



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](mailto:mikko.kivela@aalto.fi)

February 2, 2023

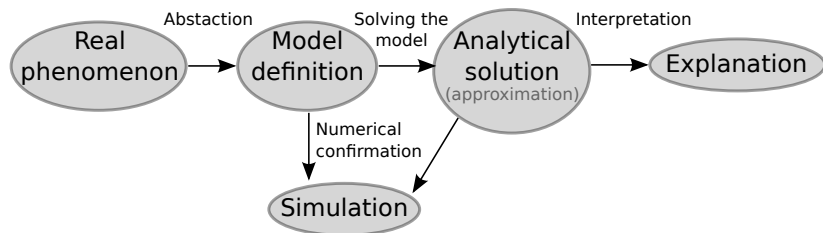
# Dynamical models of/on networks

- ▶ Learning goals this week:
  - ▶ Gain insights on the big picture on how to approach network dynamic problems analytically
  - ▶ Ability to recognize and formulate approximations for these problems
  - ▶ Ability to write down master/rate equations
- ▶ Newman: *Networks, An Introduction* - Sections 14 and 17

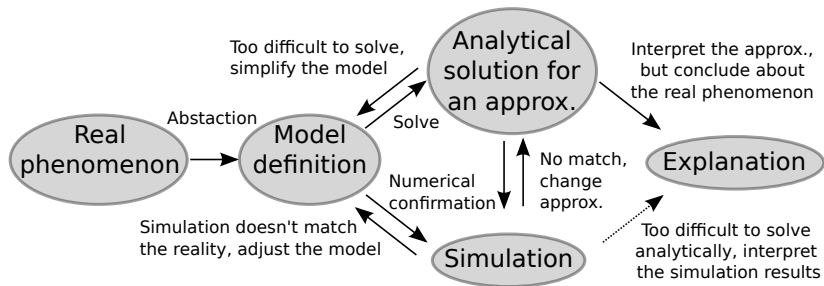
# Dynamical models of/on networks

- ▶ Models of networks: Network evolves according to some rules
- ▶ Models on networks: States of nodes evolve according to some rules
- ▶ Co-evolution models: Both network and states of nodes evolve simultaneously
- ▶ Similar techniques can be used to solve all of these

# Pipeline, as presented in a publication



# Pipeline, a more realistic picture



## Example: Barabási-Albert model

- ▶ Phenomena: Networks are born via growth, new nodes are more likely to connect to already popular nodes
- ▶ Data: Probability of link creation can be measured as a function of degree, power-law degree distributions
- ▶ Model: A variation of preferential attachment models (e.g., Price model):
  0. Start with  $t_0$  seed nodes with some connections, and set time to  $t = t_0$
  1. Increase time  $t = t + 1$ , add node with index  $t$
  2. Add  $m$  links from  $t$  to other nodes, each link to node  $i$  prob.  
$$p_i = \frac{k_i}{\sum_j k_j}$$
  3. GOTO 1.
- ▶ There could be multi-links (or the probabilities are different), but we ignore this

# Example: Barabási-Albert model

- ▶ We can solve the BA model *exactly* if we so wish
- ▶ Idea: Think about the BA model as a Markov chain where the state space is (a subset of) all graphs
  - ▶ Easy to write the transition probabilities between graphs
  - ▶ The result give us a probability distribution over all graphs for each time  $t$

## Example: Barabási-Albert model

- ▶ We can solve the BA model *exactly* if we so wish
- ▶ Idea: Think about the BA model as a Markov chain where the state space is (a subset of) all graphs
  - ▶ Easy to write the transition probabilities between graphs
  - ▶ The result give us a probability distribution over all graphs for each time  $t$
- ▶ Problem: the size of the state space explodes
  - ▶  $2^{T(T-1)/2}$  graphs with  $T$  nodes, but not all reachable
  - ▶ For every state at time  $t$ , there are  $\binom{t}{m}$  possible transitions
  - ▶ Total number of states at time  $T$  is  $N_T = \prod_{t=t_0}^T \binom{t}{m} \geq e^{mT}$  (for  $T$  large enough)
- ▶ This solution is not practical



# Approximate solutions

- ▶ Idea: Define a simple system that mimics the (expected) behavior of a complex system
- ▶ The state of the system at each time is reduced to a property or a vector of properties
  - ▶ The choice of properties is usually guided by the process
- ▶ The evolution rules are written in terms of average behavior (master/rate equation)
  - ▶ Either as difference or differential equation(s)
- ▶ Only mean behavior matters: statistical fluctuations are ignored
  - ▶ The properties are assumed to be independent
  - ▶ Dependencies can be added by including combinations of properties
- ▶ The resulting system is usually much easier to solve!

# Typical procedure in approximate solutions

- ▶ The “real” model for the system  $S$  is defined using stochastic rules  $F$  such that  $S(t) = F(S(t-1))$
- ▶ Calculate some properties  $X$  of the system  $X(S) = \mathbf{X}$ 
  - ▶ The properties can be discrete in the system  $S$  but evolve to be continuous in  $\mathbf{X}$ : e.g., probability that an edge exists
- ▶ Define deterministic rules  $f$  for the evolution of the properties  $\mathbf{X}(t) = f(\mathbf{X}(t-1))$ , s.t.  $f(\mathbf{X}) \approx \langle F(S) | X(S) = \mathbf{X} \rangle$ 
  - ▶ Usually defined in elementwise way:  $X_i(t) = f_i(\mathbf{X}(t-1))$
- ▶ Set  $X(S(t_0)) = \mathbf{X}(t_0)$  and solve  $\mathbf{X}(t) = f(\mathbf{X}(t-1))$  for any value of  $t$  or for  $t \rightarrow \infty$ 
  - ▶ We now hope that  $X(\langle F^t(S(t_0)) \rangle) \approx f^t(\mathbf{X}(t_0))$
  - ▶ For continuous time one defines  $\frac{d}{dt} \mathbf{X}(t) = f(\mathbf{X}(t))$

## Example: Barabási-Albert model

- ▶ Idea: map each graph to a vector where each element corresponds to an edge, s.t.  $X_i = 1$  if edge exists, and 0 otherwise
  - ▶ Expected value of binary variable is the probability of getting 1
  - ▶ Notation:  $A_{ij}$  is the probability that link between  $i$  and  $j$  exists
  - ▶ Set  $m = 1$  for simplicity
- ▶ The update rule for  $\mathbf{X}(t)$ :
  - ▶ Calculate expected degree for each node  $\langle k_i \rangle = \sum_j A_{ij}$
  - ▶ Set  $A_{it} = \frac{\langle k_i \rangle}{\sum_j \langle k_j \rangle}$
- ▶ The final state  $\mathbf{X}(T)$  gives the probability of each edge existing at time  $T$

# Practical guidelines for selecting variables for approximations

- ▶ Anything that can be measured from the state of the system could be selected as a property we track
  - ▶ Some selections make it difficult to come up with the update function
  - ▶ The selecting variables appearing in the update rule is a good idea
- ▶ The above approximation discards dependencies between the variables, but one can define a combination variables that keep track of these
  - ▶ Example: Probability for each triangle to exists, instead of a probability each link to exists

# Example: Barabasi-Alberts model

- ▶ Idea 1: map each graph into a vector of degrees of nodes
  - ▶ The expected degree of each node is tracked
  - ▶ Note: expected degrees are continuous variables
- ▶  $X_i(t)$  is the expected degree of node  $i$  at time step  $t$
- ▶ Now we should write the equations  $X_i(t) = f_i(\mathbf{X}(t - 1))$

# Example: Barabasi-Alberts model

- ▶ Idea 2: map each graph into a vector of counts of nodes of each degree (or a degree distribution)
  - ▶ The expected number of nodes of each degree is tracked
- ▶  $X_k(t)$  is the expected number of nodes of degree  $k$  at time step  $t$
- ▶ Now we should write the equations  $X_k(t) = f_k(\mathbf{X}(t - 1))$

# Practical guidelines for writing master equations

- ▶ When the tracked variables are not binary variables, it might be more difficult to write the update equations
- ▶ Think about each element  $X_i(t)$  separately, divide it to
  - ▶ *growth*  $X_i^+(t) = f_i^+(\mathbf{X}(t-1))$  term and
  - ▶ *decline*  $X_i^-(t) = f_i^-(\mathbf{X}(t-1))$  term
- ▶ The total is
$$X_i(t) - X_i(t-1) = X_i^+(t) - X_i^-(t) = f_i^+(\mathbf{X}(t-1)) - f_i^-(\mathbf{X}(t-1))$$
- ▶ Are there any conserved quantities? (If something grows, others decline)
- ▶ Same applies for rate equations

# Example: Barabasi-Alberts model

- ▶  $X_k(t)$  is the expected number of nodes of degree  $k$  at time step  $t$
- ▶ Addition of nodes to degree categories:
  - ▶  $X_k^+(t) = m \frac{(k-1)}{\sum_k kX_k(t-1)} X_{k-1}(t-1)$ , when  $k > m$
  - ▶  $X_k^+(t) = 1$ , when  $k = m$
- ▶ Removal of nodes from degree categories:
  - ▶  $X_k^-(t) = m \frac{k}{\sum_k kX_k(t-1)} X_k(t-1)$
- ▶ Note that  $\sum_k kX_k(t) = 2mN(t)$



# Example: Barabasi-Alberts model

- ▶ In total we have:

- ▶  $X_k(t) = X_k(t-1) + \frac{(k-1)}{2N(t-1)} X_{k-1}(t-1) - \frac{k}{2N(t-1)} X_k(t-1)$ ,  
when  $k > m$

- ▶  $X_m(t) = X_m(t-1) + 1 - \frac{m}{2N(t-1)} X_m(t-1)$

- ▶ At the stationary state we should have:

$$x_k = X_k(t)/N(t) = X_k(t-1)/N(t-1):$$

- ▶  $x_k = \frac{k-1}{k+2} x_{k-1}$ , when  $k > m$

- ▶  $x_m = \frac{1}{1+m/2}$

# Models on networks

- ▶ Above approach can also be used to solve models *on* networks
- ▶ The state of the systems is not a graph, a vector of states for the nodes
- ▶ Examples:
  - ▶ Infection spreading: Nodes are either infected or not infected
  - ▶ Opinion formation: Each nodes state represent its opinion (discreet or continuous value)

# Spreading models on networks

- ▶ Network spreading models follow the classic literature on mathematical epidemiology
- ▶ Models are determined by the possible states and transition rates between them
- ▶ States:
  - ▶ S: Susceptible
  - ▶ E: Exposed
  - ▶ I: Infected
  - ▶ R: Recovered (or Removed)
- ▶ Naming convention: state changes from left to right
- ▶ Typical models include: SI, SIS, SIRS, SEIR

# Example: SIR on networks

- ▶  $S \xrightarrow{\beta} I \xrightarrow{\gamma} R$
- ▶ Classic result (no network): Epidemic if  $R_0 = \frac{\beta}{\gamma} > 1$ 
  - ▶ i.e., when rate of infection is larger than rate of recovery
- ▶ Degree-based approximation: Epidemic if  $\frac{\beta}{\gamma} > \frac{1}{\langle q \rangle}$ 
  - ▶ where  $\langle q \rangle$  is the expected excess degree
- ▶ Node-based approximation: Epidemic if  $\frac{\beta}{\gamma} > \frac{1}{\lambda_1}$ 
  - ▶ where  $\lambda_1$  is the largest eigenvalue of an adjacency matrix

# SIR can be mapped to percolation

- ▶ Modification: all individuals stay infected for time  $\tau$  (instead of  $I \xrightarrow{\gamma} R$ )
- ▶ Infected node: probability that an *edge* leading out is activated during infection is  $\Phi = 1 - e^{-\beta\tau}$
- ▶ Set edges occupied with probability  $\Phi \rightarrow$  components give you the possible sizes of epidemics
  - ▶ We can now calculate the statistics of epidemics using percolation theory on networks

# Binary state models on networks

- ▶ A general theory for approximating processes where each node can be in one of two states<sup>1</sup>
  - ▶ Examples: SI, SIS, voter model, ...
  - ▶ Assumption: network is produced by configuration model
- ▶ Probabilities that node of degree  $k$  is susceptible (infected) and has  $m$  infected neighbors:  $s_{k,m}$  ( $i_{k,m}$ )
- ▶ Dynamics determine the rates  $S \rightarrow I$ :  $F_{k,m}$  and  $I \rightarrow S$ :  $R_{k,m}$ 
  - ▶ Example: in the SIS model  $F_{k,m} = m\beta$  and  $R_{k,m} = \gamma$

---

<sup>1</sup>Phys. Rev. Lett. 107, 068701 (2011)

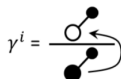
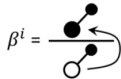
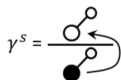
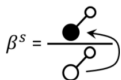
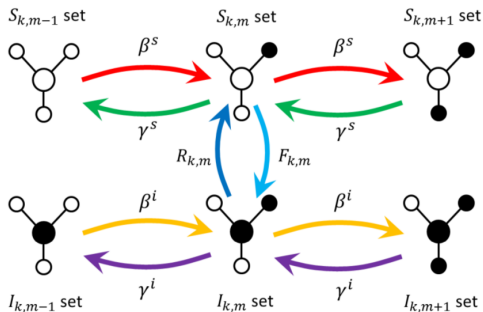
# Binary state models on networks

- ▶ The master equations become complicated:

$$\begin{aligned}\frac{ds_{k,m}}{dt} = & -F_{k,m}s_{k,m} + R_{k,m}i_{k,m} \\ & -\beta^S(k-m)s_{k,m} + \beta^S(k-m+1)s_{k,m-1} \\ & -\gamma^S m s_{k,m} + \gamma^S(m+1)s_{k,m+1}\end{aligned}$$

- ▶ Where  $\beta^S = \frac{\langle \sum_{m=0}^k (k-m) F_{k,m} s_{k,m} \rangle}{\langle \sum_{m=0}^k (k-m) s_{k,m} \rangle}$ ,  $\gamma^S = \frac{\langle \sum_{m=0}^k (k-m) R_{k,m} i_{k,m} \rangle}{\langle \sum_{m=0}^k (k-m) i_{k,m} \rangle}$
- ▶ Similar equation for  $i_{k,m}$

# Binary state models on networks





# Binary state models on networks

- ▶ The  $(k_{\max} + 1)(k_{\max} + 2)$  master equations can be solved numerically, or further approximations can be made
- ▶ Pair approximation (PA):
  - ▶  $\rho_k$ : Probability that node of degree  $k$  is infected
  - ▶  $\rho_k$ : Probability that neighbor of a node of degree  $k$  is infected
  - ▶  $s_{k,m} \approx (1 - \rho_k)B_{k,m}(\rho_k)$ , where  $B_{k,m}(\rho) = \binom{k}{m}\rho^m(1 - \rho)^{k-m}$
- ▶ Mean-field approximation (MF):
  - ▶  $\rho_k$ : Probability that node of degree  $k$  is infected
  - ▶  $s_{k,m} \approx (1 - \rho_k)B_{k,m}(\langle \frac{k}{Z} \rho_k \rangle)$

# Binary state models on networks

