Network Science

Kexing Ying

May 15, 2020

Contents

1 Fundamental Graph Theory

1.1 Basic Concepts

We begin with a few basic definitions.

Definition 1.1 (Graph)**.** A graph G is a tuple $(V(G), E(G))$ equipped with a function $\sim: E(G) \to V(G) \to V(G) \to \text{Prop}$ where $V(G)$ is the *vertex set*, $E(G)$ is the *edge set* and for all $e \in E(G)$ there exists a unique pair $v_1, v_2 \in V(G)$ such that $v_1 \sim_e v_2$. We write $v_1 \sim_e v_2$ as a short hand for $\sim (e, v_1, v_2) = \text{true}$.

Note that this definition works for both directed and undirected graphs as by this definition, a undirected graph is a directed graph with the condition that for all $e \in E(G)$, $\sim_e$ is symmetric.

Definition 1.2 (Subgraph)**.** Let G be a graph, then H is a subgraph of G if and only if H is a graph such that $V(H) \subseteq V(G)$, $E(H) \subseteq E(G)$ and the restriction $\sim^G|_H = \sim^H$. We will write $H \leq G$ for H is a subgraph of G.

Definition 1.3 (Loop)**.** Let $G := (V(G), E(G))$ be a graph and $e \in E(G)$ be an edge. We say e is a loop at some $v \in V(G)$ if and only if $v \sim_e v$.

Definition 1.4 (Multiple Edges)**.** Let $G := (V(G), E(G))$ be a graph and $e, f \in E(G)$ be edges. We call e, f be multiple edges if and only if there exists $v_1, v_2 \in V(G)$ such that $v_1 \sim_e v_2$ and $v_1 \sim_f v_2$.

Definition 1.5 (Simple)**.** We call a graph simple if it contains no loops nor multiple edges.

If a graph is simple we can then model the edge set of the graph $E(G)$ by a set of unordered tuples where each edge e with end points v_1, v_2 can be uniquely represented by $e = v_1 v_2$ (commutative if and only if G is undirected).

Definition 1.6 (Complete Graph)**.** A graph G is complete if and only if G is simple and every vertex is adjacent to every other vertices, i.e. $E(G) = \{v_i v_j \mid v_i, v_j \in V(G), i \neq j\}$.

Definition 1.7 (Adjacent)**.** Let $G := (V(G), E(G))$ be a graph and $v_1, v_2 \in V(G)$, then v_1 and v_2 are adjacent (or are neighbours) if and only if there exists some edge $e \in E(G)$ such that $v_1 \sim_e v_2$.

Definition 1.8 (Path)**.** Let G be a graph, then a path in G is a simple subgraph P of G such that $V(P)$ can be ordered in a list such that consecutive vertices are adjacent. On top of this, if this ordering resulted in the first element to be adjacent to the last, then we say P is a *cycle*.

Note that we said *ordered*, so each element of $V(P)$ can only appear once in the arrangement (so no infinite loops back and forth). This can be somewhat limiting as sometimes lists with duplicate naturally arises, so let us consider the path induced by such a list.

Lemma 1. Let G be a graph, $v_1, v_2 \in V(G)$, and L a finite list of vertices of G (possibly with repeats) such that consecutive vertices are adjacent. Then v_1, v_2 are connected.

Proof. We present an algorithm to find such a path. While L contains duplicates, find the first pair of duplicates and remove every vertex in the sequence between the two duplicates including the last duplicate. This algorithm will always terminate as L is finite. □

Definition 1.9 (Connected)**.** A graph G is connected if and only if for all $u, v \in V(G)$, there exists a path P in G such that the rearrangement of P begins in u and ends in v.

1.2 Graph as Models

Definition 1.10 (Complement)**.** Let G be a simple graph, then the complement of G, $\bar{G}$ is the simple graph $(V(G), E(\bar{G}))$ where for all $u, v \in V(G)$, $uv \in E(\bar{G})$ if and only if $uv \notin E(G)$.

Note that this complement graph is unique only if we restrict it to be simple. Suppose G is simple and let $v \in V(G)$, then $vv \notin E(G)$ by the no loop condition. Thus, if we do not restrict $\bar{G}$ to be simple, then we can add how many loops as we want at v, making the complement not unique.

Proposition 0.1. Let G be a simple graph, then the complement of G is unique.

Proof. Let G_1, G_2 be complements of G. By definition $V(G_1) = V(G) = V(G_2)$ so $G_1 = G_2$ if and only if $E(G_1) = E(G_2)$. Wlog. it suffices to show that $E(G_1) \subseteq E(G_2)$. Let $uv \in E(G_1)$, then $uv \notin E(G)$ and thus $uv \in E(G_2)$. □

Let us consider a real world problem. Suppose we have n job openings and k applicants but not all applicants are qualified for all jobs. We can easily model this problem by connecting each applicants to their respective qualified jobs and ask whether we can find a subgraph that consist of n pairwise disjoint edges.

Upon examining this question, we find that this particular model has an interesting graph structure in which none of the jobs are adjacent to each other (similarly for the applicants). This type of graphs are called *bipartite* and the set vertices representing people and jobs respectively are called independent.

Definition 1.11 (Independent)**.** Let G be a graph and $S \subseteq V(G)$. S is called an independent set in G if and only if for all $u, v \in S$, $uv \notin E(G)$.

Definition 1.12 (Bipartite)**.** A graph G is called bipartite if and only if $V(G)$ is the disjoint union of two independent sets in G. We call these two independent sets the *partite* sets of G.

1.3 Graph Isomorphism

Similar to all mathematical structures, there exists a notion of isomorphisms between two graphs that are essentially the same[1]

Definition 1.13 (Isomorphism)**.** An isomorphism from a simple graph G to a simple graph H is a bijection $f : V(G) \to V(H)$ such that for all $uv \in E(G)$, $f(u)f(v) \in E(H)$. We write $G \cong H$ as usual.

Theorem 1. Graph isomorphisms is a equivalence relation.

[1] By essentially the same we mean that almost all properties of a graph commutes via an isomorphism.

As usual, theorems of the above kind is simply by checking each of the properties so we will omit it here. As an equivalence relation induces a partition, a set of graphs can be quotiented out by the isomorphism relation and we call each element of this quotient an *isomorphism class*.

As all graphs in a isomorphism class are pairwise isomorphic to one another, they all share graph structures. Therefore, when discussing graph structures, it makes sense to talk about a isomorphism class rather than a particular graph. When we do this, we will informally call it an *unlabeled* graph.

We will now introduce some notations. The unlabeled path and cycle of n vertices is denoted by P_n and C_n respectively while the complete graph of n vertices is denoted by K_n. If G is a bipartite graph with n vertices such that two vertices are adjacent if and only if they are in different partite sets, then we call G a complete bipartite graph and denote it by $K_{r,s}$ where r, s are the sizes of the partite sets.

1.4 Trees

Definition 1.14 (Minimally Connected)**.** We say a graph G is minimally connected if and only if it is connected and for all $e \in E(G)$, the graph $(V(G), E(G) \setminus \{e\})$ is not connected.

Definition 1.15 (Tree)**.** A tree is a minimally connected graph.

Lemma 2. A tree does not have any cycles.

Proof. Let G be a tree and for contradiction suppose $C \leq G$ is a cycle. Then by the definition of a cycle, there exists $v \in V(C)$, such that elements of $V(C)$ can be arranged into a circle of consecutively adjacent vertices. Let u, v be the first and last element of $V(C)$ in the above arrangement, and $x, y \in V(G')$ where $(G' := V(G), E(G) \setminus \{uv\})$. As the there path arranged above remains, it follows that u and v are connected. So, as $x, y \in V(G') = V(G)$, there exists some path P from x to y in G. Now, if $uv \in E(P)$, we can simply replace that with path from u to v implying G' is also connected. # □

Lemma 3. The graph G is a tree if and only if it is connected and acyclic.

Proof. The forward direction follows directly from the above lemma so it suffices to show that G is minimally connected if it it connected and acyclic.

Suppose that G is not minimally connected, then there exists $uv \in E(G)$ such that $G' := (V(G), E(G) \setminus \{uv\})$ is also connected, so there exists a path P in G' from u to v. Now, as this sequence P by definition begins with u and end with v, and u, v are adjacent by uv in G, this sequence is therefore a cycle in G. # □

Lemma 4. Any two vertices on a tree is joined uniquely by a path.

Proof. Existence is true by the connectedness of a tree so all that remains is to prove that the path is unique.

Let T be a tree, $x, y \in V(T)$, and P_1, P_2 paths from x to y in T. Suppose $P_1 \neq P_2$ and let us denote p_1, p_2 the path arrangement. Let $S := \{n \in \mathbb{N} \mid p_1[n] \neq p_2[n]\}$, then, as $P_1 \neq P_2$,

S is not empty; and the fact that paths are finite, S has a minimum and a maximum value i, j by the well-ordering principle. Now, by connecting $p_1[i-1 : j+1]$ to $p_2[i-1 : j+1]$ (both of which index makes sense as $i > 1$ as $p_1[1] = x = p_2[1]$ and similarly for j), and by considering lemma 1, we have created a cycle in T. # □

In nature, many trees have *leaves* and therefore, so do our graph-theoretic trees (in fact our trees are stronger as *all* trees with two or more vertices have at least two leaves).

Definition 1.16 (Leaf)**.** A vertex in a tree is a leaf if and only if it has degree one, i.e. it is only connected to one other vertex.

Lemma 5. A tree with two or more vertices has at least two leaves.

Proof. Consider the longest path in this tree. It must have a leaf at either end of the path as otherwise it is not the longest path in this tree. □

Lemma 6. Let T be a tree and $v \in V(T)$ a leaf. Then $T - v$ is also a tree[2].

Proof. $T - v$ is connected as for all $x, y \in V(T - v)$, x, y is connected in T by some path P. Now, as $v \notin V(P)$ since if otherwise v must be on the either end of the path implying either x or y equals v. Also, as $v \notin V(P)$, e_v, the unique edge with endpoint v is also not in P, so $P \leq T - v$ connecting x, and y.

Thus, by lemma 3, it suffices to show that $T - v$ is acyclic. This is trivial as if $C \leq T - v$ is a cycle, then it is also a cycle in T, so we are done! □

Lemma 7. A tree of n vertices has $n - 1$ edges.

Proof. For convenience let us denote the trees of n vertices by T_n.

We apply natural number's induction on n. If $n = 1$ the result is trivial. Suppose for $n = k$, for all T_k, $|E(T_k)| = k - 1$, and let us consider the case for the tree T with $n = k + 1$ vertices.

By lemma 5, T has at least two leaves and let l be one of the leaves. Then by lemma 6, $T - l$ is a tree of n vertices, so it has $k - 1$ number of edges. However, as l is a leaf, there is only one edge with endpoint l, so $|E(T)| = |E(T - v)| + 1 = k - 1 + 1 = k$. □

Lemma 8. A connected graph G of n vertices has at least $n - 1$ edges.

Proof. We induct on the number of vertices. As the number of edges of a graph is a natural number, it is greater or equal to $0 = 1 - 1$, so true for $n = 1$.

Suppose G has $k + 1$ vertices, then by excluded middle, either G is minimally connected, or it is not. If it is, then it has $n - 1$ edges by the previous lemma so suppose otherwise. Then there exists $v \in V(G)$ such that $G - v$ is still connected. As $G - v$ has k vertices, it has at least $k - 1$ number of edges by the inductive hypothesis. Now, as G is connected, v is at least connected to another vertex, thus, G has at least k edges. □

Lemma 9. A connected graph G of n vertices is a tree if and only if it has $n - 1$ edges.

[2] We write $T - v$ for the graph $(V(T) \setminus \{v\}, E(T) \setminus S)$ where S is the set of edges with v as an endpoint.

Proof. The forward direction of the proof is exact lemma 7 so let us consider the reverse direction.

As G is connected, by lemma 3, it suffices to show that G is not cyclic. In order to show this, we apply induction on the number of vertices of G.

If G has one vertex, then it is trivially acyclic. Now suppose for connected graphs G_k of k vertices and $k-1$ edges are trees, let us consider the connected graph of G with $k+1$ vertices and k edges. If G is minimally connected then G is acyclic by lemma 3 so suppose that there exists $v \in V(G)$ such that $G - v$ is connected. Now as G is connected, there exists at least one edge who has endpoint v, so $G - v$ has at most $k-1$ edges. However, as $G - v$ is connected, it has at least $k-1$ edges by the above lemma so it has exactly $k-1$ edges and v is a leaf. Therefore, as $G - v$ is a tree by the inductive hypothesis, we have G is also a tree. □

2 Working with Networks

In this section we will be less rigorous and focus more on the methods used to analysis graphs (networks) especially really large ones.

2.1 Degree Distribution

Network systems vary in size but they are normally very large (that is they are large enough such that we can't draw them by hand), so, in order to analyse large networks, it is often useful to take a probabilistic approach.

Definition 2.1 (Degree of a Vertex of a Undirected Graph)**.** Let $(V(G), E(G))$ be a undirected graph and $v \in V(G)$, if v is the end point of some $e \in E(G)$, then we say v and e are *incident*. Then the degree of v is the number of incident edges.

Suppose we denote the degree of some vertex v by $d(v)$, then we find the total number of edges is simply

$$|E(G)| = \frac{1}{2} \sum_{v \in V(G)} d(v).$$

Note that the 1/2 factor is because each edge is incident to two vertices.

Definition 2.2 (Average Degree of a Undirected Graph)**.** Let $G = (V(G), E(G))$ be a undirected graph, then the average degree of G is $\frac{1}{|V(G)|} \sum_{v \in V(G)} d(v) = 2\,|E(G)|\,/\,|V(G)|$.

The above, however, does not simply transfer to directed graphs since we would loss the information of "directedness" of the graph. Therefore, the degree is defined slightly differently for directed graphs.

Definition 2.3 (Degree of a Vertex of a Directed Graph)**.** Let $(V(G), E(G))$ be a directed graph and $v \in V(G)$, then the degree of v is simply the difference between number incoming edges and the number of out going edges.

With the definition above, we see straight away the sum of the degrees of all vertices in a directed graph is zero so the definition for average degree does not apply for digraphs[3] either. Therefore, instead defining the average degree by averaging the sum of degrees, we use the average of the sum of either the incoming or outgoing degrees (both of which are equal).

Let us now consider the *degree distribution*, p_k, a characterisation of a graph that provides the probability that a randomly selected vertex has k degree.

Straight away, given some graph $(V(G), E(G))$, let $S := \{v \in V(G) \mid d(v) = k\}$, then $p_k = |S|\,/\,|V(G)|$.

[3] Digraph is an alternative word to directed graph.

2.1.1 Clustering Coefficient

The clustering coefficient of a graph attempts to capture the degree to which the adjacent nodes of a given node is connected to each other.

Definition 2.4 (Clustering Coefficent)**.** Let G be a graph and $u \in V(G)$ is a node with degree $d(u)$, then the clustering coefficient of u is

$$C_u = \frac{2L(u)}{d(u)(d(u)-1)},$$

where $L(u)$ is the number of edges in $E(G)$ connecting two neighbours of u.

We note that the clustering coefficient C_u is a number between 0 and 1 with $C_u = 0$ representing none of the adjacent nodes of u is adjacent with each other which $C_u = 1$ means every adjacent node of u is adjacent to every other adjacent node of u.

If the graph in question is *simple*, then we see that the clustering coefficient of some node u represents the probability that two (unique) adjacent nodes of u are adjacent[4].

Theorem 2. An acyclic graph has zero clustering coefficients everywhere.

Proof. This follows directly from that if there exists some node with clustering coefficient greater than 0, it has at least two neighbouring nodes which are adjacent to each other. Then we can easily create a path starting and ending at this node creating a cycle. # □

Corollary 2.1. A tree has zero clustering coefficients everywhere.

By taking the average of the clustering coefficient of every node, we have the *average clustering coefficient* which represents the degree of clustering over the whole network.

Definition 2.5 (Average Clustering Coefficien)**.** Let G be a graph and for all $u \in V(G)$ we denoted the clustering coefficient of u by C_u, then the average clustering coefficient of G is

$$\langle C \rangle = \frac{1}{|V(G)|} \sum_{u \in V(G)} C_u.$$

2.2 Path & Distance

Given a graph G and two nodes $u, v \in V(G)$, we would often like to quantitatively construct an ordering on these paths. This can be achieved through creating a distance function on the set of paths.

Definition 2.6 (Distance of Unweighed Paths)**.** Let G be a graph with unweighed edges and $u, v \in V(G)$. Suppose that P is a path between u and v in G, then the distance of P is

$$\text{dist} P = |P| - 1,$$

where $|P|$ denoted the length of the sequence of nodes induced by P.

[4] This is because $d(u)(d(u)-1)$ is the number of ordered combinations of choosing two nodes while for each edge counted by $L(u)$, it connects two unique parings (u, v) and (v, u).

It is not hard to imagine why such an distance function is useful, and why often we might want to look for the shortest path between to nodes in a network.

Definition 2.7 (Shortest Path)**.** Let G be a graph and $u, v \in V(G)$. Suppose we denote S as the set of paths between u and v. Then, we call $P \in S$ a shortest path if and only if

$$\text{dist}P = \min_{Q \in S} \text{dist}Q.$$

We denote this distance by $d_{u,v}$.

This definition makes sense as we are work with finite networks, and thus, the number of paths between two nodes is finite. We note that the shortest path between two nodes is not unique as many paths can have the same distance.

We quickly introduce some more definitions. Given a graph G, and a path P in G,

- the *diameter* of G is $\max\{d_{u,v} \mid u, v \in V(G)\}$;
- the *average path length* of G is $\frac{1}{|S|}\sum_{d \in S} d$ where $S = \{d_{u,v} \mid u, v \in V(G)\}$;
- P is an *Eulerian path* if and only if $E(G) \subseteq P$;
- P is a *Hamiltonian path* if and only if for all $u \in V(G)$, there exists $e \in P$, u is an endpoint of e.

2.2.1 Breadth-First Search (BFS) Algorithm

The BFS algorithm is an algorithm to determine the shortest distance between two nodes of a network.

Let G be a network and $u, v \in V(G)$, then

1. For each adjacent nodes w of u, construct a tuple $(w, d(w) = 1)$ and put them in a queue.
2. While the first element w of the queue is not equal to v, for each adjacent node x of w, construct a tuple $(x, d(x) = d(w) + 1)$ and remove w from the queue.
3. If the first element of the queue is v, then the distance is simply $d(v)$. However, if there is no element remain in the queue, we know that the two nodes are not connected.

Note that we can optimize the algorithm by making sure that we do not go back to previously visited nodes but that's simply an implementation issue on how one can achieve this.

We quickly recall from year 1 computing the definitions of big-O and big-Ω notations.

Definition 2.8 (Big-O notation)**.** Given a cost function $C : \mathbb{N} \to \mathbb{R}$, we say $C(N) = O(f(N))$ for some $f : \mathbb{N} \to \mathbb{R}$ if and only if there exists some $N_0 \in \mathbb{R}, r \in \mathbb{R}$ such that for all $n \geq N_0$, $C(n) \leq rf(n)$.

Definition 2.9 (Big-Ω notation)**.** Given a cost function $C : \mathbb{N} \to \mathbb{R}$, we say $C(N) = \Omega(f(N))$ for some $f : \mathbb{N} \to \mathbb{R}$ if and only if there exists some $N_0 \in \mathbb{R}, r \in \mathbb{R}$ such that for all $n \geq N_0$, $C(n) \geq rf(n)$.

We can interpret the big-O and big-Ω notations as providing some information about the upper and lower bounds of the cost function.

With that, we see that the complexity of this algorithm is $O(|V|)$ since every node will be explored in the worst case.

2.3 Modularity

Networks are often useful in describing communities where there might be regions of dense clustering and sparse clustering. In analysing the networks, it might often be helpful to partition the network into different sections but this is often not easy to do. In this section, we will take a look at one of the methods one can categorise different partitions – modularity.

Definition 2.10 (Modularity of Set of Nodes)**.** Let $G = (V, E)$ be a graph and let $S_a \subseteq V$ be a set of vertices within the graph. Then the modularity of S_a is given by

$$M_a = \frac{1}{2L} \sum_{i \in S_a} \sum_{j \in S_a} \left(A_{ij} - \frac{k_i k_j}{2L} \right).$$

By directly inspecting the definition, we see that the modularity is related to how connected the nodes within S_a are compared to a configuration model with the same degree distribution (we will see more on the configuration model in the next chapter).

Definition 2.11 (Modularity of a Partition)**.** Let $G = (V, E)$ be a graph and suppose $\mathcal{C}$ is a partition of V. Then the modularity of $\mathcal{C}$ is

$$M(\mathcal{C}) = \sum_{S_a \in \mathcal{C}} M_a.$$

The modularity of a particular partition is a possible metric on how good a partition is where the higher the modularity is better. Thus, we are now interested in how to partition a graph such that the partition has highest modularity.

Straight away, we see that there are 2^{N-1} possible partitions of two subset and much more for arbitrary partitions so it is unrealistic to brute force every partition. In fact, the is in general no good way of finding the best partition and the modularity maximization problem is "NP-hard". Thus, instead of finding the maximum modularity partition, we aim to find a partition who's modularity is close to the maximum.

Let us study one of these algorithms first introduced by Newman. Given a partition of some graph $G = (V, E)$, $S_1 \sqcup S_2 = V$, let $s_i = \pm 1$ be a indicator variable where $s_i = 1$ if $i \in S_1$ and $s_i = -1$ if $i \in S_2$. Then, we see that $1/2(s_i s_j + 1) = 1$ is the indicator of whether or not i, j are in the same set. With that, we can write the modularity of this partition is

$$M = \frac{1}{4L} \sum_{i=1}^{N} \sum_{j=1}^{N} \left(A_{ij} - \frac{k_i k_j}{2L} \right) (s_i s_j + 1).$$

Now, by introducing the modularity matrix B, where $B_{ij} = \left(A_{ij} - \frac{k_i k_j}{2L} \right)$, we have

$$M = \frac{1}{4L} s^T B s,$$

where $s = (s_1, s_2, \cdots, s_N)^T$. However, as mentioned previously, we cannot simply maximize this equation so we will have to make some approximations. Suppose we relax s such that we consider the equation for $v \in \mathbb{R}^N$ where v is some arbitrary vector such that $v^T v = \|v\|^2 = N$. Then, by enforcing the Lagrange multiplier λ, we will maximize

$$\mathcal{L}(v, \lambda) = v^T B v - \lambda(v^T v - N),$$

by requiring $\nabla \mathcal{L} = 0$ which results in $Bv = \lambda v$. This is simply a eigenvalue problem and by the fact that B is symmetric, we know that there exists at least one zero eigenvalue. Now, by considering that for an eigenpair (λ, v) of B, $v^T B v = \lambda v^T v = \lambda N$, (as we required $v^T v = N$), we see that the maximum solution is simply the eigenpair with the largest eigenvalue.

However, we are not done yet as s must have entries ± 1 and thus, v is in general not a solution. So, to approximate the maximum s, we find the a possible solution of s near v. Consider that if s' is a vector such that $\|s'\| = N$, then $v^T s' = N \cos \theta$. So, we can construct some s to minimize θ by maximizing $v^T s$. Now, by observing that the closest possible vector of s is the vector of ± 1 with the same sign as v in each entries, we have found an approximation for a partition of maximal modularity.

To summarize, the Newman's spectral modularity maximization method is the following algorithm. Given some graph G,

1. construct the modularity matrix B, where $B_{ij} = \left(A_{ij} - \frac{k_i k_j}{2L}\right)$;
2. compute the largest eigenvalue of B and its corresponding eigenvector v;
3. construct s based on the sign of v.

As we can see from above, the main cost of the spectral method is from computing the eigenvalues and eigenvectors of B. In general, for a dense matrix, the cost of computing all of the eigenvalues and eigenvectors is roughly $O(N^3)$. However, as mentioned before, we only need the largest eigenvalue and its corresponding eigenvector so this cost can be improved upon using the *power method.*

Recall that a symmetric matrix is diagonalisable, and so $\sum \dim E_\lambda = N$, and thus, the eigenvectors of B, $\{v_1, \cdots, v_n\}$ form a basis of $\mathbb{R}^N$ (where we order the eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda n$). Hence, for all $x \in \mathbb{R}^N$, there exists μ such that $x = \sum_i \mu v_i$. Assuming that $\mu_1 \neq 0$ and that $\lambda_1 > \lambda_2$, we have

$$B^m x = \sum \mu \lambda_i^m v_i \approx c_1 \lambda_1^m v_1,$$

for sufficiently large $m \in \mathbb{N}$. Now, if $B^m x = \lambda_1^m v_1$, we have $B^m x \in \mathrm{sp}(v_1) \in E_{\lambda_1}$ which is an eigenvector of λ_1.

Remark. Sometimes, $B^m x$ might become too large for computations, so it might make sense to normalize the vector after each multiplication (or alternatively divide the vector by the maximum entry – that is normalize with respect to the ∞-norm).

2.4 Centrality

While the modularity of a graph provides us with a measure of how well a partition fit for a particular network, the partition with the largest modularity is not necessarily the best partition for a particular use case. Considering the following situation.

Suppose $G = (V, E)$ is a network and let $I \sqcup J = V$ and denote L_I the number of links connecting nodes within I, L_{IJ} the number of links connecting nodes between I and J, K_I the number of stubs in I and M_{IJ} the modularity of the partition of $\{I, J\}$, that is (M_{IJ} = M_I + M_J). Then, by definition

$$M_I = \frac{1}{K}(2L_I - K_I^2/K); \quad M_J = \frac{1}{K}(2L_J - K_J^2/K),$$

and

$$M_{IJ} = \frac{1}{K}(2L_I + 2L_J + 2L_{IJ} - (K_I + K_J)^2/K),$$

and so, by expanding the parenthesis,

$$M_{IJ} = M_I + M_J + \frac{2}{K}(L_{IJ} - (K_I K_J)/K).$$

Thus, we see that if $L_{IJ} > K_I K_J/K$ then the modularity of the partition is less than the modularity of not partitioning at all. This is can cause a problem since this might result in different partitions depending on the size of the community one is working with. To solve this problem, we shall take a look at the method of centrality that measures the importance of links within communities.

Definition 2.12 (Node Betweenness Centrality)**.** Let $G = (V, E)$ be an undirected connected graph and let $n_i(a, b)$ be the number of distinct shortest paths connecting a to b through i for some $a, b, i \in V$. Furthermore, let p_{ab} be the number of total number of distinct shortest paths between a, and b for $a \neq b$ and 1 if $a = b$. Then, the centrality of the node $i \in V$ is

$$x_i = \sum_{a \in V} \sum_{b \in V} \frac{n_i(a, b)}{p_{ab}}.$$

Similarly, we can define a notion of centrality for edges.

Definition 2.13 (Edge Betweenness Centrality)**.** Taking all notations from above, given an edge $\alpha\beta \in E$ where $\alpha, \beta \in V$, the edge betweenness is

$$y_{\alpha\beta} = \sum_{a \in V} \sum_{b \in V} \frac{m_{\alpha\beta}(a, b)}{p_{ab}},$$

where $m_{\alpha\beta}(a, b)$ for $a, b \in V$ is the number of unique shortest paths from a to b passing through the edge $\alpha\beta$ while p_{ab} is the total number of unique shortest paths between a and b.

With this, we can think of a natural method for finding communities within a network, that is, removing the edges with highest centrality, and so, breaking the network into smaller partitions. This is called the Girvan-Newman community detection method. This is a powerful method as it allows us to easily find higher numbers of partitions, however, it is computationally expensive as we are required to compute the edge centrality for all of edges (time complexity of $O(|V|\,(|V| + |E|))$ to compute the edge centrality of all edges).

One obvious notion of centrality one might come up with is the degree centrality, that is, the centrality of a node in a network is simply its degree. This notion has some drawbacks as this notion will result in a node having 50 neighbours of degree 1 having the same centrality as a node with 50 neighbours of degree 50. To solve this problem while retaining the original idea of using a node's degree for its centrality, we can define the eigenvector centrality.

Definition 2.14 (Eigenvector Centrality)**.** Let $G = (V, E)$ be an undirected connected graph with adjacency matrix A, then the eigenvector centrality of some node is defined as $c(x_i)$ for all $x_i \in V$ such that,

$$c(x_i) = \alpha \sum_{j=1}^{N} A_{ij} c(x_j),$$

is satisfied for some $\alpha > 0$.

To see why this is called the eigenvector centrality, we observe that, by writing the equation in matrix form, the centrality equation becomes $Ac(x) = \alpha^{-1} c(x)$ and so, becomes an eigenvalue problem (the existence of a solution is guaranteed by Perron-Frobenius).

While the eigenvector centrality works reasonably well for undirected graphs, it works less well for directed graphs. In general, for directed graphs, we consider nodes with more edges pointing into it to be more important (say we would consider an influential webpage linking to my personal page to be more important and me linking to their page). The Katz centrality extends the eigenvalue centrality by assigning a base centrality to each node.

Definition 2.15 (Katz Centrality)**.** Taking the notations as before, the Katz centrality of a node x_i is simply $c(x_i)$ satisfying

$$c(x_i) = \alpha \sum_{j=1}^{N} A_{ij} c(x_j) + \beta,$$

for some constants $\alpha, \beta > 0$. In general, we take $\beta = 1$.

By writing the equation in matrix form, we have $(I - \alpha A)c(x) = \mathbf{1}$, and so, $c(x)$ has a solution if and only if $0 \neq \det(I - \alpha A) = \det(\alpha^{-1} I - A)$ that is α^{-1} is not an eigenvalue of A.

3 Random Networks

While sometimes we do have a concrete network on hand to work this, this is not often the case, thus, it is sometimes useful to model the behaviours of networks with random networks. We will first take a look at the classical random network – the $G(N, p)$ model and move on to some modern approaches.

3.1 Some Properties of the Model

In the section we will take a look at the $G(N, p)$ model. As the number of edges in G is characterised solely by the probability p, G can any number of edges between 0 and $\binom{n}{k}$, so it is useful to consider the probability that G has n edges.

We see straight away that the probability that G has n edges equals the sum of the probabilities that G equals to some graph G_i with $|E(G_i)| = n$, so

$$\mathbb{P}(|E(G)| = n) = \sum_{|E(G_i)|} \mathbb{P}(G = G_i) = \sum p^n(1-p)^{N(N-1)/2-n},$$

so it suffices to find the number of possible graphs with N nodes and n edges. This is simply $\binom{N(N-1)/2}{n}$, thus, the probability of G having n edges is

$$\binom{N(N-1)/2}{n} p^n(1-p)^{N(N-1)/2-n}.$$

In fact, this is the binomial distribution with parameters $N(N-1)/2$ and n.

We might also find the degree distribution of G to be useful. Let $v \in V(G)$, we recall the degree distribution is the probability that v is connected to k other nodes. As there are in total N nodes, v can at most connect to $N-1$ other nodes each with a success possibility of p. Thus, the degree distribution of G is

$$p_k = \binom{N-1}{k} p^k(1-p)^{N-1-k},$$

that is, a binomial distribution with parameters $N-1$ and p.

We recall that for large n and small p such that $n \sim p^{-1}$, so, in fact, at the limit, the degree distribution of a random network can be approximated by a Poisson distribution with the parameter $\lambda = np$.

The clustering coefficient of a random Erdos-Renyi graph is given by p. This is because given n neighbouring nodes of a node, there are $n(n-1)/2$ possible edges connecting them each with probability p, resulting in the local clustering coefficient for a node being $pn(n-1)/2$. But now, by putting this into the clustering coefficient formulae, we have

$$C_i = \frac{2pn(n-1)/2}{n(n-1)} = p,$$

as required.

3.2 Properties at the Limit

Before now, we have considered the random graph mostly heuristically without much rigour. While we shall not change this approach to proving everything from first principle, we will in this section consider random graphs from a more probabilistically point of view.

3.2.1 Basic Notions

Formally, a random Erdos-Renyi graph can be formulated as follows.

Definition 3.1 (Random Graph)**.** A random graph $G(N, p)$ is a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ where the sample space Ω is all possible simple graphs with N nodes (that is all $|\Omega| = 2^{N'}$[5] of them), $\mathcal{F}$ some σ-algebra and $\mathbb{P} : \mathcal{F} \to [0, 1]$ the probability measure.

Remark. We will begin by looking at the probability measure $\mathbb{P}$ such that for all $G \in \Omega$, $\mathbb{P}(G) = p^m(1-p)^{N-m}$, where $m = |E(G)|$.

We will also be using small o notation in which we refer a function f as $f(n) = o(g(n))$ if $f(n)/g(n) \to 0$ as $n \to \infty$.

A statement that is commonplace within network science is: "For $p > 0$, a random graph in $G(N, p)$ is connected." This means mathematically, $\mathbb{P}(G \in G(N, p), G \text{ is connected}) \to 1$ as $N \to \infty$. In short, one often says: "for $p > 0$, G is connected w.h.p[6]".

Consider a random Erdos-Renyi graph $G(N, p)$. Suppose, naively, we write k_i for the degree of the i-th node, then, it might make sense to say that, the average k_i tends to the expected value of the nodes $\langle k \rangle = p(N-1)$, i.e.

$$\lim_{N\to\infty} \frac{1}{N} \sum_{i=0}^{N} k_i = \langle k \rangle.$$

But upon further inspection, this equation is less clear. As $G(N, p)$ is simply a probability space, it doesn't make sense to take the i-th node, so k_i is in fact a function on the sample space Ω to $\mathbb{R}$. Furthermore, this sequence of functions does not converge to some number, that is, for all $N \in \mathbb{N}$, we can find a graph G_N with 0 edges so the limit is also 0 for this sequence. Indeed, what we would actually like to say is

$$\lim_{N\to\infty} \mathbb{P}\left(\left| \frac{1}{N} \sum_{i=0}^{N} k_i - \langle k \rangle \right| \geq 0 \right) = 0,$$

for all $\epsilon > 0$. We often write this as

$$\lim_{N\to\infty} \frac{1}{N} \sum_{i=0}^{N} k_i = \langle k \rangle \ \text{ w.h.p.}$$

This is true since $k_i \sim B(N-1, p)$ identically with mean $\langle k \rangle$. So, using a version of the LLN[7], the result follows.

[5] From this point forward, we write $\binom{n}{2}$ as N'.

[6] With high probability.

[7] Be careful as they are not independent since if $k_0 = N - 1$, then $k_i \geq 1$ for all other i.

3.2.2 Existence of Large Clusters

We will now prove some theorems that will provide us with some insights into when will large clusters arise.

Theorem 3. $\langle k \rangle > 1 \iff$ there exists w.h.p. a connected component which contains a proportion $S > 0$ of the total number of nodes, where S satisfies $S = 1 - e^{-\langle k \rangle S}$.

Theorem 4. Given $p > 0$, the diameter of $G \in G(N, p)$ is w.h.p. is at most two, so G is connected w.h.p.

Proof. Let $X_N(G)$ be the number of pairs (i, j) in G such that i, j has no common neighbours. So, by considering the probability that i, j do not have k as a common neighbour is

$$p(1-p) + (1-p)p + (1-p)(1-p) = 1 - p^2,$$

(where we simply add up the probabilities of all valid combinations), we have the probability that i, j has no common neighbours is $(1-p^2)^{N-2}$. So,

$$E(X_N) = N'(1-p^2)^{N-2} \to 0,$$

as $N \to \infty$. And, hence, by Markov's inequality, $\mathbb{P}(X_N \geq 1) \leq E(X_N) \to 0$ so $\mathbb{P}(X_N = 0) \to 1$ as $N \to \infty$. □

Theorem 5. If $p(N) = o(N)/N$ then, w.h.p. $G \in G(N, p)$ has no triangles.

Proof. Let T_N be the number of triplets (i, j, k) such that they are pairwise connected. Of course, then probability of a randomly chosen triplet being a triangle is p^3, so we have

$$E(T_N) = \binom{n}{3} p^3 \to 0,$$

as p tends to zero faster than N. Hence, again by Markov's inequality we have $\mathbb{P}(T_N \geq 1) \to 0$, resulting in $\mathbb{P}(T_N = 0) \to 1$. □

3.3 Other Models of Random Networks

By comparing the $G(N, p)$ model to to real networks, we find that there is a big difference between the prediction and the actual networks. The could be due to numerous reasons such as the fact that the probability of the existence of a particular link in the $G(N, p)$ model is independent from the number of links a node already have. In this section we shall take a look at some more modern models of random graphs with (perhaps) better predictions in certain situations.

We will first take a look at the *configuration model.*

Definition 3.2 (Configuration Model)**.** The configuration model with n nodes a random network with the sequence $S = \{x_1 \cdots x_n\} \subseteq (\mathbb{N}^*)^n$ such that the i-th node has degree x_i.

Remark. We see that $\sum S$ must be even as each edge has two endpoints.

While the configuration model have some benefits in that it better model a network given its degree distribution, and that it retains the small world property, we find that it is still lacking. Not only is does the configuration model require we to provide it with the degree distribution, it has a fixed number of nodes, so it cannot model the growth of a network over time. We attempts to achieve this with the *Barabasi-Albert* model.

Consider an evolving network in which the number of nodes increases over time. By considering real world examples, we notices some pattern regarding how new links are formed – the nodes with higher degree form new links more easily. This property is called *preferential attachment* and we would like to model this with our random graph.

To achieve this, at each iteration, we assign each nodes some value proportional to their degree to indicate how likely they are to form new links. The easiest way to do this is linear preferential attachment.

Definition 3.3 (Linear Preferential Attachment)**.** The probability of a new link attaching to an existing node following linear preferential attachment is given by

$$\pi_i(t+1, G(t)) = \frac{k_i}{\sum_{i=1}^{N(t)} k_i(t)},$$

where $\pi_i(t+1, G(t))$ is the probability of a link connecting to node i in graph $G(t)$ with $t = 1, 2, \cdots$.

So, at each time t, we add in a new node and a new edge with probabilities according to the linear preferential attachment. We call this an element of the family of the Barabasi-Albert models.

Without needing to consider the preferential attachment model, we see that at time $t \in \mathbb{N}$, there are $(t+1)!$ number of possible graphs.

We would also like to consider the expected properties of this model. By definition, given some property of a graph at time t, $f(t)$, we can write the expected value of f as

$$\langle f(t) \rangle = \sum_{i=1}^{(t+1)} \mathbb{P}(G_i(t)) f(t \mid G_i(t)).$$

However, this is very vague and does not provide us with much useful information, so let us look at some concrete examples.

Let us take a look at the degree distribution of a Barabasi-Albert model. Consider the expected number of nodes of degree k at time $t+1$,

$$\langle N_k(t+1) \rangle = \sum_{i=1}^{(t+1)!} P(G_i(t)) N_k(t+1 \mid G_i(t)).$$

By considering each cases we find, in general,

$$N(t+1)p_k(t+1) = N(t)p_k(t) - \frac{p_k(t)N(t)k}{2L(t)} + \frac{p_k(t)N(t)(k-1)}{2L(t)},$$

and for $k = 1$,

$$N(t+1)p_1(t+1) = N(t)p_1(t) - \frac{p_1(t)N(t)}{2L(t)} + 1.$$

Thus, by taking the limit of the two equations as $t \to \infty$, we have

$$p_{1,\infty} = \frac{2}{3};$$

and

$$p_{k,\infty} = \frac{k-1}{k+2} p_{k-1,\infty},$$

where $p_{k,\infty}$ is the limit of the probability that a node has degree k as $t \to \infty$. By rearranging, we have $p_{k+1,\infty} = \frac{k}{k+3} p_{k,\infty}$, and by induction, we find

$$p_{k,\infty} = \frac{4}{(k+2)(k+1)k} \sim k^{-3}.$$

Thus, we can conclude that the Barabasi-Albert model has the scale free property.

Lastly, let us look at how the degree of a node evolve over time. Luckily, this is very simple. If we let $k_i(t)$ be the random variable describing the degree of node i, we see straight away the relation,

$$k_i(t+1) = k_i(t) + \delta_i(t+1),$$

where $\delta_i(t+1)$ is the Bernoulli random variable with parameter given by $k_i(t)$, that is $\pi_i(t+1)$. So, (assuming linear preferential attachment), we have

$$\langle k_i(t+1) \rangle = \langle k_i(t) \rangle \left(1 + \frac{1}{2+2t}\right).$$

So, by induction,

$$\langle k_i(t_0 + j) \rangle = \prod_{n=0}^{t_0+j-1} \left(1 + \frac{1}{2+2(t_0+n)}\right) \sim \sqrt{\frac{t_0+1+j}{t_0+1}},$$

where t_0 is defined to be the smallest t such that $k_i(t = t_0) = 1$. We define such an t_0 since for all $t < t_0$, i-th node does not exist.

So far, we have looked at an element of the family of the Barabasi-Albert model, that is adding one edge and one node at each iteration. It is not hard to see that this will result in zero clustering everywhere. Suppose instead, we add one node and m new edges every iteration. Then, the degree distribution is

$$p_k \approx \frac{2m(m+1)}{(k+2)(k+1)k},$$

and the clustering is

$$\langle C \rangle \sim \frac{(\log N)^2}{N}.$$

Remark. We see that the degree distribution is consistent with our previous result where $m = 1$.

With the general Barabasi-Albert model, we observe an “ultra small-world” behaviour, that is the distance grows slower than $\log N$. Furthermore, by once again observing our degree distribution, we see that they all follow a k^{-3} trend. However, this is not necessary the case for real networks. By observing real networks, it was found that the linear preferential attachment model can be generalised such that

$$\pi_i = c(k_i^\alpha + \beta),$$

where α, β are parameters and c is the normalising constant.

With $\alpha = 1$,

$$p_k \sim k^{-\left(3+\frac{\beta}{m}\right)}.$$

With $\beta = 0$ and $\alpha < 1$, we find

$$p_k \sim k^{-\alpha} \exp\left(\frac{-2\mu(\alpha)}{\langle k\rangle(1-\alpha)k^{1-\alpha}}\right).$$

Lastly, with $\alpha > 1$, we find a unrealistic hub structure is generated.

4 Spreading Processes

4.1 Diffusion in Networks

We will first consider random walks and diffusion in the first dimension.

Definition 4.1 (Random Walk in 1-D)**.** A random walk in the first dimension is a real valued function $X_i : \mathbb{R} \to \mathbb{R}$ such that

$$X_i(t+\delta) = X_i(t) + l_i(t),$$

where l_i is the random variable such that $\mathbb{P}(l_i(t) = h) = \mathbb{P}(l_i(t) = -h) = 1/2$ for some $t, \delta, h \in \mathbb{R}$.

Given some particles following some a particular random walk, we would like to find the expected number of particles as some position x_0 at time t. Consider that, during some time Δt, a particle will move some distance $\pm\Delta x$ that is, the flux at $x_0 \pm \Delta x/2$. We see that

$$\langle j(x_0 + \Delta x/2, t)\rangle = \frac{1}{2}(N(x_0, t) - N(x_0 + \Delta x, t)),$$

$$\langle j(x_0 - \Delta x/2, t)\rangle = \frac{1}{2}(N(x_0 + \Delta x, t) - N(x_0, t)),$$

where $j(x,t)$ denotes the flux at position x time t (note that we have define movement to the right to be positive). These flux' tells us how the expected number of particles changes over a particular time step with

$$\langle N(x_0, t+\Delta t)\rangle = \langle N(x_0, t)\rangle + \langle j(x_0 - \Delta x/2, t)\rangle - \langle j(x_0 + \Delta x/2, t)\rangle,$$

and so,

$$\langle N(x_0, t+\Delta t)\rangle - \langle N(x_0, t)\rangle = \frac{1}{2}(\langle N(x_0, t+\Delta t)\rangle - 2\langle N(x_0 + \Delta x/2, t)\rangle) + \langle N(x_0 - \Delta x/2, t)\rangle.$$

However, we are not interested in this values as it is dependent on a particular pair of Δx and Δt, and so, by defining $n(x,t) := \langle N(x,t)\rangle/\Delta x$, we can take the limit as $\Delta x, \Delta t$ tends to 0, resulting in,

$$\frac{\partial n}{\partial t} = \alpha \frac{\partial^2 n}{\partial x^2}. \tag{1}$$

To obtain this, we assumed $\Delta x \sim 2\alpha\sqrt{\Delta t}$ which turns out to be empirically supported. Equation 1 is refereed to as the one dimensional diffusion equation and can be easily extended in to higher dimensions to obtain

$$\frac{\partial n}{\partial t} = \alpha \nabla^2 n. \tag{2}$$

With networks, we no longer have spatial derivatives, however, we can employ a similar idea. Instead of just considering the two fluxes, we now consider the flux' of all edges connected to some node.

Suppose we denote $n_i(t)$ to be the number of particles at node i at time t and $J_{i,j}(t)$ to be the flux per unit time Δt from j to i at time t, we see that

$$J_{i,j}(t) = -\alpha(n_i(t) - n_j(t)),$$

and so, following the 1 dimensional case,

$$n_i(t+\Delta t) - n_i(t) = \Delta t \sum_{j \in N_i} J_{i,j}(t),$$

where N_i is the set of neighbours of i. Thus, by taking the limit as $\Delta t \to 0$, we have,

$$\frac{dn_i}{dt} = \alpha \sum_{j=1}^{N} A_{i,j} J_{i,j} = \alpha \sum_{j=1}^{N} A_{i,j}(n_j - n_i) \tag{3}$$

where A is the adjacency matrix of the network. By recognizing that $\sum_{j=1}^{N} A_{i,j} n_i = n_i \sum_{j=1}^{N} A_{i,j}$, and that $\sum_{j=1}^{N} A_{i,j} = k_i$, the degree of i, we see

$$\frac{dn_i}{dt} = \alpha(\sum_{j=1}^{N} A_{i,j} n_j - n_i k_i). \tag{4}$$

Lastly, by defining $Q := \text{diag}(k_1, \cdots, k_n)$ the degree matrix, and by writing $c = (n_1, \cdots, n_N)^T$, we find,

$$\frac{d\mathbf{n}}{dt} = \alpha(A - Q)\mathbf{n}. \tag{5}$$

Interestingly, we recall from last year that we have seen $A - Q = -L$ where L is the graph Laplacian, so we can write this equation as $\frac{d\mathbf{n}}{dt} = -\alpha L\mathbf{n}$ which is easily solvable using the methods from last year's calculus.

4.2 Modelling Pandemics

The spread of disease is a form of diffusion and this can be modelled especially well with networks. We shall in this section take a look at the different models for predicting the spreading of diseases.

4.2.1 The Susceptible-Infected (SI) Model

Consider a disease the spreads in a population of N individuals. Suppose we let $S(t), I(t)$ be the number of susceptible and infected individuals respectively at time t, so $S(0) = N$ and $I(0) = 0$, then the SI model predicts the spread of this disease with,

$$\frac{dI}{dt} = \beta \langle k \rangle \frac{S(t)I(t)}{N},$$

where β is the infection rate, $\langle k \rangle$ is the average degree of each individual, that is the average number of people an individual has contact with. By inspection, we see the SI model interprets the infection rate to be proportional to the number of susceptible individuals, the number of infected individuals and their average number of contacts.

By rewriting $s = S/N$ and $i = I/N$, we find the SI model becomes

$$\frac{di}{dt} = \beta \langle k \rangle si,$$

and by solving this differential equation, we have

$$i = \frac{i_0 e^{\beta \langle k \rangle t}}{1 - i_0 + i_0 e^{\beta \langle k \rangle t}},$$

where $i_0 = i(0)$.

4.2.2 The Susceptible-Infected-Susceptible (SIS) Model

Unlike the SI model, in which once an individuals become sick, they are sick forever, the SIS model considers the fact that an individual will eventually recover and become susceptible again. This model is described by

$$\frac{di}{dt} = \beta\langle k\rangle i(1-i) - \mu i,$$

where μ is the recovery rate.

By solving the differential equation, we have

$$i = \left(1 - \frac{\mu}{\beta\langle k\rangle}\right)\frac{Ce^{(\beta\langle k\rangle-\mu)t}}{1+Ce^{(\beta\langle k\rangle-\mu)t}},$$

where

$$C = \frac{i_0}{1-i_0-\frac{\mu}{\beta\langle k\rangle}}.$$

We see that, while in the SI model, $i \to 1$ as $t \to \infty$, that is everyone will eventually become infected, the SIS model will reach the equilibrium $1-\mu/\beta\langle k\rangle$ if $\mu < \beta\langle k\rangle$ or the disease will die out if $\mu \geq \beta\langle k\rangle$. By this fact alone, we see that $R_0 = \beta\langle k\rangle/\mu$ is an important number, that is, if $R_0 > 1$ then the disease will remain within the population while if $R_0 \leq 1$, the disease will die out.

4.2.3 The Susceptible-Infected-Recovered (SIR) Model

Lastly, we have the SIR model. While the previous two model assume that individuals are either infected or susceptible, the SIR model assumes that an individuals, after being infected will either die or become immune, that is removed from the ecosystem. This model is governed by the following system of differential equations.

$$\frac{ds}{dt} = -\beta\langle k\rangle is$$
$$\frac{di}{dt} = -\mu i + \beta\langle k\rangle is$$
$$\frac{dr}{dt} = \mu i$$

where r is the proportion of the population that are recovered or removed.

In contrast to the SI and the SIS models, with this model, $i \to 0$ as $t \to \infty$ as everyone either becomes immune or died.

Unlike the the SI and the SIS model, the SIR model does not have an analytical solution so we would need to look at limiting cases to gain insight into this model. Suppose we have $s = 1 - \epsilon\tilde{s}(t) + o(\epsilon^2)$, $i = \epsilon\tilde{x}(t) + o(\epsilon^2)$ and $r = \epsilon\tilde{r}(t) + o(\epsilon^2)$ for some small ϵ. Then, the by rewritting and simplifying, SIR model becomes

$$\frac{d\tilde{s}}{dt} = \beta\langle k\rangle\tilde{x} - \epsilon\tilde{x}\tilde{s} + o(\epsilon)$$
$$\frac{d\tilde{i}}{dt} = \beta\langle k\rangle\tilde{x} - \epsilon\tilde{x}\tilde{s} - \mu\tilde{x} + o(\epsilon)$$
$$\frac{d\tilde{r}}{dt} = \mu\tilde{x} + o(\epsilon)$$

Thus, by taking $\epsilon \to 0$, we have

$$\frac{d\tilde{s}}{dt} = \beta\langle k\rangle\tilde{x}$$
$$\frac{d\tilde{i}}{dt} = \beta\langle k\rangle\tilde{x}$$
$$\frac{d\tilde{r}}{dt} = \mu\tilde{x}.$$

Resulting in $\tilde{x} = \tilde{x_0}e^{(\beta\langle k\rangle - \mu)t}$, implying that if $\beta\langle k\rangle - \mu > 0$ then the disease will exponentially spread while if $\beta\langle k\rangle - \mu < 0$ the number of infected will exponentially decrease.

4.3 Epidemics on Networks

The models described above, while useful in certain conditions, make many assumptions (such as the assumptions that all susceptible individuals are equally likely to catch the disease and all infected individuals spread the disease equally likely) that are not necessarily correct. To combat this, it might be useful to consider epidemics on specific networks.

For simplicity, we shall again suppose that an individual is either susceptible or infected and we shall assign a different definition on β, that $\beta\Delta t$ is the probability that a susceptible person is infected via a link to an infectious person over some time Δt. Let us now define the random variable $X_i(t)$ to be the indicator random variable that determines whether or not note i is infected at time t. Consider now,

$$\mathbb{P}(X_i(t + \Delta t) = 1) = \mathbb{P}(X_i(t) = 1) + \mathbb{P}(i \text{ is infected between } t, t + \Delta t)$$

where

$$\mathbb{P}(i \text{ is infected between } t, t + \Delta t) = \beta\Delta t \sum_{j=1}^{N} \mathbb{P}(X_i(t) = 0, X_j(t) = 1)A_{i,j},$$

and so, by rearranging,

$$\frac{\langle X_i(t + \Delta t)\rangle - \langle X_i(t)\rangle}{\Delta t} = \beta \sum_{j=1}^{N} A_{i,j}\langle(1 - X_i(t))X_j(t)\rangle.$$

So, by letting $\Delta t \to 0$, we have

$$\frac{d\langle X_i\rangle}{dt} = \beta \sum_{j=1}^{N} A_{i,j}\langle(1 - X_i)X_j\rangle.$$

However, this system of differential equations in general does not have closed solutions, so we shall make some approximations. For giant networks in which each node has many neighbour, the effect of one neighbour to another is minimal, so, it is reasonable to approximate $\mathbb{P}(X_i = 0, X_j = 1) = \mathbb{P}(X_i = 0)\mathbb{P}(X_j = 1)$. Thus, with this approximation, we have

$$\frac{d\langle X_i\rangle}{dt} = \beta\langle k_i\rangle \sum_{j=1}^{N} A_{i,j}\langle X_j\rangle,$$

where $\langle k_i \rangle = \langle 1 - X_i \rangle$.

While this is a neat solution, we naively assume independence which is in general not true. So we need to approach this problem with perturbation as we had done with the SIR model. Suppose we have $\langle X_i \rangle = \epsilon z_i + o(\epsilon^2$, for some small ϵ. Then,

$$\epsilon \frac{dz_i}{dt} = \beta(1 - \epsilon z_i) \sum_{j=1}^{N} A_{i,j} \epsilon z_j + o(\epsilon^2),$$

and so, by dividing by ϵ,

$$\frac{dz_i}{dt} = \beta(1 - \epsilon z_i) \sum A_{i,j} z_j + o(\epsilon).$$

Thus, as $\epsilon \to 0$, we have

$$\frac{dz_i}{dt} = \beta \sum A_{i,j} z_j.$$

Now, by writing this in matrix form,

$$\frac{d\mathbf{z}}{dt} = \beta A \mathbf{z},$$

which can be solved using first year's calculus resulting in $\mathbf{z} = \tilde{\mathbf{z}}_i e^{\lambda_i t}$ where λ_i / β and $\tilde{\mathbf{z}}_i$ are eigenvalues and eigenvectors of A respectively[8].

4.4 Degree-based analysis

Thus far, we had looked at the evolution of a epidemic on a network and have found that the early spread is related to the eigenvalues of the adjacency matrix. However, we do not have much insight into how the network structure is related to these eigenvalues. To combat this, we take a look at an alternative approach where we assume the statistical properties of a node is dependent only on its degree.

Suppose we let ϕ_k be the indicator random variable indicating whether or not nodes of degree k is infected at some time t. Then, we are interested in the probability $\mathbb{P}(\phi_k = 1)$. Consider

$$\mathbb{P}(\phi_k(t + \Delta t) = 1) = \mathbb{P}(\phi_k(t) = 1) + \sum_{k'=1}^{k_{\max}} \beta \Delta t \mathbb{P}(\theta(k, k') = 1, \phi_{k'}(t) = 1, \phi_k(t) = 0),$$

where $\theta(k, k')$ is the indicator random variable indicating whether or not k has neighbour of degree k'. By conditional probability, we see that

$$\begin{aligned} &\mathbb{P}(\theta(k, k') = 1, \phi_{k'}(t) = 1, \phi_k(t) = 0) \\ &\qquad = \mathbb{P}(\phi_{k'} = 1 \mid \phi_k = 0, \theta(k, k') = 1) \mathbb{P}(\theta(k, k') = 1 \mid \phi_k = 0) \mathbb{P}(\phi_k = 0) \\ &\qquad = \mathbb{P}(\phi_{k'-1} = 1) \mathbb{P}(\theta(k, k') = 1) \mathbb{P}(\phi_k = 0), \end{aligned}$$

so we have,

$$\mathbb{P}(\phi_k(t + \Delta t) = 1) = \mathbb{P}(\phi_k(t) = 1) + \sum_{k'=1}^{k_{\max}} \beta \Delta t \mathbb{P}(\phi_{k'-1} = 1) \mathbb{P}(\theta(k, k') = 1) \mathbb{P}(\phi_k = 0).$$

[8] The existence of the real Eigenvalue follows from the Perron-Frobenius theorem.

Now, by taking $\Delta t \to 0$, we have

$$\frac{d\langle \phi_k \rangle}{dt} = \beta k (1 - \langle \phi_k \rangle) \sum_{k'=1}^{k_{\max}} \langle \phi_{k'-1} \rangle \langle \theta(k, k') \rangle.$$

With the configuration model, we have $\mathbb{P}(\theta(k,k') = 1) = \frac{k' p_{k'}}{\bar{k}}$ resulting,

$$\frac{d\langle \phi_k \rangle}{dt} = \beta k \langle 1 - \phi_k \rangle \sum_{k'=1}^{k_{\max}} \langle \phi_{k'-1} \rangle \frac{k' p_{k'}}{\bar{k}}.$$

Let us now introduce a small perturbation $\langle \phi_k \rangle$. Suppose we write $\Theta = \sum_{k'=1}^{k_{\max}} \langle \phi_{k'-1} \rangle \frac{k' p_{k'}}{\bar{k}} = \sum_{k'=0}^{k_{\max}-1} \langle \phi_{k'} \rangle \frac{(k'+1) p_{k'+1}}{\bar{k}}$ as a short hand and so, the above equation becomes $d\langle \phi_k \rangle / dt = k\beta\Theta$. Now by differentiating with respect to t, we have,

$$\frac{d\Theta}{dt} = \sum_{k'=0}^{k_{\max}-1} \frac{(k'+1) p_{k'+1}}{\bar{k}} \frac{d\langle \phi_{k'} \rangle}{dt} = \sum_{k'=0}^{k_{\max}-1} \frac{(k'+1) p_{k'+1}}{\bar{k}} k' \beta \Theta.$$

Now, as

$$\sum_{k'=0}^{k_{\max}-1} \frac{(k'+1) p_{k'+1}}{\bar{k}} k' = \sum_{k'=1}^{k_{\max}} \frac{(k') p_{k'}}{\bar{k}} (k'-1) = \frac{1}{\bar{k}} \sum_{k'=1}^{k_{\max}} p_k (k'^2 - k') = \frac{\bar{k^2}}{\bar{k}} - 1,$$

we have

$$\frac{d\Theta}{dt} = \beta \left(\frac{\bar{k^2}}{\bar{k}} - 1 \right) \Theta.$$

Thus, solving for Θ, we have $\Theta = \Theta_0 e^{t/\tau}$, and hence,

$$\langle \phi_k \rangle = \frac{k \beta \Theta_0}{\tau} (e^{t/\tau} - 1) + \langle \phi_k(0) \rangle,$$

where $\frac{1}{\tau} = \beta(\frac{\bar{k^2}}{\bar{k}} - 1)$. With that, from the equation, we can conclude that smaller τ indicates faster spread.

4.5 Pair Approximation

We go back to our SI model on networks $\frac{d\langle x_i \rangle}{dt} = \beta \sum_{j=1}^{N} A_{i,j} \langle (1 - x_i) x_j \rangle$, where had previously made the naive assumption that $\langle (1 - x_i) x_j \rangle \approx \langle 1 - x_i \rangle \langle x_j \rangle$. While this approximation works well in certain situations, it is not a good approximation in many others. So, to solve this problem, we shall consider the pair approximation.

We know $\frac{d\langle (1 - x_i) x_j \rangle}{dt} = \frac{d\langle x_j \rangle}{dt} - \frac{d\langle x_i x_j \rangle}{dt}$ and we know $\frac{d\langle x_j \rangle}{dt}$, so we shall consider $\frac{d\langle x_i x_j \rangle}{dt}$. Consider,

$$\begin{aligned} \mathbb{P}(X_i(t + \Delta t) = 1, X_j(t + \Delta t) = 1) = {} & \mathbb{P}(X_i(t) = 1, X_j(t) \\ & + \mathbb{P}(X_i(t) = 1, X_j \to 1) \\ & + \mathbb{P}(X_j(t) = 1, X_i \to 1) \\ & + \mathbb{P}(X_i \to 1, X_j \to 1), \end{aligned}$$

where we denote $X_i \to 1$ as the event where X_i is not infected at time t but is infected at time $t + \Delta t$. By considering that $\mathbb{P}(X_i \to 1, X_j \to 1) = o(\Delta t^2)$,

$$\mathbb{P}(X_i(t) = 1, X_j \to 1) = \beta \Delta t \sum_{k=1}^{N} A_{j,l} \mathbb{P}(X_i(t) = 1, X_j(t) = 0, X_l(t) = 1),$$

and similarly

$$\mathbb{P}(X_j(t) = 1, X_i \to 1) = \beta \Delta t \sum_{k=1}^{N} A_{j,l} \mathbb{P}(X_j(t) = 1, X_i(t) = 0, X_l(t) = 1),$$

we see that, as $\Delta t \to 0$, we have,

$$\frac{d\langle x_i x_j \rangle}{dt} = \beta \sum_{l=1}^{N} \left(A_{j,l} \langle x_i s_j x_l \rangle + A_{i,l} \langle s_i x_j x_l \rangle \right),$$

where $s_i = 1 - x_i$. So

$$\frac{d\langle (1 - x_i) x_j \rangle}{dt} = \beta \sum_{l=1}^{N} \left(A_{j,l} \langle s_i s_j x_l \rangle + A_{i,l} \langle s_i x_j x_l \rangle \right).$$

5 Synchronisation

5.1 Synchronisation for Coupled Differential Equations

Let us consider a network of two node with some property $x_1(t), x_2(t)$ that varies with time according to

$$\dot{x}_i = \frac{dx_i}{dt} = ax_i,$$

where $a > 0$ for all $i = 1, 2$. We see straight away that, as $t \to \infty$, the two solutions desynchronise and diverges, that is,

$$|x_1(t) - x_2(t)| = e^{at} |x_1(0) - x_2(0)| \to \infty,$$

if $x_1(0) \neq x_2(0)$. Suppose we instead *couple* the two equations such that

$$\begin{aligned} \dot{x}_1 &= ax_1 + \alpha(x_2 - x_1), \\ \dot{x}_2 &= ax_2 + \alpha(x_1 - x_2), \end{aligned}$$

for some $\alpha > 0$. We call this coupling a *diffusive coupling* and let us consider the behaviours of this system as $t \to \infty$.

Proposition 5.1. Given the diffusive coupling system,

$$\begin{aligned} \dot{x}_1 &= ax_1 + \alpha(x_2 - x_1), \\ \dot{x}_2 &= ax_2 + \alpha(x_1 - x_2), \end{aligned}$$

where $\alpha > a/2 > 0$, $|x_1(t) - x_2(t)| \to 0$ as $t \to \infty$.

Proof. Follows straight away by defining $z = x_2 - x_1$ and solving the resultant differential equation. □

Thus, we see that for this simple case, x_1 and x_2 becomes synchronised eventually. We can generalise this result for non-linear coupled systems under certain conditions.

Proposition 5.2. Consider the diffusive coupling system of non-linear equations,

$$\begin{aligned} \dot{x}_1 &= f(x_1) + \alpha(x_2 - x_1), \\ \dot{x}_2 &= f(x_2) + \alpha(x_1 - x_2), \end{aligned}$$

Where $x_1, x_2 : \mathbb{R} \to \mathbb{R}^n$ and $f : \mathbb{R}^n \to \mathbb{R}^n$ is continuously differentiable. Then, if the image of x_1 is bounded, and there exists some $\epsilon > 0$, $M \in \mathbb{R}^+$, such that if $|x_1(0) - x_2(0)| < \epsilon$ and $\alpha > M$, then $|x_1(t) - x_2(t)| \to 0$ as $t \to \infty$.

Proof. Let $z = x_2 - x_1$, then we have $\dot{z} = f(x_2) - f(x_1) - 2\alpha z$. Using the Taylor expansion for higher dimensions, we have

$$\begin{aligned} f(x_2) &= f(x_1) + Df(x_1)(x_2 - x_1) + O(|x_2 - x_1|^2) \\ &= f(x_1) + Df(x_1)z + O(|z|^2). \end{aligned}$$

So, by substituting, $\dot{z} = Df(x_1)z + O(|z|^2) - 2\alpha z$. Now, by defining $w = e^{2\alpha t}z$, we have by product rule, $\dot{w} = 2\alpha w + e^{2\alpha t}\dot{z}$. Once again, by substituting, we have

$$\dot{w} = 2\alpha e^{2\alpha t}z + e^{2\alpha t}(Df(x_1)z + O(|z|^2) - 2\alpha z) = Df(x_1)w + O(|w|^2).$$

Consider now that, as f is continuously differentiable, and as x_1 is bounded, $Df(x_1)$ is also bounded by some $K \in \mathbb{R}$, i.e. $|Df(x_1)u| \leq K|u|$ for all u. Thus, $|\dot{w}| \leq K|w|$ and so,

$$\frac{d}{dt}|w|^2 = 2w \cdot \dot{w} = 2w \cdot Df(x_1)w \leq 2|w||Df(x_1)w| \leq 2K|w|^2,$$

so, by solving this differential inequality, we have, $|w|^2 \leq e^{Kt}|w(0)|^2$, so $|w| \leq e^{Kt}|w(0)|$, and hence, if $2\alpha > K$ then,

$$|x_1 - x_2| = |z| = \left|e^{-2\alpha t}w\right| \leq e^{-2\alpha t}e^{Kt}|w(0)| \to 0,$$

as $t \to \infty$. □

Another system which might be interesting to couple are the Lorenz equations. The Lorenz equations is a famous system of linear equations which can describe chaotic motions. We shall take a look at them now. The Lorenz equations is the following system of differential equations.

$$\begin{aligned}\dot{x} &= \sigma(y - x),\\ \dot{y} &= x(\rho - z) - y,\\ \dot{z} &= -\beta z + xy,\end{aligned}$$

for some constants σ, ρ, β. This system of equations was first introduces in the 1960's as a simplified version of the Navier-Stokes equation which can be used to model fluids.

For shorthand, let us write the Lorenz equations in the vector form $\dot{w} = f(w)$, and it turns out there exists a unique solution for each $w_0 = w(0)$.

Proposition 5.3. For each $w_0 \in \mathbb{R}^3$, there exists a unique $w : \mathbb{R}^3 \to \mathbb{R}^3$ that is a solution of

$$\dot{w} = f(w),$$

and $w(0) = w_0$, where f is the vectorised function for the Lorenz equations.

The proof of this follows from Picard's existence and uniqueness of solutions of ODE's which will be studied in the differential equations module.

It turns out that the solution of these equations has sensitive dependence on initial conditions. That is, there exists some $\delta > 0$ such that for all $w_1 \neq w_2$, there exists some $t > 0$ where $d((w_1(t), w_2(t)) \geq \delta$. This is one of the reason why weather forecast is difficult.

Let us now couple the Lorenz equations as follow.

$$\begin{aligned}\dot{w}_1 &= f(w_1) + \alpha(w_2 - w_1),\\ \dot{w}_2 &= f(w_2) + \alpha(w_1 - w_2).\end{aligned}$$

For $\alpha = 0$, as f has sensitive dependence, we have that the systems desynchronise over time. On the other hand, for α large enough, by our previous theorem, the systems can eventually converge.

5.2 Synchronisation on Networks

To discuss Synchronisation on general networks, we will use the graph Laplacian matrices $L = L_{ij} = D - A$ where D is the degree matrix and A the adjacency matrix.

Proposition 5.4. Let G be an undirected network and suppose L is the associated Laplacian. Then, L has only real positive eigenvalues, 0 is an eigenvaue corresponding to the eigenvector $mathbb1$, and 0 has multiplicity one if G is connected.

It turns out, the least positive eigenvalue λ_2 is important for characterising the synchronisation behaviour of a graph.

Proposition 5.5. The coupled network

$$\dot{x}_i = f(x_i) - \alpha \sum_{j=1}^{N} L_{ij} x_j$$

for $i = 1, \cdots, N$ globally synchronises whenever $\alpha > \Gamma/\lambda_2$ where $\Gamma \geq 0$ depends on the Lyapounov exponent of f.