices of directed networks are not symmetric, we loose many nice properties from before, such as time-reversibility that would make the time-forward random walk process and the backward-time random walk process to be equal. For these reasons, we continue by utilizing the objects introduced in Section 7.1, where p^+ and p^- correspond to p_1 and p_2 respectively. Let us define the following probability measures

$$\mu^+(x) = \frac{d_{out}(x)}{\sum_{y\in V} d_{out}(y)}, \quad \mu^-(x) = \frac{d_{in}(x)}{\sum_{y\in V} d_{in}(y)}, \tag{7.34}$$

as two measures that correspond to invariant measures defined in (7.2). As shown in Example 20, for random walk processes on undirected networks we have that $\mu^+ = \mu^- = \mu$, where μ is defined as in (2.10). Using that $\text{vol}(G) := \sum_{y\in V} d_{out}(y) = \sum_{y\in V} d_{in}(y)$, it is easy to see that

$$\mu^+ P^+ = \mu^- \quad \text{and} \quad \mu^- P^- = \mu^+,$$

i.e. μ^+ and μ^- are not the invariant measures wrt. P^+ nor P^-.

We can prove that the **extended detailed balance condition** (7.12) is satisfied

$$\begin{aligned}\mu^+(x)p^+(x,y) &= \frac{d_{out}(x)}{vol(G)}\frac{1}{d_{out}(x)} \\ &= \frac{d_{in}(x)}{vol(G)}\frac{1}{d_{in}(x)} = \mu^-(y)p^-(y,x), \quad \forall x,y \in V.\end{aligned}$$

Then, like in the previous section, we can define two transfer operators, namely T_{12}

$$(T_{12}f)(y)\mu^-(y) = \sum_{x\in V} \mu^+(x)f(x)p^+(x,y) \tag{7.35}$$

and T_{21}

$$(T_{21}f)(y)\mu^+(y) = \sum_{x\in V} \mu^-(x)f(x)p^-(x,y). \tag{7.36}$$

Furthermore, we can show that these two operators are adjoint in the sense of (7.14) and (7.15). More precisely,

$$\begin{aligned}\langle T_{12}f, g\rangle_{\mu^-} &= \sum_{y\in V} \mu^-(y)g(y)(T_{12}f)(y) \\ &= \sum_{x,y\in V} g(y)\mu^+(x)p^+(x,y)f(x),\end{aligned}$$

and

$$\begin{aligned}\langle f, T_{21}g\rangle_{\mu^+} &= \sum_{y\in V} \mu^+(y)f(y)(T_{21}g)(y) \\ &= \sum_{x,y\in V} f(y)\mu^-(x)p^-(x,y)g(x),\end{aligned}$$

so that

$$\langle T_{12}f, g\rangle_{\mu^-} = \langle f, T_{21}g\rangle_{\mu^+}, \tag{7.37}$$

where we used $\mu^+(x)p^+(x,y) = \mu^-(y)p^-(y,x)$.

Again, the operator $\mathcal{T} = T_{21}T_{12}$ is self-adjoint, where the kernel p of $\mathcal{T}$ has the form

$$p(x,y) = \sum_z \frac{1}{d_{out}(z)} w(z,x) \frac{1}{d_{in}(x)} w(z,y) = \frac{1}{d_{in}(x)} \sum_z \frac{w(z,x)w(z,y)}{d_{out}(z)}.$$

Analogously, $\mathcal{B} = T_{12}T_{21}$ is a self-adjoint operator. Following our general approach from above, we can define operators P_{12} and P_{21}, that are representations of operators T_{12} and T_{21} in L^2 and calculate their matrix representations using eigenvectors of self-adjoint operators $\mathcal{T}$ and $\mathcal{B}$ as in (7.22). In order to find metastable sets of two random walk processes, we can again define a milestoning process and use fuzzy MSM approach to identify dominant metastable regions of the original process (see Section 7.1.4 and Section 7.1.5). We will address this issue in the following section in more detail.

7.2.1 Identification of modules and hubs using random walks on directed networks

As we discussed above, analyzing directed networks is not an easy task. However, many real-world networks are directed, such as World Wide Web (WWW), see Section 1.3.3. In particular, modules and hubs in WWW could be of great importance for understanding the structure and organizational principles of this network. For example, finding web pages that belong to the same module could help identify their common topics, which could further help developing and improving search engine techniques. On the side, it has been shown that hub nodes correspond to the very important linking points of the WWW whose removal could cause seriously perturb or even cause breakdown of some parts of the network [43, 44].

Describing complex systems by directed networks provides more detailed and precise information about the underlying system. However, it also makes their analysis much more complex. Enriching edges with their directions causes asymmetry in the associated adjacency matrix, resulting in non-reversibility of the random walk process and non-self-adjointness of P. Then, the spectrum of P loses some of the nice properties that we had in the case of reversible processes (see Theorem 14), such as real-valued spectrum and uniqueness of the eigenvalue with module 1. This makes the spectral analysis approach much more complicated, as we have seen in Section 7.1. However, current experimental results show that the dominant part of the spectrum P is characterized by the real-valued eigenvalues and that their corresponding eigenvectors endorse the metastabilities of the random walk process. Theoretical justification of such results for certain classes of directed networks, would be a huge step forward in generalizing our method for module finding from Section 4.3.2 using the results presented in Section 7.1.5.

Applying our approach for finding hubs (Chapter 5) on directed networks is fairly straightforward, since the TPT objects that we are using and their properties hold also for non-reversible processes. However, it is important to note that our method

can encounter problems when identifying dominant reaction pathways when there are loops in the network. This can be seen in Proposition 9, where it is assumed that no loops (cycles) exist. Resolving this issue and providing an efficient algorithmic approach for finding the dominant pathways is the topic of future research.

7.2.2 Open questions and perspectives

Up to now, only a few methods for analyzing directed networks have been introduced. However, when considering these methods we can already see that generalizing definitions of modules and hubs to the case of directed networks is not easy and straightforward. Some of the fundamental questions concerning this topic are the following:

- **How should modules in directed networks be defined, such that they correspond to the functional subunits of the underlying system? How do edge directions influence the definition of modules?**
  Recently, these questions became a topic of scientific interest [104, 139, 97], which resulted in different definitions of modules. Some ideas are based on generalizing already existing definition from the case of undirected networks [104], whereas other are oriented towards developing information-theory-based approach [139]. In order to find a common definition of modules in directed networks, determining properties of optimal modules is certainly of great importance, especially for introducing new techniques for their identification.

- **What are hubs in directed networks? How does definition of "node importance" change when introducing directed edges?**
  The so-called link-based model for analyzing edge structure in the WWW [99], introduced two types of important nodes, namely authority and hub nodes. Authority nodes correspond to Web-pages to which many hubs point, whereas hub nodes correspond to Web-pages that point to many authority nodes. Similar ideas could be considered for developing a common definition of hub nodes.

Especially, in terms of the random-walk-based approach presented in this chapter, addressing the following questions is the topic of future research:

- **How can we resolve the problem of existing sources and sinks in the network?**
  Many real-world networks are characterized by the existence of sources and sinks. For example, in citation networks all papers that have not been cited yet represent sinks of this network. However, our definitions of forward and backward random walk processes given by (7.32) and (7.33), assume that the network doesn't have sinks nor sources. In order to overcome this problem, some methods [103, 109] introduce an additional random restart condition, that allows optional jumps of the random walk to some node of the network

(not necessarily a neighbor of a current node) with a certain restart probability. Certainly the most famous algorithm that is based on this idea is PageRank, used by the Google Internet search engine [103].

- **How can spectral properties of the forward and the backward random walk process be used in order to find network modules? How are metastable sets of two random walk processes related to modules in directed networks? Is there a difference between "forward" and "backward" modules and hubs?**

Future research should provide answers to these questions.

Summary

Real-world systems are often modeled as networks. Many algorithms for analyzing such complex networks are oriented towards finding modules that are densely inter-connected substructures having sparse connections to the rest of the network, and finding hub nodes that are key connectors of modules. In many cases these modules and hubs correspond to relevant structures in the original system. For example in biological systems, modules often correspond to functional units and hubs to essential parts of this system. In this thesis we developed a new mathematical framework that can be effectively applied for analyzing complex networks. This framework is based on defining a new type of random walk processes on networks and using spectral methods for finding modules and hubs.

When considering random walk processes on networks, modules represent metastable sets of this process. There are two crucial differences in the approach presented in this thesis compared to standard random-walk-based methods for module finding. Firstly, we have defined a new time-continuous random walk process characterized by waiting times in each node which results in increased metastability of the process in densely connected areas of the network-modules. In this way we have overcome the problem of most standard random walk processes for which also non-modular structures (for example long chains) represent metastable sets. The second difference results from the fact that most of the state-of-the-art approaches for module finding focus on finding a full partition of a network. The method introduced in this thesis finds a fuzzy decomposition of a network into modules, where nodes can be assigned to more than one module with a certain probability. In order to find such modules we used Markov State Models (MSM) as low-dimensional models for metastable Markov processes. We generalized the standard MSM approach that is based on full partitioning of the state space and developed a fuzzy MSM, where nodes that are assigned to some module with probability almost 1 correspond to dominant metastable regions. For determining the optimal modules, we used the approximation quality measure of the resulting MSM, based on the error between the original and reproduced dominant eigenvalues.

This thesis provides a new methodological approach for finding network hubs. We defined hubs as nodes that are important for the communication between network modules, that is determined by the associated random walk process. For measuring the amount of communication flow between modules in the network, we presented a method that is based on the framework of Transition Path Theory (TPT).

Finally, we proposed a generalization of our methods for analyzing undirected networks to the case of directed networks. The main difficulty is that random walk processes on directed networks are non-reversible. To this end, we adapted our methods to analyzing non-reversible processes by introducing two transfer operators whose spectral properties provide information that is needed for finding metastable sets of this process. In this way, we have provided a spectral approach for finding metastable sets of non-reversible processes. However, it is not yet clear how metastable sets are related to modules in directed networks. This will be the topic of future research.

Zusammenfassung

Technische oder natürliche Systeme werden oft als Netzwerke modelliert. Viele der vorhandenen Methoden zur Analyse solcher komplexer Netzwerke wurden dazu entwickelt, sogenannte Module und Hubs zu finden. Ein Modul ist eine Menge von Knoten, zwischen denen die Vernetzung untereinander stärker ist als zum Rest des Netzwerkes. Wichtige Knoten, die zur Verbindung von Modulen essentiell sind, werden als Hubs bezeichnet. Module und Hubs in einem Netzwerk entsprechen oft wichtigen Strukturen in dem durch dieses Netzwerk modellierten System. In biologischen Systemen entsprechen Module beispielsweise organisatorischen Einheiten und Hubs wichtigen Botenstoffen. In dieser Arbeit haben wir ein neues mathematisches Framework zur Analyse von komplexen Netzwerken entwickelt. Es basiert auf der spektralen Analyse neuartiger Random-Walk Prozesse auf Netzwerken.

Bei der Netzwerkanalyse mittels Random-Walk Prozessen entsprechen Module im Allgemeinen den metastabilen Mengen des Prozesses. Im Vergleich zu Standardmethoden zur Modulidentifikation unterscheidet sich unser neuer Ansatz durch zwei wichtige Merkmale: Erstens benutzen wir einen neuen zeitkontinuierlichen Random-Walk Prozess, der durch Wartezeiten in jedem Knoten charakterisiert ist. Dies führt zu einer erhöhten Metastabilität des Prozesses in den dicht vernetzten Bereichen der Netzwerkmodule und einer reduzierten Metastabilität in nicht-modularen Strukturen wie z.B. langen “Ketten”, die von den Standardmethoden wegen der hohen Metastabilität als Module erkannt werden. Der zweite grundlegende Unterschied unseres Ansatzes besteht darin, ein Netzwerk nicht vollständig in Module zu unterteilen und jeden Knoten zu genau einem Modul zuzuordnen (sog. full-partitioning), sondern einen Knoten einem oder mehreren Modulen mit einer bestimmten Wahrscheinlichkeit zuzuordnen (fuzzy- decomposition). Zur Identifikation dieser Module benutzen wir Markov-State-Models (MSM) als niedrig-dimensionale Repräsentation von metastabilen Markov Prozessen. Da das Standard-MSM Framework auf einer vollständigen Partition des Zustandsraumes basiert, haben wir dieses in der vorliegenden Arbeit verallgemeinert und eine fuzzy-MSM Variante entwickelt. In dieser Variante entsprechen Knoten, die einem Modul mit einer Wahrscheinlichkeit nahe Eins zugeordnet sind, den dominanten metastabilen Bereichen. Um die optimalen Module zu bestimmen, benutzen wir die Approximationsgüte des resultierenden MSM, welche auf dem Fehler zwischen den originalen und reproduzierten dominanten Eigenwerten basiert.

Neben dem Finden von Modulen wird in dieser Dissertation auch eine neue Methode zur Identifikation von sog. Hubs vorgestellt. Wir definieren Hubs als Knoten, die für die Kommunikation zwischen Modulen wichtig sind. Um den Kommunikationsfluss zwischen Modulen zu bestimmen, beschreiben wir eine neue Methode, die auf dem Transition Path Theory Framework basiert.

Im letzten Teil der Arbeit verallgemeinern wir die vorgestellten Konzepte, die bisher nur für ungerichtete Netzwerke entwickelt wurden, auf die Klassen der gerichteten Netzwerke. Das Hauptproblem hierbei ist, dass Random-Walk Prozesse auf gerichteten Netzwerken nicht reversibel sind. Um nicht-reversible Prozesse analysieren zu können, führen wir zwei Transferoperatoren ein, deren spektrale Eigenschaften die benötigten Informationen zur Identifikation von metastabilen Mengen liefern. Dadurch haben wir einen neuen, spektralen Ansatz zur Identifikation von metastabilen Mengen in nicht-reversiblen Prozessen entwickelt. Die Verbindung von diesen metastabilen Mengen zu Modulen in gerichteten Netzwerken ist noch nicht abschließend geklärt und wird Gegenstand weiterer Forschung sein.

Methods for analyzing complex networks using random walker approaches

Declaration

I declare that this thesis is the result of my own research and has not been accepted for another degree at any university or other institute of tertiary education. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

Nataša Djurdjevac (Đurđevac)

Curriculum Vitae

Der Lebenslauf ist in der Online-Version aus Gründen des Datenschutzes nicht enthalten.
For reasons of data protection, the Curriculum vitae is not published in the online version.

Bibliography

[1] S. Agarwal, C. M. Deane, M. A. Porter, and N. S. Jones. Revisiting date and party hubs: Novel approaches to role assignment in protein interaction networks. *PLoS Comput Biol*, 6:e1000817, 2010.

[2] S. Agarwal and S. Sengupta. Ranking genes by relevance to a disease. *Proceedings of the 8th Annual International Conference on Computational Systems Bioinformatics*, 2009.

[3] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network flows: theory, algorithms and applications*. Prentice Hall, 1993.

[4] A. H. Al-Mohy and N. J. Higham. Computing the action of the matrix exponential, with an application to exponential integrators. *MIMS EPrint 2010.30, The University of Manchester, to appear in SIAM J*, 2010.

[5] R. Albert. Scale-free networks in cell biology. *J. Cell Sci.*, 118:4947–4957., 2005.

[6] R. Albert and A.-L. Barabási. Statistical mechanics of complex networks. *Rev. Mod. Phys.*, 74:47–97, 2002.

[7] R. Albert, H. Jeong, and A.-L. Barabási. Diameter of the world-wide web. *Nature*, 401:130–131, 1999.

[8] R. Albert, H. Jeong, and A.-L. Barabási. Error and attack tolerance of complex networks. *Nature 406*, pages 378–382, 2000.

[9] P. Aloy and R. B. Russell. Potential artefacts in protein-interaction networks. *FEBS letters*, 530:253–254, 2002.

[10] R. Aragues, A. Sali, J. Bonet, M. A. Marti-Renom, and B. Oliva. Characterization of protein hubs by inferring interacting motifs from protein interactions. *PLoS Comput Biol*, 3:e178, 2007.

[11] A. Arenas, J. Duch, A. Fernandez, and S. Gomez. Size reduction of complex networks preserving modularity. *New Journal of Physics*, 9:176, 2007.

[12] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock. Gene ontology: tool for the unification of biology. the gene ontology consortium. *Nature Genet*, 25(1):25–29, 2000.

[13] N. Aubry, R. Guyonnet, and R. Lima. Spatiotemporal analysis of complex signals: Theory and applications. *Journal of Statistical Physics*, 64,3-4:683–739, 1991.

[14] A.-L. Barabási. *Linked: The new science of networks.* Cambridge (Massachusetts): Perseus Publishing, 2002.

[15] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286:509–512, 1999.

[16] A.-L. Barabási, R. Albert, and H. Jeong. Scale-free characteristics of random networks: The toplogy of the world wide web. *Physica A*, 281:69–77, 2000.

[17] A.-L. Barabási and E. Bonabeau. Scale-free networks. *Scientific American*, 288:60–69, 2003.

[18] A.-L. Barabási and Z. Oltvai. Network biology: Understanding the cell's functional organization. *Nature Reviews Genetics*, 5:101, 2004.

[19] A. Barrat, M. Barthélemy, R. Pastor-Satorras, and A. Vespignani. The architecture of complex weighted networks. *PNAS*, 101:3747–3752, 2004.

[20] A. Barrat, M. Barthélemy, and A. Vespignani. *Dynamical Processes on Complex Networks.* Cambridge University Press, 2008.

[21] N. N. Batada, T. Reguly, A. Breitkreutz, L. Boucher, B.-J. Breitkreutz, L. Hurst, and M. Tyers. Stratus not altocumulus: A new view of the yeast protein interaction network. *PLoS Biology*, 4:1720–31, 2006.

[22] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 6:1373–1396, 2003.

[23] M. Biskup, A. Bovier, F. Hollander, D. Ioffe, F. Martinelli, K. Netocný, and C. Toninelli. *Methods of contemporary mathematical statistical physics.* Springer, 2009.

[24] A. Björner and L. Lovász. Chip-firing games on directed graphs. *J. Algebr. Comb. 1, 305*, 1992.

[25] A. Björner, L. Lovász, and P. Shor. Chip-firing games on graphs. *Eur. J. Combin. 12, 283*, 1991.

[26] P. Blanchard and D. Volchenkov. *Random Walks and Diffusions on Graphs and Databases*. Springer Series in Synergetics, Vol. 10, 2011.

[27] N. Bou-Rabee and E. Vanden-Eijnden. Pathwise accuracy and ergodicity of metropolized integrators for sdes. *Communications on Pure and Applied Mathematics*, 63(5):655–696, 2010.

[28] A. Bovier, M. Eckhoff, V. Gayrard, and M. Klein. Metastability and low lying spectra in reversible markov chains. *Comm. Math. Phys.*, 228:219–255, 2002.

[29] A. Bovier, M. Eckhoff, V. Gayrard, and M. Klein. Metastability in reversible diffusion processes. I. sharp asymptotics for capacities and exit times. *J. Eur. Math. Soc. (JEMS)*, 6:399–424, 2004.

[30] A. Bovier, V. Gayrard, and M. Klein. Metastability in reversible diffusion processes. II. precise asymptotics for small eigenvalues. *J. Eur. Math. Soc. (JEMS)*, 7:69–99, 2005.

[31] U. Brandes, D. Delling, M. Höfer, M. Gaertler, R. Görke, Z. Nikoloski, and D. Wagner. On finding graph clusterings with maximum modularity. *In Proceedings of the 33rd International Workshop on Graph-Theoretic Concepts in Computer Science*, 2007.

[32] U. Brandes and T. Erlebach. *Network Analysis: Methodological Foundations*, volume 3418 of *Lecture Notes in Computer Science*. Springer, 2005.

[33] U. Brandes, M. Gaertler, and D. Wagner. Experiments on graph clustering algorithms. In *Algorithms - ESA 2003: 11th Annual European Symposium, Budapest, Hungary, September 16-19, 2003, Proceedings*, pages 568–579. Springer-Verlag, 2003.

[34] P. Bremaud. *Markov chains, Gibbs fields, Monte Carlo simulation, and queues*. Springer Verlag New York, 2001.

[35] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, R. Stata, A. Tomkins, and J. Wiener. Graph structure in the web. *Computer Networks*, 33:309–320, 2000.

[36] S. Brohee and J. van Helden. Evaluation of clustering algorithms for protein-protein interaction networks. *BMC Bioinformatics*, 7:488, 2006.

[37] A. E. Brouwer and W. H. Haemers. *Spectra of Graphs*. Springer, 2012.

[38] B. L. Chamberlain. Graph partitioning algorithms for distributing workloads of parallel computations. *University of Washington Technical Report UW-CSE-98-10-03*, 1998.

[39] J. Chang. Stochastic processes. *Online available material for the course "Stochastic Processes", http://www.soe.ucsc.edu/classes/engr203/Spring99/*, 1999.

[40] G. E. Cho and C. D. Meyer. Aggregation/disaggregation methods for nearly uncoupled Markov chains. *Technical Report NCSU no. 041600-0400, North Carolina State University*, 1999.

[41] F. Chung. Spectral graph theory. *CBMS Regional Conference Series in Mathematics, American Mathematical Society*, 1997.

[42] F. Chung. Laplacians and the cheeger inequality for directed graphs. *Ann. Comb. 9, 1*, 2005.

[43] R. Cohen, K. Erez, D. ben Avraham, and S. Havlin. Resilience of the internet to random breakdowns. *Physical Review Letters*, 85,21:pp.4626–4628, 2000.

[44] R. Cohen, K. Erez, D. ben Avraham, and S. Havlin. Breakdown of the internet under intentional attack. *Phys. Rev. Lett.*, 86,16:3682–3685, 2001.

[45] W. W. Cohen, R. E. Schapire, and Y. Singer. Learning to order things. *Journal of Artificial Intelligence Research*, 10:243–270, 1999.

[46] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to algorithms*. MIT Press, 1990.

[47] K. Crammer and Y. Singer. Online ranking by projecting. *Neural Compuation*, 17(1):145–175, 2005.

[48] P. Deuflhard, M. Dellnitz, O. Junge, and C. Schütte. Computation of essential molecular dynamics by subdivision techniques. In *Lecture Notes in Computational Science and Engineering*, volume 4, pages 98–115. Springer, 1999.

[49] P. Deuflhard, W. Huisinga, A. Fischer, and C. Schütte. Identification of almost invariant aggregates in reversible nearly uncoupled Markov chains. *Linear Algebra and its Applications*, 315:39–59, 2000.

[50] P. Deuflhard and M. Weber. Robust perron cluster analysis in conformation dynamics. *Linear Algebra and its Applications*, 398 Special issue on matrices and mathematical biology:161–184, 2005.

[51] N. Djurdjevac, S. Bruckner, T. O. F. Conrad, and C. Schütte. Random walks on complex modular networks. *Journal of Numerical Analysis, Industrial and Applied Mathematics (In press)*, 2011.

[52] N. Djurdjevac, M. Sarich, and C. Schütte. On Markov state models for metastable processes. *Proceeding of the ICM 2010*, 4:3105–3131, 2010.

[53] N. Djurdjevac, M. Sarich, and C. Schütte. Estimating the eigenvalue error of markov state models. *Multiscale Modeling & Simulation*, 10:61–81, 2012.

[54] S. N. Dorogovtsev and J. F. F. Mendes. *Evolution of networks: from biological nets to the Internet and WWW*. Oxford University Press, 2003.

[55] P. G. Doyle and J. L. Snell. Random walks and electric networks. *Carus Mathematical Monographs 22, Mathematical Association of America, Washington*, 1984.

[56] W. E, T. Li, and E. Vanden-Eijnden. Optimal partition and effective dynamics of complex networks. *Proc. Nat. Acad. Sci.*, 105:7907–7912, 2008.

[57] W. E and E. Vanden-Eijnden. Towards a theory of transition paths. *Journal of statistical physics*, 123:503–523, 2006.

[58] W. E and E. Vanden-Eijnden. Transition-path theory and path-finding algorithms for the study of rare events. *Annual Review of Physical Chemistry*, 61:391–420, 2010.

[59] U. Elsner. Graph partitioning: A survey. Technical report, Chemnitz, Germany: Technische Universität Chemnitz, 1997.

[60] P. Erdös and A. Rènyi. On radnom graphs. *Publicationes Mathematicae (Debrecen)*, 6:290–297, 1959.

[61] E. Estrada and J. A. Rodríguez-Velázquez. Subgraph centrality in complex networks. *Phys. Rev. E*, 71(5):056103, 2005.

[62] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the internet topology. *Comp. Comm. Rev.*, 29:251, 1999.

[63] A. K. Faradjian and R. Elber. Computing time scales from reaction coordinates by milestoning. *J. Chem. Phys.*, 120:10880–10889, 2004.

[64] D. Fell and A. Wagner. The small world of metabolism. *Nat. Biotech*, 189:1121–1122, 2000.

[65] S. Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):75–174, 2010.

[66] S. Fortunato and M. Barthélemy. Resolution limit in community detection. *PNAS*, 104,1:36–41, 2007.

[67] M. Freidlin and A. D. Wentzell. *Random perturbations of dynamical systems.* Springer, New York, 1998.

[68] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4:933–969, 2003.

[69] P. Garrido and J. Marro, editors. *Exploring Complex Graphs by Random Walks*, volume 661. American Institute of Physics, 2002.

[70] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *PNAS*, 99:7821–7826, 2002.

[71] M. Granovetter. The strength of weak ties. *American Journal of Sociology*, 78,6:1360–1380, 1973.

[72] R. Guimera and L. Amaral. Functional cartography of complex metabolic networks. *Nature*, 433:895–900, 2005.

[73] R. Guimera , S. Mossa, A. Turtschi, and L. Amaral. The worldwide air transportation network: Anomalous centrality, community structure, and cities global roles. *Proceedings of the National Academy of Sciences*, 102(22):7794–7799, 2005.

[74] R. Guimera, M. Sales-Pardo, and L. Amaral. Modularity from fluctuations in random graphs and complex networks. *Phys. Rev. E*, 70:025101, 2004.

[75] A. Gursoy, O. Keskin, and R. Nussinov. Topological properties of protein interaction networks from a structural perspective. *Biochem Soc Trans.*, 36(6)::1398–403, 2008.

[76] L. Hagen and A. Kahng. New spectral methods for ratio cut partitioning and clustering. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on In Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, Vol. 11, 9:1074–1085, 1992.

[77] L. Hakes, J. W. Pinney, D. L. Robertson, and S. C. Lovell. Protein-protein interaction networks and biology - what's the connection? *Nature Biotechnology*, 26:69–72, 2008.

[78] J.-D. Han, N. Bertin, T. Hao, D. Goldberg, G. Berriz, L. Zhang, D. Dupuy, A. Walhout, M. Cusick, F. Roth, and M. Vidal. Evidence for dynamically organized modularity in the yeast protein-protein interaction network. *Nature*, 430(6995):88–93, 2004.

[79] T. Hardiman, K. Lemuth, M. A. Keller, M. Reuss, and M. Siemann-Herzberg. Topology of the global regulatory network of carbon limitation in escherichia coli. *Journal of Biotechnology*, 132:359–374, 2007.

[80] L. Hartwell, J. Hopfield, S. Leibler, and A. Murray. From molecular to modular cell biology. *Nature*, 402:C47–C52, 1999.

[81] X. He and J. Zhang. Why do hubs tend to be essential in protein networks? *PloS Genet*, 2:e88, 2006.

[82] B. Helffer and F. Nier. Quantitative analysis of metastability in reversible diffusion processes via a witten complex approach. *Société Mathématique de France*, 2006.

[83] F. Herau, M. Hitrik, and J. Sjoestrand. Tunnel effect for Kramers-Fokker-Planck type operators: Return to equilibrium and applications. *International Mathematics Research Notices*, article ID rnn057, 2008.

[84] P. W. Holland and S. Leinhardt. Transitivity in structural models of small groups. *Small Group Research*, 2,2:107–124, 1971.

[85] B. D. Hughes. *Random walks and random environments*. Clarendon Press, Oxford, New York, 1995.

[86] W. Huisinga. *Metastability of Markovian Systems A transfer operator based approach in application to molecular dynamics*. Phd thesis, Fachbereich Mathematik und Informatik, FU Berlin, 2001.

[87] W. Huisinga, S. Meyn, and C. Schütte. Phase transitions and metastability for Markovian and molecular systems. *Ann. Appl. Probab.*, 14:419–458, 2004.

[88] W. Huisinga and B. Schmidt. Metastability and dominant eigenvalues of transfer operators. In B. Leimkuhler, C. Chipot, R. Elber, A. Laaksonen, A. Mark, T. Schlick, C. Schütte, and R. Skeel, editors, *New Algorithms for Macromolecular Simulation*, volume 49 of *Lecture Notes in Computational Science and Engineering*, chapter 11, pages 167–182. Springer-Verlag, 2006.

[89] H. Jeong, S. P. Mason, A.-L. Barabási, and Z. N. Oltvai. Lethality and centrality in protein networks. *Nature*, 411:41, 2001.

[90] H. Jeong, B. Tombor, R. Albert, Z. Oltvai, and A.-L. Barabási. The large-scale organization of metabolic networks. *Nature*, 407,6804:651–4, 2000.

[91] G. Jin, S. Zhang, X.-S. Zhang, and L. Chen. Hubs with network motifs organize modularity dynamically in the protein-protein interaction network of yeast. *PLoS ONE*, 2:e1207, 2007.

[92] M. Joy, A. Brock, D. Ingber, and S. Huang. High-betweenness proteins in the yeast protein interaction network. *J. Biomed. Biotechnol.*, 2:96–103, 2005.

[93] R. Kannan, S. Vempala, and A. Vetta. On clusterings: Good, bad and spectral. *Journal of the ACM*, 51,3:497–515, 2004.

[94] T. Kato. *Perturbation Theory for Linear Operators*. Springer-Verlag, 1995.

[95] B. Kayser, S. Hüffner, and T. O. F. Conrad. Towards modular graph clustering. *In preparation*, 2012.

[96] M. Kim and J. Leskovec. Multiplicative attribute graph model of real-world networks. *Internet Mathematics*, 8(1-2):113–160, 2012.

[97] Y. Kim, S.-W. Son, and H. Jeong. Finding communities in directed networks. *Phys. Rev. E*, 81:016103, 2010.

[98] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220 (4598):671–680, 1983.

[99] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *JOURNAL OF THE ACM*, 46(5):604–632, 1999.

[100] A. Knyazev and M. E. Argentati. Rayleigh-ritz majorization error bounds with applications to fem. *SIAM Journal on Matrix Analysis and Applications*, 31:1521, 2010.

[101] H. A. Kramers. Brownian motion in a field of force and the diffusion model of chemical reactions. *Physica*, 7(4):284–304, 1940.

[102] S. Lafon and A. Lee. Diffusion maps and coarse-graining: A unified framework for dimensionality reduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28:1393–1403, 2006.

[103] A. N. Langville and C. D. Meyer. *Google's PageRank and Beyond: The Science of Search Engine Rankings*. Princenton University Press, 2006.

[104] E. A. Leicht and M. E. J. Newman. Community structure in directed networks. *Phys. Rev. Lett.*, 100:118703, 2008.

[105] J. Leskovec, K. Lang, A. Dasgupta, and M. Mahoney. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Mathematics*, 6(1):29–123, 2009.

[106] T. Li, J. Liu, and W. E. A probabilistic framework for network partition. *Phys. Rev. E*, 80, 2009.

[107] L. Lovász. Random walks on graphs: A survey. *Bolyayi Society Mathematical Studies*, 2, 1993.

[108] L. Lovász and P. Winkler. Mixing of random walks and other diffusions on a graph. *Surveys in Combinatorics, Stirling. London Mathematical Society Lecture Note Series*, 218:119, 1995.

[109] K. Macropol, T. Can, and A. Singh. Rrw: repeated random walks on genome-scale protein networks for local cluster discovery. *BMC Bioinformatics*, 10(1):283, 2009.

[110] I. Marek and P. Mayer. Aggregation/disaggregation iterative methods applied to Leontev and Markov chain models. *Appl. Math.*, 47, 2001.

[111] R. Mattingly. A revised stochastic complementation algorithm for nearly completely decomposable markov chains. *ORSA Journal on Computing*, 7(2), 1995.

[112] E. Meerbach, C. Schütte, and A. Fischer. Eigenvalue bounds on restrictions of reversible nearly uncoupled Markov chains. *Lin. Alg. Appl.*, 398, 2005.

[113] M. Meila and W. Pentney. Clustering by weighted cuts in directed graphs. *Proceedings of the 7th SIAM International Conference on Data Mining*, pages 135–144, 2007.

[114] M. Meila and J. Shi. A random walks view of spectral segmentation. *AI and Statistics (AISTATS)*, 2001.

[115] P. Metzner. *Transition Path Theory for Markov Processes: Application to molecular dynamics*. PhD thesis, Free University Berlin, Germany, 2007.

[116] P. Metzner, C. Schütte, and E. Vanden-Eijnden. Illustration of transition path theory on a collection of simple examples. *J. Chem. Phys.*, 125, 2006.

[117] P. Metzner, C. Schütte, and E. Vanden-Eijnden. Transition path theory for markov jump processes. *Multiscale Modeling and Simulation*, 7(3):1192–1219, 2009.

[118] C. D. Meyer. Stochastic complementation, uncoupling markov chains, and the theory of nearly reducible systems. *SIAM Rev.*, 31, 1989.

[119] J. M. Montoya and R. V. Solé. Small world patterns in food webs. *Journal of Theoretical Biology*, 214,3:405–412, 2002.

[120] B. Nadler, S. Lafon, R. R. Coifman, and I. G. Kevrekidis. *Diffusion Maps - a probabilistic interpretation for spectral embedding and clustering algorithms*, volume 58 of *Principal Manifolds for Data Visualization and Dimension Reduction*. Springer, 2008.

[121] M. E. J. Newman. Mixing patterns in networks. *Phys. Rev. E*, 67,:026126, 2003.

[122] M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45,:167–256, 2003.

[123] M. E. J. Newman. Fast algorithm for detecting community structure in networks. *Phys. Rev. E*, 69:066133, 2004.

[124] M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *PHYS.REV.E*, 74:036104, 2006.

[125] M. E. J. Newman, A.-L. Barabási, and D. Watts. *The Structure and Dynamics of Networks*. Princeton Univ Press, Princeton, NJ, 2006.

[126] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69 (026113), 2004.

[127] F. Noé, C. Schütte, E. Vanden-Eijnden, L. Reich, and T. Weikl. Constructing the full ensemble of folding pathways from short off-equilibrium trajectories. *PNAS*, 106(45):19011–19016, 2009.

[128] J. D. Noh and H. Rieger. Random walks on complex networks. *Phys. Rev. Lett.*, 92, 2004.

[129] J. R. Norris. *Markov chains*. Cambridge University Press, 1998.

[130] G. Palla, I. Derényi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435:814–818, 2005.

[131] S. L. Pimm. The complexity and stability of ecosystems. *Nature*, 307:321–326, 1984.

[132] P. Pons and M. Latapy. Computing communities in large networks using random walks. *J. of Graph Alg. and App. bf*, 10:284–293, 2004.

[133] N. Przulj. *Knowledge Discovery in Proteomics*, chapter Graph Theory Analysis of Protein-Protein Interactions, pages 73–128. CRC Press, 2005.

[134] S. Pu, J. Wong, B. Turner, E. Cho, and S. J. Wodak. Up-to-date catalogues of yeast protein complexes. *Nucleic Acids Research*, 37:825–831, 2009.

[135] E. Ravasz and A.-L. Barabási. Hierarchical organization in complex networks. *Phys. Rev. E*, 67(2):026112, Feb 2003.

[136] E. Ravasz, A. Somera, D. Mongru, Z. Oltvai, and A.-L. Barabási. Hierarchical organization of modularity in metabolic networks. *Science*, 297:1551–1555, 2000.

[137] S. Redner. How popular is your paper? an empirical study of the citation distribution. *The European Physical Journal B*, 4:131, 1998.

[138] H. Risken. *The Fokker-Planck Equation: Methods of Solution and Applications*. Springer, 1996.

[139] M. Rosvall and C. Bergstrom. Maps of random walks on complex networks reveal community structure. *Proc Natl Acad Sci*, 105:1118–1123, 2008.

[140] M. Sales-Pardo, R. Guimera, A. Moreira, and L. Amaral. Extracting the hierarchical organization of complex systems. *Proc. Natl. Acad. Sci. U. S. A.*, 104:15224–15229, SEP 2007.

[141] M. Sarich. *Projected Transfer Operators*. PhD thesis, FU Berlin, 2011.

[142] M. Sarich, N. Djurdjevac, S. Bruckner, T. O. F. Conrad, and C. Schütte. Use multilevel random walks to find modules and hubs in complex networks. *Submitted*, 2011.

[143] M. Sarich, F. Noé, and C. Schütte. On the approximation quality of markov state models. *Multiscale Modeling and Simulation*, 8(4):1154–1177, 2010.

[144] M. Sarich and C. Schütte. Approximating selected non-dominant timescales by markov state models. *Comm Math Sci*, 10(3):1001–1013, 2011.

[145] M. Sarich, C. Schütte, and E. Vanden-Eijnden. Optimal fuzzy aggregation of networks. *Multiscale Modeling and Simulation*, 8(4):1535–1561, 2010.

[146] V. Satuluri and S. Parthasarathy. Symmetrizations for clustering directed graphs. *Proceedings of the 14th International Conference on Extending Database Technology*, pages 343–354, 2011.

[147] S. E. Schaeffer. Graph clustering. *Computer Science Review*, 1,1:27–64, 2007.

[148] C. Schütte. *Conformational Dynamics: Modelling, Theory, Algorithm, and Applications to Biomolecules.* Habilitation thesis, Fachbereich Mathematik und Informatik, FU Berlin, 1998.

[149] C. Schütte, A. Fischer, W. Huisinga, and P. Deuflhard. A direct approach to conformational dynamics based on hybrid Monte Carlo. *J. Comp. Physics Special Issue on Computational Biophysics*, 151:146–168, 1999.

[150] C. Schütte and W. Huisinga. Biomolecular conformations can be identified as metastable sets of molecular dynamics. In *Handbook of Numerical Analysis*, pages 699–744. Elsevier, 2003.

[151] C. Schütte, F. Noe, J. Lu, M. Sarich, and E. Vanden-Eijnden. Markov state models based on milestoning. *J. Chem. Phys*, 2011.

[152] J. Scott. *Social Network Analysis: A Handbook.* Sage Publications Ltd, 1993.

[153] E. Seneta. *Non-negative matrices and Markov chains.* Springer, New York, 1981.

[154] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence,*, 22,8:888–905, 2000.

[155] Y. Singer. Dynamic measure of network robustness. *IEEE 24th Conference of Electrical and Electronic Engineers in Israel*, pages 366–370, 2006.

[156] S. S. Skiena. *The algorithm design manual.* Springer, 1998.

[157] O. Sporns, C. J. Honey, and R. Kötter. Identification and classification of hubs in brain networks. *PLoS ONE*, 2:e1049, 2007.

[158] B. Tadic. Exploring complex graphs by random walks. In P. Garrido and J. Marro, editors, *Modeling of complex systems: Seventh Granada Lectures, Granada, Spain, AIP Conference Proceedings,*, volume 661, pages 24–26. American Institute of Physics, 2002.

[159] E. Tonti. On the variational formulation for linear initial value problems. *Annali di Matematica Pura ed Applicata Series 4*, 95,1:331–360, 1973.

[160] R. R. Vallabhajosyula, D. Chakravarti, S. Lutfeali, A. Ray, and A. Raval. Identifying hubs in protein interaction networks. *PLoS ONE*, 4(4):e5344, 2009.

[161] S. van Dongen. *Graph Clustering by Flow Simulation*. PhD thesis, University of Utrecht, 2000.

[162] D. Venturi. A fully symmetric nonlinear biorthogonal decomposition theory for random fields. *Physica D*, 2010.

[163] S. Wasserman and K. Faust. *Social network analysis: methods and applications*. Cambridge University Press, 1994.

[164] S. Wasserman and J. Galaskiewicz. *Advances in social network analysis : research in the social and behavioral sciences*. Sage Publications, Thousand Oaks, Calif., 1994.

[165] D. J. Watts. *Small Worlds: The Dynamics of Networks between Order and Randomness*. Princeton University Press, 1999.

[166] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1998.

[167] R. J. Williams, N. D. Martinez, E. L. Berlow, J. A. Dunne, and A.-L. Barabási. Two degrees of separation in complex food webs. *Proceedings of the National Academy of Sciences 99*, 99:12913–12016, 2002.

[168] H. Yu, D. Greenbaum, L. H. Xin, X. Zhu, and M. Gerstein. Genomic analysis of essentiality within protein networks. *Trends Genet.*, 20:227–231, 2004.

[169] H. Yu, P. Kim, E. Sprecher, V. Trifonov, and M. Gerstein. The importance of bottlenecks in protein networks: correlation with gene essentiality and expression dynamics. *PLoS Comput. Biol.*, 3:e59, 2007.