

# Epidemics and gossip algorithms

Social and Technological Networks

Rik Sarkar

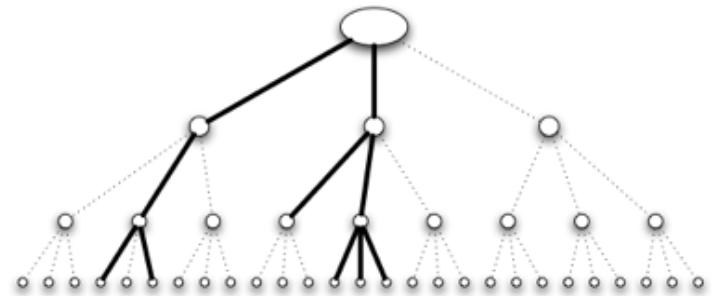
University of Edinburgh, 2018.

# Spread of diseases

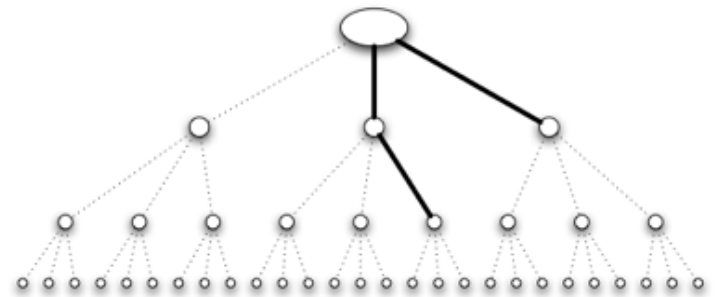
- Pattern depends on network structure
- e.g. spread of flu
- Network of people
- Network of airlines
  
- Different from idea/innovation contagion
  - Does not need a “decision”
  - Does not need multiple support
    - Infectious disease passes easily with some probability

- Suppose everyone meets  $k$  new people and infects each with probability  $p$
- That is, they infect  $R = kp$  people on average

- If  $p$  is high
- The disease will persist through rounds
- If  $p$  is low, it will die out after some rounds



(b) *With high contagion probability, the infection spreads widely*



# Property

- When  $R > 1$  number of infected people keeps increasing
  - Outbreak
- When  $R < 1$  Number of infected people decreases
  - Disease dies out
- Phase transition at  $R = 1$
- assuming there are enough “new” people supply to meet
- Generally true in the initial stages

- Around  $R = 1$ : small efforts can have large effects on epidemic
  - Awareness causing slight decrease in  $p$
  - Quarantine/fear causing slight decrease in  $k$

# SIR Model

- Susceptible (initially)
- Infectious (after being infected)
  - While Infectious, it can pass disease to each neighbor in each step with prob.  $p$
- Removed (after given duration as Infectious)
  - Immune/dead

# SIS model

- No “Removed” state. Susceptible follows Infectious

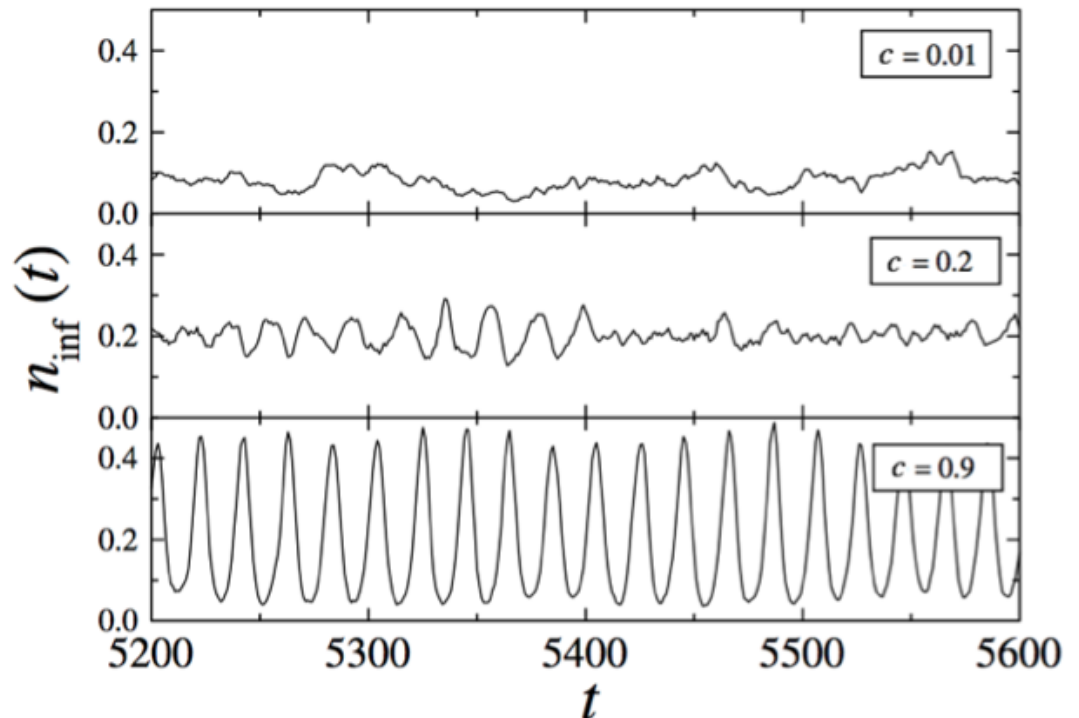


# SIRS model

- Susceptible
- Infectious
- Recovered (immune)
- Susceptible

# SIRS oscillations in Watts-Strogatz Small worlds

- Nodes connected to few neighbors on a ring
- Fraction  $c$  of links modified to connect to random nodes



# Epidemic or gossip algorithm

- Emulates the spread of epidemic or a rumor in a network
- A node speaks to a random neighbor to spread the rumor message
- Useful for spreading information in computer networks

# Spreading a message via gossip

- Complete graph: Anyone can call anyone
- Problem: One node has a message or rumor to spread. How does it spread it to all nodes in the network?
- Calling everyone will take  $O(n)$  rounds.
- After first round, nodes with the rumor can help
  - But how do you avoid collision?

# Spreading a message via gossip

- Complete graph: Anyone can call anyone
- Problem: One node has a message or rumor to spread. How does it spread it to all nodes in the network?
- Strategy: In each round, anyone with the rumor calls one random node and passes the rumor

# Theorem

- Everyone gets the message in  $O(\log n)$  rounds
- Idea:
- $n/3$  nodes get the message in  $\log n$  rounds
  - Current number of infected nodes  $m < n/3$
  - Probability that a call goes to a new node is at least  $2/3$ 
    - Number of calls to new nodes:  $2m/3$
  - Probability of a collision at the new node is  $1/(n-m)$
  - $O(m^2)$  possible pairs for collision
  - Max possible collisions:  $\frac{m^2}{2(n-m)} \leq \frac{m^2}{2} \cdot \frac{1}{2m} = \frac{m}{4}$
- Number of newly infected nodes at least  $\frac{2m}{3} - \frac{m}{4} = \frac{5m}{12}$

- while  $m < n/3$ 
  - $m$  grows to  $m(1 + 5/12) = 17m/12$  every round
  - $m$  grows to  $n/3$  in  $O(\log n)$  rounds
- After  $m > n/3$ 
  - Probability that a node is not called in 1 round is
 
$$\left(1 - \frac{1}{n}\right)^{\frac{n}{3}} \leq e^{\frac{-1}{n} \cdot \frac{n}{3}} = e^{-1/3}$$
  - Probability that 1 or more nodes are not called after  $O(\log n)$  rounds
  - Less than  $1/n^c$ , where  $c$  depends on the constant in the  $O$

- See Kempe 18.



- In a computer network (imagine wireless network)
- Spreading a piece of information
- Naive method: A flood: a node calls all its neighbors to send message
- Wasteful: since many nodes can have common nearby neighbors, this wastes messages
- Better to do it as one neighbor per round
- Still spreads slowly...
- At least  $\sqrt{n}$  rounds in a grid of  $n$  nodes

# Geographic gossip in wireless networks

- Imagine nodes on a plane
- Instead of only sending messages to neighbors
- Send to nodes at random (assume there is some routing mechanism in place)
- Easily reaches far away nodes
- Faster:  $O(\log n)$  rounds:  $O(n \log n)$  messages
- but the routing costs more messages:
  - $\sqrt{n}$  hops on average per message
  - $(n\sqrt{n}) \log n$  total transmissions

# Spread in small world distributions

- Instead send message to a random node
  - Picked according to a small world-like distribution
  - Picking a node at distance  $d$  with probability  $1/d^3$
  - Note the slight difference in exponent (3 instead of 2)
  - Short paths are more common in these distributions
  - The average message cost is  $O(\text{poly}(\log n))$
  - Still there are enough long messages spreading the message to far away regions
- Kempe, Kleinberg, Demers; Spatial gossip and resource location protocols STOC 2001

# Advantages of gossip

- Simple to implement
- Robust algorithm
  - A node failure does not stop the computation
  - Easy to add nodes to the system
  - At the cost of a logarithmic factor of increased costs

# Averaging gossip

- Suppose the nodes all have a “value”
- And we wish to compute a linear function of these values
- E.g. the average

- The push-sum protocol
  - In every round
  - Every node takes a fraction of its value and sends to a random neighbor
  - It adds all received values to its current value
- The pairwise averaging protocol
  - In every round, a node talks to one other random neighbor
  - Both nodes set their values to the average of the two

# Gossip averaging protocols

- On a complete graph
  - Both protocols converge to the average fast
  - $O(\log n)$  rounds
- On small world graphs/small world distributions
  - Convergence not known