Aalto University
School of Science
and Technology

CS-E5745
Mathematical Methods for Network Science

Mikko Kivelä

Department of Computer Science
Aalto University, School of Science
mikko.kivela@aalto.fi

January 5, 2021

This lecture

1. Basic concepts and notation (remind CS-E5740)
2. Basic network models (remind CS-E5740)
3. Common approximations:
   - Tree-like approximations
   - “Thermodynamic limit”

Basic definitions and notation (1/8)

- Graph $G = (V, E)$, where V is the set of nodes and $E \subseteq V \times V$ is the set of edges
  - Usually no self-edges: $(v, v) \notin E$ for any $v \in V$.
  - In this course: undirected networks, no multiedges or weighted edges (unless otherwise mentioned)
  - We will use both vertices/edges and nodes/links.
  - $N = |V|$ and $L = |E|$
- Last lecture: Multilayer networks

Basic definitions and notation (2/8)

- Two nodes v, u are *adjacent* or *neighbors* if there is a link $(v, u) \in E$, and (v, u) is *incident* to v and u.
- *Neighborhood* of node v is the set of nodes adjacent it: $\Gamma(v) = \{u | (v, u) \in E\}$
- *Degree* of node v is the number of adjacent nodes: $k_v = |\Gamma(v)|$

Basic definitions and notation (3/8)

- *Walk* is a sequence of nodes and connected by links $(v_0, e_1, v_1, e_2, \ldots, e_l, v_l)$, where $e_i = (v_{i-1}, v_i) \in E$
  - Length of the walk is the **number of edges** in it
- *Path* is a walk where all nodes are distinct, with the exception that the first and the last node can be the same.
- *Cycle* is a path where the first and the last node are the same
- *Distance* between two nodes is the length of the shortest path between those nodes
- Note! Some sources might have different definitions for walk and path! Always define these concept (outside of this course) for clarity.

Basic definitions and notation (4/8)

- Two nodes are *connected* if there is a path between them
- *Component* is a maximal set of nodes that are connected
  - Connectivity partitions an undirected graph into components (i.e., each node is in exactly one component)
  - The *size of the component* is the number of nodes in it
  - The *largest component* is the one with largest size
- *Connected graph* is a graph with a single component
- *Giant component* is the component that spans non-zero fraction of an infinitely large network

Basic definitions and notation (5/8)

- *Percolation* theory in networks describes the properties of connected components
  - *Site percolation*: paths are allowed only through *occupied* nodes
  - *Bond percolation*: paths are allowed only through *occupied* edges
  - Identical to removing nodes/edges
- Physics: regular lattices, nodes or edges set independently and uniformly occupied with *occupation probability*

Basic definitions and notation (6/8)

- We assume that the nodes are (or are implicitly mapped to) numbers from 0 to $N - 1$
- *Adjacency matrix*:

$$A_{uv} = \begin{cases} 1, & \text{if } (u, v) \in E \\ 0, & \text{if } (u, v) \notin E. \end{cases}$$

- Useful when working with networks, e.g.,
  - Degree: $k_v = \sum_u A_{uv}$
  - Number of walks of length n starting at v and ending at u: $(A^n)_{uv}$
  - ...

Basic definitions and notation (7/8)

- *Global clustering coefficient* or *transitivity* is

$$C = \frac{\text{\# of triangles}}{\text{\# connected triplets}} = \frac{\sum_{uvh} A_{uv} A_{vh} A_{hu}}{\sum_{uvh} A_{uv} A_{hu}} = \frac{\mathrm{Tr}(A^3)}{\mathrm{Tr}(AFA)}$$

  - Tr is the trace operator and F is the adjacency matrix of a full graph
- *Local clustering coefficient* for node u is

$$c_u = \frac{\sum_{vh} A_{uv} A_{vh} A_{hu}}{\sum_{vh} A_{uv} A_{hu}} = \frac{(A^3)_{uu}}{(AFA)_{uu}} = \frac{(A^3)_{uu}}{k_u(k_u - 1)/2}$$

Basic definitions and notation (8/8)

- *Tree* is a connected graph with no loops
  - Equivalently, a connected graph with $N - 1$ edges
- *Forest* is a graph that consists of trees

Random graph models

- Create an artificial random network with desired properties
  - = probability distributions over all graphs $P(G)$
  - (*Physicist jargon*: probability distribution is an “ensemble”)
- Can be roughly divided to two categories:
  1. Null models that have some set of structural properties but otherwise maximally random: Usually closed form formula for $P(G)$
  2. Stylised models to analyse particular microscopic generation rules: No closed form formula for $P(G)$, only algorithm for sampling

Erdős-Rényi random graphs

- "soft" and "hard" versions:
  - $G(N, p)$: N nodes, each link exists with probability p
  - $G(N, L)$: N nodes and L links distributed uniformly randomly between the nodes
  - (*Physicist jargon*: These are some times called "canonical ensemble" and "microcanonical ensemble")
- $G(N, p = \frac{L}{N(N-1)/2}) \approx G(N, L)$, because $\langle L \rangle = pN(N-1)/2$
  - Often used interchangeably for large networks
  - Differences in these two discussed later in the course
- $G(N, p)$ AKA *Bernoulli random graphs*

Configuration model (1/6)

- *Configuration model*: a completely random graph with given degree sequence $\{k_u\}_u$
- Again "soft" and "hard" variants can be constructed:
  - Each graph with the **exactly** the given degree sequence is sampled uniformly randomly
  - The **expected value** of degrees is given by the degree sequence, but there can be slight deviations
  - More on these on the 5th lecture
- In practise also other variants and complications, see a recent review article: `https://arxiv.org/abs/1608.00607`
- Note: Often only the "hard" variant is said to be a configuration model, and "soft" variants have different names (e.g., Chung-Lu model)

Configuration model (2/6)

- The "hard" variant of the configuration:

$$P(G|\{k_u\}) = \begin{cases} \frac{1}{\Omega(\{k_u\})}, & \text{if } k(G) = \{k_u\} \\ 0, & \text{otherwise.} \end{cases}$$

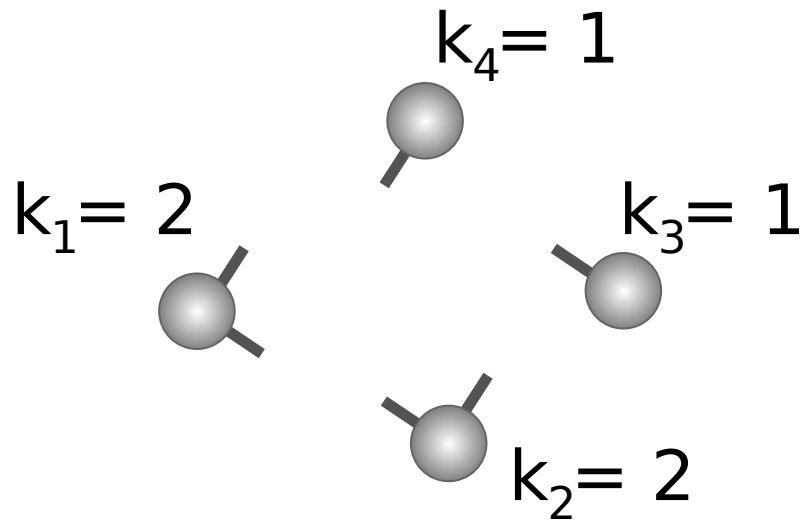
- The sequence $\{k_u\}$ is a *graphic sequence* iff
  - $\sum_u k_u$ is even, and
  - $\sum_{u=0}^{r} k_u \leq r(r-1) + \sum_{i=r+1}^{N-1} min(r, k_u)$, for all $r \leq N-2$ (where in the sums $\{k_u\}$ ordered such that $k_u \geq k_{u+1}$)

Configuration model (3/6)

- The configuration model can be relaxed by allowing multi-links and self-loops
  - Only requirement is that $\sum_u k_u$ is even
  - Large sparse networks will have small number of multi-links and loops
- Easy generation algorithm based on *stubs*
  - Node u has k_u stubs
  - Select two stubs uniformly randomly and connect

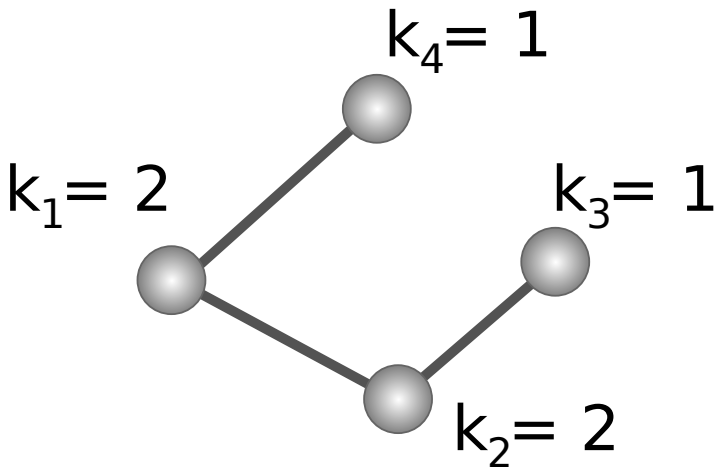
Configuration model (4/6)

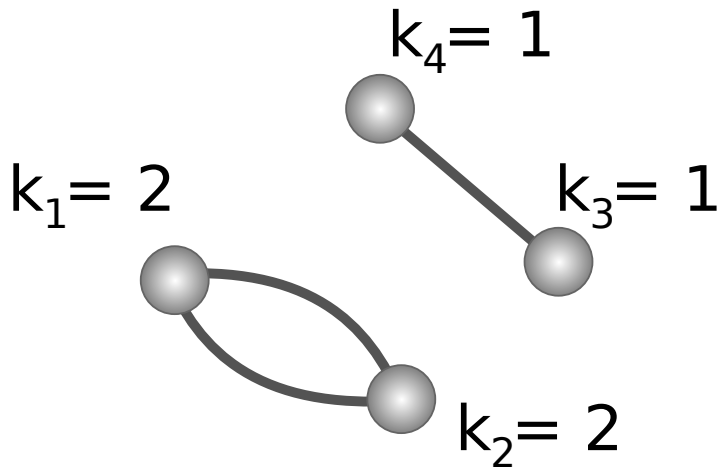
- Example: $\{k_u\} = \{2, 2, 1, 1\}$. Stubs:



Configuration model (5/6)

- Example: $\{k_u\} = \{2, 2, 1, 1\}$. Two of the possible solutions:





Configuration model (6/6)

- A "soft configuration model": each edge (u, v) (including $u = v$) is present independently with probability

$$P((u,v) \in G | \{k_u\}) = \begin{cases} \frac{k_u k_v}{\sum_u k_u}, & \text{if } u \neq v \\ \frac{k_u k_v}{2\sum_u k_u}, & \text{if } u = v. \end{cases}$$

- This leads to the expected value of the degree of each node to follow the given sequence $\langle k(G) \rangle = \{k_u\}$
- Very similar formula can be derived for the expected number of edges between two nodes in the "hard" configuration model variant with multiedges
  - ... but the edges do not appear independently of each other
- Few variants exist (see Hofstad: inhomogeneous random graphs, Chung-Lu model, Norros-Reittu model)

Assumption and approximations

- Analytical calculations are often impossible if you want to do them *precisely* for finite networks
- We want the *big picture*, and don't care about minor details or extreme accuracy of our calculations
- We do simplifying assumptions and approximations, such as
  - Concentrate on what happens at the infinite network size
  - Assume that we can disregard some aspects of the network structure
  - Leave out higher order terms in series expansions
  - ...

Infinitely large networks

- **Assumption:** Network is big enough that it behaves like an infinitely large system
- It is often convenient to study some class of networks when $N \to \infty$
  - (*Physicist jargon*: Taking infinite limit on a size of the system keeping some other variables constant is called the "thermodynamic limit")
- Calculations and results often become simpler: only the largest effects matters, details and higher order effects can be omitted
- Example: $G(N, p)$ and $G(N, L)$ become in effect the same ensemble at the thermodynamic limit
  - Warning: it is often assumed that all "soft" and "hard" distributions become the same, but this is not necessarily true, see Squartini et al. "Breaking of Ensemble Equivalence in Networks" PRL (2015)

Tree-like approximations

- **Assumption:** Network doesn't have any loops, or the loops only have a minor effect to the phenomena that is studied
- Many calculations for trees are often easier than for general graphs
  - Example: Calculate the number of nodes that can be reached from a node
- Sparse random networks are locally tree-like [Exercise 1.3]
  - Many results can be shown to be precise for infinitely large networks using this idea (see Hofstad)
- The tree-like assumption is very common in networks literature and often implicit
- Real networks have high clustering coefficient but the theory still seems to work, see Melnik et al. "The unreasonable effectiveness of tree-based theory for networks with clustering" PRE (2011)

Mean-field-type approximations

- **Assumption: Parts of the network can be grouped together in a way that we can concentrate on the average behavior of each group**
  - Example: All nodes of the same degree have the same probability of being infected in epidemics
- Calculations relying on this assumption are called “mean field theory”
- Very common approach in network science

Does my theory work for real networks?

- Often in the literature no formal tools are given to determine if the theory works for particular network
  - Example: How much fluctuations from the theory I should expect to have when my network has N nodes?
- Typical approach: compare analytical results to example data or detailed simulations

Does my theory work for real networks?

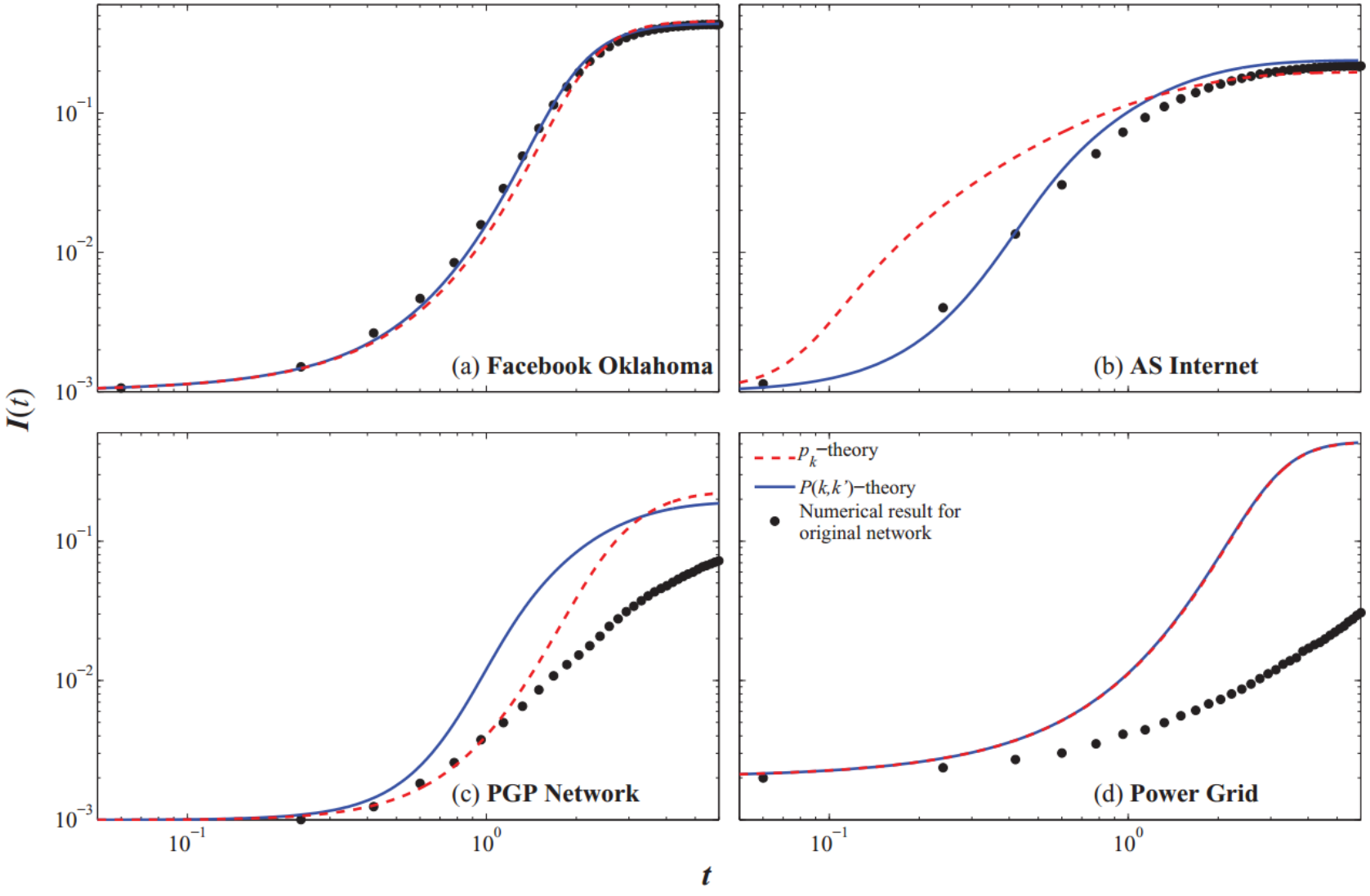


Figure from: Melnik et al. "The unreasonable effectiveness of tree-based theory for networks with clustering" PRE (2011)

Following a link leads to high degree nodes

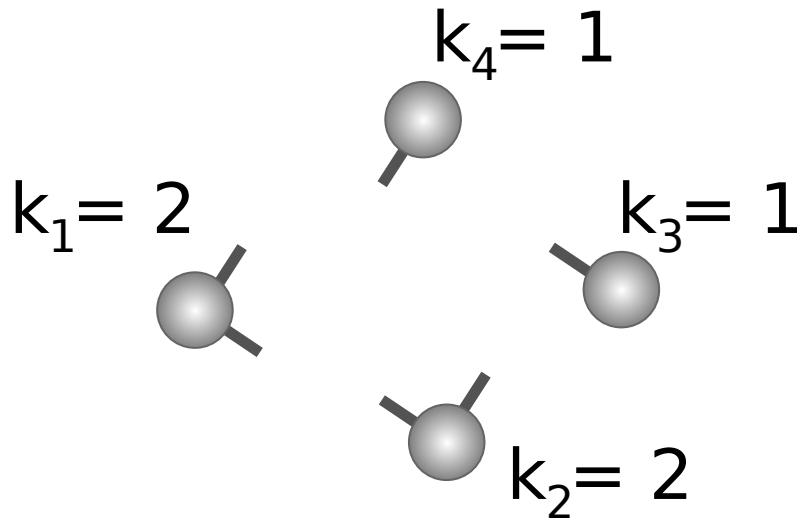
- A node with degree k has probability proportional to k of being reached when a link is followed
  - Selecting random link, and one of its end points
  - Selecting a random node, and one of its neighbors

$$p'(k) \propto kp(k)$$

- Recurring theme in calculations
  - Spreading process is more likely to reach high degree nodes, high degree nodes are more effective spreaders ...
  - Neighboring nodes have higher degree than uniformly random nodes, high degree nodes are more likely to belong to the giant component ...
- Your friends have more friends than you do

Following a link leads to high degree nodes

- Example: $\{k_u\} = \{2, 2, 1, 1\}$. Probabilities to reach node when following a link $\{\frac{2}{6}, \frac{2}{6}, \frac{1}{6}, \frac{1}{6}\}$.



Excess degrees

- Follow a link, how many new links does the node have (i.e., not counting the link used to come to the node)

$$q(k) \propto (k+1)p(k+1)$$

- Network is a forest, start breadth first search from any node...
  - Excess degree is the branching factor
  - Tree can be infinitely large iff avg. excess degree larger than one

Excess degrees

- Example: $\{k_u\} = \{2, 2, 1, 1\}$. Excess degree sequence (one element for each stub) $\{1, 1, 1, 1, 0, 0\}$.