M.Sc. Thesis
Master of Science in Engineering

DTU Compute
Department of Applied Mathematics and Computer Science

Convex Relaxation Techniques

for clustering

Arturo Arranz (s160412)

Kongens Lyngby 2018

DTU Compute
Department of Applied Mathematics and Computer Science
Technical University of Denmark

Ørsteds Plads
Building 324
2800 Kongens Lyngby, Denmark
Phone (+45) 45 25 30 31
compute@compute.dtu.dk
www.compute.dtu.dk

Preface

They say that when you are having a good time, time pass very fast and these two years have definitely pass very fast. It has been two lovely years in Copenhagen. Academically, I can not be happier with the education received at DTU. Here I have come across excellent professors and colleagues who demonstrated true passion for science and mathematics. What it started as a big challenge, after switching my field of studies, it ended as new great love for mathematics. It has given me a proper mental map and proper framework of thinking for approaching any problem in life.

Personally, I have accumulated tons of new experiences but more importantly I have met tons of new people from all parts of the globe. Some of them for short period of times, other have became very good friends. I can perfectly imagine ourselves in a few years telling stories about our days in Copenhagen. The "*good ol' days*" we will say.

And now, it is time for acknowledgments. First of all, of course, to my parents. Without them I wouldn't be able to be where I am. Their hard work on giving us the best, has paid off. They can be proud of their both sons. This bring me to my brother. Marcos, you have been a role model that often I have tried to imitate. It is because of you that I followed the path of science and I can not be happier for that. Especial thanks to my supervisors Martin and Anders. I really appreciate your time and effort investment in making sure I kept on track. You have taught me the importance of mathematical rigor and attention to details that I often miss. All I can say is that it has been a complete pleasure working with both of you and I hope that in the future we can continue collaborating.

Lastly, I want to mention to all my friends in Madrid, that I always miss and often make me want to go back.

Arturo Arranz (s160412)

Kongens Lyngby, August 3, 2018

Contents

List of Figures

List of Tables

Acronyms

DC-SBM Degree-Corrected stochastic blockmodel. v, 16, 17, 38

GN Girvan-Newman. 24

QCQP Quadratically Constrained Quadratic Program. 27, 28, 29, 31

SBM stochastic blockmodel. v, 15, 16, 17, 20, 37

SDP semidefinite programming. 3, 28

SDR semidefinite relaxation. 27, 28, 29, 30

CHAPTER 1

Introduction

In the XVIII century, the mathematician Carl Gottlieb Ehler was obsessed with a particular problem related to his home city's bridge system, the famous problem *of the 7 bridges of Königsberg*. The idea was easy to formulate: which path did an imaginary traveler had to take in order to cross all the bridges, passing through each of them only once. The answer was not so easy. After many attempts, Ehler decided to ask for help from the greatest mathematician he knew, Leonard Euler.

At first, Euler considered the problem unrelated to mathematics, but over time, the riddle caught his attention. The first thing that he noticed was that it did not matter the direction from which the bridges were crossed. This insight allowed him to simplify the problem and represent the bridge system as a graph like in Figure 1.1, where each piece of land was represented as a node and each bridge as a link.

Eventually, by studying the local characteristics of each node, he demonstrated that the problem was infeasible unless an even number of bridges was connected to each node except the starting and finishing node. However, Euler did not think at that time that a whole new field of mathematics would emerge from these results. This was the beginning of graph theory.

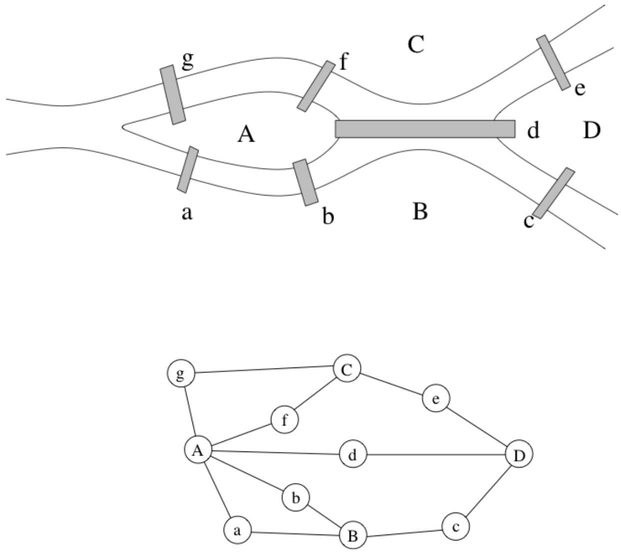


Figure 1.1: Graph representation of Königsberg bridge system.

In essence, a graph is an abstract representation of a system's elements, often called nodes or vertices, and the relations between them called links or edges. However, from the apparent model simplicity studying its properties often lead to very interesting applications in the real world. Here we outline some of the most typical applications:

- *Transportation networks*: Given a network of roads and cities, how can you find the shortest path from point A to B? You can, for example, try all the possible routes but probably a smarter option is modeling it as a graph with the cities as nodes and the roads as links and apply the Dijkstra algorithm for shortest path [38]. This is what Google Maps does.

- *Social networks analysis*: Analyzing a social network can be useful to know how information is spread, who has more influence in the network, finding subgroups, etc.

- *Biological networks*: The human body has more than 120K proteins and studying their interactions is fundamental to understand the cell biological processes.

- *Neural networks*: with neurons as vertex and synapses as edges, neural graphs can be used to study the functioning of our brain.

- *Epidemiology*: As an example, in 1970, a graph of sexual relations among people was created to study the spread and origin of AIDS [21].

- *WWW*: internet web pages can be modeled as a graph where connections are made based on link references from web pages to other web pages. The famous *web crawlers* running on the networks are the key for the Google indexing algorithms [9].

These are just a few examples of applications of graph theory. However, the scope of this thesis is constrained to the specific problem of finding *communities* or also known as *clusters or modules*. Such communities are abstractly understood as groups of nodes that are "alike" which are generally associated with subgraph where its nodes very interconnected but poorly connected with the external ones. In the Figure 1.2 the reader can observe a graph which clearly exhibits communities. It represents research collaborations in The Santa Fe Institute. A researcher who knows his or her community identity could find potential collaborations easily. This is just an example of an application, but finding communities is central for studying a large diversity of topics such as social behavior [19, 35], protein to protein interactions [11, 31], gene expression [12], recommender systems [28, 39, 45], page ranking [26], image segmentation [41] and many more.

Community detection has advanced significantly since 1980 and has become more relevant lately as a larger amount of data are available. Due to the ubiquitous relevance of the topic, the development has come from several disciplines such as computer science, social science, mathematics, ecology, statistical physics, etc. The huge variety of proposed algorithms is probably explained by the ill-definition of the community detection problem

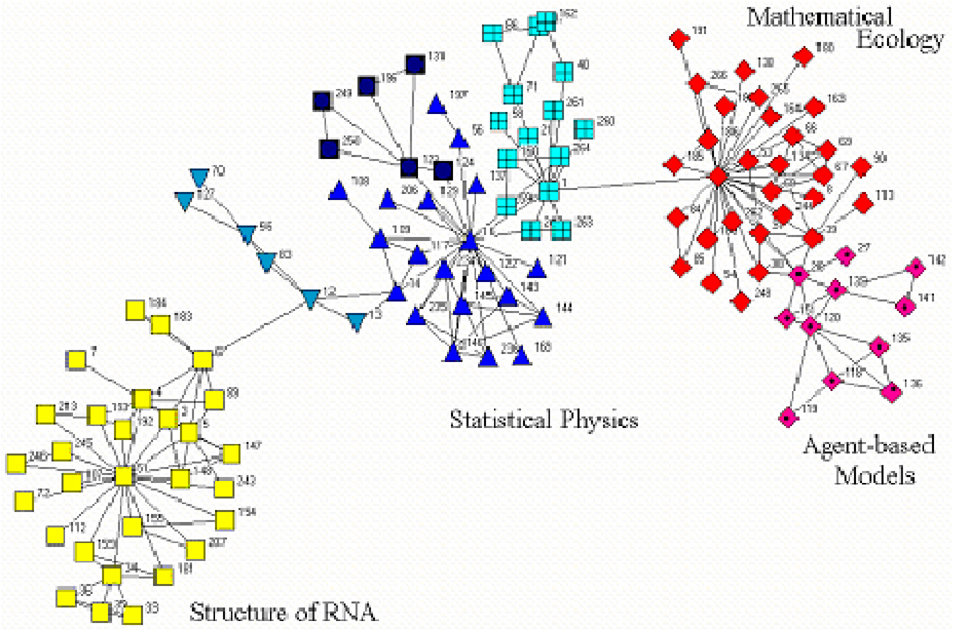


Figure 1.2: Network of Santa Fe researchers collaborations.

[16]. This means that there is no universal definition of what a community is, and usually, it is defined by the specific question we want to answer or the type of network we are analyzing. Intuitively, as we said a community is considered a sub-group which is compactly connected. However, consider a bipartite network as in the figure 1.3. In this kind of networks, each node is connected to the opposite group nodes but not to its own, which is contrary to our previous definition of community. This illustrates the importance of asking the right questions when studying a system, process or network.

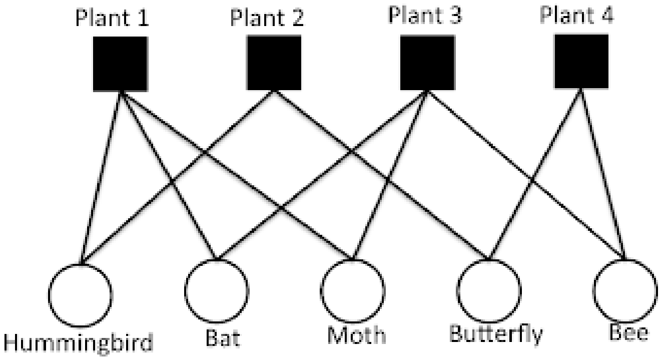


Figure 1.3: Bipartite network of animals and plants.

In any case, we will focus, as most of the literature, on trying to find communities that are tightly connected. To this end, most of the available algorithms can be classified into 3 approaches: statistical inference, dynamical systems, and optimization. In particular, we are interested in the latter one, where recent developments on semidefinite programming

(SDP) have to lead to very robust and effective graph partitions even in noisy environments. Concretely, the aim of this master thesis is to make a comparative analysis of several algorithms with the SDP formulations, explain its strengths, show connections with other formulations, and explore new ways to define the objective function, i.e new ways to define a community.

CHAPTER 2
Background

2.1 Graph Theory, Notation and Definitions

In this chapter we will give a more formal introduction to graph theory by establishing basic definitions, notations, properties and we will give some examples of archetypal networks, the stochastic block model or the Erdös–Rényi model and explain its relevance for the community detection problem.

Mathematically speaking an undirected graph is a pair $G = (V, E)$ where $V = \{v_1, v_2, .., v_n\}$ is a set of nodes and E is the set of edges where each edge $e_k = \{v_k, v_p\}$ contains the information of the connected pair of nodes. Furthermore, graphs can also be classified in terms of the nature of its connections. It can be directed or undirected if the direction of the connections matter. The links can also be weighted to denote the connection strength between a pair of nodes.

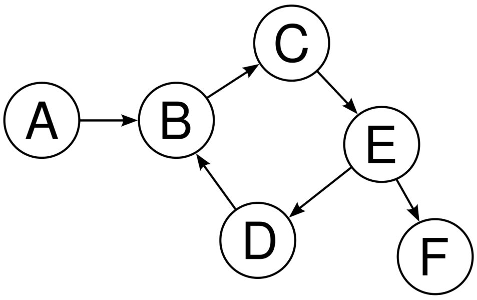


Figure 2.1: Directed graph.

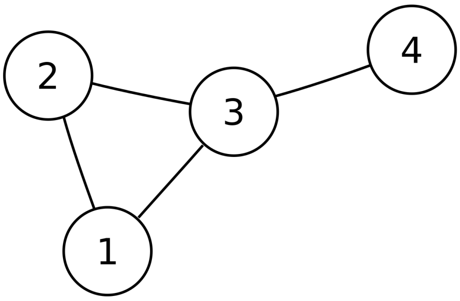


Figure 2.2: Undirected graph.

For convenience we introduce the shorthand notation $i \in A$ for the set of indices $\{i|v_i \in A\}$. A graph contain many properties, nomenclature and concepts. Here we enlist the more relevant:

- Two nodes i, $j \in V$ are adjacent/neighboring if $\{i, j\} \in E$.
- The **degree**, d_i, for an undirected unweighted graph, it is the number of nodes adjacent to i.

- For a undirected weighted graph, the *degree*, it is the sum of the edges connected to the the node.
- Two different measures of a graph "size" are:

$$\begin{aligned}|G| &:= \text{the number of nodes in G}\\ \text{vol}(G) &:= \sum_{i\in G} d_i\end{aligned}$$

- A **path** is the set of crossed nodes in order to get from i to j. The **distance**, $\delta(i,j)$ is the minimum number of nodes needed to cross from i to j.
- It is said that a node i is **reachable** from j if there is a path that joins them. If all the nodes of an undirected graph are reachable the graph is *connected.*
- We talk about **self-loops** when an edge joins a node with itself. Graphs with self-loops are not analyzed in the present work.
- We denote $\mathbb{1}$ as the vectors of ones.
- $\bar{A}$ will denote *the complement* of a set of nodes $A \in V$. The complement is the rest of nodes from V that are not included in A.

2.2 Linear Algebra Representation

A graph can be represented in a matrix form. Concretely, it can be represented by the square *adjacency matrix*, W, where w_{ij} represents the connection weight between the nodes i and j. For example, an unweighted graph would have $w_{ij} = 1$ if nodes are connected and $w_{ij} = 0$ if nodes i and j are unconnected. The following matrices W_d and W_u are the corresponding adjacency matrices of graphs in Figure 2.1 and 2.2 respectively.

$$W_d = \begin{bmatrix}0&1&0&0&0&0\\0&0&1&0&0&0\\0&0&0&0&1&0\\0&1&0&0&0&0\\0&0&0&1&0&1\\0&0&0&0&0&0\end{bmatrix} \qquad W_u = \begin{bmatrix}0&1&1&0\\1&0&1&0\\1&1&0&1\\0&0&1&0\end{bmatrix}$$

Note that the adjacency matrix of an undirected graph is always symmetric as $w_{ij} = w_{ji}$ for every j and i. This representation allows us to use linear algebra to study the graph properties. For example, if two nodes, *i*, and *j*, happen to share the exact same nodes implies that W is not full rank as its two columns, W_i and W_j would be linearly dependent.

2.2.1 Counting Steps with Matrix Multiplication

Assuming that every node is reachable(connected graph). We define the *path-length* n between two nodes i and j as a list of ordered nodes $i, k_1, k_2, ..., k_{n-1}, j$ which are consecutively connected, i.e:

$$a_{i,k_1} = a_{k_1,k_2} = ... = a_{k_{n-2},k_{n-1}} = a_{kn-1,j} = 1$$

In other words, it is the number of nodes needed to cross in order to get from i to j. Hence, the adjacency matrix indicates the path-lengths 1 if $a_{ij} = 1$ or higher if $a_{ij} = 0$. However, what happens with W^2? Knowing that its entries are $(W^2)_{ij} = \sum_{k=1}^{N} a_{ik}a_{kj}$ we can interpret as following: for each connected node to i check if there is a connection with j. this is basically the sum of path-length 2 between i and j. Readily, we can see how $(W^2)_{ii}$ is similar to the deg_i as it counts the number of walks that start and end in the same node. It is always possible to visit the adjacent nodes and come back through the same connection. This result can be generalized to higher order exponents

$$(W^n)_{ij} = \sum_{k_1=1}^{N} \sum_{k_2=1}^{N} \sum_{k_{n-2}=1}^{N} \sum_{k_{n-1}=1}^{N} a_{i,k_1} a_{k_1,k_2} ... a_{k_{n-2},k_{n-1}} a_{k_{n-1},j}$$

which measures the number of walks of length n that connect the nodes. This leads to the following lemma:

Lemma 1: The quantity $(W^n)_{ij}$ counts the number of different walks $(i \neq j)$ or closed walks $(i = j)$ of length n between nodes i and j.

2.2.2 The Graph-Laplacian

The Laplacian [32] is another interesting matrix for analyzing graphs. It is defined as $L = D - W$, where D is the diagonal matrix which entries d_{ii} are the node i degree. For example, the Laplacian of the undirected graph from Figure 2.2 would be

$$L = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} - \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 & -1 & -1 & 0 \\ -1 & 2 & -1 & 0 \\ -1 & -1 & 3 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

Laplacian representation of graphs have been studied throughly in the literature. Here, we enumerate some its properties:

1. *For every vector $x \in \mathbb{R}^n$ we have*

$$x^T L x = \frac{1}{2} \sum_{i}^{n} \sum_{j}^{n} w_{ij}(x_i - x_j)^2 \tag{2.1}$$

2. *L is contained in the set of positive semidefinite cones, S^n_+. i.e. L is symmetric and positive semidefinite*

3. *0 is always an eigenvalue associated to the eigenvector $\mathbb{1}$.*

4. *The algebraical multiplicity of $\lambda = 0$ is the number of connected components in the graph*

Proof statement (1):

$$\begin{aligned} x^T Lx = x^T Dx - x^T Wx &= \sum_{i=1}^{n} d_i x_i^2 - \sum_{i=1}^{n}\sum_{j=1}^{n} x_i x_j w_{ij} \\ &= \frac{1}{2}\Big(\sum_{i=1}^{n} d_i x_i^2 - 2\sum_{i=1}^{n}\sum_{j=1}^{n} x_i x_j w_{ij} + \sum_{j=1}^{n} d_j x_j^2\Big) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}(x_i - x_j)^2 \end{aligned}$$

Proof statement (2): the symmetry of W is maintained as D is just a diagonal matrix. The positive semi-definiteness follows from the statement one, $x^t Lx \geq 0$ for any $x \in \mathbb{R}^n$.

Proof statement (3): all the rows sum to one. If all the rows are linearly combined with equal weighting, we get the vectors of zeros, i.e.

$$L\mathbb{1} = \vec{0}\mathbb{1}$$

Proof statement (4): each connected component will be formed by a set of linearly dependent columns in the adjacency matrix which will result in k eigenvectors associated to eigenvalue 0, where k denotes the number of components. In addition to the eigenvector, $\mathbb{1}$ from statement 3, it will result in k eigenvectors with eigenvalue 0. For example:

$$L = \begin{bmatrix} 2 & -1 & -1 & 0 \\ -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Would have the eigenvectors $v_1 = [1, 1, 1, 1]$, $v_2 = [1, 1, 1, 0]$ and $v_3 = [0, 0, 0, 1]$ which satisfy:

$$L\vec{v}_n = \vec{v}_n 0$$

2.3 Centrality, Communicability, and Betweenness

We can have different measures that tell us information about a network. Some examples are the centrality, communicability, and betweenness. We will see that there are many ways to formally formally these measures, but we can abstractly define them as:

- *Centrality*: the importance of a single node in terms of surrounding density
- *Communicability*: measures the well-connectedness between 2 nodes
- *Betweenness*: how much information travels through a node. If for instance, we imagine a road network connecting cities, we can think of betweenness as how many cars travel through a particular city, i.e how important such node is for connecting with the rest

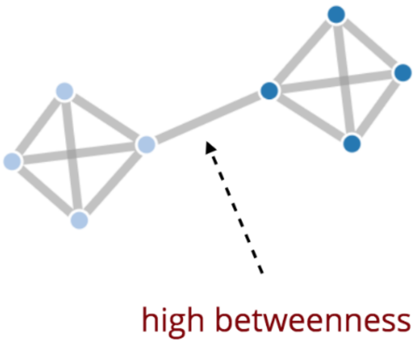


Figure 2.3: Betweenness illustration.

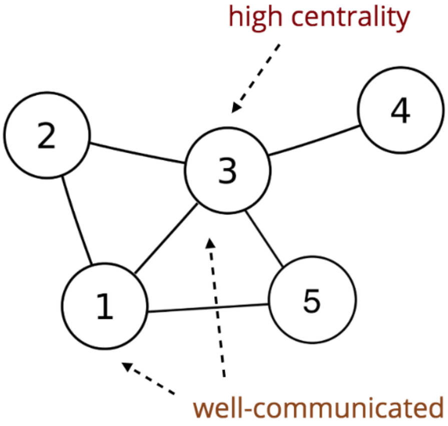


Figure 2.4: Communicability and centrality illustration.

For instance, in Figure 2.3 the edge bridging the two clusters would have a high betweenness since it would be the path for every walk across clusters. In Figure 2.4 we have an example of high centrality where the node is surrounded by many nodes. Inversely, the node 4 would have a low centrality. Finally, in the same Figure we can observe how the node 1 and 3 would score a high communicability as many short paths are available between both nodes. Mathematically the easiest way to define the centrality is by means of the degree,

$$d_i = \sum_{i}^{N} a_{ik} = (We)i$$

However, this measure does not take in account anything but the immediate surroundings of the node. Katz proposed in 1953 [25] a more sophisticated measure

$$k_i := \sum_{j=1}^{N} \sum_{k=0}^{\infty} \alpha^k (W)_{ij}^k$$

which also measures the impact of further nodes but giving more weight to the closer ones. However, a more general framework can be defined.

2.3.1 General Framework

As stated above, centrality, communicability, and betweenness are measures of *well-connectedness* which can be related to powers of adjacency matrix. For instance, the centrality is equivalent to the diagonal of the squared adjacency matrix W^2, which is the sum of all the closed loop walks of length 2. However, the closed loop walks of length 3 are an indicator of well-connected neighbors. And the same can be extended for longer closed loop walks. Adding this information result in a richer measure of connectedness. Nonetheless, information is less likely to travel through longer paths so it seems natural to introduce a down-weighting factor. With this in mind, we introduce the following function

$$f(W) = \sum_{n=1}^{\infty} c_n W^n \tag{2.2}$$

where $\{c_n\}_{n\geq 1}$ is the sequence of down-weighting coefficients where its elements must be non-negative. The sequence must make the sum convergent. Then we can define the *f-measures* as:

- Centrality of node i: $f(W)_{ii}$
- Communicability between i and j: $f(W)_{ij}$
- Betweenness of the node r:

$$\frac{1}{(N-1)^2-(N-1)}\sum\sum\nolimits_{i\neq j,j\neq r,\neq r}\frac{f(W)_{ij}-f(W-E(r))_{ij}}{f(W)_{ij}}$$

While the definitions of centrality and communicability might now seem obvious, there might be a gap in the betweenness formal and abstract definition. The formal definitions measure what is the overall communicability lost of the network if we remove the node r. Then $(W - E(r))$ eliminates the connections of r from W where $E(r) \in R^{NxN}$ has non-zero only in the row and column r, and row and column r have 1 wherever W has 1. Each of the terms in the sum is a normalized value and the coefficient $1/(N-1)^2-(N-1)$ averages over all the terms. An interesting property of the infinite sum worth to study is its spectrum

$$\begin{aligned} f(W) = \sum_{n=0}^{\infty} c_n W^n &= c_0 W^0 + c_1 W + c_2 W^2 + ... + c_k W^k + ... \\ &= c_0 I + c_1 Q\Lambda Q^T + c_2 (Q\Lambda Q^T)^2 + ... + c_k (Q\Lambda Q^T)^k + ... \\ &= c_0 I + Qc_1 \Lambda Q^T + Qc_2 \Lambda^2 Q^T + ... + Qc_k \Lambda^k Q^T + ... \end{aligned}$$

where the $Q\Lambda Q^T$ is the eigendecompostition of the symmetric adjacency matrix W. Clearly, now we can see how the function in (2.2) is basically a mapping of its spectrum as

$$\text{eig}(f(W))_i = \sum_{n=0}^{\infty} c_n \lambda_i^n = f(\lambda_i)$$

2.3.2 Special Case: Matrix Exponential

In particular, if we choose a weight decay of $c_n = \frac{1}{n!}$ it results in

$$f(W) = \left(I + W + \frac{W^3}{3!} + ... + \frac{W^k}{k!} + ... \right)$$

which is the definition of the matrix exponential $\exp(W)$. This is a perfectly valid down-weighting as it constantly increases the dumping over long walks and leads to a convergent sum. This construction, defined by Estrada [15], has also the advantage of resulting in a well known matrix function. The spectrum is mapped as

$$\text{eig}(\exp(W))_i = \sum_{n=0}^{\infty} \frac{1}{n!} \lambda_i^n = e^{\lambda_i}$$

2.3.3 Special Case: The Resolvent Function

The previous case has not much more motivation besides that it leads to a nice computationally form. Also, Estrada proposed in [15] to down-weight each path of length k compared with maximum possible paths of the same length admitted by K_N defined as the fully connected graph of the same size. While these sequences would naturally decrease as longer paths outnumber shorter ones, in K_N, the exponential function neglect sharply the long walks.

From Lemma 1.1, we can easily calculate the path-length k of K_N by knowing that it correspondent adjacency matrix is $(J - I)$ where $J \in \mathbb{R}^{NxN}$ is the matrix of all ones. However, Estrada approximated the infinite sum for large N as

$$f(W) = (I - \alpha W)^{-1} \tag{2.3}$$

where the parameter α controls the down-weighting sequence.

2.4 Graph Cuts

Graph-cuts are intimately related with community detection. Formally defined, a cut is the partition of a graph, $G = (V, E)$, in two subgraphs, $S \in V$ and $\bar{S} \in W \backslash V$, i.e dividing the vertices into two disjoint sets. Such cuts can be performed in different manners.

2.4.1 Minimum and Maximum Cuts

If for example we would require to make a division where each sub-graph contains as few as possible connections, we would be talking about the well-known *maxiumum cut* problem defined as:

$$\max \quad \frac{1}{2}\sum_{i\in S}\sum_{j\in \bar{S}} w_{ij} \tag{2.4}$$

which can be reformulated as

$$\begin{aligned} &\max_{x} \quad \frac{1}{4}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}(1 - x_i x_j) \\ &\text{s.t} \qquad x_i \in \{-1, 1\} \quad \text{for} i = 1, ...n \end{aligned} \tag{2.5}$$

where $x_j = 1$ if $j \in S$ and $x_i = -1$ if $i \in \bar{S}$. This is a combinatorial problem which was demonstrated to be NP-complete [23]. However, Goemans and Williamson [18] proved that the problem can be approximated within 0.875 of the global solution via semidefinite programming. This is a remarkable result for a NP-hard problem.

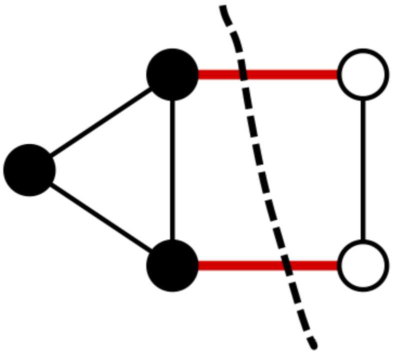


Figure 2.5: Min-cut illustration.

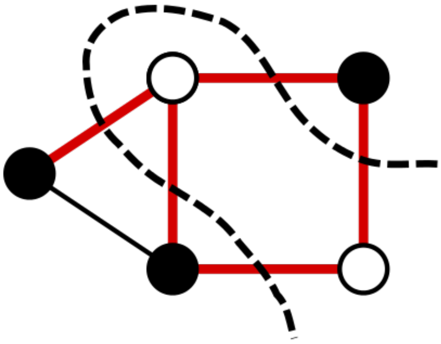


Figure 2.6: Max-cut illustration.

Contrary, our purpose is actually partitioning in a way that connections between the sub-graphs is minimum or alternatively that each sub-graph is densely connected. In this case we are talking about another well-known problem called *minimum cut*.

$$\min_{x} \quad \frac{1}{2}\sum_{i\in S}\sum_{j\in \bar{S}} w_{ij} \tag{2.6}$$

which is also a combinatorial problem, with the trivial solution of no cutting edges. There exist modified version, as the *sp-cut* where you have to find the minimum-cut that divide the nodes s and p. This problem can be solved efficiently in polynomial time [42]. However, it come with a major drawback. It often leads to trivial solutions where only one node is separated from the rest of the graph. In practice communities contain several nodes so we need to include some kind of group size balancing constraint, as follows

$$\begin{aligned} \min_x \quad & x^T L x \\ \text{s.t} \quad & x_i \in \{-1, 1\} \\ & \mathbb{1} x = 0 \end{aligned} \tag{2.7}$$

since, as proved

$$x^T L x = \frac{1}{2} \sum_i^n \sum_j^n w_{ij} (x_i - x_j)^2$$

We could built an equivalent problem by writing in terms of the adjacency matrix by minimizing $-x^T W x = -\sum_{i=1}^n \sum_{j=1}^n w_{ij} x_i x_j$. During the rest of the document we will stick to the Laplacian formulation.

Unfortunately, introducing this balancing constraint in 2.7 comes with a big computational expense as the problem becomes NP-hard [43]. Many relaxed versions have been proposed, in particular, spectral clustering is a very popular one which we will explain shortly. Semidefinite relaxations are another breed which has gain popularity lately, [3, 5, 22]. The latter kind of relaxations is the main focus of this thesis and the core of the proposed solution. Finally, one could extend a graph-cut to admit several partitions. If we define $C(A, B) = \sum_{i \in A, j \in B}$ where A and B denote subgraphs, a k-cut would be defined as

$$\text{cut}(A_1, ...A_k) := \frac{1}{2} \sum_{i=1}^k C(A_i, \bar{A}_i) \tag{2.8}$$

Nonetheless, the scope of the present work is attained to the case of $k = 2$.

2.4.2 Spectral Relaxation of Balanced Min-cut

Before explaining these results we have to make a small parenthesis and explain what is a relaxation, which is central for the understanding not only this section but the core of the present work.

What is a relaxation?

A relaxation, in mathematical optimization, is an approximation of a problem by a another one that is easier to solve. For instance, if we had the following minimization problem

$$z = \min\{f(x) : x \in X \subseteq \mathbb{R}^n\}$$

and its relaxation version

$$z_R = \min\{f_R(x) : x \in X_R \subseteq \mathbb{R}^n\}$$

Then the two following conditions must satisfy:

1. $X_R \supseteq X$
2. $f_R(x) \leq f(x) \quad \text{for all} \quad x \in X.$

Meaning that the the original feasible set, X, is contained in the expanded set, X_R, and that the original function is always greater or equal than the relaxed version for points in the original set.

This implies that we can get lower bounds to the original problem by solving the relaxed version and forcing feasibility. Better relaxation will produce smaller gaps between the true solution and the lower bound. Furthermore, if $f(x) = f_R(x)$ for all $x \in X$, then solutions for the relaxed problem that are feasible on the original problem are also solutions for the original problem.

Now, the spectral relaxation consist on relaxing the discreteness constraint of (2.7) as

$$\begin{aligned} \min_x \quad & x^T L x = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}(x_i - x_j)^2 \\ \text{s.t} \quad & ||x||_2 = \sqrt{n} \\ & \mathbb{1}x = 0 \end{aligned} \tag{2.9}$$

This is a valid relaxation as the new problem domain includes the original one and it is also a tight approximation as the Figure 2.7 illustrated. In the there, the red dots represent the feasible solutions for the problem where $x \in \mathbb{R}^2$. The feasible domain in the relaxed version is just the circle that contain them. For more dimensions it is generalized with an hyper-sphere.

It is not straightforward to understand why the new problem is easier to solve. However, it finds out that the solution is given by the second eigenvector of L(assuming

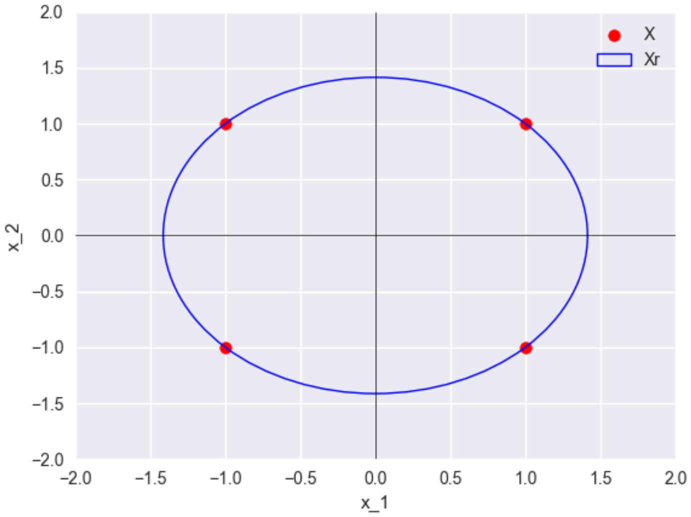


Figure 2.7: Spectral relaxation in $\mathbb{R}^2$.

that the graph is connected). This can be shown with a simple eigenbasis decomposition:

$$\begin{aligned} x^T L x &= x^T Q \Lambda Q^T x \\ &= y^T \Lambda y \\ &= \sum_{i=1}^{n} \lambda_i y_i^2 \end{aligned}$$

where the change of variable $y = Q^T x$ was made. We can see that the minimal objective value would be the vector associated with the smallest eigenvalue. However, recall that for a connected graph the smallest eigenvalue is 0 associated to the eigenvector $\mathbb{1}$. Nonetheless, the constraint $\mathbb{1}x = 0$ removes such trivial solution and leads to the second eigenvector since L is symmetric and its eigenvectors are mutually orthogonal.

2.5 Graphs with Planted Partitions

2.5.1 The Stochastic Block Model

While there are many types of graphs we are exclusively interested in the ones that display communities. For this purpose, the SBM, which falls into the category of random graphs, provides an excellent framework. It has been extensively studied in the literature [24, 2, 17] and different variations have arisen.

Each node of the SBM v contains a latent variable $z_v = 1, \cdots, k$ indicating to which cluster it belongs. This assignment is made by multinomial distribution parametrized by γ, i.e. $\gamma_k = P(z_v = k)$. Then, the probability of linkage among nodes depends if they belong to the same or different clusters. The links are generated by independent draws of a Bernoulli distribution which parameters are specified in the affinity matrix $\rho \in \mathbb{R}^{kxk}$. Hence the model is defined as

$$SBM(n, \gamma, \rho) \left\{ \begin{array}{l} z_i \sim_{iid} \text{Multinomial}(\gamma) \\ w_{ij} | z_i, z_j \sim_{ind} \text{Bernoulli}(\rho_{z_i z_j}) \end{array} \right. \tag{2.10}$$

The focus of the present thesis is the undirected bi-cluster case of $k = 2$:

$$\rho = \begin{bmatrix} p_1 & q \\ q & p2 \end{bmatrix}$$

where p ad q are denominated the inter-cluster and intra-cluster probabilities.

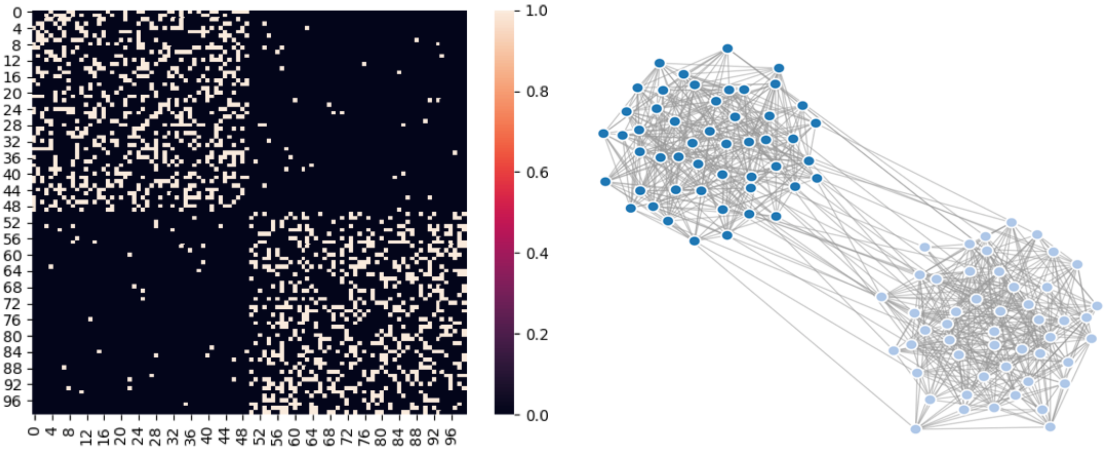


Figure 2.8: Balanced SBM with $p_1 = p_2 = 0.3$ and $q = 0.01$ and $\gamma = \{0.5, 0.5\}$.

The major advantage of SBM is its generative nature as ground truth labels can be utilized to measure the accuracy of the algorithms. On the other hand, one may wonder if the model is a good fit for real-world networks, which leads to the next variant, the DC-SBM.

2.5.2 The Degree-Corrected Stochastic Blockmodel

One of the features of networks generated by the SBM is the degree homogeneity among the nodes. Since the linking probability is the same for all the nodes in the same community, on average every node connects to the same amount of nodes. Although this characteristic can be desirable in some cases many real networks present highly skewed degree distributions. In fact, Barabási and Albert [6] demonstrated that most of the real networks follow a power law distributions.

In order to accommodate the necessity of degree heterogeneity, Newman introduced the DC-SBM [24] which is defined as

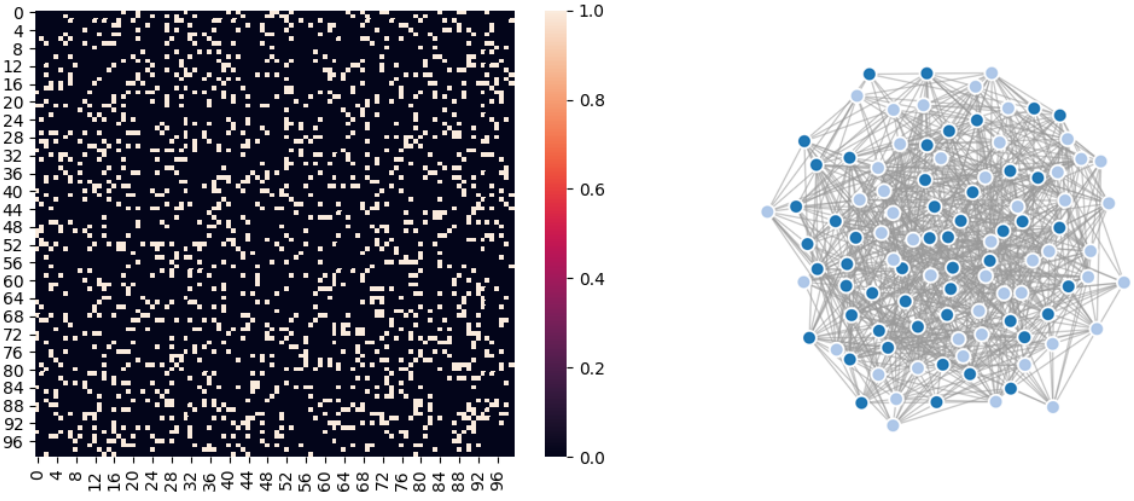


Figure 2.9: Special SBM case where inter-cluster and intra-cluster probabilities are the same with $p_1 = p_2 = q = 0.15$ and $\gamma = \{0.5, 0.5\}$. This corresponds to the Erdös–Rényi model..

$$DC\text{-}SBM(n, \gamma, \rho, \theta) \begin{cases} z_i \sim_{iid} \text{Multinomial}(\gamma) \\ w_{ij} | z_i, z_j \sim_{ind} \text{Poisson}(\theta_i \theta_j \rho_{z_i z_j}) \end{cases} \tag{2.11}$$

The difference with the general SBM is the introduction of parameter $\theta \in \mathbb{R}^n$ which scales the linking probability of each individual node. The higher θ_u, the higher linking probability of u with every other node. Additionally, the Bernoulli is swapped by a Poisson distribution in order to be able to do the scaling. Notice that this allows the existence of more than one edge between two pairs of nodes and self-loops. However, in the limit of a large sparse graph, the probability of an edge and the expected number of edges. By making sure that the expected value is no higher than 1, there is essentially no difference. In the Figure 2.10 it is shown a DC-SBM ensemble.

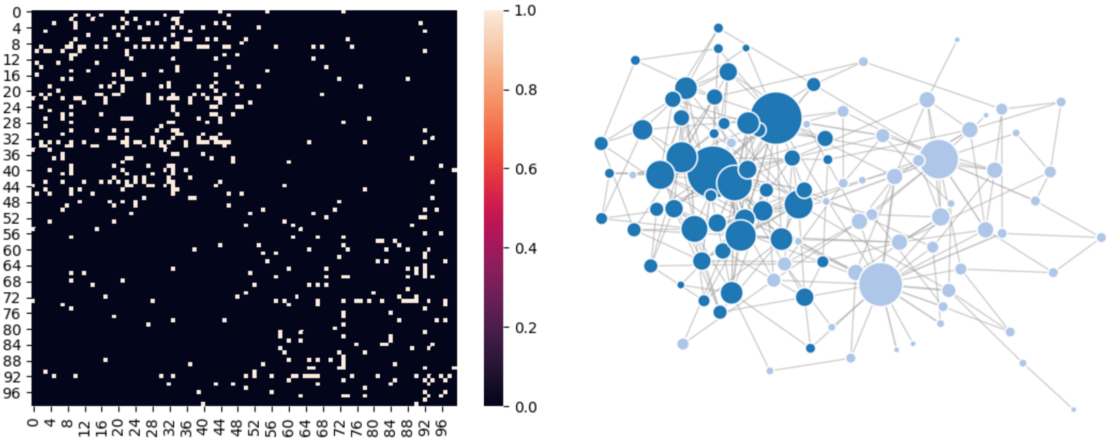


Figure 2.10: DC-SBM for $p_1 = p_2 = 0.03$, $q = 0.005$, $\gamma = \{0.5, 0.5\}$ and $\theta_n = 3Pois(0.1)$. Nodes with big radius represent high degree.

CHAPTER 3
Community detection state of the art

3.1 Spectral Methods

Spectral clustering is an old unsupervised machine learning technique popularized at the beginning of the century [40, 29, 36] and nowadays it is used in most of the data analysis task as first attempt to find groups. Besides its computational simplicity, it is not straight forward to see how it works. In this section we will explain the method and we will see how it is highly connected to the spectral relaxation of the *balanced min-cut* from previous chapter.

Firstly, spectral clustering maps data from Euclidean space into a graph, therefore it can also be used for community detection by just skipping this step. In order to make the mapping, the graph adjacency matrix is constructed by the pairwise similarity between each data point. For instance, given a dataset $X = \{x_1, ..., x_n\}$ where $x \in \mathbb{R}^d$ the adjacency matrix, W, can be constructed as

$$w_{ij} = f(x_i, x_j) = \exp\Big(-\frac{||x_i - x_j||^2}{2\sigma}\Big)$$

where the strength of the connection is determined by the distance between the data points. This similarity function is known as Gaussian kernel, but there exist many other options.

Now, let us assume that a connected graph has been generated and we would like to find k clusters. Next step is to build the Laplacian matrix, L, from W. From its the properties, we have already seen that the eigenvectors associated to the eigenvalue 0 denote the number of connected components. Here we are assuming that we have only one connected component(a connected graph), but this idea gives a hint of what a non-zero but small eigenvalue of the Laplacian means: directions of high separability.

To illustrate this concept consider the toy example from the Figure 3.1 which consist of 400 independently drawn samples from 4 different Gaussian distributions. After constructing the adjacency matrix the eigenvalues of its Laplacian are computed, Figure 3.2. As expected the first eigenvalue is zero and the following 3 are very small (but non-zero).

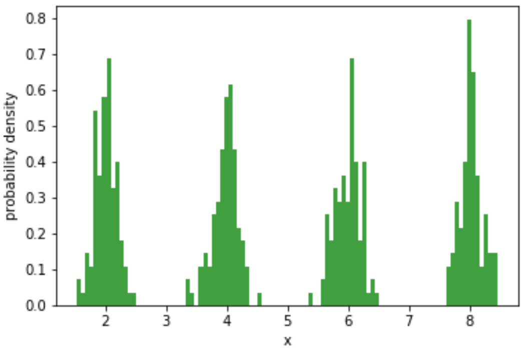


Figure 3.1: Generated data.

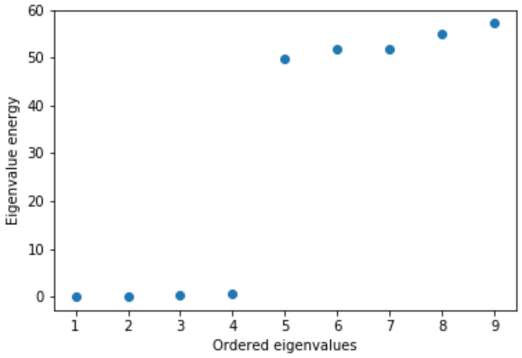


Figure 3.2: Laplacian Eigenvalues.

In the Figure 3.3 it is shown the eigenvectors associated to the 4 smallest eigenvalues where the x-axis represent the nodes which have not been ordered. The first eigenvector, as always is $\mathbb{1}$. But know we can see how encoding the nodes with the 3 following eigenvectors make them completely separable. In fact, for this specific simple case only the first one would be sufficient.

Finally, the last step is to label each node/data-point. If we construct $U \in \mathbb{R}^{n\times(k-1)}$ where the columns are the first k eigenvectors (after ordering and excluding the first), we can feed to a simpler linear classifier such as *k-means* [30]. We can then summarize all the steps

1. *Create a adjacency matrix using a similarity function and setting $w_{ii} = 0$ for all $i = 1, ...n$*

2. *Compute its Laplacian $L \in \mathbb{R}^{n\times n}$.*

3. *Compute the first eigenvectors, $u_2, u_3, ...u_k$, of the k smallest eigenvalues excluding the first. And let $U^{n\times(k-1)}$ be the matrix composed by such eigenvectors.*

4. *Defining $y_i \in \mathbb{R}^n$ for $i = 1, 2, ...n$ as the row vectors of U, make clustering of the points y_i using the k-means algorithm into the clusters $C_1, ...C_k$.*

Spectral methods are highly connected with the dimensionality reduction method of Principal Component Analysis (PCA) where one extract the eigenvector associated to the biggest eigenvalues of a covariance matrix, which is also a symmetric matrix. Also we can see the spectral relaxation of the *balanced min-cut* as a special case of Spectral Clustering where $k = 2$ and instead of using *k-means* we simply use rounding to 1 or -1.

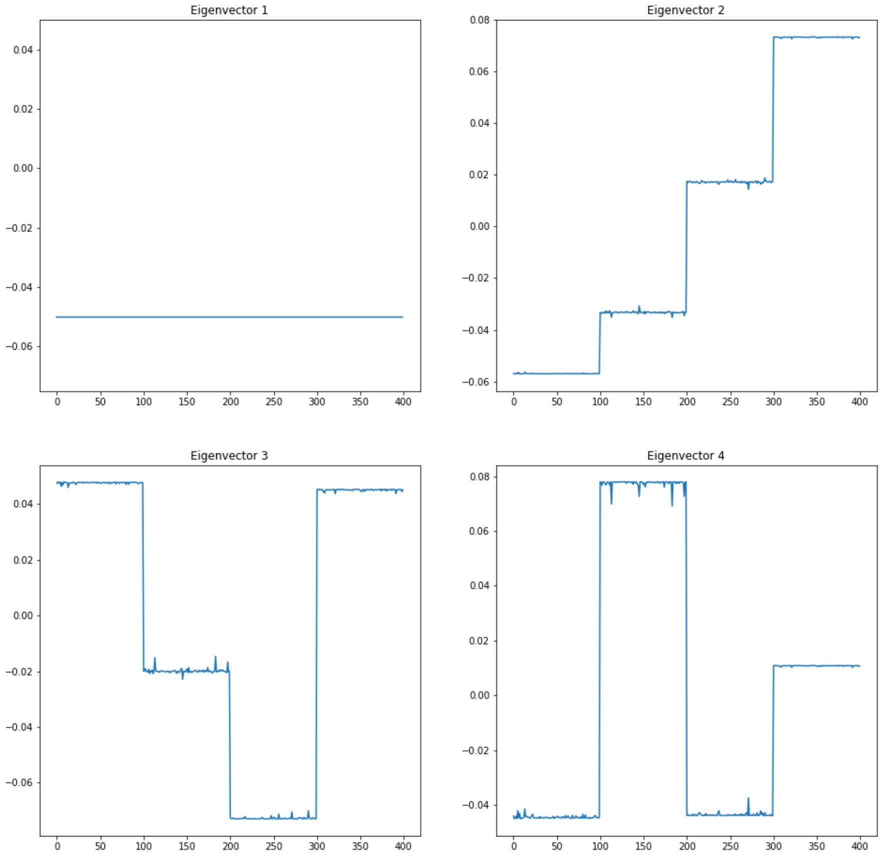


Figure 3.3: Eigenvectors of the 4 smallest eigenvalues.

3.2 Statistical Inference

Community detection can be also tackled from a statistical point of view. The idea is to fit a graph generative model with planted partitions. The SBM and its variants introduced in the previous chapter is the most used in the literature [44, 46, 24]. If we where given the vector $z \in \mathbb{R}^N$ denoting the label of each node, the log-likelihood of the general SBM would be

$$logP(W, z|\gamma, \rho) = \sum_{r}^{k} n_r \text{log}\gamma_r + \frac{1}{2}\Bigg(\sum_{r,s}^{k} m_{rs}\text{log}\rho_{rs} - n_r n_s \rho_{rs}\Bigg)$$

where m_{rs} is the number of edges running from group r to group s and $n_r(n_s)$ the number of nodes in r (s). Maximizing over γ and ρ gives,

$$\hat{\gamma}_r = \frac{n_r}{n}, \quad \hat{\rho}_{rs} = \frac{m_{rs}}{n_r n_s} \tag{3.1}$$

But assuming that we already have the partition ,z , does not get us anywhere, as this is exactly what we would like to find. Instead, we are interested on

$$P(W|\rho,\gamma) = \sum_{z} P(W,z|\rho,\gamma) \tag{3.2}$$

However, marginalizing over z becomes computationally intractable as it amounts for a sum over all the possible partitions (k^n). Several approaches to tackle the problem have been proposed. In particular different variants of the EM algorithm [33], which consist in iterating over two steps: and Expectation (E) step which approximates the full marginal, $P(W|\rho,\gamma)$, and the Maximization (M) step which estimates γ and ρ in order to maximize the approximation. Several approaches to the E step have been proposed such as using Monte Carlo Markov Chain (MCMC) algorithm to sample from the joint posterior $P(W,z|\rho,\gamma)$ [37], belief propagation techniques [46] or variational methods [27]. Once that z is fixed, the E step can be solved as in (3.1). Eventually, after convergence, we recover the estimations $\hat{z}$, $\hat{\gamma}$ and $\hat{\rho}$.

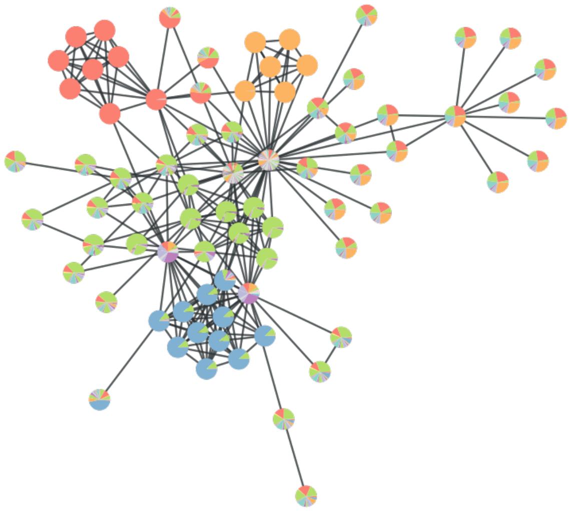


Figure 3.4: Marginal probabilities of group membership in a network.

Statistical inference approach has advantages such as the probabilistic framework which provides uncertainty measures as in Figure 3.5. However, its most important drawback is the difficulty of choosing the number of clusters k. Maximization over all possible k would lead to the trivial solution of $k = n$, i.e one cluster per node(*overfitting*). In the machine learning community, this problem is generally tackle by multi-fold crossvalidation but network data is globally dependent which makes it difficult to split into training and validation sets. Several heuristic have been proposed [20, 10]. Finally, one also have to hope that the chosen model is representative of the true network.

3.3 Optimization Based Methods

Optimization methods are probably the most popular approach to solve the community detection problem. Generally, it consist on finding the optimal value of a function which measures the goodness of the partition, such as the graph-cut objectives introduced in the previous chapter. However, the most popular one is the *modularity function* introduced by Girvan and Newman in 2004 [34]. Its general formulation is as follows

$$Q = \frac{1}{2n} \sum_{ij} (W_{ij} - P_{ij}) \delta(C_i, C_j) \tag{3.3}$$

where n is the number of edges of the network, the sum runs over all pairs of nodes i and j, W_{ij} is the element of the adjacency matrix, P_{ij} is the null model term and in the Kronecker delta at the end C_i and C_j indicate the communities of i and j. The matrix P represents the *null model* derived from averaging randomized versions of the graph in such way that we preserve some of its features. Hence, the modularity function measures how different is the original graph, W, from its randomized version, P. A widely spread choice of the null model is $P_{ij} = d_i d_j / 2m$ where d_i and d_j denotes the already defined degree from nodes i and j and corresponds to the expected number of edges between the pair if the network would be assembled again. Therefore, such choice preserve, on average, the degree of each node. It yields to the classic form

$$Q = \frac{1}{2n} \sum_{ij} (W_{ij} - P_{ij}) \delta(C_i, C_j) \tag{3.4}$$

Modularity maximization is, unfortunately, an NP-hard problem [8] so for sufficiently large networks we can just hope to find optimal approximations.

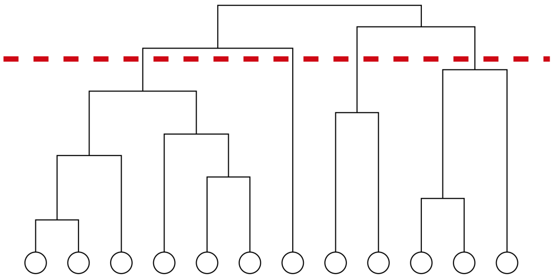


Figure 3.5: Dendrogram representing the hierarchical algorithms type of output. In the bottom the circles represent the individual nodes and as we move upwards they connect forming larger communities. The horizontal red line denotes the optimal split. .

One of the most popular methods for approximating Q is the Louvain algorithm [7] which performs a greedy optimization in a hierarchical fashion. At first every node i

is given its own community. Then the change of modularity is evaluated by removing i from its own community and merging into the community of each neighbor j. The combination with higher modularity change is then executed. This process is repeated for each node until no improvement of the modularity is possible. In second phase, representing next level of hierarchy, a smaller network is built where each community from first phase is converted into a node. Then weighted links between the communities are created and the first stage can be applied again. The algorithm finish when the level of higher modularity is obtained.

Two major advantage of hierarchical are the computational speed which make it easy to escalate for big networks and the ability to find automatically the best number of partitions. On the other side, these methods tent to give sub-optimal solutions.

3.4 Dynamics Based Methods

Communities can also be detected by running dynamical processes in the network. This include diffusion processes, random-walk dynamics, synchronization, etc. The Girvan-Newman (GN) algorithm [34] is one of such kind, concretely it is based in the already introduced *betweenness* measure. If you recall from chapter 1, the idea of betweenness is to estimate a fraction of all possible walks that pass through a node/edge, which can be understood as the amount of information flowing in that specific node/edge. Edges connecting two communities generally have higher betweenness as walks from one community to another have less available path options which produce a bottleneck effect. For example, in the *Barbell-graph* from Figure 3.6, any walk across communities is forced to pass through the middle edge which translates into a the high edge-betweenness from Figure 3.7.

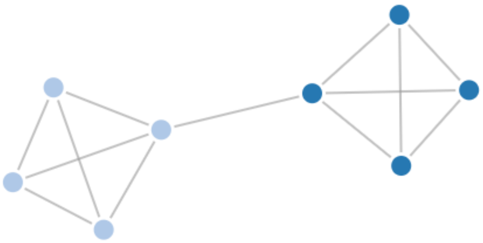


Figure 3.6: A barbell-graph.

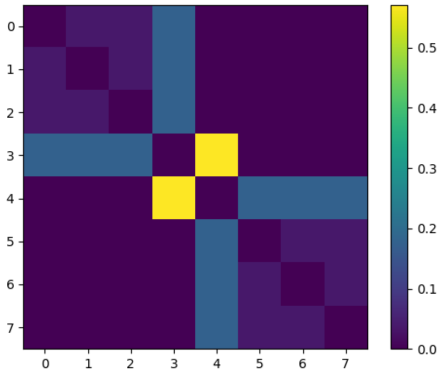


Figure 3.7: Edge betweenness.

In the GN algorithm, the graph is divided by cutting the edges with highest betweenness. Then this is repeated iteratively in the remaining subgraphs until no edges are left. In each subdivision the modularity is evaluated and the one with highest value is selected. We can enumerate its steps as follow:

1. *Calculate betweenness scores for all edges in the network.*
2. *Find the edge with the highest score and remove it from the network.*
3. *Recalculate betweenness for all remaining edges.*
4. *Repeat from step 2.*

CHAPTER 4 Semidefinite Relaxation

In the Background chapter we have already introduced one possible relaxation of the *balanced min-cut* problem. In this chapter we introduce another kind of relaxation, the semidefinite relaxation (SDR). The core idea is to transform a non-convex Quadratically Constrained Quadratic Program (QCQP) into a convex problem by expanding the feasible domain. Convex problems are easier to solve by most of the available solvers. Additionally, in a convex problem if we find an optimal, it is guaranteed to be the global optima and not just local. We will introduce SDR by with the Schur relaxation of the max-cut problem.

4.1 Schur Relaxation of QCQPs

Recall how the formulation of the *max-cut* in (2.5) leads to a NP-hard combinatorial problem. We can introduce the variable $Y = xx^T$

$$\begin{aligned} \max_x \quad & \frac{1}{4}\sum_{i=1}^n\sum_{j=1}^n w_{ij} - \sum_{i=1}^n\sum_{j=1}^n w_{ij}y_{ij}) \\ \text{s.t} \quad & x_i^2 = 1 \quad \text{for } i = 1, ..., n \\ \text{s.t} \quad & Y = xx^T \end{aligned} \tag{4.1}$$

which by knowing that $\text{tr}(WY) = \sum_{i=1}^n \sum_{j=1}^n w_{ij}y_{ij}$ and that Y is positive semidefinite matrix of rank 1, one can further simplify by

$$\begin{aligned} \max_x \quad & \frac{1}{4}\sum_{i=1}^n\sum_{j=1}^n w_{ij} - \text{tr}(WY) \\ \text{s.t} \quad & Y_{ii} = 1 \quad \text{for } i = 1, ..., n \\ & Y \succeq 0 \\ & \text{rank}(Y) = 1 \end{aligned} \tag{4.2}$$

At this point the problem (4.2) and (2.5) are equivalent and still NP-hard. However, now it is easy to identify the bottleneck. The rank constraint. While the rest of the constraints are convex respect to Y, the latter is non-convex. So if we drop the constraint, the problem is relaxed as

$$\begin{aligned}
\max_{x} \quad & \frac{1}{4}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij} - \operatorname{tr}(WY) \\
\text{s.t} \quad & Y_{ii} = 1 \quad \text{for } i = 1, ..., n \\
& Y \succeq 0
\end{aligned} \tag{4.3}$$

This is the so called SDR since the problem (4.3) is an specific instance of an SDP. The new convex formulation allows us to solve the problem numerically in an efficient manner, with the standard available solvers such as interior-point methods. The only thing left to do is to unbundle the vector solution to the QCQP (2.5) from $Y = xx^T$. If by chance the solution would be rank 1, then the first eigenvector of Y would be the the global solution to 4.5. This is rarely the case, so lower rank approximations and rounding is one way to extract feasible solutions. Randomization is another example, but we will cover with more details each method in the following section.

It is reasonable to wonder how good the solution of the approximation is to the real solution. Goeman and Williamson proved that with randomization the relaxed solution gives the tight upper bound of

$$0.87856\text{RELAX} \leq \text{MAXCUT} \leq \text{RELAX}$$

Which means that the relaxed solution is guarantee to be in the worst case, 12%, worst than the global solution, which is remarkable for a NP-hard problem.

4.2 Extracting Labels from SDR Solutions

As already mentioned, once the SDP is solved the solution to the vector solution for the quadratic non-convex problem has to be extracted, i.e., we have to undo the change of variables $X = xx^T$. In this section, we will explain three different methods.

4.2.1 Rank-1 Approximation

Let's assume that we got lucky and the solution of our SDP satisfy the dropped constraint $rank(X) = 1$, meaning that we got the optimal solution. At this point recovering x is straight forward as the singular value decomposition of a rank-1 matrix contain a single non-zero vector

$$X = v\sigma v^T$$

and vector $x = \sqrt{\sigma}v$ would satisfy $X = xx^T$. However, while rank-1 solutions are rather rare, it might be reasonable to make a rank-1 approximation i.e

$$\begin{aligned}
\min_{x} \quad & ||X - X_p||_F \\
\text{s.t} \quad & \operatorname{rank}(X_p) = 1
\end{aligned}$$

By the Eckart–Young–Mirsky theorem [13], we know that this problem has a analytical solution which is, precisely, the factorization using the first singular value

$$X \simeq X_p = v_1 \sigma_1 v_1^T$$

The only difference now is that the solution $x = \sqrt{\sigma_1}2v_1$ will not satisfy the discreteness constraint. We can solve this by using the sign function which map the entries of a vector to 1 or -1. Therefore the solution would be $\hat{x} = \operatorname{sgn}(v_1)$.

4.2.2 Low Rank Approximation and K-means

One can also argue that a rank-1 approximation would erase a lot of information and higher order rank approximations would be more convenient. Consider the the case of matrix in the Figure 4.1

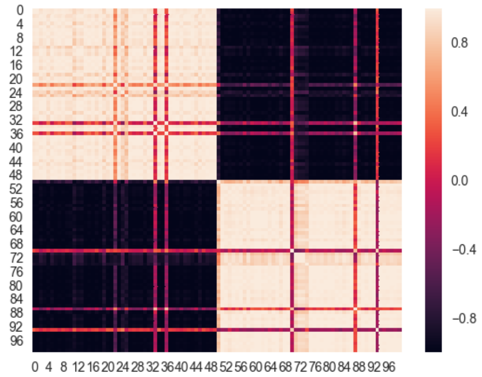


Figure 4.1: Low rank matrix.

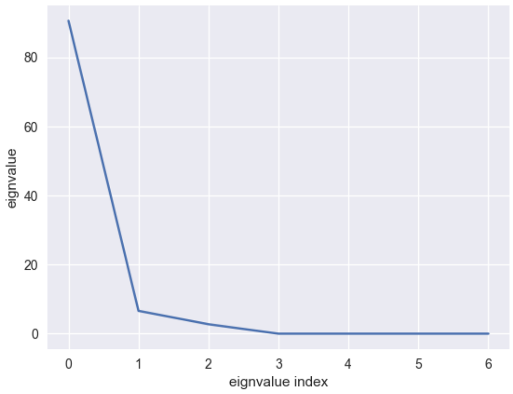


Figure 4.2: Ordered eigenvalues.

By inspecting its eignvalues in Figure 4.2 it is obvious that a rank-3 approximation would be an almost perfect approximation. It's a common approach, once the singular values are ordered, to sequentially add them to the approximation while the difference in magnitude between the current and the next is higher than a threshold. In our example,

$$X \simeq V \Lambda V^T = \begin{bmatrix} v_1 & v_2 & v_3 \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \begin{bmatrix} v_1^T \\ v_2^T \\ v_3^T \end{bmatrix}$$

Unfortunately, $V \in \mathbb{R}^{nx3}$ is not a valid solution as a vector, $x \in \mathbb{R}^n$ is needed. Nonetheless, this can be solved be applying k-means in the same fashion that spectral clustering does. The input data would consist on n samples of 3 dimensions, and for the case of k=2, the output would be a vector $x \in R^n$ with entry values $x_i = \{1, -1\}$.

4.2.3 Randomization

Randomized rounding is a very powerful method for extracting solutions to QCQP from SDR solutions and is the key technique to find upper and lower bounds such as the one of Goeman and Williamson in the max-cut. It can be understood by interpreting the variable X as a covariance matrix from a Gaussian distribution. Recall, the covariance matrix is defined as

$$Cov[x] = E[(x - E[x])(x - E[x])^T] = E[xx^T]$$

which for the case of mean zero it resemble the change of variables $Y = xx^T$. Following this idea, let us consider the stochastic formulation of the SDR max-cut

$$\begin{aligned} \max_{Y \simeq 0, Y \in \mathbb{S}^n} \quad & \mathrm{E}_{\xi \sim \mathcal{N}(0,Y)} \Big[\sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} - \xi W \xi^T \Big] \\ \text{s.t} \quad & \mathrm{E}_{\xi \sim \mathbb{N}(0,Y)} [\xi_i^2] = 1 \quad \text{for } i = 1, ..., n \end{aligned} \tag{4.4}$$

i.e want to find the covariance matrix, Y, that by sampling vectors, ξ, minimize in expectation the original problem while remaining feasible. This is equivalent to the SDR in (4.3). Now we can generate potential solutions by sampling from $(0, X)$ and pick the one which generate lower value after plugging in the objective function. The only problem is that some of the samples might not be feasible and we need to enforce it. For our example,(4.3), a simple $\mathrm{sgn}(x)$ would ensure feasibility of each sample. However, in practice each feasibility enforcing is problem dependent.

```
given a SDR solution, $Y^*$, and N number of randomizations;
for $t$ =1,...$N$ do
    generate sample $\xi_t \sim \mathcal{N}(0, Y)$;
    make sample feasible, e.g: $x_t = \mathrm{sgn}(\xi_t)$;
end
determine $t^* = \arg\max_{t=1,...N}(-x_t W x)$ ;
Result: $\hat{x} = x_{t^*}$ is the approximate solution to the QCQP problem
```

Algorithm 1: Randomized rounding algorithm

4.3 Minimum Cut SDP Relaxation

The SDR for the balanced *min-cut* is not much different. The only difference is that now we have to deal with an additional constraint. For convenience, we can change slightly the formulation in (2.7) to

$$\begin{aligned} \min_{x} \quad & \frac{1}{2} x^T L x \\ \text{s.t} \quad & x_i^2 = 1 \quad \text{for } i = 1, ...n \\ & (\mathbb{1}x)^2 \leq \delta^2 \end{aligned} \tag{4.5}$$

where δ denotes the desired balancing among the groups. For example $\delta^2 = 0$ would result in two groups of exact same size, in the case that n is odd. The SDR is now straight forward

$$\begin{aligned} \min_{Y} \quad & -\operatorname{tr}(CY) \\ \text{s.t} \quad & Y_{ii} = 1 \quad \text{for } i = 1, ..., n \\ & \operatorname{tr}(\mathbb{1}\mathbb{1}^T Y) \leq \delta^2 \\ & Y \succeq 0 \end{aligned} \tag{4.6}$$

where $Y = xx^T$. This leads to what we will call the *balanced-min-cut-SDP* formulation. And now, by using one of the available methods for rounding, one can obtain a approximate solution for the QCQP, (4.5).

In recent developments on semidefinite programming for graph cuts, Bandeira et al. [4] produced the following semidefinite relaxation of (2.7)

$$\begin{aligned} \max_{x} \quad & \operatorname{tr}(BX) \\ \text{s.t} \quad & X_{ii} = 1 \quad \text{for } i = 1, ..., n \\ & X \succeq 0 \end{aligned} \tag{4.7}$$

where $X = xx^T$ and

$$B = \begin{cases} b_{ij} = 1 & \text{if } e = \{i, j\} \in E \\ b_{ij} = -1 & \text{otherwise} \end{cases}$$

Basically, this formulation comes from integrating the balancing constraint into the objective function as one would do when constructing the Lagrangian function. And it can be proven that (4.7) is equivalent to

$$\begin{aligned} \max_{x} \quad & \operatorname{tr}((W - \alpha\mathbb{1}\mathbb{1}^T)X) \\ \text{s.t} \quad & X_{ii} = 1 \quad \text{for } i = 1, ..., n \\ & X \succeq 0 \end{aligned} \tag{4.8}$$

for some choice of α. In the same way (4.6) and (4.8) are equivalent for some choice of δ. The latter formulation, which we will call *regularized-min-cut-SDP*, has the advantage that by using randomization with such objective function we bias the samples towards feasible solutions while in the *balanced-min-cut* formulation most of the samples would not satisfy the balancing constraint. On the other hand, it becomes hard to match the regularization parameter, α, with the group balancing, while in the *balanced-min-cut-SDP* it is straight forward. In practice, most of the available solvers compute both the primal and dual problems, so one could solve the *balanced-min-cut-SDP* and use the *regularized-cut-SDP* objective to sample.

4.3.1 Uncertainty Interpretation of Randomized Rounding

One of the major advantages of the randomized rounding is the uncertainty interpretation of the SDP solution. Since we are considering X as the covariance estimation to the stochastic maximization problem such as (4.4) we have a way to measure how tight the estimation is. If for example the solution X is rank-1, all the samples would be exactly the same. The lower the rank of the estimation, the more confident we would be about our estimation.

4.4 Augmented Adjacency Matrix

For now we have perform graph-cuts based on just the adjacency matrix information, i.e just the direct connections between the nodes. But we are not restricted to just this family of similarity matrices. In practice, we can use any matrix which penalize clustering together two nodes of different communities and enhance the ones of same community. In other words, we want a similarity matrix $S \in \mathbb{R}^{nxn}$ which entries s_{ij} are high when $x_i = x_j$ and low when $x_i \neq x_j$.

$$f(x) = \sum_{i=1}^{n} \sum_{j=1}^{n} s_{ij} x_i x_j$$

4.4.1 Communicability

We have already talk previously about another kind of similarity matrix, the communicability matrix, C. In practice a densely connected cluster would be highly communicated as well, as paths would be more often within clusters than across clusters. Estrada already proposed a communicability based community detection [14], which consist in some kind of spectral decomposition of the communicability matrix, followed by K-means (same procedure as Spectral clustering). This idea can be easily extended to the SDP formulation.

Recall that the communicability matrix is an infinite weighted sum of the adjacency matrix power. Also recall, that for a unweighted and undirected graph the entries of A^n

$$(W^n)_{ij} = \text{walks of length n between node i and j}$$

One might wonder which dumping weight would enhance separability of the clusters . Intuitively, shorter paths would be more often within clusters than longer ones. Once that walks are long enough would connect any pair of nodes indifferently of the community that they belong. A simple visual inspection on the Figure 4.3 confirms the intuition, as the the higher order powers lead to more homogeneous matrices. Fortunately, both

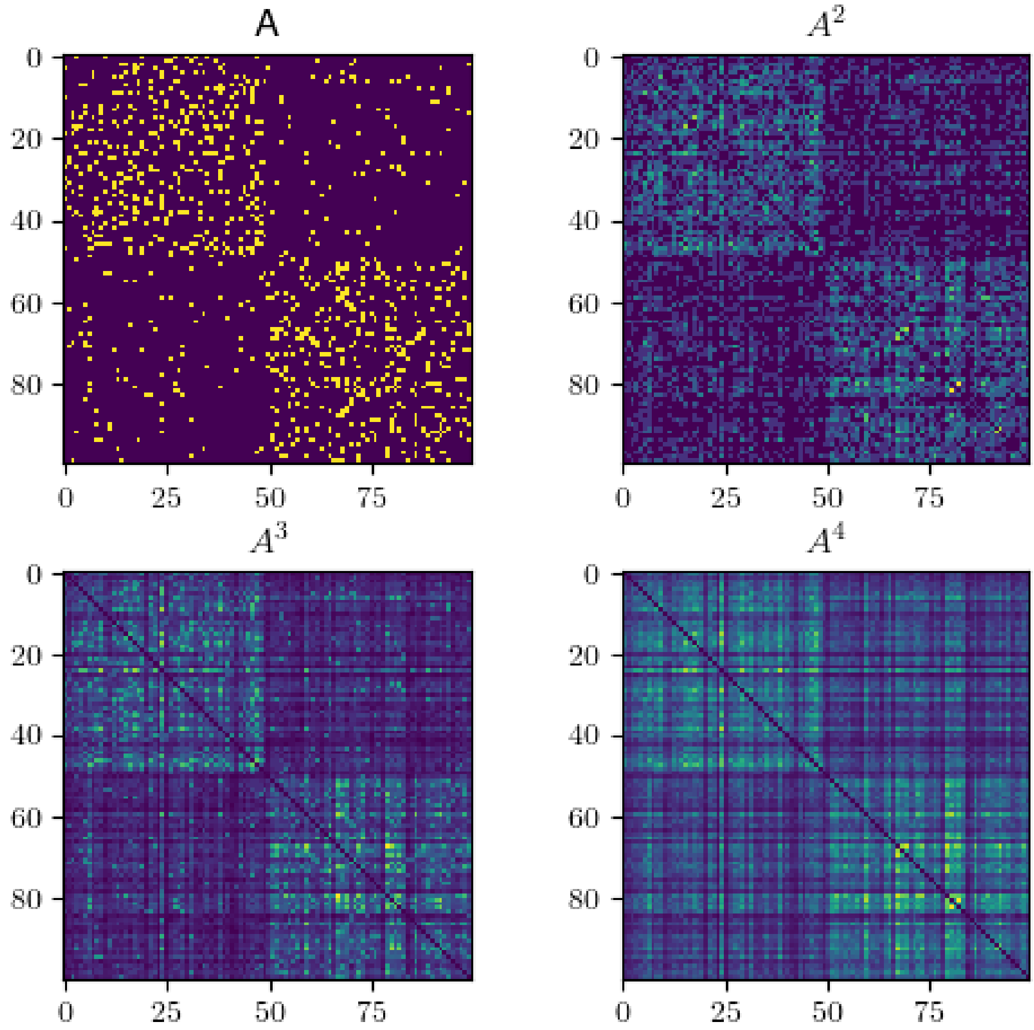


Figure 4.3: Matrix powers of the adjacency matrix.

definitions of communicability discussed in previous chapters enhance the shorter paths. This are some examples of *enhanced* adjacency matrix based on random walks:

- $C = \exp(\frac{W}{\max(\text{eig}(W))})$
- $C = (I - sW)^{-1}$
- $C = W + \alpha W^2$
- $C = W + \alpha W^2 + \gamma W^3$

where α and γ are weighting coefficients. The first term corresponds to the normalized Communicability matrix based in the exponential matrix. The second term is the communicability based in the Resolvent and finally, the last two terms are the adjacent matrix in addition of walks of length 2 and 3.

4.4.2 Shortest-Distance

Considering longer connections among nodes is one way to exploit topological information of the network, but is not the only one. Another interesting property from graphs with communities is that the distance between nodes of the same groups are generally shorter than to other group nodes. The intuition is that a walk between nodes of the same community has much more available paths making easier to reach, while intercluster walks would have to first reach a node which is connected to the other group and then make its way to the desired node.

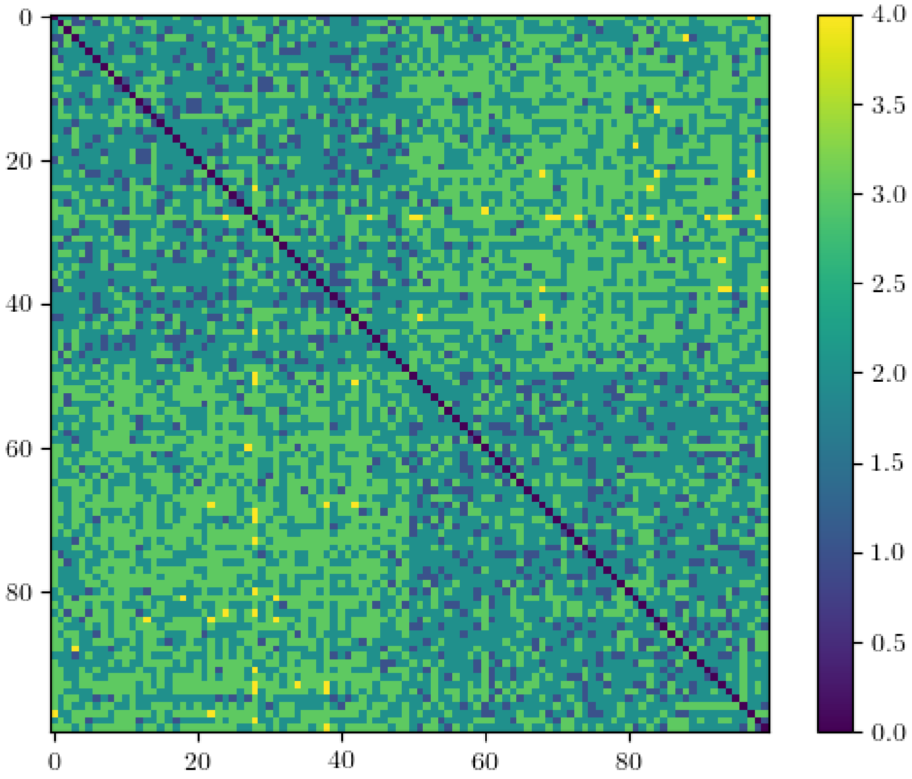


Figure 4.4: Distance matrix.

We can visually confirm the hypothesis in the Figure 4.4 where the distance matrix, $D \in \mathbb{R}^{nxn}$, of the graph from Figure 4.3 is represented, which entries are defined as

$$d_{ij} := \text{distance between node i and j}$$

We could utilize this information in order to construct a new similarity matrix S defined as

$$S = \begin{cases} s_{ij} = 1 & \text{if } w_{i,j} = 1 \\ s_{ij} = -d_{ij} & \text{if } w_{i,j} = 0 \end{cases} \tag{4.9}$$

Plugging such matrix into (4.6) would encourage to cluster together nodes that are connected and discourage the ones that are far away. As we already explained none of this new augmented adjacency matrix would break the group balancing when sampling with the randomized rounding, as we would just reject those ones.

CHAPTER 5
Experimental results

In order to test the performance and robustness of the Semidefinite formulation we will run the algorithms on both synthetic and real networks. Furthermore, we will compare with some of the methods mentioned in the state of the art chapter and we will combine the SDP formulation with different augmented adjacency matrices. This mean that we will solve the *balanced-cut-SDP* (4.6) substituting the matrix in the objective by:

- *SDP-commu*: normalized communicability matrix $C = \exp(W/\max(eig(W)))$
- *SDP-distance*: the similarity matrix based in the distance from (4.9)
- *SDP-distance-W2*: a combination of the similarity matrix (4.9) based on the distance and the walks of length 2 $B = S + W^2/\max(W^2)$

5.1 Synthetic Generated Datasets

Access to ground truth labels is one of the advantages of using computer-generated networks. Many available real datasets also count with annotations. However, with synthetics model, we can control the difficulty of the community recovery. Concretely, the SBM shows a sharp phase transition where the problem becomes unsolvable. Bandeira et al. [1] calculated, both theoretically and computationally, such limit for the special case of $k = 2$ and $p_1 = p_2$ where p_1 and p_2 denotes the internal linking probability of the first and second community respectively. If we recall, the SBM is parametrized by the number of nodes n, the affinity matrix ρ and the group assignment distribution γ. For our generated model we will choose

$$n = 100 \quad \gamma = [0.5, 0.5] \quad \rho = \begin{bmatrix} p & q \\ q & p \end{bmatrix}$$

and using Bandeira's results we will define three different scenarios in ascendant difficulty

- Easy scenario, "*easy-SBM*": $p = 0.15$ and $q = 0.05$
- Challenging scenario, "*diff-SBM*": $p = 0.13$ and $q = 0.05$
- Extreme scenario, "*extr-SBM*": $p = 0.11$ and $q = 0.05$

Notice that we do not have to decrease too much probability p, the only parameter that changes, in order to complicate the problem considerably. This is due to the aforementioned sharp threshold of the SBM. In order to be able to generalize about the results we assemble, with the specified parameters, twenty different networks for each scenario. In the Figure 5.1 one can observe an assemble for each case. Nodes from 0 to 50 correspond to the first community and the rest to the second community. It can be appreciated how the group linkage loose density.

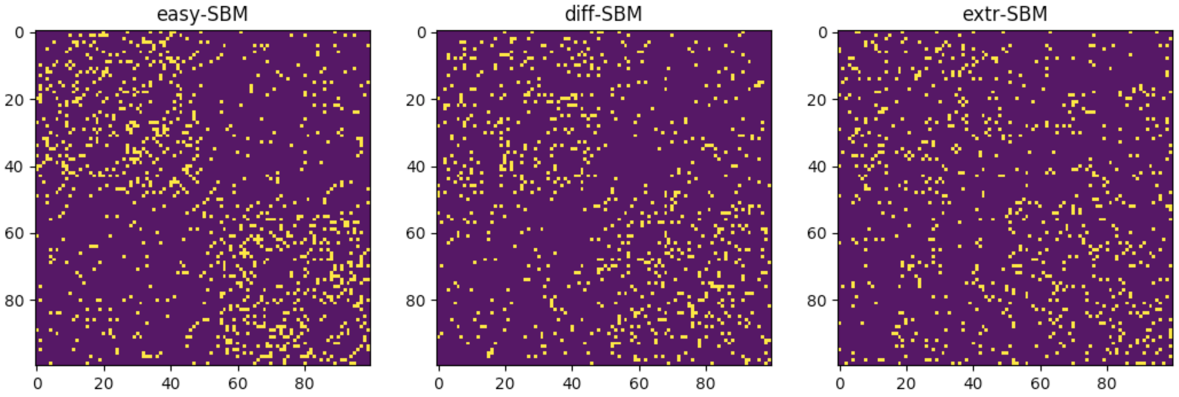


Figure 5.1: Adjacency matrices of the three defined scenarios for the SBM.

We use the same parameter setting for defining different scenarios of the DC-SBM. Additionally, the individual node scaling θ is defined generated as $\theta_u \sim 3 \times Poisson(0.1)$ which will enhance the degree of only a few nodes by a factor of 3. Again, one of the twenty assemble for each case can be visualized in the Figure 5.2.

- Easy scenario, "*easy-DCSBM*": $p = 0.02$ and $q = 0.0035$
- Challenging scenario, "*diff-DCSBM*": $p = 0.02$ and $q = 0.07$
- Extreme scenario, "*extr-DCSBM*": $p = 0.02$ and $q = 0.085$

One can observe, in the results of Table 5.1 how the SDP formulation obtains much better results than the rest of the algorithms. In particular, the hierarchical methods, completely fail to recover any community even for the less challenging scenario. The spectral method is still able to produce decent results, but still inferior to the SDP methods. Nonetheless, these are very impressive results considering the computational speed of Spectral Clustering. Ultimately, one can verify how the augmented adjacency matrices enhance the accuracy of the general *balanced-cut-SDP*.

5.2 Real Datasets

In 1970 the anthropologist and computer scientist, Wayne W. Zachary, studied for 3 years the social dynamics of a local Karate club. Wayne created a network based on

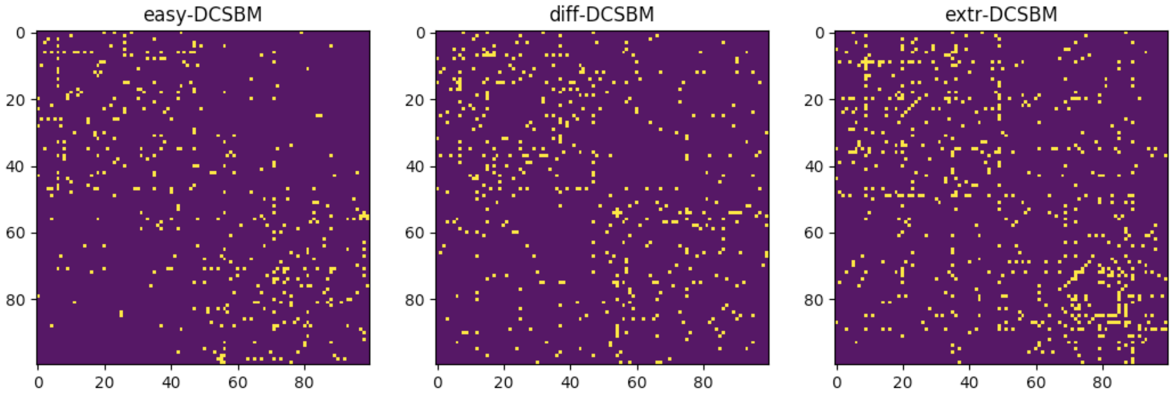


Figure 5.2: Adjacency matrices of the three defined scenarios for the DC-SBM.

	SBM			DC-SBM		
	easy	diff	extr	easy	diff	extr
Louvain	0.51	0.51	0.512	0.51	0.51	0.51
Girvan-Newman	0.529	0.51	0.51	0.6645	0.519	0.51
Spectral-clustering	0.926	0.82	0.6735	0.794	0.6275	0.522
SDP	0.9595	0.836	0.693	0.9145	0.7165	0.539
SDP-comm	**0.96**	0.8355	0.7095	0.913	**0.736**	0.54
SDP-distance	0.948	0.839	**0.71**	0.91	0.715	0.5575
SDP-distance-W2	0.9465	**0.846**	0.703	**0.916**	0.7095	**0.5645**

Table 5.1: Average accuracy of 20 trials per scenario.

the pairwise social interactions of the 34 members outside of the club. After a while, a conflict between the club administrator and the instructor arose and the club split. Wayne was able to predict to who each of the members supported, except one [47].

For some reason, after Newman used this dataset in his seminal paper from 2004 [34] it became very popular and it has been used extensively by the researchers in community detection. In fact, the dataset became so popular that network scientist created the Zachary's Karate club trophy for the researchers who used the dataset as an example at conferences on networks. The club network is shown in the Figure 5.3.

The second analyzed dataset consist of a dolphin community living in Doubtful Sound, New Zealand. The ecology researchers created a network of 62 dolphins based in the number of times that they were sighted together. The network can be visualized in 5.4.

Access to the underlying model or process that generated these networks is impossible.

Instead of ground truth data, we have to rely on metrics that, for our understanding, measure the goodness of the partitions. One will be the modularity as defined in 3.4. Another metric will be the silhouette index. For each node, i, clustered in the community the C_k, the silhouette is defined as

$$s(i) = \frac{\bar{b}_{min} - \bar{b}_{i,C_k}}{\max(\bar{b}_{min}, \bar{b}_{i,C_k})}$$

where $\bar{b}_{i,C_k}$ is the average distance between node i and all the nodes of its community C_k and $\bar{b}_{min}$ is the $\bar{b}_{i,C_j}$ for every other community j. The metric is bounded as $-1 \leq s(i) \leq 1$ where 1 means *well-clustered*. The silhouette index of a community is the average over its nodes and the silhouette of the whole network is the average over all the communities.

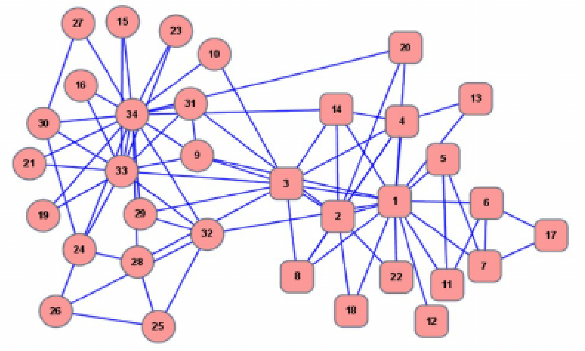


Figure 5.3: Karate-club network.

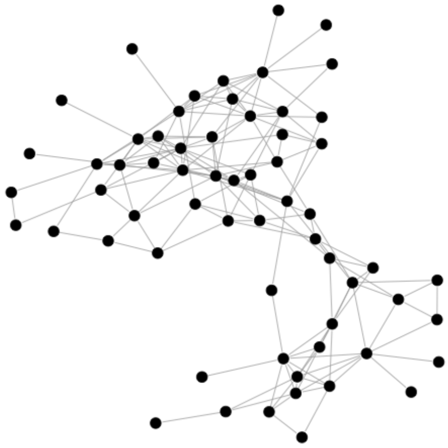


Figure 5.4: The dolphins network.

	Karate-Club		Dolphins network	
	Q	S	Q	S
Louvain	0.359	**0.345**	-0.001	-0.167
Girvan-Newman	0.359	**0.345**	0.378	**0.446**
Spectral-clustering	0.151	0.190	0.337	0.321
SDP	**0.371**	0.328	**0.401**	0.307
SDP-comm	**0.371**	0.328	**0.401**	0.307
SDP-distance	**0.371**	0.328	**0.401**	0.307
SDP-distance-W2	**0.371**	0.328	**0.401**	0.307

Table 5.2: Modularity and Silhouette for the Karate and Dolphin networks.

The results (Table 5.2) show how the SDP formulations achieve the best modularity score in both networks. This makes sense as the objective optimized is somehow related to the modularity. Both functions favor communities with high internal linkage. However, although the SDP methods achieve high Silhouette scores, the Girvan-Newman algorithm is better in the two cases. The gap difference is especially very acute in the Dolphin network. If we look at its topology these results make sense as a bottleneck effect is produced in the middle making the betweenness based method very suitable.

These results underly the importance of how we define a community and its correspondence goodness functions. I believe that a community is better represented by the modularity the SDP is the best choice. If contrary we rely on the silhouette, the GN method is the best in this case.

CHAPTER 6

Conclusions

In the present thesis, we have achieved the following goals:

- Shown the connections between graph theory and linear algebra
- We have made a thorough analysis of the state of the art on community detection summarizing the most popular method and showing connections across them.
- We have explained the powerful method of relaxing optimization problems and how they can approximate the real problems. Specifically, we have shown how semidefinite programming is a very suitable method for the community detection problem.
- We have shown how already existing SDP formulations for the community detection problem can be reformulated in order to control the groups' size.
- We purposed several modified adjacency matrices, as input for the SDP formulations, which enhance separability of the problem.
- Finally, from the experimental results, we showed the importance of how different goodness metrics lead to different partitions.

As future work, we have opened a window for enhancing the SDP formulations by augmented adjacency matrices. We have just proposed two different types, but many other options and combinations are available to explore. Additionally, the formulation can be extended to admit more than one cut and exploiting sparsity is definitely possible in many cases.

Bibliography

[1] E. Abbe, A. S. Bandeira, and G. Hall. “Exact Recovery in the Stochastic Block Model”. In: *IEEE Transactions on Information Theory* 62.1 (January 2016), pages 471–487. ISSN: 0018-9448. DOI: `10.1109/TIT.2015.2490670`.

[2] E. Abbe and C. Sandon. “Community Detection in General Stochastic Block models: Fundamental Limits and Efficient Algorithms for Recovery”. In: *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*. October 2015, pages 670–688. DOI: `10.1109/FOCS.2015.47`.

[3] Emmanuel Abbe. “Community Detection and Stochastic Block Models: Recent Developments”. In: *Journal of Machine Learning Research* 18 (2017), 177:1–177:86.

[4] Emmanuel Abbe et al. “Decoding binary node labels from censored edge measurements: Phase transition and efficient recovery”. In: *CoRR* abs/1404.4749 (2014). arXiv: `1404.4749`. URL: `http://arxiv.org/abs/1404.4749`.

[5] Naman Agarwal et al. “Multisection in the Stochastic Block Model using Semidefinite Programming”. In: *CoRR* abs/1507.02323 (2015).

[6] Albert-László Barabási and Réka Albert. “Emergence of Scaling in Random Networks”. In: *Science* 286.5439 (1999), pages 509–512. ISSN: 0036-8075. DOI: `10.1126/science.286.5439.509`. eprint: `http://science.sciencemag.org/content/286/5439/509.full.pdf`. URL: `http://science.sciencemag.org/content/286/5439/509`.

[7] Vincent D Blondel et al. “Fast unfolding of communities in large networks”. In: *Journal of Statistical Mechanics: Theory and Experiment* 2008.10 (2008), P10008. URL: `http://stacks.iop.org/1742-5468/2008/i=10/a=P10008`.

[8] U. Brandes et al. *Maximizing Modularity is hard.* cite arxiv:physics/0608255 Comment: 10 pages, 1 figure. 2006. URL: `http://arxiv.org/abs/physics/0608255`.

[9] Sergey Brin and Lawrence Page. “The Anatomy of a Large-scale Hypertextual Web Search Engine”. In: *Comput. Netw. ISDN Syst.* 30.1-7 (April 1998), pages 107–117. ISSN: 0169-7552. DOI: `10.1016/S0169-7552(98)00110-X`. URL: `http://dx.doi.org/10.1016/S0169-7552(98)00110-X`.

[10] Etienne CÃŽme and Pierre Latouche. “Model selection and clustering in stochastic block models based on the exact integrated complete data likelihood”. In: *Statistical Modelling* 15.6 (2015), pages 564–589. DOI: `10.1177/1471082X15577017`. eprint: `https://doi.org/10.1177/1471082X15577017`. URL: `https://doi.org/10.1177/1471082X15577017`.

[11] Jingchun Chen and Bo Yuan. “Detecting functional modules in the yeast protein-protein interaction network”. In: *Bioinformatics* 22 18 (2006), pages 2283–90.

[12] Melissa Cline et al. “Integration of biological networks and gene expression data using Cytoscape”. In: 2 (September 2007), pages 2366–2382.

[13] Carl Eckart and Gale Young. “The approximation of one matrix by another of lower rank”. In: *Psychometrika* 1.3 (September 1936), pages 211–218. ISSN: 1860-0980. DOI: `10.1007/BF02288367`. URL: `https://doi.org/10.1007/BF02288367`.

[14] Ernesto Estrada. “Community detection based on network communicability”. In: 21 (March 2011), page 016103.

[15] Ernesto Estrada and Desmond Higham. “Network properties revealed through matrix functions”. In: *SIAM Review* 52.4 (November 2010), pages 696–714. ISSN: 0036-1445. DOI: `10.1137/090761070`.

[16] Santo Fortunato and Darko Hric. “Community detection in networks: A user guide”. In: *CoRR* abs/1608.00163 (2016).

[17] A. Gadde et al. “Active learning for community detection in stochastic block models”. In: *2016 IEEE International Symposium on Information Theory (ISIT)*. July 2016, pages 1889–1893.

[18] Michel X. Goemans and David P. Williamson. “Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming”. In: *J. ACM* 42.6 (November 1995), pages 1115–1145. ISSN: 0004-5411. DOI: `10.1145/227683.227684`. URL: `http://doi.acm.org/10.1145/227683.227684`.

[19] Anna Goldenberg et al. *A Survey of Statistical Network Models*.

[20] Peter D. Grnwald, In Jae Myung, and Mark A. Pitt. *Advances in Minimum Description Length: Theory and Applications (Neural Information Processing)*. The MIT Press, 2005. ISBN: 0262072629.

[21] StÃ©phane Helleringer and Hans-Peter Kohler. “Sexual Network Structure and the Spread of HIV in Africa: Evidence from Likoma Island, Malawi”. In: 21 (December 2007), pages 2323–32.

[22] Adel Javanmard, Andrea Montanari, and Federico Ricci-Tersenghi. “Phase Transitions in Semidefinite Relaxations”. In: (2016).

[23] Richard M. Karp. “Reducibility among Combinatorial Problems”. In: *Complexity of Computer Computations: Proceedings of a symposium on the Complexity of Computer Computations, held March 20–22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, and sponsored by the Office of Naval Research, Mathematics Program, IBM World Trade Corporation, and the IBM Research Mathematical Sciences Department.* Edited by Raymond E. Miller, James W. Thatcher, and Jean D. Bohlinger. Boston, MA: Springer US, 1972, pages 85–103. ISBN: 978-1-4684-2001-2. DOI: 10.1007/978-1-4684-2001-2_9. URL: https://doi.org/10.1007/978-1-4684-2001-2_9.

[24] Brian Karrer and M.E.J. Newman. “Stochastic blockmodels and community structure in networks”. In: 83 (January 2011), page 016107.

[25] Leo Katz. “A new status index derived from sociometric analysis”. In: *Psychometrika* 18.1 (March 1953), pages 39–43. ISSN: 1860-0980. DOI: 10.1007/BF02289026. URL: https://doi.org/10.1007/BF02289026.

[26] Ravi Kumar et al. “Trawling the Web for Emerging Cyber-communities”. In: *Comput. Netw.* 31.11-16 (May 1999), pages 1481–1493. ISSN: 1389-1286. DOI: 10.1016/S1389-1286(99)00040-7. URL: https://doi.org/10.1016/S1389-1286(99)00040-7.

[27] Pierre Latouche, Etienne BirmelÃ©, and Christophe Ambroise. “Variational Bayesian Inference and Complexity Control for Stochastic Block Models”. In: 12 (December 2009).

[28] G. Linden, B. Smith, and J. York. “Amazon.com recommendations: item-to-item collaborative filtering”. In: *IEEE Internet Computing* 7.1 (January 2003), pages 76–80. ISSN: 1089-7801. DOI: 10.1109/MIC.2003.1167344.

[29] Ulrike von Luxburg. “A tutorial on spectral clustering”. In: *Statistics and Computing* 17.4 (December 2007), pages 395–416. ISSN: 1573-1375. DOI: 10.1007/s11222-007-9033-z. URL: https://doi.org/10.1007/s11222-007-9033-z.

[30] J. MacQueen. “Some methods for classification and analysis of multivariate observations”. In: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics.* Berkeley, Calif.: University of California Press, 1967, pages 281–297. URL: https://projecteuclid.org/euclid.bsmsp/1200512992.

[31] Edward M. Marcotte et al. “Detecting protein function and protein-protein interactions from genome sequences”. English (US). In: *Science* 285.5428 (July 1999), pages 751–753. ISSN: 0036-8075. DOI: 10.1126/science.285.5428.751.

[32] Bojan Mohar. “The Laplacian spectrum of graphs”. In: *Graph Theory, Combinatorics, and Applications.* Wiley, 1991, pages 871–898.

[33] Radford M. Neal and Geoffrey E. Hinton. “Learning in Graphical Models”. In: edited by Michael I. Jordan. Cambridge, MA, USA: MIT Press, 1999. Chapter A View of the EM Algorithm That Justifies Incremental, Sparse, and Other Variants, pages 355–368. ISBN: 0-262-60032-3. URL: http://dl.acm.org/citation.cfm?id=308574.308679.

[34] M. E. J. Newman and M. Girvan. “Finding and evaluating community structure in networks”. In: *Phys. Rev. E* 69 (2 February 2004), page 026113. DOI: 10.1103/PhysRevE.69.026113. URL: https://link.aps.org/doi/10.1103/PhysRevE.69.026113.

[35] M. E. J. Newman, D. J. Watts, and S. H. Strogatz. “Random graph models of social networks”. In: *Proceedings of the National Academy of Sciences* 99.suppl 1 (2002), pages 2566–2572. ISSN: 0027-8424. DOI: 10.1073/pnas.012582999. eprint: http://www.pnas.org/content/99/suppl_1/2566.full.pdf. URL: http://www.pnas.org/content/99/suppl_1/2566.

[36] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. “On Spectral Clustering: Analysis and an Algorithm”. In: *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*. NIPS’01. Vancouver, British Columbia, Canada: MIT Press, 2001, pages 849–856. URL: http://dl.acm.org/citation.cfm?id=2980539.2980649.

[37] Tiago P Peixoto. “Reconstructing networks with unknown and heterogeneous errors”. In: *arXiv preprint arXiv:1806.07956* (2018).

[38] Sven Peyer, Dieter Rautenbach, and Jens Vygen. “A Generalization of Dijkstra’s Shortest Path Algorithm with Applications to VLSI Routing”. In: *J. of Discrete Algorithms* 7.4 (December 2009), pages 377–390. ISSN: 1570-8667. DOI: 10.1016/j.jda.2007.08.003. URL: http://dx.doi.org/10.1016/j.jda.2007.08.003.

[39] Shaghayegh Sahebi and William Cohen. “Community-Based Recommendations: a Solution to the Cold Start Problem”. In: *Workshop on Recommender Systems and the Social Web (RSWEB), held in conjunction with ACM RecSys?11*. October 2011. URL: http://d-scholarship.pitt.edu/13328/.

[40] Larry S. Shapiro and Michael Brady. “Feature-based correspondence: an eigenvector approach”. In: *Image Vision Comput.* 10 (1992), pages 283–288.

[41] Jianbo Shi and Jitendra Malik. “Normalized Cuts and Image Segmentation”. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 22.8 (August 2000), pages 888–905. ISSN: 0162-8828. DOI: 10.1109/34.868688. URL: https://doi.org/10.1109/34.868688.

[42] Mechthild Stoer and Frank Wagner. “A Simple Min-cut Algorithm”. In: *J. ACM* 44.4 (July 1997), pages 585–591. ISSN: 0004-5411. DOI: 10.1145/263867.263872. URL: http://doi.acm.org/10.1145/263867.263872.

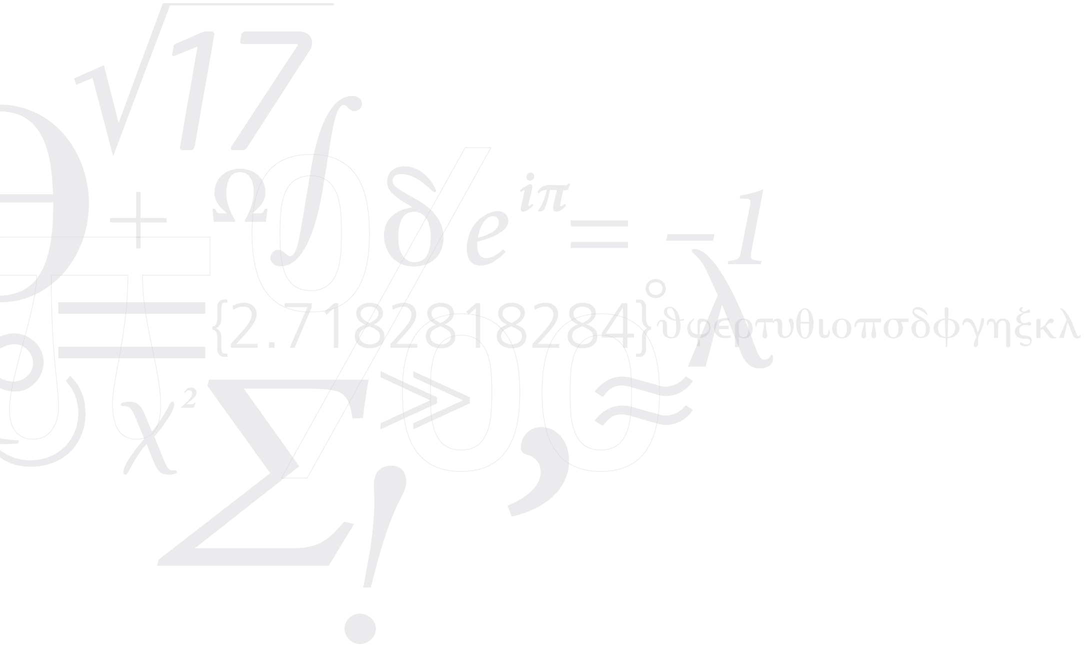
[43] Dorothea Wagner and Frank Wagner. “Between Min Cut and Graph Bisection”. In: *Mathematical Foundations of Computer Science 1993*. Edited by Andrzej M. Borzyszkowski and Stefan Sokołowski. Berlin, Heidelberg: Springer Berlin Heidelberg, 1993, pages 744–750. ISBN: 978-3-540-47927-7.

[44] Y. X. Rachel Wang and Peter J. Bickel. “Likelihood-based model selection for stochastic block models”. In: *Ann. Statist.* 45.2 (April 2017), pages 500–528. DOI: 10.1214/16-AOS1457. URL: https://doi.org/10.1214/16-AOS1457.

[45] Rui Wu et al. “Clustering and Inference From Pairwise Comparisons”. In: *SIGMETRICS Perform. Eval. Rev.* 43.1 (June 2015), pages 449–450. ISSN: 0163-5999. DOI: 10.1145/2796314.2745887. URL: http://doi.acm.org/10.1145/2796314.2745887.

[46] Xiaoran Yan et al. “Model selection for degree-corrected block models”. In: *Journal of Statistical Mechanics: Theory and Experiment* 2014.5 (2014), P05007. URL: http://stacks.iop.org/1742-5468/2014/i=5/a=P05007.

[47] W.W. Zachary. “An information flow model for conflict and fission in small groups”. In: *Journal of Anthropological Research* 33 (1977), pages 452–473.


$\sqrt{17}$
$e^{i\pi} = -1$
{2.7182818284}
ϑφερτυθιοπσδφγηξκλ