2015

Community Detection Detailed for Online Social Networks

Christopher J. Hogan
Wilfrid Laurier University, hoga1470@mylaurier.ca

Community Detection Detailed for Online Social Networks

by

Christopher Hogan

Masters of Science in Mathematics

Submitted to the Department of Mathematics
in partial fulfillment of the requirements for
Masters of Science in Mathematics
Wilfrid Laurier Univertsity

Contents

1 Introduction

1.1 Motivation and Outline

Ever since the internet became publicly available it has allowed users to interact with each other across virtual networks. In particular, development allows individuals to post, trade and discuss ideas to the point where a virtual presence, or even an avatar, can feel like a necessity. As of today these interactions happen almost anywhere. Some examples are people's email addresses, Facebook pages, LinkedIn profiles, Twitter accounts, etc. Although there exists many more examples of such virtual places they all have the same effect as their real world equivalent: communities. Even in virtual spaces, we as humans have tendencies to form connections to other people or their avatars. Although that trait alone is not unexpected, virtual networks open the doors to so many sciences as every piece of information is written down and recorded at some point. We see this in the large studies that take place today, that use information such as polling data, the recommended circles of buyers or friends, or simply workplace correspondences[22]. The huge amount of data that flows through online communities is often referred to as big data and has been the highlight of many recent studies. Particularly in this boom of information growth a vast field of research has emerged and has seen significant attention with this data now available. To be able to identify the circles of users that more frequently interact with each other enables us to understand how influence, knowledge, and even happiness, flow through a social network. Finding these groups has been known as Community Detection in Social Network Analysis. It provides an approach to a real world problem with immediate application. This has never been more needed, especially with more and more of society being incorporated into a virtual world where high amounts of data is being recorded. The purpose of this thesis is to provide a

Community Detection algorithm particularly for finding these communities within online social networks.

For the remainder of Chapter 1 we discuss the necessary basic mathematical concepts needed for reviewing the methods currently used in Community Detection. Chapter 2 covers the findings and recent research in Social Network Analysis that are relevant to finding online communities. Chapter 3 details the specific community detection tool that we will use in our own model. In Chapter 4 we present our own model using many of tools of community detection but also introducing some of our own findings. Chapter 5 shows how the implementation of our method maintains a necessary lower computation time. Chapter 6 compares our result to the commonly used algorithms and details our overall performance. In Chapter 7 we review our method, findings, what contributions were made to this field of research and what foreseeable improvements can be immediately applied.

1.2 Mathematical Background

This section provides the necessary basic mathematical background and the various notation used in prior methods of community detection and that we incorporate into our own method. We cover the basics of graph theory as it provides a necessary mathematical representation of the network, a small amount of linear algebra that plays a large part of our algorithm, and an introduction to run-time complexity.

1.2.1 Graph Theory

A **graph** $G = (V, E)$ consists of a non-empty set of points V of points called **vertices** and set of unordered pairs of vertices E called edges. A graph is sometimes called an undirected graph. For an edge, $e = (u, v) \in E$, the vertices u and v are called the **ends** of e. For each edge of a graph it can be said that the edge *connects* the vertices or the vertices are *adjacent* to each other. An edge is a loop if it connects a vertex to itself, $e = (v, v)$. An edge that has the same ends as another edge is called a **multiple edge**. A graph with no multiple edges or loops is called **simple**. Undirected graphs can be represented as a point for each vertex of a the graph and as a line between points for each edge, not necessarily straight.

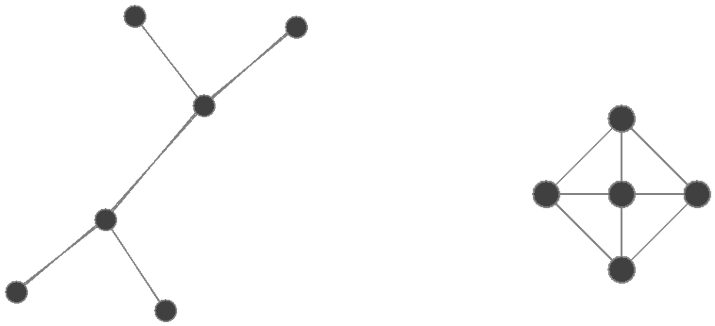
A **directed graph** $G = (V, E)$ is a graph where the edges E are ordered pairs of vertices. Then an edge $e = (u, v) \in E$ indicates an edge from u to v where u is the tail of the edge and v is the head of the edge. A way to represent directed graphs as a point for each vertex and an arrow for each edge that starts at the tail and points to the head.

A **path** in a graph (or directed graph) is an alternating sequence of vertices and edges, $v_0, e_1, v_1, e_2, v_2, \ldots, e_p, v_p$ where v_i are distinct vertices and e_i indicates that v_{i-1} is adjacent to v_i. A **directed path** is path where edge, e_i, indicate an edge from v_{i-1} to v_i exist. A graph is **connected** if there exists a path between every two vertices. A graph that is not connected is **disconnected**.

The **complement** of graph G is a graph H where H has the same vertex set as G and the vertices of H are only adjacent if they were not adjacent in G.

In this thesis we adopt the notation that orders our set of vertices $V = \{1, 2, \ldots |V|\}$. Then a vertex v_i refers to the vertex at the ith element of V. We denote $e = (v_i, v_j)$ as ij where in an undirected graph vertices v_i and v_j are adjacent to each other, and in directed graph there exists an edge from v_i to v_j. Thorough out this paper we refer to many undirected graphs and will often represent them as in Figure 1.

A **subgraph**, $H = (V', E')$, of a graph, $G = (V, E)$, is graph with vertices that are a subset of G, $V' \subseteq V$ and edges that are a subset of the edges of G, $E' \subseteq E$. An **induced** or **full** subgraph, $H = (V', H')$, of graph $G = (V, E)$ has a vertex set



(a) Graph A

(b) Graph B

Figure 1: Two simple undirected graphs.

that is strictly a subset of the vertex set of G, $V' \subset V$, and every edge of E' meeting with the vertex set V' in H. A **connected component** of a graph is subgraph that is connected.

The **order** of a graph is the number of vertices, $|V|$, within a graph, often known as n. The **size** of a graph is the number edges, $|E|$, within the graph, often known as m. The **degree** of vertex, v, is the number of edges connecting to, or adjacent to v noted as $deg(v)$ or $d(v)$. In this work we adapt the notation of the degree of v_i as k_i to align our work with previous community detection works and many of the definitions in Chapter 3. Similarly for directed graphs, we say a vertex has an **outgoing** degree, k_i^{out}, and **incoming** degree, k_i^{in}, indicating the number of edges coming from a vertex and coming into vertex respectively. A **weighted** graph has a weighting function $w(ij)$ on the edges ij where $w(ij) \in \mathbf{R}$. In a weighted graph the **strength** of a vertex, s_i, is defined to be the sum of weights of adjacent edges. The *size* of a weighted graph is the sum of weights of all edges W. Note that by the *First*

Fundamental Theorem of Graph Theory the following hold true,

$$\begin{aligned}
&\sum_i k_i = 2m, \text{ where G is an undirected graph,} \\
&\sum_i k_i^{in} = \sum_i k_i^{out} = m, \text{ where G is a directed graph,} \\
&\sum_i s_i = 2W, \text{ where G is a weighted graph.}
\end{aligned} \tag{1.1}$$

Graphs are often represented by their adjacency matrices. The **adjacency matrix** of a graph, $A(G)$ or $\mathbf{A}$, is an $n \times n$ matrix where $A_{ij} = 1$ if there exists an edge from v_i to v_j and 0 otherwise. Similarly the weighted adjacency matrix $W(G)$ contains the strengths of the edges $W_{ij} = s_{ij}$ from v_i to v_j. The degree sequence of graph, d_G, is a monotonic nonincreasing sequence of the vertex degrees. The graph A from Figure 1 has the following adjacency matrix and degree sequence,

$$A(G) = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \text{ and } d_G = (3,3,1,1,1,1) \tag{1.2}$$

While Graph B from Figure 1 has the following adjacency matrix and degree sequence,

$$A(G) = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix} \text{ and } d_G = (4, 3, 3, 3, 3) \qquad (1.3)$$

A **Cut Edge** of a graph is an edge of a graph that when removed causes the graph to become disconnected. A set of edges that when removed causes the graph to become disconnected is known as **cut set**.

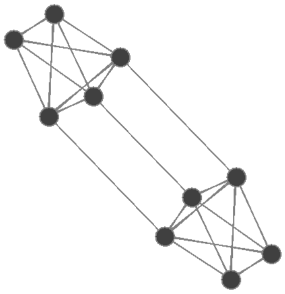


Figure 2: A graph with where 3 edges make a cut set

Lastly, there are many types of graphs relevant to this area of research and well studied in the past. A graph is a *clique* if every vertex is connected to every other vertex. Figure 2 has two subgraphs that are cliques with 5 vertices. These are called 5-cliques. A q-clan refers to a subset of vertices that are all reachable from to each other within q edges. Large q-clans often indicate a highly dense group of vertices and are commonly used in social network analysis [26].

1.2.2 Linear Algebra

Often in mathematics data is represented in an $n \times n$ matrix, such as the adjacency matrix of a graph. A matrix, A, is a diagonal matrix if for $A_{ij} = 0$ if $i \neq j$. A matrix, A, is **symmetric** if $A_{ij} = A_{ji}$ for all ij. The adjacency matrix of an undirected graph is symmetric. If A is composed of real values and is symmetric then A has a decomposition $A = Q \Lambda Q^T$, where the columns of Q are the known as the eigenvectors and Λ is an diagonal matrix where Λ_{ii} is corresponding eigenvalue. This decomposition is known as the *eigendecomposition*.

1.3 Complexity Class

A large determining factor for which algorithms to use on a network is their efficiency. Often in computer science the analysis of an algorithm is determined by how it will run with an arbitrary large size input. For this purpose we introduce *Big* O *Notation.* Big O notation is the comparison the growth of an algorithm, $f(n)$, to another function $g(n)$ as a way of determining whether functions are equivalent. We say $f(n) = \mathrm{O}(g(n))$ if for some n_0 and all $n \geq n_0$ then there exist $C \in R$ such that $|f(n)| \leq C|g(n)|$. For example if we consider $f(n) = 4n^2 - n + 7$ then as n get arbitrarily large it is clear that the largest term is the n^2. Then where $g(n) = n^2$ then for any $C > 4$ there will be corresponding n_0 where $f(n) \leq Cg(n)$ is always true. Similarly we define Ω where $f(n) = \Omega(g(n))$ if for some n_0 and all $n \geq n_0$, there exists C such that $|f(n)| \geq C|g(n)|$. Then following our example, for any $C < 4$ there exists a corresponding n_0 such that $|f(n)| > C|g(n)|$. If a function, $f(n)$, is both $\mathrm{O}(g(n))$ and $\Omega(g(n))$ then we say $f(n) = \Theta(g(n))$. We use Θ notation to indicate what the efficiency or running time of function is. In our example, our function $f(n) = \Theta(n^2)$. This becomes key in our algorithm for finding social networks since with large data, although a function with run time n^3 may produce a more accurate result, such an algorithm would be infeasible to consider when n is large and an n^2 algorithm is available.

2 Social Network Analysis and Online Communities

2.1 Introduction

The origins of Social Networks Analysis is ambiguous, however it has been prevalent field of research since Social Networks Analysis received specific attention in the 1930's [26]. During the 1930's researchers devised the concept of a sociogram, a graphical network representation in order to further understand communities [26]. Later in the 1960's, researcher began to first apply mathematical concepts to these networks and develop today's social network analysis [26]. In this chapter we review these concepts of social networks and introduce specific considerations when developing methods of community detection within social networks with particular relevance towards online communities.

2.2 Properties of a Social Network

Social Network Analysis is the specific field of sociology and anthropology where network theory is applied to social networks. Network theory uses graphs as representation of real world systems [26]. Then for social network analysis a graph is constructed by creating a vertex for each individual, commonly referred to as an actor. The possible relations and interactions between the actors are represented by edges between the vertices of the respective actors. For social media networks it is not uncommon to expand these definitions such that content items, such as blog posts or comments belonging to an actor, are included as vertices and additional edges are included to represent various relationships. For the purpose of this thesis we will only consider the simple graph scenario where the network is $G = (V, E)$, in which

V represents actors and E represents a level of social interactions between the two actors. This provides a basic structure that can be built upon in further works as needed.

These social networks are known to share a number of characteristics common to real networks, such as the small world effect and power law distributions [10]. Of these characteristics, some real networks are known to exhibit the concept of *community structure* within the network. Community structure refers to the fact that the actors of a network belong to groups that have high levels of interaction within their own groups and comparably fewer interactions with other groups. That is, networks may be partitioned into groups of vertices that contain more edges between vertices of the same group than with vertices of other groups. There are various concepts for how community structure appears in complex networks, such as n-cliques and n-clans [22]. Examples of this property are easily visible within social network such as social interactions that occur on Facebook between friends, or a person's email and their correspondence with others.

The concept of community structure has various properties, depending on the complex network it is embedded in. The community structure within a social network will often contain many of the following properties: hierarchical, overlapping, and elements of the multi-tier structure. To detect communities it is necessary to consider how these properties manifest within social networks as well as the drawbacks they present on community detection algorithms. To do this we detail each of the properties below. A network is *hierarchical* when communities of smaller sizes can be grouped together to represent larger communities resulting in communities on varying scales of sizes. In social network this is a well-known attribute and can be easily seen as any formal organization of a social system [27, 31]. Uncovering hierarchical networks is problematic as it becomes difficult to return meaningful partitions that correctly identify the communities. Some of the pitfalls of a detecting communities in a hierarchical network include separating a community into smaller communities that should be considered as one or, grouping communities together when they should be separate. A network is *overlapping* when actors can be considered members of multiple communities. Social networks comprised of social interactions are rarely ever completely composed of disjoint communities, especially as social networks are considered to be small-world networks thus the average path length between any two actors is small, known often to be log(the number of actors). Often actors can represent roles that bridge any number of communities, thus belonging partially to all of them. Unique and special consideration is needed in order to uncover overlapping networks as, unlike many other complex networks, they can not always be discretely partitioned. A multitiered network may have communities of various sizes. This presents a significant disadvantage as many earlier algorithms for detecting communities rely on the a priori information of the sizes of the clusters as well as what mesoscopic level they exist on.

The properties of social networks also impose necessary considerations for community detection. It is often the case that the information of how many communities there are within a complex network can easily reduce the difficulty of discerning the communities within the network; however for real social networks this a priori information is often unavailable as they may be composed of any number of communities. Algorithms that find communities of varying sizes as well as any number of communities often require a unique approach to succeed without having a priori knowledge.

Social networks are also often known to be dynamic and often changing as time of goes on. That is, new actors appear and new relations occur, while other actors will disappear and older relations may fade away from the network. Communities that appear in a dynamic community structure change in size and potentially merge together, while other will fade, spilt, or disappear.

2.3 Social Brain Hypothesis and Community Size

The Social Brain Hypothesis presented by R.I.M. Dunbar in 1993 is a notable theory we use to clarify some of the properties of community structure presented in the prior section. The Social Brain Hypothesis theorized that the brain size of the neocortex of nonhuman primates covaries to the size of the group that they organized themselves into [4]. Further studies built upon the Social Brain Hypothesis indicated hunter-gatherer societies have a similar relation, that given the size of a human's neocortex, communities that form contain up to 147.8 individuals. This number is usually rounded to 150 and known as Dunbar's number. The 95% confidence interval for this number is 100.2-231.1. Dunbar's number has received significant attention by researchers seeking to provide an accurate estimate of real world communities of humans beyond hunter-gatherer societies. Furthermore the work of W. -X. Zhou, D.Sornette, R. A. Hill and R.I.M. Dunbar provides a organization of hierarchal structure that occur within human communities [31]. The authors found from their studies that communities scale, on average, from smallest, S_1, to largest, S_6, as $S_1 = 4.6$, $S_2 = 14.3$, $S_3 = 42.6$, $S_4 = 132.5$, $S_5 = 566.6$ and $S_6 = 1728$ [31]. The authors provide range bands for each of these average community size. The smaller community, found to be between 3-5 individuals, represents people from whom one would seek personal advice or turn to in times of emotional and financial stress [31]. The second smallest community, that ranges from 12-20 individuals, is referred often as the 'sympathy group' or 'support clique'. These are individuals who are close to a person and that may be contacted at least once a month [6, 13]. The third group, ranging from 30-50 individuals, are bands that are often represented in hunter-gatherer societies. These individuals are the most likely to change over time but are always drawn from the 150 individuals a person knowns [4]. Larger groups that are over 500 people can be thought of as tribes or identity communities that are not necessary dense in social interactions, but in which all the actors belong to a common identity [31, 24].

Since the development of online communities enables any person to connect to hundreds, if not thousands of others, one can ask whether the social brain hypothesis is still accurate in suggesting communities are limited to at most 150 individuals, especially when it is possible for anyone to have over 500 friends on Facebook. This question is addressed by R.I.M. Dunbar, who illustrates that communication between social network sites do not in fact increase the capacity to have more friends [5]. The ease of communication allows for the representation of the 500 plus common identity-based communities to become visible, but these acquaintances do not actually increase the number of friends anyone is capable of having. In fact it comes to light that social networks mostly allow for any person to remain in contact with friends that are geographically separated and simply allow the person to maintain these relations [5]. This can be seen as noise, or interference, that an online social network may possess, and we consider this unique problem in a later section. Lastly, the social brain hypothesis has been validated already within social network of Twitter where authors B. Gonçalves, N. Perra, A. Vespignani provided a model that exemplifies Dunbar's number within the a online community [11]. This study was one of the first to bring Dunbar's number to online communities and big data. From these results we build our own model on the premise that Dunbar's number influences what communities we expect to find in our data.

2.4 Noise and Edge weighting

A particular issue for any dataset drawn from real world data is the level of noise that appears. For social network analysis, noise can be seen not only from the various forms of data sampling of interactions but also from actors within the network themselves. As indicated earlier, it is normal within social networks for actors to claim more friends than they actual have, or for inconsistencies to appear in the relationships between online social networks and the real communities outside of the network [5, 9]. Many methods and procedures have been developed to eliminate noise within a network and to more clearly show community structure [28]. The most applicable and widely used means to eliminate noise in social networks is to consider applying a weighting function to a network such that each edge is given a weight that best represents the various types of relationships and interactions. [9, 29, 3]. That is, for each pair of users, consider the interactions and evaluate the strength of their relation. One of the first large developments for understanding the relationship between actors of a network comes from M. Granovetter's ideas of the Strength of Weak Ties [12]. In which Granovetter proposed that flow communication between networks, especially the flow between weak ties, could be incredibly beneficial. In the case of Social Network Analysis and Community Detection, we expect that the internal strength of communications will be stronger *within* a community rather than *between* communities. Without evaluating the strength of a relationship, a social network would not be able to distinguish between varying relationships and would result in noise as not all relationships are equal. This leads to degradation in the overall performance, and in the finding of communities, as data may become too dense or contain arbitrary thresholding [29]. To evaluate these relationships a recent study done by E. Gilbert and K. Karahalios conducted a comprehensive model of network interactions on subset of users of Facebook [9]. This resulted in an algorithm that is based on learning how a large set of actors interact with one another and then using this knowledge, to properly evaluate relationships strength, which, in turn, can be used to weight the edges within a social network. The results generated were compared to a survey each user completed that indicated the strengths of their own relationship with friends they had on Facebook. This comparison indicated the algorithm proved successful. To further develop the understanding of relationships, methods of Opinion Mining and Sentiment Analysis can be used to understand the text between actors and their opinion of one another which can be added to any model [21]. For these reasons we provide details in future chapters of how to adapt our model of weighted symmetric networks.

In dynamic social networks the strength of relationship over time is the driving force for change in a network. As edges become weaker, the communities in the network may split; conversely, as relationships becomes stronger, communities may become larger and merge together. This change in the network represents the decay of relationships and the loss of friends, as well as the making of new friends and forming relationships with others. This trend supports the prior section of the social brain hypothesis by indicating that as a person makes new friends, the strength of the ties to the older friends may decrease, and since they can only support a certain number of friends overall, they must replace older friends.

3 Graph clustering and Modularity of Social Networks

3.1 Introduction

The problem of clustering groups in a graph has been a very popular problem in recent years. Clustering groups has many real world applications resulting in the interest of many interdisciplinary fields of research. Detecting communities through community structure is used to find naturally forming groups of actors, represented by vertices, that have more interactions, represented by edges, inside the group that they belong to than with the rest of the network. These groups of vertices are commonly referred to as *clusters*, c. In the context of social network analysis we also refer to these clusters as *communities.*

A **partition**, P of a graph, G, is set of clusters where each vertex is assigned to one cluster. A **covering**, $\mathcal{C}$, is set of clusters where each vertex belongs to at minimum one cluster but may belong to multiple. Many of the definitions will be presented for discovering and evaluating partitions and then later are expanded for coverings.

For successful detection of viable partitions it is necessary that the graph, and thus the network, must be sparse. That is, if the size of the network is n and the number of relations in the network, m, then $n \gg m$, where $\gg$ indicates that the number of edges is significantly smaller than the number of vertices. In community detection we expect that as our communities we use grow in size that $m \in \mathrm{O}(n)$, otherwise more applicable methods would be in the field of research of data clustering[7]. Real networks are known to be sparse and and social networks are expected to be sparse.

Since recent research in community structure has been extensive we will only be reviewing necessary concepts and algorithms relating to the development of our model of detection: see [7] for an relatively recent extensive review of community detection and [23] for explicit community detection for social media. In this section we will be review the necessary ideas to provide a foundation for the understanding of how algorithms are developed for Graph Clustering.

3.2 Community Evaluation: Modularity

Although communities can appear and be thought of as cliques and clans, for online networks, these structures are difficult to find and there is no current polynomial-time algorithm. Although many algorithms exist to roughly find these structures, they often can be too restrictive on defining what a social network's community structure is [23]. **Weak** communities and **Strong** communities terms were introduce to relax some of these constraints. A community is said to weak if it has overall more internal interactions than external. A community is said to be strong if every actor of a community has more internal interactions than external. Many quality functions have been proposed to determine whether a partition of a network is an accurate representation of communities. Of these quality functions, *density*, *conductance*, and *modularity* are popular definitions and provide an easy introduction to this material as each have unique, but relatively simple, ways of defining communities that is not restricted to particular graph structures. We further illustrate which has the most applicable properties for finding communities in online social networks. For these definition, let $G = (V, E)$, denote a graph, G, with vertex set, V, where $|V| = n$, and edge-set, E, where $|E| = m$. Let $A(G)$, or A, donate the adjacency matrix of G, where A_{ij} is 1 if there is an edge, ij, between vertices v_i and v_j and 0 otherwise. Let P be a partition composed of clusters $c_1, ..., c_{|P|}$. For partitions we define a vertex,

v_i's, corresponding cluster as c_i.

We say an edge is an *internal edge*, also known as a *intra-cluster edge*, of a partition if both ends of the edge belong to the same cluster. An edge of a covering is an internal edge with respect to a cluster if both ends partial belong to that cluster. An edge is an *external edge*, also known as *inter-cluster edge*, of a partition if the ends of the edge belong to different clusters. An edge of covering is external with respect to a cluster c if one end does not belong to c.

The measurement of *density* for any cluster of vertices is expressed as *external density* and *internal density*. *Internal density* is the comparison of the total number of internal edges of the partition to the total possible number of internal cluster edges. We then define the number of internal edges as

$$\#\text{ of intra-cluster edges} = \sum_{i,j=1}^{n} A_{ij}\delta(c_i, c_j) \tag{3.1}$$

where $\delta(c_i, c_j)$ is the Dirac delta function

$$\delta(c_i, c_j) = \begin{cases} 1 & \text{if } c_i = c_j \text{ for vertices } v_i \text{ and } v_j \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

Then density of internal edges for a cluster c is expressed as

$$\delta_{int}(c_i) = \frac{\#\text{ of inter-cluster edges of } c}{|c|(|c| - 1)/2}. \tag{3.3}$$

The parameter $\delta_{int}(c_i)$ measures the ratio of the number of edges inside of a clustering c_i to the number of possible edges of c_i which is $n_c(n_c - 1)/2$ [7]. We then measure the external edges as

$$\#\text{ of external edges } = \sum_{i,j=1}^{n} A_{ij}\overline{\delta(c_i, c_j)} \tag{3.4}$$

where $\overline{\delta(c_i, c_j)} = 1 - \delta(c_i, c_j)$. Then similarly external density of cluster, c_i, is measured as

$$\delta_{ext}(c) = \frac{\#\text{ of external edges of } c_i}{(n - |c|)|c|} \tag{3.5}$$

The parameter $\delta_{ext}(c_i)$ measures the ratio of the number of external edges to the maximum number of possible external edges c can have [7]. Density is a simplistic method to determine community structure. We now investigate more intensive measurements.

For a given graph, the *conductance*, or the *normalized metric cut*, is the measurement of splitting a graph with the smallest cut-set. To avoid simply cutting off a single vertex with only a few edges to the rest of the network conductance is measured favouring equally sized communities. Conductance is then a measurement of two subgraphs S and and the vertex induced subgraph $\overline{S} = V - S$. Conductance is then expressed as,

$$\phi(S) = \frac{\sum_{i \in S, j \notin S} A_{ij}}{\min\left(|A(S)|, |A(\overline{S}|)\right)} \tag{3.6}$$

where the $|A(S)| = \sum_{i \in S} \sum_{j \in V} A_{ij}$ [16]. The value of conductance, ϕ, is the ratio between the number of edges between the subset S and $\overline{S}$ compared to the minimum of the size of either S or $\overline{S}$ including the edges between each other. Conductance can be used for community detection for splitting a network into two clusters, favouring clusters of equal sizes.

$Modularity$ is a relatively recent measurement founded by M. Girvan and M.E Newman that has seen a lot of attention for its intuitive design and its accurate results [7]. Modularity relies on a graph with similar properties to the original graph but with absolutely no community structure [7]. This graph is often referred to as the null model, and modularity is the measurement of the difference between a partition of the original graph and the same partition on this null model. The most commonly used null model is a random graph with the same degree sequence as the original graph, also known as the *configuration model* [7]. To formulate the null model consider each edge of the original graph, ij, to be split in two; that two stubs of the edges are created, one stub connected to v_i, and one to v_j. A vertex v_i has k_i stubs. Then to remap an edge from v_i to v_j, where v_i and v_j may not have been adjacent before, consider randomly selecting both a stub from v_i, which has probability $\frac{k_i}{2m}$, and randomly selecting a stub from v_j, which has probability $\frac{k_j}{2m}$. It follows that the probability of remapping any edge is $\frac{k_i k_j}{4m^2}$. Modularity is the difference of the ratio of the internal edges in a partition to the total number subtracted by the expected value of edges in the null model, which is defined as,

$$Q = \frac{1}{2m} \sum_{i,j=1}^{n} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(c_i, c_j) \tag{3.7}$$

Note that the first term of equation (3.7) bounds Q between 1 and -1. To see this we show the following proof:

Proof. When the partition has no edges the proof is trivial as $A_{ij} = \frac{k_i k_j}{2m} = 0$ for all $i, j \in V$. If there exists an edge in the partition then for some $i, j \in V$, $k_i k_j > 0$ and $A_i j = 1$ Then for $Q < 1$

$$\frac{1}{2m} \sum_{i,j=1}^{n} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(c_i, c_j) < \frac{1}{2m} \sum_{i,j=1}^{n} (A_{ij} \delta(c_i, c_j)) \leq \frac{1}{2m} \sum_{i,j=1}^{n} A_{ij} = \frac{2m}{2m} = 1$$

Then for $Q > -1$

$$Q = \frac{1}{2m} \sum_{i,j=1}^{n} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(c_i, c_j) > \sum_{i,j=1}^{n} \left(-\frac{k_i k_j}{4m^2} \delta(c_i, c_j) \right) \geq - \sum_{i,j=1}^{n} \frac{k_i k_j}{4m^2} = -\frac{4m^2}{4m^2} = -1$$

Thus Q is bounded by 1 and -1. □

A positive modularity defines a good partition, as the clusters of the partition have a high number of internal edges where the expected number of edges between vertices is low. We notice that if we consider the whole graph as one cluster, then $\delta(c_i, c_j) = 1$ for all $i, j \in V$ and

$$Q = \sum_{i,j=1}^{n} A_{ij} - \sum_{i,j=1}^{n} \frac{k_i k_j}{2m} = 2m - \frac{\sum_{i}^{n} k_i \sum_{j}^{n} k_j}{2m} = 2m - \frac{(2m)(2m)}{2m} = 0 \qquad (3.8)$$

Then if any partition of the network results in a modularity value less than 0, generally, the network is a dense community, and no split can increase the modularity.

Of these objective functions both conductance and modularity have seen a large amount of attention for developing algorithms. Both have seen the most success at finding communities, but modularity has seen particular success for social networks [16]. Thus in this thesis we expand upon the current modularity objective function. This choice is made as algorithms using conductance often make use of a priori knowledge such as the number of communities and the size of communities. Particularly, conductance favours the partition of clusters of equal size and focuses on determining a minimal cut while modularity provides a measurement for which partitions find module elements of a community and in many cases favours splits of various sizes. This approach better aligns with the difficulties of finding community structure outlined in Chapter 2.

3.3 Properties of Modularity

Since modularity is particularly popular there has been much research to further understand properties of this measurement. We now review the adaptations of modularity for directed or weighted and overlapping networks, the spectral decomposition of modularity, and the well-known resolution limit of modularity.

3.3.1 Adaptations of modularity

Social networks are not always represented as simple unweighted undirected graphs. As in Chapter 2, there are many different interpretation of data for a network [22]. Although we investigate simple unweighted undirected graphs for the purposes of this thesis, here we provide the details and extensions for weighted and directed networks [7].

For directed networks, an equation analogous to (3.7) can be derived as follows. For all vertices, v_i, the edges meeting v_i are broken into outgoing stubs, the number of these is the out degree, k_i^{out} of vertex v_i, and incoming stubs, the number of these is the degree, k_i^{in} of vertex v_i. The null model is created by considering the probability randomly of creating an edge from a vertex, v_i, to another vertex v_j. This is the product of the probability, $\frac{k_i}{m}$, of selecting an outgoing stub of v_i from all other outgoing stubs and the probability, $\frac{k_j}{m}$, of selecting an incoming stub of v_j from all other incoming stubs. Then the probability of creating an edge from v_i to v_j is $\frac{k_i^{out}k_j^{in}}{m^2}$. For directed networks the modularity is defined as

$$Q = \frac{1}{m}\sum_{i,j=1}^{n}\left(A_{ij} - \frac{k_i^{out}k_j^{in}}{m}\right)\delta(c_i, c_j). \tag{3.9}$$

Equation (3.9) is the difference of the of ratio the of internal directed edges compared to all the edges of the graph subtracted by the expected values of directed edges for in appearing in the random graph.

For weighted networks we construct a null model that incorporates the total weights of the edges meeting a vertex. Instead of the degree of vertices affecting our equations it is now the the total strength, s_i, of edges adjacent to a vertex, v_i. As before, then the expected weight between two vertices, v_i and v_j, is $\frac{s_i s_j}{4W^2}$ where W is the the total weight within the network. We then conclude that for a weighted adjacency matrix with entries W_{ij}, modularity can be represented as

$$Q = \frac{1}{2W} \sum_{i,j=1}^{n} \left(W_{ij} - \frac{s_i s_j}{2W} \right) \delta(c_i, c_j) \tag{3.10}$$

Equation (3.10) is the difference of the ratio of internal weights to all weights subtracted by the expected strength between the vertices in the random graph. Without weighting a network noise will be introduced to the network as not all social ties are the same strength [12, 29, 21]. It follows from equation (3.9) and equation (3.10) it that the modularity for a weighted directed graph can be constructed by the directed incoming strengths, s_i^{in}, and outgoing strengths, s_i^{out}. Then, following the same reasoning provided for both prior equations, modularity is defined for directed weighted graphs as follows,

$$Q = \frac{1}{2W} \sum_{i,j=1}^{n} \left(W_{ij} - \frac{s_i^{in} s_j^{out}}{2W} \right) \delta(c_i, c_j) \tag{3.11}$$

With a thorough understanding of weighted and directed modularity, an extension for *overlapping communities* can be constructed. Recall that overlapping communities are represented as a covering. We now follow the work of V. Nicosia, G. Mangioni, V. Carcgiolo and M. Malgeri to show this extension [20]. For overlapping communities it is necessary to introduce a membership vector for each vertex. Let $[\alpha_{i,1}, \alpha_{i,2} \ldots \alpha_{i,|\mathcal{C}|}]$ be the membership vector for a vertex v_i, where $\alpha_{i,c}$ expresses the belonging value of vertex v_i has to a community c. In a partition $\alpha_{v_i,c} = 1$ if and only if $v_i \in c$. However for overlapping communities this constraint is relaxed and the $\alpha_{v_i,c}$ value changes to indicate how much a vertex belongs to a community. It follows that $\alpha_{i,c}$ has the following intuitive properties, with respect to all the clusters c of partition P,

$$0 \leq \alpha_{v_i,c} \leq 1 \quad \forall v_i \in V, \quad \forall c \in P \tag{3.12}$$

and that

$$\sum_{c=1}^{|\mathcal{C}|} \alpha_{v_i,c} = 1 \tag{3.13}$$

This means that a vertex vector can never have total belonging value more than it has in the case of finding a partition.

With each vertex having a belonging coefficient to a cluster c, we now consider the belonging of an edge between them to a cluster c. Let ℓ be the edge between vertices v_i and v_j. Denote the belonging value of ℓ to a cluster c by $\beta_{l,c}$. Then a edge's belonging to a cluster $\beta_{l,c}$ is dependent on the adjacent vertices of the edge and their belonging values to c. V. Nicosia *et al.* define this as [20],

$$\beta_{l,c} = \mathcal{F}(\alpha_{v_i,c}, \alpha_{v_j,c}) \tag{3.14}$$

where $\mathcal{F}$ is some function bounded between 0 and 1 and dependent on $\alpha_{v_i,c}$ and $\alpha_{v_j,c}$. V.Nicosia *et al.* found in their experiments the following two dimensional logistic function gave the best results for overlapping communities [20],

$$\mathcal{F}(\alpha_{i,c}, \alpha_{j,c}) = \frac{1}{\left(1 + e^{-f(\alpha_{i,c})}\right)\left(1 + e^{-f(\alpha_{j,c})}\right)} \tag{3.15}$$

where $f(\alpha_{i,c})$ is a linear scaling function,

$$f(x) = 2px - p, \quad p \in \mathcal{R} \tag{3.16}$$

where x is the membership of a vertex and in our experiments we apply larger value of p for larger communities then scale it down to a minimum of 9. These values of p maintain distinct clusters that do not completely overlap each other.

To integrate these variables into the modularity measurement, consider equation (3.9) rewritten as sum of clusters,

$$Q = \frac{1}{m} \sum_{c=1}^{|\mathcal{C}|} \sum_{i,j=1}^{|c|} \left(\delta(c_i, c_j) A_{ij} - \delta(c_i, c_j) \frac{k_i^{out} k_j^{in}}{m} \right). \tag{3.17}$$

Then by relaxing $\delta(c_i, c_j)$ to variables $r_{ij,c}$ and $s_{ij,c}$, where for partitioning $r_{ij,c} = s_{ij,c} = \delta(c_i, c_j, c)$, we can consider the modularity gain of edges that belong to multiple clusters. Let us consider $r_{\ell,c}$ as the contribution to modularity given by of the edge, ℓ, joining vertices v_i and v_j that belongs in some way to cluster c. Then $r_{\ell,c}$ is called the belonging value of the internal edges of the clusters of our partition. To illustrate this consider equation (3.17) rewritten now with overlapping edges internal edges,

$$Q = \frac{1}{m} \sum_{c=1}^{|\mathcal{C}|} \sum_{i,j=1}^{|c|} \left(r_{ij,c} A_{ij} - s_{ij,c} \frac{k_i^{out} k_j^{in}}{m} \right). \tag{3.18}$$

Note that $s_{ij,c}$ will be defined below. Equation (3.18) illustrates now how the gain of modularity is based on the values of $r_{ij,c}$. However, in a partition the gain is simply dependent on the membership of its vertices (either 1 or 0), and it is under that intuition that the authors defined $r_{ij,c}$ as dependent on the membership of the adjacent vertices; that is to say,

$$r_{ij,c} = \beta_{\ell,c} = \mathcal{F}(\alpha_{i,c}, \alpha_{j,c}). \tag{3.19}$$

The formulation of $s_{ij,c}$ is more complicated. From equation (3.18) the value $s_{ij,c}$ represents the overlapping nature of the null model used to compare against the partition. The difficulty comes in that the null model must preserve the total contribution to any cluster, c. The *configuration model* for the case of clusters that do not overlap involves placing an edge between vertex v_i and any other vertex without any consideration of the vertices adjacent to v_i. That is, the cluster that v_i is a part of has no relation to the probability to that v_i has an edge to any other vertex. This ensures that the null model maintains the degree sequence but contains no community structure. For the similar case of overlapping communities, the belonging, $\alpha_{i,c}$, of vertex v_i to cluster c of does not depend on the belonging, $\alpha_{j,c}$, of any vertex v_j to cluster c. The expected belonging coefficient of any possible edge ij starting from a cluster c is the average of all possible edge belonging coefficients starting from c. This is formulated as,

$$\beta^{out}_{ij,c} = \frac{\sum_{j=1}^{n} \mathcal{F}(\alpha_{i,c}, \alpha_{j,c})}{n}. \tag{3.20}$$

Similarly the expected value of an incoming edge belonging coefficient for a cluster c is the average edges belongings coefficients to c, which is defined as

$$\beta^{in}_{ij,c} = \frac{\sum_{i=1}^{n} \mathcal{F}(\alpha_{i,c}, \alpha_{j,c})}{n}. \tag{3.21}$$

Then our null model becomes

$$s_{ij,c} \frac{k_i^{out} k_j^{in}}{m} = \frac{k_i^{out} \beta^{out}_{ij,c} k_j^{in} \beta^{in}_{ij,c}}{m} \tag{3.22}$$

We can then substitute the coefficient r_{ijc} and s_{ijc} of equation (3.17) as the belonging coefficients of internal edges and the expected belonging of any directed edge in and out of a cluster to construct a measurement of overlapping modularity. V.Nicosia *et al.* [20] define this as,

$$Q = \frac{1}{m}\sum_{c=1}^{|\mathcal{C}|}\sum_{i,j=1}^{|c|}\left(\beta_{ij,c}A_{ij} - \frac{k_i^{out}\beta_{ij,c}^{out}k_j^{in}\beta_{ij,c}^{in}}{m}\right) \tag{3.23}$$

3.3.2 Spectral Decomposition

The spectral decomposition of modularity was presented by M.E.J Newman in 2006 [19]. This method was shown to be more intuitive for finding social network communities than other spectral methods such as applying the *Laplacian matrix* [18]. In fact, regular spectral methods excel at optimizing conductance and finding optimal splits of a network into two subgraphs with a minimum number of cut edges, known as the well-established field of graph partitioning [22, 19]. However this is not always ideal for a social network that has more than two various communities and communities that vary in size [19]. For this reason we will use the spectral decomposition of modularity. To introduce how to optimize modularity with the eigenvectors of matrices we follow the same steps presented by M.E.J Newman. For spectral analysis we initially consider the modularity of a simple network or graph as in equation (3.7). Note that, however, $\delta(c_i, c_j)$ may be represented as value $\frac{1}{2}(s_is_j+1)$ where $s_i \cdot s_j$ are 1 if i and j belong to the same group or -1 if they belong to different groups (not to be confused with strength indicators from equation (3.10)).

$$\frac{1}{4m}\sum_{i,j=1}^{n}\left(A_{ij} - \frac{k_ik_j}{2m}(s_is_j+1)\right) = \frac{1}{4m}\sum_{i,j}^{n}\left(A_{ij} - \frac{k_ik_j}{2m}(s_is_j)\right) \tag{3.24}$$

where (3.24) is reduced by following the property presented in equation (3.8). This result can be rewritten in the following matrix form,

$$Q = \frac{1}{4m}\mathbf{s}^T\mathbf{B}\mathbf{s}. \tag{3.25}$$

where $\mathbf{B}$ is the symmetric matrix and

$$B_{ij} = A_{ij} - \frac{k_i k_j}{2m} \tag{3.26}$$

$\mathbf{B}$ is known as the *modularity matrix.* We then consider $\mathbf{s}$ as a linear combination of the the normalized eigenvectors $\mathbf{v}_i$ of the *modularity matrix.* Then $\mathbf{s}$ can be expressed as

$$\mathbf{s} = \sum_{i}^{n} a_i \mathbf{v}_i \tag{3.27}$$

where $a_i = \sum_{i=1}^{n} \mathbf{v}_i^T s_i$ Then equation (3.25) becomes

$$Q = \sum_{i=1}^{n}\left(a_i \mathbf{v}_i^T\right)\mathbf{B}\sum_{j=1}^{n}\left(a_j \mathbf{v}_j\right) = \sum_{i,j=1}^{n}\left(a_i a_j \lambda_j \delta(\mathbf{v}_i, \mathbf{v}_j)\right) = \sum_{i=1}^{n}\left(a_i^2 \lambda_i\right) \tag{3.28}$$

where λ_i is the eigenvalue corresponding to the eigenvector $\mathbf{v}_i$. Equation (3.28) makes use of the fact that $\mathbf{v}_j^T \mathbf{v}_j = \delta(\mathbf{v}_i, \mathbf{v}_j)$. Then if we assume that the eigenvalues have been ordered such that $\lambda_1 > \lambda_2 \cdots > \lambda_n$, the largest modularity split would come from choosing $\mathbf{s}$ to be as parallel to the first eigenvector $\mathbf{v}_i^1$ as possible. That is, by constraints of the construction, the most parallel choice is

$$s_i = \begin{cases} +1 & \text{if } \mathbf{v}_i^1 > 0 \\ -1 & \text{if } \mathbf{v}_i^1 < 0 \end{cases} \tag{3.29}$$

This solution for **s** of equation 3.29 is an approximate optimal split for considering the network along one vector. However real communities often have more than just two communities. This procedure is illustrated by M. E. J. Newman but follows closely what has been shown for the case of the Laplacian and other spectral methods, shown by C. J. Alpert S. -Z. Yao [19, 1]. Let **S** be an $n \times |P|$ index matrix such that each column represents a community. Then the values of **S** are defined to be

$$S_{ij} = \begin{cases} 1 & \text{if vertex } v_i \text{ belongs to community } c_j \\ 0 & \text{otherwise} \end{cases} \tag{3.30}$$

The columns of **S** must be mutually orthogonal in the construction of the spectral space. Then in a manner similar to (3.28) the δ function can be replaced as,

$$\delta(c_i, c_j) = \sum_{c=1}^{|P|} S_{i,c} S_{jc} \tag{3.31}$$

The modularity of a partition of a network is

$$Q = \frac{1}{2m} \sum_{i,j=1}^{n} \sum_{c=1}^{|P|} B_{ij} S_{ic} S_{jc} = \frac{1}{2m} \operatorname{tr}(\mathbf{S}^T \mathbf{B} \mathbf{S}). \tag{3.32}$$

This equation can then be rewritten using the eigenvalues, λ_i, and eigenvectors, **a**, where we removed the unit eigenvector $\mathbf{v}_i$, of the modularity matrix as

$$Q = \sum_{i=1}^{n} \sum_{c=1}^{|P|} (s_{ic} a_i)^2 \lambda_i, \tag{3.33}$$

where the $\frac{1}{2m}$ is suppressed as it does not change the position of an optimal partition. Similar to ordinary spectral partitioning [1], *vertex vectors* are introduced as,

$$[r_i] = \sqrt{(\lambda_i - \alpha)} a_i \tag{3.34}$$

where $i = 1 \dots n$ and $\alpha < \lambda_n$ to ensure that all $[r_i]_j$ remain real. We can then redefine modularity from equation (3.33) in terms of vertex vectors as

$$\begin{aligned} Q &= n\alpha + \sum_{i=1}^{n} \sum_{c=1}^{|P|} (s_{ik} a_i)^2 (\lambda_i - \alpha) \\ &= n\alpha + \sum_{i,j=1}^{n} \sum_{c=1}^{|P|} ([r_i]_j s_{ic})^2 \\ &= n\alpha + \sum_{c=1}^{|P|} \sum_{j=1}^{n} \sum_{i=1}^{|c|} ([r_i]_j)^2 \\ &= n\alpha + \sum_{c=1}^{|P|} |\mathbf{R}_k|^2, \end{aligned} \tag{3.35}$$

where $\mathbf{R}_k$ is known as a *community vector* and is the sum of all vertex vectors of a community. Many algorithms are built from this representation of modularity as it provides a dimensional space equal to the number consisting of the eigenvectors with vertex vectors that represent modularity splits. Often if the network has $|P|$ distinct communities of roughly equal size they will be represented in the first $|P| - 1$ vectors [7].

3.3.3 Resolution of Modularity

In this section we review the the well-known resolution limit of modularity. Although modularity has proved successful in community detection, it does have a resolution limit, which will inaccurately identify communities [7]. To show what the resolution limit is and how it affects the measurement of modularity, we the follow the proof given by S. Fortunato and M. Barthélemy [8]. First consider equation (3.7) as the sum of contribution of each cluster but rewritten as

$$Q = \sum_{c}^{k} \left[\frac{l_c}{m} - \left(\frac{d_c}{2m} \right)^2 \right], \tag{3.36}$$

where k is the number of clusters, l_c is the number of internal edges in a cluster c, and d_c is the total degree of the vertices included in c. Then if the first term of equation (3.36) is larger than the second, there are more edges in the cluster, c, than expected. Thus a cluster c is a modular cluster if

$$\frac{l_c}{m} - \left(\frac{d_c}{2m} \right)^2 > 0. \tag{3.37}$$

Let the outgoing edges of the cluster, the inter-cluster edges, l_c^{out}, be represented as a fraction of the total number of edges of a cluster, i.e $l_c^{out} = ad_c$ where $a \geq 0$. Equation (3.37) then becomes

$$\frac{l_c}{m} - \left(\frac{(a+2)l_c}{2m} \right)^2 > 0 \tag{3.38}$$

which rearranges to

$$l_c < \frac{4m}{(a+2)^2} \tag{3.39}$$

If $a = 0$, then the cluster of s is disconnected from the network and is a separate component of the network. Consider $a > 0$. Then equation (3.40) becomes an upper limit for how many internal edges can be included in a module cluster. For $0 < a < 2$, l_s is has an upper bound in $(m/4, m)$. A sufficient condition for a cluster to be conditions for a community is then

$$l_c < \frac{m}{4} \text{ and } a < 2 \tag{3.40}$$

This presents a problem as the l_c is now dependent on the size of the network. The authors then construct a network that illustrates this problem and proves how the maximum modularity measurement can group very distinct groups together. Consider a network with a pair of clusters, $\mathcal{M}_1$ and $\mathcal{M}_2$, where both hold to the condition presented in equation (3.40), and call the remaining network $\mathcal{M}_0$. Let $\mathcal{M}_1$ have l_1 edges with l_1^{out} edges to $\mathcal{M}_0$. Similarly let $\mathcal{M}_2$ have l_2 edges with l_2^{out} edges to $\mathcal{M}_0$. Lastly, let $\mathcal{M}_1$ and $\mathcal{M}_2$ be l_{int}. It follows that non-negative a_1, b_1, a_2, b_2, exist such that, $l_{int} = a_1 l_1 = a_2 l_2$, $b_1 l_1 = l_1^{out}$, and $b_2 l_2 = l_2^{out}$. Since both $\mathcal{M}_1$ and $\mathcal{M}_2$ are module clusters by design then $a_1 + a_2 \leq 2$, $b_1 + b_2 \leq 2$ and $l_1, l_2 < \frac{L}{4}$. Then consider two separate partitions, A and B of this network, where A has vertices $\mathcal{M}_1$ and $\mathcal{M}_2$ as separate clusters and B has vertices $\mathcal{M}_1$ and $\mathcal{M}_2$ in the same cluster. The modularity of each partition, Q_A and Q_B, is given as follows, where Q_0 is the modularity of $\mathcal{M}_0$,

$$Q_A = Q_0 + \frac{l_1}{m} - \left[\frac{(a_1 + b_1 + 2)l_1}{2m}\right]^2 + \frac{l_2}{m} - \left[\frac{(a_2 + b_2 + 2)l_2}{2m}\right]^2 \tag{3.41}$$

$$Q_B = Q_0 + \frac{l_1 + l_2 + a_1 l_1}{m} - \left[\frac{(a_1 + b_1 + 2)l_1 + (a_2 + b_2 + 2)l_2}{2m}\right]^2 \tag{3.42}$$

As both $\mathcal{M}_1$ and $\mathcal{M}_2$ are modular clusters by construction it is expected that $Q_A > Q_B$ or that $\Delta Q = Q_B - Q_A < 0$. ΔQ can be expressed in terms of equation

(3.41) and (3.42) as

$$\Delta Q = \frac{a_1 l_1}{m} - \frac{(a_1 + b_1 + 2)(a_2 + b_2 + 2)l_1 l_2}{4m^2} \tag{3.43}$$

Notice that if $a_1 = a_2 = 0$, then equation (3.43) always holds true. Otherwise equation (3.43) can be solved for values of l_2 (without loss of generality) such that $\Delta Q < 0$ as follows,

$$l_2 > \frac{2ma_1}{(a_1 + b_1 + 2)(a_2 + b_2 + 2)}. \tag{3.44}$$

Equation (3.44) shows that l_2 is dependent on m as well as the various coefficients. S. Fortunato and M. Barthélemy show for which cases the inequality of equation (3.44) does not always hold [8]. For simplicity let $l_1 = l_2 = l$. Consider now the extreme case that $\mathcal{M}_1$ and $\mathcal{M}_2$ are joined to the other components of the network with a single edge; that is, $a_1 = a_2 = b_1 = b_2 = 1/l$. Equation (3.44) now becomes.

$$l > \frac{2L(1/l)}{(2/l + 2)(2/l + 2)} \tag{3.45}$$

which is equivalent to

$$l > \sqrt{\frac{L}{2}} \tag{3.46}$$

for larger enough values. Then for any $l < \sqrt{\frac{L}{2}}$ modularity will incorrectly merge two very distinct communities together. S. Fortunato and M. Barthélemy expand on this for cases where $l_1 \neq l_2$ but the result is similar [8]. This resolution limit has been seen in real networks and is a situation that must be addressed by any modularity-based community detection.

4 Overlapping and Hierarchical Spectral Method

4.1 Introduction

With the prior sections covering some of the aspects of how community detection works and some of the methods used, we provide additional functionality to these existing methods to tailor them toward community detection in social networks. We present our spectral method which is particularly suited to finding communities that are overlapping, as well as find finding multi-tiered communities of varying sizes, while overcoming some of the obstacles presented by the resolution limit of modularity.

Our method first looks at the gain in modularity by adding vertices together into the same clusters of a partition. The particular patterns of the spectral space allow us to determine estimates of communities and vertices that are highly related to these communities. It then becomes possible to determine the number of internal edges to a predicted group without it being fully defined. We construct a membership matrix on the indicated number internal edges. We then determine how to reduce the amount of overlap between our communities and then recursively partition them until all communities can no longer be broken up.

4.2 Spectral Belonging

To begin, recall that we consider the community G, with vertices v and edges e where $A(G)$ is the corresponding adjacency matrix. Let $\mathbf{B}(G)$ be the corresponding modularity matrix for a graph G. We then construct the spectral decomposition as detailed in the spectral decomposition section. We follow a subset of the spectral decomposition shown in section (3.3.2) but we only consider the two first eigenvalues. We use only this limited information as it provides the most optimum splits for modularity but maintains an overall run time efficiency that is on par with other algorithms. Let λ_1 and λ_2 be the corresponding maximum eigenvalues with eigenvectors $\mathbf{a}_1$ and $\mathbf{a}_2$ respectively. Note that if $\lambda_2 < 0$ then only λ_1 is necessary. If both eigenvalues are less than zero, then equation (3.28) would never be positive indicating no modular splits. The spectral decomposition is constructed where each vertex v_i has a corresponding vertex vector, r_i. We re-adjust equation (3.35) to

$$\begin{aligned} Q &= \sum_{i=1}^{n} \sum_{c=1}^{|P|} (s_{ic} a_i)^2 (\lambda_i) \\ &= \sum_{i=1}^{n} \sum_{c=1}^{|P|} \left(\sum_{j=1}^{2} ([r_i]_j s_{ic}) + \sum_{j=3}^{n} (s_{ic} a_{i,j})^2 (\lambda_i) \right) \end{aligned} \tag{4.1}$$

We remove the α as we expect at least the first two eigenvalues to be positive. Equation (4.1) has separated the term we are maximizing in this method, $\sum_{i=1}^{|c|} \sum_{j=1}^{2} ([r_i]_j s_{ic})$, from the other as the greatest contribution to modularity comes from the first two eigenvalues. Then we define the gain from a community c from the two lead eigenvalues as,

$$Q_k = |\mathbf{R}_c| = \sum_{j=1}^{2} \sum_{i=1}^{|c|} [r_i]_j^2. \tag{4.2}$$

This is easily reducible to the sum of dot products of the spectral space as follows,

$$
\begin{aligned}
Q_k = |\mathbf{R}_c| &= \sum_{j=1}^{2} \left[\sum_{i=1}^{|c|} r_{i,j} \right]^2 \\
&= \left[\sqrt{\lambda_1} \sum_{i=1}^{|c|} a_{i,1} \right]^2 + \left[\sqrt{\lambda_2} \sum_{i=1}^{|c|} a_{i,2} \right]^2 \\
&= \sqrt{\lambda_1} \sum_{i=1}^{|c|} a_{i,1} \cdot \sqrt{\lambda_1} \sum_{j=1}^{|c|} a_{j,1} + \sqrt{\lambda_2} \sum_{i=1}^{|c|} a_{i,2} \cdot \sqrt{\lambda_2} \sum_{j=1}^{|c|} a_{j,2} \\
&= \sum_{i,j=1}^{|c|} (\sqrt{\lambda_1} a_{i,1} \cdot \sqrt{\lambda_1} a_{j,1}) + (\sqrt{\lambda_2} a_{i,2} \cdot \sqrt{\lambda_2} a_{j,2}) \\
&= \sum_{i,j=1}^{|c|} (r_i \cdot r_j)
\end{aligned}
\tag{4.3}
$$

It follows that the gain of adding a vertex, v, with vertex vector, r_v, to a community vector $\mathbf{R}_c$ is defined as follows

$$
\begin{aligned}
Q_{c+v} - (Q_c + Q_v) &= \sum_{i,j=1}^{|c+v|} r_i \cdot r_j - \left(\sum_{i,j=1}^{|c|} r_i \cdot r_j + r_v \cdot r_v \right) \\
&= 2 \sum_{i=1}^{|c|} r_v \cdot r_i = 2 r_v \cdot \left(\sum_{i=1}^{|c|} r_i \right)
\end{aligned}
\tag{4.4}
$$

Using both equations (4.3), (4.4) we are able to determine how much a vector belongs to any predefined community.

4.3 Community Estimation

To begin finding communities in a network we further investigate the patterns found in the spectral decomposition of the network. It has been illustrated that a modularity spectral dimension that the data that indicates a community must be further than 90 degrees apart from data that indicate another community or otherwise it would be more beneficial to join them as one and the communities would not be different.[25, 19]. Then given the constraint that communities must appear 90 degrees apart it follows that the number of communities that can appear in a p-dimensional space is p+1. Then for our algorithm, as we only inspect the first two eigenvectors of any modularity matrix we can expect up to three representations of communities to appear in the spectral decomposition. Equation (4.3) indicates that a group of vertices is a community if there is large sum of dot products between the vertices vectors. Thus there we expect to see up to three groups that together form a relatively high sum of dot products. Although there are many algorithms for detecting large groups of vertex together, such as the 'MELO' algorithm presented by C. J. Alpert S. -Z. Yao [1], we present our own variant as it captures information about the vertex vectors that is necessary for developing overlapping communities.

The key aspect of our vertex vector partition algorithm is we that consider the immediate community surrounding any vertex vector. As stated before, we expect the communities to exist in the spectral decomposition, there is expected to be groups of vertices that result in a high modularity. We define the *potential neighbouring community* of a vertex vector, r_i, as the sum of positive modularity of including $4r_i$ and any other vertex vector r_j subtracted by a partition where they are in different communities.

$$\begin{aligned}\text{potential community of } v &= \sum_{i,j=1}^{n} (Q_{i+j} - (Q_i + Q_j)) \\ &= \sum_{i,j=1}^{n} (r_i \cdot r_j - (r_i \cdot r_i + r_j \cdot r_j)) \qquad (4.5) \\ &= \sum_{i,j=1}^{n} 2r_i \cdot r_j \text{ where } r_i \cdot r_j > 0\end{aligned}$$

For the indication of a potential community we only consider vertices that give a positive value be included with r_i, as we only investigate a nearby community. Thus the potential neighbouring communities of a vertex vector, r_i, exist up to 90 degrees from the vertex vector r_i. However, for the estimate to be accurate, it is also necessary to consider that there exist additional communities that other vertices belong to, that exist at least 90 degrees away. If we introduce another community vector, r_{i_0}, then any vector, r_j, in the small angle of the two vertex vectors has a positive gain to both communities. Thus adding it to the local community of r_i implies that modularity is lost from not adding to r_{i_0} which creates the following inequality from measuring local communities

$$(Q_{i+j} - (Q_i + Q_j)) - (Q_{i_0+j} - (Q_{i_0} + Q_j)) = 2(r_i \cdot r_j - r_{i_0} \cdot r_j) \tag{4.6}$$

We then define a vertex vector's strictly local community as if there were two equally-sized communities directly 90 degrees away from the vertex vector. We expect the actual neighbouring communities to be of roughly the same size in magnitude as all three summed together are equal to 0. Then the total modularity gain to adding a vertex vector to community centred at r_i must consider that the vertex vector is removed from the other possible community that is less than 90 degrees away. The strictly local community of a vertex, r_i, is then sum of the positive differences of adding any vertex to the local community minus adding it to the peripheral adjacent communities

$$\begin{aligned} local(r_i) &= 2\sum_j (r_i \cdot r_j - r_{i_0} \cdot r_j) \\ &= 2\sum_j |r_j|(|r_i|\cos\theta_{ij} - |r_0|\cos\theta_{i_0 j}) \\ &= 2\sum_j |r_j||r_i|(\cos\theta_{ij} - \cos(\frac{\pi}{2} - \theta_{ij})) \end{aligned} \tag{4.7}$$

Equation (4.7) provides what can be considered as the minimal measurement for a community forming around a vertex vector. It does not require any priori knowledge about the network and is not based on any partition of the vertices, which makes it an ideal approach for social networks. Once each vertex vector is given a measurement, if the vertex has the largest local community within a 90 degree arc centred on itself then we can estimate that there is a community surrounding it. We refer to these vertex vectors as *predictor vectors.* We denote predictor vectors as $\tilde{r}_c$. Generally three predictor vectors are found through measuring all the local communities. However, if the second eigenvector indicates a significantly smaller community, then there may only be two, since vectors adjacent to large predictor vector will have a large local community but may be within 90 degrees of a small predictor vector. Thus our geometric constraint prevents finding the smaller predictor vector. In the rare case of a very noisy network, then there may only be one vector found.

Once predictor vectors are found, we can form *predictor communities* around the predictor vectors. A predictor community consists of all vertex vectors that give a positive contribution corresponding to local community of the respective predictor vector, denoted as

$$\tilde{\mathbf{R}}_j = \sum_i r_i, \text{ for } \theta_{i,j} < \frac{\pi}{4} \tag{4.8}$$

Predictor communities provide an outline, or silhouette, of what communities the lead eigenvectors correspond to.

We then begin determining every vertex vector's belonging to each of the predictor communities. An intuitive measurement is the modularity gained, defined by equation (4.4), of adding a vertex vector to as to a predictor community. However this presents a number of problems. Communities exist on many scales but in modularity optimization if they are too small in comparison to the rest of the network there is no gain in separating these communities. Not all communities are represented in the first two leading eigenvectors. If one could compute the first $|p| - 1$ eigenvectors where $|p|$ is the number of communities then an ideal partition would be the choice of $|p| - 1$ independent mutually orthogonal columns of the spectral decomposition. Thus in 2 dimensional spectral decomposition there are many vertex vectors that contribute very little to modularity as they refer to communities represented in the positive eigenvectors that have not been calculated. Vertex vectors that have relatively small gain to be added a community we say have *small spectral participation*, as they provide small amounts of modularity to our found predictor communities and may correspond to different communities.

Then an accurate method for estimating the communities in the spectral space would be to consider assigning each vertex vector to the predictor community or to an additional group of vertices that still have unknown membership. Vertex vectors that clearly belong to a predictor community have a high value by equation (4.4). However this is not the case for all vertex vectors that belong to that community. If we consider adding a vertex, v, to community, c, then the change of modularity, given by equation (3.36) is,

$$\begin{aligned} Q_{c+v} - (Q_c + Q_v) &= \left(\frac{l_c + k_v^{int}}{m} - \left(\frac{d_c + k_v}{2m} \right)^2 \right) \\ &\quad - \left(\left(\frac{l_c}{m} - \left(\frac{d_c}{2m} \right)^2 \right) + \left(- \left(\frac{k_v}{2m} \right)^2 \right) \right) \\ Q_{c+v} - (Q_c + Q_v) &= \frac{k_v^{int}}{m} - \frac{d_c k_v}{2m^2} \end{aligned} \tag{4.9}$$

where k_v is the degree of v and k_v^{int} is the number of edges between v and the vertices of c. Then if we consider k_v^{int} as fraction of k_v that is $k_v^{int} = ak_v$ where $0 \leq a \leq 1$ equation (4.9) becomes

$$Q_{c+v} - (Q_c + Q_v) = \frac{k_v}{m} \left(a - \frac{d_c}{2m} \right) \tag{4.10}$$

Then equation (4.10) shows how modularity gain is partially dependent on k_v and the fraction of internal edges a. Then we can expect our predictor vertices be highly connected to vertices to in their relative community. However, a vertex with a small degree but with a large proportion of the edges meeting it going to the community have smaller gain. Then vertices of smaller degree are also grouped with other vertices that have a low spectral participation. This is reflected in the spectral decomposition, with a low value in equation (4.4). Then it follows from both equation (4.10) and equation (4.4) that

$$2r_v \cdot (\tilde{\mathbf{R}}_c) \propto \frac{k_v}{m} \left(a_v - \frac{d_c}{2m} \right) \tag{4.11}$$

Then to get around the difficulty of distinguishing vertices with a lower spectral participation but a large number of edges into the community from vertices that have a low spectral degree but do not belong to the predictor community, we consider that our predicator vertex vector, v_p, belongs entirely to the the respective predictor community formed around it. By considering the ratio of vertex vectors to that of a nearby predictor vertex vector. If we consider a ratio of the modularity gained from adding the predictor vector to the community it belongs to that of another vertex vector v_h we see the follow estimated relative gain

$$\begin{aligned}\frac{2\, r_h \cdot \tilde{\mathbf{R}}_c}{2\, r_p \cdot \tilde{\mathbf{R}}_c} &\propto \frac{\frac{k_h}{m}\left(a_h - \frac{d_c}{2m}\right)}{\frac{k_p}{m}\left(a_p - \frac{d_c}{2m}\right)} \\ \frac{2\, k_p\, r_h \cdot \tilde{\mathbf{R}}_c}{2\, k_h\, r_p \cdot \tilde{\mathbf{R}}_c} &\propto \frac{a_h - \frac{d_c}{2m}}{a_p - \frac{d_c}{2m}},\end{aligned} \tag{4.12}$$

The right hand side term of equation (4.12) is a ratio of the internal edges of v_h to that of our predictor vector, v_p for the respective predictor community without actually knowing what vertices belong to the community. This is the heart of our algorithm as it we can estimate the the ratio of internal edges a vertex may have to a community without knowing the community itself. Vertices with a small number of internal edges can be determined if they belong to a predictor community as we see if a vertex has an equivalent or better ratio of internal edges as that of the predictor vertex, which we assume fully belongs to the predictor community. We then construct a belonging matrix to show this information as

$$\beta_{i,j} = \frac{k_j \tilde{\mathbf{R}}_j \cdot r_i}{k_i \tilde{\mathbf{R}}_j \cdot r_j} \tag{4.13}$$

Although $\beta_{i,j}$ provides the measurement from equation (4.12) it often possible that $\sum_j \beta_{i,j} > 1$ as these equations are only approximations and network data will often contain outliers. We normalize and include the potential set unknown vertices in the following membership matrix to produce,

$$\begin{aligned} M_{i,j} &= \frac{\beta_{i,j}}{\max(\sum_k \beta_{i,k}, 1)} \\ M_{i,|c|+1} &= 1 - \sum_j M_{i,j}. \end{aligned} \tag{4.14}$$

The measurement found in equation (4.14) is then the heart of our algorithm as it provides a method to divide a network into three three modular groups and a set of vertices that only weakly correspond to any of the three groups.

4.4 Recursive Partitions

For the first pass we find the Membership Matrix from the prior section and assign vertex vectors. The exception is a network with only negative eigenvectors because then there is no modular split. We then reconstruct adjacency matrices for each of the subgraphs by applying an edge-weighting function for overlapping modularity such as equation (3.14). We expect the modularity of the new subgraph to have greater modularity compared to the prior iteration of communities. However, the group of unknown vertices is comprised of what we consider to be smaller communities that are unrelated to the current partition. It is no surprise that the modularity value of this subset of the partition will often be negative. Therefore we must continually apply our community estimation algorithm to the group of unknown vertices, until either all vertices have been assigned to a group or until a positive modularity is reached.

As the community of unknown vertices tends to favour vertices that provide little modularity gain and a large number of vertices that do not correspond to the community indicated by the eigenvector, we can expect the communities within the resolution community to appear in the unknown community. However, to limit the number of vertices that do not correlate strongly to vertices of those that are unaffiliated with the communities, we remove vertices that have less than one full edge contained within the unknown group. That is to say, for a vertex vector, r_i, to belong to the unknown group then $\sum_j \mathcal{F}(\alpha_{v_i,|c|+1}, \alpha_{v_j,|c|+1}) \geq 1$ otherwise it can not be certain it belongs to any group that may exist in unknown group. Rather than consider a potential vertex v_i that at best it is on the peripheral of, to be in the unknown group, we move r_i into a known community by rebalancing the membership matrix. We do this rebalancing proportional to the known memberships, as expressed as the following equation,

$$\begin{aligned} M_{i,j} &\leftarrow M_{i,j} + \frac{M_{i,j} \times M_{i,|c|+1}}{\sum_k M_{i,|c|}} \\ M_{i,|c|+1} &\leftarrow 0 \end{aligned} \tag{4.15}$$

Equation (4.15) removes the membership in the unknown group but also ensure that the total membership is still 1.

After this step we must do a similar de-noising procedure for the communities that were found. Often the communities may absorb smaller communities, gaining partial membership of the vertices, while these smaller communities have partial memberships in the unknown groups of vertices or other found communities. If we consider the possible minimal contribution a vertex vector, r_i, may have while still belonging to a community, found or unknown, is $M_{i,j} = 1/3$. It follows that the minimal edge, in the first pass, contribution is $\mathcal{F}(1/3, 1/3)$. Then we construct a threshold value of what vertices can still belong to a cluster as having at least one edge of minimal internal edge weight such as

$$v_i \in G_c \text{ if and only if } \mathcal{F}(\alpha_{i,c}, \alpha_{j,c}) \geq \mathcal{F}(1/3, 1/3) \text{ for some j} \in V. \tag{4.16}$$

For each vertex removed this way, the membership matrix must be rebalanced but only after all vertices that do not meet the requirement of (4.16) have been removed. Then using equation (4.15) we can rebalance the membership matrix for the vertex vector removed. For further recursive steps the threshold value varies for each vertex vectors it depends on the membership value assigned in the prior recursive step.

To apply a recursive step the total membership of each vertex is limited by the by their membership value of the prior community, $\alpha_{i,p}$, they were found in instead of 1. Otherwise the constraint (3.13) of overlapping communities would not be applied properly, that is $\sum_c \alpha_{i,c} = 1$. We adjust our threshold of a minimal edge to be $\mathcal{F}(1/3, \alpha_{i,p}/3)$.

For each recursive step the new modularity is calculated and compared to that of the prior cluster's modularity. When we start our current modularity is 0 as all the graph is considered to be in one cluster. Then as we produce more clusters we expect to see an increase in modularity to a level where there are no positive eigenvalues or where communities are being split apart. The first condition means that a highly dense community has been found with no modular splits. However the later condition, which is far more likely to occur, is when the sum of modularity of the new clusters in less than that of when they are only a single cluster. The vertices of the unknown group, however, have little or nothing to do with groups found in the spectral space this does not imply that they modular. It is far more likely to expect small unrelated communities to have grouped together in the unknown group. Thus before any comparison can be done the unknown group must be recursed again to be sure that there are not many unrelated clusters within. That is, for each group of the unknown vertices, we must recursively check new partitions until the modularity of the unknown group adheres to our conditions of either there is no longer a increasing modularity or there are no positive eigenvalues.

4.5 Main Algorithm

To recap, our method uses the following steps.

1. Find the spectral decomposition corresponding to the first two eigenvectors with vertex vectors $[r_i]_p = \sqrt{\lambda_j} a_{ip}$, where a_{ip} is the element of the eigenvector matrix.

2. For each vertex vector find the the local community $local(r_i) = \sum_j |r_i||r_j|(\cos(\theta_{ij}) - \cos(\frac{\pi}{2} - \theta_{ij}))$, where θ_{ij} is the angle between r_i and r_j.

3. If a local community of a vertex vector, r_i is the largest within 90 degrees of the vertex vector then it is a predictor vertex vector. The sum of the vertex vectors within 45 degrees of the predictor vertex vector makes up the corresponding predictor community, $\tilde{\mathbf{R}}_i$. Let $|c|$ be the number of predictor vertex vector found.

4. An $n \times |c|$ belonging matrix, β, is constructed where $\beta_{i,j} = \frac{k_j r_i \cdot \tilde{\mathbf{R}}_j}{k_i r_j \cdot \tilde{\mathbf{R}}_j}$ for $j \leq k$, and where k_i and k_j are the degrees for vertex v_i and that of corresponding vertex vector, v_j.

5. Then the $n \times (|c| + 1)$ membership matrix is set as $M_{i,j} = \frac{\beta_{i,j}}{\max(\sum_c \beta_{i,c}, 1)}$ for $j \leq |c|$ and $M_{i,|c|+1} = 1 - \sum_j M_{i,j}$.

6. For each $\mathcal{F}(\alpha_{i,c}, \alpha_{j,c}) < \mathcal{F}(1/3, \alpha_{i,c}/3)$ for all $v_j \in V$ then v_i is removed from R_c. For some edge belong function $\mathcal{F}$ and vertex, v_i belonging values $\alpha_{i,c}$ for a cluster c.

7. If there has been any re-assigning of vertices start from step 6 again.

8. If the sum of modularity of the cluster as defined in equation (3.23) is higher than the previous partition, repeat the steps 1 through 7, until there is a negative modularity change or no feasible split.

Using this recursive method we find good results that are on par with other spectral algorithms. This is further explained in the results of chapter 6.

4.6 Run-Time Analysis

In this section we cover the running time used by our algorithm and how to maintains the standards of other spectral algorithms. We outline each step of section 4.5 to analyze its time efficiency.

Step 1 follows from the same spectral decomposition used in the past by M. E. J. Newman [19]. By finding the eigenvalues and eigenvectors using the Lanczos method it is necessary to multiple the modularity matrix $\mathbf{B}$ by a trial vector $\mathbf{x}$. This takes significant time as the modularity matrix is dense and would require $\mathrm{O}(n^2)$ time, rather than $\mathrm{O}(n+m)$ time for a sparse matrix. While $\mathrm{O}(n^2)$ is not a terrible time efficiency, the Lanczos method would typically require n iterations to converge, thus resulting in a runtime of $\mathrm{O}(n^3)$. However M. E. J. Newman shows that we can represent the modularity matrix as sparse matrix if we consider the multiplication of trail vectors as follows [19],

$$\mathbf{Bx} = A(G)\mathbf{x} - \frac{\mathbf{k}^T\mathbf{x}}{2m}\mathbf{k}. \tag{4.17}$$

If the lead eigenvalues, λ_n, are negative then an additional term of $-\lambda_n\mathbf{I}$ can be added to calculate the largest two vectors. To retrieve all the eigenvectors using Lanczos would require $\mathrm{O}(n^3)$ time, however, there exists variants of the Lanczos method to retrieve the first few using only $\mathrm{O}(n^2)$ time [19]. Using this reasoning we have built our own method to run only in O(n^2) and have built the remaining steps to run in O(n^2) time as well.

Steps 2 and 3 require checking all vertex vectors against all other vertex vectors. That is to say, for each n we do some constant time operations, such multiplying or adding, for of the each other n vertices. These steps have O(n^2) time efficiency. The construction of the belonging and and membership matrices only require O(n) time to calculate.

We determine the running time of our smoothing function where vertices are removed and the membership matrices are rebalanced. As a vertex is removed from a community or the unknown group its membership is shifted elsewhere, increasing its membership to a known community. In the next iteration we check if the vertex is now valid under equation 4.16 for all communities. Then each vertex is added to a community they are strong with, whether it be a known community or the set of unknown vertices. Each of these vertices now has a strength to community higher than the threshold and will not be moved again. We repeat the check for any new vertices. Then every vertex only has one chance to move. The worst case scenario is if we only move one vertex per iteration making for n iterations. Thus then our balancing method is O(n^2) time.

Then as all steps are performed in O(n^2) time our algorithm runs on par with others in existence.

5 Benchmarks Graphs and Results

In this section we provide our findings, and discuss how our algorithm compares with the methods that are currently in use. We use well-known benchmark graphs that have been tested with other methods and consider which are the most practical.

5.1 Example Graphs

In this first section we present our method's findings through some well-known example graphs. We will also introduce a small graph that directly captures the strength of method. We look at using the normalized mutual information to compare our algorithm with others, however, as overlapping normalized mutual information has faced some criticism, we reduce our overlapping method by replacing the check in the recursive step, 7, in the overview such that all final memberships are 1.

5.1.1 Zachary's Karate Club

For an introduction we began with the well-known Zachary's Karate Club social network. Zachary's Karate Club is a widely used benchmarked graph that first appear in 1977 as social network representing different groups (by W.W. Zachary [30]). This graph has 34 vertices and 150 edges and is shown in Figure 5. The Newman-Girvan Algorithm for Zachary's Karate Club social network found the maximal modularity with a modularity score of 0.4198 and partition and the graph into 4 separate communities. Our algorithm produces 5 separate communities with the same very similar structure and modularity score of 0.4147. Although this number is a little lower it's important to note that for a difference of 0.051 we enable our algorithm to contain overlapping communities.

Figure 4 shows the spectral decomposition of the network, which is what to be

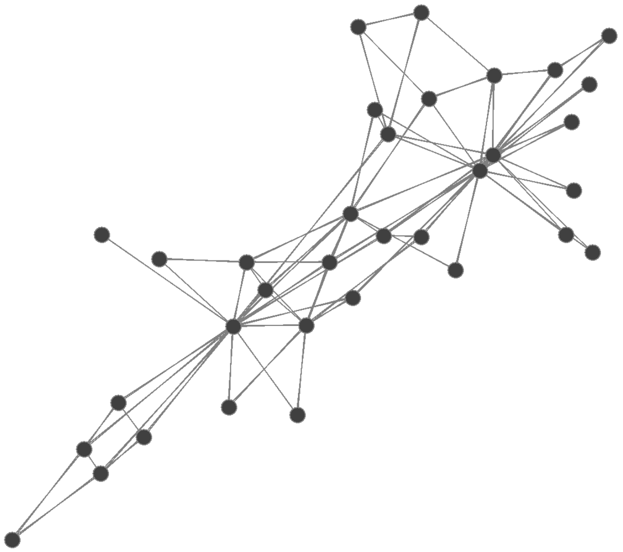


Figure 3: Zachary's Karate Club

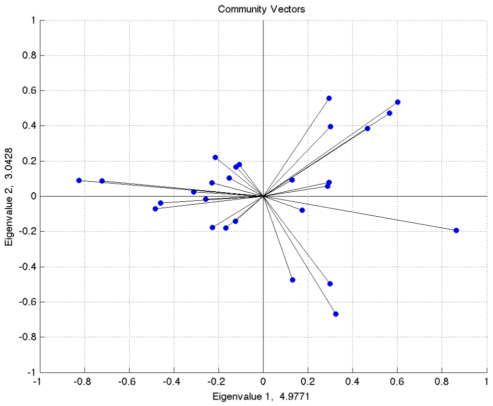


Figure 4: The standard modularity spectral decomposition of Zachary's Karate Club

expect from a smaller network. The three clusters are not initially clear from this figure as there is a larger outlier and many of the points cluster around the origin of the spectral decomposition. We apply our algorithm to the data to try to clarify the clusters within this spectral decomposition.

In Figure 5 we present the first pass of our algorithm. The predictor vertices are indicated as the light green lines emanating from the origin while the colour of the vertex vectors indicate the membership assigned to the vertex vector corresponding

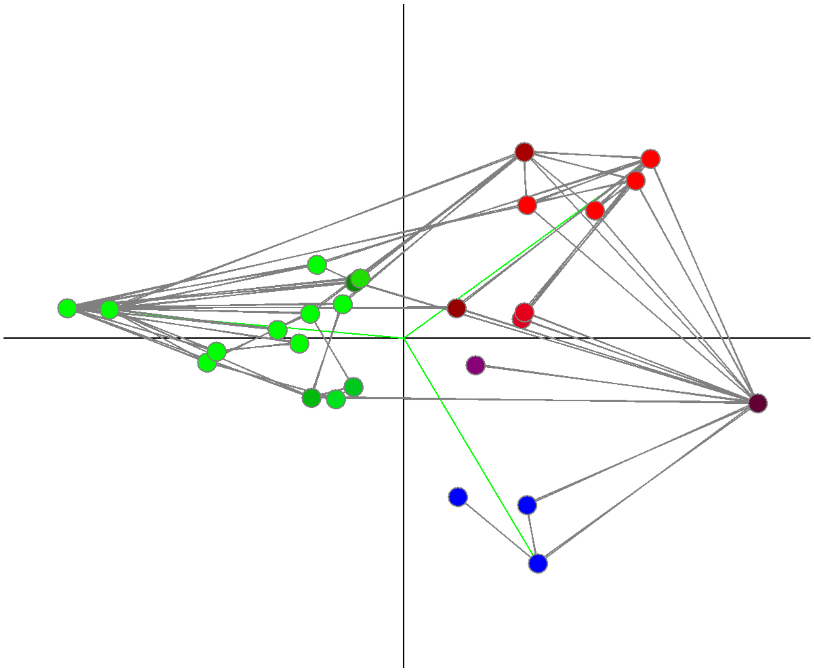
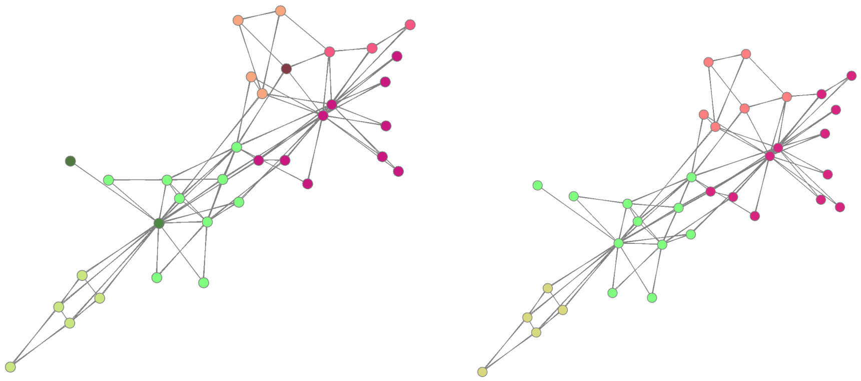


Figure 5: The spectral decomposition with the three leading predictor vertices and marked as light green (colour version) as lines from the origin. Vertices are marked with colours indicating their membership value to predictor communities.

to each predictor community, which is either red blue or green. After this step we apply our membership rebalancing before making another partition.

Then our final partition of the Zachary's Karate Graph compared with known partition of maximum modularity is shown in Figure 6 where the overlapping vertices have been darkened for visibility between the colours.



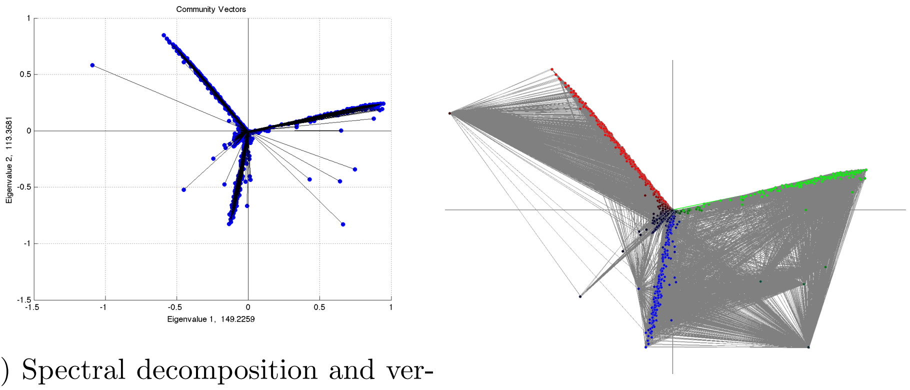
(a) Our final partition. (b) Partition maximum modularity

Figure 6: The comparison of our method's final partition to that of the maximum modularity partition. Our method achieves a overlapping cover of modularity of 0.4147 while the maximum modularity is 0.4198 found using partitioning.

5.1.2 SNAP Facebook

In this section we introduce a graph from the Social Network Analysis Platform at Stanford University. J. McAuley and J. Leskovec used a number of ego-networks and social networks to validate their own method for detection circles of friends [15]. In particular, they use a graph of 10 users ego-networks collected from Facebook surveys and anonymized. A user's ego network consists of all circles the user is a part of and the users inside those circles. J. McAuley and J. Leskovec illustrated that those related to a user should appear in social circles, such as family, fellow students, and friends [15]. In their study they also collected data about the circles that would further help their model identify the different circles. We use only the network data provided to illustrate how the method works on large data sets. The FaceBook Snap network has 4039 vertices and 88234 edges. On our first pass the data show the following graphs.

The Facebook Snap graph illustrates how a group of unknown vertices form in the center of the spectral decomposition. In the example the number of vertices



(a) Spectral decomposition and vertex vectors.

(b) Initial Membership of vertices

Figure 7: The first iteration of the SNAP Facebook graph.

that correspond to the group of unknown vertices is over 3000, three quarters of the vertices in the graph. This of course makes a lot of sense considering that the Facebook SNAP graph was found using 10 ego-networks and there should be some significantly separated clusters. Thus when there are only three vectors to represent the data, a large amount of the data does not significantly contribute to the modularity of these predicted communities. A strength of our algorithm is that it does not try to associate vertices into clusters they have no similarities with.

5.1.3 Spectral Example Graph

Now we introduce a graph that really captures the strength of our adaptive membership matrix. Consider the following graph composed of 3 cliques of 20 vertices all connected to a clique of 5 vertices. One vertex for each clique of 20 is connected to the clique by a single edge. The spectral decomposition of this graph is somewhat misleading in the information it provides. The spectral decomposition shown Figure 9 in is dominated by the large cliques. Thee vertex vector of the degree 1 vertices and the vertices of the 5-clique joined to the 20-clique have nearly the same spectral space making it quite difficult to distinguish them. This is mostly because they bring the

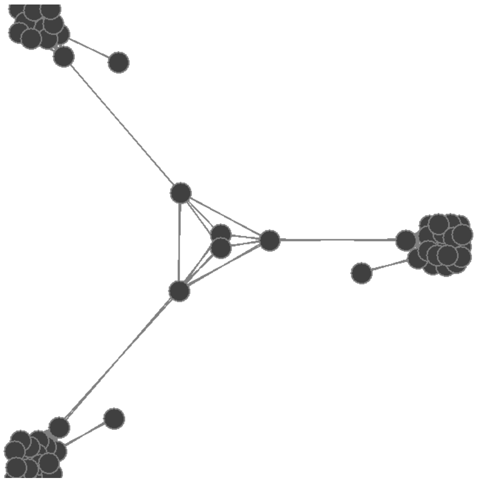


Figure 8: A graph 3 cliques of order 20, and one central clique of order 5

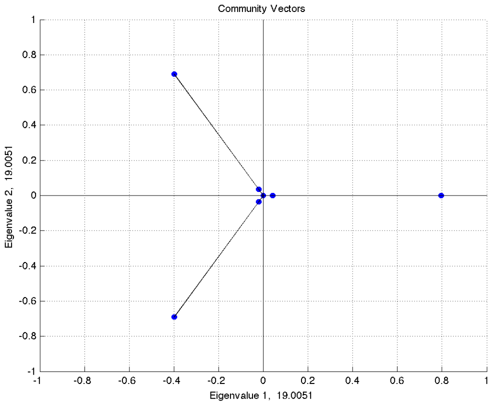


Figure 9: The spectral decomposition of Figure 9

same number of internal edges to each of the predictor communities. However, with our modification to membership matrix by introducing the fraction edges in equation (4.13) we are to distinguish the differences between these two vertex vectors. This is evident in Figure 10 as we show the initial membership matrix applied to the graph.

This shows the improvement of our algorithm over other spectral algorithms most of which will generate a partition from the bisecting line originating at the origin point of the spectral space [25, 19]. In those cases the algorithms do some similar

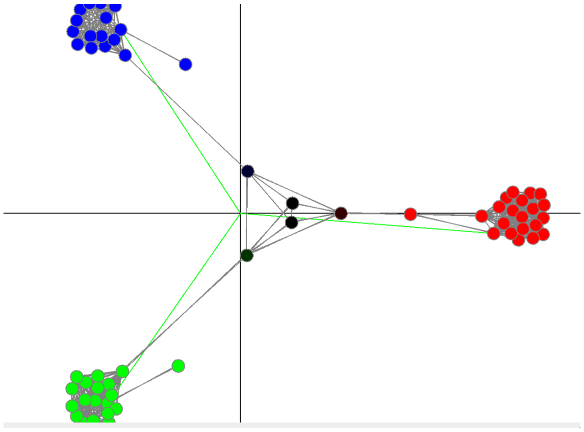


Figure 10: The initial membership matrix of Figure 8

smoothing by moving the membership of vertices back and forth via Kernighan-Lin iterations, but this still does not guarantee that the vertices will be passed on together into the same cluster.

5.2 Benchmark Graphs

In the previous section we considered how our algorithm works on specific graphs; however specific graphs provide only a glimpse of an algorithm's functionality for finding communities. To compare community detection algorithms there have been many developments in generating graphs that have known communities that interact.

Then to measure an algorithm's success the groups identified by the algorithm are compared against the known true communities. To do this we use *Normalized Mutual Information*. Let A and B be partitions of the same graph. Then a $confusion$ matrix, N is of size $|A| \times |B|$ and N_{ij} corresponds to the number vertices in community A_i that appear in B_j. Then the similarity of partitions based on information theory is

$$I(A,B) = \frac{-2\sum_{i=1}^{|A|}\sum_{j=1}^{|B|}\log(N_{ij}n/N_iN_j)}{\sum_{i=1}^{|A|}N_i\log(N_i/n)+\sum_{j=1}^{|B|}N_j\log(N_j/N)}, \tag{5.1}$$

where N_i is the sum over row i, N_j is the sum over column j and n is the number of vertices. We use these measurements to discern how similar B is to A. In the case of algorithm comparisons we investigate that how close our partition B is to ground truth partition A, in the following two benchmarks. This measurement is only available for partitions of communities and not overlapping covers, so we adjust our algorithm to return partitions.

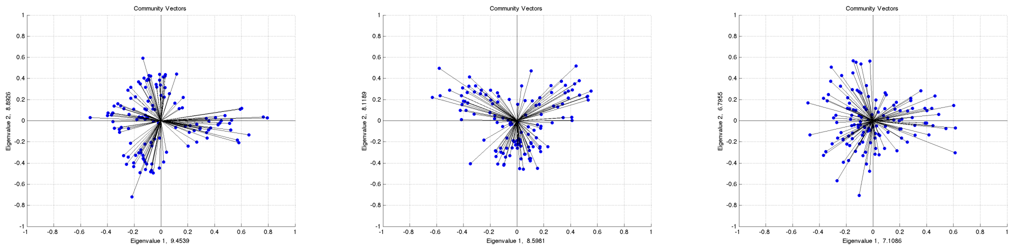
5.2.1 Planted l-Partition

A significantly popular group of benchmark graphs are those generated by a *planted l-partition* [7]. In the planted l-partition we consider a graph of n vertices where $n = g \cdot l$ vertices in l groups with g vertices each. Each vertex is given an average degree of k where $k = p_{in}(g-1) + p_{out}g(l-1)$, and p_{in} is the probability of an internal edge and p_{out} the probability of an external edge. Then if $p_{in} > p_{out}$ we expect to see more internal edges. Newman and Girvan introduced a particular planted l-partition graph that has been widely used to identify communities. This particular planted l-partition graph has is $l = 4, g = 32$ and $k = 16$. The number of expected internal edges and external edges are z_{in} and z_{out} where $z_{in} = p_{in}(g-1) = 15p_{in}$ and $z_{out} = p_{out}(g-1)l = 96p_{out}$. Let $z_{out} = 16 - z_{in}$. Then we can expect communities to appear until $z_{out} > 8$ as then there are more external edges then internal edges (indicating less than weak communities). We then compare our results on these test cases with those found in a comparative test done by L. Danon, A, Diaz-Guilera, J. Duch, A Arena [2]. That is, we see what the normalized mutual information are when $z_{out} = 6, 7, 8$.

By correlating our result to those found in L. Danon *et al.* our results fall a little below the the models presented. We address this issue by examining the spectral decomposition of these graphs as, shown in Figure 11. Although it exhibits three

External Edges	Modularity	Normalized Mutual Information
$Z_{out} = 6$	0.2783	0.5854
$Z_{out} = 7$	0.2585	0.5308
$Z_{out} = 8$	0.2027	0.3547

directions that communities cluster, since each is equally connected to the fourth community it produces noise on the peripheral of each community. Normally we are able to filter this behaviour with our belonging ratio equation (4.12) but all vertices have the same expected number of edges meeting them so this results in the loss of one of our algorithm's strongest features.



(a) Spectral space $z_{out} = 6$ (b) Spectral space $z_{out} = 7$ (c) Spectral space $z_{out} = 8$

Figure 11: Realizations of spectral space of the specific l-partition graphs

5.2.2 Benchmarking: LFR Graphs

Although the planted l-partition benchmark provides a easy measurement for detecting communities, the communities that do exist are on the borderline of becoming less than weak communities. In social structures it has been seen that there are power law distributions in both the size of the communities involved and the degrees of vertices. Authors A. Lancichinetti, S.Fortunato, and F. Radicchi introduce a similar model to the l-partition but had the communities vary on these power law distributions. Vertices are given a degree from a power law distribution, of α, between k_{min} and k_{max} and the size of communities are determined from a power law distribution, of β, between s_{min} and s_{max}. The algorithm considers the the ratio of internal to external edges as the parameter μ where $\mu > 0.5$ would no longer guarantee a weak community. We compared our results to some of those found in the comparative analysis done by A. Lancihinetti and S. Fortunato in 2009 [14]. We consider the same four parameters for realizations of the LFR mode, where we have 1000 vertices, each with an expected degree of 20 and forming communities sizes either between 10 and 50 or 20 to 100 and the mixing parameter 0.3 and 0.5.

Size of communities and $\mu = 0.3$	Modularity	Normalized Mutual Information
Small and $\mu = 0.3$	0.5442	0.8968
Large and $\mu = 0.3$	0.5570	0.8304
Small and $\mu = 0.5$	0.3305	0.5567
Large and $\mu = 0.5$	0.3308	0.5403

Our results for this graph are significantly stronger as the community entities now vary significantly. Vertex vectors that exist on the periphery can now be segregated by our belonging ratio equation (4.12). We see that our algorithm is stable for this possible scenario. Our algorithm does not outperform all others in every type of realization, but it remains consistent in finding communities.

6 Conclusions

In this work we have provided a new method whose goal is to be more specific for finding social online communities. In this section we address the strengths that our algorithm has and the contributions made to this field of research. We also review some of the weaknesses and the difficulties in building algorithms for social communities. Last we discuss some necessary further improvements and give a sense of where this algorithm can be developed.

6.1 Strengths and Contributions

The greatest strength of our algorithm is our belonging gain ratio equation (4.12) which provides a unique measurement that ties the spectral space and the original space together. With this strength we can easily distinguish between two vertex vectors that both contribute the same number of internal edges but have different external edges. We are able to do this with very little information by only relying on the communities indicated by the leading eigenvalues.

Our algorithm provides a direct measurement for overlapping communities. Previous spectral algorithms have not made use of the ability to see the modularity gain that a vertex vector has to communities represented in the space.

This method is the the first to make use of the information given in the spectral space for overlapping communities and gives consideration to vertices that are not related to the the larger communities represented in the spectral space. Section 6.1.3 is an example of how our method directly favours communities that can be lost from partitions that originate from the origin of the spectral space. Our algorithm protects communities with low modularity values from being separated into larger communities. This also benefits the resolution limit of modularity, as communities that are identified as being in the unknown group will be individually investigated without the influence of larger communities which would cause an increase in modularity if they were joined together.

6.2 Weaknesses and Difficulties

Our method does have some weaknesses especially toward the particular l-partition of Section 6.2. Our algorithm often performs poorly on graphs that do not represent communities, although they present interesting mathematical graphs, our algorithm expects communities to have varying individuals and relationships. Our algorithm can still be susceptible to the resolution limit but often the smaller graphs are protected by being placed in group of unknown vertices with other smaller communities. This often results in many smaller communities that together produce a negative modularity and result in being separated. However, our findings in the large LFR graphs indicate that those graphs would have to be very large.

Although our algorithm is designed to come from a strong sociology background, such as consider communities that have hierarchal overlapping and occur in various sizes, there still exists large a divide between the fields of mathematics and sociology. Often mathematical models rely upon the statistical properties of networks such as the power law distributions of degrees and community sizes while overlooking many social network consideration. Mathematical models of real networks rarely consider noise as a possible side effect and even then those that do do not always contain significant validation. The finding of E. Gilbert showed that noise does exist and weightings of edges must be a consideration, and this information is not yet available in benchmark testing [9]. Without weighting edges, interactions in socail network that do not represent or contradict community structure appear as strong as regular interactions and can lead to misleading findings [17]. Such example networks are still needed. While sociology considers many excellent ego-based communities, the question of the how community structure forms and how social aspect of network form has not been yet a research question. Often studies consider the types of communities that exist but not the graphical representations such as the difference between bond based communities and identity based communities [24].

Lastly due to the limitation of our master's thesis we have not develop our algorithm as much as we would like to. There are still many validations our method needs such as the comparative analysis in A. Lancihinetti and S. Fortunato for $\mu > 0.5$ as in [14], or using normalized mutual information in the recursive smoothing for better results, especially for overlapping communities. There are still many comparisons to existing algorithms to be made and differences in result need to be justified.

6.3 Further Research

To continue this area of research there many more improvements and tests that can be considered for this algorithm. As before, using normalized mutual information as the part of smoothing the partition will definitely enable a more thorough overlapping algorithm. Similarly partitions can be made more the fashion of a greedy algorithm by considering partitions into using large number of eigenvectors. As our algorithm uses only vector comparisons we can then expand our algorithm as we consider the most beneficial spectral space, provided the runtime does not exceed our goal of $O(n^2)$.

References

[1] Charles J. Alpert and So-Zen Yao. “Spectral partitioning: the more eigenvectors, the better”. In: *Proceedings of the 32nd Annual ACM/IEEE Design Automation Conference*. ACM. 1995, pp. 195–200.

[2] Leon Danon et al. “Comparing community structure identification”. In: *Journal of Statistical Mechanics: Theory and Experiment* 2005.09 (2005), P09008.

[3] Christopher P. Diehl, Galileo Namata, and Lise Getoor. “Relationship identification for social network discovery”. In: *AAAI*. Vol. 22. 1. (2007), pp. 546–552.

[4] Robin IM Dunbar. “Coevolution of neocortical size, group size and language in humans”. In: *Behavioral and brain sciences* 16.04 (1993), pp. 681–694.

[5] Robin I.M. Dunbar. “Social cognition on the Internet: testing constraints on social network size”. In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 367.1599 (2012), pp. 2192–2201.

[6] Robin I.M. Dunbar and Matt Spoors. “Social networks, support cliques, and kinship”. In: *Human Nature* 6.3 (1995), pp. 273–290.

[7] Santo Fortunato. “Community detection in graphs”. In: *Physics Reports* 486.3 (2010), pp. 75–174.

[8] Santo Fortunato and Marc Barthelemy. “Resolution limit in community detection”. In: *Proceedings of the National Academy of Sciences* 104.1 (2007), pp. 36–41.

[9] Eric Gilbert and Karrie Karahalios. “Predicting tie strength with social media”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM. 2009, pp. 211–220.

[10] Michelle Girvan and Mark E.J. Newman. “Community structure in social and biological networks”. In: *Proceeding of the National Academy of Sciences* 99.12 (2002), pp. 7821–7826.

[11] Bruno Goncalves, Nicola Perra, and Alessandro Vespignani. “Validation of dunbar’s number in twitter conversations”. In: *arXiv preprint arXiv:1105.5170* (2011).

[12] Mark S. Granovetter. “The strength of weak ties”. In: *American Journal of Sociology* (1973), pp. 1360–1380.

[13] Russell A Hill and Robin I.M. Dunbar. “Social network size in humans”. In: *Human nature* 14.1 (2003), pp. 53–72.

[14] Andrea Lancichinetti and Santo Fortunato. “Community detection algorithms: a comparative analysis”. In: *Physical Review E* 80.5 (2009), p. 056117.

[15] Jure Leskovec and Julian J Mcauley. “Learning to discover social circles in ego networks”. In: *Advances in Neural Information Processing Systems*. 2012, pp. 539–547.

[16] Jure Leskovec et al. “Statistical properties of community structure in large social and information networks”. In: *Proceedings of the 17th International Conference on World Wide Web*. ACM. 2008, pp. 695–704.

[17] David Liben-Nowell and Jon Kleinberg. “The link-prediction problem for social networks”. In: *Journal of the American society for information science and technology* 58.7 (2007), pp. 1019–1031.

[18] Mark E.J. Newman. “Finding community structure in networks using the eigenvectors of matrices”. In: *Physical Review E* 74.3 (2006).

[19] Mark E.J. Newman. “Modularity and community structure in networks”. In: *Proceedings of the National Academy of Sciences* 103.23 (2006), pp. 8577–8582.

[20] Vincenzo Nicosia et al. “Extending the definition of modularity to directed graphs with overlapping communities”. In: *Journal of Statistical Mechanics: Theory and Experiment* 2009.03 (2009).

[21] Bo Pang and Lillian Lee. “Opinion mining and sentiment analysis”. In: *Foundations and Trends in Information Retrieval* 2.1-2 (2008), pp. 1–135.

[22] Symeon Papadopoulos et al. “Community detection in social media”. In: *Data Mining and Knowledge Discovery* 24.3 (2012), pp. 515–554.

[23] Symeon Papadopoulos et al. “Community detection in social media”. In: *Data Mining and Knowledge Discovery* 24.3 (2012), pp. 515–554.

[24] Yuqing Ren, Robert Kraut, and Sara Kiesler. “Applying common identity and bond theory to design of online communities”. In: *Organization studies* 28.3 (2007), pp. 377–408.

[25] Thomas Richardson, Peter J. Mucha, and Mason A Porter. “Spectral tripartitioning of networks”. In: *Physical Review E* 80.3 (2009), p. 036111.

[26] John Scott and Peter J. Carrington. *The SAGE Handbook of Social Network Analysis*. SAGE publications, 2011.

[27] Herbert A. Simon. *The Architecture of Complexity*. Springer, 1991.

[28] Riitta Toivonen et al. “The role of edge weights in social networks: modelling structure and dynamics”. In: *SPIE Fourth International Symposium on Fluctuations and Noise*. International Society for Optics and Photonics. 2007, 66010B–66010B.

[29] Rongjing Xiang, Jennifer Neville, and Monica Rogati. “Modeling relationship strength in online social networks”. In: *Proceedings of the 19th International Conference on World Wide Web*. ACM. 2010, pp. 981–990.

[30] Wayne W Zachary. “An information flow model for conflict and fission in small groups”. In: *Journal of anthropological research* (1977), pp. 452–473.

[31] W-X Zhou et al. “Discrete hierarchical organization of social group sizes”. In: *Proceedings of the Royal Society B: Biological Sciences* 272.1561 (2005), pp. 439–444.