Notes on Network Analysis

Alina Kochocki

Summer 2025

Abstract

This is a summary of Leonid Zhukov's lecture series, on network science. The course is intended as an introduction to networks. Many of my notes are taken or copied directly from his slides for the purpose of study. I do not take credit for the material except in how I might have chosen to summarize certain portions.

Contents

1 Lecture 1 – Introduction to Network Science

Topics covered in the course:

- Statistical properties and modeling of the network
- Network structure and dynamics
- Processes on networks

Terminology:

- Network = Graph
- Nodes = Vertices, Actors
- Links = Edges, Relations
- Clusters = Communities

Network examples exists in biology, social systems, traffic and transport systems, etc.

1.1 Graphs

- A **network** is represented by a graph, $G(V, E)$.
- A **graph** $G = (V, E)$ is an ordered pair of sets, a set of vertices V and a set of edges E, where $n = |V|, m = |E|$.
- An **edge** $e_{ij} = (v_i, v_j)$ is a pair of vertices, an ordered pair for a directed graph.

- An **adjacency matrix** $A^{n\times n}$ is a matrix with nonzero element a_{ij} where there is edge e_{ij} (to represent all directly connected combinations). Such a graph is symmetric if the graph is not directed. Weights may also exist for each edge connection.

1.2 Nodes, Degree

- Two nodes or vertices are adjacent if they share an edge.
- An edge and a node on that edge are **incident**.
- The **degree**, k_i of a node v_i is the total number of nodes adjacent to it (number of incident edges), $k_i = |N(v_i)|$.
- The average node degree in the graph,

$$\langle k \rangle = \frac{1}{n}\sum_i k_i = \frac{2m}{n} = \frac{2|E|}{|V|}. \tag{1}$$

- In directed graphs, the total node degree, $k_i = k_i^{in} + k_i^{out}$. k^{in} is the incoming degree, the number of edges pointing to the node. k^{out} is the outgoing degree, the number of edges exiting or pointing out from the node.
- In directed graphs, the average number of in and out-degrees must be equal,

$$\langle k^{in} \rangle = \frac{1}{n}\sum_i k_i^{in} = \langle k^{out} \rangle = \frac{1}{n}\sum_i k_i^{out} = \frac{m}{n} = \frac{|E|}{|V|}. \tag{2}$$

1.3 Degree Distribution

- k_i is the degree of a given node, $k_i = 1, 2, \cdots k_{max}$.
- n_k is the number of nodes with degree k, total nodes $n = \sum_k n_k$.
- The **degree distribution** is the fraction of nodes with degree k,

$$P(k_i = k) = P(k) = P_k = \frac{n_k}{\sum_k n_k} = \frac{n_k}{n}. \tag{3}$$

1.4 Graph Connectivity

- A **path** from v_i to v_j is an ordered list of vertices or edges joining two vertices.
- A graph is connected if there is a path between any two vertices.
- The **connected component** is the maximal connected subgraph of a graph. You may also have strongly or weakly connected graphs depending on directionality of the graph.

1.5 Paths and Distances

- The distance $d_G(v_i, v_j)$ between two vertices is the number of edges along the shortest path.
- A graph **diameter** is the largest shortest path between any two vertices,

$$D = \max_{i,j} d_G(v_i, v_j). \tag{4}$$

- The average path length is bounded above by this diameter (essentially an average over all distances, note symmetry),

$$\langle L \rangle = \frac{1}{n(n-1)} \sum_{i \neq j} d_G(v_i, v_j). \tag{5}$$

1.6 Graph Transitivity

The **transitivity** of a graph, or the global clustering coefficient,

$$T = \frac{3 \times \text{number of triangle-connected units (triads)}}{\text{number of connected triplets of vertices}}. \tag{6}$$

Here, connected triplets are three connected nodes, but not necessarily by a closed triangle (numerator).

1.7 Clustering Coefficient

How neighbors of a given node are connected to each other.

- The **local clustering coefficient (per vertex)**,

$$C_i = \frac{\text{number of links in } N_i}{k_i(k_i - 1)/2}. \tag{7}$$

  This is essentially a measure of the interconnections of the direct neighbors of a specific node.

- The **average clustering coefficient**,

$$\bar{C} = \frac{1}{n} \sum_{i=1}^{n} C_i. \tag{8}$$

1.8 Complex Networks

- Scale-free networks – power-law node degree distributions :)
- Small-world networks – small diameters and average path lengths
- Transitive networks – high clustering coefficient (common in social networks)

Many real social networks follow power-law degree distributions. These are 'scale-free'. Probably because the shape of the distribution appears the same at any degree-scale when plotted on a log-log axis. We can describe these k (node) distributions with,

$$p(k) = Ck^{-\gamma}. \tag{9}$$

Here, C and γ are constants.

It is common to have triads in social networks – friends of friends are also friends.

2 Lecture 2 – Power-Law and Scale-Free Networks

2.1 Scale-Free Networks

Early work shows that internet network degrees of connectivity are scale-free, following power laws. Again, we have the power-law form for nodes, k,

$$P(k) = Ck^{-\gamma}. \tag{10}$$

We can also think about an expression in log-log space,

$$\log\ P_k = -\gamma \log\ k + \log\ C. \tag{11}$$

We also have normalization for our constant, C,

$$\sum_{k=1}^{\infty} P_k = C\sum_{k=1}^{\infty} k^{-\gamma} = C\zeta(\gamma) = 1; C = \frac{1}{\zeta(\gamma)}. \tag{12}$$

To point out, we have this Riemann zeta function for $\gamma > 1$,

$$P_k = \frac{k^{-\gamma}}{\zeta(\gamma)}. \tag{13}$$

Assuming that k is a real number, and taking continuous values,

$$P(k) = Ck^{-\gamma},\ \text{for}\ k \geq k_{min}. \tag{14}$$

Then you can perform an integral to solve for your constant, C. We can define our power-law PDF,

$$p(k) = \frac{\gamma - 1}{k_{min}}\left(\frac{k}{k_{min}}\right)^{\gamma}. \tag{15}$$

2.2 Poisson Distribution

Poisson node degree distributions occur given that each pair of nodes has some set, constant probability of connection.

2.3 Hubs in Networks

- How does the network size affect the size of its hubs? What is the natural cutoff?
- Probability of network with a node of degree greater than k_{max}?

$$Pr(k \geq k_{k_{max}}) = \int_{k_{max}}^{\infty} p(k)dk. \tag{16}$$

- The expected number of nodes with degree greater than or equal to this maximal number.. k_{max}, should be 1.
- The expected largest node degree in an exponential network of size N, $p(k) = Ce^{-\lambda k}$,

$$k_{max} = k_{min} + \frac{\ln N}{\lambda}. \tag{17}$$

- The expected largest node degree in a power-law (PL) network of size N, $p(k) = Ck^{-\lambda}$,

$$k_{max} = k_{min}N^{\frac{1}{\gamma - 1}}. \tag{18}$$

- In a random network, we have $k_{max} \propto \ln\ N$.

2.4 Moments

- We can also think about the moments of these power-law distributions. The expectation for the first moment,

$$\langle k \rangle = \int_{k_{min}}^{\infty} kp(k)dk = \frac{\gamma - 1}{\gamma - 2} k_{min}. \tag{19}$$

  Note that this is only defined for $\gamma > 2$. For higher moments of order m, we must require $\gamma > m + 1$. $\langle k^2 \rangle$ can then not be calculated as a finite number for $2 < \gamma < 3$!

- This can be used for determining the variance of a graph, i.e. $\sigma^2 = \langle k^2 \rangle - \langle k \rangle^2$.

- We can consider the degree of a randomly chosen node, $k = \langle k \rangle \pm \sigma_k$, $\sigma_k^2 = \langle k^2 \rangle - \langle k \rangle^2$.

- In a Poisson degree distribution, a random network has a scale $\langle k \rangle$,

$$k = \langle k \rangle \pm \sqrt{\langle k \rangle}. \tag{20}$$

- However, a power-law network with $2 < \gamma < 3$ is scale free,

$$k = \langle k \rangle \pm \infty. \tag{21}$$

  "There is no degree with respect to which most nodes have their degree."

2.5 Scale-Free Networks

Often, we have networks that are partially scale free, but may show regions where there is in fact some characteristic scale. We can also classify properties of networks based on their spectral index, γ.

- $\gamma < 2$: The "anomalous regime", where no network can exist. Both the mean and variance of the network degree distribution diverges. The average path length is constant. There maximal degree grows faster than the size of the network.

- $2 \leq \gamma < 3$: The "scale-free regime', where a network can exist. Here, the mean of of the degree distribution exists, but the variance is not defined. The average distance follows lnln N.

- $\gamma > 3$: The "random regime", where a random network can exist. Here, both the mean and variance are defined. The average distance grows as $\langle d \rangle \propto \ln N / \ln \langle k \rangle$.

2.6 Power Laws

- Recall that we also have a cumulative distribution function (CDF) for each probability density function (PDF). For our power law, $p(k)$,

$$F(k) = Pr(k_i \leq k) = \int_0^k p(k)dk. \tag{22}$$

- We have the complimentary cumulative distribution function (cCDF),

$$\bar{F}(k) = Pr(k_i > k) = 1 - F(k) = \int_k^{\infty} p(k)dk. \tag{23}$$

- If we would like to fit or estimate this value, γ, of course we can do some maximum likelihood estimation :)
- $\{x_i\}$ is a set of n observations or measurements of the distribution,

$$P(x_i) = \frac{\gamma - 1}{x_{min}} \left(\frac{x_i}{x_{min}} \right)^{-\gamma}. \tag{24}$$

- Then we have the probability of our sample, for a given γ, which is a product over probability density functions,

$$P(\{x_i\}|\gamma) = \prod_i^n \frac{\gamma - 1}{x_{min}} \left(\frac{x_i}{x_{min}} \right)^{-\gamma}. \tag{25}$$

- And also Bayes' theorem, which gives the probability of a specific spectral index given our observations,

$$P(\gamma|\{x_i\}) = P(\{x_i\}|\gamma) \frac{P(\gamma)}{P(\{x_i\})}. \tag{26}$$

- We can then define our log-likelihood, $\mathcal{L}$,

$$\mathcal{L} = \ln\ P(\gamma|\{x_i\}) = n \ln\ (\gamma - 1) - n \ln\ x_{min} - \gamma \sum_{i=1}^{n} \ln\ \frac{x_i}{x_{min}}. \tag{27}$$

- Which is maximized such that, $\frac{\partial \mathcal{L}}{\partial \gamma} = 0$,

$$\gamma = 1 + n \left[\sum_{i=1}^{n} \ln\ \frac{x_i}{x_{min}} \right]^{-1}. \tag{28}$$

- Finally, we have an error estimate,

$$\sigma = \sqrt{n} \left[\sum_{i=1}^{n} \ln\ \frac{x_i}{x_{min}} \right]^{-1} = \frac{\gamma - 1}{\sqrt{n}}. \tag{29}$$

- An important question is our choice of x_{min}, the lowest degree. To do this, we can compare the experimental and model CDFs (as a function of degree), to select this minimal degree. We can determine the largest difference between theoretical/model CDF and our experimental CDF for each assumed x_{min}, and then choose the x_{min} yielding this smallest difference. This is a Kolmogorov-Smirnov test.

$$D = \max\ |D(x|\gamma, x_{min} - F_{exp}(x)|, \tag{30}$$

$$x^*_{min} = \text{argmin}_{x_{min}} D. \tag{31}$$

3 Lecture 3 – Random Graphs

3.1 Network Models

Empirical network features:

- Power-law, heavy-tailed degree distribution
- Small average distance (diameter)
- Large clustering coefficient, transitivity

3.2 Random Graph Model

We have a graph $G = (V, E)$, a set of vertices V and a set of edges E, where $n = |V|, m = |E|$. In a random graph model,

- $G(n, m)$ is a randomly selected graph from the set of C_N^m graphs, $N = \frac{n(n-1)}{2}$, with n nodes and m edges.
- $G(n, p)$ each pair out of $N = \frac{n(n-1)}{2}$ pairs of nodes is connected with probability p and m is random,

$$\langle m \rangle = p\frac{n(n-1)}{2}, \tag{32}$$

$$\langle k \rangle = \frac{1}{n}\sum_i k_i = \frac{2\langle m \rangle}{n} = p(n-1) \approx pn, \tag{33}$$

$$\rho = \frac{\langle m \rangle}{n(n-1)/2} = p. \tag{34}$$

  Here, $\langle k \rangle$ is average node degree in the graph. ρ is a graph density.

- In a $G(n, p)$ model, the probability for a network to have m nodes is given by a binomial distribution,

$$P(m) = C_N^m p^m (1-p)^{N-m}. \tag{35}$$

  $N = \frac{n(n-1)}{n}$. p^m is the probability that m links are present. $(1-p)^{N-m}$ is the probability that other links are not. C_N^m number of ways to select m links out of N

- We then have the expectation for the number of links,

$$\langle m \rangle = \sum_{m=0}^{N} mP(m) = pN = p\frac{n(n-1)}{2}. \tag{36}$$

3.3 Random Graph Degree Distribution

- Probability that ith node has degree k is given by a binomial distribution,

$$P(k_i = k) = P(k) = C_{n-1}^k p^k (1-p)^{n-1-k}. \tag{37}$$

  Again p_k is the probability of connecting to k nodes, $(1-p)^{n-k-1}$ is the probability of not connecting to any other node. And, $C_{n-1}^k = \frac{(n-1)!}{k!(n-k-1)!}$

- When $\langle k \rangle << N$ or $n \to \infty$ and $p \to 0$, the binomial distribution approaches a Poisson distribution:

$$P(k) = \frac{\langle k \rangle^k e^{-\langle k \rangle}}{k!} = \frac{\Lambda^k e^{-\lambda}}{k!}, \langle k \rangle = pn = \lambda. \tag{38}$$

- Random graphs generally follow a Poisson degree distribution :)
- p is the probability of two connected nodes. If zero, the graph is unconnected. If one, the graph is completely connected or full – every node is connected to every other node. As p increases, we move from zero connection to full connection, this is a **phase transition** on a graph.

3.4 Phase Transition

- Consider u as the fraction of nodes not connected to the greatest connected component (GCC). This probability of not being connected,

$$u = \frac{n - n_G}{n} = P(k=0) + P(k=1) \cdot u + P(k=2) \cdot u^2 + \cdots \tag{39}$$

$$= \sum_{k=0}^{\infty} P(k) u^k = \sum_{k=0} \frac{\lambda^k e^{-\lambda}}{k!} u^k = e^{-\lambda} e^{\lambda u} = e^{\lambda(u-1}. \tag{40}$$

- This first sum represents connections to clusters of various sizes that are just not the GCC itself. And let s be the fraction of nodes belonging to the GCC,

$$u = e^{\lambda(u-1)}, \tag{41}$$

$$s = 1 - u. \tag{42}$$

- $\lambda = pn = \langle k \rangle$, when $\lambda \to \infty, s = 1$, when $\lambda \to 0, s = 0$
- We can also define a critical value, $\lambda_c = \langle k \rangle = p_c n = 1$. This is the point of phase transition when the GCC begins to form. This transition occurs when the probability of connection to another node is one, and the GCC participation grows to a maximal value as this probability increases.
- For a graph $G(n, p)$, for $n \to \infty$, the critical value is $p_c = 1/n$.
- We can also define a subcritical regime: $p < p_c, \langle k \rangle < 1$, there are no components with more than $\sim \ln n$ nodes. The largest component is a **tree**.
- At the critical point, $p = p_c, \langle k \rangle = 1$, the largest component has $\sim n^{2/3}$ nodes.
- In the supercritical $p > p_c, \langle k \rangle > 1$, the GCC has essentially all $\sim (p - p_c)n$ nodes.
- In the connected regime, $p >> \ln\, n/n, \langle k \rangle > \ln\, n$.

3.5 Threshold Probabilities

- We can also define probabilities for when different subgraphs of k nodes and l edges appear in a random graph $p_s \sim n^{-k/l}$. When $p > p_s$...
- $p_s \sim n^{-k/(k-1)}$, we are likely to have a tree with k nodes ...
- $p_s \sim n^{-1}$, a cycle with k nodes
- $p_s \sim n^{-2/(k-1)}$, subgraph with k nodes

3.6 Clustering Coefficient

- Probability that two neighbors link to each other,

$$C_i(k) = \frac{\text{links between NN}}{\text{max number of links}} = \frac{pk(k-1)/2}{k(k-1)/2} = p. \tag{43}$$

3.7 Graph Diameter

- $n = 1 + \langle k \rangle + \langle k \rangle^2 + \cdots = \dfrac{\langle k \rangle^{D+1} - 1}{\langle k \rangle - 1} \approx \langle k \rangle^D$.
- Around p_c, $\langle k \rangle^D \sim n$,

$$D \sim \frac{\ln n}{\ln \langle k \rangle}. \tag{44}$$

- In a small world, the network distance follows a logarithmic trend.

3.8 Configuration Model

- Can construct a random network with a predefined degree sequence, $D = \{k_1, k_2, k_3, \cdots k_n\}$, n nodes and $m = 1/2 \sum_i k_i$ edges.
- Randomly match two stubs and connect them by an edge. Can have self **loops** and multiple edges :)
- The probability that two nodes i and j are connected,

$$p_{ij} = \frac{k_i k_j}{2m - 1}. \tag{45}$$

- Forms a graph for your special graphical degree sequence. These can be nice for comparison against some real non-random graph.

4 Lecture 4 – Network Models

4.1 Preferential Attachment Model

Barabasi and Albert, 1999, a dynamic growth model – start at $t = 0$ with n_0 nodes and $m_0 \geq n_0$ edges,

- Growth – at each time step, add a new node with m edges $m_0 \leq n_0$, connecting to m nodes within the network.
- Preferential attachment – the probability of linking to an existing node i is proportional to the node degree k_i,

$$\Pi(k_i) = \frac{k_i}{\sum_i k_i}. \tag{46}$$

- Nodes and edges are then increasing with time as described.
- We can also describe this with a continuous time, degree approximation,

$$k_i(t + \delta t) = k_i(t) + m\Pi(k_i)\delta t, \tag{47}$$

$$\frac{dk_i t}{dt} = m\Pi(k_i) = m\frac{k_i}{\sum_i k_{i]}} = \frac{mk_i}{2mt} = \frac{k_i(t)}{2t}. \tag{48}$$

A node i is added at time t_i: $k_i(t_i) = m$,

$$\int_m^{k_i(t)} \frac{dk_i}{k_i} = \int_{t_i}^{t} \frac{dt}{2t}. \tag{49}$$

The solution is then,

$$k_i(t) = m\left(\frac{t}{t_i}\right)^{1/2}. \tag{50}$$

- In such a network, earlier nodes ultimately develop more connections, vertices over time. We want to consider the probability of a randomly selected node to have $k' \leq k$, $P(k' \leq k)$ at time t,

$$m\left(\frac{t}{t_i}\right)^{1/2} \leq k \rightarrow t_i \geq \frac{m^2}{k^2}t. \tag{51}$$

- We then have the cumulative function,

$$F(k) = P(k' \leq k) = \frac{n_0 + t - m^2 t/k^2}{n_0 + t} \approx 1 - \frac{m^2}{k^2}. \tag{52}$$

- As well as the distribution function,

$$P(k) = \frac{d}{dk}F(k) = \frac{2m^2}{k^3}. \tag{53}$$

- Last, we also have the average path length,

$$\langle L \rangle \sim \log(N)/\log(\log(N)), \tag{54}$$

and clustering coefficient, $C \sim N^{-0.75}$.

4.2 Growing a Random Graph

- Growth – at each time step add a new node of m edges ($m \leq n_0$), connected to m nodes already in the network.
- Attachment uniformly at random – the probability of linking to an existing node, i is,

$$\Pi(k_i) = \frac{1}{n_0 + t - 1}. \tag{55}$$

- The node degrees grow as,

$$k_i(t) = m\left(1 + \log\left(\frac{t}{i}\right)\right), \tag{56}$$

with node degree distribution function,

$$P(k) = \frac{e}{m}\exp(-k/m). \tag{57}$$

4.3 Non-linear Preferential Attachment

- If we have a non-linear preferential attachment model,

$$\Pi(k) \sim k^{\alpha}. \tag{58}$$

- $\alpha = 0$, no hubs, exponential distribution
- $0 < \alpha < 1$, sublinear, smaller hubs, a stretched exponential
- $\alpha = 1$, scale-free, hubs, PL distribution
- $\alpha > 1$, superlinear, super hubs

	Random	BA Model	WS Model	Empirical Networks
P(k)	$\frac{\lambda^k e^{-\lambda}}{k!}$	k^{-3}	Poisson-like	power law
C	$\langle k \rangle / N$	$N^{-0.75}$	const.	large
$\langle L \rangle$	$\frac{\log(N)}{\log(\langle k \rangle)}$	$\frac{\log(N)}{\log \log(N)}$	$\log(N)$	small

Table 1: Characteristics of different model networks.

4.4 Link Selection Model

- Growth – add a new node at each time step.
- Link selection – select links at random and connect one of the two nodes at the ends.
- Probability to connect to a node with degree k,

$$q_k \frac{k p_k}{\langle k \rangle}. \tag{59}$$

- This gives a scale-free distribution.

4.5 Copying Model

- Random connection – with probability p connect to a random node u.
- Copying – with probability $p - 1$, randomly choose an outgoing link from u and connect to its target.
- The probability to connect to a node of degree k,

$$\Pi(k) = \frac{p}{n} + (1 - p)\frac{k}{2m}. \tag{60}$$

- Again, the result is scale free.

4.6 Small World Model

- Motivation is to keep high clustering. You will also end up with a large diameter. Really need an image for this ...
- Watts and Strogatz, 1998 – single parameter model, interpolation between a regular lattice and a random graph...
- Start with a regular lattice with n nodes, k edges per vertex and $k << n$.
- Randomly connect with other nodes with probability p, forms $pnk/2$ "long-distance" connections from a total of $nk/2$ edges.
- Note, $p = 0$ is a regular lattice, $p = 1$ is a random graph. Transitioning from the lattice to random graph changes the path length and clustering coefficient.

5 Lecture 5 – Node Centrality, Network Ranking

5.1 Node Centrality

- Some nodes may be more central or with a higher degree than others.
- **Degree Centrality** – number of nearest neighbors,

$$C_D(i) = k(i) = \sum_j A_{ij} = \sum_j A_{ji}. \tag{61}$$

- We also have the normalized degree centrality,

$$C_D^*(i) = \frac{1}{n-1} C_D(i) = \frac{k(i)}{n-1}. \tag{62}$$

- High degree-centrality – indicates a node in contact with many others.

5.2 Closeness Centrality

- **Closeness Centrality** – how close is an actor or agent to all other actors in a network,

$$C_C(i) = \frac{1}{\sum_j d(i,j)}. \tag{63}$$

- We also have the normalized closeness centrality,

$$C_C^* = (n-1) C_C(i). \tag{64}$$

- A high closeness centrality indicates a short communication path to all others, a minimal number of steps to all others.
- We can also define a **harmonic centrality**,

$$C_H(i) = \sum_j \frac{1}{d(i,j)}. \tag{65}$$

5.3 Betweenness Centrality

- **Betweenness Centrality** – the number of shortest paths going through the actor $\sigma_{st}(i)$,

$$C_B(i) = \sum_{s \neq t \neq i} \frac{\sigma_{st}(i)}{\sigma_{st}}. \tag{66}$$

- We also have the normalized betweenness centrality,

$$C_B^*(i) = \frac{2}{(n-1)(n-2)} C_B(i). \tag{67}$$

- With a high betweenness centrality, the vertex lies on many shortest paths.

5.4 Eigenvector Centrality

- The importance of a node depends on the importance of its neighbors (recursive definition),

$$v_i \leftarrow \sum_j A_{ij} v_j \tag{68}$$

$$v_i = \frac{1}{\lambda} \sum_j A_{ij} v_j \tag{69}$$

$$\mathbf{A}\mathbf{v} = \lambda \mathbf{v}. \tag{70}$$

- Select an eigenvector associated with the largest eigenvalue, $\lambda = \lambda_1$, $\mathbf{v} = \mathbf{v}_1$. The node that is connected to other important nodes is important. Note that the determined importance is normalized. Also, this works only for undirected networks, where A is then symmetric, and the existence of eigenvectors is guaranteed.

5.5 Katz Status Index

- The weighted count of all paths coming to the node, the weight of path of length n is counted with attenuation factor β^n, $\beta < \dfrac{1}{\lambda_1}$,

$$k_i = \beta \sum_j A_{ij} + \beta^2 \sum_j A_{ij}^2 + \beta^3 \sum_j A_{ij}^3 + \cdots \tag{71}$$

$$\mathbf{k} = (\beta \mathbf{A} + \beta^2 \mathbf{A}^2 + \cdots)\mathbf{e} = \left(\sum_{n=0}^{\infty} (\beta \mathbf{A})^n - \mathbf{I} \right) \mathbf{e} \tag{72}$$

$$\mathbf{k} = \beta (\mathbf{I} - \beta \mathbf{A})^{-1} \mathbf{A} \mathbf{e}. \tag{73}$$

5.6 Directed Graphs

- **Sinks** – zero out degree nodes, $k_{out}(i) = 0$
- **Sources** – zero in degree nodes, $k_{in}(i) = 0$
- Centrality measures on directed graphs?
- Define both an in- and out-degree centrality,

$$C_D^{in}(i) = \frac{k^{in}(i)}{n-1}; C_D^{out}(i) = \frac{k^{out}(i)}{n-1}. \tag{74}$$

- Closeness centrality, normalized,

$$C_C(i) = \frac{n-1}{\sum_j d(i,j)}. \tag{75}$$

- Betweenness centrality, normalized,

$$C_B(i) = \frac{1}{(n-1)(n-2)} \sum_{s \neq t \neq i} \frac{\sigma_{st}(i)}{\sigma_{st}}. \tag{76}$$

5.7 Some Graph Theory Notions

- A graph is **strongly connected** if every vertex is reachable from every other vertex.
- Strongly connected components are partitions of a graph which are strongly connected.
- If strongly connected, there is a path in either direction between any two pairs of nodes/vertices.
- A graph is **aperiodic** if the greatest common divisor of the lengths of its cycles is one (no integer greater than one divides the length of *every* cycle of the graph).

5.8 PageRank

- The internet hyperlinks as a graph ... **PageRank** is the probability of a random web surfer visiting a page from a hyperlink, or jumping somewhere else new.
- A **random walk** on a directed graph ...

$$p_i^{t+i} = \sum_{j \in N(i)} \frac{p_j^t}{d_j^{out}} = \sum_j \frac{A_{ji}}{d_j^{out}} p_j. \tag{77}$$

$$\mathbf{D}_{ii} = diag\{d_i^{out}\} \tag{78}$$
$$\mathbf{p}^{t+1} = (\mathbf{D}^{-1}\mathbf{A})^T \mathbf{p}^t \tag{79}$$
$$\mathbf{P} = \mathbf{D}^{-1}\mathbf{A} \tag{80}$$

- We can then iterate,

$$\mathbf{p}^{t+1} \to \mathbf{P}^T \mathbf{p}^t. \tag{81}$$

- These are power iterations,

$$\mathbf{p} \to \alpha \mathbf{P}^T \mathbf{p} + (1-\alpha)\frac{\mathbf{e}}{n}. \tag{82}$$

  Here, we have some probability of following the graph (first term), or some probability of jumping to a new location on the graph (second term). α is this **teleportation coefficient**. We will have this sparse linear system,

$$(\mathbf{I} - \alpha \mathbf{P}^T)\mathbf{p} = (1-\alpha)\frac{\mathbf{e}}{n}. \tag{83}$$

  As well as this resulting eigenvalue problem,

$$(\alpha \mathbf{P}^T + (1-\alpha)\mathbf{E})p = \lambda \mathbf{p} \tag{84}$$

  .

- Know that there are many iterations on PageRank for different use cases.

5.9 Perron-Frobenius Theorem

- **Perron-Frobenius theorem** – the fundamental theorem of Markov chains, if the matrix is,
  - stochastic (non-negative and rows sum up to one, describes a Markov chain)
  - irreducible, a strongly connected graph
  - aperiodic, $\exists \lim_{t\to\infty} \bar{p}^t = \bar{\pi}$.

  And, there are eigenvectors such that,

$$\bar{\pi}P = \lambda\bar{\pi},\ ||\bar{\pi}||_1 = 1, \lambda = 1. \tag{85}$$

  Here, $\bar{\pi}$ is a stationary distribution of Markov chain, a row vector.

- Essentially, on an infinite random walk (strongly connected), we determine the frequency of each location visited, and normalize this distribution for the probability of visitation.

5.10 Hubs and Authorities (HITS)

- Good authorities are referred to by good hubs. Good hubs point to good authorities.

$$a_i \leftarrow \sum_j A_{ji} h_j, \tag{86}$$

$$h_i \leftarrow \sum_j A_{ij} a_j. \tag{87}$$

- Then we again have a system of linear equations,

$$\mathbf{a} = \alpha \mathbf{A}^T \mathbf{h}, \tag{88}$$

$$\mathbf{h} = \beta \mathbf{A}\mathbf{a}. \tag{89}$$

  As well as a symmetric eigenvalue problem,

$$(\mathbf{A}^T\mathbf{A})\mathbf{a} = \lambda\mathbf{a}, \tag{90}$$

$$(\mathbf{A}\mathbf{A}^T)\mathbf{h} = \lambda\mathbf{h}. \tag{91}$$

  Here, $\lambda = (\alpha\beta)^{-1}$.

6 Lecture 6 – Structural Properties of Networks

6.1 Patterns of Relations

- Global, statistical properties of networks – average node degree, average clustering, average path length
- Local vertex properties – node centrality, page rank
- Pairwise properties – node equivalence, node similarity, correlation between pairs of vertices

6.2 Structural Equivalence

- Two vertices are **structurally equivalent** if their respective sets of in- and out-neighbors are the same.
- The rows and columns of the adjacency matrix for the structurally equivalent nodes are identical.
- Self loops make strong structural equivalence. possible

6.3 Structural Similarity

- Nodes are similar if they share many neighbors. We can consider different similarity measures,
  - Jaccard similarity,
$$J(v_i, v_j) = \frac{|N(v_i) \cap N(v_j)}{|N(v_i) \cup N(v_j)|}. \tag{92}$$
  - Cosine similarity (vectors in an n-dimensional space),
$$\sigma(v_i, v_j) = cos(\theta_{ij}) = \frac{\mathbf{v}_i^T \mathbf{v}_j}{|\mathbf{v}_i||\mathbf{v}_j|} = \frac{\sum_k A_{ik} A_{kj}}{\sqrt{\sum A_{ik}^2}\sqrt{\sum A_{jk}^2}}. \tag{93}$$
  - Pearson Correlation Coefficient ...
- We can then construct a node similarity matrix.

6.4 SimRank

- We assume we have a directed graph. Two vertices are similar if they are referenced by similar vertices.
- $s(a, b)$ – the similarity between a and b, $l()$, the set of in-neighbors,
$$s(a, b) = \frac{C}{|l(a)||l(b)|} \sum_{i=1}^{l(a)} \sum_{j=1}^{l(b)} s(l_i(a), l_j(b)), a \neq b, \tag{94}$$
$$s(a, a) = 1. \tag{95}$$
- Matrix notation,
$$S_{ij} = \frac{C}{k_i k_j} \sum_{k,m} A_{ki} A_{mj} S_{km}. \tag{96}$$
- We have an iterative solution starting with $s_0(i, j) = \delta_{ij}$.

6.5 Degree Correlation

- **Degree correlation** is the likelihood that nodes link to nodes with similar or dissimilar nodal degree.
- Pearson degree correlation coefficient ($-1 \leq r \leq 1$),

$$r = \frac{\text{cov}}{\text{var}} = \frac{\sum_{ij} A_{ij}(k_i - \langle k \rangle)(k_j - \langle k \rangle)}{\sum_{ij} A_{ij}(k_i - \langle k \rangle)^2}. \tag{97}$$

- The degree correlation matrix – the fraction of edges connecting nodes, degrees k, k',

$$e_{k,k'} = \frac{m(k,k')}{m}. \tag{98}$$

- The degree correlation function,

$$k_{nn}(k) = \sum_{k'} k' P(k'|k); P(k'|k) = \frac{e_{k,k'}}{\sum'_k e_{k,k'}}. \tag{99}$$

6.6 Assortative and Disassortative Networks

- **Assortative network** ($r > 0$) – hubs tend to connect to hubs, low degree nodes connect to low degree nodes. Similar agents prefer similar agents.
- **Disassortative network** ($r < 0$) – high degree nodes tend to connect to low degree nodes.
- **Assortative mixing** – links or mixing between similar nodes.
- **Disassortative mixing** – links or mixing between dissimilar nodes.

6.7 Mixing by Categorical Attributes

- Consider s categorical attribute c_i. How often do attributes match across edges compared to random?
- Modularity,

$$Q = \frac{m_c - \langle m_c \rangle}{m} = \frac{1}{2m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(c_i, c_j). \tag{100}$$

- m_c is the number of edges between vertices with the same attributes, $\langle m_c \rangle$ is the expected number of edges within the same class in a random network.
- We have the Assortativity coefficient,

$$\frac{Q}{Q_{max}} = \frac{\sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(c_i, c_j)}{2m - \sum_{ij} \frac{k_i k_j}{2m} \delta(c_i, c_j)}. \tag{101}$$

6.8 Mixing by Scalar Values

- We can consider a vertex scalar value, attribute, x_i,
- How much more similar are attributes across edges than expected at random?
- Average and covariance over edges,

$$\langle x \rangle = \frac{\sum_i k_i x_i}{\sum_i k_i}, \tag{102}$$

$$\text{var} = \frac{1}{2m} \sum_{ij} A_{ij}(x_i - \langle x \rangle)^2 = \frac{1}{2m} \sum_i k_i (x_i - \langle x \rangle)^2, \tag{103}$$

$$\text{cov} = \frac{1}{2m} \sum_{ij} A_{ij}(x_i - \langle x \rangle)(x_j - \langle x \rangle). \tag{104}$$

- We also have the assortativity coefficient, cov/var.

6.9 Friendship Paradox

- On average, your friends have more connections than you.
- Average neighbor degree of a node with degree k,

$$k_{nn} = \sum_{k'} k' P(k'|k). \tag{105}$$

- For an uncorrelated network,

$$k_{nn} = \sum_{k'} k' \frac{k' p(k')}{\langle k \rangle} = \frac{\langle k^2 \rangle}{\langle k \rangle}. \tag{106}$$

- In a random network,

$$\langle k^2 \rangle = \langle k \rangle (1 + \langle k \rangle). \tag{107}$$

- In scale free networks,

$$\langle k^2 \rangle / \langle k \rangle >> \langle k \rangle. \tag{108}$$

- In general, we are more likely to be friends with hubs because hubs have more connections.

7 Lecture 7 – Graph Partitioning Algorithms

7.1 Graph Partition

- **Graph partitioning** is the split of a graph into subgraphs by partitioning its nodes into mutually exclusive groups. The specific split is defined by the edges it crosses. Its size is the number of edges crossed.
- Exact solution is NP-hard (non-deterministic polynomial time).
- Number of ways to divide N nodes in 2 groups (bi-partition), $2^{N-1} - 1$ distinct cuts.

- A combinatorial optimization problem – ideal optimization criterion or optimization method. Can be solved by algorithms or heuristics.
- Concept of a balanced vs unbalanced partition.
- Can also have a multi-way partition, beyond 2 ways.

7.2 Graph Partitioning Algorithms

- Greedy optimization, local search
- Approximate optimization, spectral graph partitioning, multicommodity flow
- Randomized algorithms, randomized minimum cut
- Heuristic algorithms, multilevel graph partitioning
- See many corresponding citations online :)

7.3 Optimization Criterion

- We consider graph $G(E, V)$ and partition $V = V_1 + V_2$, and potential partitioning schemes...
- Graph cut,
$$Q = \text{cut}(V_1, V_2) = \sum_{i \in V_1, j \in V_2} e_{ij}. \tag{109}$$
- Ratio cut,
$$Q = \frac{\text{cut}(V_1, V_2)}{||V_1||} + \frac{\text{cut}(V_1, V_2)}{||V_2||}. \tag{110}$$
- Normalized cut,
$$Q = \frac{\text{cut}(V_1, V_2)}{\text{Vol}(V_1)} + \frac{\text{cut}(V_1, V_2)}{\text{Vol}(V_2)}. \tag{111}$$
- Quotient cut (conductance),
$$Q = \frac{\text{cut}(V_1, V_2)}{\min(\ \text{Vol}(V_1),\ \text{Vol}(V_2))}. \tag{112}$$
- Here, $\text{Vol}(V_1) = \sum_{i \in V_1, j \in V} = \sum_{i \in V_1} k_i$.

7.4 Randomized Minimum Cut

- We have Karger's algorithm for finding a minimum cut...
- Edge contraction – remove an edge and merge the two vertices or nodes that it previously joined.
- $P(\text{final cut} = \text{min cut}) \geq 2/n^2$. This can be run $\Omega(n^2)$ times and the smallest cut chosen.

7.5 Karger Minimum Cut

- Our input is the graph G and output is the class indicator vector mincutX.
- We randomly select edges to be removed until only two nodes are left.
- We repeat this process and then compare the resulting number of edges left between the two nodes.

7.6 Linear Algebra of Graph Cuts

- Let $V = V^+ + V^-$ by the partitioning of the nodes
- $s = \{+1, -1, +1, \cdots -1, +1\}$, a vector indicating,

$$s(i) = \begin{cases} +1: & \text{if } v(i) \in V^+ \\ -1: & \text{if } v(i) \in V^- \end{cases} \tag{113}$$

- We then have the number of edges connecting V^+, V^-,

$$\text{cut}(V^+, V^-) = \frac{1}{4} \sum_{e(i,j)} (s(i) - s(j))^2 = \frac{1}{4} \sum_{i,j} (k_i \delta_{i,j} - A_{ij}) s(i) s(j) \tag{114}$$

$$\text{cut}(V^+, V^-) = \frac{1}{4} \sum_{i,j} (D_{ij} - A_{ij}) s(i) s(j). \tag{115}$$

  Here, the sum is actually the size of the cut, the number of crossed edges.

7.7 Graph Cuts

- We define a **Graph Laplacian**, $\mathbf{L}_{ij} = \mathbf{D}_{ij} - \mathbf{A}_{ij}$, where $\mathbf{D}_{ii} = diag(k_i)$,

$$\mathbf{L}_{ij} = \begin{cases} k(i), & \text{if } i = j \\ -1, & \text{if } \exists e(i,j) \\ 0, \text{ otherwise} & \end{cases} \tag{116}$$

- Our graph cut is then,

$$Q(\mathbf{s}) = \text{cut}(V^+, V^-) = \frac{1}{4} \sum_{i,j} L_{ij} s(i) s(j) = \frac{\mathbf{s}^T \mathbf{L} \mathbf{s}}{4}. \tag{117}$$

- We take the minimal cut, $\min_{\mathbf{s}} Q(\mathbf{s})$.
- We then have the cut constraint, $\sum_i s(i) = 0$. An integer minimization problem, the exact solution is NP-hard.

7.8 Spectral Method – Relaxation

- Discrete $\rightarrow$ continuous problem
- In our discrete problem we find,
$$\min_{\mathbf{s}}\left(\frac{1}{4}\mathbf{s}^T\mathbf{L}\mathbf{s}\right). \tag{118}$$
Here, $s(i) = \pm 1, \sum_i s(i) = 0$.
- If we relax to a continuous problem ($x(i)$ a real number), we find,
$$\min_{\mathbf{x}}\left(\frac{1}{4}\mathbf{x}^T\mathbf{L}\mathbf{x}\right), \tag{119}$$
with constraints, $\sum_i x(i)^2 = n, \sum_i x(i) = 0$. We have $s(i) = sign(x(i))$. The exact constraint satisfies the relaxed equation. The relaxed equation is easier to solve.
- Our constraint problem is a Lagrange multiplier problem :),
$$Q(\mathbf{x}) = \frac{1}{4}\mathbf{x}^T\mathbf{L}\mathbf{x} - \lambda(\mathbf{x}^T\mathbf{x} - n), \mathbf{x}^T\mathbf{e} = 0. \tag{120}$$
This is a constraint optimization problem. We differentiate with respect to L and with respect to x.
- We have the eigenvalue problem,
$$\mathbf{L}\mathbf{x} = \lambda\mathbf{x}, \mathbf{x} \perp \mathbf{e}. \tag{121}$$
- With solution, $Q(\mathbf{x}_i) = \frac{n}{4}\lambda_i$.
- Our smallest eigenvector.. $\mathbf{L}\mathbf{e} = 0, \lambda = 0, \mathbf{x}_1 = 0$.
- To search for the next minimal eigenvalue, we minimize the Rayleigh-Ritz quotient,
$$\min_{\mathbf{x}\perp\mathbf{x}_1}\left(\frac{\mathbf{x}^T\mathbf{L}\mathbf{x}}{\mathbf{x}^T\mathbf{x}}\right). \tag{122}$$
- In a spectral graph partitioning algorithm, our input is the adjacency matrix, A, our output is the class indicator vector s. We compute D and L and solve for the second smallest eigenvector for our minimum cut.

7.9 Optimization Criterion

- As a final take, note that minimizing the number of edges cut does not guarantee the volume or weight of the final partitions is equivalent. This is why we also need optimization criterion like the normalized cut.

8 Lecture 8 – Community Detection

8.1 Community Detection

- How to detect clusters, e.g. social clusters?
- **Network communities** are groups of vertices such that vertices inside the group are connected with many more edges than between groups.
- Community detection is the assignment of vertices to communities. Note that we can also have overlapping communities.
- Partitioning can be used as a tool, but may not be helpful in the case of overlapping communities.

8.2 What Makes a Community?

- A community is a cohesive subgroup.
- A mutuality of ties. Almost everyone in the group has ties or edges with one another.
- Compactness. There is a closeness between members, e.g. with the number of steps.
- There is a density of edges. A high frequency of ties within the group. Also, a higher frequency relative to non-members (separation).

8.3 Community Density

- Consider a graph $G(E, V), n = |V|, m = |E|$.
- A community, a set of nodes S, n_s is the number of nodes, m_s is the number of edges in S.
- Graph density, (i.e. the ratio between the number of edges in the graph and the number of nodes),
$$\rho = \frac{m}{n(n-1)/2}. \tag{123}$$
- Community internal density,
$$\delta_{int} = \frac{m_s}{n_s(n_s-1)/2}. \tag{124}$$
- External edges density,
$$\delta_{ext} = \frac{m_{ext}}{n_s(n-n_s)}. \tag{125}$$
- Community (cluster): $\delta_{int} > \rho, \delta_{ext} < \rho$.
- Note that it is difficult to actually test all iterations, and density may not actually be 'sensitive' enough to detect groups.

8.4 Modularity

- Can compare fraction of edges within the cluster to the expected fraction in a random graph with an identical degree sequence. Modularity may be a preferable indicator of communities over density.
- Can then consider the modularity score,

$$Q = \frac{1}{2m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(c_i, c_j) = \sum_u \left(\frac{m_u}{m} - \left(\frac{k_u}{2m} \right)^2 \right). \tag{126}$$

- Here, m_u is the number of internal edges in a community u. k_u is the sum of node degrees within a community.
- We then have the modularity score range $Q \in [-1/2, 1)$. If there is a single community, $Q = 0$. A higher modularity score indicates better community selection.

8.5 Community by Optimal Partition

- We can also use the partitioning schemes described in the last lecture for community selection.
- At each partition, look at the modularity score of each separated piece.

8.6 Heuristic Approach

- We can consider edge-betweenness, the number of shortest paths $\sigma_{st}(e)$ going through edge e. These are often the bridges between communities,

$$C_B(e) = \sum_{s \neq t} \frac{\sigma_{st}(e)}{\sigma_{st}} \tag{127}$$

- Recover communities by progressively removing edges with largest edge-betweenness until graph left in two components.

8.7 Fast Community Unfolding

- A multi-resolution scalable method.
- First, assign every node to its own community
- For every node, determine modularity gain from removing node from its community and placing in the community of its neighbor. Place node in the community maximizing modularity gain. Repeat until there is no more improvement (local max modularity).
- Then, nodes form communities of merged "super nodes". The weight on the links is added up.
- The above algorithm is repeated until no further changes (maximum modularity).

8.8 Agglomerative Hierarchical Clustering

- Similarity matrix,

$$x_{ij} = \frac{N(i,j)}{\min k_i, k_j + 1 - \theta(A_{ij})} \tag{128}$$

- Here, N_{ij} is the number of common neighbors, θ is a step function.
- With the Ravasv algorithm,
  - Assign each node to its own community and evaluate x_{ij} for all node pairs.
  - Find the node pair with the highest similarity and merge them into a single community.
  - Calculate the similarity between the new community and all other communities.
  - Repeat the first steps until all nodes form a single community.
  - Find the optimal cut of the dendogram.

8.9 Communities and Random Walks, Walktrap

- Consider a random walk on a graph,
- At each time step the walk moves to a new node uniformly on a graph,

$$P_{ij} = \frac{A_{ij}}{d(i)}, P = D^{-1}A, D_{ii} = \text{diag}(d(i)). \tag{129}$$

- P_{ij}^t is the probability to get from i to j in t steps, $t << t_{mixing}$.
- For two i and j in the same community P_{ij}^t is high.
- If i and j are in the same community, then $\forall\ k, P_{ik}^t \approx P_{jk}^t$.
- We can then define a distance between nodes,

$$r_{ij}(t) = \sqrt{\sum_{k=1}^{n} \frac{(P_{ik}^t - P_{jk}^t)}{d(k)}} = ||D^{-1/2}P_i^t - D^{-1/2}P_j^t||. \tag{130}$$

- If we use a more approximate computation, we can compute k random walks of length t from a starting node i. We can approximate $P_{ik}^t \approx \frac{N_{ik}}{K}$, number of walks ending up on k.
- The distance between communities is then,

$$P_{Cj}^t = \frac{1}{|C|}\sum_{i \in C} P_{ij}^t, \tag{131}$$

$$r_{C_1C_2}(t) = \sqrt{\sum_{k=1}^{n} \frac{(P_{C_1k}^t - P_{C_2k}^t)}{d(k)}} = ||D^{-1/2}P_{C_1}^t - D^{-1/2}P_{C_2}^t||. \tag{132}$$

- An algorithm for hierarchical clustering,
  - We assign each vertex to its own community $S_1 = \{\{v\}, v \in V\}$.

- We compute the distance between all adjacent communities $r_{C_iC_j}$.
- We choose the two closest communities (based on Ward's methods) and merge them, $S_{k+1} = (S_k \ \{C_1, C_2\}) \cup C_3, C_3 = C_1 \cup C_2$.
- We update the distance between communities.
- After $n-1$ steps, we finish with a single community.

8.10 k-clique Community

- A **k-clique** is a complete subgraph with k nodes.
- A $k-$clique community is a union of all $k-$cliques that cab be reaches from each other through a series of adjacent $k-$cliques.
- Two $k-$cliques are adjacent if they share $k-1$ nodes.
- We can use the algorithm $CFinder$, find all maximal cliques, create a matrix of clip overlap, consider a threshold of $k-1$. We then select communities as connected components.

8.11 Real World Communities

- We can consider the best conductance of a vertex set S of size k,

$$\Phi(k) = \min_{S \in V, |S| \in V} \phi(S), \phi(S) = \frac{\text{cut}(S, V/S)}{\min(\text{vol}(S), \text{vol}(S/V))}. \tag{133}$$

  Here, $\text{vol}(S) = \sum_{i \in S} k_i$ is the sum of all node degrees in the set.

9 Lecture 9 – Epidemics on Networks I

9.1 Epidemic Branching Process

- $R_0 = p\langle k \rangle$ – the average number of newly infected nodes at each step.
- On the nth step, the average number of infected people, $R_0^n = (p\langle k \rangle)^n$.
- If $R_0 > 1$, the average grows geometrically as R_0^n.
- If $R_0 < 1$, the average shrinks geometrically as R_0^n.
- If $n \to t$, geometric growth transitions to exponential growth.
- R_0 is the basic reproduction number, the average number of secondary infections when one infected individual is introduced into a host population where everyone is susceptible. $R_0 = 1$ is the threshold that determines when an infection can invade and persist in a new host population.

9.2 Compartmental Models in Epidemiology

- Mathematical epidemiology
- We can consider a deterministic compartmental model with population classes $\{S, I, R\}$.
- $S(t)$ – the susceptible class, the individuals not yet infected with the disease at time t.
- $I(t)$ – the infected class, the individuals who have been infected with the disease and are capable of. spreading it at t.
- $R(t)$ – the recovered class, the number of individuals who have been infected and then recovered from the disease. They cannot be infected again or transmit to others.
- We can consider a fully-mixing model and a closed population (no birth, death, migration).

9.3 SI Model

- $S(t)$, susceptible group and $I(t)$, infected group,

$$S(t) + I(t) \to N \tag{134}$$

- β is a transmission or infection rate, the number of transmitting contacts per unit time. $T_c = 1/\beta$, is the time between transmitting contact.
- We can then consider an infection equation,

$$I(t + \delta t) = I(t) + \beta \frac{S(t)}{N} I(t), \delta t \tag{135}$$

$$\frac{dI(t)}{dt} = \beta \frac{S(t)}{N} I(t). \tag{136}$$

- Can consider $i(t) = I(t)/N$, $s(t) = S(t)/N$.
- We have,

$$\frac{di(t)}{dt} = \beta s(t) i(t) \tag{137}$$

$$\frac{ds(t)}{dt} = -\beta s(t) i(t) \tag{138}$$

$$s(t) + i(t) = 1 \tag{139}$$

- We can then introduce the differential equation, $i(t = 0) = i_0$,

$$\frac{di(t)}{dt} = \beta(1 - i(t)) i(t). \tag{140}$$

- Given our set of differential equations, we must also consider boundary conditions or initial conditions, i_0.
- Our solution is a logistic growth function,

$$i(t) = \frac{i_0}{i_0 + (1 - i_0)e^{-\beta t}}. \tag{141}$$

As $t \to \infty$, $i(t) \to 1$, $s(t) \to 0$.

9.4 SIS Model

- In an SIS model, we have susceptible populations, as well as infected populations which can later recover (to then become susceptible). $S \to I \to S$. $S(t) + I(t) = N$.
- β is an infection rate upon contact. γ is a recovery rate upon contact. $T_r = 1/\gamma$ is the average time to recovery.
- We have our infection equations,

$$\frac{ds}{dt} = -\beta si + \gamma i \tag{142}$$
$$\frac{di}{dt} = \beta si - \gamma i \tag{143}$$
$$s + i = 1 \tag{144}$$

- With $i(t = 0) = i_0$ we then have,

$$\frac{di}{dt} = (\beta - \gamma - \beta i)i. \tag{145}$$

- We have the solution,

$$i(t) = \left(1 = \frac{\gamma}{\beta}\right)\frac{C}{C + e^{-(\beta-\gamma)t}}, \tag{146}$$
$$C = \frac{\beta i_0}{\beta - \gamma - \beta i_0}. \tag{147}$$

- In the limit $t \to \infty$, we have logistic solutions,

$$\beta > \gamma, i(t) \to \left(1 - \frac{\gamma}{\beta}\right), \tag{148}$$
$$\beta < \gamma, i(t) = i_0 e^{(\beta-\gamma)t} \to 0. \tag{149}$$

- We can also define an epidemic threshold at $R_0 = \beta/\gamma = 1$.

9.5 SIR Model

- $S(t), I(t), R(t)$ (recovered) groups, where $S \to I \to R$ and $S(t) + I(t) + R(t) = N$. We have β as an infection rate and γ as a recovery rate.
- We have the infection equation,

$$\begin{cases} \frac{ds}{dt} = -\beta si \\ \frac{di}{dt} = \beta si - \gamma i \\ \frac{dr}{dt} = \gamma i \\ s + i + r = 1 \end{cases} \tag{150}$$

Model	Early Time	Late Time	Epidemic Threshold
SI	$i_0 e^{\beta_c t}$	1	–
SIS	$(1-\frac{\gamma}{\beta_c})e^{(\beta_c-\gamma)t}$	$1-\frac{\gamma}{\beta_c}$, 0	$\beta_c/\gamma = 1$
SIR	exponential	0	$\beta_c/\gamma = 1$

Table 2: A summary of simple epidemic models presented earlier.

- We then have the equations and solution, t,

$$\frac{ds}{dt} = -\beta s \frac{dr}{dt}\frac{1}{\gamma} \tag{151}$$

$$s = s_0 e^{-\beta r/\gamma} \tag{152}$$

$$\frac{dr}{dt} = \gamma(1 - r - s_0 e^{-\beta r/\gamma}) \tag{153}$$

$$t = \frac{1}{\gamma}\int_0^r \frac{dr}{1 - r - s_0 e^{-\beta r/\gamma}}. \tag{154}$$

- As $t \to \infty$, $dr/dt = 0$,

$$1 - r_\infty = s_0 e^{-\beta r/\gamma} r_\infty. \tag{155}$$

- We also define the threshold $R_0 = \beta/\gamma$ and consider the critical point of $R_0 = 1$.

9.6 SEIIHURD Model

- We can also have more complex models
- In an SEIIHURD model, we consider a multi-component model for disease,
  - S – susceptible
  - E – exposed
  - I – infectious (both symptomatic and asymptomatic)
  - H – hospitalized
  - U – ICU units
  - R – recovered
  - D – dead

10 Lecture 10 – Epidemics on Networks II

10.1 Compartmental Models Summary

10.2 Probabilistic Node-Level Model

- We have a network of *potential* contacts, (adjacency matrix A).
- We can consider the probabilistic model (state of a node),
  - $s_i(t)$ – probability that at time t a node i is susceptible
  - $x_i(t)$ – probability that at time t a node i is infected

- $r_i(t)$ – probability that at time t a node i is recovered

- β – probability that the disease will be transmitted on a contact in time δt (for a compartmental model $\beta_c = \beta\langle k\rangle$)

- γ – recovery rate (probability to recover in a unit time δt)

- This is a probabilistic description. We can also assume this occurs for a connected graph where all nodes are reachable and that the network is undirected (A is symmetric).

- $P_{\text{inf}} \approx \beta s_i(t) \sum_{j\in N(i)} x_j(t)\delta t$ (node infection).

- $P_{\text{rec}} = \gamma x_i(t)\delta t$ (node recovery).

10.3 SI Model

- $S \to I$

- Probabilities that node i, susceptible, $s_i(t)$, or infected, $x_i(t)$, such that $x_i(t) + s_i(t) = 1$.

- β is our infection rate, probability of becoming infected in a unit of time,

$$x_i(t+\delta t) = x_i(t) + \beta s_i \sum_j A_{ij}x_j\delta t. \tag{156}$$

- We then have the additional infection equation,

$$\frac{dx_i(t)}{dt} = \beta s_i(t) \sum_j A_{ij}x_j(t). \tag{157}$$

- The system equation is,

$$\frac{dx_i(t)}{dt} = \beta(1 - x_i(t)) \sum_j A_{ij}x_j(t). \tag{158}$$

- At early time we have,

$$\frac{dx_i(t)}{dt} = \beta \sum_j A_{ij}x_j. \tag{159}$$

- As well as the solution basis,

$$\mathbf{A}\mathbf{v}_k = \lambda_k \mathbf{v}_k, \tag{160}$$

$$\mathbf{x}(t) = \sum_k a_k(t)\mathbf{v}_k. \tag{161}$$

  Plugging this solution back in, we can also solve for each coefficient,

$$\frac{da_k(t)}{dt} = \beta\lambda_k a_k(t). \tag{162}$$

$$\mathbf{x}(t) = \sum_k a_k(0)e^{\lambda_k\beta t}\mathbf{v}_k. \tag{163}$$

- The growth rate of infections depends on λ_1 and the probability of infection of nodes depends on v_1 (eigenvector centrality).

10.4 SIS and SIR Models

- Similarly, we can reformulate our SIS and SIR models in terms of probabilistic infection and recovery.

10.5 Epidemic Threshold

- One can show that an epidemic threshold depends on network homogeneity $\sigma_k^2 = \langle k^2 \rangle - \langle k \rangle^2$,

$$R = \frac{\langle k \rangle}{\langle k^2 \rangle}. \tag{164}$$

- In a random network, $\langle k^2 \rangle = \langle k \rangle (\langle k \rangle + 1) : R = 1/\langle k \rangle > 0$.
- In a scale-free network, $P(k) \approx k^{-\gamma}$, when $2 < \gamma \leq 3$ and $N \to \infty$: $\langle k^2 \rangle \to \infty, R \to 0$.