

# Networks and Community Detection

Joshua Mankelow

March 16, 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.

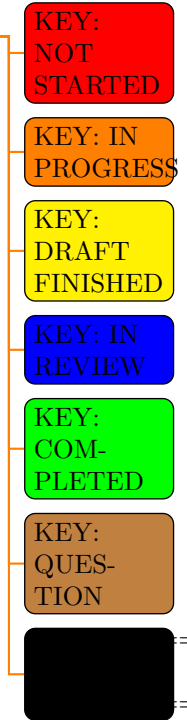
## Contents

<b>1</b>	<b>Introduction to Networks</b>	<b>4</b>
1.1	Social Networks . . . . .	4
1.2	Technological Networks . . . . .	5
1.3	Information Networks . . . . .	6
<b>2</b>	<b>Properties of Networks</b>	<b>7</b>
2.1	Adjacency Matrices . . . . .	7
2.2	The Network Laplacian . . . . .	8
2.3	Paths . . . . .	8
2.4	Components . . . . .	10
2.5	Cut Sets . . . . .	10
2.6	Degree Distribution . . . . .	11
2.7	Measures Derived from Walks and Paths . . . . .	12
2.8	Clustering coefficient . . . . .	12
2.9	Centrality . . . . .	12
2.9.1	Closeness Centrality . . . . .	12
2.9.2	Betweenness Centrality . . . . .	13
2.9.3	Katz Centrality . . . . .	13
2.9.4	PageRank . . . . .	13
2.10	Spectral Properties . . . . .	13
<b>3</b>	<b>Community Detection</b>	<b>14</b>
<b>4</b>	<b>Applications of Community Detection</b>	<b>15</b>

## Todo list

	KEY: NOT STARTED . . . . .	3
	KEY: IN PROGRESS . . . . .	3
	KEY: DRAFT FINISHED . . . . .	3
	KEY: IN REVIEW . . . . .	3
	KEY: COMPLETED . . . . .	3
	KEY: QUESTION . . . . .	3
	===== DIVIDER =====	3
	SEC: Introduce Network Theory . . . . .	4
	SEC: Models of Networks . . . . .	4
	Figure: Rendering of Southern Women Dataset . . . . .	5
	Figure: Rendering of Some Internet Dataset . . . . .	6
	SEC: Definition of a Network . . . . .	7
	SEC: Different Types of Network . . . . .	7
	SEC: Interesting Properties of Networks . . . . .	7
	Figure: Strongly connected network . . . . .	11
	Figure: Loosely connected network . . . . .	11
	Figure: Zachary Karate Club . . . . .	11
	Figure: Degree distribution of the Zachary Karate Club dataset . . . . .	11
	in this section I'm finding it really hard not to just copy from Renaud's notes. . . . .	12
	CITATION NEEDED . . . . .	13

SEC: Introduction to Community Detection . . . . .	14
SEC: Traditional Methods of Community Detection . . . . .	14
SEC: Spectral Methods of Community Detection . . . . .	14
SEC: Applications of Community Detection . . . . .	15
SEC: Figure out an interesting thing to write some of my own code for	15



# 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

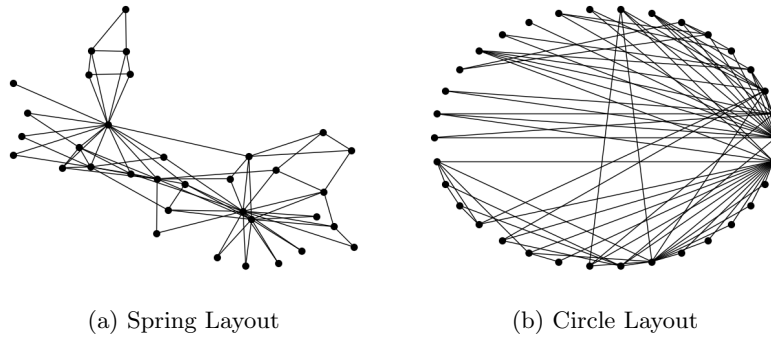


Figure 1: Two renderings of the Zachary Karate Club network using data from KONECT.cc[Kun13] and a Python library NetworkX[HSS08]

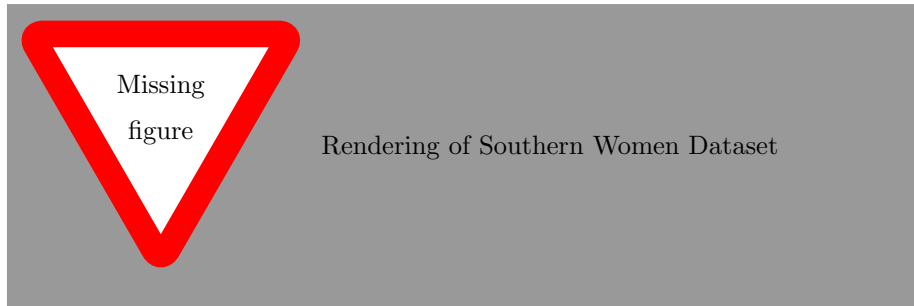


Figure 2: A rendering of the Southern Women Dataset

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

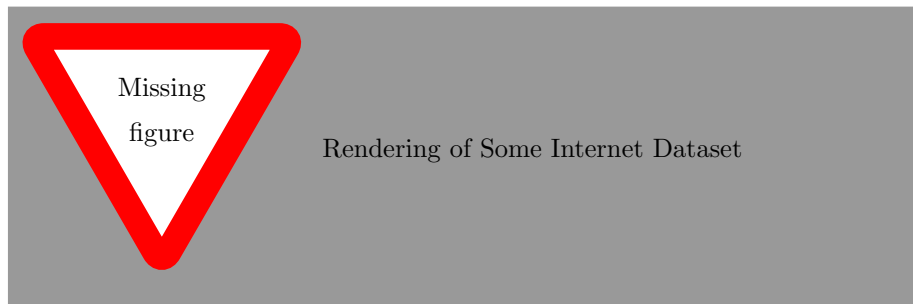


Figure 3: A rendering of the Internet

is where the idea of community 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 loosely 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 nodes 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

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 extract value about community structure later.

SEC: Definition of a Network

**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]

SEC: Interesting Properties of Networks

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

### 2.1 Adjacency Matrices

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]

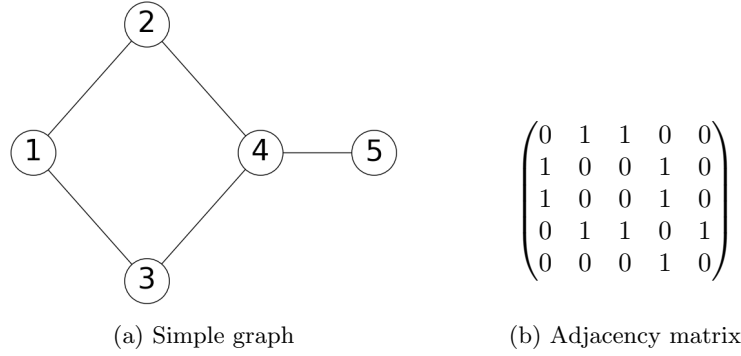


Figure 4: A simple network and its adjacency matrix

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 its 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  $v_1, \dots, v_n \in V$  such that  $(v_i, v_{i+1}) \in E$  for all  $i = 1, \dots, n-1$ . 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



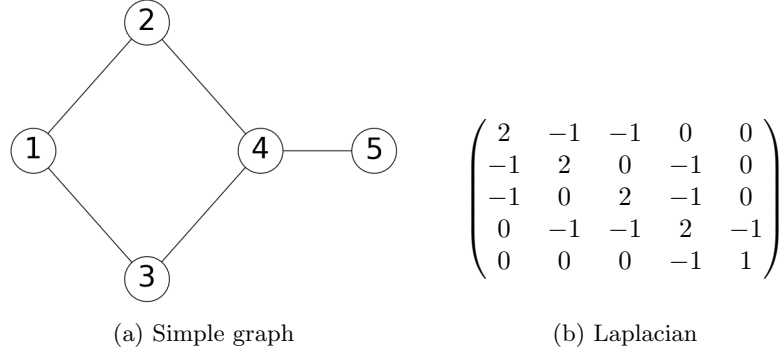


Figure 5: A simple network and its Laplacian

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  $1 \rightarrow 4 \rightarrow 5 \rightarrow 2 \rightarrow 1$  which will also get counted separately by this method as the following  $1 \rightarrow 2 \rightarrow 5 \rightarrow 4 \rightarrow 1$ . 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

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

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 very important concept when we're considering community detection as communities are, in some sense, almost clusters

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.

## 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.

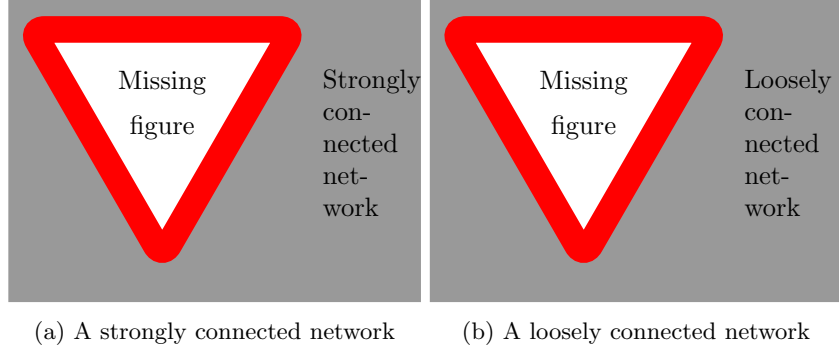


Figure 6: Two networks with differing connectedness

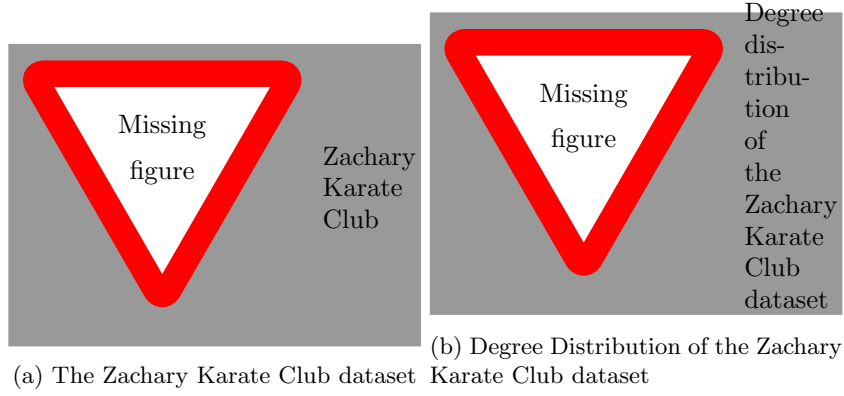


Figure 7: A network and its associated degree distribution

## 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  $\gamma$  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.

### 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  $\sigma_{jl}$  denotes the total number of shortest paths connecting nodes  $j$  and  $l$  and  $\sigma_{jl}^i$  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

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

### 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

### 3 Community Detection

---

SEC: Introduction to Community Detection

SEC: Traditional Methods of Community Detection

SEC: Spectral Methods of 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

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