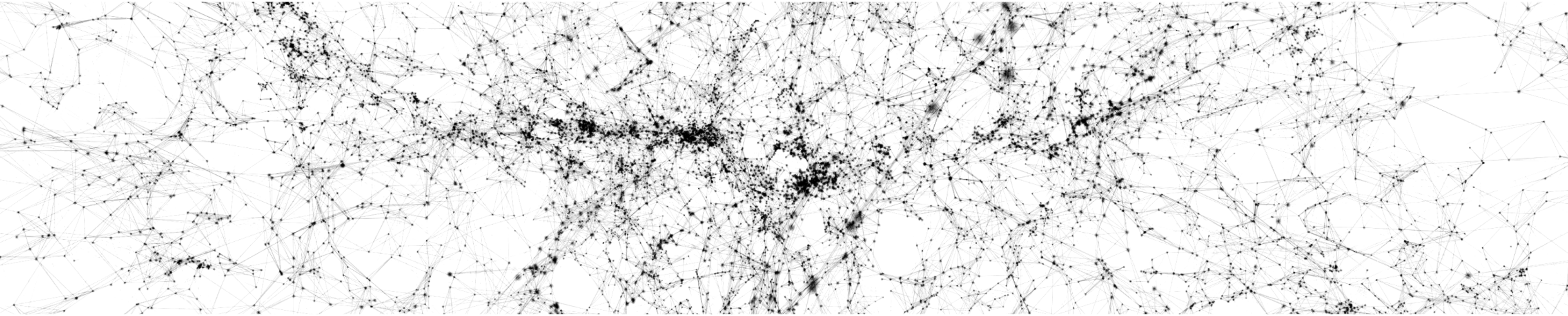
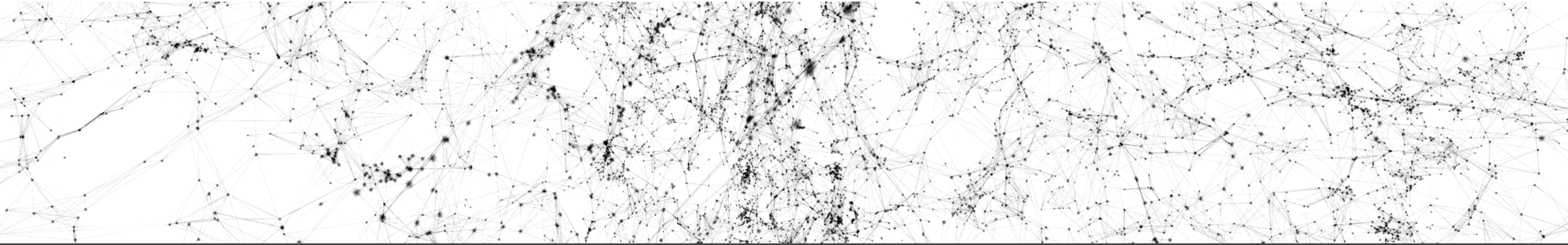


Models

Analysis of complex interconnected data

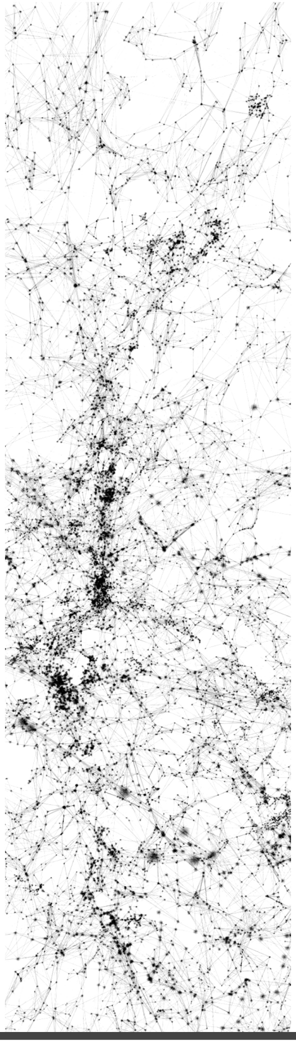


Quick Notes

- Reminder, first assignment due in a week
  - http://www.reirab.com/Teaching/NS22/Assignment_1.pdf
  - **Any questions for the assignment?**
  - Submit single entry as a Group in Mycourses
  - On the report, make sure it is well-written
    - Plots have legends, axes are marked clearly, datasets explained
    - Explain what you have done, reference each (set of) plot(s) in text
- Use Ed for easier communications

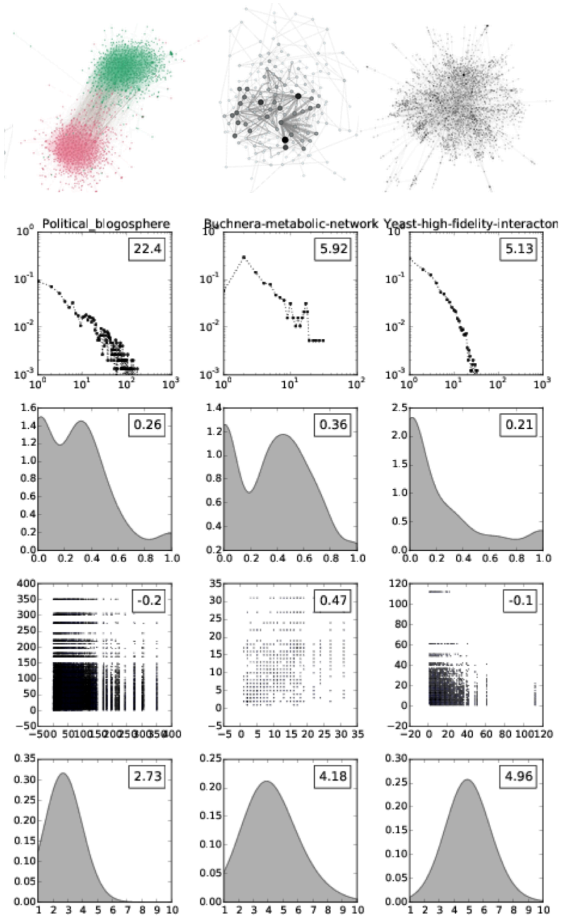
Outline

- **Patterns Quick recap**
- Models
  - ER model
  - BA model
  - SBM
  - Configuration model
  - FF model
  - Kronecker graph model
  - Fitting to observed graphs
  - LFR model



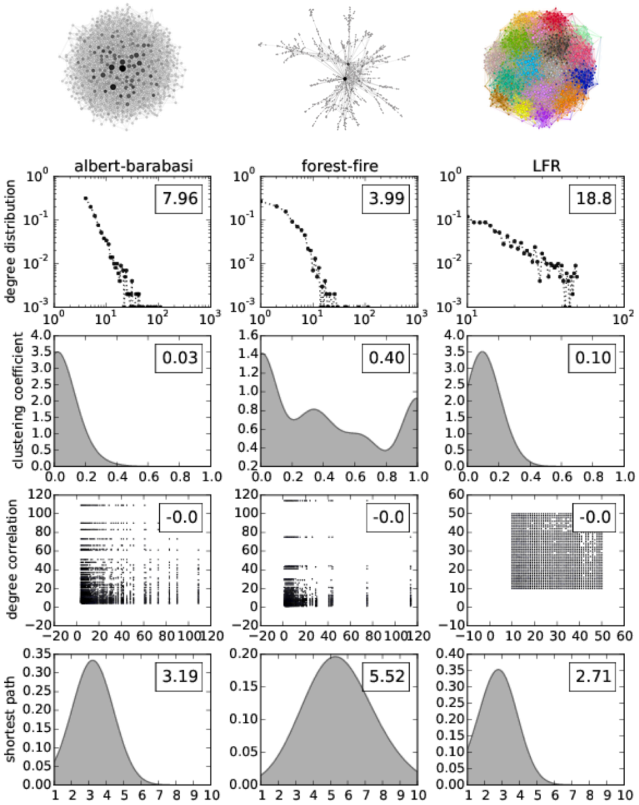
Patterns: quick recap

- Sparsity Pattern
    - mean degree << number of nodes (E << Emax)
- Scale Free Pattern
    - heavy tailed degree distribution
- Assortativity Pattern
    - positive or negative correlation between degree of connecting nodes
- Transitivity Pattern
    - high ratio of closed triangles (clustering coefficient)
- Small world Pattern
    - small average shortest path



Outline

- Patterns Quick recap
- Models
    - **ER model**
    - SBM
    - Configuration model
    - AB model
    - FF model
    - Kronecker graph model
    - Fitting to observed graphs
    - LFR model

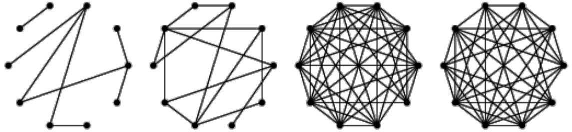


Erdös-Rényi Model (ER)

- Introduced in 1960
- Basis of **random graph** theory
- Simple model that results in **small-world** graphs
- Parameters: $\mathscr{G}(n, p)$ or $\mathscr{G}(n, m)$
  - n: number of nodes
  - p: probability of an edge between any two nodes
  - m: number of edges
- Generation: How can we generate an ER graph?

all edges are equally likely

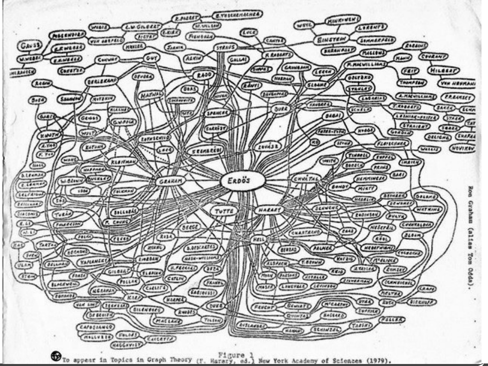
ER(n, p)



Paul Erdős
(1913-1996)

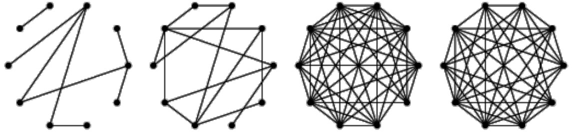
Alfréd Rényi
(1921-1970)

Side note:
What is Erdős number?



Erdös-Rényi Model (ER)

- Introduced in 1960
- Basis of **random graph** theory
- Simple model that results in **small-world** graphs
- Parameters: $\mathscr{G}(n, p)$ or $\mathscr{G}(n, m)$
  - n: number of nodes
  - p: probability of an edge between any two nodes
  - m: number of edges
- Generation: How can we generate an ER graph?
  - $\mathscr{G}(n, p)$: for each pair of node connect them with probability p ($\mathcal{O}(n^2)$): toss M (n choose 2) coins {has linear time implementation}
  - $\mathscr{G}(n, m)$: for each edge, select a random source and destination($\mathcal{O}(m)$): roll 2m n-sided die



$$N = 10, \quad M = \binom{10}{2} = 45$$

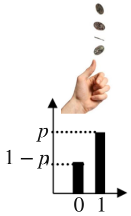
What is p here?

Erdös-Rényi Model (ER): Binomial Graphs

- Generation: How can we generate an ER graph?
  - ○ $\mathscr{G}(n, p)$: toss M (n choose 2) biased coins (with success probability p)

- ER Graphs are also called **Binomial Graphs**
  - ○ A coin's outcome has a Bernoulli distribution, x is a Bernoulli random variable that takes values of 0 or 1 with:

$$Bernoulli(x|p) = p^x(1-p)^{(1-x)} \quad \text{or} \quad Bernoulli(x|p) = \begin{cases} p & x = 1 \\ 1-p & x = 0 \end{cases}$$



  - ○ Number of heads in a sequence of independent coin tosses follows a Binomial distribution

$$\underbrace{Binomial(M, m|p)}_{\text{Probability of generating a graph with m edges}} = \underbrace{\binom{M}{m}}_{\text{Select m edges out of M possible}} \underbrace{p^m}_{\text{Probability of having m links}} \underbrace{(1-p)^{M-m}}_{\text{Probability of not having the rest of links}}$$

Erdös-Rényi Model (ER): Degree Distribution

- ER Graphs are also called **Binomial Graphs**
  - Probability of an edge:

$$Bernoulli(x|p) = p^x(1-p)^{(1-x)}$$

  - Probability of generating a graph with m edges:

$$Binomial(M, m|p) = \binom{M}{m} p^m (1-p)^{M-m}$$

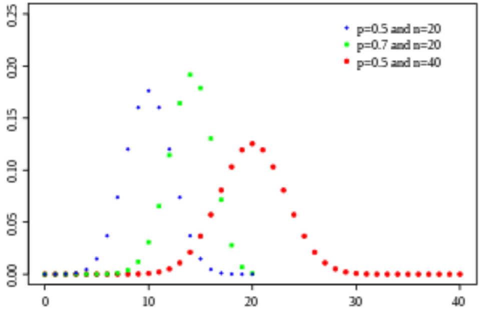
  - Degree distribution:

$$p(k) = Binomial(n-1, k|p) = \binom{n-1}{k} p^k (1-p)^{n-1-k}$$

Select k neighbours out of n-1 possible nodes

Probability of having k links

Probability of not having the rest of links



Erdös-Rényi Model (ER): Degree Distribution

- Degree distribution:

$$p(k) = Binomial(n-1, k \,|\, p) = \binom{n-1}{k} p^k (1-p)^{n-1-k}$$

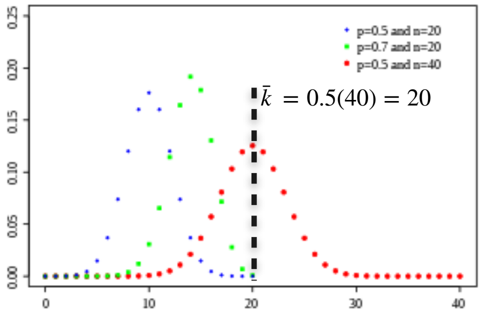
Select k neighbours out of n-1 possible nodes

Probability of having k links

Probability of not having the rest of links

We know the mean and variance of a Binomial distribution, so we easily get:

  - ○ Mean Degree: $p(n-1)$
  - ○ Variance of Degree: $p(1-p)(n-1)$



Erdös-Rényi Model (ER): Degree Distribution

- Degree distribution:

$$p(k) = Binomial(n-1, k\,|\,p) = \binom{n-1}{k} p^{k}(1-p)^{n-1-k}$$

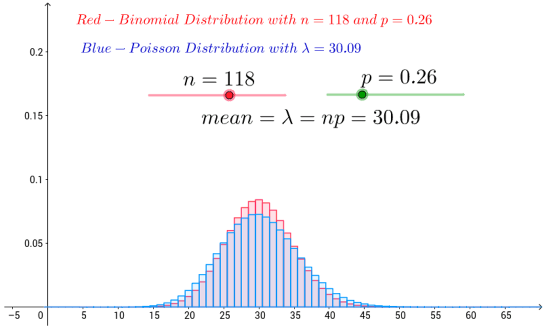
Select k neighbours out of n-1 possible nodes

Probability of having k links

Probability of not having the rest of links

- For large *n* and small *k*, which is often the case in real world graphs, we can **approximate** this with Poisson distribution with mean of average degree

$$p(k) = e^{-\bar{k}}\frac{\bar{k}^{k}}{k!}$$



- ER graphs are therefore also sometimes called **Poisson random graphs**

Erdös-Rényi Model (ER): Clustering Coefficient

- Local clustering coefficient: $c_i = \dfrac{A^3_{ii}}{k_i(k_i-1)} = \dfrac{2\mathscr{E}_i}{k_i(k_i-1)}$

where $\mathscr{E}_i$: number of edges between neighbours of i

- Expected number of edges between i's neighbours, given since edges are i.i.d and equally likely:

$$E[\mathscr{E}_i] = p\frac{k_i(k_i-1)}{2}$$

Probability of an edge between a pair

Number of distinct pairs of neighbours of i

- Expected clustering coefficient becomes:

$$E[c_i] = p\frac{k_i(k_i-1)}{k_i(k_i-1)} = p = \frac{\bar{k}}{n-1}$$

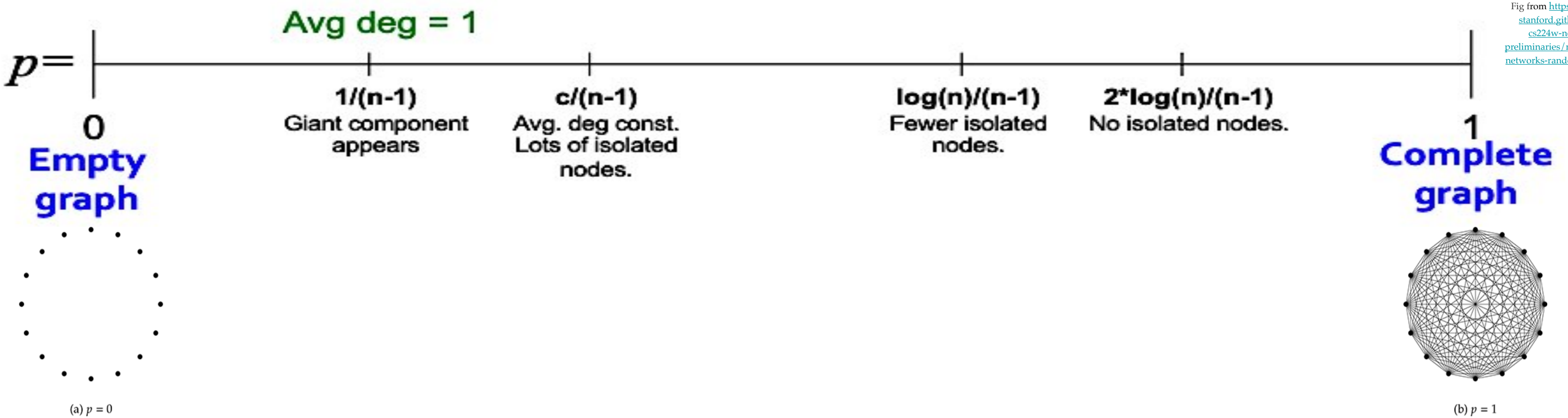
- Small [Zero] clustering coefficient
  - The clustering coefficient is average degree divided by number of nodes therefore with fixed average degree, and when n grows, clustering coefficient goes to zero

Erdös-Rényi Model (ER): Connectivity

Emergence of a giant component at $p = \dfrac{1}{n-1}$ that is when $\bar{k} = 1$

A network component whose size grows in proportion to n we call a giant component.

In expectation, every node has one edge



Avg deg = 1

$p=$

0 **Empty graph**

1/(n-1) Giant component appears

c/(n-1) Avg. deg const. Lots of isolated nodes.

log(n)/(n-1) Fewer isolated nodes.

2*log(n)/(n-1) No isolated nodes.

1 **Complete graph**

(a) $p = 0$

(b) $p = 1$

Fig from https://snap-stanford.github.io/cs224w-notes/preliminaries/measuring-networks-random-graphs

$\bar{k} = 0$

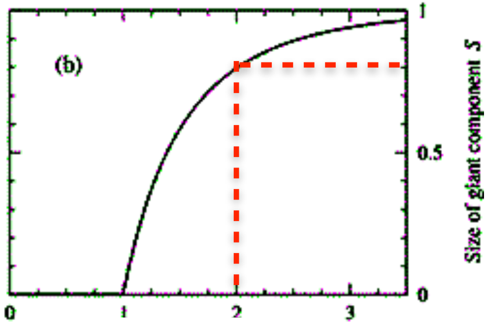
$\bar{k} = n - 1$

Erdös-Rényi Model (ER): Connectivity

Emergence of a giant component at $p = \dfrac{1}{n-1}$ that is when $\bar{k} = 1$

A network component whose size grows in proportion to n we call a giant component.

In expectation, every node has one edge



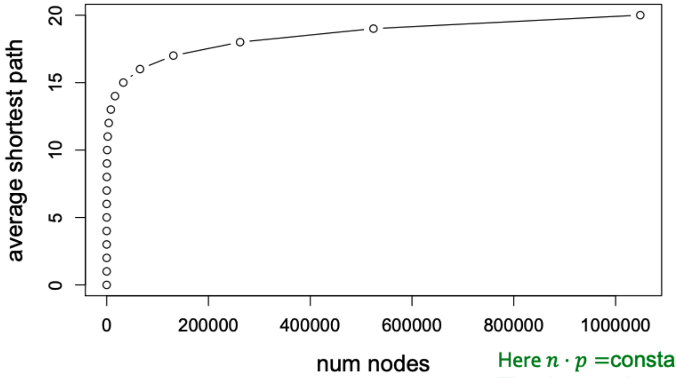
$\bar{k} = 1$

- With average degree of 2, 80% of nodes are in the GCC
- in the limit of large n, the probability that we will have two separate giant components in such a network goes to zero

Erdös-Rényi Model (ER): path length

- ER graphs are Small world
    - The diameter is $\log(n)/\log(pn)$

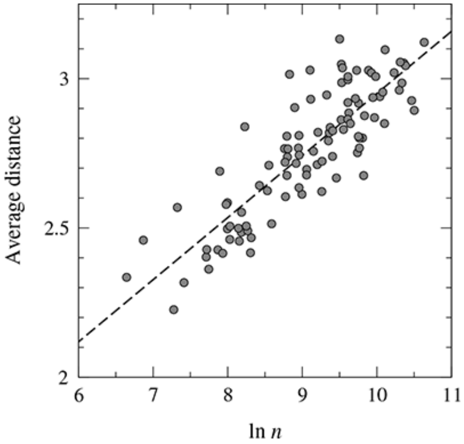
- Example: we increase the number of nodes, while keeping the average degree constant, average shortest path increase is logarithmic, that is in order of $\mathcal{O}(\log(n))$



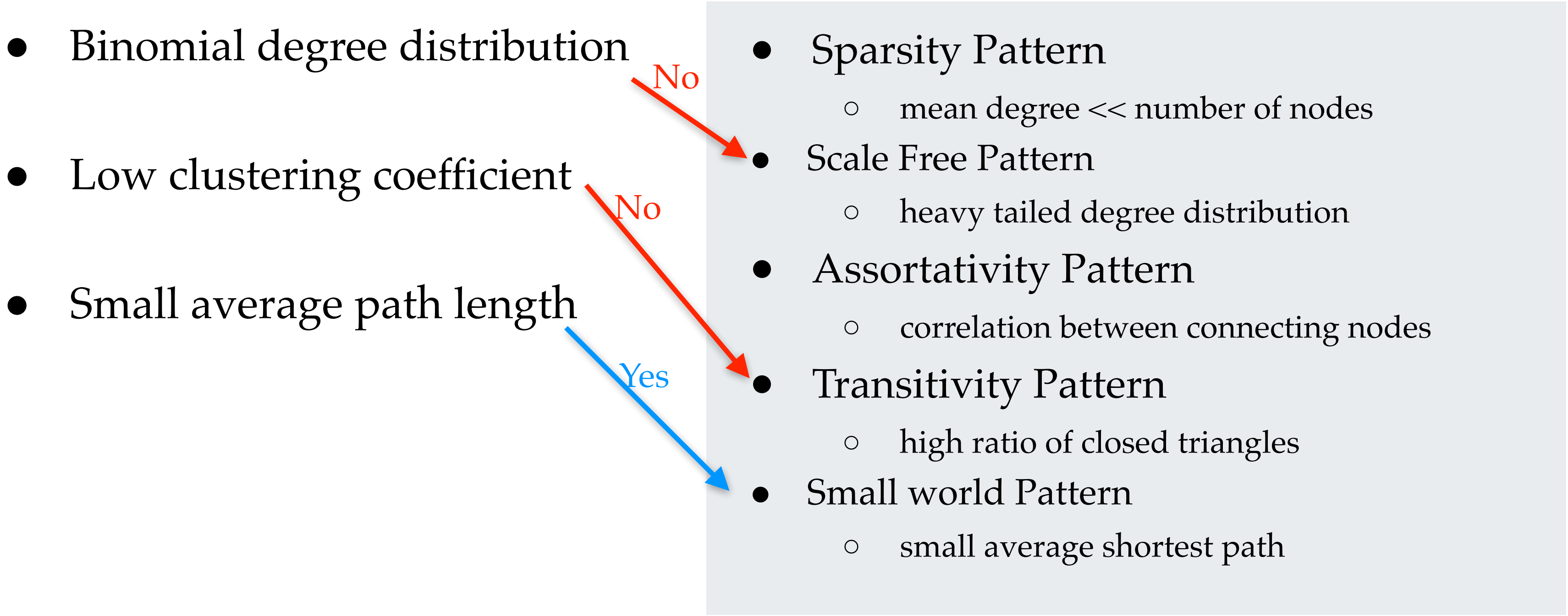
Here $n \cdot p$ =constant
That is, avg deg k is const

Compare it with the pattern in real world networks: Average shortest path distance in Facebook friendship networks of 100 US universities (with different sizes)

from Newman's book



Erdös-Rényi Model (ER) VS Real Graphs



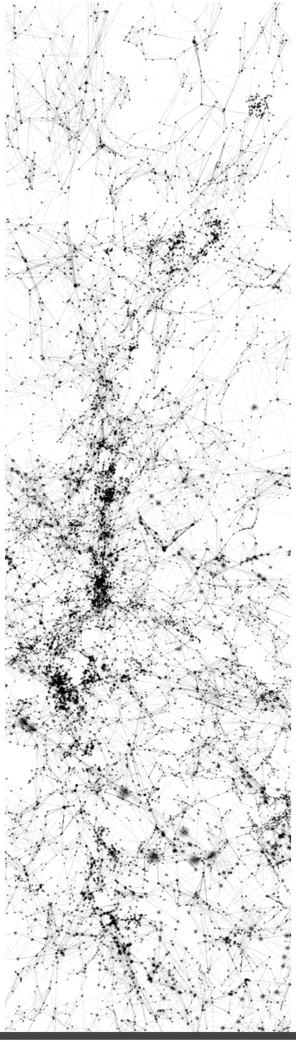
- Binomial degree distribution
- Low clustering coefficient
- Small average path length

- Sparsity Pattern
    - mean degree << number of nodes
- Scale Free Pattern
    - heavy tailed degree distribution
- Assortativity Pattern
    - correlation between connecting nodes
- Transitivity Pattern
    - high ratio of closed triangles
- Small world Pattern
    - small average shortest path

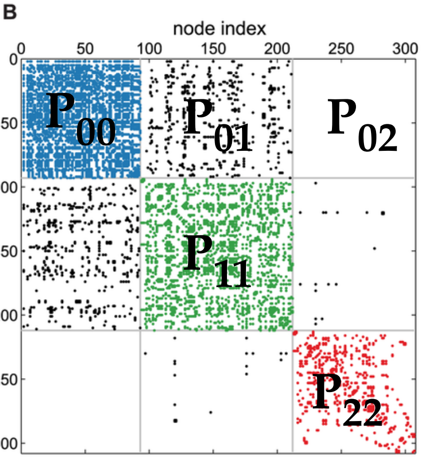
Real world graphs are not random

Outline

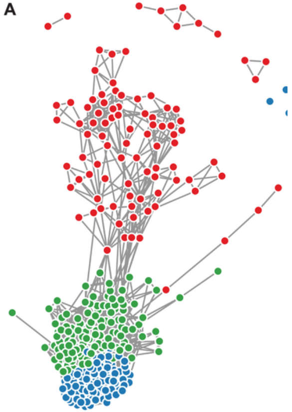
- Patterns Quick recap
- **Models**
  - ER model
  - **SBM**
  - Configuration model
  - AB model
  - FF model
  - Kronecker graph model
  - Fitting to observed graphs
  - LFR model



Stochastic Block Models (SBM)

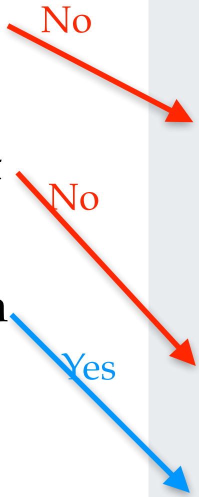


- Generalized ER to created block-structured graphs
- Parameters:
  - n: number of nodes
  - B: number of blocks, disjoint sets that divide the n nodes
  - P: $B \times B$ probabilities per each (and between any pairs of) block
- Generation: create an ER graph in each (within, between) block with the corresponding probability, i.e. probability of edge depends on the block memberships of its adjacent nodes
  - $p(A_{ij} = 1) = P_{b_i b_j}$, where b_i gives the block id of node i



Stochastic Block Models (SBM) VS Real Graphs

- Each block has Binomial degree distribution
- Low clustering coefficient
- Small average path length



- Sparsity Pattern
  - mean degree << number of nodes
- Scale Free Pattern
  - heavy tailed degree distribution
- Assortativity Pattern
  - correlation between connecting nodes
- Transitivity Pattern
  - high ratio of closed triangles
- Small world Pattern
  - small average shortest path

There is degree corrected block models, see here

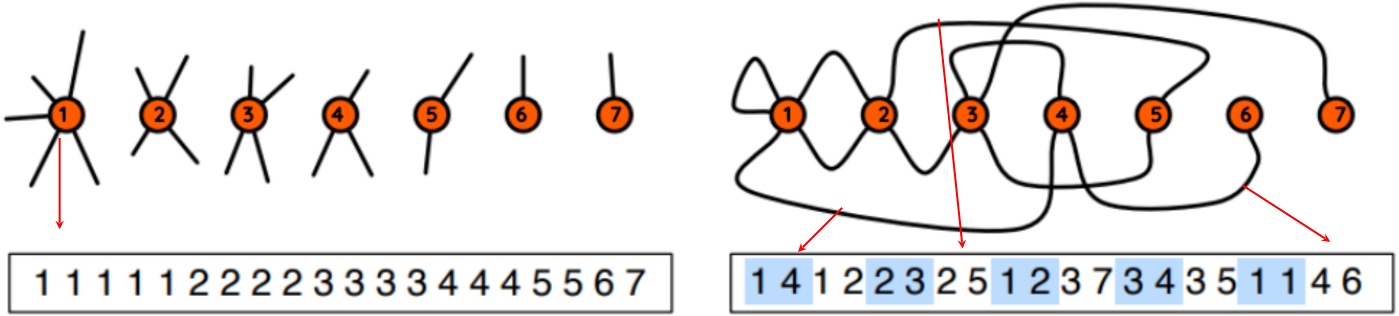
$$p(A_{ij} = 1) = Bernoulli(\theta_i \theta_j P_{b_i b_j})$$

Similar to ER

Configuration model

- By Mark Newman, generalizing ER to specific degree distribution
- Parameters: degree sequence (can be easily sampled from any distribution)
- Generation: assign slots, randomly connect them
- Serves as a null model for community detection
    - edges are distributed randomly given the degrees are fixed
    - communities that are not formed randomly should deviate from this

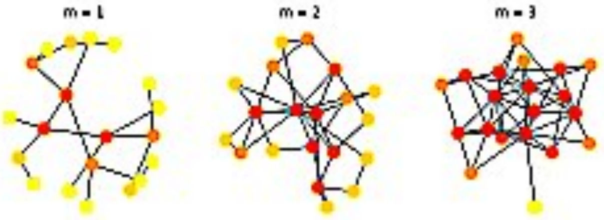
$$p_{ij} = \frac{k_i k_j}{2m - 1} \approx \frac{k_i k_j}{2m}$$



Slot
endpoint
node ids

Albert Barabasi Model (AB)

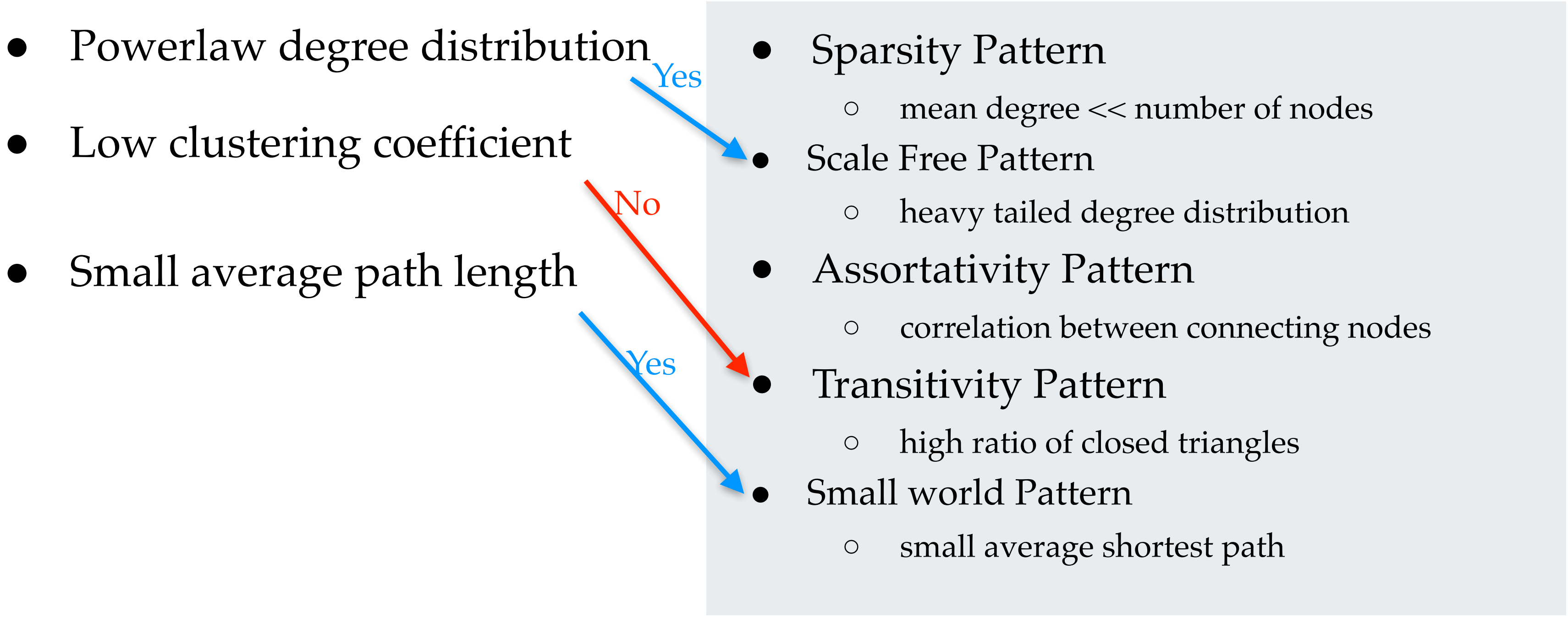
- Introduced in 1999, a.k.a Barabási–Albert (BA) model
- Uses preferential attachment which gives scale-free graphs
- Parameters: BA (n,m)
    - n: number of nodes
    - m: average degree



- Generation:
    - add one node at the time, add **m** connections per new node if possible
    - probability of forming a connection to an existing node is proportional to its degree:

$$p(i) = \frac{k_i}{\sum_j k_j} = \frac{k_i}{2m}$$

Albert Barabasi Model (AB) VS Real Graphs



- Powerlaw degree distribution
- Low clustering coefficient
- Small average path length

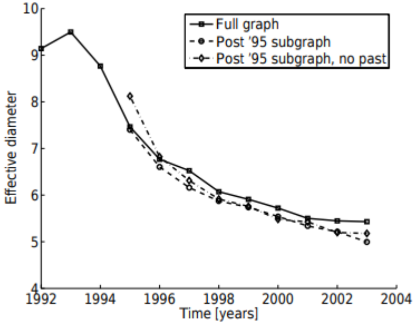
- Sparsity Pattern
  - mean degree << number of nodes
- Scale Free Pattern
  - heavy tailed degree distribution
- Assortativity Pattern
  - correlation between connecting nodes
- Transitivity Pattern
  - high ratio of closed triangles
- Small world Pattern
  - small average shortest path

Similar to Configuration Model

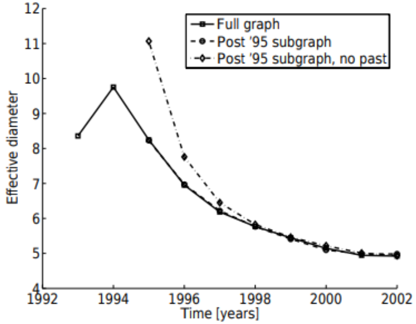
Evolution Patterns of Real Graphs: beyond static patterns

Looking at measures over time or as graph grows (x-axis usually time or number of nodes)

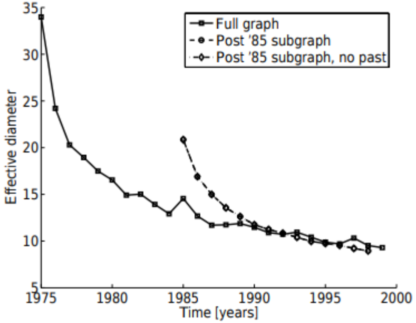
e.g. diameter shrinks over time in many real work graphs



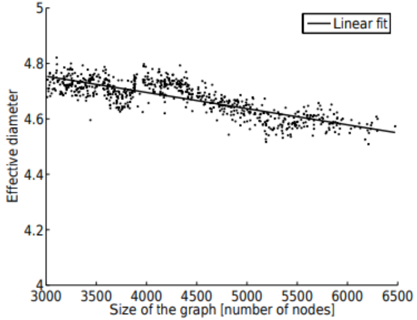
(a) arXiv citation graph



(b) Affiliation network



(c) Patents

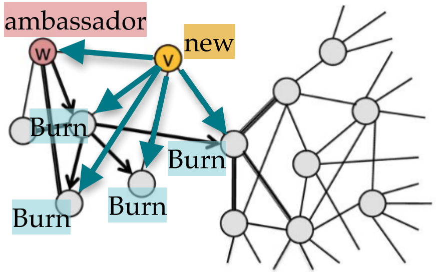


(d) AS

See more here: Graphs over Time: Densification Laws, Shrinking Diameters, and Possible Explanations

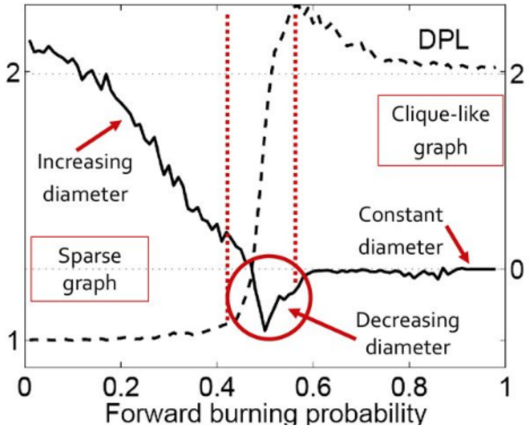
Forest Fire model (FF)

- By Leskovec, 2005
- To follow **evolution patterns** observed in real-world graphs
    - denser over time, the average degree increasing, and the diameter decreasing
- Parameters: n, p and rp
    - n: number of nodes
    - p: forward burning probability
    - r : backward burning probability
- Generation:
    - add a node at a time, connect the node to an ambassador, chosen uniformly at random
    - draw number of inlink and outlink from geometric distributions with means of p/(1 −p) and r/(1 −r) respectively
    - the new node recursively forms (out)links to the (in & out) neighbours of every node it connects to until fire dies



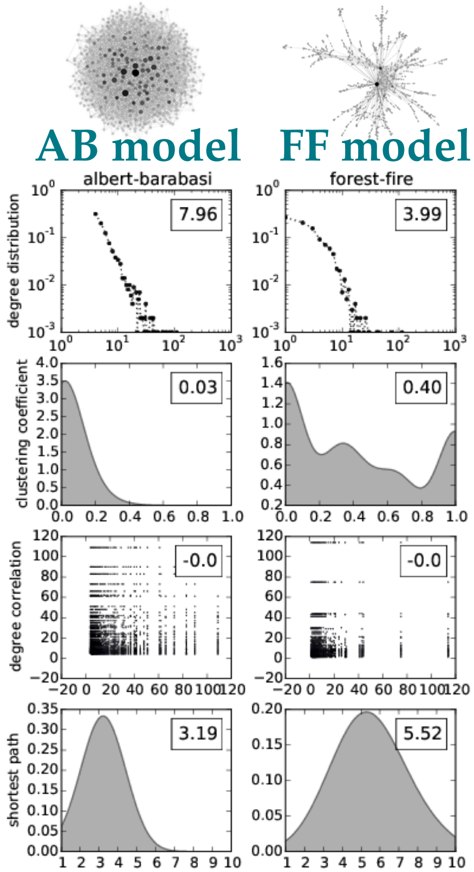
Forest Fire model (FF): properties

- Heavy-tailed degree distribution
    - rich get richer: older nodes have more chances to become ambassadors
- Densifies
    - newly entered node has more links to neighbours close to its ambassador
- Can result in shrinking diameter
    - Which is observed in real-world networks



Outline

- Patterns Quick recap
- Models
  - ER model
  - SBM
  - Configuration model
  - **AB model**
  - **FF model**
  - Kronecker graph model
  - Fitting to observed graphs
  - LFR model



AB model **FF model**

Kronecker graph model

Kronecker product of matrices

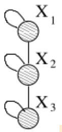
$$\mathbf{C} = \underset{\text{NxM}}{\mathbf{A}} \otimes \underset{\text{KxL}}{\mathbf{B}} \doteq \begin{pmatrix} a_{1,1}\mathbf{B} & a_{1,2}\mathbf{B} & \dots & a_{1,m}\mathbf{B} \\ a_{2,1}\mathbf{B} & a_{2,2}\mathbf{B} & \dots & a_{2,m}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1}\mathbf{B} & a_{n,2}\mathbf{B} & \dots & a_{n,m}\mathbf{B} \end{pmatrix}.$$

N*K x M*L

Based on self-similarity, generate graphs recursively [Leskovec, 2010]

whole has the same shape of its part

Consider a small initiator matrix, use kronecker products to get the adjacency matrix as $K_k = \underbrace{K_1 \otimes K_1 \otimes \dots K_1}_{k\ times} = K_{k-1} \otimes K_1$

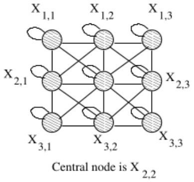


3x3

(a) Graph K_1

1	1	0
1	1	1
0	1	1

(d) Adjacency matrix of K_1



9x9

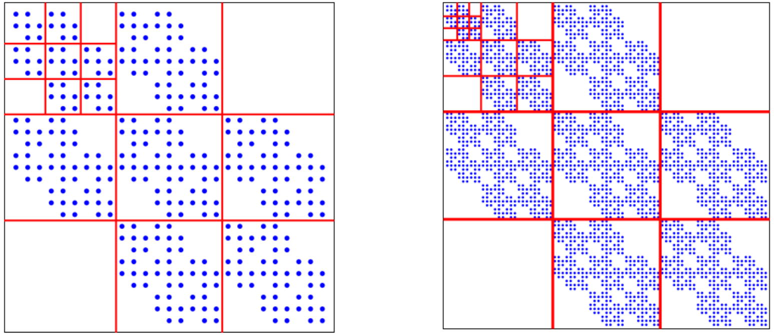
(c) Graph $K_2 = K_1 \otimes K_1$

K_1	K_1	0
K_1	K_1	K_1
0	K_1	K_1

(e) Adjacency matrix of $K_2 = K_1 \otimes K_1$

27x27

81x81



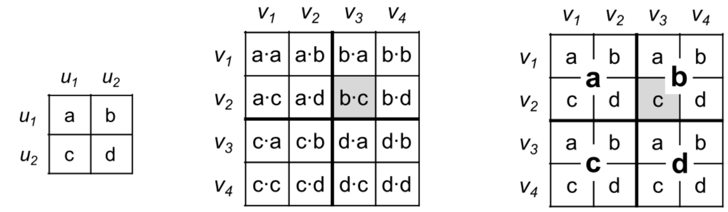
(a) K_3 adjacency matrix (27×27)

(b) K_4 adjacency matrix (81×81)

More here: https://snap-stanford.github.io/cs224w-notes/preliminaries/measuring-networks-random-graphs

Stochastic Kronecker graph model

Stochastic Kronecker graph, initiator matrix is probabilities and edges are drawn for the final graph with the corresponding probabilities



$$\begin{array}{c|cc} & u_1 & u_2 \\ \hline u_1 & a & b \\ u_2 & c & d \end{array}$$

$$\begin{array}{c|cccc} & v_1 & v_2 & v_3 & v_4 \\ \hline v_1 & a\cdot a & a\cdot b & b\cdot a & b\cdot b \\ v_2 & a\cdot c & a\cdot d & b\cdot c & b\cdot d \\ v_3 & c\cdot a & c\cdot b & d\cdot a & d\cdot b \\ v_4 & c\cdot c & c\cdot d & d\cdot c & d\cdot d \end{array}$$

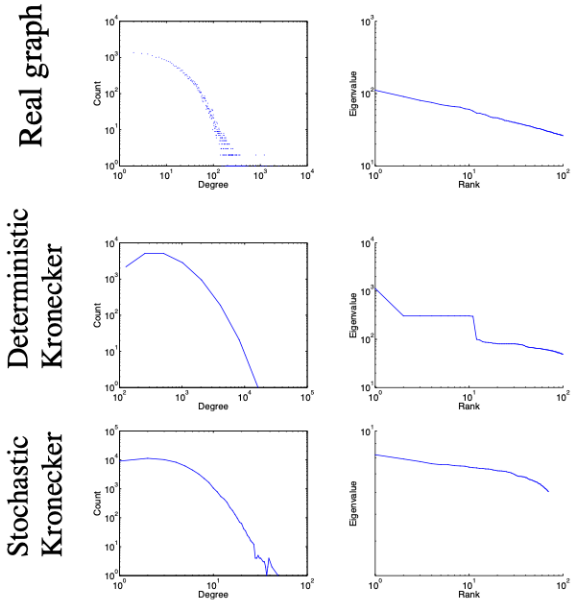
(a) 2×2 Stochastic Kronecker initiator $\mathcal{P}_1$

(b) Probability matrix $\mathcal{P}_2 = \mathcal{P}_1 \otimes \mathcal{P}_1$

(c) Alternative view of $\mathcal{P}_2 = \mathcal{P}_1 \otimes \mathcal{P}_1$

if all probabilities are equal in the initial matrix, this becomes equivalent to ER

how to generate efficiently? instead of n^2 toss coins, we can go hierarchal, sample graphs linearly, by considering how the probability matrix is generated, for more detail see here



(a) Degree distribution

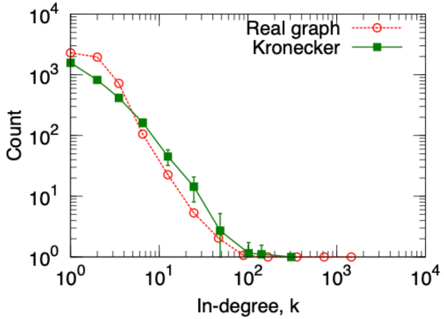
(b) Scree plot

Kronecker graph model

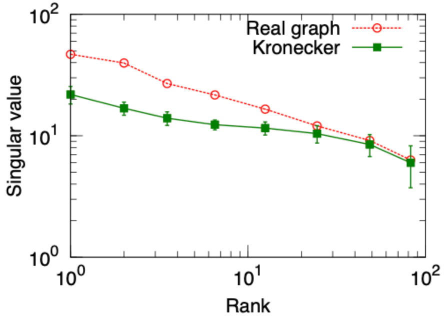
the initiator matrix can be set based on real-world data to sample similar graphs, by searching over what matrix is more likely to give the observed

$$\arg\max_{\Theta} P(G \mid \Theta^{[k]} \xleftarrow{\text{Kronecker}} \Theta)$$

for more detail see here



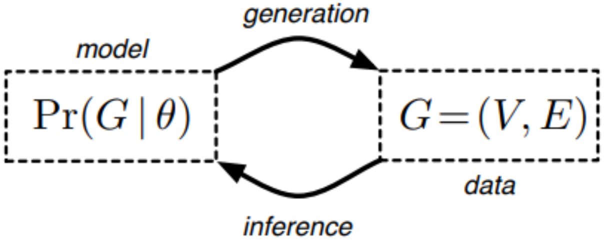
(a) Degree distribution



(c) Scree plot

Fitting to observed graphs: more general

- Option 1:
    - Measure and plot different characteristics of the observed graphs
    - Tune the parameters of the model to find a close enough fit to the observed patterns
- Option 2:
    - Define the likelihood of observing a graph, usually assuming edges are independent
    - Use maximum likelihood to find the model parameters



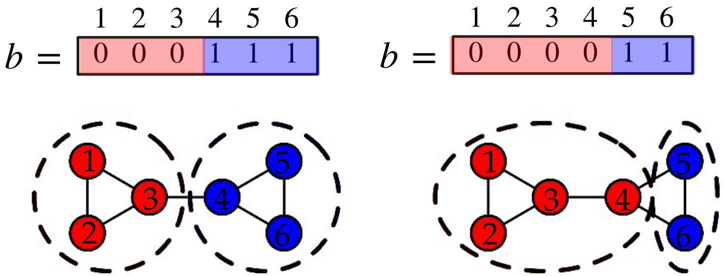
Fitting the SBM to data

Likelihood of G given Probability matrix P and partitioning b

$$\mathscr{L}(G \mid P, b) = \prod_{ij} P(i \to j \mid P, b)$$

$$\mathscr{L}(G \mid P, b) = \prod_{ij \in E} P(i \to j \mid P, b) \prod_{ij \notin E} 1 - P(i \to j \mid P, b)$$

$$\mathscr{L}(G \mid P, b) = \prod_{ij \in E} P_{b_i b_j} \prod_{ij \notin E} 1 - P_{b_i b_j}$$



$\mathcal{L}_{\text{good}} = 0.043304\ldots$
$\ln \mathcal{L}_{\text{good}} = -3.1395\ldots$

P M_{good}	red	blue
red	3/3	1/9
blue	1/9	3/3

$\mathcal{L}_{\text{bad}} = 0.000244\ldots$
$\ln \mathcal{L}_{\text{bad}} = -8.3178\ldots$

P M_{bad}	red	blue
red	4/6	2/8
blue	2/8	1/1

Recall in SBM:

$p(A_{ij} = 1) = P_{b_i b_j}$, where b_i gives the block id of node i

Lancichinetti, Fortunato, and Radicchi (LFR) model

- Extends the configuration model
- Sample degree sequence and block sizes from power law distributions
- Randomly assign nodes to blocks according to sampled block sizes
- Wire nodes based on configuration model and the sampled degree sequence
- Rewire until each node has a fixed fraction, μ, of links going outside its block