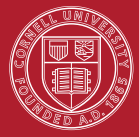


Lecture 9- Random Graph Models and Real Networks

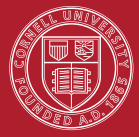
Graph-Based Data Science For Networked Systems
ECE 5260 – ORIE 5735

Anna Scaglione

Cornell Tech

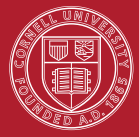
February 23, 2026

Content



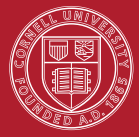
Cornell University

Outline



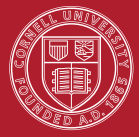
Cornell University

Introduction



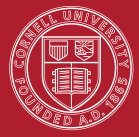
- The last lecture slides introduced two random graph models:
  1. First the Erdös Renyi or Poisson random graph model $\mathcal{G}(N, p)$, that is amenable to derive analytically a number of average features
  2. The second is called *the configuration model*, which allows to fix the degree sequence or its distribution.
- Both are useful to understand trends and allow analytical studies when network data are lacking
- Today we complete their treatment with additional models and talk about features observed in real networks

Introduction



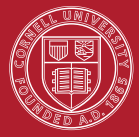
- The last lecture slides introduced two random graph models:
  1. First the Erdös Renyi or Poisson random graph model $\mathcal{G}(N, p)$, that is amenable to derive analytically a number of average features
  2. The second is called *the configuration model*, which allows to fix the degree sequence or its distribution.
- Both are useful to understand trends and allow analytical studies when network data are lacking
- Today we complete their treatment with additional models and talk about features observed in real networks

Introduction



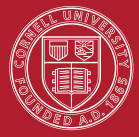
- The last lecture slides introduced two random graph models:
  1. First the Erdös Renyi or Poisson random graph model $\mathcal{G}(N, p)$, that is amenable to derive analytically a number of average features
  2. The second is called *the configuration model*, which allows to fix the degree sequence or its distribution.
- Both are useful to understand trends and allow analytical studies when network data are lacking
- Today we complete their treatment with additional models and talk about features observed in real networks

Introduction



- The last lecture slides introduced two random graph models:
  1. First the Erdös Renyi or Poisson random graph model $\mathcal{G}(N, p)$, that is amenable to derive analytically a number of average features
  2. The second is called *the configuration model*, which allows to fix the degree sequence or its distribution.
- Both are useful to understand trends and allow analytical studies when network data are lacking
- Today we complete their treatment with additional models and talk about features observed in real networks

Outline

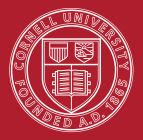


Cornell University

Outline

Cornell University

Generalizations of $\mathcal{G}(N, p)$

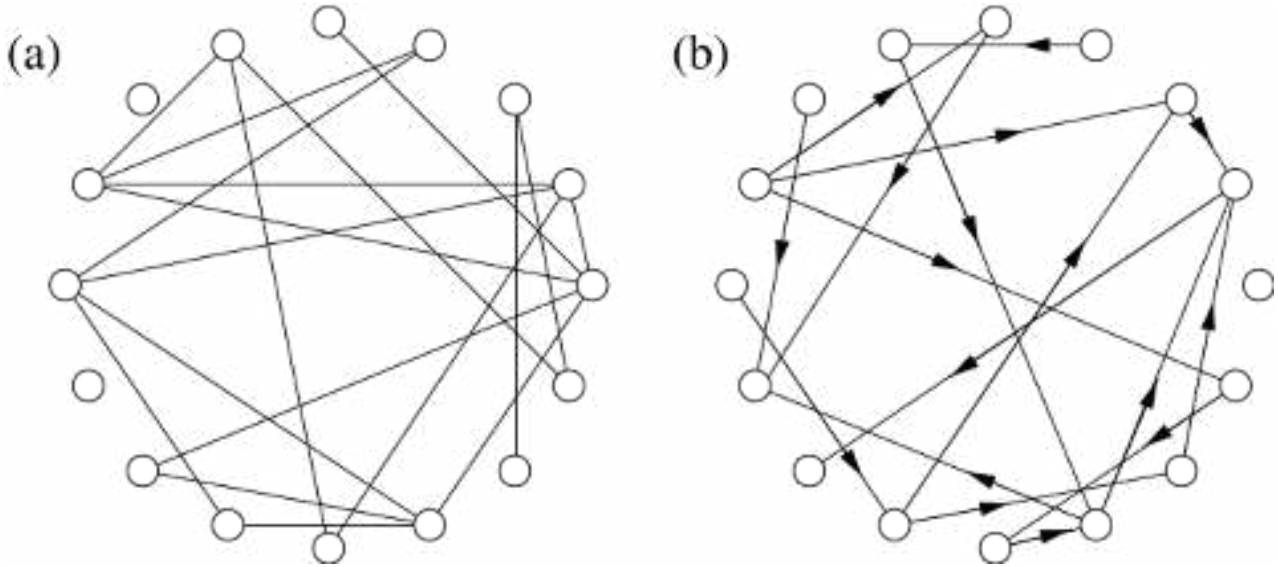


- Last time we only discussed undirected graphs
- The random graph model can be generalized to the following cases
  1. Directed networks
  2. Bipartite networks
  3. Acyclic networks

Outline

Cornell University

Random Directed networks



- The $\mathcal{G}(N, p)$ is easy to generalize: the random edge is assigned a direction

Directed configuration model

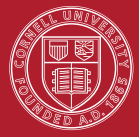
- The generalization is also simple: one must create stubs we the specified in and out degree sequences $\boldsymbol{d}^{\text{in}}$ and $\boldsymbol{d}^{\text{out}}$
- Note that we need to make sure that the number of outgoing and ingoing stubs match:

$$|\mathcal{E}| = M = \sum_u d_u^{\text{in}} = \sum_u d_u^{\text{out}}$$

- Many of the calculations we made can be generalized. For instance there are $d_u^{\text{in}} \cdot d_v^{\text{out}}$ possible connections between nodes u, v. Thus probability of an edge between nodes u and v

$$p_{uv} = \frac{d_u^{\text{in}} \cdot d_v^{\text{out}}}{M}$$

Directed configuration model



- The generalization is also simple: one must create stubs we the specified in and out degree sequences $\boldsymbol{d}^{\text{in}}$ and $\boldsymbol{d}^{\text{out}}$
- Note that we need to make sure that the number of outgoing and ingoing stubs match:

$$|\mathcal{E}| = M = \sum_u d_u^{\text{in}} = \sum_u d_u^{\text{out}}$$

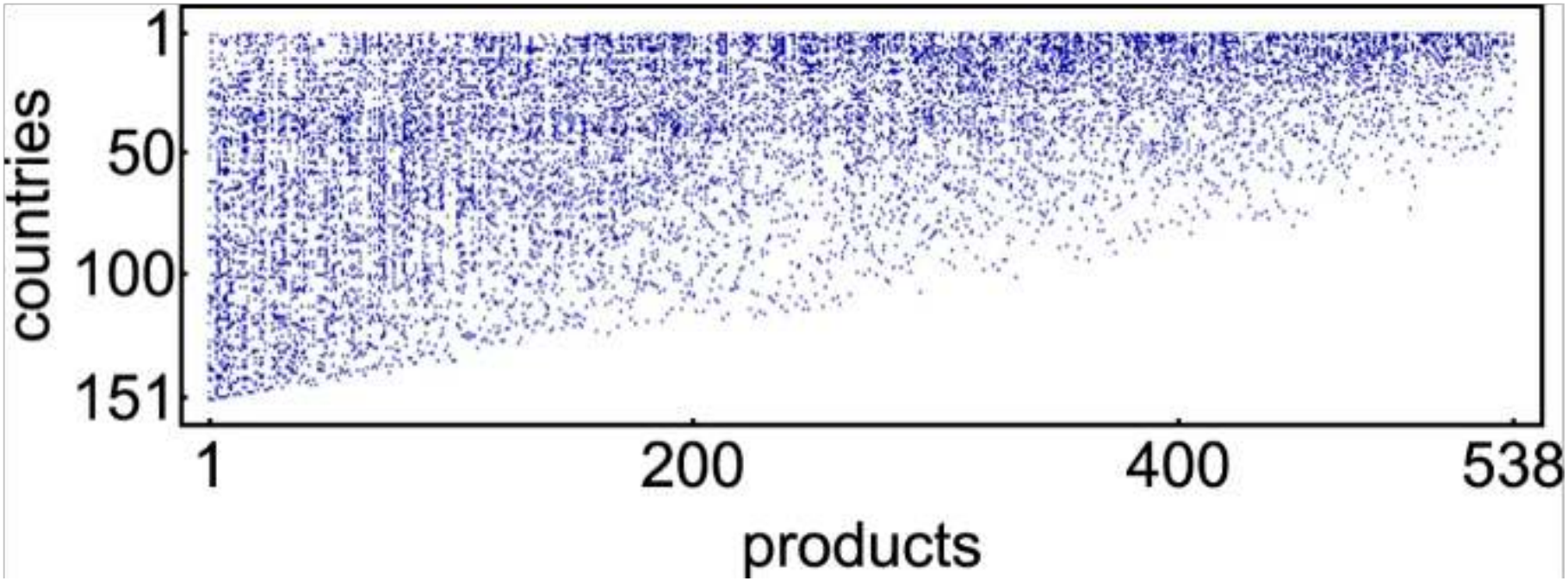
- Many of the calculations we made can be generalized. For instance there are $d_u^{\text{in}} \cdot d_v^{\text{out}}$ possible connections between nodes u, v. Thus probability of an edge between nodes u and v

$$p_{uv} = \frac{d_u^{\text{in}} \cdot d_v^{\text{out}}}{M}$$

Outline

Cornell University

Bipartite Graphs



- We already discussed the important role of Bipartite graphs in describing associations

Bipartite Random Graphs

- Bipartite Random Graph (BiRG): The construction of a random bipartite graph entails
  - defining the two sets of nodes $\mathcal{U}, \mathcal{V}$
  - selecting the nodes in $\mathcal{U}$ connecting them with probability p with the nodes in $\mathcal{V}$
- Bipartite Directed Configuration Model (BiDCM): The configuration model for bipartite networks is simple too:
  - create two sets of stubs
  - select uniformly at random the matching stubs from one set to the other
- See e.g. the python implementation

Bipartite Random Graphs

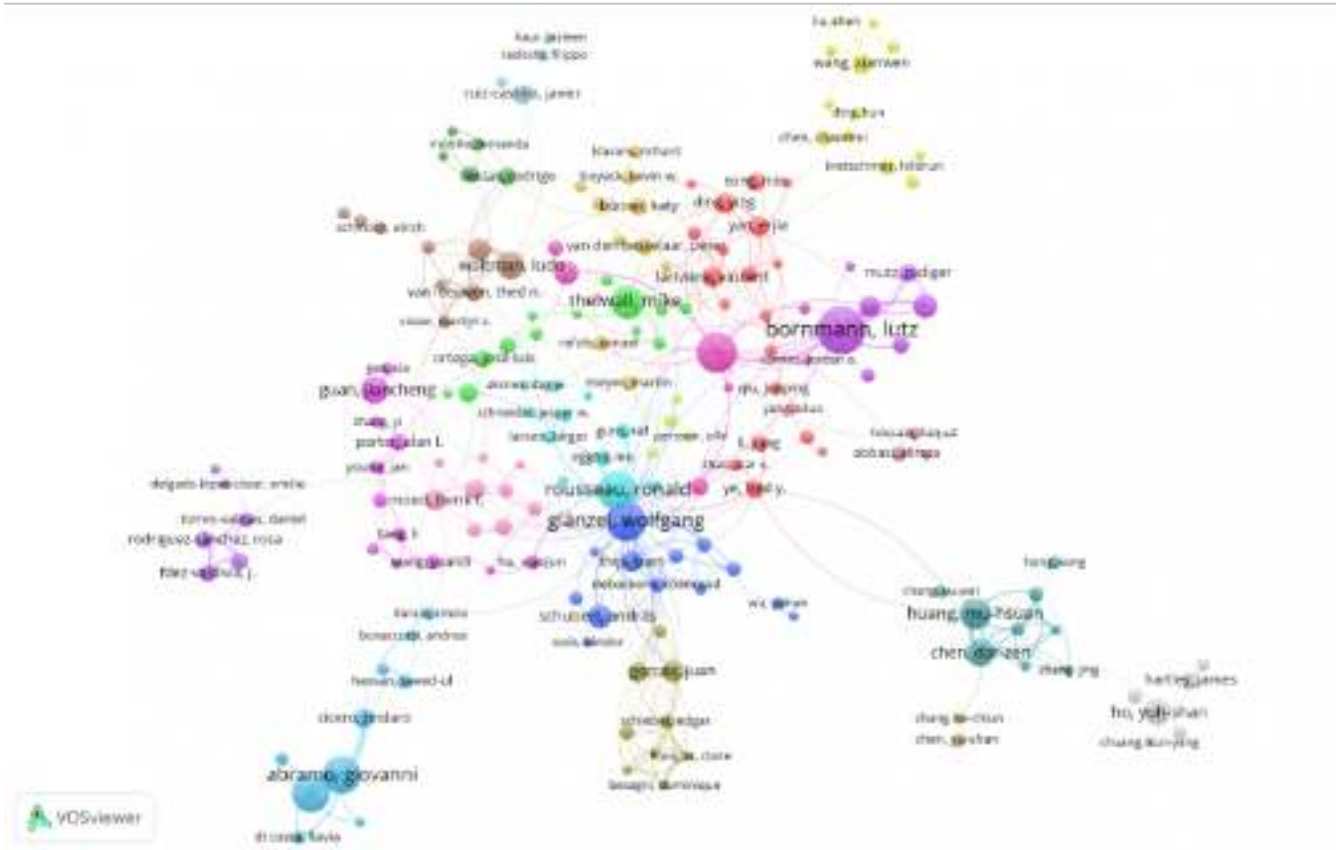
- Bipartite Random Graph (BiRG): The construction of a random bipartite graph entails
  - defining the two sets of nodes $\mathcal{U}, \mathcal{V}$
  - selecting the nodes in $\mathcal{U}$ connecting them with probability p with the nodes in $\mathcal{V}$
- Bipartite Directed Configuration Model (BiDCM): The configuration model for bipartite networks is simple too:
  - create two sets of stubs
  - select uniformly at random the matching stubs from one set to the other
- See e.g. the python implementation

Outline

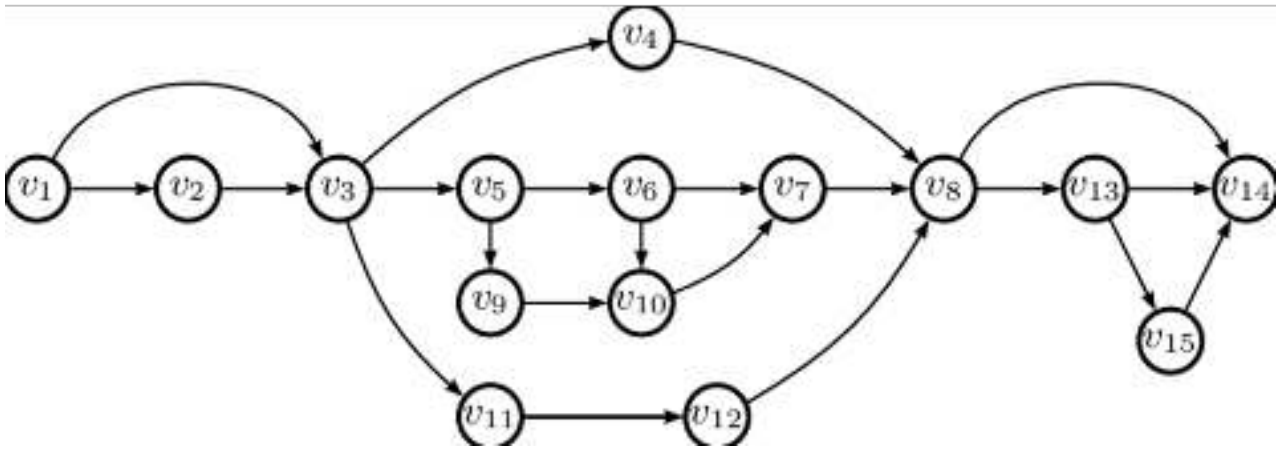
Cornell University

Random acyclic networks

- Knowledge networks examples are directed and often naturally acyclic.
- The classical example is that of citation networks.

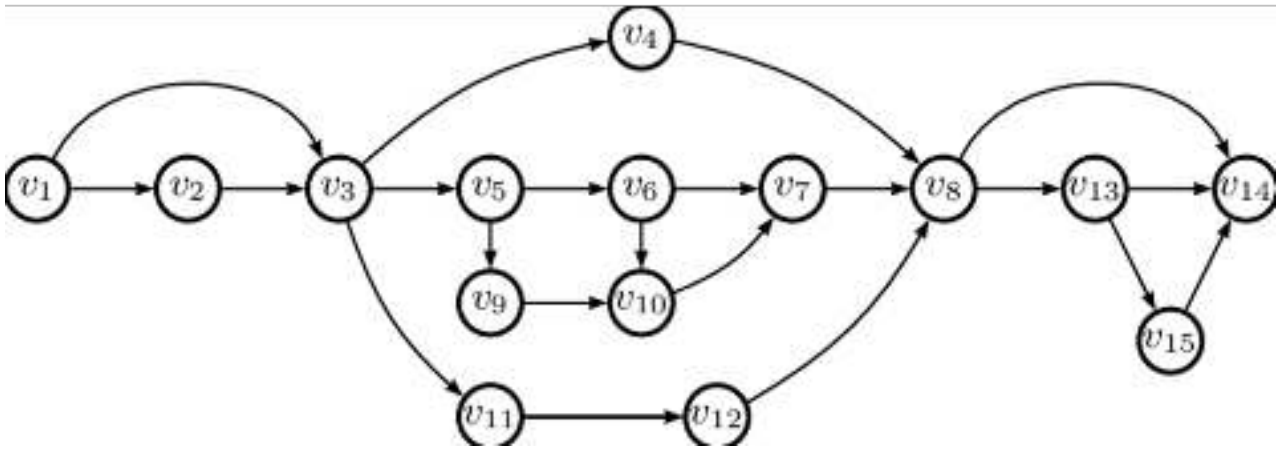


Random acyclic networks



- In an acyclic network there are no closed directed loops
- The nodes of such a graph can be arranged on a line, so that the edges all go in the same direction (topological ordering)
- If one thinks of citation networks, it seems natural that the ordering happens in reverse chronological order of publication, with citations directed from the latest to the earliest

Random acyclic networks

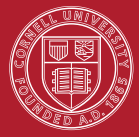


- In an acyclic network there are no closed directed loops
- The nodes of such a graph can be arranged on a line, so that the edges all go in the same direction (topological ordering)
- If one thinks of citation networks, it seems natural that the ordering happens in reverse chronological order of publication, with citations directed from the latest to the earliest

Random acyclic networks

- To create a random acyclic graph one can leverage the property of the existence of an ordering
  1. start arranging the nodes in order;
  2. place the corresponding in and outgoing stubs at each node;
  3. work backward connecting the nodes one by one, starting from the earliest to the latest, connecting outgoing stubs we encounter to an in-going one chosen at random among the earliest nodes stubs.

Random acyclic networks



- To create a random acyclic graph one can leverage the property of the existence of an ordering
  1. start arranging the nodes in order;
  2. place the corresponding in and outgoing stubs at each node;
  3. work backward connecting the nodes one by one, starting from the earliest to the latest, connecting outgoing stubs we encounter to an in-going one chosen at random among the earliest nodes stubs.

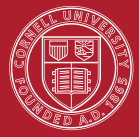
Random acyclic networks

- The ordering in the representation is analogous to the property that there exist an ordering for which the adjacency matrix $\boldsymbol{A}$ is lower triangular
- Let $v = 1, \ldots, N$ be the ordering. Not all $\boldsymbol{d}^{\text{in}}, \boldsymbol{d}^{\text{out}}$ are possible. First, for a directed graph $M = \sum_u d_u^{\text{in}} = \sum_u d_u^{\text{out}}$
- Also the out degree of node v cannot be greater than the number of unused stubs of all the earlier nodes:

$$d_v^{\text{out}} \leq \sum_{u=1}^{v-1} d_u^{\text{in}} - \sum_{u=1}^{v-1} d_u^{\text{out}} \quad \Rightarrow \quad \sum_{u=1}^{v-1} d_u^{\text{in}} \geq \sum_{u=1}^{v} d_u^{\text{out}}$$

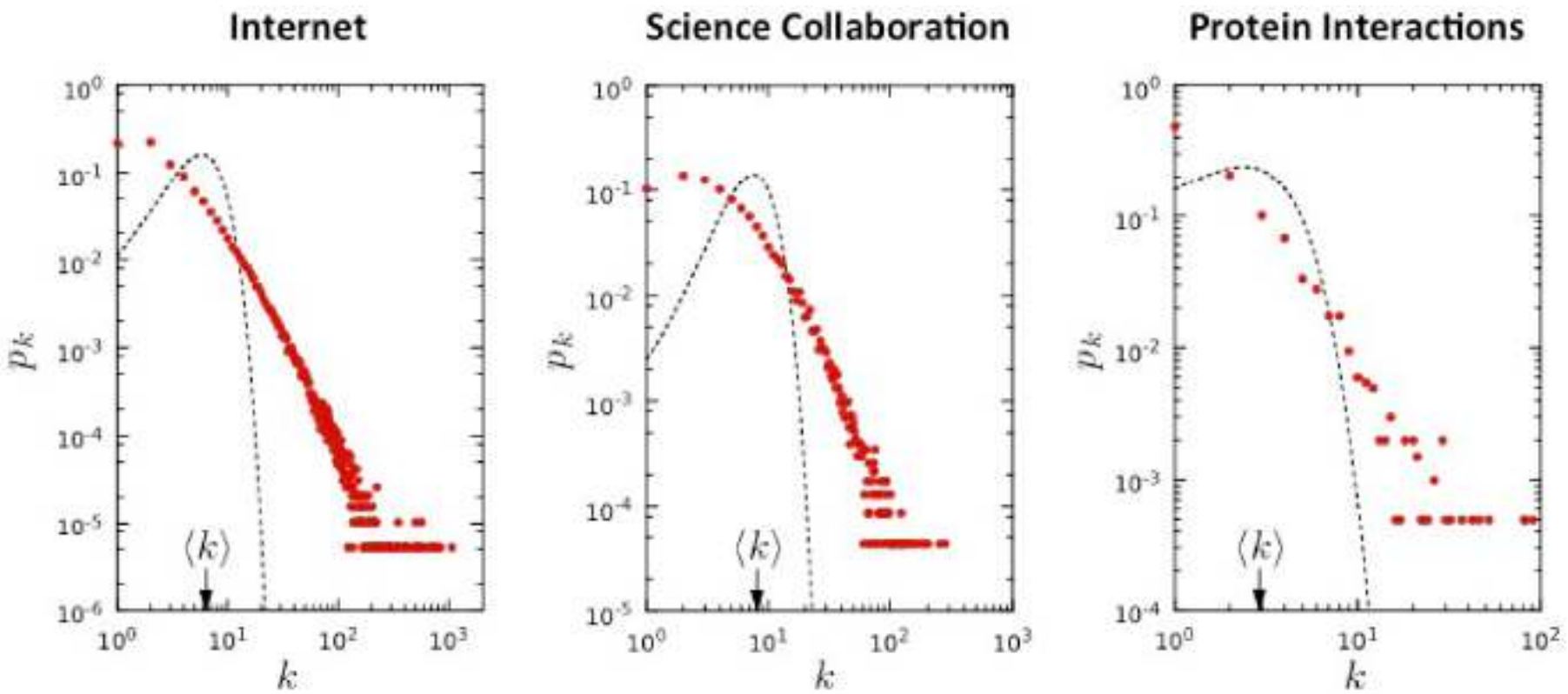
- Unlike the previous cases, unfortunately for these graphs it is hard to extract expectations of various features

Outline



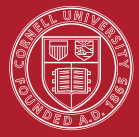
Cornell University

Reproducing properties of real networks



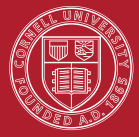
- The configuration model can replicate arbitrary degree distributions while the Poisson random model is not realistic.

Reproducing properties of real networks



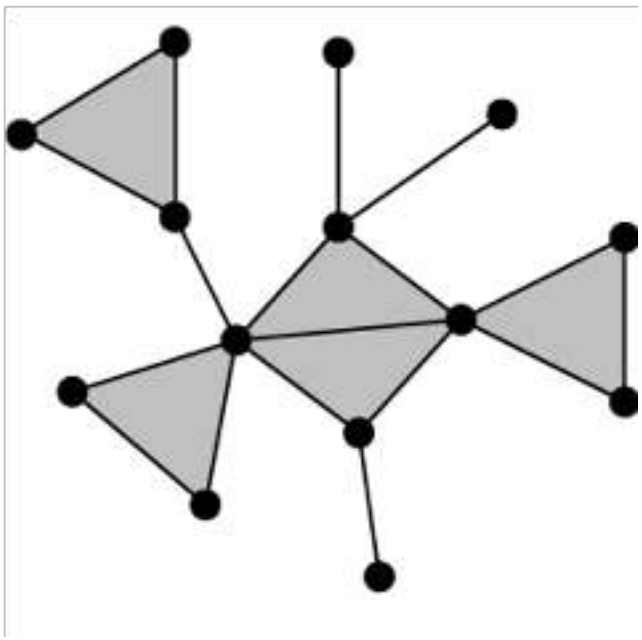
- Can we reproduce other features of real networks? Examples are the following:
  1. Clustering coefficient
  2. Assortative mixing and community structure
  3. The Small World effect
  4. Dynamic networks

Outline



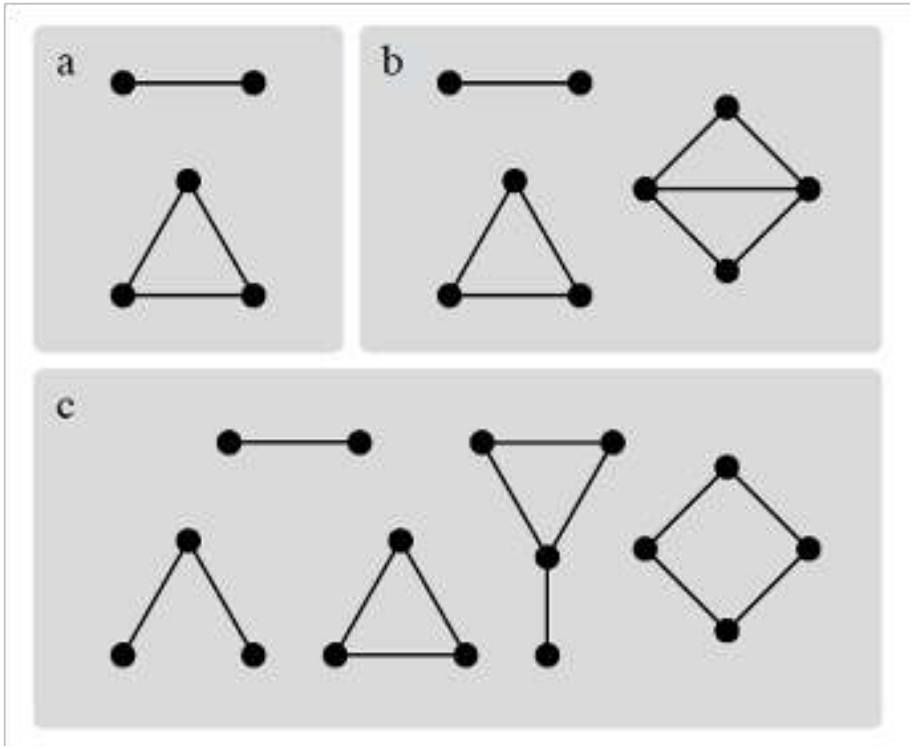
Cornell University

Fixed degree and clustering coefficient



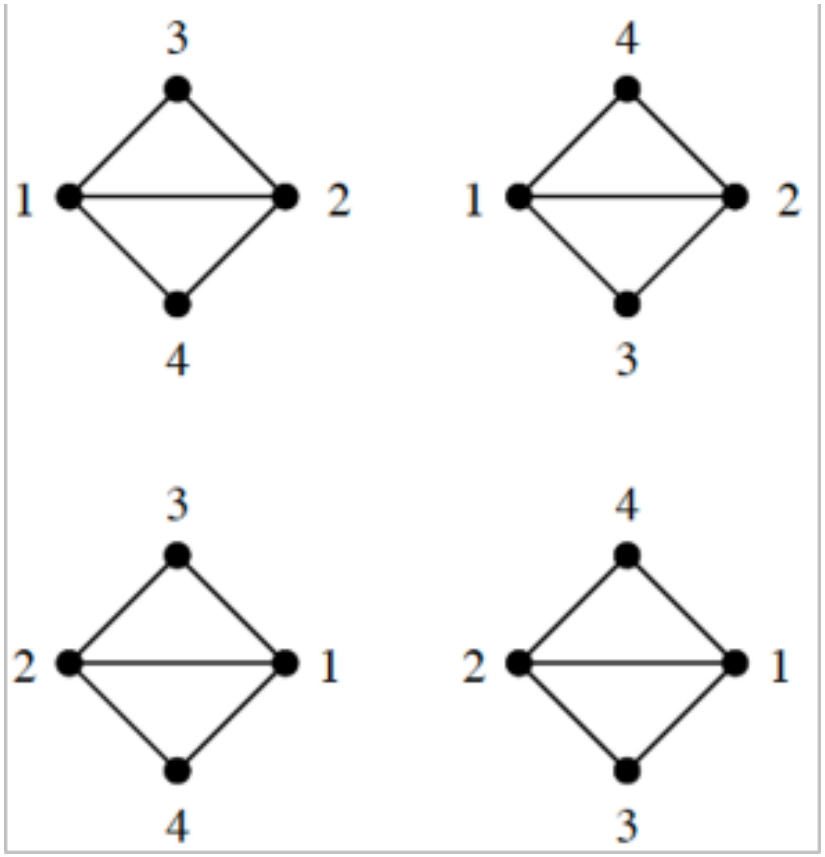
- In the configuration model the nodes become small star graphs that are then connected at random
- The idea can be generalized by creating a certain desired number of triangles, to yield the desired **clustering coefficient**
- A **stub** includes a certain number of single edges and triangles that need to be matched

Generalization to arbitrary subgraph motifs



- Other graph motifts could be considered in addition to the one in Fig. a we just mentioned
- For the general random construction rather than the **degree sequence**, we need what is called the **roles sequence**

Role sequence

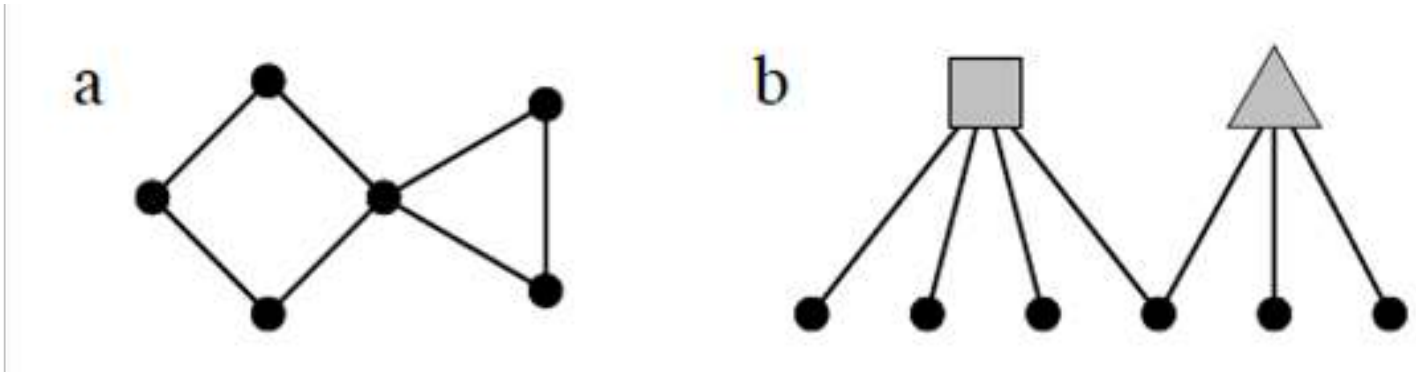


Nodes 1 and 2 can have two possible positions, and so do 3 and 4. Each node has two roles.

- The roles of a node in a sub-graph enumerate each of the topologically distinct situations in which a node can be, while maintaining the edge pairs the same.
- For small subgraphs, the enumeration of the roles is not hard to do brute force, but is

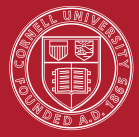
Matching with bipartite representation

An alternative representation is as a bipartite graph:



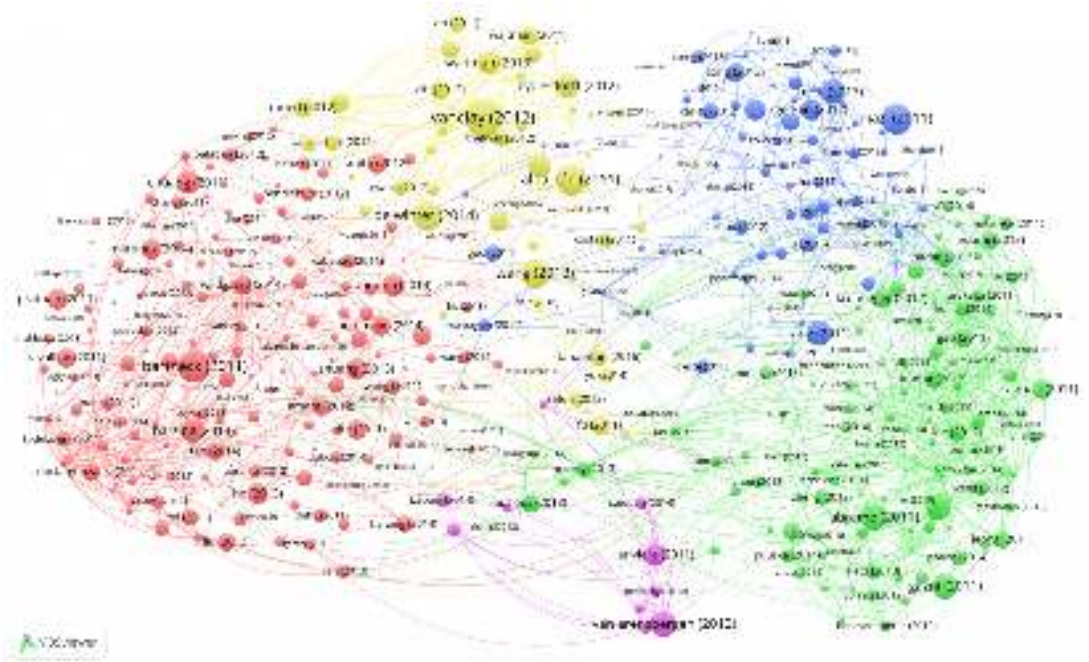
- Set $\mathcal{V}$ is the set of nodes of the original network and $\mathcal{U}$ contains the subgraphs motifs present (e.g. square and triangles, in the figure)

Outline



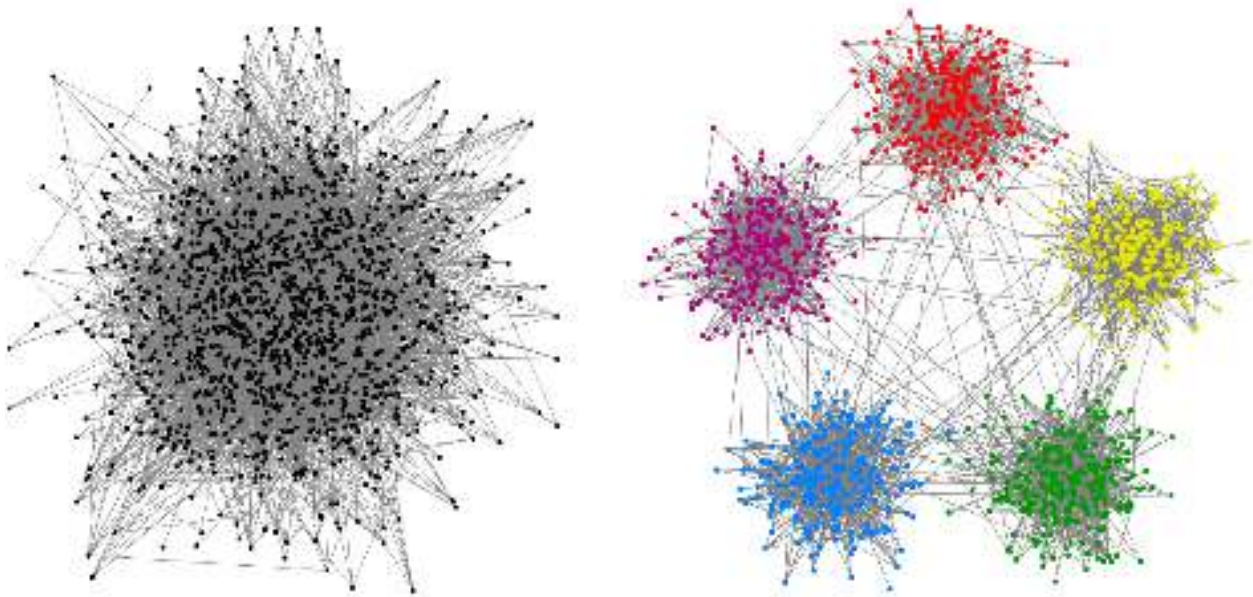
Cornell University

Community struture



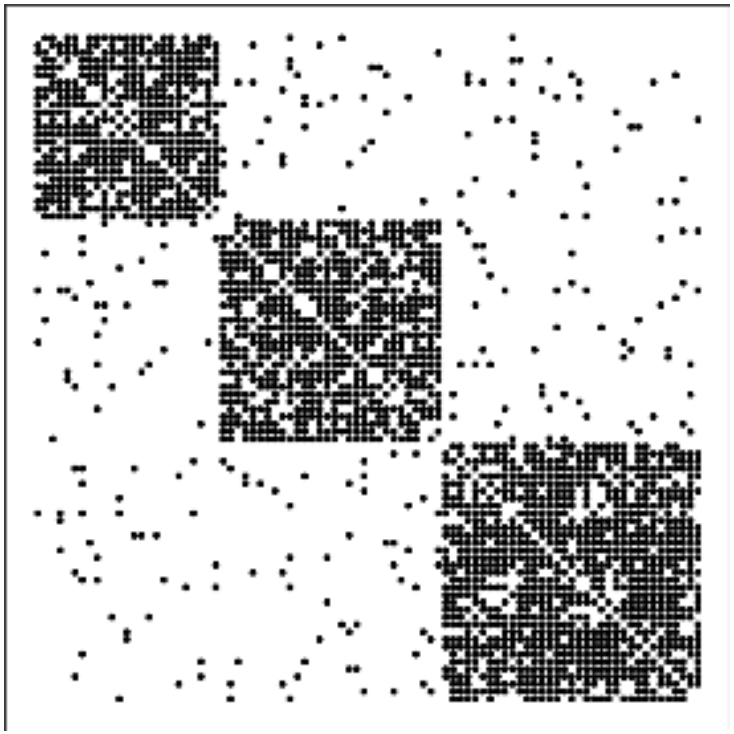
- Nodes in a certain group are associated often with different node features. We called that property assortatitivity and used the pearson coefficient to measure it.
- If the network is assortative with respect to a certain feature that determines connectivity, a community structure emerges

Stochastic Block Model



- Similar Poisson random graph but generates graphs with clusters
- Nodes are divided in K groups $\{\mathcal{V}_1, \ldots, \mathcal{V}_K\}$, based on how many nodes each group should have, or the nodes for each group are assigned at random, with probability $P(k), k = 1, \ldots, K$

Stochastic Block Model



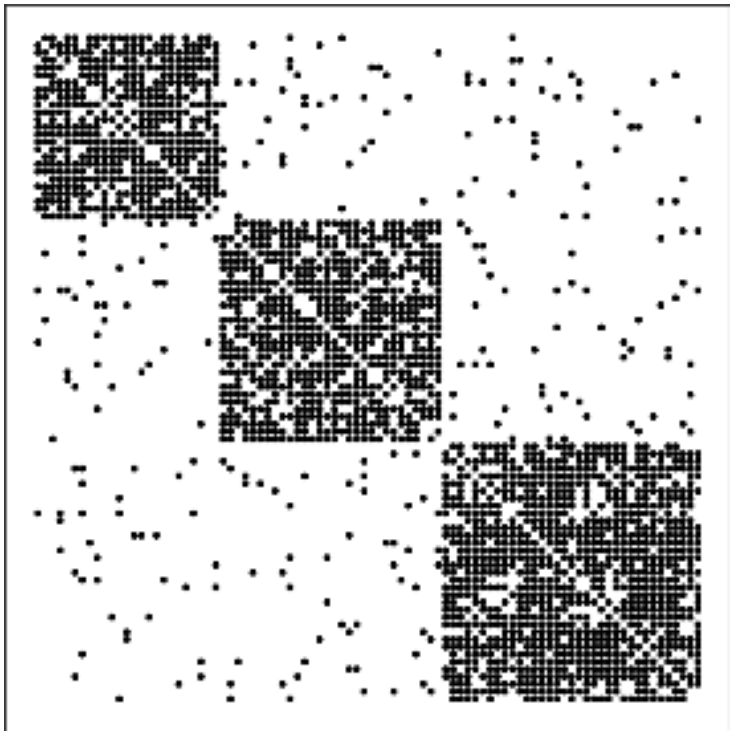
The degree distribution can be obtained considering that $v \in V_k,\ d_v = \sum_{k=1}^{K} |\mathcal{N}_u \cap \mathcal{V}|$, which is a sum of independent random variables, each with a Binomial distribution.

- The basic idea is to have a certain probability γ_k of connections within the group, and a different probability, $\gamma_{kk'}$ that connections exist between nodes in $\mathcal{V}_k$ and $\mathcal{V}_{k'}$
- Normally these quantities are arranged into a matrix $\mathbf{\Gamma}$, with

$$[\mathbf{\Gamma}]_{kk} = \gamma_k > [\mathbf{\Gamma}]_{kk'} = p_{kk'}$$

- Like in $\mathcal{G}(N, p)$, within the group the probability tends to be Poisson, which does not fit real networks

Stochastic Block Model



The degree distribution can be obtained considering that $v \in V_k,\ d_v = \sum_{k=1}^{K} |\mathcal{N}_u \cap \mathcal{V}|$, which is a sum of independent random variables, each with a Binomial distribution.

- The basic idea is to have a certain probability γ_k of connections within the group, and a different probability, $\gamma_{kk'}$ that connections exist between nodes in $\mathcal{V}_k$ and $\mathcal{V}_{k'}$
- Normally these quantities are arranged into a matrix $\mathbf{\Gamma}$, with

$$[\mathbf{\Gamma}]_{kk} = \gamma_k > [\mathbf{\Gamma}]_{kk'} = p_{kk'}$$

- Like in $\mathcal{G}(N, p)$, within the group the probability tends to be Poisson, which does not fit real networks

Configuration and Chung Lu models

- Also in this case, for a better fit, one applies a generalization of the configuration model
- More commonly, a generalized Chung Lu model is used
- Recall that the Chung-Lu model only seeks to match the degree of a node on average $\mathbb{E}[d_u] = c_u$ and places randomly an edge (u, v) between two nodes with probability:

$$p_{uv} = \frac{c_u c_v}{2M}, \qquad M = |\mathcal{E}|$$

- The difference is that there is now also another parameter $[\mathbf{\Pi}]_{kk'} = \pi_{kk'}$ that is used to determine at random in which groups u and v should be.

Configuration and Chung Lu models

- Also in this case, for a better fit, one applies a generalization of the configuration model
- More commonly, a generalized Chung Lu model is used
- Recall that the Chung-Lu model only seeks to match the degree of a node on average $\mathbb{E}[d_u] = c_u$ and places randomly an edge (u, v) between two nodes with probability:

$$p_{uv} = \frac{c_u c_v}{2M}, \quad M = |\mathcal{E}|$$

- The difference is that there is now also another parameter $[\mathbf{\Pi}]_{kk'} = \pi_{kk'}$ that is used to determine at random in which groups u and v should be.

Configuration and Chung Lu models

- Also in this case, for a better fit, one applies a generalization of the configuration model
- More commonly, a generalized Chung Lu model is used
- Recall that the Chung-Lu model only seeks to match the degree of a node on average $\mathbb{E}[d_u] = c_u$ and places randomly an edge (u, v) between two nodes with probability:

$$p_{uv} = \frac{c_u c_v}{2M}, \qquad M = |\mathcal{E}|$$

- The difference is that there is now also another parameter $[\mathbf{\Pi}]_{kk'} = \pi_{kk'}$ that is used to determine at random in which groups u and v should be.

Configuration and Chung Lu models

Cornell University

- There are constraints in choosing $\pi_{kk'}$ if we want $\mathbb{E}[d_u] = c_u$
- In fact, from the random construction, for a node in $u \in \mathcal{V}_k$:

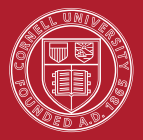
$$\mathbb{E}[d_u] = \sum_{k'=1}^{K} \pi_{kk'} \frac{c_u c_v}{2M}$$

- Hence:

$$\sum_{k,k'} \pi_{kk'} \frac{c_u c_v}{2M} = c_u, \quad \Rightarrow \quad c_u \overbrace{\sum_{k'} \pi_{kk'} \frac{c_v}{2M}}^{\sum_{k'} \pi_{kk'} \frac{c_v}{2M} = 1} = c_u$$

$$\Rightarrow \quad \sum_{k'} \pi_{kk'} c_v = 2M$$

Configuration and Chung Lu models



- There are constraints in choosing $\pi_{kk'}$ if we want $\mathbb{E}[d_u] = c_u$
- In fact, from the random construction, for a node in $u \in \mathcal{V}_k$:

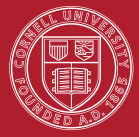
$$\mathbb{E}[d_u] = \sum_{k'=1}^{K} \pi_{kk'} \frac{c_u c_v}{2M}$$

- Hence:

$$\sum_{k,k'} \pi_{kk'} \frac{c_u c_v}{2M} = c_u, \quad \Rightarrow \quad c_u \overbrace{\sum_{k'} \pi_{kk'} \frac{c_v}{2M} \equiv 1}^{\sum_{k'} \pi_{kk'} \frac{c_v}{2M}} \sum_{k'} \pi_{kk'} \frac{c_v}{2M} = c_u$$

$$\Rightarrow \quad \sum_{k'} \pi_{kk'} c_v = 2M$$

Outline

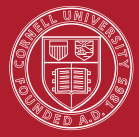


Cornell University

The Small World effect

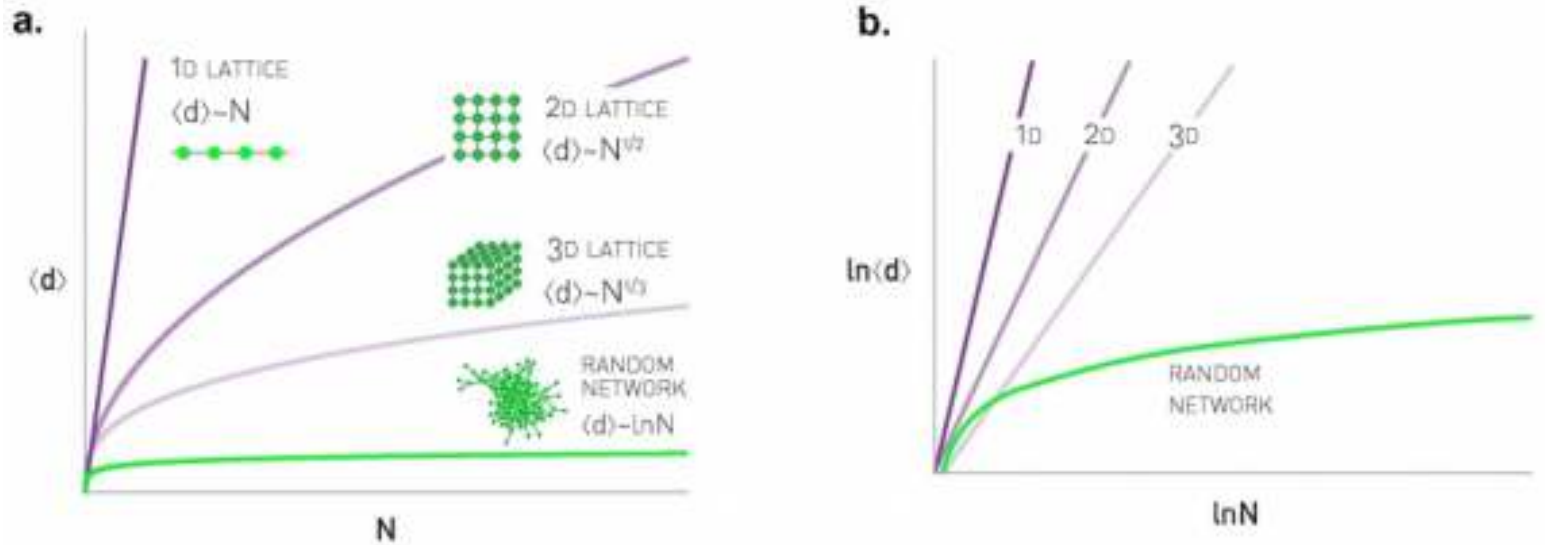
- **1960s Milgram experiment** A group of volunteers reached with a letter sent to acquaintances, asked them to forward it to their acquaintances, to reach possibly a target person in Boston, that the volunteers did not know. It took about 6 steps before someone knew that person.
- While social networks are clustered they seem to feature individuals that can *funnel information* far away

Random Small World Graph models



Cornell University

- We said that for a Poisson random graphs with N nodes the expected distance s and average degree c are such that $s \approx \frac{\ln N}{\ln c}$
- Say that the average degree say 10, in 6 steps we can reach a million people, since $\ln 10^6 / \ln 10 = 6$
- This distance is **very different** for networks that are connected regularly, on a line a grid a cube.

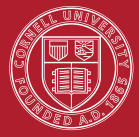


$< d >$ = average distance

Small World random graphs

- We said that real networks hardly resemble $\mathcal{G}(N.p)$
- However, the small world property is found in important cases, like social networks and citation networks
- Unlike the Poisson random graphs, these networks do not have high algebraic connectivity, and they tend to have high clustering coefficient.
- However, the exhibit the same $O(\log N)$ trend.

Small World random graphs



- We said that real networks hardly resemble $\mathcal{G}(N.p)$
- However, the small world property is found in important cases, like social networks and citation networks
- Unlike the Poisson random graphs, these networks do not have high algebraic connectivity, and they tend to have high clustering coefficient.
- However, the exhibit the same $O(\log N)$ trend.

Small World random graphs

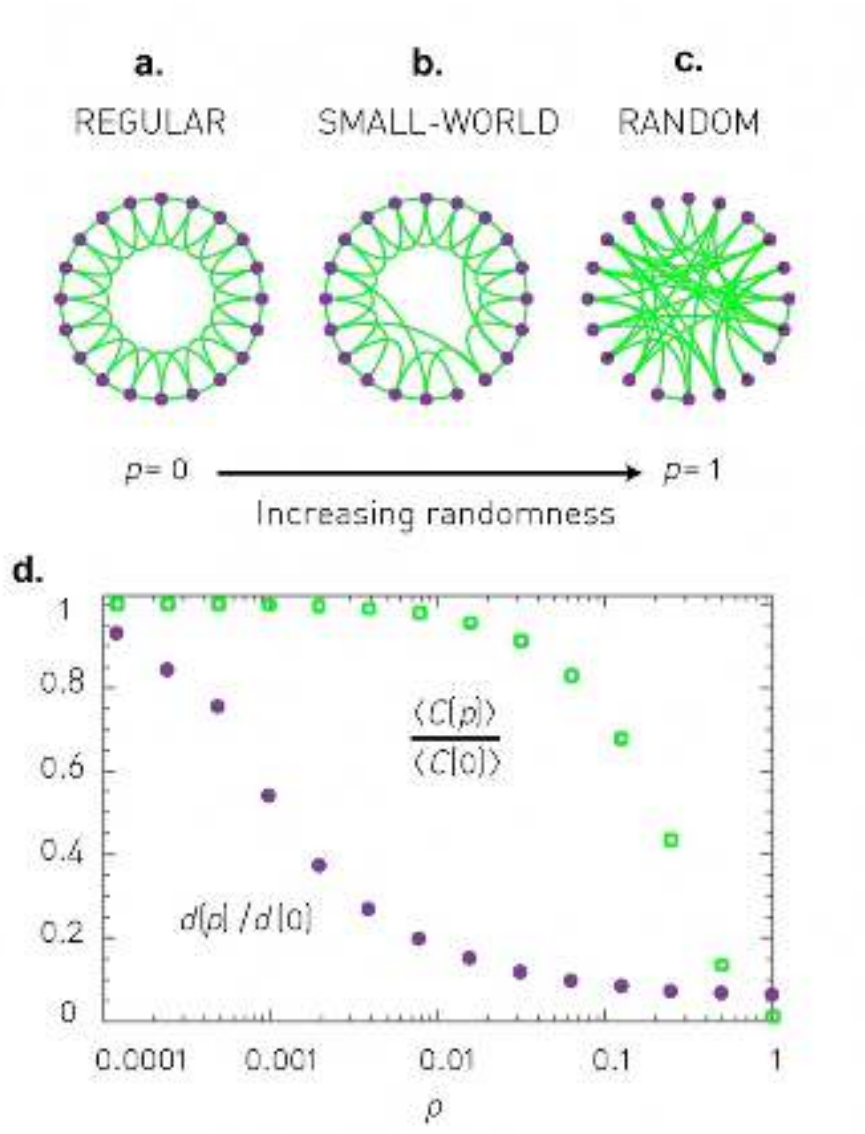
- The different features in the table provide the evidence

Network	N	L	‹k›	‹d›	d_{max}	lnN/ln‹k›
Internet	192,244	609,066	6.34	6.98	26	6.58
WWW	325,729	1,497,134	4.60	11.27	93	8.31
Power Grid	4,941	6,594	2.67	18.99	46	8.66
Mobile-Phone Calls	36,595	91,826	2.51	11.72	39	11.42
Email	57,194	103,731	1.81	5.88	18	18.4
Science Collaboration	23,133	93,437	8.08	5.35	15	4.81
Actor Network	702,388	29,397,908	83.71	3.91	14	3.04
Citation Network	449,673	4,707,958	10.43	11.21	42	5.55
E. Coli Metabolism	1,039	5,802	5.58	2.98	8	4.04
Protein Interactions	2,018	2,930	2.90	5.61	14	7.14

In this figure (from Barabasi's book, so it is not consistent with my notation)
$N := |\mathcal{V}|, L := |\mathcal{E}|, < k >:= \mathbb{E}$[node degree], $< d >:= \mathbb{E}$[distance between u and v]

Small World Watts and Strogatz model

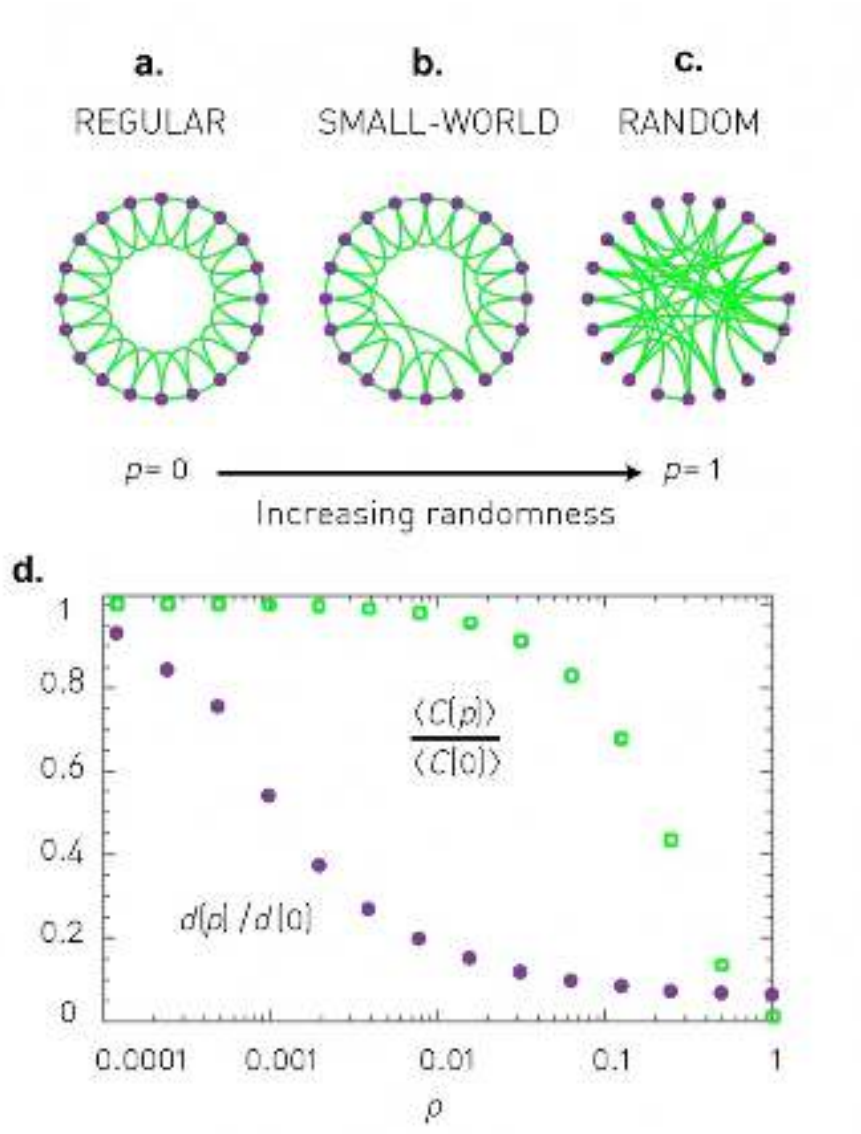
- While there are better models to fit real networks, WS small world random graph is nice to make the point
- It starts from a regular lattice, with clustering coefficient $C[0]$. It has a certain **rewiring** probability p that reassign the edges uniformly at random
- The interesting observation is that also in this networks the distance is $O(\log N)$ like in $\mathcal{G}(N,p)$



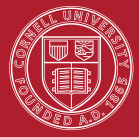
Small World Watts and Strogatz model

Cornell University

- While there are better models to fit real networks, WS small world random graph is nice to make the point
- It starts from a regular lattice, with clustering coefficient $C[0]$. It has a certain **rewiring** probability p that reassign the edges uniformly at random
- The interesting observation is that also in this networks the distance is $O(\log N)$ like in $\mathcal{G}(N,p)$

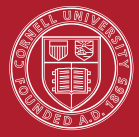


Outline



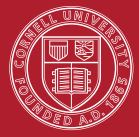
Cornell University

Time-variations in the graph



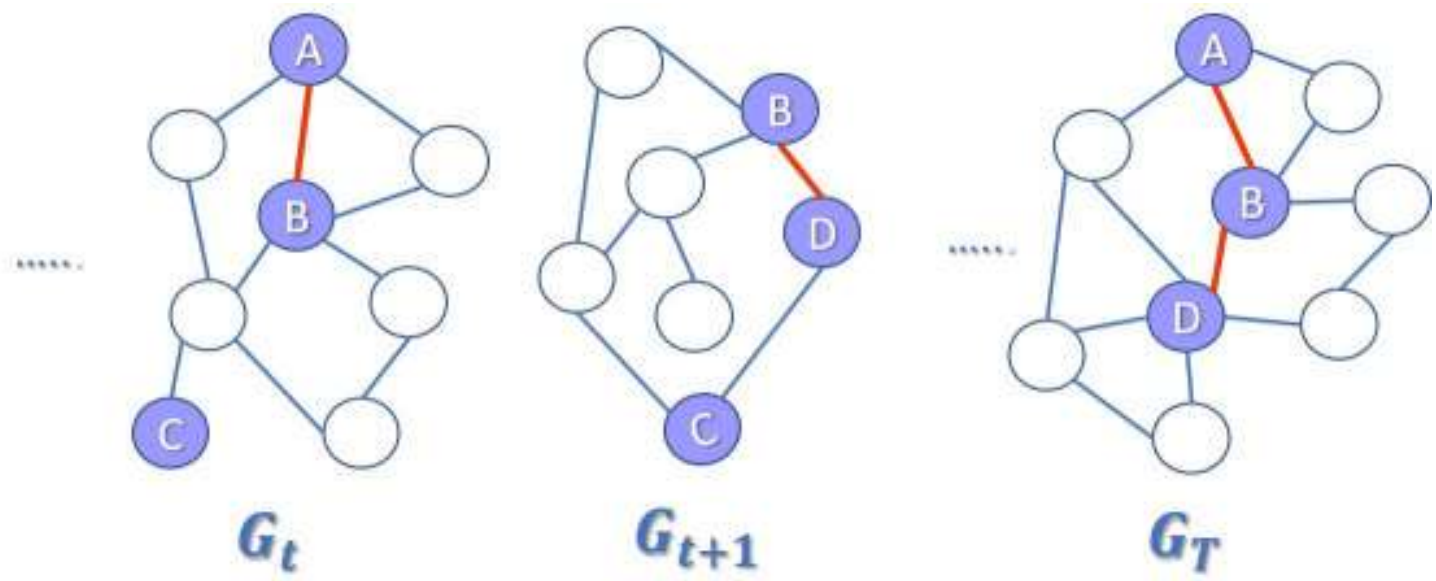
- In general there are many real scenarios where the network is dynamic $\mathcal{G}(t) = \{\mathcal{V}(t), \mathcal{E}(t)\}$
- For instance, friendships evolve in time. Webpages or web links are added or deleted
- Often the faster dynamics are on the edges $\mathcal{E}(t)$
- Formally, fixing $\mathcal{V}(t) = \mathcal{V}$ we can describe this as a multiplex graph (t is the index of the layer)
- How can we generate these models? There are many options

Time-variations in the graph



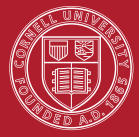
- In general there are many real scenarios where the network is dynamic $\mathcal{G}(t) = \{\mathcal{V}(t), \mathcal{E}(t)\}$
- For instance, friendships evolve in time. Webpages or web links are added or deleted
- Often the faster dynamics are on the edges $\mathcal{E}(t)$
- Formally, fixing $\mathcal{V}(t) = \mathcal{V}$ we can describe this as a multiplex graph (t is the index of the layer)
- How can we generate these models? There are many options

Edge Switching Process



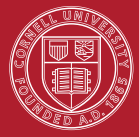
- The model must capture how the edges go on and off at random
- This is a stochastic process
- Most models often assume each edge $e(t)$ is an independent stochastic process, i.e. all edges are independently switching

Edge Switching Process



- The simplest model is that the edges are not only independently switching, but such that $e(t)$ for each t is an independent Bernoulli random variable
- This is useful when modeling random independent failures
- This is not always very realistic, and many models assume that there is correlation among the consecutive values

Edge Switching Process



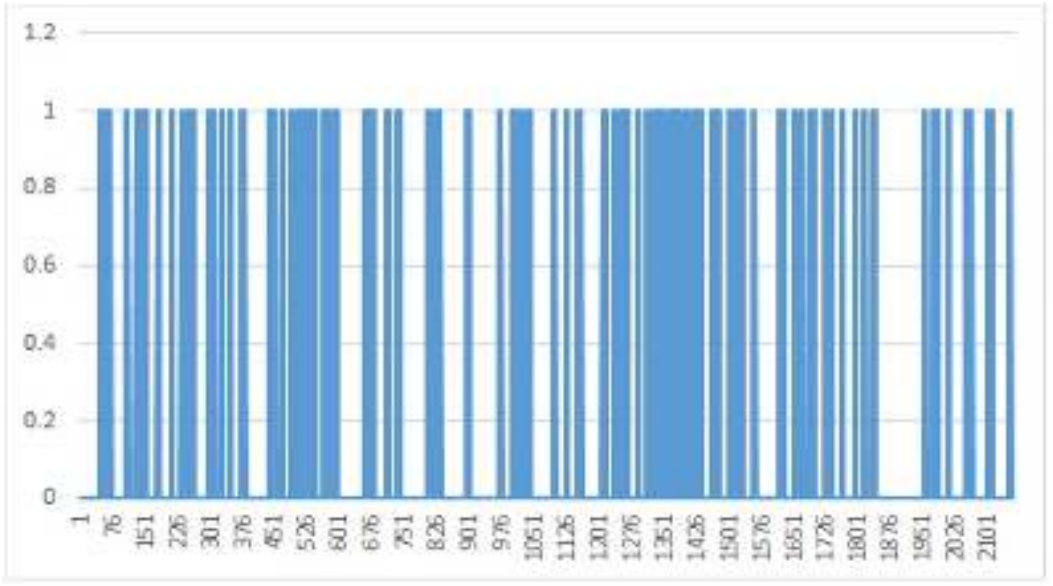
- The simplest model is that the edges are not only independently switching, but such that $e(t)$ for each t is an independent Bernoulli random variable
- This is useful when modeling random independent failures
- This is not always very realistic, and many models assume that there is correlation among the consecutive values

Edge Switching Process

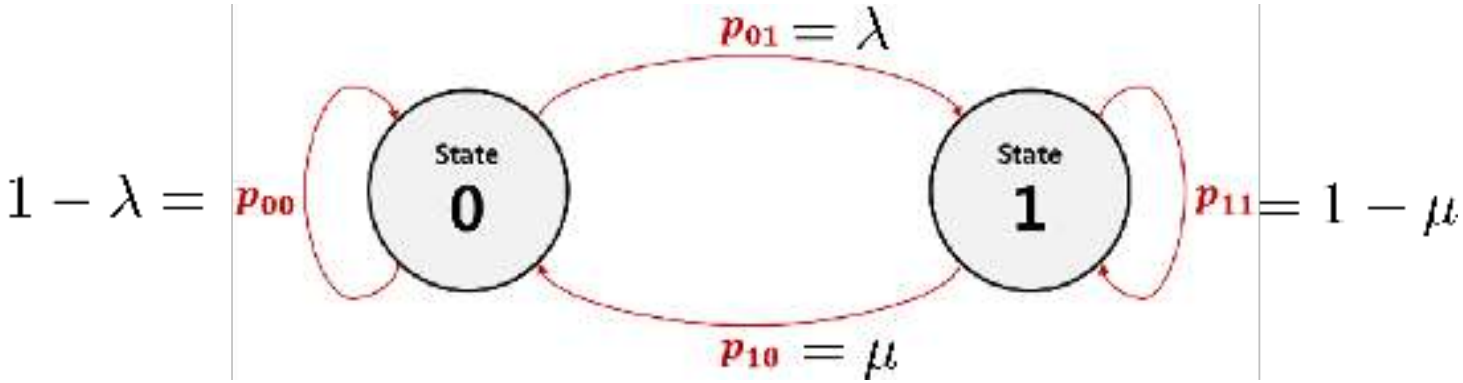
- The simplest model is that the edges are not only independently switching, but such that $e(t)$ for each t is an independent Bernoulli random variable
- This is useful when modeling random independent failures
- This is not always very realistic, and many models assume that there is correlation among the consecutive values

Edge Switching Process

- Basically, we want to model something similar to this figure, where the edge state remains the same $e(t) = e(t-1) = \ldots = e(t-T)$ for a certain period of time
- This period is random



Markov Switching Process



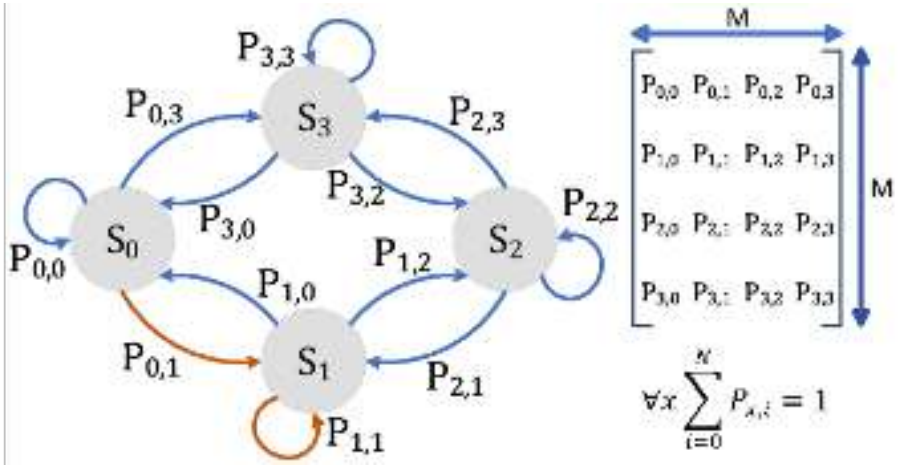
- The second simplest switching process is that edges come and go following independently a random process such that:

$$P(e(t) = 1 | e(t-1) = 0) = \lambda, \quad P(e(t) = 0 | e(t-1) = 1) = \mu$$

$$\Rightarrow \quad P(e(t) = 0 | e(t-1) = 0) = 1-\lambda, \quad P(e(t) = 1 | e(t-1) = 1) = 1-\mu$$

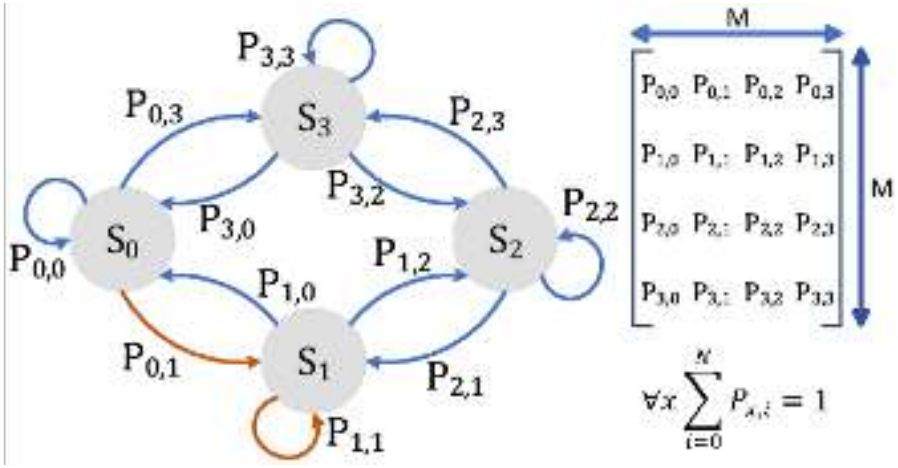
Transition Probability : $\boldsymbol{P} = \begin{bmatrix} p_{00} & p_{10} \\ p_{10} & p_{11} \end{bmatrix} = \begin{bmatrix} 1-\lambda & \lambda \\ \mu & 1-\mu \end{bmatrix}$

Markov Chains



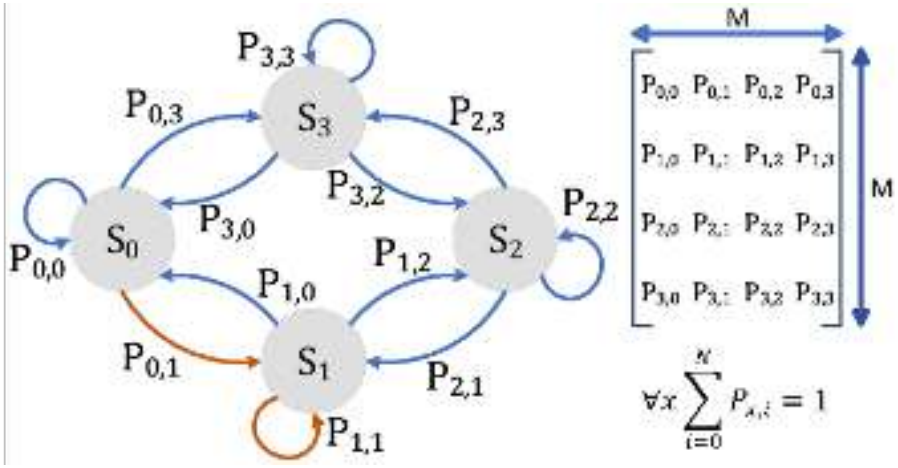
- This is the simplest example of a Markov chain (MC), (the symplest case of a graphical model)
- In general the MC has $\mathcal{S} = \{s_1, \ldots, s_n\}$ possible outcomes (called states)
- The probability of $S(t) = s_i$, given the previous state $S(t-1) = s_j$ is conditionally independent to all $S(t')$ for $t' < t - 1$

Markov Chains



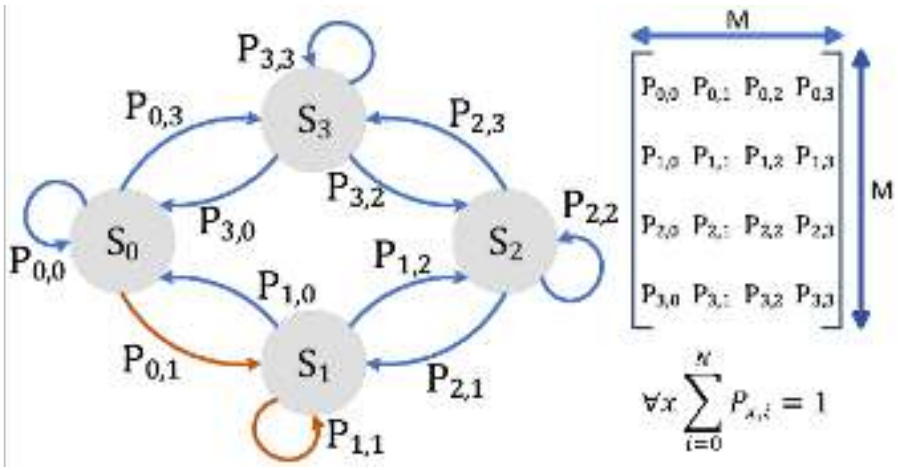
- This is the simplest example of a Markov chain (MC), (the symplest case of a graphical model)
- In general the MC has $\mathcal{S} = \{s_1, \ldots, s_n\}$ possible outcomes (called states)
- The probability of $S(t) = s_i$, given the previous state $S(t-1) = s_j$ is conditionally independent to all $S(t')$ for $t' < t - 1$

Markov Chains



- This is the simplest example of a Markov chain (MC), (the symplest case of a graphical model)
- In general the MC has $\mathcal{S} = \{s_1, \ldots, s_n\}$ possible outcomes (called states)
- The probability of $S(t) = s_i$, given the previous state $S(t-1) = s_j$ is conditionally independent to all $S(t')$ for $t' < t - 1$

Markov Chains paramters



- The probability of $P(S(t) = s_i | S(t-1) = s_j)$ is what is needed to define the entire stochastic process
- For an homogeneous MC, the conditional probability $P(S(t) = s_i | S(t-1) = s_j) = [\boldsymbol{P}]_{ij}$, is constant where $\boldsymbol{P}$ is, again, the Transition Probability

Markov Chains and edge switching

- Note, $\mathbf{1}^\top \boldsymbol{P} = \mathbf{1}^\top$, given that each column is a Probabilty Mass Function (PMF) which adds up to one.
- So P has a left eigenvector equal to $\mathbf{1}$ with eigenvalue $\lambda = 1$. Turns out, this is the largest one
- Using the theorem of total probability, one can show what are called *Chapman Kolmogorov* equations we can derive dynamics of the PMF of a state at time t:

$$[\boldsymbol{\pi}(t)]_i := P(S(t) = s_i), \quad \Rightarrow \quad \boldsymbol{\pi}(t) = \boldsymbol{P}\boldsymbol{\pi}(t-1)$$

- Using *Chapman Kolmogorov* equations repeatedly:

$$\boldsymbol{\pi}(t) = \boldsymbol{P}\boldsymbol{\pi}(t-1) = \boldsymbol{P}^2\boldsymbol{\pi}(t-2) = \ldots = \boldsymbol{P}^t\pi(0)$$

Markov Chains and edge switching

- Note, $\mathbf{1}^\top \boldsymbol{P} = \mathbf{1}^\top$, given that each column is a Probabilty Mass Function (PMF) which adds up to one.
- So $\boldsymbol{P}$ has a left eigenvector equal to $\mathbf{1}$ with eigenvalue $\lambda = 1$. Turns out, this is the largest one
- Using the theorem of total probability, one can show what are called *Chapman Kolmogorov* equations we can derive dynamics of the PMF of a state at time t:

$$[\boldsymbol{\pi}(t)]_i := P(S(t) = s_i), \quad \Rightarrow \quad \boldsymbol{\pi}(t) = \boldsymbol{P}\boldsymbol{\pi}(t-1)$$

- Using *Chapman Kolmogorov* equations repeatedly:

$$\boldsymbol{\pi}(t) = \boldsymbol{P}\boldsymbol{\pi}(t-1) = \boldsymbol{P}^2\boldsymbol{\pi}(t-2) = \ldots = \boldsymbol{P}^t\pi(0)$$

Markov Chains and edge switching

- Note, $\mathbf{1}^\top \boldsymbol{P} = \mathbf{1}^\top$, given that each column is a Probabilty Mass Function (PMF) which adds up to one.
- So $\boldsymbol{P}$ has a left eigenvector equal to $\mathbf{1}$ with eigenvalue $\lambda = 1$. Turns out, this is the largest one
- Using the theorem of total probability, one can show what are called *Chapman Kolmogorov* equations we can derive dynamics of the PMF of a state at time t:

$$[\boldsymbol{\pi}(t)]_i := P(S(t) = s_i), \quad \Rightarrow \quad \boldsymbol{\pi}(t) = \boldsymbol{P}\boldsymbol{\pi}(t-1)$$

- Using *Chapman Kolmogorov* equations repeatedly:

$$\boldsymbol{\pi}(t) = \boldsymbol{P}\boldsymbol{\pi}(t-1) = \boldsymbol{P}^2\boldsymbol{\pi}(t-2) = \ldots = \boldsymbol{P}^t\pi(0)$$

Markov Chains and edge switching

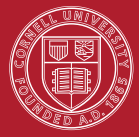
- Note, $\mathbf{1}^\top \boldsymbol{P} = \mathbf{1}^\top$, given that each column is a Probabilty Mass Function (PMF) which adds up to one.
- So $\boldsymbol{P}$ has a left eigenvector equal to $\mathbf{1}$ with eigenvalue $\lambda = 1$. Turns out, this is the largest one
- Using the theorem of total probability, one can show what are called *Chapman Kolmogorov* equations we can derive dynamics of the PMF of a state at time t:

$$[\boldsymbol{\pi}(t)]_i := P(S(t) = s_i), \quad \Rightarrow \quad \boldsymbol{\pi}(t) = \boldsymbol{P}\boldsymbol{\pi}(t-1)$$

- Using *Chapman Kolmogorov* equations repeatedly:

$$\boldsymbol{\pi}(t) = \boldsymbol{P}\boldsymbol{\pi}(t-1) = \boldsymbol{P}^2\boldsymbol{\pi}(t-2) = \ldots = \boldsymbol{P}^t\pi(0)$$

Asymptotic state probability



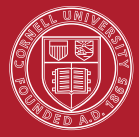
- Under some conditions on $\boldsymbol{P}$ (which we will delve into when defining the dynamics of random walks) $\boldsymbol{P}^t$ converges to a rank 1 matrix equal to:

$$\lim_{t\to+\infty} \boldsymbol{P}^t = \boldsymbol{\pi}^\infty \mathbf{1}^\top \quad \text{where} \quad \boldsymbol{\pi}^\infty = \boldsymbol{P}\boldsymbol{\pi}^\infty$$

- $\boldsymbol{\pi}^\infty$ is a fixed point for Chapman Kolmogorov equations
- it is also the right eigenvector of $\boldsymbol{P}$ associated with the eigenvalue $\lambda_1(\boldsymbol{P}) = 1$
- The limit above implies that the state probability asymptotically is equal to $\boldsymbol{\pi}^\infty$:

$$\lim_{t\to+\infty} \boldsymbol{\pi}(t) = \lim_{t\to+\infty} \boldsymbol{P}^t \boldsymbol{\pi}(0) = \boldsymbol{\pi}^\infty \overbrace{\mathbf{1}^\top \boldsymbol{\pi}(0)}^{=1} = \boldsymbol{\pi}^\infty$$

Asymptotic state probability



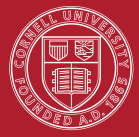
- Under some conditions on $\boldsymbol{P}$ (which we will delve into when defining the dynamics of random walks) $\boldsymbol{P}^t$ converges to a rank 1 matrix equal to:

$$\lim_{t\to+\infty} \boldsymbol{P}^t = \boldsymbol{\pi}^\infty \mathbf{1}^\top \quad \text{where} \quad \boldsymbol{\pi}^\infty = \boldsymbol{P}\boldsymbol{\pi}^\infty$$

- $\boldsymbol{\pi}^\infty$ is a fixed point for Chapman Kolmogorov equations
- it is also the right eigenvector of $\boldsymbol{P}$ associated with the eigenvalue $\lambda_1(\boldsymbol{P}) = 1$
- The limit above implies that the state probability asymptotically is equal to $\boldsymbol{\pi}^\infty$:

$$\lim_{t\to+\infty} \boldsymbol{\pi}(t) = \lim_{t\to+\infty} \boldsymbol{P}^t \boldsymbol{\pi}(0) = \boldsymbol{\pi}^\infty \overbrace{\mathbf{1}^\top \boldsymbol{\pi}(0)}^{=1} = \boldsymbol{\pi}^\infty$$

Asymptotic state probability



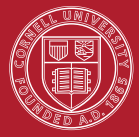
- Under some conditions on $\boldsymbol{P}$ (which we will delve into when defining the dynamics of random walks) $\boldsymbol{P}^t$ converges to a rank 1 matrix equal to:

$$\lim_{t\to+\infty} \boldsymbol{P}^t = \boldsymbol{\pi}^\infty \mathbf{1}^\top \quad \text{where} \quad \boldsymbol{\pi}^\infty = \boldsymbol{P}\boldsymbol{\pi}^\infty$$

- $\boldsymbol{\pi}^\infty$ is a fixed point for Chapman Kolmogorov equations
- it is also the right eigenvector of $\boldsymbol{P}$ associated with the eigenvalue $\lambda_1(\boldsymbol{P}) = 1$
- The limit above implies that the state probability asymptotically is equal to $\boldsymbol{\pi}^\infty$:

$$\lim_{t\to+\infty} \boldsymbol{\pi}(t) = \lim_{t\to+\infty} \boldsymbol{P}^t \boldsymbol{\pi}(0) = \boldsymbol{\pi}^\infty \overbrace{\mathbf{1}^\top \boldsymbol{\pi}(0)}^{=1} = \boldsymbol{\pi}^\infty$$

Asymptotic state probability



- Under some conditions on $\boldsymbol{P}$ (which we will delve into when defining the dynamics of random walks) $\boldsymbol{P}^t$ converges to a rank 1 matrix equal to:

$$\lim_{t\to+\infty} \boldsymbol{P}^t = \boldsymbol{\pi}^\infty \mathbf{1}^\top \quad \text{where} \quad \boldsymbol{\pi}^\infty = \boldsymbol{P}\boldsymbol{\pi}^\infty$$

- $\boldsymbol{\pi}^\infty$ is a fixed point for Chapman Kolmogorov equations
- it is also the right eigenvector of $\boldsymbol{P}$ associated with the eigenvalue $\lambda_1(\boldsymbol{P}) = 1$
- The limit above implies that the state probability asymptotically is equal to $\boldsymbol{\pi}^\infty$:

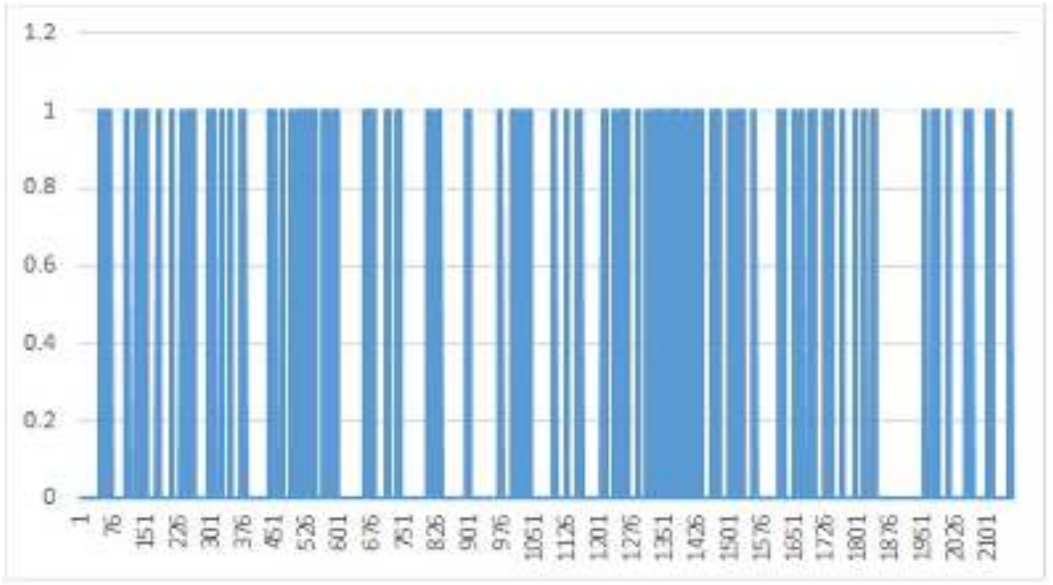
$$\lim_{t\to+\infty} \boldsymbol{\pi}(t) = \lim_{t\to+\infty} \boldsymbol{P}^t \boldsymbol{\pi}(0) = \boldsymbol{\pi}^\infty \overbrace{\mathbf{1}^\top \boldsymbol{\pi}(0)}^{=1} = \boldsymbol{\pi}^\infty$$

Limit of the edge switching process

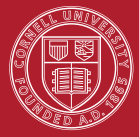
- For the edge switching process,

$$\boldsymbol{\pi}^{\infty} = \begin{bmatrix} \frac{\mu}{\lambda+\mu} \\ \frac{\lambda}{\lambda+\mu} \end{bmatrix}$$

- Note that the expected time the edge is off is $\frac{1}{\lambda}$ while the expected time it stays on is $1/\mu$



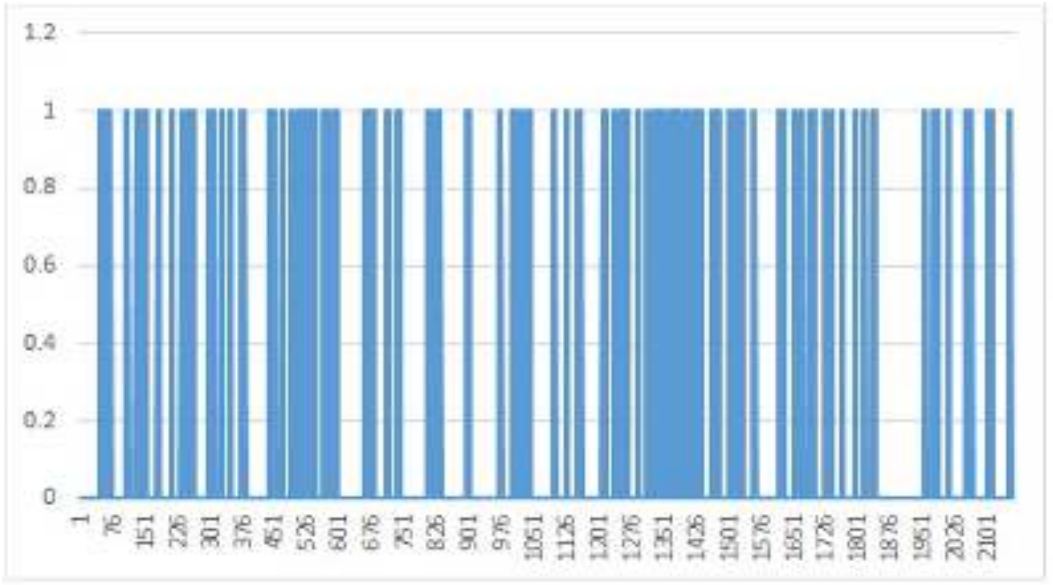
Limit of the edge switching process



- For the edge switching process,

$$\boldsymbol{\pi}^{\infty} = \begin{bmatrix} \frac{\mu}{\lambda+\mu} \\ \frac{\lambda}{\lambda+\mu} \end{bmatrix}$$

- Note that the expected time the edge is off is $\frac{1}{\lambda}$ while the expected time it stays on is $1/\mu$

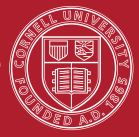


Edge switching process of Chung Lu type

- Nothing prevents from matching empirical observations that suggest that an edge has its own specific unique transition probabiliities λ_{uv} and μ_{ub}
- For the generalization of Chung Lu random graph model, we fix the turnover rate μ, and change λ_{uv} in order to ensure that each $\mathcal{G}(t)$ has the prescribed average $\mathbb{E}[d_u(t)] = c_u$, which implies that for every $\mathcal{G}(t)$:

$$p_{uv}(t) := \text{Probability}(e(t) = (u, v)) = \frac{c_u c_v}{2M}$$

Edge switching process of Chung Lu type

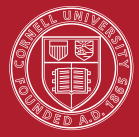


- In a dynamic setting, it can be shown that the probability that an edge appears is $p_{uv}(t) = \lambda_{uv}/\mu$
- Hence requires that:

$$p_{uv}(t) = \lambda_{uv}/\mu \equiv \frac{c_u c_v}{2M}$$

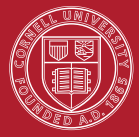
- This leaves a single parameter μ in addition to the average degree sequence c_v to define to construct the model

Outline



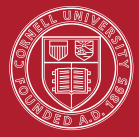
Cornell University

Synthetic systems



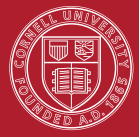
- Focusing only on matching topological features can become a rabbit hole, which misses the point of the exercise: gaining insight from the model
- When wanting a model to assess, for example, infrastructure vulnerabilities, these approaches have significant limitations
- One needs to create samples that have similar operational results as well, including physical parameters that define them

Synthetic systems



- Focusing only on matching topological features can become a rabbit hole, which misses the point of the exercise: gaining insight from the model
- When wanting a model to assess, for example, infrastructure vulnerabilities, these approaches have significant limitations
- One needs to create samples that have similar operational results as well, including physical parameters that define them

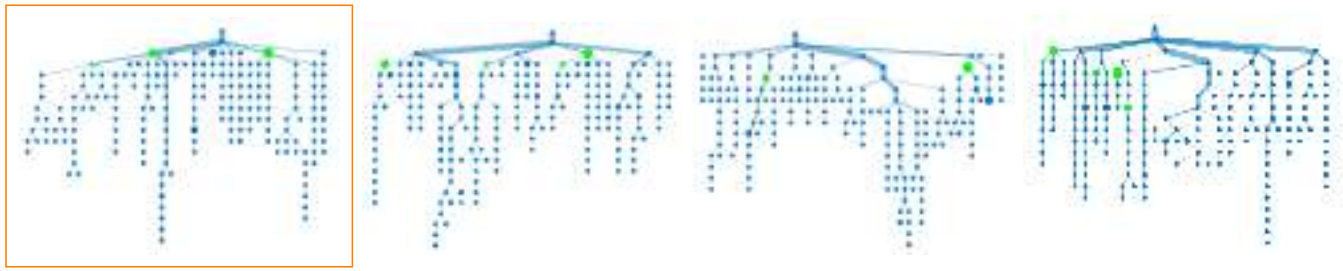
Synthetic systems



- Focusing only on matching topological features can become a rabbit hole, which misses the point of the exercise: gaining insight from the model
- When wanting a model to assess, for example, infrastructure vulnerabilities, these approaches have significant limitations
- One needs to create samples that have similar operational results as well, including physical parameters that define them

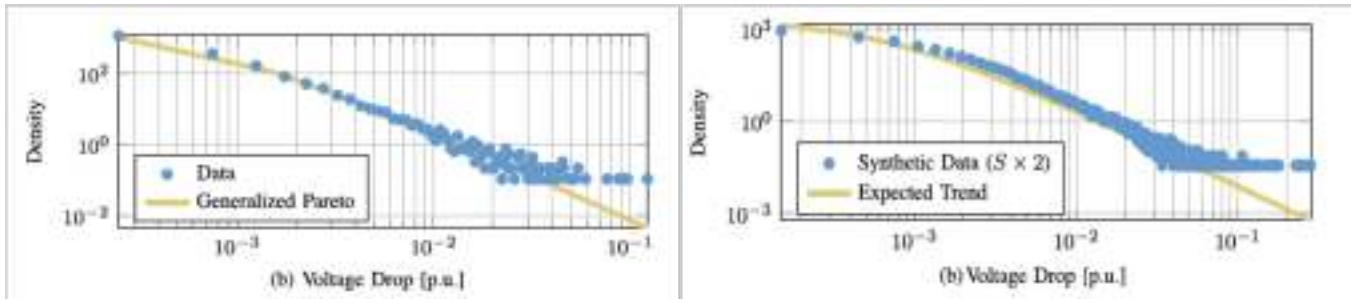
Synthetic systems

- New research on power grids, for instance, has shown how to produce samples that topologically and operationally are realistic



(a) (b) (c) (d)

The width of each line represents the relative power flow magnitude. Edges with reverse flow are marked in red. The size of each node represents the magnitude of real load/injection. Injection nodes are identified with green. Feeder (a) with the frame around it is taken directly from the data, the other three are synthetically generated by the algorithm.



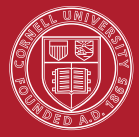
- This approach is practical when the true data are not available, they are *big data* and one wants to perform Montecarlo studies.

Real network data

- For many infrastructure studies on water, energy, transportation there is a significant amount of data that are made publicly available through the Homeland Infrastructure Foundation-Level Data (HIFLD) https://hifld-geoplatform.opendata.arcgis.com

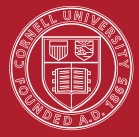
- For transportation OpenStreetMap is used most commonly, but HIFLD has other data beyond the GIS street information.

Outline



Cornell University

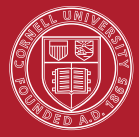
Conclusions



Cornell University

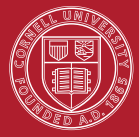
- We have covered a number of different random graphs
- The next time will mark the end of module 1
- We will discuss the phenomenon of *criticality* that relates to network connectivity for real and random networks.

Conclusions



- We have covered a number of different random graphs
- The next time will mark the end of module 1
- We will discuss the phenomenon of *criticality* that relates to network connectivity for real and random networks.

Conclusions



- We have covered a number of different random graphs
- The next time will mark the end of module 1
- We will discuss the phenomenon of *criticality* that relates to network connectivity for real and random networks.