Networks and Community Detection

Joshua Mankelow

April 8, 2022

Abstract

Networks are an example of a rigorous model for analysing and understanding real world complex systems. A very important quality of these network models is the naturally emergent community structure. Community detection allows us to identify clusters in the network that are well connected amongst eachother. If a network is being used to model a real world system then finding this structure has many implications about the behaviour of the system such as ... [WANT TO FIND EXAMPLES BUT HAVEN'T LOOKED AT ENOUGH PAPERS YET]. In this essay I will discuss multiple methods for community detection in networks and their applications to the analysis and understanding of real world complex systems.

Contents

Todo list

KEY: NOT STARTED

KEY: IN PROGRESS

KEY: DRAFT FINISHED

KEY: IN REVIEW

KEY: COMPLETED

KEY: QUESTION

1 Introduction to Networks

SEC: Introduce Network Theory

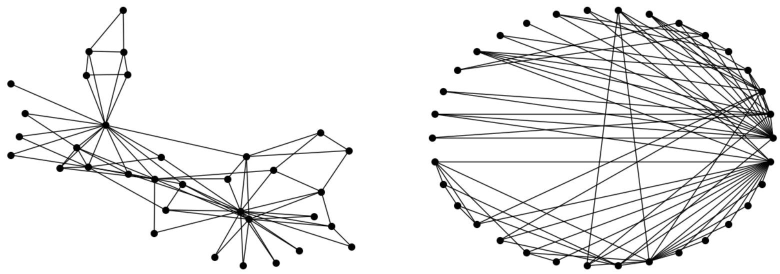
Networks are considered as the combination of two separate objects - a set of nodes (vertices) and a set of links (edges) that connect nodes. The idea is to define a structure that can represent a set of things and how they're connected amongst each other. It turns out that this idea is invaluable for modeling real world systems. Examples of such real world systems include *Technological Networks*, *Social Networks*, *Information Networks* and *Biological Networks* as different systems that are modelled by a network.[New10, Contents] A brief example of a network would be something like the following: Imagine you and a number of people you speak to regularly are represented as dots (nodes or vertices) on a piece of paper. Then if any two people are friends, the dots representing those people are connected by a line (edge). If you then repeat this process by asking your acquaintances to list all their friends and so on, you will end up with a simple model of a *social network*.

Now that we have this model, it is easy to identify and detect any natural structure that emerges which we can then use to develop an understanding of the behaviour of the real world system that the network represents. The structure that I will explore in this essay is that of *communities*. Generally speaking, communities are subsets of a network that are *densely connected* amongst themselves. I.e. there is some notion of any node within a community being more closely connected to other nodes in the community than nodes outside the community in the average case. Before we dive into the details of communities and detecting them, I wish to provide some motivation by way of example of the kinds of situations that networks can arise and why they are the natural model for the related systems.

1.1 Social Networks

SEC: Models of Networks

To better illustrate the simple notion of a social network mentioned above, I will introduce the canonical community detection example of *Zachary's Karate Club*. Zachary's Karate Club is a dataset where "The data was collected from the members of a university karate club by Wayne Zachary in 1977. Each node represents a member of the club, and each edge represents a tie between two members of the club."[kon17, Metadata]. In Figure 1, there are two different renderings of the Zachary Karate Club. Figure 1a shows the network rendered using a "spring" layout (which is a type of force directed graph drawing[Kob12])and figure 1b shows the network rendered using a "circle" layout. These different layouts show us different parts of the underlying structure of the network. For example, in Figure 1a, it is clear which nodes in the network have the highest degree and which are of lower degree. It also allows you to see some of the community structure in the network. Meanwhile, in Figure 1b, it is much easier to see the which nodes edges in the network would need to be removed to disconnect the network in a minimal way. The reason this dataset is the canonical example of community detection is that the question that comes with it is the following: Suppose two members of the club have a disagreement which causes the club to split in two. How does the club split? In Zachary's original paper on the topic *An Information Flow Model for Conflict in Small Groups*[Zac77] he uses community detection techniques to predict how the network will split after the disagreement. Out of 34 people, Zachary correctly predicts how 33 of



(a) Spring Layout (b) Circle Layout

Figure 1: Two renderings of the Zachary Karate Club network using data from KONECT.cc[Kun13] and a Python library NetworkX[HSS08]

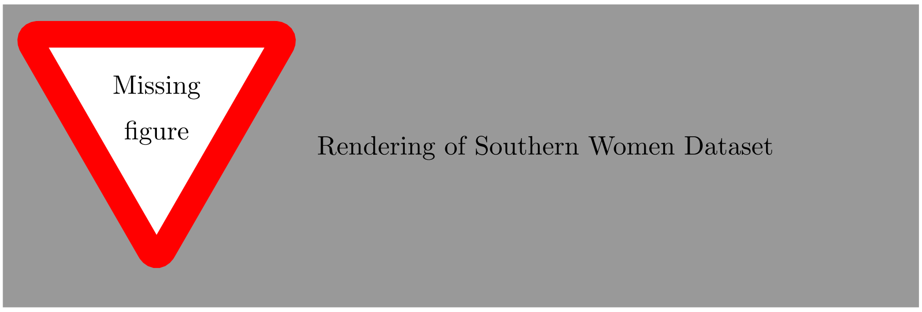


Figure 2: A rendering of the Southern Women Dataset

them will choose a side after the disagreement.

There, of course, exist different ways to represent social networks. The way in which you choose to represent them depends on the question you are trying to answer. For example, one might imagine having two types of nodes in a network. One type of node will represent a person and another type of node will represent an event. An edge is drawn between a person and an event if a person attended a given event and person A is considered connected to person B if they both attended the same event. One such example of this is the *Southern Women Dataset*.[DGG41] This dataset is another example of a community detection problem because after analysis of the data, it was found that women in the group were split into two discrete subgroups.

1.2 Technological Networks

As a result of our intensely and digitally connected world, technological networks are of significant interest to researchers. The easiest example to consider is the Internet. The internet consists of many computers all connected by copper or fibre cables which signals are sent through to transmit data. As one might imagine, in the model, the computers are nodes and the cables are the edges. The internet needs to be robust against software and hardware failures and this

Figure 3: A rendering of the Internet

is where the idea of commnity detection can help us. Saying that we want the internet to be robust is the same as saying that we want every node in the network to be strongly connected to every other node i.e. the number of possible routes between any two nodes is large. This means that, in the philosophy of community detection, we want the internet to act as one large community rather than multiple smaller communities that are loosley connected. An alternative way of looking at this is that once we've managed to identify the communities, we can then figure out which edges and noes are the critical ones that allow passage from one community to another. This allows us to reinforce those edges and nodes to reduce the potential for failure.

Yet another example of a technological network would be the UK Power Grid. Network theory is a useful model here as with the UK Power Grid we're trying to solve exactly the same problem as with the internet — we want the system to be robust against hardware or software failures.

1.3 Information Networks

The most accessible example of an information network is that which is generated by looking back through the citations on a paper recursively. If Paper A references Paper B, then we will draw a directed edge connecting Paper A to Paper B. This will generate a network that shows which papers are referenced by which other papers and how information is reused. Applying community detection to such a network would show us the different academic working groups and perhaps even different fields or subfields of a subject.

Another example of an information network is the World Wide Web which differs from the internet in that it refers to the webpages hosted on the internet rather than the servers and cables themselves. Mapping the world wide web as a network shows us communities of websites that regularly reference each other.

2 Properties of Networks

SEC: Definition of a Network

Community detection relies on us knowing lots about the underlying structure of a network and to do that we have to understand its properties. This chapter will establish a more formal understanding of networks and will highlight some key properties and methods that we will use to exctract value about community structure later.

Definition 1. *(Undirected network) Let V be a set of vertices (nodes) and let E be a set of pairs of vertices such that if $e = (x, y) \in E$ then $x, y \in V$. An undirected network is the pair $(V, E) = N$. An edge $e = (x, y) \in E$ is said to join x and y and y to x.[Lam21, 1]*

The undirected network is the simplest type of network and on its own has interesting enough properties. However, for the sake of example and application, we will also introduce some other types of network that allow for more detailed models.

SEC: Different Types of Network

Definition 2. *(Directed network) Let V be a set of vertices (nodes) and let E be a set of pairs of vertices such that if $e = (x, y) \in E$ then $x, y \in V$. A directed network is the pair $(V, E) = N$. An edge $e = (x, y) \in E$ is said to join x to y. I.e. if x is joined to y then y is not necessarily joined to x.[Lam21, 1]*

The intuition for directed graphs, is that edges may only be travelled along in one way. This comes in handy for modelling more intricate systems. The final network type of interest is that of the weighted network.

Definition 3. *(Weighted network) Let V be a set of vertices (nodes) and let E be a set of triples of the form $V^2 \times \mathbb{R}$ such that if $e = (x, y, w) \in E$ then $x, y \in V$. The value w is said to be the weight of the edge.[Lam21, 1]*

The weighted network allows us to introduce some notion of how hard it is to move along a certain edge. This is useful when modeling things like traffic flow. [citation needed]

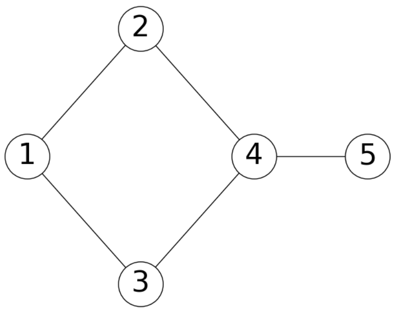
The above definitions of a network are likely more technical than we will ever need because once we have introduced the notion of of an adjacency matrix, that becomes our go to representation of a network.

2.1 Adjacency Matrices

SEC: Interesting Properties of Networks

The objects defined above are meaningless without a rigorous way of mathematically representing them. To that end, we have to come up with a way of describing a network mathematically. This leads us to the definition of the adjacency matrix:

Definition 4. *(Adjacency matrix) Let $N = (V, E)$ be a network and label every vertex $v \in V$ with a number from 1 to $n = |V|$. The adjacency matrix of a network is the matrix of elements $(A)_{ij}$ such that $a_{ij} = 1$ if $(i, j) \in E$ and $a_{ij} = 0$ if $(i, j) \notin E$. In other words, if nodes i and j are connected by an edge in the network, then the corresponding element in the matrix is 1. Otherwise, it is 0.[New10, 111]*



(a) Simple graph

$$\begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

(b) Adjacency matrix

Figure 4: A simple network and its adjacency matrix

The adjacency matrix gives us our first way of representing a network. Figure 4 shows a basic example of a network and its associated adjacency matrix. This will form the basis for most of the analytical work we do going forwards. It's worth noting that there are also different types of adjacency matrix corresponding to the different types of network. For example, in the case of a directed network we will have a non-symmetric matrix where $a_{ij} = 1$ if $(i, j) \in E$ but this does not necessarily mean that $a_{ji} = 1$. We also get something similar for weighted networks where we set $a_{ij} = w$ where w is the weight of the edge connecting i and j in N.

2.2 The Network Laplacian

The Network Laplacian is a simple extension of the adjacency matrix with more interesting properties.

Definition 5. *(Network Laplacian) The Laplacian of a network $N = (V, E)$, denoted by L is given by the following:*

$$L = D - A$$

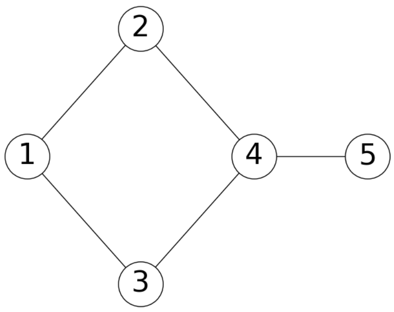
where A is the adjacency matrix of the network and D is a diagonal matrix containing the degrees of each vertex in the network such that $d_{ii} = deg(v_i)$ and $d_{ij} = 0$ if $i \neq j$.

Figure 5 shows the same simple graph as before and it's Laplacian matrix.

2.3 Paths

When we're analysing a network, we're very often interested in which vertices are reachable from any given vertex. As such, we become interested in the idea of a path. A path in a network is defined in the following way

Definition 6. *(Path) Let $N = (V, E)$ be a network. A path is a sequence of vertices $v_1, \ldots, v_n \in V$ such that $(v_i, v_{i+1}) \in E$ for all $i = 1, \ldots, n-1$. In other words, a path is a sequence of vertices such that every consecutive pair of vertices is connected by an edge in E. We say that the length of a path is the*



(a) Simple graph

$$\begin{pmatrix} 2 & -1 & -1 & 0 & 0 \\ -1 & 2 & 0 & -1 & 0 \\ -1 & 0 & 2 & -1 & 0 \\ 0 & -1 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{pmatrix}$$

(b) Laplacian

Figure 5: A simple network and its Laplacian

number of edges (v_i, v_{i+1}) *that are traversed by the path. Note that under this definition, we may pass through each vertex in the network more than once.*

Paths are an important concept in community detection as they allow us to phrase questions in rigorous terms as opposed to loose concepts of connectedness. Paths also give us our first look into the usefulness of the adjacency matrix. Using the adjacency matrix, it is very simple to determine whether there exists a path between two vertices i and j. Paraphrasing Newman [New10, 137], suppose our adjacency matrix is given by A. If i and j are directly connected then $A_{ij} = 1$ and we are done. If $A_{ij} = 0$ then pick some k such that $A_{ik} = 1$. Then it is simple to see that if $A_{kj} = 1$ then $A_{ik}A_{kj} = 1$ which implies that i and j are connected via k. In fact, we can even go so far as to calculate the total number of ways to draw a path of length two between i and j, $N_{ij}^{(2)}$, in the following way:

$$N_{ij}^{(2)} = \sum_{k=1}^{n} A_{ik}A_{kj} = [A^2]_{ij}$$

where $[\cdot]_{ij}$ denotes the (i, j)-th element of the given matrix. Clearly, this process actually generalises to paths of arbitrary length r and we can see that

$$N_{ij}^{(r)} = [A^r]_{ij}$$

Also note that this solution counts each path but going in opposite directions. For example, you might have a path going $1 \to 4 \to 5 \to 2 \to 1$ which will also get counted separately by this method as the following $1 \to 2 \to 5 \to 4 \to 1$. This result isn't very useful, but it goes to show that the adjacency matrix we introduced before is useful and provides insight about the structure of our network. We call a path that starts and ends at the same place a loop and we can actually calculate the number of loops of length r using the spectral properties of the adjacency matrix. Paraphrasing Newman again [New10, 137], our adjacency matrix A can be written as $A = UDU^T$ because A is symmetric meaning that it has n real and non-negative eigenvalues with real valued eigenvectors. In this form, U is our matrix of eigenvectors and D is the diagonal matrix containing the eigenvalues. We know that $A^r = (UKU^T)^r = UK^RU^T$ and then the number of loops is given by

$$
\begin{aligned}
L_r &= \mathrm{Tr}(UK^rU^T) = \mathrm{Tr}(U^TUK^r) = \mathrm{Tr}(K^r) \\
&= \sum_i k_i^r
\end{aligned}
$$

where k_i is the i-th entry of the matrix K. There exist analogous results for all the different types of networks which Newman discusses further. [New10, 138]. Typically, we are interested in types of path known as *geodesic paths.*

Definition 7. *(Geodesic Path) A geodesic path (more commonly referred to as a shortest path) is a path through a network such that no shorter path exists.*

Geodesic paths are more interesting than general paths as they are necessarily self-avoiding as any time a path intersects with itself it adds unnecessary length. Geodesic paths are also used to define some other properties of networks such as the *diameter.*

2.4 Components

Components are a natural consequence of the notion of paths. Simply put, components are sets of vertices in the graph that are all connected to each other via paths.

Definition 8. *(Component) A component is a subset C, of the vertex set V such that if $v_1, v_2 \in C$ then v_1 and v_2 are connected by a path. Furthermore if $v_3 \notin C$ then v_3 is not connected to either v_1 or v_2 by any path.*

Components are an important concept in the study of community detection. Recall the intuition for a community introduced in section 1. From here, it is clear to see a similarity between the notion of a community and that of a component. Loosely put, a community is a subset of a network that is *nearly* a component.

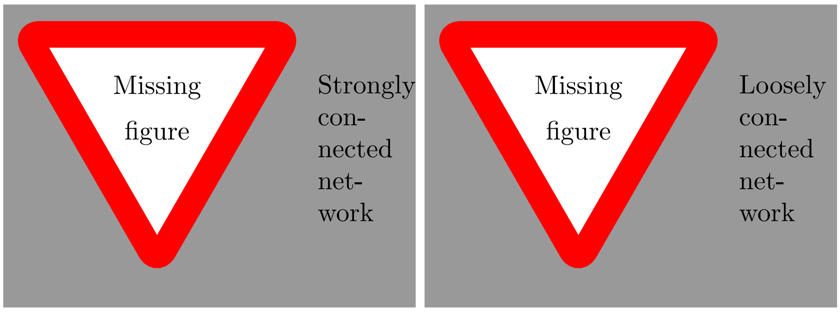
This section probably need replacing with one on *partitioning* and *clustering*

2.5 Cut Sets

Cut sets, as the name would suggest are sets of vertices that cut the network into multiple components.

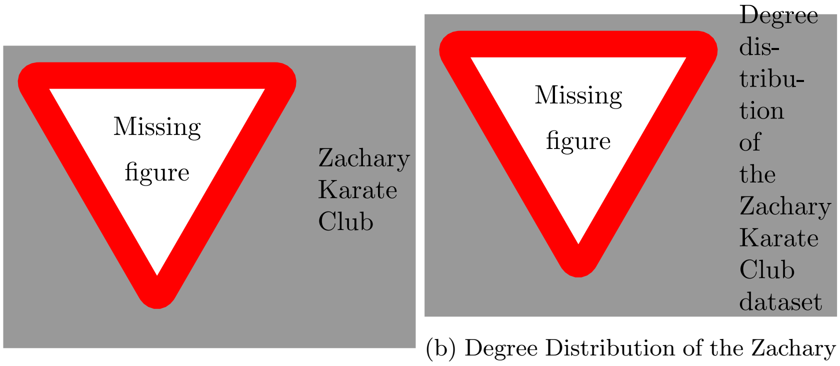
Definition 9. *(Cut Set) A cut set is a subset, $\mathcal{C}$, of the edge set, E, such that the network $N = (V, E \setminus \mathcal{C})$ has more than one component.*

We also have the notion of a minimum cut set which is a cut set of minimum cardinality. I.e. it's the smallest subset of the vertices that can be removed which will disconnect the network into two components. Similarly to components, minimum cut sets are also important in the analysis of communities as the size of the minimum cut gives us some notion of how strongly connected our network is. A larger cut set means that we require more edges to be removed from the network to get multiple connected components. This suggests that a network with a larger minimum cut is more strongly connected and a network with a smaller minimum cut is more loosely connected. An example of such structure can be seen in Figure 7.



(a) A strongly connected network (b) A loosely connected network

Figure 6: Two networks with differing connectedness



(a) The Zachary Karate Club dataset (b) Degree Distribution of the Zachary Karate Club dataset

Figure 7: A network and it's associated degree distribution

2.6 Degree Distribution

We already know that the degree of a node is the number of edges incident on that node. In the context of adjacency matrices, the degree of node i is defined by

$$\deg(i) = d_i = \sum_{j=1}^{N} A_{ij}$$

There also exist similar definitions for both weighted and directed networks. Using this definition of the degree of a matrix, we can define something called the *degree distribution*. The degree distribution, as the name suggests, is a frequency distribution of all the degrees in the network. This distribution is often denoted by the function $p(k)$.

According to Renaud Lambiotte, these degree distributions often have very long tails which are regularly described by a power law. i.e.

$$p(k) \propto k^{-\gamma}$$

where γ usually takes values between 2 and 3.[Lam21, 16] The relationship above holds approximately until some "cutoff degree" where the structure changes and $p(d)$ quickly decreases to 0. Curiously, this leads us to a formalisation of the friendship paradox wherein the average number of friends of any given node is less than the average degree of nodes adjacent to any given node.

in this section I'm finding it really hard not to just copy from Renaud's notes.

2.7 Measures Derived from Walks and Paths

I think I'll probably leave this one out. It's not very interesting.

2.8 Clustering coefficient

When thinking about community detection, we're actually interested in the connectedness of different parts of the network and in particular we're interested in the interconnectedness of a set of vertices. One way to measure the interconnectedness of a vertex with the surrounding nodes is using the clustering coefficient. The clustering coefficient counts the number of triangles in the network that include a given vertex. We define the clustering coefficient in the following way

$$C_i = \frac{\text{number of triangles including the } i\text{th node}}{k_i(k_i-1)/2}$$

This quantity measures and normalises the number of triangles in the immediate vicinity of the vertex i. Using the clustering coefficient, we can extend this to consider the whole network.

$$C = \frac{1}{N}\sum_{i=1}^{N} C_i$$

Again this total measure of clustering is normalised such that $0 \leq C \leq 1$.

2.9 Centrality

Different measures of centrality aim to convey the importance of certain nodes in the network. In this section, I will introduce some examples from Renaud Lambiotte's notes.[Lam21, 19]

2.9.1 Closeness Centrality

The closeness centrality is the inverse of the mean distance between a node i and every other node in the network. Notationally, this looks like the following

$$\text{closeness}_i = \frac{N-1}{\sum_{j=1;j\neq i}^{N} d(i,j)}$$

where $d(i,j)$ is the smallest number of moves from one node to another required to reach j from i.

2.9.2 Betweenness Centrality

A slightly more complex measure of centrality is the betweenness centrality. This measures each node's contribution to the number of shortest paths that exist in the network.

$$\text{betweenness}_i = \frac{2}{(N-1)(N-2)} \sum_{j=1;j\neq i}^{N} \sum_{l=1;l\neq i}^{j-1} \frac{\sigma_{jl}^i}{\sigma_{jl}}$$

where σ_{jl} denotes the total number of shortest paths connecting nodes j and l and σ_{jl}^i is the number of such paths containing the node i.

2.9.3 Katz Centrality

The Katz measure of centrality considers all walks between two nodes i and j, but gives each one less weighting as it increases in length by scaling it by a constant $\alpha \in (0, 1)$. The Katz centrality of a node i is defined by

$$\text{Katz}_i = \sum_{i=1}^{N} [(I - \alpha A)^{-1}]_{ij}$$

2.9.4 PageRank

PageRank is a centrality measure developed with the advent of the internet in an attempt to improve search engine indexing. The actual algorithm for calculating the PageRank of a node is too intricate to go into detail in this essay, but the technical definition is the "stationary density of a discrete-time random walk."[Lam21, 19]

CITATION NEEDED

2.10 Spectral Properties

The final properties of interest are spectral in nature. Spectral properties are properties that are based on the eigendecomposition of either the adjacency matrix, the Laplacian or a modified version of the Laplacian called the normalised Laplacian. If the Laplacian is defined by

$$L = D - A,$$

Then the normalised Laplacian is given by

$$\tilde{L} = D^{-1/2} L D^{-1/2} = I - D^{-1/2} A D^{-1/2}.$$

By definition, the adjacency matrix, the Laplacian and the normalised Laplacian are symmetric. This means that the eigenvalues of each matrix are all real and the corresponding eigenvectors form an orthonormal basis. It's important to note that the two Laplacian matrices always have $\lambda_1 = 0$. In fact, if the network is connected, $\lambda_1 = 0$ and $\lambda_i > 0 \ \forall i > 1$. This is our first spectral property of a matrix.

3 Community Detection

SEC: Introduction to Community Detection

As allured to in the previous chapters, detecting communities is of great interest and as such there a number of ways to do it. The process of community detection involves analysing the network and finding groups of nodes in the network that are more densely connected amongst themselves than they are to the rest of the network. This notion forms the basis for most community detection methods and algorithms. However, it turns out that it's difficult to come up with a good definition of a community. In fact, it turns out that "In most cases, communities are algorithmically defined, i.e. they are just the final product of the algorithm without a precise *a priori* definition." [For10, 84] In this section, we will discuss the notions of "community" that underpin a number of interesting methods in community detection before going into detail about the algorithms themselves.

SEC: Background for Community Detection

A simple example of the aforementioned notion comes in the form of inter and intra-cluster densities as discussed by Fortunato.[For10, 84]. This idea assumes that we have a network N and a subnetwork $C \subseteq N$ i.e. C is also a network where every node and edge in C is also in N, but the converse is not necessarily true. Suppose we want to determine whether C is a community inside N. To aid us in our investigation, we can define the following two quantities,

1. Intra-cluser density: $\delta_{\text{int}}(C) = \frac{\text{\# of internal edges of C}}{n_c(n_c-1)/2}$,
2. Inter-cluser density: $\delta_{\text{ext}}(C) = \frac{\text{\# of external edges of C}}{n_c(n_c-1)/2}$,

where $n_c = |C|$ and internal and external edges refer to edges originating in C that end inside and outside C respectively. Thus the intra-cluster density is the ratio of edges originating in C that remain in C whilst the inter-cluster density is the ratio of edges originating in C that end outside C. Finally, to make sense of these metrics, we introduce a fiducial marker for community structure, the average link density,

$$\delta(N) = \frac{\text{\# of edges in } N}{n(n-1)/2}.$$

Following the same intuition as before, this quantity represents the ratio of edges that are actually present in the network agains the total number of possible edges. Now, if C were a community, we would expect that $\delta_{\text{int}}(C)$ is noticably larger than $\delta(N)$. Similarly, we would expect $\delta_{\text{ext}}(C)$ to be noticably smaller than $\delta(N)$. Getting this result is the goal of most community detection algorithms.

Stricter definitions of communities come in three flavours: local definitions, global definitions and definitions based on vertex similarity. To summarise: local definitions rely on considering the structure of a given subnetwork and perhaps it's immediately adjacent neighbours; global definitions consider the structure of the whole network; and degree similarity relies on some notion of similarity between any two vertices. Each method seeks to formalise our intuition that communities should be strongly connected amongst themselves, but weakly connected to the rest of the network. To extend this intuition, we will refer to Wasserman's summary of the four general properties of what he calls "cohesive subgroups" (but we call communities) that have influenced the formalisations and definitions of the concept in the social network literature.[WF94, 251-252] The following are reprinted verbatim:

1. The mutuality of ties
2. The closeness or reachability of subgroup members
3. The frequency of ties amongst members
4. The relative frequency of ties amongst subgroup members compared to non-subgroup members

The four points above are written in the parlance of the Social Network Analysis literature. As such, we will reprint them using terms more in line with our context:

1. Cliques
2. Closeness or reachability of community members
3. Frequency of connections between community members
4. Relative frequency of connections amongst community members compared to non-community members.

Each of these points refers to a different strategy for identifying communities. Paraphrasing Wasserman and using our terminology: Strategies based on cliques require each member of a community to be directly adjacent to each other member of a community; strategies based on closeness or reachability require that each member of a community is reachable from every other member of the same community, but adjacency is not required; strategies based on frequency of connetions between community members require that each member of the community is adjacent to many other members of the community; and strategies based on the relative frequency of connections require that members of the community are more connected amongst each other than they are to the rest of the network.

These four ideas for definitions are such that the communities they generate are considered maximal subnetworks i.e. adding another node to the subnetwork would remove the subnetwork's community propery. The ideas are also ordered (roughly) in decreasing strictness and Fortunato provides an excellent summary of the journey from a rudimentary strategy using cliques to strategies involving fitness measures in the case of local definitions.[For10, 3.2.2]

Can I cite sections like this?

Global definitions, of course, follow the same notion, but they have a different motivation. Whilst local definitions are used when we're interested in just the structure of the subnetwork, global definitions are used when the community structure doesn't make sense without the context of the rest of the network and vice versa. Global definitions are more often indirect definitions in the sense that the communities are defined by the output of an algorithm implementing a global method. Curiously though, there are a set of direct global definitions. These definitions are called *null model* definitions. These definitions rely on taking the original network and generating a random version of it that has some similarity in it's structural features to the original. These two networks are then compared in some way which will reveal any underlying community structure. In fact, these null model methods underpin the idea of modularity; a concept which underpins the most popular method of network clustering.

3.1 Quality Functions and Basic Modularity

It's all well and good talking about which methods exist, but ultimately we're interested in the efficacy of a given method. Anyone can define an algorithm that breaks a network down into communities (for example by putting each node into it's own community), but some of these might not provide us with high quality insight to the structure of a network. Hence we develop the idea of *quality functions*. A community detection algorihm takes a network N and returns a set of partitions and/or clusters in that network based on a set of rules. A quality function determines the quality of an algorithm by assigning a number to each partition of a network. Typically, partitions with a high score are considered "good". According to Fortunato, the most used quality function is Nerman and Girvan's definition of modularity[NG04, 8]:

$$Q = \sum_i (e_{ii} - a_i^2) = \mathrm{Tr}(e) - ||e^2||$$

This definition relies on constructing a matrix e such that e_{ij} is the fraction of edges in the network that link a node in community i to a node in community j, $a_i = \sum_j e_{ij}$ the fraction of edges that connect to vertices in community i, and finally $||X||$ denotes the sum of all elements of the matrix X. Based on this definition, if our algorithm has partitioned the network successfully into communities then we expect Q to be large. Even though $\mathrm{Tr}(e)$ is large when the communities are very well connected, it achieves a maximum of $\mathrm{Tr}(e) = 1$ when all the nodes are in the same community. This is clearly not useful as it doesn't give us any information about the structure of the networks. Thus we subtract $||e^2||$ from $\mathrm{Tr}(e)$. To quote Newman and Girvan: "this quantity measures the fraction of edges in the network that connect vertices of the same type minus the expecte value of the same quantity in a network with the same community divisions but with random connections between them". The idea for this definition is to use a *null model* and exploit the fact that random networks are not expected to have any community strucutre.

SEC: Methods of Community Detection

3.2 Community Detection via Smallest Cut

The simplest way to consider community detection is to think about dividing the network into subnetworks such that the number of edges between all the subnetworks is minimised. This is called a smallest cut. Referring again to Renaud's notes, we will define a few pieces of machinery that will allow us to algorithmically/analytically find the set of edges that separates the communities.[Lam21, 26-27] For simplicity, we restrict ourselves to trying to identify only two communities. Let us assume that we have partitioned the graph into two groups labelled 1 and 2. We then define the number of edges starting in group 1 and ending in group 2 or vice versa by the following quantity:

$$R = \frac{1}{2} \sum_{\substack{i,j \text{ in} \\ \text{different} \\ \text{groups}}} A_{ij}.$$

Of course, this isn't very easy to work with, so we define a vector s such that

$$s_i = \begin{cases} +1 & \text{if vertex } i \text{ belongs to group 1,} \\ -1 & \text{if vertex } i \text{ belongs to group 2.} \end{cases}$$

After noting the following

$$\frac{1}{2}(1 - s_i s_j) = \begin{cases} 1 & \text{if } i \text{ and } j \text{ are in different groups,} \\ 0 & \text{if } i \text{ and } j \text{ are in the same group,} \end{cases}$$

we can rewrite R as

$$\begin{aligned} R =& \frac{1}{4} \sum_{ij} (1 - s_i s_j) A_{ij} \\ =& \quad \vdots \\ =& \frac{1}{4} \sum_{ij} s_i s_j (k_i \delta_{ij} - A_{ij}) \end{aligned}$$

3.3 Louvian Community Detection

3.4 Surprise Community Detection

3.5 Leiden Community Detection

3.6 Walktrap Community Detection

4 Applications of Community Detection

SEC: Applications of Community Detection

SEC: Figure out an interesting thing to write some of my own code for

5 Conclusion

References

[DGG41] Allison Davis, Burleigh B. Gardner, and Mary R. Gardner. *Deep South; a Social Anthropological Study of Caste and Class.* The Univ. of Chicago Press, 1941.

[For10] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3):75–174, 2010.

[HSS08] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using networkx. In Gaël Varoquaux, Travis Vaught, and Jarrod Millman, editors, *Proceedings of the 7th Python in Science Conference*, pages 11 – 15, Pasadena, CA USA, 2008.

[Kob12] Stephen G. Kobourov. Spring embedders and force directed graphs. 2012.

[kon17] Zachary karate club network dataset – KONECT, October 2017.

[Kun13] Jérôme Kunegis. KONECT – The Koblenz Network Collection. In *Proc. Int. Conf. on World Wide Web Companion*, pages 1343–1350, 2013.

[Lam21] Renaud Lambiotte. C5.4 networks lecture notes. 2021.

[New10] Mark Newman. *Networks: An Introduction.* 2010.

[NG04] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Phys. Rev. E*, 69:026113, Feb 2004.

[WF94] Stanley Wasserman and Katherine Faust. *Social Network Analysis: Methods and Applications.* Structural Analysis in the Social Sciences. Cambridge University Press, 1994.

[Zac77] Wane W. Zachary. An information flow model for conflict in small groups. 1977.