Math 168 – Intro to Networks

University of California, Los Angeles

Duc Vu

Spring 2022

This is math 168 – Introduction to Networks taught by Professor Chodrow. We meet weekly on MWF from 10:00 am to 10:50 am for lecture. The required textbook for the class is *Networks* 2^{nd} by *Newman*. Other course notes can be found at my blog site. Please let me know through my email if you spot any typos in the note.

Contents

1 Lec 1: Mar 28, 2022 **4**
1.1 Introduction 4

2 Lec 2: Mar 30, 2022 **5**
2.1 Networks and Matrices 5

3 Lec 3: Apr 1, 2022 **7**
3.1 Measures and Metrics 7

4 Lec 4: Apr 5, 2022 **9**
4.1 Measures and Metrics (Cont'd) 9

5 Lec 5: Apr 6, 2022 **11**
5.1 Erdos-Renyi Random Graph 11

6 Lec 6: Apr 8, 2022 **13**
6.1 Paths and Branching Processes in ER Random Graphs 13

7 Lec 7: Apr 11, 2022 **14**
7.1 Giant Component in Sparse Erdos-Renyi 14

8 Lec 8: Apr 13, 2022 **15**
8.1 Experimental Lecture on ER Theory 15

9 Lec 9: Apr 15, 2022 **16**
9.1 Configuration Model 16

10 Lec 10: Apr 18, 2022 **18**
10.1 Configuration Model (Cont'd) 18

11 Lec 11: Apr 20, 2022 **19**
11.1 Modularity Maximization 19

12 Lec 12: Apr 22, 2022 **21**
12.1 Modularity Maximization (Cont'd) 21

13 Lec 13: Apr 25, 2022 **23**
13.1 Resolution Limit . . . 23

14 Lec 14: Apr 27, 2022 **24**
14.1 Random Walks on Graphs . . . 24

15 Lec 15: Apr 29, 2022 **26**
15.1 PageRank . . . 26

16 Lec 16: May 2, 2022 **28**
16.1 Agent-Based Modeling . . . 28

17 Lec 17: May 4, 2022 **29**
17.1 Opinion Dynamics . . . 29

18 Lec 18: May 6, 2022 **30**
18.1 Opinion Dynamics (Cont'd) . . . 30

19 Lec 19: May 9, 2022 **31**
19.1 Opinion Model Implementation . . . 31

20 Lec 20: May 11, 2022 **32**
20.1 Midterm . . . 32

21 Lec 21: May 13, 2022 **33**
21.1 Intro to Epidemics on Networks . . . 33

22 Lec 22-23: May 18-20, 2022 **34**
22.1 Link Prediction in Networks . . . 34

List of Theorems

4.7 Laplacian Formula for Cuts 10

List of Definitions

2.1 Graph 5
2.2 Adjacency Matrix 5
2.4 Walk 5
3.1 Degree 7
3.2 Path-connected 7
3.3 Geodesic Path 7
3.4 Diameter 8
4.1 Triadic Closure 9
4.2 Local Clustering Coefficient 9
4.4 Laplacian Matrix 9
4.5 Clustering/Partition 10
4.6 Cut Value 10
5.1 Random Graph 11
5.4 Cycle 12
6.1 Path 13
6.2 Branching Process 13
7.1 Giant Component 14
9.1 Degree Sequence 16
9.2 Configuration Model Random Graph 16
11.1 Modularity 19
11.2 Modularity Matrix 20
14.1 Simple Random Walk 24
14.2 Transition Matrix 24
14.3 Stationary Distribution 24
14.4 Ergodic Graph 24
15.1 Directed Adjacency Matrix 26
15.2 (Directed) Degree 26
15.4 PageRank 26

§1 Lec 1: Mar 28, 2022

§1.1 Introduction

Some introduction and logistics stuffs of the class. Nothing mathy is discussed in this lecture.

§2 Lec 2: Mar 30, 2022

§2.1 Networks and Matrices

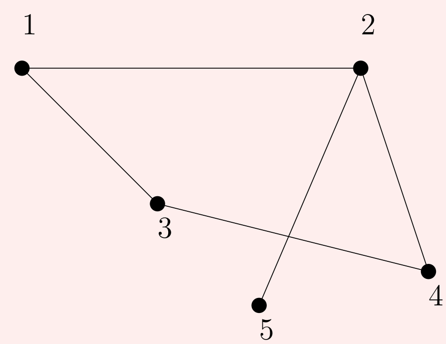
Definition 2.1 (Graph) — A (simple, undirected) graph is $G = (N, E)$, a node set N and an edge set $E \subseteq N \times N$ s.t. $i \neq j \forall (i,j) \in E$.

Definition 2.2 (Adjacency Matrix) — The adjacency matrix $\mathbf{A}$ of a graph $G = (N, E)$ is a matrix in $\mathbb{R}^{n \times n}$ where $n = |N|$ with entries

$$a_{ij} = \begin{cases} 1, & \text{if } (i,j) \in E \\ 0, & \text{otherwise} \end{cases}$$

Example 2.3

Consider the following graph



The adjacency matrix is

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

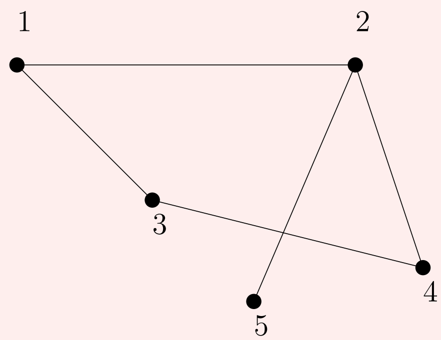
Definition 2.4 (Walk) — A walk in $G = (N, E)$ is a sequence of edges $(i_1, j_1), (i_2, j_2), \ldots, (i_k, j_k)$ where

$$j_1 = i_2, j_2 = i_3, \ldots, j_{k-1} = i_k$$

This is a walk of length k (number of edges) from i_1 to j_k.

Example 2.5

Consider the above example



Walk $3 \to 2$ of length

- $1 : \emptyset$
- $2 : (3,1),(1,2);(3,4),(4,2)$
- $3 : \emptyset$
- $4 : (3,1),(1,2),(2,1),(1,2)$

Fact 2.1. The $ijth$ entry of $\mathbf{A}$ counts the number of walks of length 1 from node i to j.

Conjecture 2.1. The $ijth$ entry of $\mathbf{A}^k$ counts the number of walks of length k from i to j.

Proof. Suppose inductively that $W(k) \triangleq \mathbf{A}^k$ has entries $w_{ij}(k)$ counting k-walks from $i \to j$. Consider $W(k+1) = W(k)A$. Its entries are

$$w_{ij}(k+1) = \sum_{l \in N} w_{il}(k) a_{lj}$$

$\square$

§3 Lec 3: Apr 1, 2022

§3.1 Measures and Metrics

A walk of length 2, $i \leftrightarrow i$, is the number of edges attached to node $i \triangleq$ degree of node i, k_i

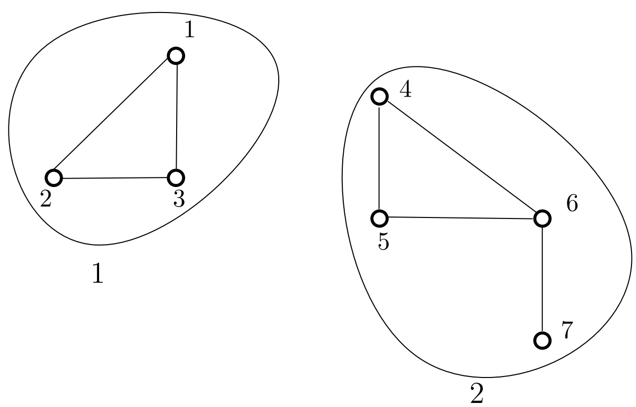
$$k_i = \sum_{j\in N} a_{ij} = \sum_{j\in N} a_{ji} = ii\text{th entry of } \mathbf{A}^2$$

Definition 3.1 (Degree) — The degree k_i of a node i is the number of edges attached to it

$$k_i = |\{j : (i,j) \in E\}|$$

Definition 3.2 (Path-connected) — Nodes i and j are path-connected if $\exists$ a walk $i \leftrightarrow j$ of any length. The connected component of i is the set of notes to which i is path-connected.
G is connected if it has 1 connected component.

Consider a disconnected graph G



Then, the adjacency graph is

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 & 0 \\ 0 & \mathbf{A}_2 \end{bmatrix} \text{ up to permutations of node labels}$$

Question 3.1. How big is a graph?

- Number of nodes
- Number of edges
- Diameter

Definition 3.3 (Geodesic Path) — Geodesic (shortest) path between $i \leftrightarrow j$ is a walk s.t. no walk has shorter length.

Definition 3.4 (Diameter) — Diameter of G is

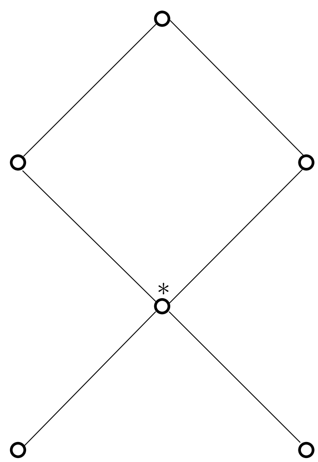
$$\max_{ij} \text{geodesic distance}(i, j)$$

which is undefined if i and j are not connected.

Example 3.5

"6 degrees of separation": in social networks, the diameter is usually about 6.

Node Importance:

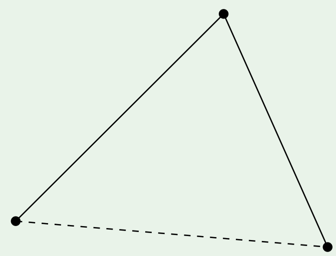


- * has highest degree
- If * were removed, graph would be disconnected
- Short average distance to other nodes
- Betweeness: # of geodesic paths passing through $i \in N$.

§4 Lec 4: Apr 5, 2022

§4.1 Measures and Metrics (Cont'd)

Definition 4.1 (Triadic Closure) — In networks, the observation that this phenomenon



happens a lot is called triadic closure.

Calculation: # of triangles attached to node i = # of walks of length 3 $i \leftrightarrow i$ (A^3 diagonal) $\cdot \frac{1}{2} = \frac{1}{2} \sum_j \sum_k a_{ij} a_{jk} a_{ki}$.
To compute the # of possible triangles attached to i

1. Calculate k_i
2. $\binom{k_i}{2}$

   Exercise 4.1. Express in terms of the adjacency matrix $\mathbf{A}$.

Definition 4.2 (Local Clustering Coefficient) — The local clustering coefficient CC_i at node i is

$$\frac{\text{\# of triangles at } i}{\text{\# of possible triangles}}$$

Note: $0 \leq CC_i \leq 1$.

Remark 4.3. On average, CC_i is high (many triangles can be observed) and global measures are high.

Definition 4.4 (Laplacian Matrix) — The (combinatorial) Laplacian matrix of a graph $\mathbf{L} \in \mathbb{R}^{n \times n}$

$$\mathbf{L} = \mathbf{D} - \mathbf{A}$$

where

$$\mathbf{D} = \begin{bmatrix} k_1 & & & \\ & k_2 & & \\ & & \ddots & \\ & & & k_n \end{bmatrix}$$

Definition 4.5 (Clustering/Partition) — A clustering/partition of a graph is a partition of N, $\{C_1, C_2, \dots, C_l\}$

$$N = \bigcup_{j=1}^{l} C_j, \qquad C_j \cap C_{j'} = \emptyset \text{ if } j \neq j'$$

Let $c_i \triangleq$ cluster of node i.

Definition 4.6 (Cut Value) — The cut value of a partition $\{C_1, \dots, C_l\}$ is

$$\frac{1}{2} \sum_{i,j \in N} a_{ij} \underbrace{\mathbb{1}\left[c_i \neq c_j\right]}_{\substack{=1,\ c_i \neq c_j \\ =0 \text{ otherwise}}}$$

Idea: Good clustering have small cut values.
Setting: 2 clusters $\{C_1, C_2\}$

$$s \in \mathbb{R}^n, \quad s_i = \begin{cases} +1, & c_i = 1 \\ -1, & c_i = 2 \end{cases}$$

Theorem 4.7 (Laplacian Formula for Cuts)
The cut value of $\{C_1, C_2\}$ is $\frac{1}{4} s^\top \mathbf{L} s$.

Proof. Consider

$$\begin{aligned} x^\top \mathbf{L} x &= x^\top (\mathbf{D} - \mathbf{A}) x \\ &= \sum_{i \in N} k_i x_i^2 - \sum_{i,j \in N} a_{ij} x_i x_j \\ &= \sum_{i,j \in N} a_{ij} x_i^2 - \sum_{i,j \in N} a_{ij} x_i x_j \\ &= \frac{1}{2} \left(\sum_{i,j \in N} a_{ij} x_i^2 + \sum_{i,j \in N} a_{ij} x_j^2 - 2 \sum_{i,j \in N} a_{ij} x_i x_j \right) \\ &= \frac{1}{2} \sum_{i,j \in N} a_{ij} (x_i - x_j)^2 \end{aligned}$$

So

$$\begin{aligned} s^\top \mathbf{L} s &= \frac{1}{2} \sum_{i,j \in N} a_{ij} (s_i - s_j)^2 \\ &= \frac{1}{2} \sum_{i,j \in N} a_{ij} \left(4 \mathbb{1}\left[c_i \neq c_j\right]\right) \end{aligned}$$

□

§5 Lec 5: Apr 6, 2022

§5.1 Erdos-Renyi Random Graph

Definition 5.1 (Random Graph) — Random graph is a probability distribution over graphs.

Definition 5.2 — An Erdos-Renyi random graph on n nodes with edges probability p is written $G(n, p)$.

To sample, we take each pair of nodes and draw an edge between them i.i.d with probability p.

Question 5.1. How many pairs are there?

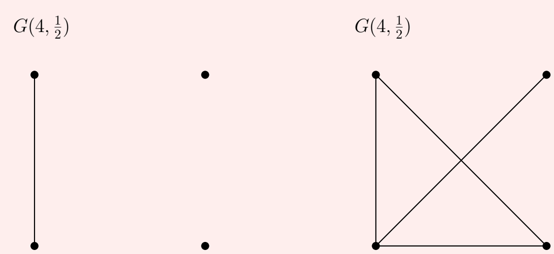
There are $\binom{n}{2}$. Also,

$$\mathbb{E}\left[\#\text{ of edges}\right] = p\binom{n}{2}$$

So $\#$ edges $\sim \text{Binomial}\left(\binom{n}{2}, p\right)$.

Example 5.3

Consider



Question 5.2. What is the average degree c?

We can see that $c = \frac{1}{2}$ for the left graph and $c = 2$ for the right graph.

<u>*Note*</u>: The average degree can be calculate as $c = \frac{2m}{n}$ where m is the number of edges and n is the number of nodes.

<u>Expected # of Triangles in ER</u>

$$\mathbb{E}\left[\#\text{ of triangles}\right] = \binom{n}{3}p^3$$

where each edge is independent. Recall the global clustering coefficient is

$$C = \frac{\#\text{ of triangles} \cdot 3}{\#\text{ of wedges}}$$

where a wedge is a graph with 3 nodes and 2 edges. Then,

$$\mathbb{E}\left[\#\text{ of wedges}\right] = 3\binom{n}{3}p^2$$

Note that

$$\mathbb{E}[C] \neq \frac{\mathbb{E}[\triangle] \cdot 3}{\mathbb{E}[\# \text{ of wedges}]} = p$$

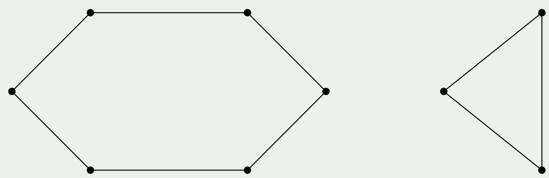
Fact 5.1. As $n \to \infty$, $C \overset{\text{in dist.}}{\longrightarrow} p$.

Node degree in ER

$$\mathbb{E}[k_i] = (n-1)p = c$$

As p increase, $\mathbb{E}[k_i]$ also increases (linearly). However, this is not always the case. Consider sparse ER with $p = \frac{c}{n-1}$. So, as $n \to \infty$, $p = \frac{c}{n-1} \to 0$, i.e., sparse ER model has very little clustering.

Definition 5.4 (Cycle) — A cycle on node i is a walk from $i \leftrightarrow i$ with no repeated nodes or edges.



We can observe that 3-cycles are rare in large sparse ER ($n \to \infty$).

§6 Lec 6: Apr 8, 2022

§6.1 Paths and Branching Processes in ER Random Graphs

Definition 6.1 (Path) — A path is a walk with no node repetitions.

Path lengths: Pick $i, j \in N$. Define $R_k \triangleq \#$ of paths of length k between i and j. Let's compute $r_k = \mathbb{E}[R_k]$.

$$\mathbb{E}[R_k] \approx \mathbb{E}[\#(\text{ path to } l \text{ and } (l,j) \in E)] = \mathbb{E}[\# \text{ of length to } l \text{ of length } k-1]\, p = r_{k-1}p$$

So

$$r_k \approx \underbrace{r_{k-1}p}_{\text{for path through } l} \underbrace{(n-2)}_{\# \text{ of ways to choose } l \neq i,j}$$

Also, notice that

$$r_k \approx r_{k-1}p(n-1) = r_{k-1}c = c^{k-1}r_1 = c^{k-1}p$$

Question 6.1. What length k makes path likely?

We have

$$\log r_k \approx (k-1)\log c + \underbrace{\log p}_{\log c - \log n}$$

$$k \approx \frac{\log r_k + \log n}{\log c}$$

Assume $r_k = 1$. Then, consider the world population of 8 billions with average degree of 1000

$$k \approx \frac{\log n}{\log c} \approx \frac{\log 8 \cdot 10^9}{\log 10^3} \approx 3.4$$

Notice that if $c \leq 1$, the expression above doesn't make any sense.
Galton-Waston Branching Process

Definition 6.2 (Branching Process) — Let p be a probability distribution on $\mathbb{Z}$, called the offspring distribution. A branching process with distribution p is a sequence of random variables $X_0, X_1, X_2, \ldots$ s.t. $X_0 = 1$ and for $t \geq 1$,

$$X_t = \sum_{i=1}^{X_{t-1}} Y_i$$

where each Y_i is distributed i.i.d. according to p.

Branching processes create tree-graphs without cycles, which we can utilize to better understand the behavior of ER random graph.

§7 Lec 7: Apr 11, 2022

§7.1 Giant Component in Sparse Erdos-Renyi

Fact 7.1. Say we have a Poisson(c) process, then

$$\mathbb{E}[X_k] = c^k$$

Total number of individuals in $\mathbb{E}$

$$\mathbb{E}\left[\sum_{k=0}^{\infty} X_k\right] = \sum_{k=0}^{\infty} \mathbb{E}[X_k] = \sum_{k=0}^{\infty} c^k = \begin{cases} \frac{1}{1-c} & 0 < c < 1 \\ \text{divergent "}\infty\text{"} & c \geq 1 \end{cases}$$

Let's consider

$$\begin{aligned} P\left(\frac{\text{size of component containing node } i}{n} > a\right) &= P(\text{size} > an) \\ &\leq \frac{\mathbb{E}[\text{size}]}{an} \qquad \text{(Markov's)} \\ &= \frac{1}{an}\frac{1}{1-c} \to 0 \text{ unless } a = 0 \end{aligned}$$

In the case of $a = 0$, $P\left(\frac{\text{size}}{n} > 0\right) = 1$.

Definition 7.1 (Giant Component) — A sequence $G\left(n, \frac{c}{n-1}\right)$ as $n \to \infty$ has a giant component(GC) if

$$P\left(\frac{\text{component containing random node } i}{n} > a(> 0)\right) \geq b > 0$$

In other words,

$$\mathbb{E}[\text{size of largest component}] = an \text{ for some } 0 < a \leq 1$$

Fact 7.2. Sequence $G\left(n, \frac{c}{n-1}\right)$ has a giant component if and only if $c > 1$.

Let u be the probability P that a node is not in giant component, $s = 1 - u$ is probability that a node is in giant component, $sn =$ size of giant component.

$$u = \left(\underbrace{1-p}_{\text{not connected}} + \underbrace{pu}_{\text{connected not in GC}}\right)^{n-1}$$

Let's simplify the above expression.

$$\begin{aligned} u &= (1 - p(1-u))^{n-1} \\ &= \left(1 - \frac{c(1-u)}{n-1}\right)^{n-1} \\ &= e^{-c(1-u)} \text{ as } n \to \infty \end{aligned}$$

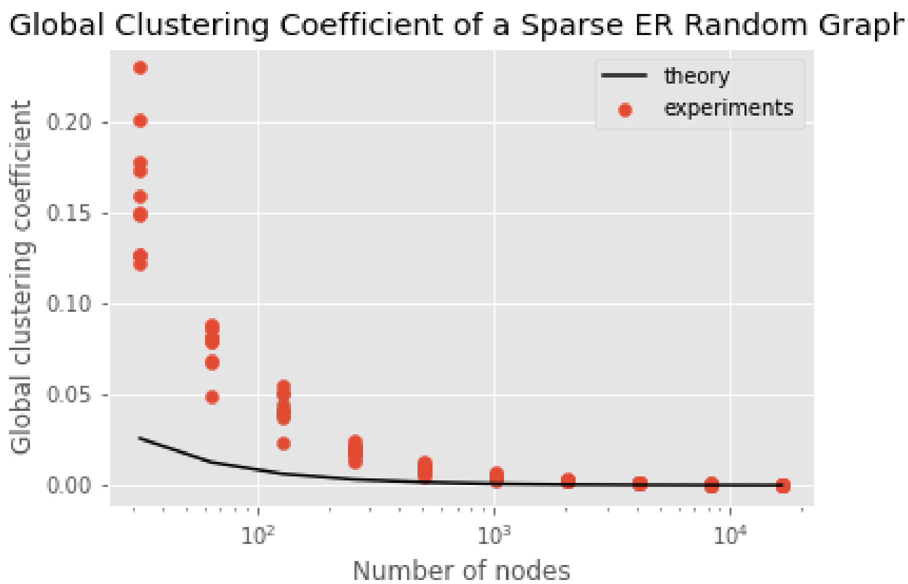
Replace $s = 1 - u$

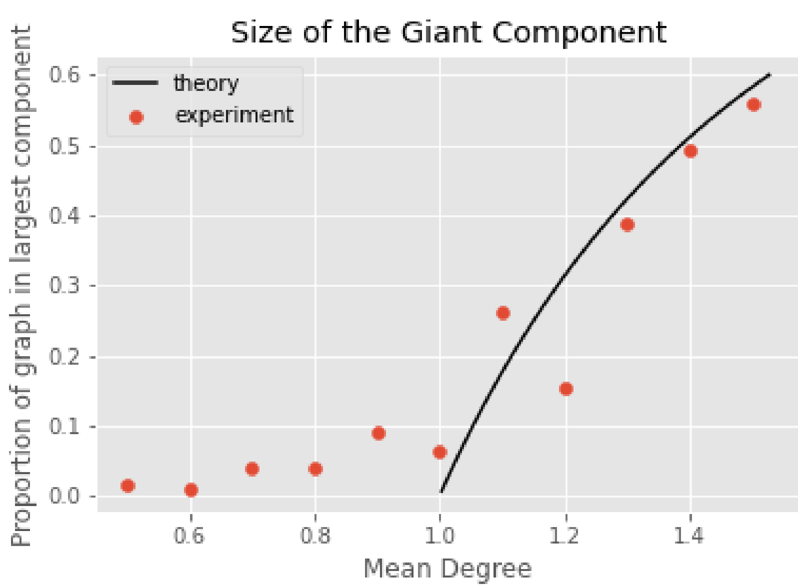
$$s = 1 - e^{-cs}$$

§8 Lec 8: Apr 13, 2022

§8.1 Experimental Lecture on ER Theory

In this lecture, we did some experiments with Python to check whether they agree with the theoretical results that we discussed previously on ER random graph.





§9 Lec 9: Apr 15, 2022

§9.1 Configuration Model

Definition 9.1 (Degree Sequence) — The degree sequence of $G = (N, E)$ with $|N| = n$ is $\vec{k} \in Z^n$ s.t. degree of $i \in N = k_i$.

Definition 9.2 (Configuration Model Random Graph) — The configuration model random graph with degree sequence $\vec{k}$ is a uniformly random graph among all graph with degree sequence $\vec{k}$.

Stub-Matching:

Select uniformly random pairs of half edges and turn them into edges until we run out of edge pair (we then have a graph). However, this method is not perfect as we can have a problem with self-loop or parallel edges, i.e., we only want simple graphs.

Fact 9.1. For $n \to \infty$, if the degree sequence doesn't grow in its entries (sparsity), then $P\left(\text{simple graph}\right) > \varepsilon > 0$.

Fact 9.2. Stub matching (conditioned on getting a simple graph) samples from configuration model.

Moment of the degree sequence:

Definition 9.3 — Degree distribution $p_k = P\left(\text{random node has degree } k\right) = \frac{\#\text{ nodes of degree } k}{n}$.

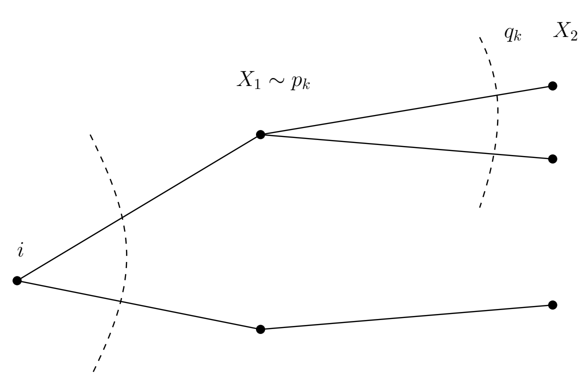
The l^{th} moment is defined as

$$\langle k^l \rangle \triangleq \sum_k p_k k^l$$

So

$$\langle k^0 \rangle = \sum_k p_k = 1$$
$$\langle k^1 \rangle = \sum_k p_k k$$

Branching Process:



$$\mathbb{E}\left[X_1\right] = \langle k \rangle$$

and

$$\begin{aligned}
q_k &= P\left(\text{if I follow an edge to node j, the number of additional edges on j} = k\right)\\
q_k &= \frac{(k+1)p_{k+1}n}{\text{\# of half-edges=2m}}\\
&= \frac{(k+1)p_{k+1}}{\langle k\rangle}
\end{aligned}$$

Then,

$$\begin{aligned}
\mathbb{E}\left[\text{\# offspring in 2nd gen from single parent}\right] &= \sum_{k=0} kq_k = \sum_{k=0} \frac{k(k+1)p_{k+1}}{\langle k\rangle}\\
&= \sum_{k'=1} \frac{(k'-1)k'p_{k'}}{\langle k\rangle}\\
&= \frac{1}{\langle k\rangle}\sum_{k'=1}\left(k'^2 - k'\right)p_{k'}\\
&= \frac{1}{\langle k\rangle}\left(\langle k^2\rangle - \langle k\rangle\right)
\end{aligned}$$

Branching heuristic for giant component: Giant component iff $\frac{\langle k^2\rangle - \langle k\rangle}{\langle k\rangle} > 1$.

§10 Lec 10: Apr 18, 2022

§10.1 Configuration Model (Cont'd)

From Cauchy-Schwarz, $\langle k^2 \rangle \geq \langle k \rangle^2$. In real world social networks, $\langle k^2 \rangle \gg \langle k^2 \rangle$ (heterogeneous degree). For example, say we have vaccine for 1% of population, and we want to vaccinate high degree individuals but in reality we don't know their degree. The problem here is we don't have network data/structure, and our assumption is $\langle k^2 \rangle \gg \langle k \rangle^2$.
Instead of randomly picking people to get vaccine, we encourage people to nominate a friend to get the vaccine (walk along social network).

§11 Lec 11: Apr 20, 2022

§11.1 Modularity Maximization

Definition 11.1 (Modularity) — The modularity of graph G and cluster labels z_i for each node, $\vec{z} \in \mathbb{R}^n$ with respect to random graph model M

$$Q(G, z) = \frac{1}{2m} \sum_{i,j \in N} \left[a_{ij} - \mathbb{E}_m\left[A_{ij}\right]\right] \delta(z_i, z_j)$$

where

$$\delta(x, y) = \begin{cases} 1, & x = y \\ 0, & x \neq y \end{cases}$$

Consider an ER random graph $G(n, p)$

$$\mathbb{E}\left[A_{ij}\right] = P\left(A_{ij} = 1\right) = p$$

Now, consider the configuration model, we know the degrees $k_1, k_2, \ldots, k_n$

$$\sum_{i=1}^{n} k_i = 2m$$

Then,

$$\mathbb{E}\left[A_{ij}\right] = \left(\frac{k_i}{2m} \frac{k_j}{2m - 1}\right) 2m \approx \frac{k_i k_j}{2m}$$

So we can substitute the expression above into the modularity formula

$$Q = \frac{1}{2m} \sum_{i,j} \left(a_{ij} - \frac{k_i k_j}{2m}\right) \delta(z_i, z_j)$$

which is known as the standard modularity.
Now, let's dig into how to maximize the modularity. We need to find cluster/communities in G by solving

$$\hat{z} = \operatorname*{argmax}_z Q(G, z)$$

Assume we have n nodes and 2 groups, then there are 2^n candidate solutions. However, this is NP-hard problem. We must use heuristics. First, let's turn $\delta(z_i, z_j)$ into some expression that involves linear algebra.

$$s_i \triangleq \begin{cases} +1, & z_i = 1 \\ -1, & z_i = 2 \end{cases}$$

$$\begin{aligned} \delta(z_i, z_j) &= \begin{cases} 1, & z_i = z_j \\ 0, & \text{otherwise} \end{cases} \\ &= \frac{1}{2}\left(s_i s_j + 1\right) \end{aligned}$$

So

$$\begin{aligned}
Q &= \frac{1}{2m}\sum_{i,j}\left(a_{ij}-\frac{k_ik_j}{2m}\right)\frac{1}{2}\left(s_is_j+1\right)\\
&= \frac{1}{4m}\sum_{i,j}\underbrace{\left(a_{ij}-\frac{k_ik_j}{2m}\right)}_{b_{ij}}s_is_j + \underbrace{\frac{1}{4m}\sum_{i,j}\left(a_{ij}-\frac{k_ik_j}{2m}\right)}_{=0}\\
&= \frac{1}{4m}\sum_{i,j}b_{ij}s_is_j
\end{aligned}$$

Definition 11.2 (Modularity Matrix) — The modularity matrix $\mathbf{B}\in\mathbb{R}^{n\times n}$ has entries

$$b_{ij} = a_{ij} - \frac{k_ik_j}{2m}$$

This allows to write

$$Q = \frac{1}{4m}\vec{s}^\top\mathbf{B}\vec{s}$$

Let s have any entries, solve $\max_s \vec{s}^\top\mathbf{B}\vec{s}$, set

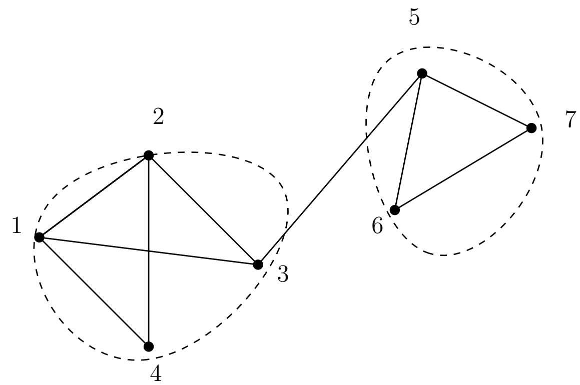
$$z_i = \begin{cases}1, & s_i < 0\\ 2, & s_i \ge 0\end{cases}$$

So from homework 0, we know that $s =$ 1st eigenvector of $\mathbf{B}$.

§12 Lec 12: Apr 22, 2022

§12.1 Modularity Maximization (Cont'd)

Consider a community detection problem



Say we find a label vector $\vec{z} = (1,1,1,1,2,2,2)$. In modularity maximization, our goal is to pick $\vec{z}$ to maximize

$$Q = \frac{1}{2m} \sum_{i,j \in N} \left[a_{ij} - \frac{k_i k_j}{2m} \right] \delta(z_i, z_j)$$

Let l be a label. Let's define

$$e_l \triangleq \frac{1}{2m} \sum_{i,j \in N} a_{ij} \delta(z_i, l) \delta(z_j, l) = \text{\% of all edges w/ both ends in community } l$$

and

$$f_l \triangleq \frac{1}{2m} \sum_{i \in N} k_i \delta(z_i, l) = \text{\% of edges that end in cluster } l$$

So from the above figure, we can see that

$$\begin{aligned} e_1 &= \frac{1 \cdot 10}{2 \cdot 9} = \frac{5}{9} \\ f_2 &= \frac{1 \cdot 7}{18} = \frac{7}{18} \end{aligned}$$

So we can rewrite Q as follows

$$\begin{aligned} Q &= \frac{1}{2m} \sum_{i,j} \left[a_{ij} - \frac{k_i k_j}{2m} \right] \delta(z_i, z_j) \\ &= \frac{1}{2m} \left[\sum_l \sum_{i,j} a_{ij} \delta(z_i, l) \delta(z_j, l) \right] \\ &= \frac{1}{2m} \left[\sum_l \sum_{i,j} \frac{k_i \delta(z_i, l) k_j \delta(z_j, l)}{2m} \right] \\ &= \sum_l \left[e_l - f_l^2 \right] \end{aligned}$$

Consider $\max -\sum_l f_l^2$ or $\min \sum_l f_l^2$ s.t. $\sum_l f_l = 1$ and $f_l \geq 0$. Then, by using Lagrange multiplier, we have

$$\begin{aligned} \nabla \sum_l f_l^2 &= \lambda \nabla \sum_l f_l \\ 2 \begin{pmatrix} f_1 \\ f_2 \\ \vdots \end{pmatrix} &= \lambda \begin{pmatrix} 1 \\ 1 \\ \vdots \end{pmatrix} \\ \implies f_l &= f_{l'} \quad \text{for } l \neq l' \end{aligned}$$

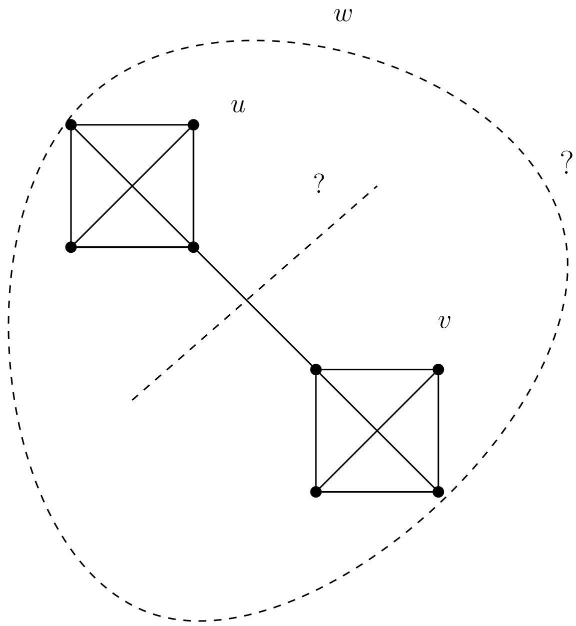
Thus, modularity maximization says

1. Try to make lots of in-cluster edge ($\max e_l$)
2. Try to make the cluster similar sizes (f_l)

§13 Lec 13: Apr 25, 2022

§13.1 Resolution Limit

Modularity maximization can't find communities that are "too small" relative to graph size.



Let ΔQ be the change in Q due to merging u and v into w.

$$\Delta Q = \underbrace{e_w - (e_u + e_v)}_{\frac{1}{2m}} - \left(f_w^2 - \left(f_u^2 + f_v^2\right)\right)$$

Note that

$$f_u = \frac{1}{2m}\,(\text{sum of node degrees in cluster } u) = \frac{1}{2m}\left[(k-1)k+1\right] = f_v \triangleq \frac{s}{2m}$$

and

$$f_w = \frac{1}{2m}2s\,(= f_u + f_v)$$

So when is $\Delta Q > 0$?

$$\frac{1}{2m} - \frac{(2s)^2}{(2m)^2} + 2\frac{s^2}{(2m)^2} > 0$$
$$2m > s^2$$

Example 13.1

Consider a graph with $n = 5 \times 10^6$ and $c = 20$

$$\implies 2m = nc = 10^8$$

We need $s > \sqrt{2m} = 10^4$ where $s = k^2 - k + 1$. So roughly speaking, $k > 100$ to detect k-clique communities in this graph.

§14 Lec 14: Apr 27, 2022

§14.1 Random Walks on Graphs

Simple Random Walk:

Start at node i. Pick a neighbor of i uniformly at random and move there. Repeat this process infinitely.

Definition 14.1 (Simple Random Walk) — A simple random walk on graph G is a countable sequence of random variables $X_1, \ldots, X_t, \ldots$ with values in N ($X_t = i$ implies we are at node i at time t). The distribution of X_{t+1}

$$\begin{aligned} P\left(X_{t+1}=i \mid X_t=j_t, X_{t-1}=j_{t-1}, \ldots, X_0=j_0\right)=P\left(X_{t+1}=i \mid X_t=j_t\right) &= \begin{cases} \frac{1}{k_{j_t}} & (j_t, i) \in E \\ 0 & (j_t, i) \notin E \end{cases} \\ &= \frac{a_{ij_t}}{k_{j_t}} \end{aligned}$$

Definition 14.2 (Transition Matrix) — The transition matrix of a simple random walk is $\mathbf{P} = \mathbf{A}\mathbf{K}^{-1}$ where

$$\mathbf{K} = \begin{pmatrix} k_1 & & \\ & \ddots & \\ & & k_n \end{pmatrix}$$

and $p_{ij} = \frac{a_{ij}}{k_j}$.

Consider

$$\begin{aligned} P\left(X_{t+1}=i\right) &= \sum_{j \in N} P\left(X_{t+1}=i \mid X_t=j\right) P\left(X_t=j\right) \\ q_i(t+1) &= \sum_{j \in N} p_{ij} q_j(t) \end{aligned}$$

So

$$\vec{q}(t+1) = \mathbf{P}\vec{q}(t) = \mathbf{P}^{t+1} q(0)$$

Definition 14.3 (Stationary Distribution) — A simple random walk has a stationary distribution $\vec{\pi} \in \mathbb{R}^n$ if $\lim_{t\to\infty} \vec{q}_i(t) = \vec{\pi}_i$, regardless of the starting point.

Definition 14.4 (Ergodic Graph) — A graph is ergodic if

1. it is connected and
2. it is aperiodic (gcd of cycle length = 1)

Theorem 14.5

A simple random walk on an ergodic graph has a unique stationary distribution $\vec{\pi}$. Furthermore, $\vec{\pi}$ is the unique solution of $\vec{\pi} = \mathbf{P}\vec{\pi}$.

Proof. Use Perron-Frobenius Theorem. □

Structure of $\vec{\pi}$: Recall $\mathbf{P} = \mathbf{A}\mathbf{K}^{-1}$. We want to show

$$\vec{\pi} = \mathbf{A}\mathbf{K}^{-1}\vec{\pi}$$

Guess: $\vec{\rho} = \vec{k}$. Then, let's check.

$$\mathbf{A}\mathbf{K}^{-1}\vec{\rho} = \mathbf{A}\mathbf{K}^{-1}\vec{k} = \mathbf{A}\mathbf{1} = \vec{k} = \vec{\rho}$$

This is not normalized, so we can deduce that $\vec{\pi} = \frac{1}{2m}\vec{k}$.

§15 Lec 15: Apr 29, 2022

§15.1 PageRank

Definition 15.1 (Directed Adjacency Matrix) — Directed adjacency matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$

$$a_{ij} = \begin{cases} 1 & \text{if } j \to i \\ 0 & \text{otherwise} \end{cases}$$

Consequently, $\mathbf{A}$ is, in general, not symmetric.

Definition 15.2 ((Directed) Degree) — We define

$$k_i^{\text{in}} = \sum_{j\in N} a_{ij}$$
$$k_i^{\text{out}} = \sum_{j\in N} a_{ji}$$

Directed Random Walk: From node j, follow a random outgoing arrow to the next node

$$P\left(X_{t+1} = i | X_t = j\right) = \frac{a_{ij}}{k_j^{\text{out}}} \triangleq p_{ij}$$

Then, we define the transition matrix as follows

$$\mathbf{P} = \mathbf{A}(\mathbf{K}^{\text{out}})^{-1}$$

where

$$\mathbf{K}^{\text{out}} = \begin{bmatrix} k_1^{\text{out}} & & \\ & \ddots & \\ & & k_n^{\text{out}} \end{bmatrix}$$

in which we assume $k_i^{\text{out}} \geq 1\ \forall i$.

Theorem 15.3

Suppose that there exists integer $t > 0$ s.t. $\mathbf{A}^t$ has all positive entries. Then the directed random walks has a stationary distribution $\vec{\pi}$, and $\mathbf{P}\vec{\pi} = \vec{\pi}$.

Definition 15.4 (PageRank) — With probability $1-\alpha$, take a directed random walk step with probability α, teleport somewhere else. Note that

- $\alpha \in [0, 1]$ is the teleportation rate
- $\vec{v} \in \mathbb{R}^n$ is the teleportation vector (assume $\vec{v}$ is entry-wise positive, $\sum v_i = 1$)

This walk has the probability transition

$$P\left(X_{t+1} = i | X_t = j\right) = (1-\alpha)\frac{a_{ij}}{k_j^{\text{out}}} + \alpha v_i = \tilde{p}_{ij}$$

We define

$$\tilde{\mathbf{P}} = (1-\alpha)\mathbf{P} + \alpha\mathbf{V}$$

Traditionally, $\alpha = 0.15$ and $\vec{v} = \frac{1}{n}\begin{pmatrix}1 & 1 & \dots & 1\end{pmatrix}^\top$. If we choose this particular choice, PakeRank has a stationary distribution. Notice that $\pi_i > 0$ where $\vec{\pi} = \tilde{\mathbf{P}}\vec{\pi}$.

§16 Lec 16: May 2, 2022

§16.1 Agent-Based Modeling

Coding session :)

§17 Lec 17: May 4, 2022

§17.1 Opinion Dynamics

figure here

Each node i has opinion $x_i \in [-1,1]$. For example,

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} 0.5 \\ -0.7 \\ 0.2 \\ -0.3 \\ 0.0 \end{pmatrix} = \vec{x}(t)$$

Then,

$$\vec{x}(t+1) = F\left(\vec{x}(t)\right)$$

This is discrete-time deterministic (not random) synchronous (all nodes update simultaneously) model, and our function F depends on the graph structure.

$$F\left(\vec{x}(t)\right) = \begin{pmatrix} f_1\left(\vec{x}(t)\right) \\ f_2\left(\vec{x}(t)\right) \\ \vdots \\ f_n\left(\vec{x}(t)\right) \end{pmatrix}$$

Note that $f_i\left(\vec{x}(t)\right)$ is a function only of the neighbors of node i and i itself.

$$x_i(t+1) = f_i\left(\vec{x}(t)\right) = (1-\beta)x_i(t) + \beta \underbrace{\frac{1}{k_i}\sum_{j\sim i} x_j(t)}_{\text{average of neighbor opinion}}$$

Notice that $\sum_{j\sim i} x_j(t) = \sum_{j\in N} a_{ij}x_j(t) = \left(\mathbf{A}\vec{x}\right)_i$ and $(1-\beta)x_i(t) = \left((1-\beta)\mathbf{I}\vec{x}\right)_i$. Then,

$$\vec{x}(t+1) = \left[(1-\beta)\mathbf{I} + \beta\left(\mathbf{K}^{-1}\mathbf{A}\right)\right]\vec{x}$$

§18 Lec 18: May 6, 2022

§18.1 Opinion Dynamics (Cont'd)

Recall

$$\vec{x}(t+1) = (1-\beta)\left[\mathbf{I}\vec{x}(t)\right] + \beta\left[\mathbf{K}^{-1}\mathbf{A}\vec{x}(t)\right]$$

Then, we have

$$\begin{aligned}\vec{x}(t+1) - \vec{x}(t) &= \beta\left(\mathbf{K}^{-1}\mathbf{A} - \mathbf{I}\right)\vec{x}(t)\\ &= \beta\mathbf{K}^{-1}\left(\mathbf{A} - \mathbf{K}\right)\vec{x}(t)\\ &= \beta\mathbf{K}^{-1}(-\mathbf{L})\vec{x}(t)\\ &= -\beta\overline{\mathbf{L}}\vec{x}(t)\end{aligned}$$

Fact 18.1. $\vec{x}(t) \overset{t\to\infty}{\longrightarrow} \vec{x}^*$ where $\overline{\mathbf{L}}\vec{x}^* = \vec{0}$ and $\vec{x}^*$ is unique if G is connected.

Suppose $\overline{\mathbf{L}}\vec{x} = \vec{0}$, and in particular

$$\begin{aligned}\mathbf{K}^{-1}(\mathbf{K}-\mathbf{A})\vec{x} &= \vec{0}\\ (\mathbf{K}-\mathbf{A})\vec{x} &= \vec{0}\end{aligned}$$

So $\mathbf{L}\vec{x}^* = \vec{0}$. Thus,

$$\vec{x}^* = \gamma\vec{1}$$

Fact 18.2. On connected graphs, linear consensus dynamics converges to consensus, i.e., $x_i^* = x_j^*$ for all $i, j \in N$.

Model Modifications

- β depends on node where some nodes have $\beta_i = 0$
- Interaction depends on x_i and x_j (Hegselmann-Krause)
- Introduce noise

§19 Lec 19: May 9, 2022

§19.1 Opinion Model Implementation

Coding session – refer to this link!

§20 Lec 20: May 11, 2022

§20.1 Midterm

§21 Lec 21: May 13, 2022

§21.1 Intro to Epidemics on Networks

Coding session :D – refer to this link!

§22 Lec 22-23: May 18-20, 2022

§22.1 Link Prediction in Networks

Coding Session :) – refer to this link!