

## Spreading Processes on Networks: Models, Techniques and Algorithms

Hans De Sterck School of Mathematical Sciences hans.desterck@monash.edu





### collaborators

Joel Miller, Institute for Disease Modeling, Seattle, USA



John Lang, UCLA, Communications Studies (PhD Waterloo, July 2016)





sources (background on network science, SIR disease propagation, ...)

http://barabasi.com/networksciencebook/ **NETWORK SCIENCE CLASS 3: RANDOM NETWORKS** (CHAPTER 3 IN TEXTBOOK) Albert-László Barabási with Roberta Sinatra WWW.BARABASILAB.COM

arXiv:1702.01252v1



Random Spatial Networks: Small Worlds without Clustering, Traveling Waves, and Hop-and-Spread Disease Dynamics

John Lang, Hans De Sterck, Jamieson L. Kaiser, Joel C. Miller

book by Kiss, Miller and Simon (2017)



on Networks

From Exact to Approximate Models



### motivation: spread of 2013-2016 Ebola epidemic

• Guinea, Sierra Leone, Liberia

Virus genomes reveal factors that spread and sustained the Ebola epidemic

**Gytis Dudas, Luiz Max Carvalho, Trevor Bedford, Andrew J. Tatem, Guy Baele,** *Nature* **544**, 309–315 (20 April 2017) | doi:10.1038/nature22040

- goal: develop modeling framework
- random networks
- spatial structure!
- disease propagation (stochastic, DEs for insight)



4



### two parts of my presentation

- part A: models and algorithms for networks
  - "introduction to network science"
  - random spatial networks
  - algorithms for efficient network generation
  - application: small worlds with spatial structure
- part B: disease propagation on networks
  - propagation of Susceptible-Infectious-Recovered (SIR) disease
  - stochastic simulation algorithms
  - exact analytic models, and simulations
  - applications



random spatial networks (RSNs)



FIG. 1. An example RSN and its properties. The distance kernel is a Gaussian,  $f(d) = \exp(-d^2/2\sigma^2)/2\pi\sigma^2$  with  $\sigma = 0.03$ . The imposed distribution of expected degrees is P(2) = P(15) = 0.5. The density is  $\rho = 10000$ . One node and its neighbors are highlighted. A random network without spatial structure would exhibit neighbors throughout the domain.

6





### B1: Susceptible-Infectious-Recovered (SIR) models

SIR compartmental model – well-mixed populations





Susceptible-Infectious-Recovered (SIR) models

- SIR model on networks
  - infection transmission processes  $\beta$ : infection rate per edge
  - recovery processes  $\gamma$ : recovery rate

O

 $\lambda_1 = 3.1$ 

- each process is modeled as a Poisson process (Markovian, independent): events occur continuously and independently at a constant (average) rate
- for a given state of the network: / processes, each is a Poisson process number of occurrences Q<sub>i</sub> is Poisson distributed

$$Q_i \sim \operatorname{Pois}(\lambda_i) \qquad (1 \le i \le l)$$

Q,~ Pois (3.1)



### Susceptible-Infectious-Recovered (SIR) models

- SIR model on networks : at correct state of network  $Q_i \sim \operatorname{Pois}(\lambda_i)$   $(1 \le i \le l)$ 
  - $Q_{1} \sim Pois(3.1)$

Poisson distribution:

$$P(Q = q) = P(q \text{ events occur in interval } [0,1])$$
$$= \frac{\lambda^q}{q!} \exp(-\lambda)$$

 $\lambda$ : number of occurrences per unit time

$$E(Q) = \lambda$$
  $Var(Q) = \lambda$ 



- Poisson process:
  - first arrival time is exponentially distributed, with rate :



 $Q_1 \sim \text{fois}(3.1)$ 

 $q_1 = 3$ 

 $\lambda_1 = 3.1$ 

0

E



### Gillespie algorithm

**Input:** Network G, per-edge transmission rate  $\mathbf{B}$ , recovery rate  $\mathbf{X}$ , set of index node(s) initial\_infecteds, maximum time  $t_{max}$ . **Output:** Lists times, *S*, *I*, and *R* giving number in each state at each time.

function Gillespie\_network\_epidemic( $G, \mathbf{B}, \mathbf{X}$ , initial\_infections,  $t_{max}$ ) times, S, I,  $R \leftarrow [0], [|G|-len(initial_infections)], [len(initial_infections)], [0]$ infected nodes  $\leftarrow$  initial infections at risk nodes  $\leftarrow$  uninfected nodes with infected neighbours for each node *u* in at risk nodes **do** infection\_rate[u] =  $\mathbf{B} \times$  number of infected neighbours total\_infection\_rate  $\leftarrow \sum_{u \in \text{at risk nodes}} \text{ infection_rate}[u],$ total\_recovery\_rate  $\leftarrow \mathbf{x} \times \text{len(infected_nodes)}$  $\downarrow$  total rate  $\leftarrow$  total transmission rate + total recovery rate time  $\leftarrow$  exponential\_variate(total\_rate)



 $\lambda = \Sigma \lambda i$  $T \sim E_{SQL}(\lambda)$ 



function Gillespie network epidemic( $G, \tau, \gamma$ , initial infections,  $t_{max}$ ) times,  $S, I, R \leftarrow [0], [|G|-len(initial_infections)], [len(initial_infections)], [0]$ infected nodes ← initial infections at risk nodes  $\leftarrow$  uninfected nodes with infected neighbours for each node *u* in at risk nodes **do** infection rate  $[u] = \tau \times$  number of infected neighbours total\_infection\_rate  $\leftarrow \sum_{u \in \text{at risk nodes}} \text{ infection_rate}[u],$ total recovery rate  $\leftarrow \gamma \times \text{len(infected nodes)}$ 2Btotal\_rate  $\leftarrow$  total\_transmission\_rate + total\_recovery\_rate time  $\leftarrow$  exponential variate(total rate) while time  $< t_{max}$  and total\_rate > 0 do  $r = uniform random(0, total_rate)$ if r <total\_recovery\_rate then u = random.choice(infected nodes)Recovery remove *u* from infected nodes reduce infection\_rate[v] for u's susceptible neighbours velse choose u from at\_risk\_nodes with probability  $\frac{\text{Infection_rate}[u]}{\text{total_infection_rate}}$ infection\_rate[u] remove *u* from at risk nodes add *u* to infected nodes for susceptible neighbours v of u do if v not in at risk nodes then add v to at risk nodes update infection\_rate[v] update times, S, I, and R update total\_recovery\_rate, total\_infection\_rate, and total\_rate time  $\leftarrow$  time + exponential\_variate(total\_rate) return times, S, I, R [Kiss, Miller, Simon]



stochastic simulation algorithms for SIR propagation on networks

Gillespie can be slow when there are many "active" processes in a large network

(2) event-driven simulation method

 $Q_i \sim \operatorname{Pois}(\lambda_i) \qquad (1 \le i \le l)$ 

infection Gransmussion Los No

60i

- observation: when a node gets infected, times for further infection transmission events and the recovery event can be computed independent from anything else that will happen in the network
- compute these times, and store them in a priority queue (heap) (unless an event won't have an effect)
- process events in the priority queue in order (generating more events as needed)
- <u>efficient</u>: when an event happens, we know immediately what kind of event it is (and which node)



### max-heap: adding an element









( min - bear and)



**Input:** Network G, per-edge transmission rate  $\tau$ , recovery rate  $\gamma$ , set of index node(s) initial\_infecteds, and maximum time  $t_{\text{max}}$ .

**Output:** Lists times, *S*, *I*, and *R* giving number in each state at each time.

```
- D event - based
function fast SIR(G,\tau,\gamma, initial infecteds, t_{max})
     times, S, I, R \leftarrow [0], [|G|], [0], [0]
     Q \leftarrow empty priority queue

      u.status \leftarrow susceptible
      u.pred_{inf_time} \leftarrow \infty
      v.pred_{inf_time} \leftarrow \infty
      v.pred_{inf_time} \leftarrow \infty
      v.pred_{inf_time} \leftarrow 0

      u in initial_infecteds do
      v.pred_{inf_time} \leftarrow 0
      v.pred_{inf_time} \leftarrow 0

     for u in G.nodes do
     for u in initial infecteds do
           add Event to O
                                                                                                      \triangleright ordered by time
     while Q is not empty do
          Event \leftarrow earliest remaining event in Q
           if Event.action is transmit then
                if Event.node.status is susceptible then
                     process_trans_SIR(G, Event.node, Event.time, \tau, \gamma, times, S, I, R, Q, t<sub>max</sub>)
          else
                process rec SIR(Event.node, Event.time, times, S, I, R)
     return times, S, I, R
```



```
function process trans SIR(G, u, t, \tau, \gamma, times, S, I, R, O, t<sub>max</sub>)
    append times, S, I, and R with t, S.last-1, I.last+1, and R.last
    u.status \leftarrow infected
    u.rec_time \leftarrow t+exponential_variate(\gamma)
    if u.rec time < t_{max} then
        newEvent \leftarrow {node: u, time: u.rec_time, action: recover}
        add newEvent to Q
    for v in G.neighbours(u) do
        find trans SIR(Q, t, \tau, u, v, t_{max})
function find_trans_SIR(Q, t, \tau, source, target, t_{max})
    if target.status is susceptible then
        inf time \leftarrow t+exponential variate(\tau)
        if inf_time < minimum(source.rec_time, target.pred_inf_time, t_{max}) then
            newEvent \leftarrow {node: target, time: inf time, action: transmit}
            add newEvent to Q
            target.pred inf time \leftarrow inf time
function process_rec_SIR(u, t, times, S, I, R)
    append times, S, I, and R with t, S.last, I.last-1, and R.last+1
    u status \leftarrow recovered
                                                                            [Kiss, Miller, Simon]
```



### B3: SIR spread on RSNs: simulations

- small-world Random Spatial Network
  - most connections are local
  - small fraction of long-range connections
  - MOVIE
  - observe: small-world effects
  - observe: traveling waves!





# 8(d) 4 SIR spread on RSNs: simulations local connections only: $p_{ij} = \min\left(\frac{\kappa_i \kappa_j}{\rho < \kappa} f(d_{ij}), 1\right)$

FIG. 2. Disease spread in spatial networks with only short range connections  $[f(d) = 1/0.05^2 \pi$  for d < 0.05 and 0 otherwise] and varying values of  $\rho$ . Top:  $\rho = 10000$ , Middle:  $\rho = 50000$ , and Bottom:  $\rho = 250000$ . Disease is introduced to all nodes in a small region in the center, with  $\beta = 1/3$  and  $\gamma = 1$ . We plot the status of all nodes at time t = 3, including a detailed. As  $\rho$  increases, the spread is more coherent and we expect a deterministic model to be more accurate.



large - p limit: continuum approximation (calerent, isotropic)



### B4: exact analytic PIDE models for SIR propagation on RSNs

- RSNs are "locally tree-like": probability a node is in a cycle of length D vanishes as n increases
- let's derive an exact SIR differential equation model in the limit of high node density! all & neighbors have not transmitted &
- consider a random node u

 $\Theta(\mathbf{x},t)$  : probability a random neighbor of u has not transmitted infection to u

probability a node with  $\kappa_u$  is susceptible (has not been transmitted to):

$$\sum_{k=0}^{\infty} e^{-\kappa_u} \kappa_u^k \Theta^k / k! = \exp[-\kappa_u (1 - \Theta)]$$

$$\int_{k=0}^{\infty} e^{-\kappa_u} \kappa_u^k \Theta^k / k! = \exp[-\kappa_u (1 - \Theta)]$$

$$\int_{k=0}^{\infty} e^{-\kappa_u} \kappa_u^k \Theta^k / k! = \exp[-\kappa_u (1 - \Theta)]$$

$$\int_{k=0}^{\infty} e^{-\kappa_u} e^{-\kappa_u} (1 - \Theta)$$

$$\int_{k=0}^{\infty} e^{-\kappa_u} e^{-\kappa_$$



evolution equation for  $\Theta(\mathbf{x},t)$ 

$$\Psi(\Theta) = \int_0^\infty e^{-\kappa(1-\Theta)} P(\kappa) \,\mathrm{d}\kappa$$
$$\Theta = \Phi_S + \Phi_I + \Phi_R$$
$$\Phi_S = \int_V S(\hat{\mathbf{x}}, 0) \int_\kappa \frac{\kappa P(\kappa)}{\langle \kappa \rangle} f(|\hat{\mathbf{x}} - \mathbf{x}|) e^{-\kappa(1-\Theta(\hat{\mathbf{x}}, t))} \,\mathrm{d}\kappa \,\mathrm{d}\hat{\mathbf{x}}$$





- evolution equation for  $\Theta({f x},t)$ 

$$\begin{split} \frac{\partial}{\partial t} \Theta(\mathbf{x}, t) &= -\beta \Theta(\mathbf{x}, t) + \gamma (1 - \Theta(\mathbf{x}, t)) \\ &+ \beta \frac{\int_V S(\hat{\mathbf{x}}, 0) \Psi'(\Theta(\hat{\mathbf{x}}, t)) f(|\hat{\mathbf{x}} - \mathbf{x}|) \, \mathrm{d}\hat{\mathbf{x}}}{\langle \kappa \rangle} \end{split}$$

$$\begin{split} S(\mathbf{x},t) &= S(\mathbf{x},0)\Psi(\Theta(\mathbf{x},t)), \\ \frac{\partial}{\partial t}R(\mathbf{x},t) &= \gamma(1-S(\mathbf{x},t)-R(\mathbf{x},t)) \\ I(\mathbf{x},t) &= 1-S(\mathbf{x},t)-R(\mathbf{x},t) \end{split}$$







$$\frac{\partial \Theta(x_j, t)}{\partial t} = -\beta \Theta(x_j, t) + \gamma (1 - \Theta(x_j, t)) + \beta \sum_{i=j-N_{pad}}^{j+N_{pad}} e^{-\kappa (1 - \Theta(x_i, t))} f(d(x_i, x_j))(1 - \epsilon(x_i)) \Delta x$$

### B5: traveling wave properties

$$\Theta(x,t) = \theta(\xi(x,t)) \qquad \xi(x,t) = x - ct$$
  
$$\Theta(x,t) = \theta(\xi(x,t)) = 1 - \epsilon h(\xi(x,t)) \qquad h(\xi) \approx e^{-\alpha \xi}$$





B6: numerical experiments - "small worlds" with spatial structure

MOVIES

$$p_{ij} = \min\left(\frac{\kappa_i \kappa_j}{\rho < \kappa >} f(d_{ij}), 1\right)$$

- local connections
- local + global connections: small worlds, also without clustering (with "proximity")
- random connections



### small world networks with spatial structure (RSNs)

recall: Watts-Strogatz "small world" networks 



- small world:
  - large average local clustering coefficient —
  - small average (shortest) path length \_



### small world networks with spatial structure (RSNs)

• RSNs: define local proximity coefficient

- first normalize all distances:  $\overline{d}_{ij}$  normalized to [0, 1]

- define local proximity coefficient:

 $p_i = 1 - \operatorname{avg}(\overline{d}_{ij} \text{ of graph neighbors } j \text{ of } i)$   $p_i \in [0, 1]$  $p_i \approx 1$ : graph neighbors of i are located close  $p_i \approx 0$ : graph neighbors of i are located far

average local proximity coefficient:



### small worlds with spatial structure , LC> proximity 8 *C*(*p*) / *C*(0) 0.8 0.6 0.4 \* п \*\* L(p) / L(0)0.2 0.8 0.8 \* \*\* ж 0.0001 0.001 0.01 0.1 0.6 0.6 ⋇ р \*\* 0.4 0.4 \* \*\*\*\*\*\*\*\*\*\*\*\*\* 0.2 0.2 0 0 10<sup>-8</sup> 10<sup>-10</sup> 10<sup>-6</sup> $10^{-4}$ 10<sup>-10</sup> 10<sup>-8</sup> $10^{-6}$ $10^{-4}$ $\epsilon$ $\epsilon$

- low density,  $N=N_0$ : small world
  - high clustering, high proximity
  - low (average) shortest path

- high density, N>>N<sub>0</sub>: unclustered small world
  - low clustering
  - high proximity
  - low (average) shortest path

still small world! (local, and global, structure)





small-world effect:

- local propagation (traveling wave) due to local structure (proximity, not clustering)
- long-range jumps due to small-world property

for spatial networks, proximity is important in determining whether small-world effects occur, rather than clustering



### B7: summary and conclusions

- random networks with spatial structure (RSN)
- efficient network generation algorithms
- efficient stochastic simulation algorithms
- analytical equations may be derived, may provide insight (traveling waves, parameter regimes, ...)
- applications: spreading processes on spatial networks
  - realistic spatial connectivity patterns, degree distributions
  - Ebola, plague, ...







Density

Cumulative



[Barabasi]

### some final words ...

- network science is
  - interesting (also algorithmically)
  - potentially powerful, for broad applications
  - links with P(I)DEs, waves, ...
  - potential applications in computational social science

thank you!

