Fast and near-optimal monitoring for healthcare acquired infection outbreaks

Thu 4/23

CS:4980 Computational Epidemiology

PLOS COMPUTATIONAL BIOLOGY

RESEARCH ARTICLE

Fast and near-optimal monitoring for healthcare acquired infection outbreaks

Bijaya Adhikari[1]*, Bryan Lewis[2], Anil Vullikanti[2,3], José Mauricio Jiménez[4], B. Aditya Prakash[1]*

Published in Sep 2019.

Side note: Bijaya Adhikari is joining our department this fall!

Overview

The paper has 5 parts:

1. Overall goal
2. Modeling and simulation
3. Modeling as optimization problems
4. Approximation algorithms for optimization problems
5. Results

Part I: Overall goal

- Let P denote the set of human agents and L denote the set of locations.
- Let $n = |P \cup L|$

Goal: Find a *rate vector* $\boldsymbol{r} = (r[1], r[2], \ldots, r[n])$, where $r[v]$denotes the rate at which an "agent" $v \in P \cup L$ is monitored, that

- maximizes the probability of detecting an infection or
- moves the detection day forward in time as much as possible.

Notes: (a) $r[v]$ is the probability that agent v will be monitored in a day.

(b) Monitoring could mean testing a stool sample or swabbing a surface.

Part I: Overall goal

- The problem would be trivial, if we were allowed to make the rate vector as high as possible (e.g., $\boldsymbol{r} = (1, 1, \ldots, 1)$).

- There is a given *cost vector* $\boldsymbol{c} = (c[1], c[2], \ldots, c[n])$ that associates with each agent v, a cost $c[v]$ of monitoring that agent.

- Then

$$c[1]r[1] + c[2]r[2] + \ \ldots + c[n]r[n]$$

is the expected per day cost of monitoring agents according the chosen rate vector $\boldsymbol{r}$.

- We are given a *budget* B and it is required that

$$c[1]r[1] + c[2]r[2] + \ \ldots + c[n]r[n] \leq B$$

Questions on Part I

- Does this overall goal make sense to you?
- How should we take into account the fact that hospital population is changing as patients get discharged and new patients are admitted?
- Should the rate vectors be dynamic, i.e., change over time for a particular agent?
- Any other aspects you think should be modeled in this problem?

Part II: Modeling and simulation

(a) Contacts

Table 1. Summary statistics on the mobility log and the resulting social networks.

total no. of locations	72,146
total no. of agents	96,281
total no. of days	200
average no. of mobility entries per day	138,765.4 ± 6384.15
average no. of visits per location	384.6 ± 1949.22
average no. of nodes in social networks	6,924.8 ± 96.51
average no. of edges in social networks	45,503.8 ± 4267.97
average degree in social networks	13.2 ± 1.44
average clustering co-efficients in social networks	0.4 ±0.03

https://doi.org/10.1371/journal.pcbi.1007284.t001

Questions: How is this table generated? What data is it based on? What types of agents/locations are included?

Part II: Modeling and simulation

(b) Disease model

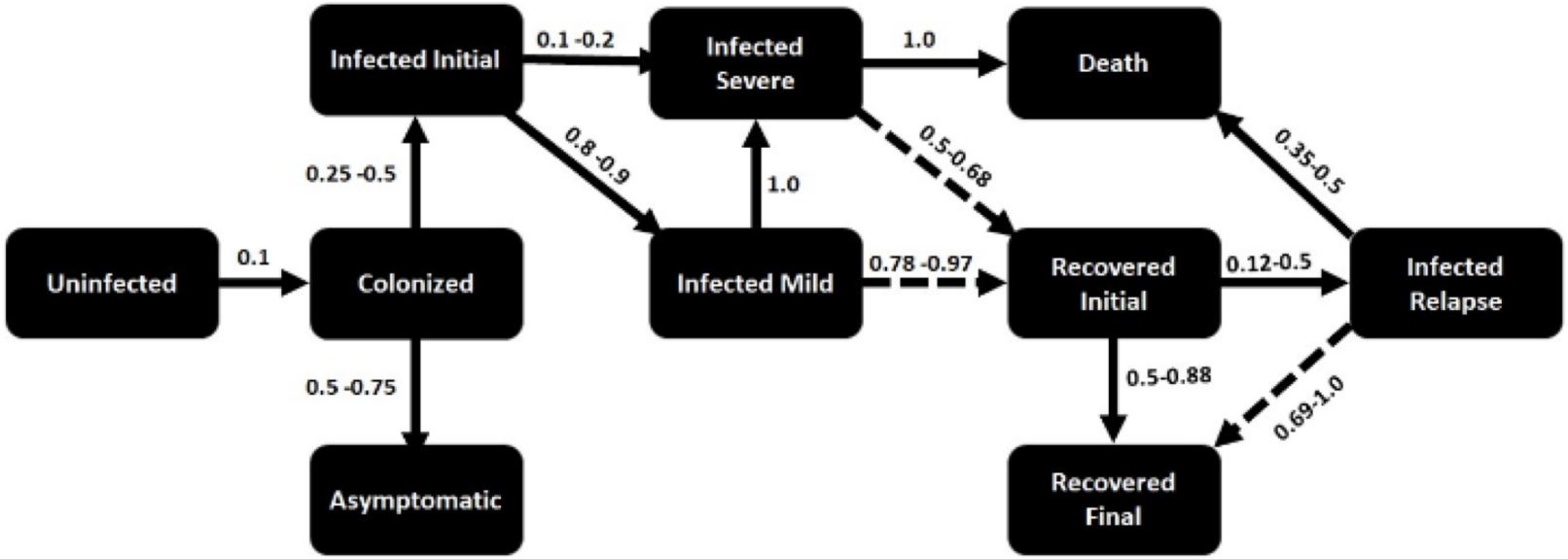

Fig 2. Human infection model for *C. difficile*. Each state in the finite state machine shown above indicates the stages in the infection/recovery process. The arrows indicate possible transition in state and the weight on the arrows indicate the transition probability. The dashed arrows represent transition under medication.

https://doi.org/10.1371/journal.pcbi.1007284.g002

Questions: Does this disease model for C.difficile make sense? What data is it based on? How are the transition probabilities inferred?

Part II: Modeling and simulation

(c) Pathogen load model

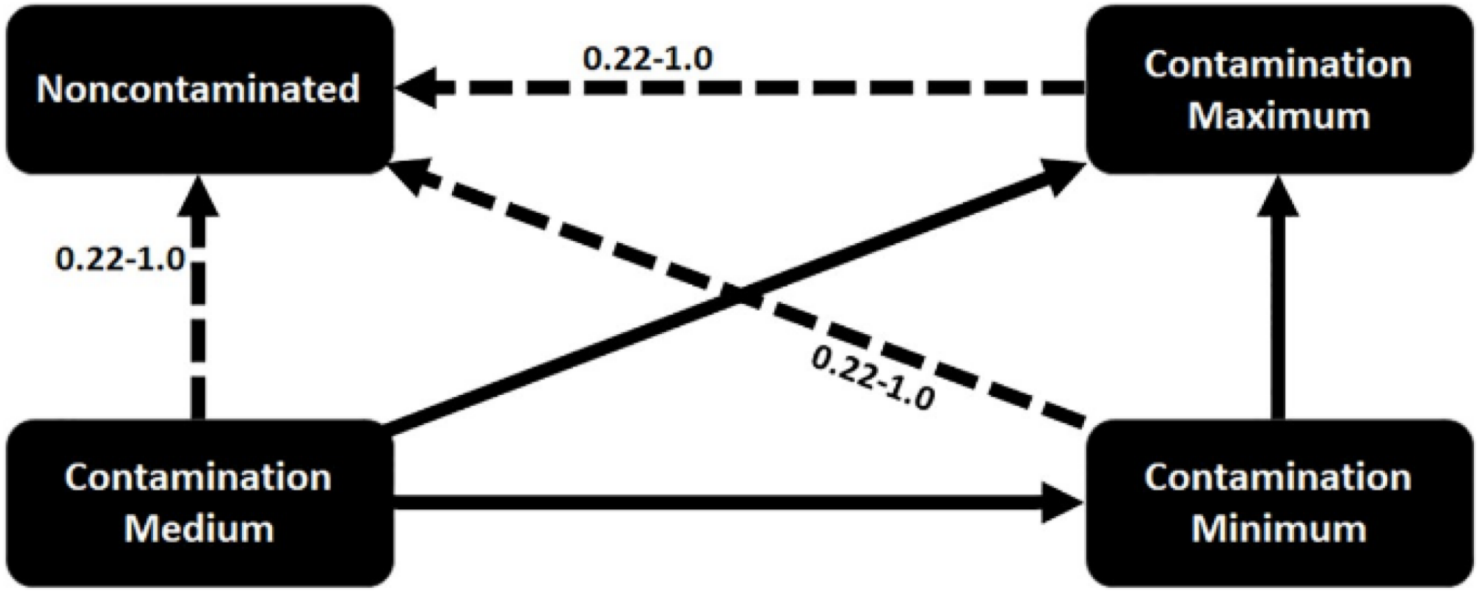

Fig 3. Fomite contamination model for *C. difficile*. Each state in the finite state machine shown above indicates the stages in the contamination/decay process. The solid arrows indicate possible transition in state. The transition between the states depend on number of infected people in the location. The dashed arrows represent transition assuming cleaning.

https://doi.org/10.1371/journal.pcbi.1007284.g003

Questions: Does this model for pathogen load make sense? What data is it based on? How does the transition probability depend on number of infected people? Do they have to be severely infected? Asymptomatic?

Part III: Modeling as optimization problems

- Run a bunch of simulations. Each *simulation instance* i is the output of a particular simulation, consisting of who got infected, when, and pathogen load on locations over time.
- Let $\mathcal{I}$ be the set of all simulation instances. These form the input to our optimization problems.
- For an agent $v \in P \cup L$ and simulation instance $i \in \mathcal{I}$, let $\tau(v, i)$ denote the number of days v was infected in simulation instance i.
- Then the probability of *detecting* v in a given simulation instance i, given a rate vector $\boldsymbol{r}$, is

$$P_d(v|i, \boldsymbol{r}) = 1 - (1 - r[v])^{\tau(v,i)}$$

Part III: Modeling as optimization problems

- Then the probability of detecting *some* infected human agent in simulation instance i, given a rate vector $\boldsymbol{r}$, is

$$P_d(i, \boldsymbol{r}) = 1 - \prod_{v \in P \cup L} (1 - P_d(v|i, \boldsymbol{r}))$$

- Plugging in the expression for $P_d(v|i, \boldsymbol{r})$, this simplifies to

$$P_d(i, \boldsymbol{r}) = 1 - \prod_{v \in P \cup L} (1 - r[v])^{\tau(v,i)}$$

Part III: Modeling as optimization problems

Maximizing Detection Probability (MDP) problem

Find $\boldsymbol{r}$ that maximizes

$$F(\boldsymbol{r}) := \sum_{i \in \mathcal{I}} P_d(i, \boldsymbol{r})$$

subject to

$$\sum_{v=1}^{n} c[v] r[v] \leq B.$$

Questions: What is this problem saying? Is there a danger of "overfitting" to the simulations? Are there other aspects that should be considered in this problem formulation?

Note: The **Early Detection** (ED) problem is also formulated as an optimization problem. Read about it.

Part IV: Approximation algorithms for optimization problems

- Both MDP and ED are NP-hard (no surprise there!)
- So we look for approximation algorithms (i.e., heuristics with guarantees on error).
- For this we take a detour into *submodular functions*.

Definition: Let Ω be a finite set. A function $f: 2^{\Omega} \longrightarrow \mathbb{R}$ is a *submodular set function* if it satisfies the following *diminishing marginal* returns property:

For every $X, Y \subseteq \Omega$, where $X \subseteq Y$, and every $x \in \Omega - Y$,

$$f(X \cup \{x\}) - f(X) \geq f(Y \cup \{x\}) - f(Y)$$

Part IV: Approximation algorithms for optimization problems

Example: The *coverage function* is submodular

Let $S_1 = \{a, b, e\}, S_2 = \{c, d, e\}, S_3 = \{a, c\}, S_4 = \{a, d, e\}, S_5 = \{a, f\}$ be arbitrary subsets of $U = \{a, b, c, d, e, f\}$.

Define $f: 2^{\{1,2,3,4,5\}} \to \mathbb{R}$ as $f(X) = |\bigcup_{i \in X} S_i|$.

Note: $f(X)$ is the size of coverage of the subsets indexed by X.

So $f(\{3,5\}) = |S_3 \cup S_5| = |\{a, c, f\}| = 3$.

So $f(\{1,4\}) = |S_1 \cup S_4| = |\{a, b, d, e\}| = 4$.

Question: f is submodular. Why?

Part IV: Approximation algorithms for optimization problems

What do submodular functions have to do with anything?

For any submodular set function f, the problem

$$\text{maximize } f(X)$$
$$|X| \leq B$$

has a simple, greedy approximation algorithm.

Example: The MaxCoverage problem

Given a collection of sets $S_1, S_2, \ldots, S_n$, find a subcollection of B sets $S_{i_1}, S_{i_2}, \ldots, S_{i_B}$ such that $|S_{i_1} \cup S_{i_2} \cup \cdots \cup S_{i_B}|$ is maximized.

Part IV: Approximation algorithms for optimization problems

Cost-effective Outbreak Detection in Networks

Jure Leskovec
Carnegie Mellon University

Andreas Krause
Carnegie Mellon University

Carlos Guestrin
Carnegie Mellon University

Christos Faloutsos
Carnegie Mellon University

Jeanne VanBriesen
Carnegie Mellon University

Natalie Glance
Nielsen BuzzMetrics

- Appeared in KDD 2007

- They show that placing a few “sensors” in a network
  - network of water pipes in a city
  - network of blogs that link to each other

to maximize probability of detecting water contamination or a viral piece of news is equivalent to the problem of maximizing a submodular function subject to a budget constraint.

- This is the connection to disease-surveillance.

Part IV: Approximation algorithms for optimization problems

Simple, greedy algorithm?

$X \leftarrow \emptyset$

while $|X| \leq B$ **do**

Pick an $x \in U - X$ that maximizes $f(X \cup \{x\}) - f(X)$

$X \leftarrow X \cup \{x\}$

- This algorithm guarantees a $\left(1 - \frac{1}{e}\right) \approx 0.632$ approximation.

- In other words, even in the worst case this algorithm is guaranteed to produce a set X such that $f(X)$ is at least 63% as large as $f(X^*)$, where X^* is an optimal set.

Part IV: Approximation algorithms for optimization problems

Maximizing Detection Probability (MDP) problem

Find $\boldsymbol{r}$ that maximizes

$$F(\boldsymbol{r}) := \sum_{i \in \mathcal{I}} P_d(i, \boldsymbol{r})$$

subject to

$$\sum_{v=1}^{n} c[v] r[v] \leq B.$$

- The objective function is a function of the rate vector $\boldsymbol{r} \in \mathbb{R}^{\boldsymbol{n}}$.
- The authors assume that each rate can take a discrete value, say,

$$L = \left\{ \frac{0}{100}, \frac{1}{100}, \cdots, \frac{99}{100}, \frac{100}{100} \right\}$$

- So $\boldsymbol{r} \in L^n$ and $F(\boldsymbol{r})$ is a function over a *discrete lattice*.

Part IV: Approximation algorithms for optimization problems

- The authors show that $F(\boldsymbol{r})$ has the diminishing returns property in the following sense.

For every $\boldsymbol{x}$, $\boldsymbol{y}$, such that $\boldsymbol{x} \preccurlyeq \boldsymbol{y}$, for every e_i, $1 \leq i \leq n$,
$F(\boldsymbol{x} + e_i) - F(\boldsymbol{x}) \geq F(\boldsymbol{y} + e_i) - F(\boldsymbol{y})$

Note: (i) $\boldsymbol{x} \preccurlyeq \boldsymbol{y}$ means every element of $\boldsymbol{x}$ is less than or equal to the corresponding element in $\boldsymbol{y}$. (ii) e_i is the length-n vector with $\frac{1}{100}$ at index i and 0's everywhere else.

- F is called a *submodular lattice function*.
- A simple, greedy approximation algorithm exists for maximizing submodular lattice functions, subject to the budget constraint.

Part IV: Approximation algorithms for optimization problems

Algorithm 1 HAIDETECT

Require: I, budget B

1: **for** each feasible initial vector $\mathbf{r}_0$ **do**
2: Initialize the rate vector $\mathbf{r} = \mathbf{r}_0$
3: **while** $\sum_v \mathbf{r}[v] \cdot \mathbf{c}[v] < B$ **do**
4: Find a node v and rate r maximizing average marginal gain
5: Let $\mathbf{r}[v] = r$
6: Remove all candidate pairs of nodes and rates which are not feasible
7: Return the best rate vector $\mathbf{r}$

HAIDETECT has desirable properties in terms of both effectiveness and speed. The performance guarantee of HAIDETECT is given by the following lemma.

Lemma 4. HAIDETECT *gives a (1-1/e) approximation to the optimal solution.*

Questions: Try to understand this algorithm. What could they mean by “feasible initial vector”? What does Step 4 mean? What about Step 6?

Part V: Results

- We will not discuss the results today.
- This part of the paper is for you to study carefully. We will discuss on Tuesday.

Thanks for your attention…

Any final questions?