Percolation approaches to SIR disease spread

Joel C. Miller & Tom Hladish

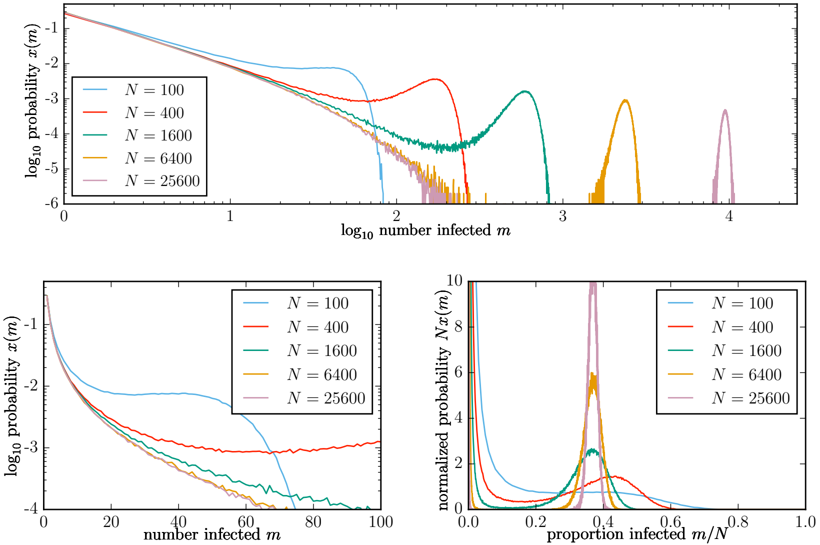
20–22 July 2016

Percolation

We are going to explore a relationship between SIR disease and percolation.
This will lead to methods to

- predict epidemic probability from a single infection.
- predict final size of an epidemic.
- predict the dynamics of an epidemic.

Recall SIR behavior



Modified model

We have a network

- An edge represents a potential transmission path (unweighted, bidirectional).

Modified model

We have a network

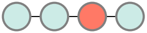
- An edge represents a potential transmission path (unweighted, bidirectional).
- An infected node remains infected for a single time step.

Modified model

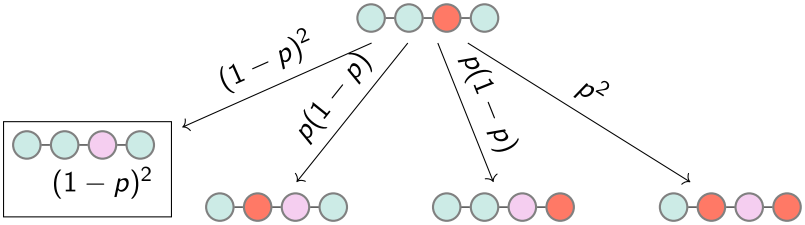
We have a network

- An edge represents a potential transmission path (unweighted, bidirectional).
- An infected node remains infected for a single time step.
- An infected node transmits to a neighbor with probability p.

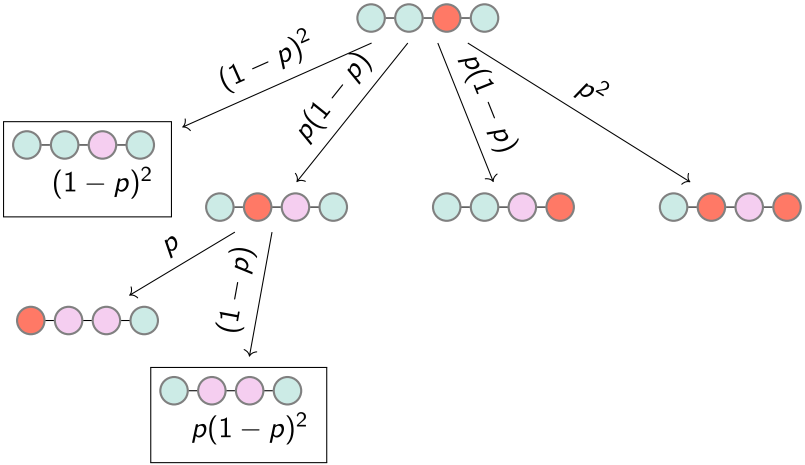
Modeling Disease Spread in a network



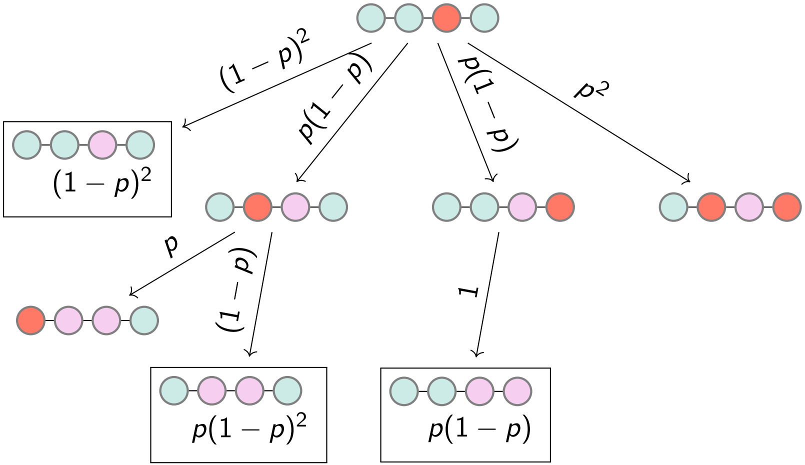
Modeling Disease Spread in a network



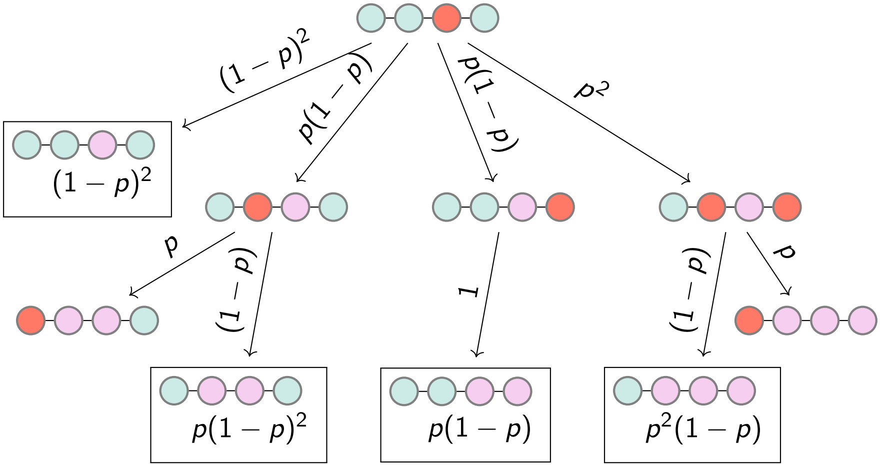
Modeling Disease Spread in a network



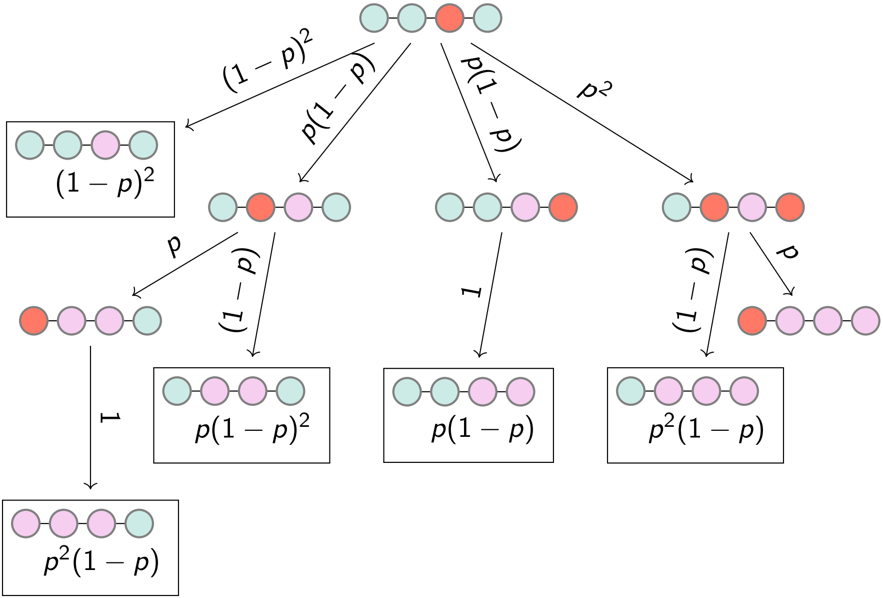
Modeling Disease Spread in a network



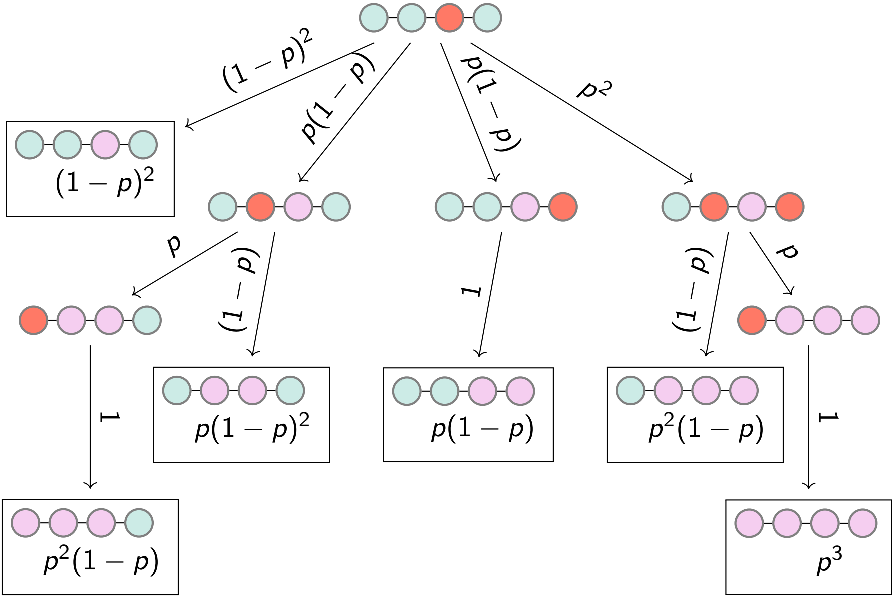
Modeling Disease Spread in a network



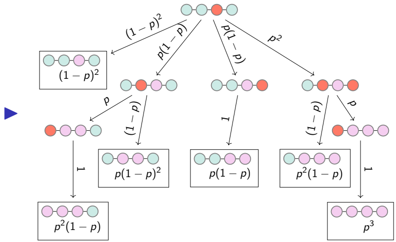
Modeling Disease Spread in a network



Modeling Disease Spread in a network

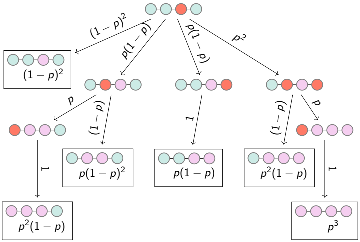


Alternative perspective



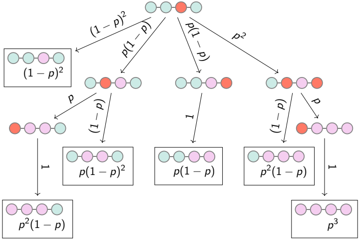
At each step, if there is an edge to cross, it is crossed with probability p. No edge is ever crossed twice.

Alternative perspective

- 

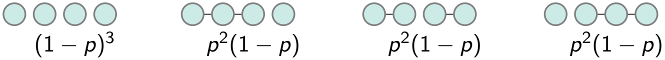
At each step, if there is an edge to cross, it is crossed with probability p. No edge is ever crossed twice.
- It is equivalent to decide in advance whether the edges will be crossed if encountered.

Alternative perspective

- 

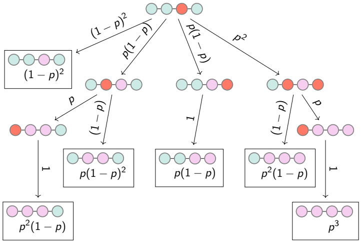
  At each step, if there is an edge to cross, it is crossed with probability p. No edge is ever crossed twice.

- It is equivalent to decide in advance whether the edges will be crossed if encountered.



- 

Alternative perspective

- 

  At each step, if there is an edge to cross, it is crossed with probability p. No edge is ever crossed twice.
- It is equivalent to decide in advance whether the edges will be crossed if encountered.

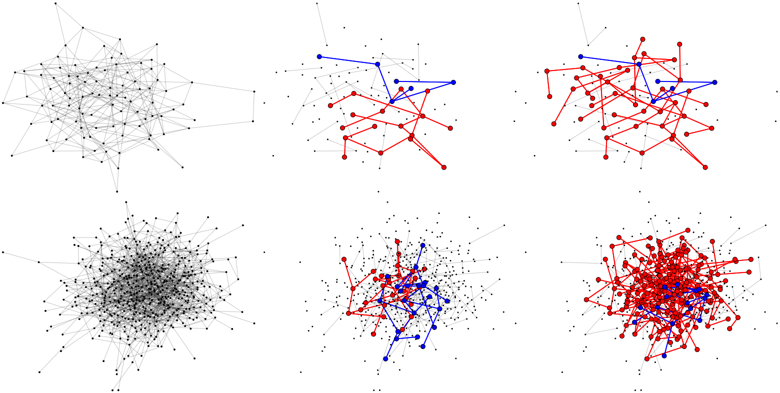
  

  $(1-p)^3$ $\quad p^2(1-p)$ $\quad p^2(1-p)$ $\quad p^2(1-p)$
- 

  $p(1-p)^2$ $\quad p(1-p)^2$ $\quad p(1-p)^2$ $\quad p^3$

Percolation in different size networks

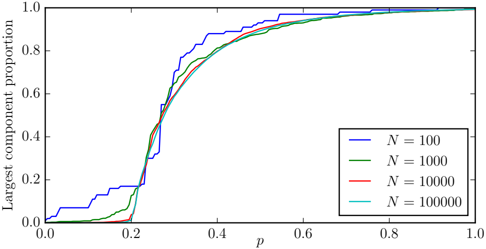
Comparison of largest (red) and second largest (blue) components in different size networks below and above percolation threshold.



- Below threshold largest and second largest are about the same size as each other
- Below threshold largest and second largest are about the same size in both networks

Now return back to transmitting with rate β and recovering with rate γ.

More detailed comparison of network size



Transmission probability

- We'll want to know the probability an infected node v transmits to its neighbor u (assuming v is infected).

Transmission probability

- We'll want to know the probability an infected node v transmits to its neighbor u (assuming v is infected).
- Transmission is at rate β, and recovery is at rate γ. By "swimming pool analogy", probability of transmitting before recovering is $\beta/(\beta+\gamma)$.

Transmission probability

- We'll want to know the probability an infected node v transmits to its neighbor u (assuming v is infected).
- Transmission is at rate β, and recovery is at rate γ. By "swimming pool analogy", probability of transmitting before recovering is $\beta/(\beta+\gamma)$.
- Note: v transmitting to u and to w are correlated events (both depend on duration of v's infection), but transmissions from different nodes to a single node are independent.

Direct percolation analogy

Given a network G, I want to simulate the spread of an SIR disease.

- I use Tom as a random number generator.

Direct percolation analogy

Given a network G, I want to simulate the spread of an SIR disease.

- I use Tom as a random number generator.
  - When a node u becomes infected, I ask Tom: "how long will its infection last?"
  - Then for each neighbor v I ask "will u transmit to v? When?"

Direct percolation analogy

Given a network G, I want to simulate the spread of an SIR disease.

- I use Tom as a random number generator.
  - When a node u becomes infected, I ask Tom: "how long will its infection last?"
  - Then for each neighbor v I ask "will u transmit to v? When?"
- Tom decides he doesn't like the rush to generate a random number on the fly. So he does it in advance.

Direct percolation analogy

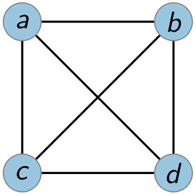
Given a network G, I want to simulate the spread of an SIR disease.

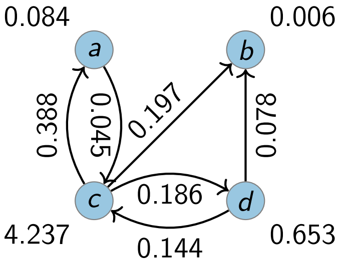
- I use Tom as a random number generator.
  - When a node u becomes infected, I ask Tom: "how long will its infection last?"
  - Then for each neighbor v I ask "will u transmit to v? When?"
- Tom decides he doesn't like the rush to generate a random number on the fly. So he does it in advance.
  - For each node Tom assigns the duration its infection will last if infected.
  - Once the duration is chosen, Tom decides which neighbors it will transmit to and how long it will take.
  - Then he reports those to me when I ask.

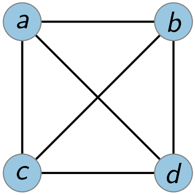
Direct percolation analogy

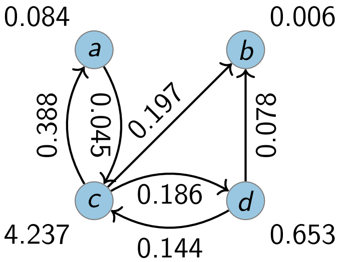
Given a network G, I want to simulate the spread of an SIR disease.

- I use Tom as a random number generator.
  - When a node u becomes infected, I ask Tom: "how long will its infection last?"
  - Then for each neighbor v I ask "will u transmit to v? When?"
- Tom decides he doesn't like the rush to generate a random number on the fly. So he does it in advance.
  - For each node Tom assigns the duration its infection will last if infected.
  - Once the duration is chosen, Tom decides which neighbors it will transmit to and how long it will take.
  - Then he reports those to me when I ask.
- Is it possible for me to know whether he is calculating in advance or not?


a
b
c
d

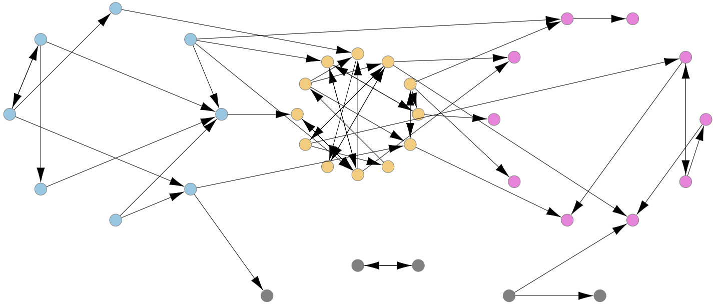

0.084
0.006
a
b
0.388
0.045
0.197
0.078
c
0.186
d
4.237
0.144
0.653





Every number that Tom gives me is a random number that is generated independently of every other number. It doesn't matter when he generates it.

Typical structure



Directed Percolation Equivalence

The following processes produce Indistinguishable output:

- Normal Epidemic Simulation:

Directed Percolation Equivalence

The following processes produce Indistinguishable output:

- Normal Epidemic Simulation:
  - Take a network G.
  - Choose an initial infected individual.
  - Allow edges to transmit until the random time the individual recovers.

Directed Percolation Equivalence

The following processes produce Indistinguishable output:

- Normal Epidemic Simulation:
  - Take a network G.
  - Choose an initial infected individual.
  - Allow edges to transmit until the random time the individual recovers.
- Percolation-based simulation:

Directed Percolation Equivalence

The following processes produce Indistinguishable output:

- Normal Epidemic Simulation:
  - Take a network G.
  - Choose an initial infected individual.
  - Allow edges to transmit until the random time the individual recovers.
- Percolation-based simulation:
  - Take a network G.

Directed Percolation Equivalence

The following processes produce Indistinguishable output:

- Normal Epidemic Simulation:
  - Take a network G.
  - Choose an initial infected individual.
  - Allow edges to transmit until the random time the individual recovers.
- Percolation-based simulation:
  - Take a network G.
  - Generate a new directed network H:

Directed Percolation Equivalence

The following processes produce Indistinguishable output:

- Normal Epidemic Simulation:
  - Take a network G.
  - Choose an initial infected individual.
  - Allow edges to transmit until the random time the individual recovers.
- Percolation-based simulation:
  - Take a network G.
  - Generate a new directed network H:
    - For each individual u, assign a duration d of infection.
    - For each edge from u, determine delay τ until transmitting.
    - If $\tau < d$, place directed edge into network with associated time.

Directed Percolation Equivalence

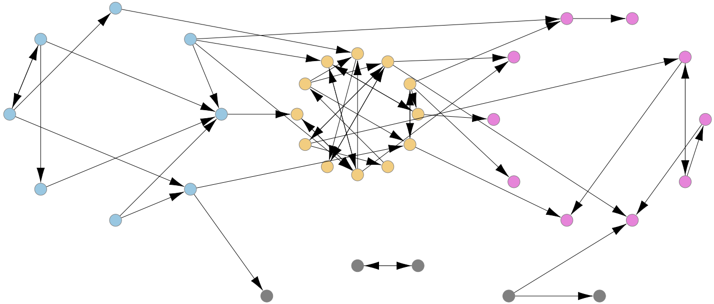
The following processes produce Indistinguishable output:

- Normal Epidemic Simulation:
  - Take a network G.
  - Choose an initial infected individual.
  - Allow edges to transmit until the random time the individual recovers.
- Percolation-based simulation:
  - Take a network G.
  - Generate a new directed network H:
    - For each individual u, assign a duration d of infection.
    - For each edge from u, determine delay τ until transmitting.
    - If $\tau < d$, place directed edge into network with associated time.
  - Choose an initial infected individual.
  - Trace the disease spread, transmitting after given time if an edge is in the new network.

Comments on directed percolation

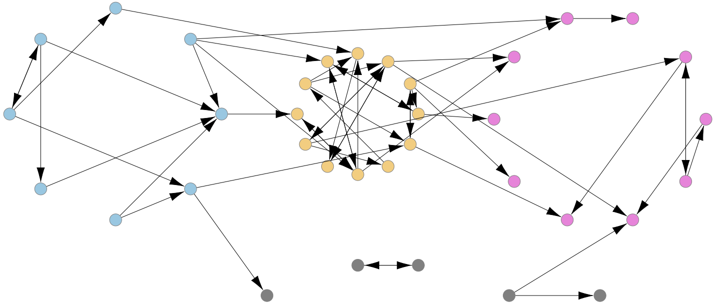
- Directed percolation can be used more generally when there are other sources of heterogeneity in infectiousness and/or susceptibility.
- The eventually infected nodes are exactly those nodes in the out-component of the index case.
- The probability a random node is infected follows from the size of its in-component.

Typical structure



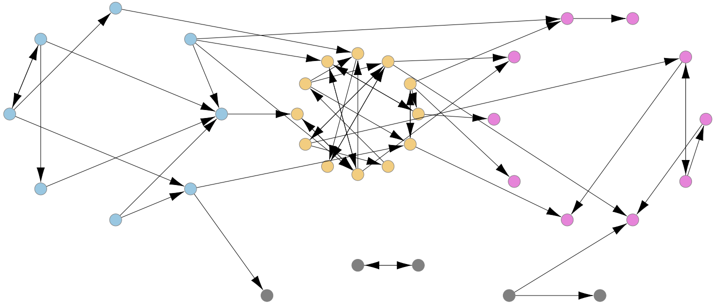
- We can understand the dynamics with a “bowtie” diagram.

Typical structure



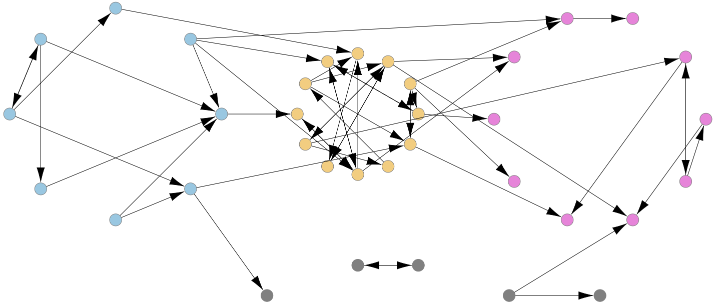
- We can understand the dynamics with a “bowtie” diagram.
- Above a threshold there is a Giant Strongly Connected Component H_{SCC}

Typical structure



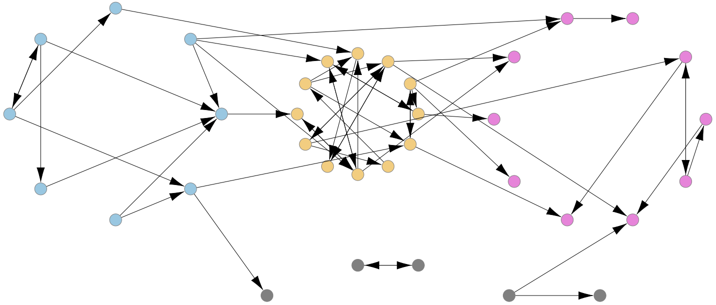
- We can understand the dynamics with a “bowtie” diagram.
- Above a threshold there is a Giant Strongly Connected Component H_{SCC}
- It has an in-component H_{IN} and an out-component H_{OUT}.

Typical structure



- We can understand the dynamics with a "bowtie" diagram.
- Above a threshold there is a Giant Strongly Connected Component H_{SCC}
- It has an in-component H_{IN} and an out-component H_{OUT}.
- If the index case is in H_{IN} or H_{SCC} then all of H_{SCC} and H_{OUT} are eventually infected.

Typical structure



- We can understand the dynamics with a "bowtie" diagram.
- Above a threshold there is a Giant Strongly Connected Component H_{SCC}
- It has an in-component H_{IN} and an out-component H_{OUT}.
- If the index case is in H_{IN} or H_{SCC} then all of H_{SCC} and H_{OUT} are eventually infected.
- So Epidemic Probability $\mathcal{P} = \mathbb{E}(|H_{IN} \cup H_{SCC}|)/N$ and Attack rate $\mathcal{A} = \mathbb{E}(|H_{SCC} \cup H_{OUT}|)/N$.

SIR epidemic size

- Consider a Configuration Model network in which we infect a (probably small) fraction of the population ρ.
- Allow the SIR disease to spread.
- We assume ρN is large enough that stochastic die out does not play a major role.
- What proportion end up susceptible or recovered?

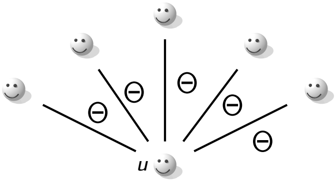
Changing the question

Instead of asking what proportion end up susceptible or recovered ask:

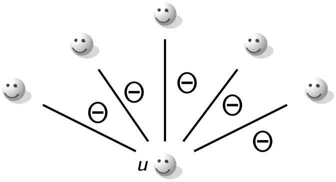
Changing the question

Instead of asking what proportion end up susceptible or recovered ask:

What is the probability a random node does not have a transmission path to it from one of the index nodes?



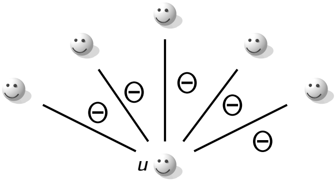
$\ominus = P(v \text{ did not transmit to } u)$



$\Theta = P(v \text{ did not transmit to } u)$

Probability a random degree k test individual is susceptible at the end is

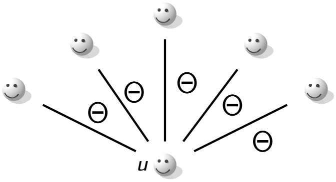
$$(1 - \rho)\Theta^k$$



$\Theta = P(v \text{ did not transmit to } u)$

Probability a random ~~degree k~~ test individual is susceptible at the end is

$$S = \sum_k P(k)(1-\rho)\Theta^k$$



$\Theta = P(v \text{ did not transmit to } u)$

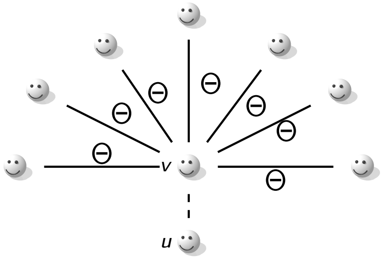
Probability a random ~~degree k~~ test individual is susceptible at the end is

$$S = \sum_k P(k)(1-\rho)\Theta^k = \hat{\psi}(\Theta)$$

where

$$\hat{\psi}(x) = (1-\rho)\sum_k P(k)x^k$$

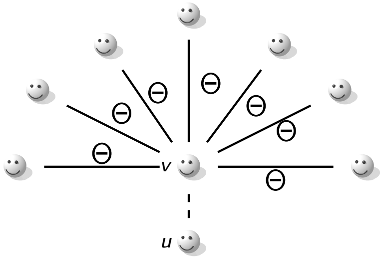
Finding Θ



Probability a random degree k partner never infected is

$$(1-\rho)\Theta^{k-1}$$

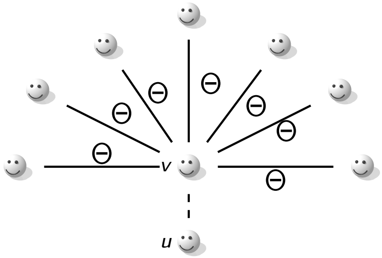
Finding Θ



Probability a random ~~degree k~~ partner never infected is

$$\phi_S = \sum_k P_n(k)(1-\rho)\Theta^{k-1}$$

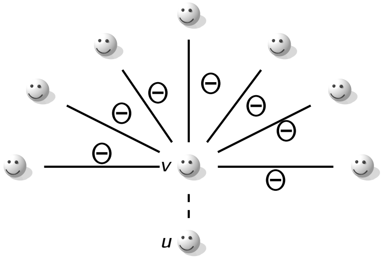
Finding Θ



Probability a random ~~degree k~~ partner never infected is

$$\phi_S = \sum_k \frac{kP(k)}{\langle K \rangle}(1-\rho)\Theta^{k-1}$$

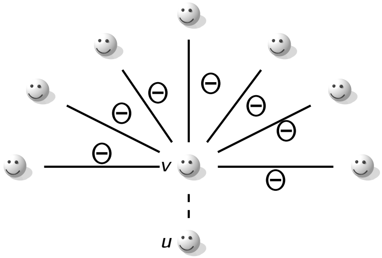
Finding Θ



Probability a random ~~degree k~~ partner never infected is

$$\phi_S = \sum_k \frac{kP(k)}{\langle K \rangle}(1-\rho)\Theta^{k-1} = \frac{\hat{\psi}'(\Theta)}{\langle K \rangle}$$

Finding Θ



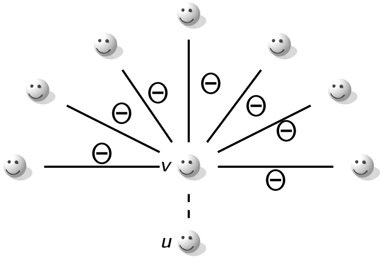
Probability a random ~~degree k~~ partner never infected is

$$\phi_S = \sum_k \frac{kP(k)}{\langle K \rangle}(1-\rho)\Theta^{k-1} = \frac{\hat{\psi}'(\Theta)}{\langle K \rangle}$$

Given β and γ probability partner does not transmit to u is

$$\Theta = \phi_S + \left(1 - \frac{\beta}{\beta+\gamma}\right)(1-\phi_S)$$

Finding Θ



Probability a random ~~degree k~~ partner never infected is

$$\phi_S = \sum_k \frac{kP(k)}{\langle K \rangle}(1-\rho)\Theta^{k-1} = \frac{\hat{\psi}'(\Theta)}{\langle K \rangle}$$

Given β and γ probability partner does not transmit to u is

$$\Theta = \phi_S + \left(1 - \frac{\beta}{\beta+\gamma}\right)(1-\phi_S) = 1 - \frac{\beta}{\beta+\gamma} + \frac{\beta}{\beta+\gamma}\frac{\hat{\psi}'(\Theta)}{\langle K \rangle}$$

Final Size

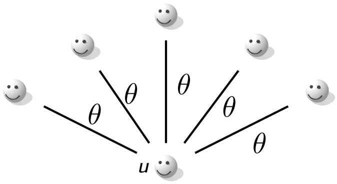
So

$$R = 1 - \hat{\psi}(\Theta)$$

where

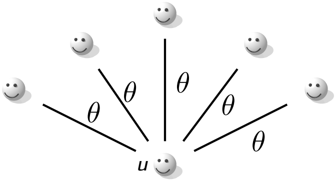
$$\Theta = \frac{\gamma}{\beta+\gamma} + \frac{\beta}{\beta+\gamma}\frac{\hat{\psi}'(\Theta)}{\langle K \rangle}$$

Finding $S(t)$



$\theta(t) = P(v \text{ not yet transmitted to } u)$

Finding $S(t)$

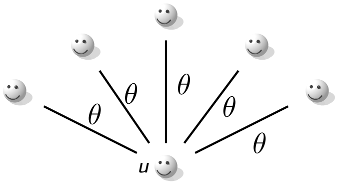


$\theta(t) = P(v \text{ not yet transmitted to } u)$

Probability a random degree k test individual still susceptible is

$$(1-\rho)\theta(t)^k$$

Finding $S(t)$

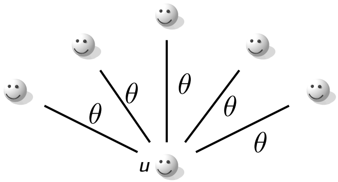


$\theta(t) = P(v \text{ not yet transmitted to } u)$

Probability a random ~~degree k~~ test individual still susceptible is

$$S(t) = \sum_k P(k)(1-\rho)\theta(t)^k$$

Finding $S(t)$



$\theta(t) = P(v \text{ not yet transmitted to } u)$

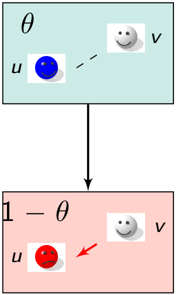
Probability a random ~~degree k~~ test individual still susceptible is

$$S(t) = \sum_k P(k)(1-\rho)\theta(t)^k = \hat{\psi}(\theta(t))$$

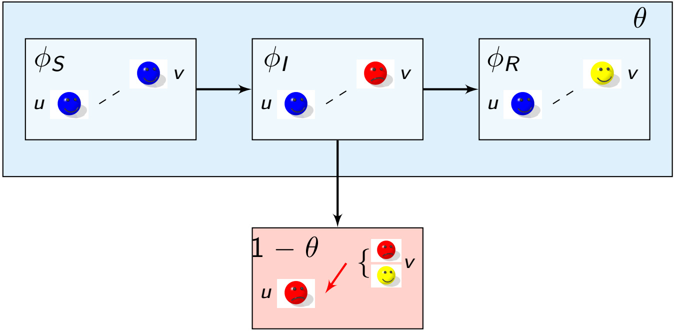
where

$$\psi(x) = \sum_k P(k)x^k$$

How does θ evolve?

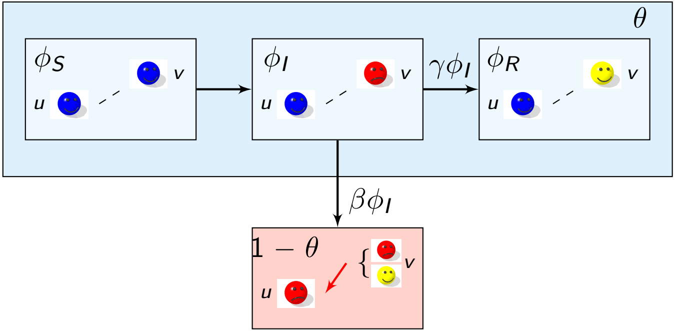


How does θ evolve?



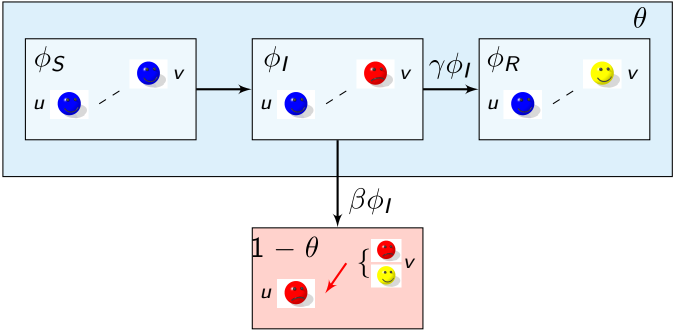
- $\theta = \phi_S + \phi_I + \phi_R$.

How does θ evolve?



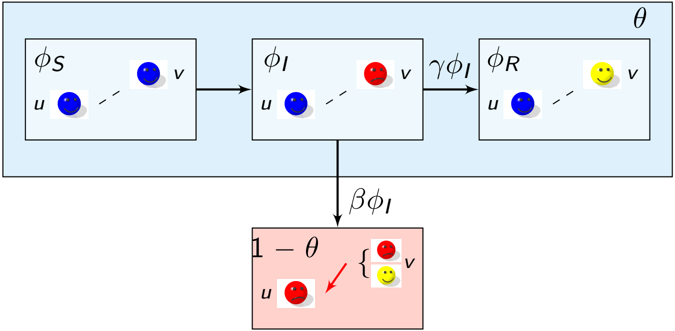
- $\theta = \phi_S + \phi_I + \phi_R$.
- $\dot{\theta} = -\beta\phi_I$.

How does θ evolve?



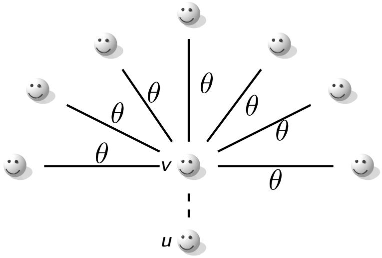
- $\theta = \phi_S + \phi_I + \phi_R$.
- $\dot{\theta} = -\beta\phi_I$.
- Our goal is to find ϕ_I in terms of θ.

Finding $\phi_R(t)$



Because derivatives are proportional (swimming pool analogy),
$\phi_R = \frac{\gamma}{\beta}(1-\theta)$

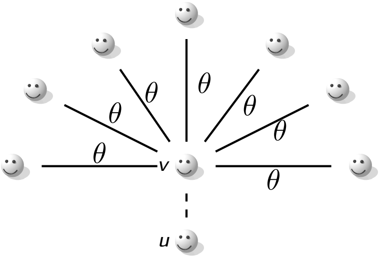
Finding $\phi_S(t)$



Probability a random degree k partner still susceptible is

$$(1-\rho)\theta(t)^{k-1}$$

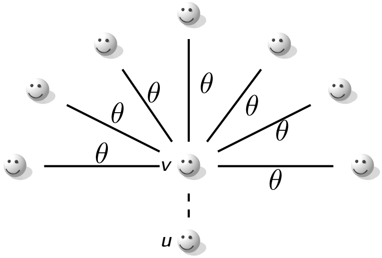
Finding $\phi_S(t)$



Probability a random ~~degree k~~ partner still susceptible is

$$\phi_S(t) = \sum_k P_n(k)(1-\rho)\theta(t)^{k-1}$$

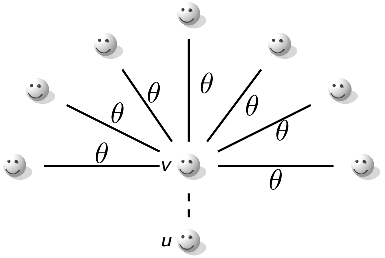
Finding $\phi_S(t)$



Probability a random ~~degree k~~ partner still susceptible is

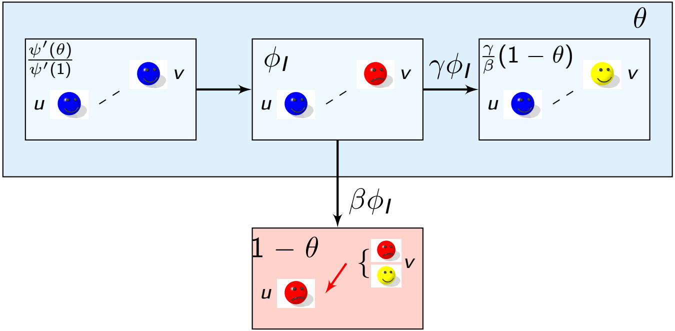
$$\phi_S(t) = \sum_k \frac{kP(k)}{\langle K \rangle}(1-\rho)\theta(t)^{k-1}$$

Finding $\phi_S(t)$



Probability a random ~~degree k~~ partner still susceptible is

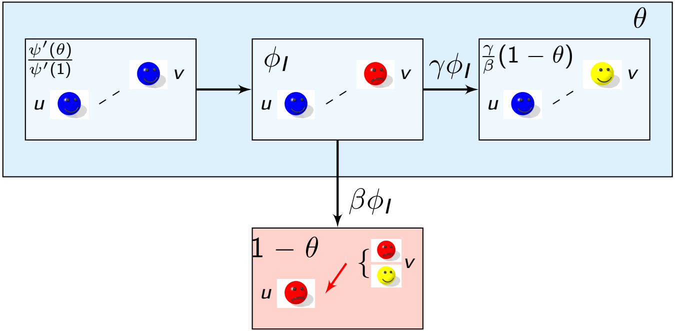
$$\phi_S(t) = \sum_k \frac{kP(k)}{\langle K \rangle}(1-\rho)\theta(t)^{k-1} = \frac{\hat{\psi}'(\theta)}{\langle K \rangle}$$



We have

$$\phi_I = \theta - \phi_S - \phi_R$$

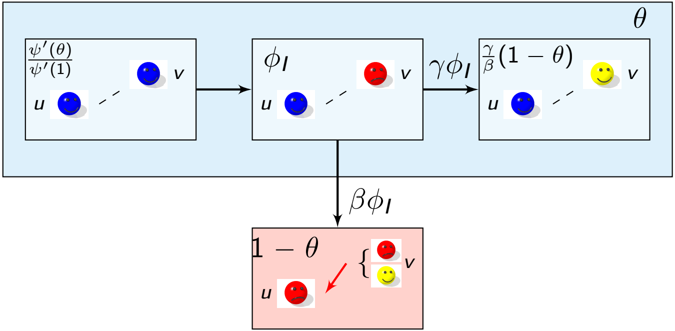
$$\dot{\theta} = -\beta\phi_I$$



We have

$$\phi_I = \theta - \phi_S - \phi_R = \theta - \frac{\psi'(\theta)}{\psi'(1)} - \frac{\gamma}{\beta}(1-\theta)$$

$$\dot{\theta} = -\beta\phi_I$$



We have

$$\phi_I = \theta - \phi_S - \phi_R = \theta - \frac{\psi'(\theta)}{\psi'(1)} - \frac{\gamma}{\beta}(1-\theta)$$

$$\dot{\theta} = -\beta\phi_I = -\beta\theta + \beta\frac{\psi'(\theta)}{\psi'(1)} + \gamma(1-\theta)$$

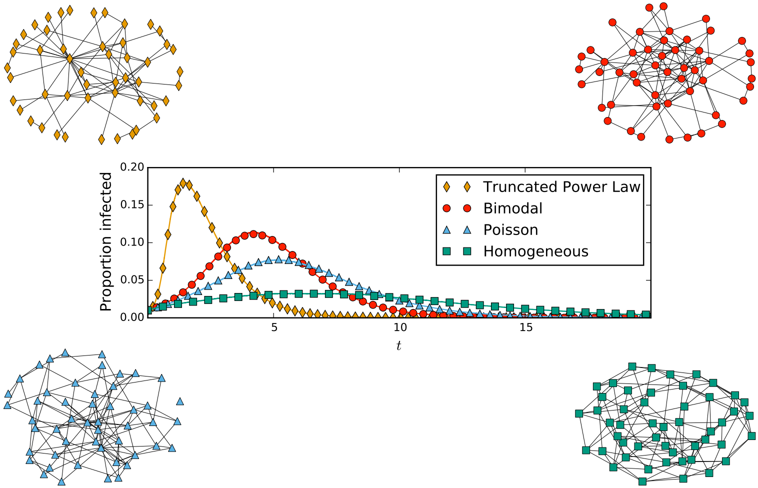
Final System

We finally have

$$\dot{\theta} = -\beta\theta + \beta\frac{\psi'(\theta)}{\psi'(1)} + \gamma(1-\theta)$$

$$\dot{R} = \gamma I \qquad S = \psi(\theta) \qquad I = 1 - S - R$$

More details in [1, 2]


Truncated Power Law
Bimodal
Poisson
Homogeneous
Proportion infected
0.20
0.15
0.10
0.05
0.00
5
10
15
t

A good exercise

Repeat this derivation for a model in which infections last for one time step and transmission occurs with probability T.

References

References I

[1] Joel C. Miller, Anja C. Slim, and Erik M. Volz.
Edge-based compartmental modelling for infectious disease spread.
Journal of the Royal Society Interface, 9(70):890–906, 2012.

[2] Joel C. Miller.
Epidemics on networks with large initial conditions or changing structure.
PLoS ONE, 9(7):e101421, 2014.