Networks

Computational Social Systems 1 (VU) (706.616)

Denis Helic

ISDS, TU Graz

March 28, 2020

Outline

Graphs & Networks

Definition

A network is a set of items called nodes and connections between those items called links.

Terminology clarification:

- Mathematics: vertices (vertex) and edges
- Physics: sites and bonds
- Sociology: actors and ties
- **Computer science: nodes and links**

Graphs & Networks

Definition

A graph (network) is a pair of sets $G = (V, E)$, whereas V denotes the set of nodes and E the set of links.

- In an *undirected* graph, the set $E \subseteq [V]^2$
- $[V]^k$ is the set of all subsets of V with k elements
- In an undirected graph links are pairs of nodes
- In a *directed* graph, the set $E \subseteq V \times V$
- In a directed graph, links are ordered pairs of nodes
- In graph theory literature often $V(G)$ and $E(G)$ are used.

Example of a simple undirected graph

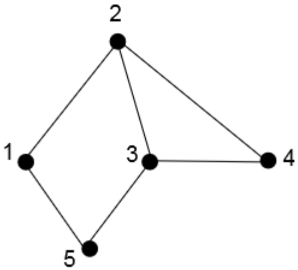


Figure: Simple undirected graph

- $V = \{1, 2, 3, 4, 5\}$
- $E = \{\{1, 2\}, \{1, 5\}, \{2, 3\}, \{2, 4\}, \{3, 4\}, \{3, 5\}\}$

Example of a simple directed graph

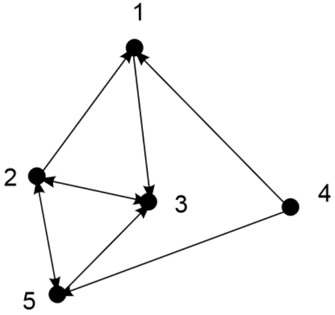


Figure: Simple directed graph

- $V = \{1, 2, 3, 4, 5\}$
- $E = \{(1,3), (2,1), (2,3), (2,5), (3,2), (4,1), (4,5), (5,2), (5,3)\}$

Some further notation

- Simple graphs: graphs with no self-links or loops
- $\forall i \in V$, $\{i\} \notin E$ (undirected graph). By defining that $E \subseteq [V]^2$ this is never the case.
- $\forall i \in V$, $(i, i) \notin E$ (directed graph)
- Number of nodes in G: $n = |V|$
- Number of links in G: $m = |E|$

Graphs vs. Networks

- Mathematical graph theory
- Analytical approach to studying of small graphs (typically tens or hundreds of nodes)
- With the emergence of ICT technology we are able to analyze large graphs that exist in nature, societies, technologies, etc.
- Now, we are considering large-scale statistical properties of graphs
- Network science deal with the empirical analysis of large graphs (networks) that occur in different areas

Types of networks

- Nodes connected by links is the simplest type of network
- Different types of nodes
- Different types of links
- Nodes and links can carry weights

Types of networks

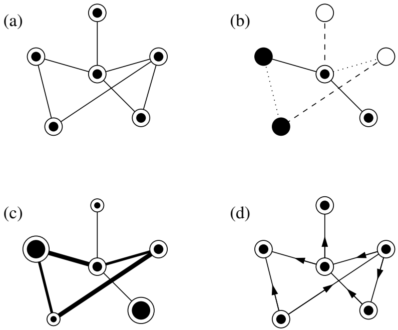


Figure: Various types of networks. From: The structure and function of complex networks, Newman, 2003.

Networks

- *Social networks*. Nodes are people and links are acquaintances, friendship, and so on.
- *Communication networks*. Internet: nodes are computers and links are cables connecting computers
- *Biological networks*. Metabolism: nodes are substances and links are metabolic reactions
- *Information networks*. Web: nodes are Web pages and links are hyperlinks connecting pages

Networks

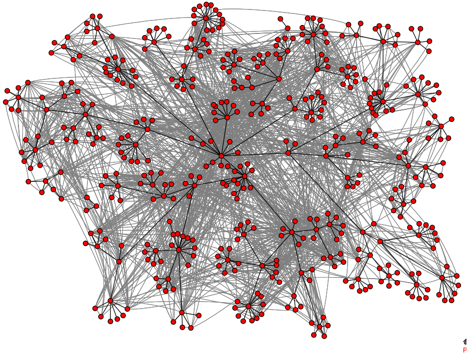


Figure: Social network of HP Labs constructed out of e-mail communication. From: How to search a social network, Adamic, 2005.

Networks

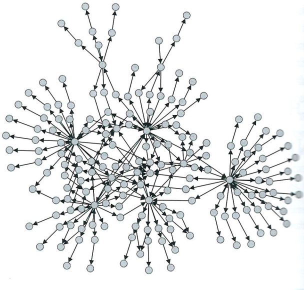


Figure: Network of pages and hyperlinks on a Website. From: Networks, Mark Newman, 2011.

Arpanet

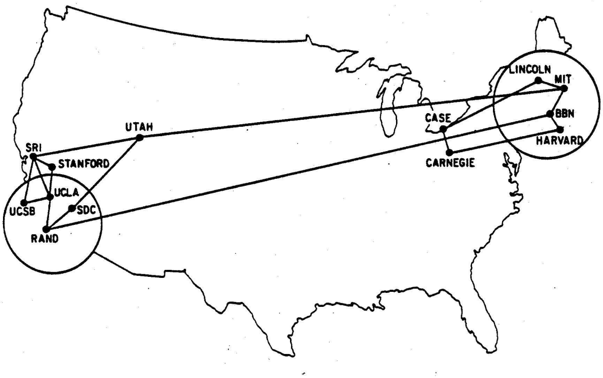
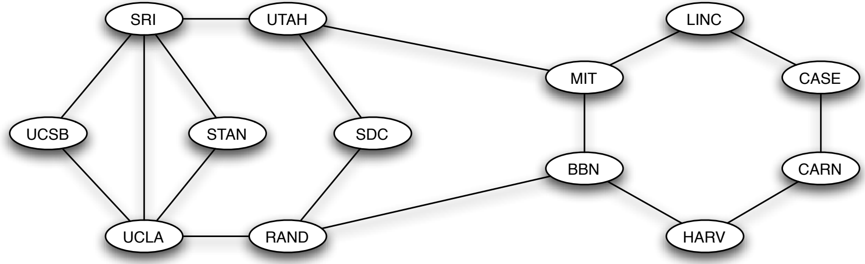


Figure: Image from:
http://som.csudh.edu/cis/lpress/history/arpamaps/

Arpanet

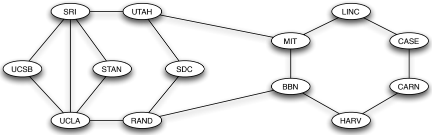


Paths

- Often things travel across the links of a graph
- A passenger taking a sequence of airline flights
- A computer user navigating the Web, or Wikipedia
- A data packet moving across the computer network, e.g. the Internet

Paths

- **Path**: a sequence of nodes such that each consecutive pair in the sequence is connected by a link
- For example, the sequence: (MIT, BBN, RAND, UCLA) is a path in the Internet graph
- Another sequence: (CASE, LINC, MIT, UTAH, SRI, UCSB) is also a path
- But the sequence: LINC, BBN, HARV, CARN is not a path

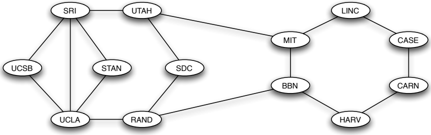


Paths Formally

Definition

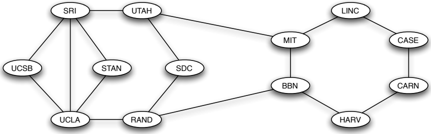
Let $G = (V, E)$ be a graph. Given two nodes $s, t \in V$ we define $\pi_{s,t} = (s, u_1, u_2, \ldots, u_{l-1}, t)$ to be a path between s and t if $\{u_1, u_2, \ldots, u_{l-1}\} \subset V$ and $\{(s, u_1), (u_1, u_2), \ldots, (u_{l-1}, t)\} \subset E$. Let $\Pi_{s,t}$ be a set of all paths from s to t.

- $\pi_{SRI,UCLA} = (SRI, UCLA)$ because $\{\{SRI, UCLA\}\} \subset E$
- $\pi_{SRI,UCLA} = (SRI, STAN, UCLA)$ because $\{\{SRI, STAN\}, \{STAN, UCLA\}\} \subset E$



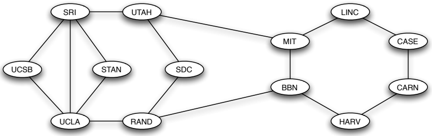
Paths

- We can repeat nodes in a path
- For example, the sequence: (SRI, STAN, UCLA, SRI, UTAH, MIT) is a path
- SRI is repeated
- If a path does not repeat nodes: **simple path**



Cycles

- An important kind of nonsimple path is a cycle
- **Cycle**: is a path with at least three links, in which the first and the last node are the same
- For example, (SRI, STAN, UCLA, SRI) is a cycle
- By design, every link belongs to a cycle to make it robust to failure (alternative routes)



Path length

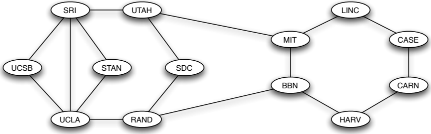
- Very often we want to know how **long** a path is
- In transportation and communication network it is important how many hops a packet or a person travels
- **Path length**: the number of links in a path

Definition

Let $G = (V, E)$ be a graph. Given two nodes $s, t \in V$ and a path $\pi_{s,t} = (s, u_1, u_2, \ldots, u_{l-1}, t)$ from s to t. We define the length of path $\pi_{s,t}$ as $|\pi_{s,t}| = l$.

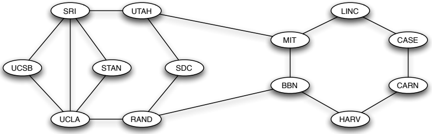
Path length

- (MIT, BBN, RAND, UCL) has length 3; (MIT, UTAH) has length 1



Distance

- **Distance**: the length of the shortest path between two nodes s and t. We denote the distance with $\ell_{s,t}$.
- In other words: $\ell_{s,t} \leq |\pi_{s,t}|$ for all paths $\pi_{s,t} \in \Pi_{s,t}$
- LINC and SRI have distance 3, i.e. $\ell_{LINC,SRI} = 3$
- UTAH and RAND have distance 2, i.e. $\ell_{UTAH,RAND} = 2$
- UTAH and SRI have distance 1, i.e. $\ell_{UTAH,SRI} = 1$



Breadth-First Search

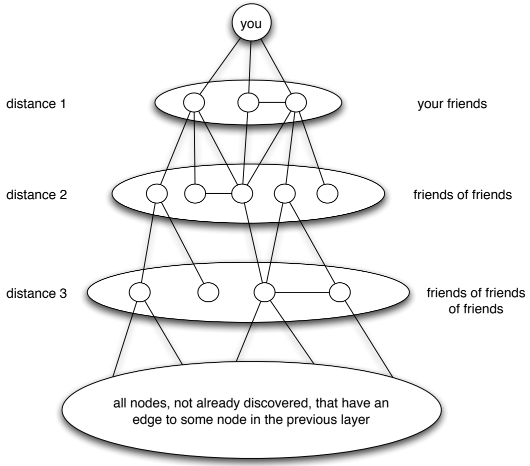
- For a small graph we can figure out the distance by looking at the picture
- For larger graphs we need an algorithm
- An efficient algorithm is **breadth-first search**
- The algorithm computes the distances from a single starting node to all other nodes
- From now on we assume that starting from an arbitrary node we can always reach all other nodes

Breadth-First Search

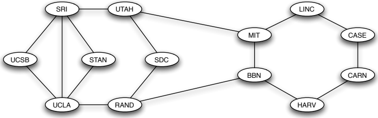
- We begin at a given node i in the network
  1. We declare all neighbors of i (nodes connected to i) to be at distance 1
  2. Then we find all neighbors of these neighbors (not counting nodes that are already neighbors of i) and declare them to be at distance 2
  3. Then we find all neighbors of the nodes from the previous step (again, not counting nodes that we already found at distance 1 and 2) and declare them to be at distance 3

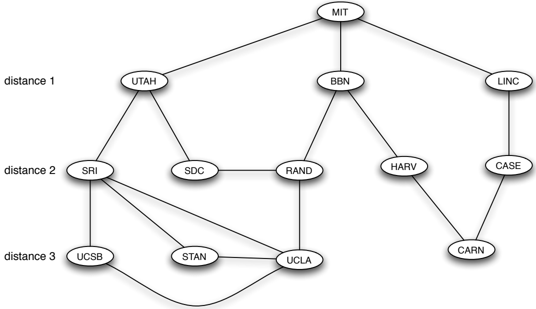
  (...) We continue in this way and search in successive layers each of which is at the next distance out until we can not discover any new nodes

Breadth-First Search



Breadth-First Search





Complexity of Breadth-First Search

- Recollect that we denote the number of nodes in a graph with n and a number of links with m
- During a breadth-first search we have to investigate all nodes at least once and follow all of their links at least once
- Thus, we perform $n + m$ operations
- Complexity of the breadth-first search algorithm is $O(n + m)$

Complexity of Breadth-First Search

- Using breadth-first search we can compute the distances between all pairs of nodes in a network (all-pairs-shortest-path)
- We iterate over the nodes and start a BFS from each node
- The complexity is $O(n(n+m)) = O(n^2 + nm)$

Complexity of Breadth-First Search

- Using breadth-first search we can compute the distances between all pairs of nodes in a network (all-pairs-shortest-path)
- We iterate over the nodes and start a BFS from each node
- The complexity is $O(n(n+m)) = O(n^2 + nm)$
- In a connected simple graph without selflinks: $(n-1) \leq m \leq \frac{n(n-1)}{2}$
- The overall complexity $O(nm)$

Summarizing distances

- One interesting quantity with respect to distances is the **diameter**
- **Diameter**: maximum distance between any pair of nodes in the graph (we denote it with ℓ_{max})
- Another interesting quantity is the **average distance**
- **Average distance** over all pairs of nodes in a graph:

$$\bar{\ell} = \frac{1}{n(n-1)} \sum_{ij} \ell_{ij}$$

Summarizing distances

- In many networks diameter and average distance are close to each other
- In some graphs, however, they can be very different
- Can you think of a graph where the diameter is three (or arbitrary many) times longer than the average distance

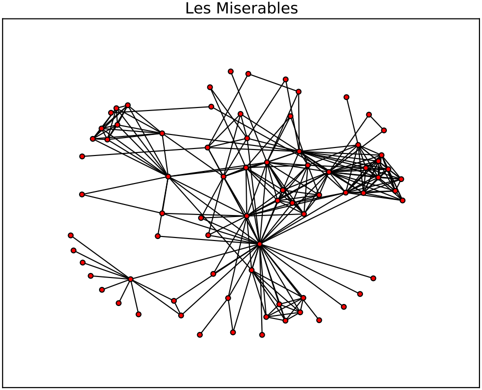
Summarizing distances

- In many networks diameter and average distance are close to each other
- In some graphs, however, they can be very different
- Can you think of a graph where the diameter is three (or arbitrary many) times longer than the average distance
- You need outliers in the distribution, i.e. a distant node connected by a chain of nodes to a tightly connected graph core

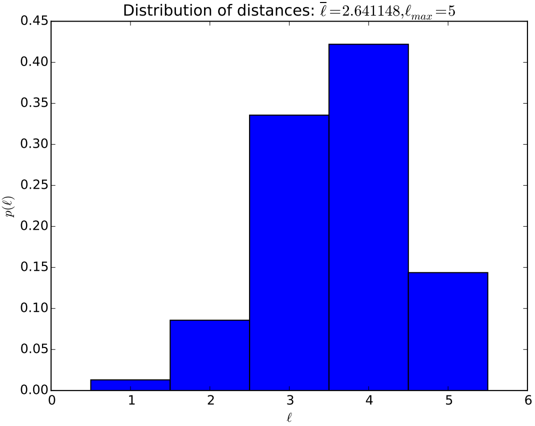
Distribution of distances

- Interesting statistics: distribution of distances
- How many pairs have a given distance
- Typically, we will normalize by the total number of pairs to obtain probabilities
- We can visualize it with a histogram

Les Miserables



Distribution of distances



Ipython notebook

- IPython Notebook example
- http://kti.tugraz.at/staff/socialcomputing/courses/webscience/websci1.zip

Command Line

ipython notebook –pylab=inline websci1.ipynb

Complexity of Breadth-First Search

- The overall complexity $O(nm)$
- If we have $m \sim n$ this is $O(n^2)$
- If $m \sim n^2$ this is $O(n^3)$
- However, if n is in the order of millions or billions both situations are prohibitive for breadth-first search
- We will need some method for approximating the distances
- The basic idea: estimate the distance bounds and make those bounds tight

Distance bounds

Definition

Let $SP_{s,t} \subseteq \Pi_{s,t}$ be the set of paths $\pi_{s,t}$ such that $|\pi_{s,t}| = \ell_{s,t}$.

- $SP_{s,t}$ is the set of shortest paths from s to t
- The shortest-path distance or just distance for short is a metric

Distance is a metric

Definition

A metric on a set X is a function $d : X \times X \rightarrow [0, \infty)$ and for all $x, y, z \in X$ the following conditions hold:

1. $d(x, y) \geq 0$
2. $d(x, y) = 0 \iff x = y$
3. $d(x, y) = d(y, x)$
4. $d(x, z) \leq d(x, y) + d(y, z)$ (Triangle inequality)

- The triangle inequality can be written as: $d(x, z) \geq |d(x, y) - d(y, z)|$

Distance bounds

- Given any three nodes s, t, and u

$$\ell_{s,t} \leq \ell_{s,u} + \ell_{u,t} \tag{1}$$

$$\ell_{s,t} \geq |\ell_{s,u} - \ell_{u,t}| \tag{2}$$

Distance bounds

Observation 1

Let $s, t, u \in V$. If there exist a path $\pi_{s,t} \in SP_{s,t}$ such that $u \in \pi_{s,t}$ then $\ell_{s,t} = \ell_{s,u} + \ell_{u,t}$.

Observation 2

Let $s, t, u \in V$. If there exist a path $\pi_{s,u} \in SP_{s,u}$ such that $t \in \pi_{s,u}$ or there exist a path $\pi_{t,u} \in SP_{t,u}$ such that $s \in \pi_{t,u}$ then $\ell_{s,t} = |\ell_{s,u} - \ell_{u,t}|$.

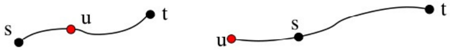


Figure: From "Fast Shortest Path Distance Estimation in Large Networks" by Potamias et al.

Landmarks

- We will use a set of landmarks $D = \{u_1, u_2, \dots u_d\}$
- Given a graph G and a set of d landmarks D we precompute the distances between each node in V and each landmarks
- We perform breadth-first search from all landmarks in $O(md)$
- d is small, e.g. $d \sim log(n)$

Landmarks

- Due to the triangle inequality we have:

$$\max_i |\ell_{s,u_i} - \ell_{t,u_i}| \leq \ell_{s,t} \leq \min_i \{\ell_{s,u_i} + \ell_{t,u_i}\} \tag{3}$$

- In other words, with $L = \max_i |\ell_{s,u_i} - \ell_{t,u_i}|$ and $U = \min_i \{\ell_{s,u_i} + \ell_{t,u_i}\}$ the true distance $\ell_{s,t} \in [L, U]$
- Estimation is very fast: $O(d)$

Landmarks

- Thus, if we have a “nice” set of landmarks D the approximation is very quick
- If we take U upper bound as our approximation following the Observation 1 this approximation is exact if there is a landmark in D that is on a shortest path from s to t
- If for all pairs of nodes from V there exist at least one landmark in D that lies on one shortest path from s to t then our approximation is exact
- In such case we say that landmarks *cover* all pairs of nodes from V

Landmark selection problem

LANDMARKS-COVER

Given a graph $G = (V, E)$ select the minimum number of landmarks $D \subseteq V$ such that all pairs of nodes $(s, t) \in V \times V$ are covered.

Theorem

*LANDMARKS-COVER is **NP**-hard.*

Landmark selection problem

NODE-COVER

Given a graph $G = (V, E)$ we say $V' \subseteq V$ covers V if every link has at least one endpoint in V'.

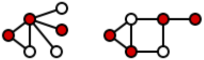


Figure: Node Cover (Source Wikipedia)

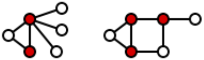


Figure: Minimal Node Cover (Source Wikipedia)

Landmark selection problem

Proof.

We reduce LANDMARKS-COVER to NODE-COVER by transforming an instance of NODE-COVER to LANDMARKS-COVER.

1. Consider a solution D to LANDMARKS-COVER. D covers all pairs of nodes and thus it covers also pairs at distance 1, which are connected by a single link. Therefore all links from E are covered by D and D is the solution to NODE-COVER.
2. Consider a solution V' to NODE-COVER. Some nodes from V' are on the links of the shortest path $\pi_{s,t}$ from s to t, and therefore V' is also a solution to LANDMARKS-COVER.

□

Landmark selection strategies

- We can not select the landmarks optimally so we have to select them using heuristics
- The basic idea: select "central" nodes, which lie on many shortest paths
- Baseline: random selection
- Select nodes with many links because the chance is higher that they are on many shortest paths
- Estimate average shortest path for each node and select the nodes with the smallest average

Landmark selection strategies

- We can not select the landmarks optimally so we have to select them using heuristics
- The basic idea: select “central” nodes, which lie on many shortest paths
- Baseline: random selection
- Select nodes with many links because the chance is higher that they are on many shortest paths
- Estimate average shortest path for each node and select the nodes with the smallest average
- Average path estimation: select randomly few nodes, perform BFS from those nodes, calculate averages to those nodes

Experimental results

- Datasets: Flickr-E $\sim 600K$ nodes, Flickr-I $\sim 800K$ nods, DBLP $\sim 220K$

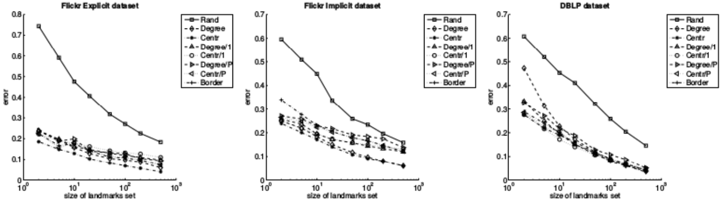
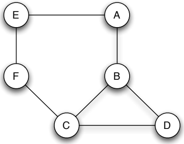


Figure: From "Fast Shortest Path Distance Estimation in Large Networks" by Potamias et al.

Pivotal nodes

- We say that a node k is pivotal for a pair of distinct nodes i and j if k lies on every shortest path between i and j
- k is not equal to either i and j
- B is pivotal for (A,C) and (A,D)
- However, it is not pivotal for (D,E)



Pivotal nodes

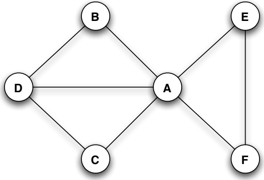
- Pivotal nodes play an important role in connecting other nodes
- Some nodes are more "important" than the other nodes
- Can you think of an example of a graph in which every node is pivotal for at least one pair of nodes

Pivotal nodes

- Pivotal nodes play an important role in connecting other nodes
- Some nodes are more "important" than the other nodes
- Can you think of an example of a graph in which every node is pivotal for at least one pair of nodes
- Can you think of an example of a graph in which every node is pivotal for at least two different pairs of nodes

Gatekeepers

- Similar to pivotal nodes is an idea that some nodes play a "gatekeeping" role in networks
- We say that a node k is a gatekeeper if, for some other distinct nodes i and j, k lies on every path between i and j
- k is not equal to either i and j
- A is a gatekeeper because it lies on every path between B and E, or D, and E

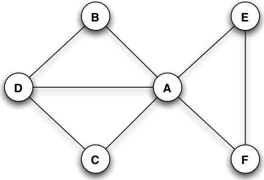


Gatekeepers

- The last definition has a “global” flavor
- We have to consider paths in the full graph to decide if a node is a gatekeeper
- We can think also about a “local” version of a gatekeeper
- We say that a node k is a local gatekeeper if it has two distinct neighbors i and j that are not connected to each other
- k is not equal to either i and j

Gatekeepers

- Node A is also a local gatekeeper, e.g. B and E are neighbors but they are not connected to each other
- Node D is a local gatekeeper for B and C but it is not a gatekeeper



Gatekeepers

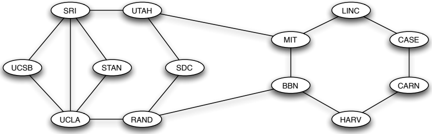
- Can you think of an example of a graph in which more than half of all nodes are gatekeepers

Gatekeepers

- Can you think of an example of a graph in which more than half of all nodes are gatekeepers
- Can you think of an example of a graph in which there are no gatekeepers but in which every node is a local gatekeeper

Connectivity

- Given a graph one important question is whether every node can reach every other node by a path
- If that is the case the graph is **connected**
- ARPANET is a connected graph, as it should be always the case with communication and transportation networks



Connectivity

- But, in e.g. a social network that is not always the case
- Then we say that a graph is **disconnected**

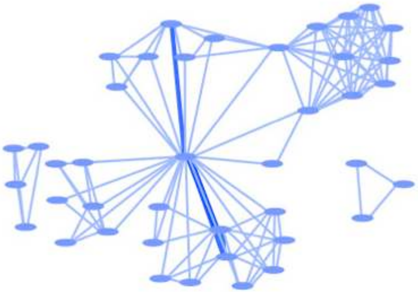
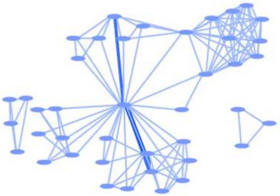


Figure: Collaboration graph of a biological research center

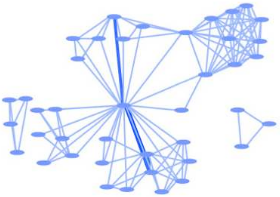
Components

- If a graph is disconnected than it breaks apart into a set of connected **components**
- **Component**: a subset of nodes such that
  1. every node in the subset has a path to every other node in that subset (internally connected)
  2. the subset is not a part of some larger connected set (stands in isolation from the rest of the graph)



Components

- Components are a first, global way of describing the structure of a network
- Within a given component there might be a richer structure
- The large component: a prominent node at the center and tightly linked groups at the periphery
- This large component would break apart without the central node



Giant components

- Consider the global friendship network, i.e. a social network of the entire world
- Is this network connected?
- Probably not, since a single person without friends constitutes a one-node component
- "Remote tropical island"

Giant components

- But, people (you) have friends in other countries
- You are in the same component as those friends
- As well as their friends, their parents, their parent friends, their descendants, and so on
- You are in the same component with the people that you never heard of, with totally different experiences, etc.
- This component seems likely to contain a significant fraction of the world's population, and this is in fact true!
- We call such a component a **giant component**

Giant components

- It is an informal definition: a component that contains a significant fraction of nodes
- Typically, when a network contains a giant component it contains almost always only one
- Why?

Giant components

- It is an informal definition: a component that contains a significant fraction of nodes
- Typically, when a network contains a giant component it contains almost always only one
- Why?
- If we have two giant components with e.g. 1 billion people in each
- It takes only a single link from a node from the first component to a node from the second to connect those two components
- Practically, such a link always exists

Degree

- **Degree**: of a node is the number of links connected to it
- Measures how "important" a node is
- We denote the degree of node i by k_i
- Every link has two ends, hence there are $2m$ link ends in an undirected network
- The number of link ends is equal to the sum of the degrees of all the nodes

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

- Average degree

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

Degree in directed networks

- In directed networks we have **in-degree** and **out-degree**
- **In-degree**: k_i^{in} is the number of ingoing links
- **Out-degree**: k_j^{out} is the number of outgoing links
- The number of links is equal to the sum of in-degrees, and is also equal to the sum of out-degrees

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

- Average in-degree and average out-degree

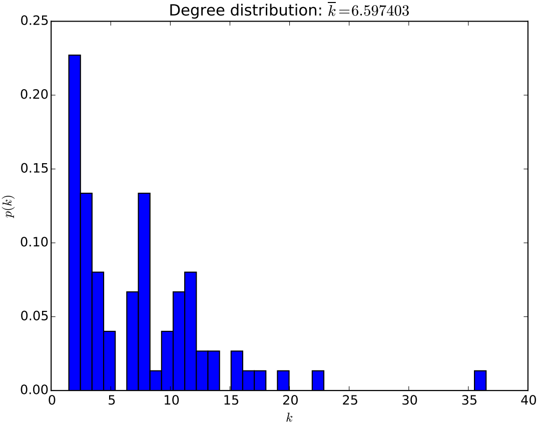
$$\overline{k^{in}} = \frac{1}{n} \sum_{i=1}^{n} {k_i}^{in} = \frac{1}{n} \sum_{i=1}^{n} {k_i}^{out} = \overline{k^{out}}$$

$$\bar{k} = \overline{k^{in}} = \overline{k^{out}} = \frac{m}{n}$$

Degree distribution

- Interesting statistics: degree distribution
- How many nodes have a given degree
- Typically, we will normalize by the total number of nodes to obtain probabilities
- We can visualize it with a histogram

Degree distribution



Ipython notebook

- IPython Notebook example
- http://kti.tugraz.at/staff/socialcomputing/courses/webscience/websci1.zip

Command Line

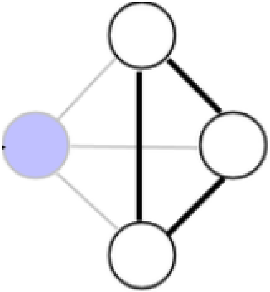
ipython notebook –pylab=inline websci1.ipynb

Clustering coefficient

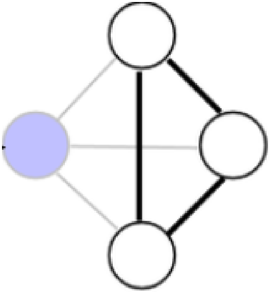
- A special kind of paths are triangles (closed triads)
- Three nodes all connected to each other
- Closely related to local gatekeepers
- Local clustering coefficient of a node is a measure how transitive connections in a network areas
- I.e. a friend of a friend is also a friend
- **Clustering coefficient**: fraction of node i neighbors that are themselves connected (we denote it with C_i, $\Gamma(i)$ is the set of neighbors of i, E is the set of all links)

$$C_i = \frac{2|e_{jk}|}{k_i(k_i - 1)}, j, k \in \Gamma(i), e_{jk} \in E$$

Clustering coefficient

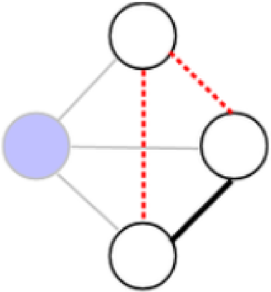


Clustering coefficient

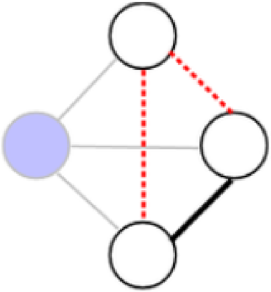


- $C_{gray} = 1$

Clustering coefficient

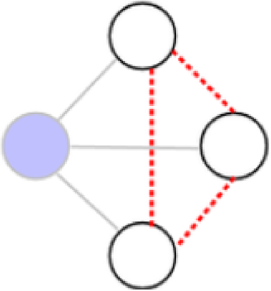


Clustering coefficient

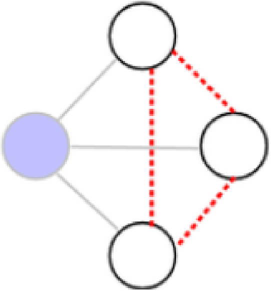


- $C_{gray} = \frac{1}{3}$

Clustering coefficient

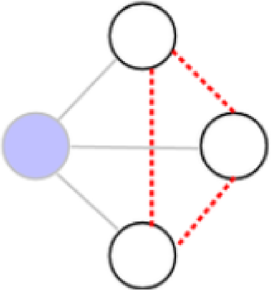


Clustering coefficient



- $C_{gray} = 0$
- What can you say about the clustering coefficient of a local gatekeeper?

Clustering coefficient



- $C_{gray} = 0$
- What can you say about the clustering coefficient of a local gatekeeper?
- It is less than 1

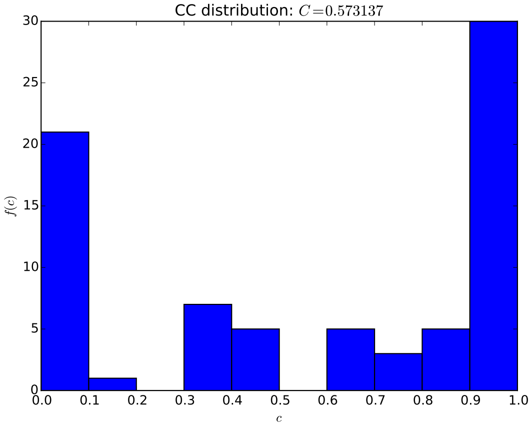
Clustering coefficient

- Some statistics: average clustering coefficient

$$C = \frac{1}{N} \sum_i C_i$$

- Clustering coefficient distribution
- How many nodes have a clustering coefficient in a certain range
- We can visualize it with a histogram

Clustering coefficient distribution



Ipython notebook

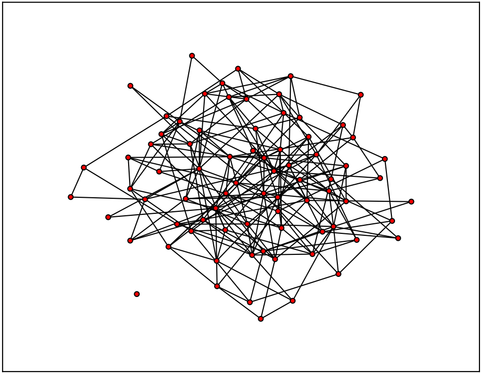
- IPython Notebook example
- http://kti.tugraz.at/staff/socialcomputing/courses/webscience/websci1.zip

Command Line

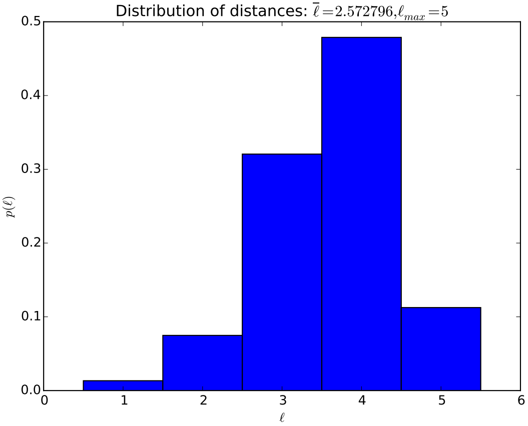
ipython notebook –pylab=inline websci1.ipynb

Les Miserables

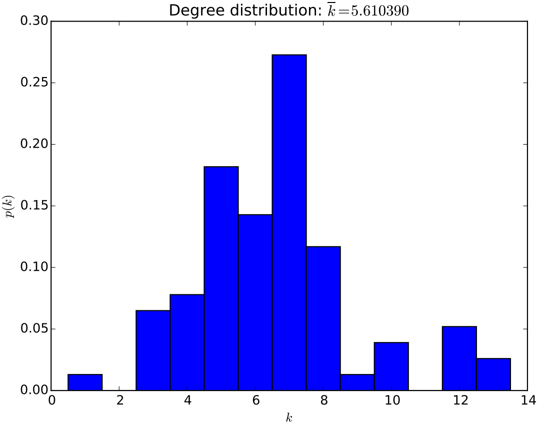
Random Graph $n = 77, m = 216$



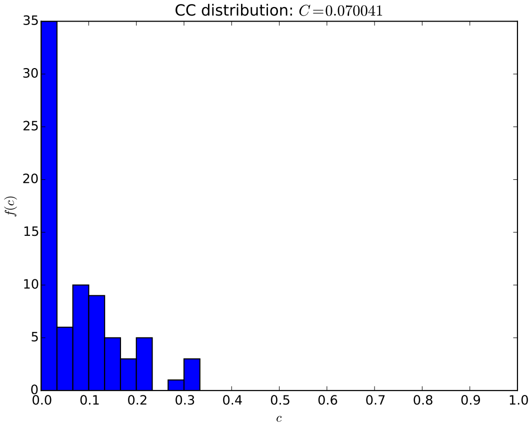
Distribution of distances



Degree distribution



Clustering coefficient distribution



Ipython notebook

- IPython Notebook example
- http://kti.tugraz.at/staff/socialcomputing/courses/webscience/websci1.zip

Command Line

ipython notebook –pylab=inline websci1.ipynb

Modeling networks

- Observe: e.g. collect data by crawling
- Measure: e.g. how many nodes, how many links
- Quantify: e.g. distances, degrees, clustering coefficient
- Make a model: e.g. how networks are created
- Predict with the model and validate: e.g. implement and evaluate (compare with the real networks)
- Apply: engineering approach to implementing the model in the software

Random graphs

- Random graph: a model network where some properties take fixed values and other properties are random
- The simplest model: we fix n and m but place links at random
- Iterate over links, for each link select a random pair of nodes
- Create a simple graph, i.e. no self-links are allowed
- We will call this model $G(n, m)$

$G(n, m)$ model

- Equivalent definition: we choose a graph uniformly at random among all graphs with n nodes and m links
- Mathematically, this is a proper definition
- A random graph defines an *ensemble* of networks, i.e. a probability distribution $P(G)$ over possible networks:
- $P(G) = \frac{1}{\Omega}$ if a network has n nodes and m links and 0 otherwise
- Ω is the total number of such networks

$G(n, m)$ model

- Equivalent definition: we choose a graph uniformly at random among all graphs with n nodes and m links
- Mathematically, this is a proper definition
- A random graph defines an *ensemble* of networks, i.e. a probability distribution $P(G)$ over possible networks:
- $P(G) = \frac{1}{\Omega}$ if a network has n nodes and m links and 0 otherwise
- Ω is the total number of such networks
- $\Omega = \binom{\binom{n}{2}}{m}$

Random graphs

- Properties of random graphs: mean values of the ensemble (typical behavior)
- Calculated as expectations of the probability distribution or a random variable

Definition

The expectation of a random property $X(G)$ of a random graph ensemble G is

$$E[X] = \sum_{G} X(G)p(G)$$

when this sum is "well-defined", otherwise the expectation does not exist.

$G(n,m)$ vs. $G(n,p)$ model

- Some mean values are easy to calculate, i.e. the average number of links is m
- Average degree $\bar{k} = \frac{2m}{n}$
- Other properties are more difficult to calculate
- A better approach is to fix n and p, which is the probability of links between nodes
- This model is $G(n,p)$

$G(n, p)$ model

- Technical definition is again in terms of an ensemble, i.e. a probability distribution over all possible networks
- What is the total number of possible simple graphs?

$G(n,p)$ model

- Technical definition is again in terms of an ensemble, i.e. a probability distribution over all possible networks
- What is the total number of possible simple graphs?
- $\Omega = 2^{\binom{n}{2}}$
- But the $P(G)$ is not uniform anymore, i.e. in general $P(G) \neq \frac{1}{\Omega}$
- Some graphs are more probable then other graphs in case of $G(n,p)$
- What is the probability of a graph that has exactly m links

$G(n,p)$ model

- Technical definition is again in terms of an ensemble, i.e. a probability distribution over all possible networks
- What is the total number of possible simple graphs?
- $\Omega = 2^{\binom{n}{2}}$
- But the $P(G)$ is not uniform anymore, i.e. in general $P(G) \neq \frac{1}{\Omega}$
- Some graphs are more probable then other graphs in case of $G(n,p)$
- What is the probability of a graph that has exactly m links
- $P(G) = p^m(1-p)^{\binom{n}{2}-m}$
- Other names for $G(n,p)$: Erdős–Rényi, Bernoulli, Poisson

Mean number of links

- Probability that a simple graph G has m links: $P(G) = p^m(1-p)^{\binom{n}{2}-m}$
- The number of graphs with n nodes and m links: $\binom{\binom{n}{2}}{m}$
- The total probability of drawing a graph with m links from the ensemble

$$P(m) = \binom{\binom{n}{2}}{m} p^m (1-p)^{\binom{n}{2}-m} \tag{4}$$

- This is binomial distribution
- The expected (mean) number of links:

$$E[m] = \sum_{m=0}^{\binom{n}{2}} mP(m) \tag{5}$$

Linearity of expectation

Theorem

Suppose X *and* Y *are discrete r.v. such that* $E[X] < \infty$ *and* $E[Y] < \infty$*. Then,*

- $E[aX] = aE[X], \forall a \in \mathbb{R}$
- $E[X + Y] = E[X] + E[Y]$

- Proof left for exercise ;)

Bernoulli random variable

PMF

$$p(x) = \begin{cases} 1 - p \text{ if } x = 0 \\ p \text{ if } x = 1 \end{cases}$$

- Bernoulli r.v. with parameter p
- Models situations with two outcomes
- E.g. coin flip

Binomial random variable

- Suppose $X_1, \ldots, X_n$ are independent and identical Bernoulli r.v.
- The Binomial r.v. with parameters (p, n) is

$$Y = X_1 + \cdots + X_n$$

- Models the number of successes in n Bernoulli trials
- E.g. the number of heads in n coin flips

PMF

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

Expectation: Bernoulli r.v.

PMF

$$p(x) = \begin{cases} 1 - p \text{ if } x = 0 \\ p \text{ if } x = 1 \end{cases}$$

- $E[X] = (1 - p) \cdot 0 + p \cdot 1 = p$

Expectation: Binomial r.v.

- Binomial is the sum of $X_1, \ldots, X_n$, independent and identical Bernoulli r.v.

$$Y = X_1 + \cdots + X_n$$

$$E[Y] = E[X_1 + \cdots + X_n] = E[X_1] + \cdots + E[X_n] = np$$

Mean number of links and mean degree

$$E[m] = \sum_{m=0}^{\binom{n}{2}} mP(m) = \binom{n}{2}p = \frac{n(n-1)}{2}p \tag{6}$$

$$\bar{k} = E[k] = \frac{2E[m]}{n} = \frac{2}{n}\frac{n(n-1)}{2}p = (n-1)p \tag{7}$$

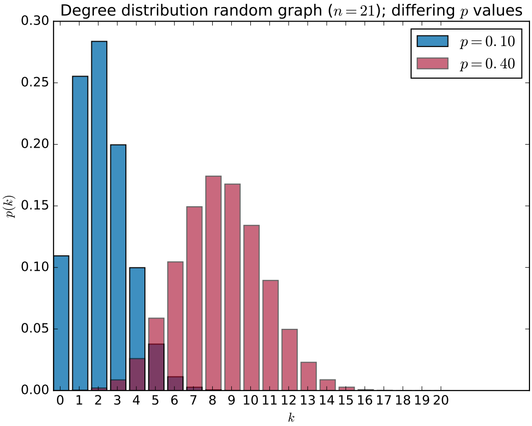
Degree distribution

- A given node is connected with independent probability p to $(n-1)$ other nodes
- Probability of being connected to exactly k other nodes: $p^k(1-p)^{n-1-k}$
- There are $\binom{n-1}{k}$ ways of selecting k nodes from $n-1$ nodes
- The total probability of having a degree k:

$$P(k) = \binom{n-1}{k} p^k (1-p)^{n-1-k} \tag{8}$$

- This is again binomial distribution
- Thus, $G(n,p)$ has a binomial degree distribution

Degree distribution (Binomial)



Degree distribution

- In many cases n is large, p is small and the average degree, i.e. $(n-1)p$ is constant
- Let us introduce $\lambda = (n-1)p$

$$
\begin{aligned}
P(k) &= \frac{(n-1)(n-2)\dots(n-k)}{k!}\frac{\lambda^k}{(n-1)^k}(1-\frac{\lambda}{n-1})^{n-1-k} \\
&= \frac{(n-1)(n-2)\dots(n-k)}{(n-1)^k}\frac{\lambda^k}{k!}\frac{(1-\frac{\lambda}{n-1})^{n-1}}{(1-\frac{\lambda}{n-1})^k}
\end{aligned}
$$

Degree distribution

- What is $\lim_{n \to \infty} P(k)$

$$\lim_{n \to \infty} \frac{(n-1)(n-2)\dots(n-k)}{(n-1)^k} = 1$$

$$\lim_{n \to \infty} (1 - \frac{\lambda}{n-1})^k = 1$$

$$\lim_{n \to \infty} (1 - \frac{\lambda}{n-1})^{n-1} = ?$$

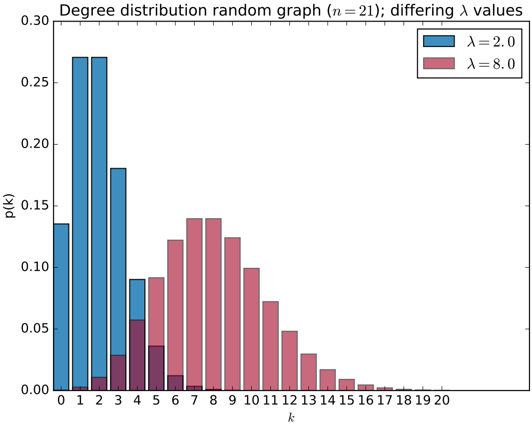
Degree distribution

- The definiton of e: $e = \lim_{x\to\infty}(1+\frac{1}{x})^x$
- Let us substitute: $n-1 = -x\lambda$

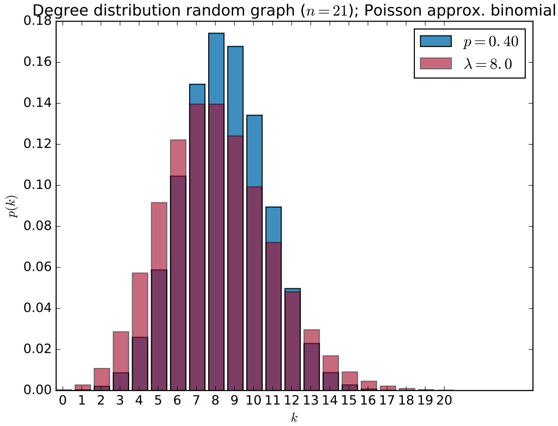
$$\begin{aligned}\lim_{n\to\infty}(1-\frac{\lambda}{n-1})^{n-1} &= \lim_{n\to\infty}(1+\frac{1}{x})^{-x\lambda} \\ &= \lim_{n\to\infty}((1+\frac{1}{x})^x)^{-\lambda} \\ &= e^{-\lambda}\end{aligned} \tag{9}$$

- Put it all together: $P(k) = \frac{\lambda^k}{k!}e^{-\lambda}$
- Poisson distribution

Degree distribution (Poisson)



Degree distribution (Poisson approximation)



Clustering coefficient

- Probability that two neighbors of a node are themselves connected
- What is this probability?

Clustering coefficient

- Probability that two neighbors of a node are themselves connected
- What is this probability?
- $C = p$

Giant component in $G(n, p)$

- What is the size of the largest component in $G(n, p)$?
- How does it relate to n and p?
- Is there a giant component (GC)?
- How many nodes does it occupy?

Giant component in $G(n, p)$

- Two special cases
- When $p = 0$ there are no links in the graph and we have n components of size 1
- The size of the largest component is constant and does not depend on n
- When $p = 1$ there are links between all pairs (complete graph) and the largest component is of size n
- There is a GC and its size depends on n, i.e. it is exactly n

Giant component in $G(n, p)$

- Two qualitatively different states (phases)
- What needs to happen with the random graph when we start with $p = 0$ and slowly increase p until we reach $p = 1$

Giant component in $G(n, p)$

- Two qualitatively different states (phases)
- What needs to happen with the random graph when we start with $p = 0$ and slowly increase p until we reach $p = 1$
- With increasing p the largest component will become bigger until it turns into a GC
- Until it reaches the size of n
- Transition between two extremes, a.k.a. phase transition
- There will be a critical value for p at which phase transition occurs

Giant component in $G(n, p)$

- In the second case the size of the GC is in proportion to n, i.e. it is exactly n
- That will be our definition of a GC
- With this definition we can calculate the size of the GC in $G(n, p)$
- With u we denote the fraction of nodes that do not belong to the GC
- The size of the GC: $S = 1 - u$
- u is the probability that a randomly chosen node does not belong to the GC

Giant component in $G(n,p)$

- Probability of $i \in V$ and $i \notin GC$ is u
- For the above to hold it must $\forall j \in V$:
    1. $(i,j) \notin E$ or
    2. $(i,j) \in E \implies j \notin GC$

Giant component in $G(n,p)$

- Probability of (1): $1-p$

Giant component in $G(n, p)$

- Probability of (1): $1 - p$
- Probability of (2): pu

Giant component in $G(n, p)$

- Probability of (1): $1 - p$
- Probability of (2): pu
- Total prob. of i not connected to GC via j: $1 - p + pu$

Giant component in $G(n, p)$

- Probability of (1): $1 - p$
- Probability of (2): pu
- Total prob. of i not connected to GC via j: $1 - p + pu$
- Total prob. of i not connected to GC via any other node: $(1 - p + pu)^{n-1}$

$$u = (1 - p + pu)^{n-1} = (1 - p(1 - u))^{n-1} \tag{10}$$

Giant component in $G(n, p)$

- With $\bar{k} = p(n-1)$:

$$
\begin{aligned}
u &= \left(1 - \frac{\bar{k}}{n-1}(1-u)\right)^{n-1} \\
ln(u) &= (n-1)ln\left(1 - \frac{\bar{k}}{n-1}(1-u)\right)
\end{aligned}
$$

- What happens in the limit of large network size, i.e. when $n \to \infty$
- $\frac{\bar{k}}{n-1}(1-u) \to 0$
- Taylor's expansion about 1: $ln(1-x) \approx -x$ for small x

Giant component in $G(n,p)$

$$
\begin{aligned}
ln(u) &\approx -(n-1)\frac{\bar{k}}{n-1}(1-u) \\
ln(u) &\approx -\bar{k}(1-u) \\
u &\approx e^{-\bar{k}(1-u)}
\end{aligned}
$$

- With $S = 1 - u$:

$$
\begin{aligned}
1 - S &= e^{-\bar{k}S} \\
S &= 1 - e^{-\bar{k}S}
\end{aligned}
$$

Giant component in $G(n, p)$

- Expression for the size og GC in the limit of large network size
- No close form solution but we can solve it graphically

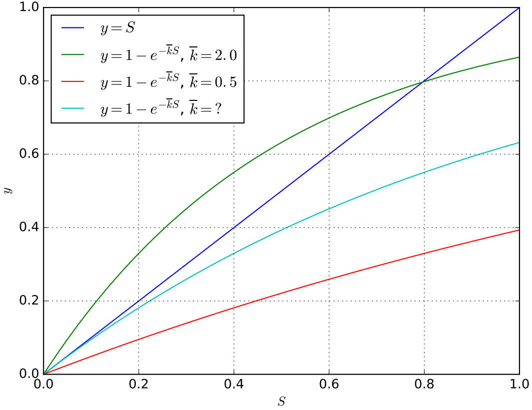


Figure: Giant component in a random graph

Giant component in $G(n,p)$

- Phase transition occurs when gradients are equal for $S = 0$
- Take derivatives of both sides and substitute $S = 0$:

$$\begin{aligned} 1 &= \bar{k}e^{-\bar{k}S} \\ \bar{k} &= 1 \end{aligned}$$

- Recollect $\bar{k} = p(n-1)$
- For $\bar{k} < 1$ no GC
- For $\bar{k} > 1$ GC
- For $\bar{k} = 1$ phase transition

Giant components demo

- NetLogo Example
- http://www.netlogoweb.org/launch#http://www.netlogoweb.org/assets/modelslib/SampleModels/Networks/GiantComponent.nlogo

Diameter of $G(n,p)$

- Average degree: $\bar{k}$
- Starting at a random i
- At distance 1 we have $\bar{k}$ other nodes
- At distance 2 we have $\bar{k} \cdot \bar{k}$ other nodes
- At distance s we $\bar{k}^s$ other nodes
- We can repeat this until $\bar{k}^s \approx n$
- Or equivallently $s \approx \frac{ln(n)}{ln(\bar{k})}$
- Diameter grows as a logarithm of the number of nodes