Neo-Phoebe

AzeezDa

June 13, 2023

Notation and conventions

Through this document these notational conventions will be used:

- Uppercase letters ($A, B, C, \ldots$) will denote sets or random variables
- Lowercase letters ($a, b, c, \ldots$) will denote elements in sets
- Fraktur letters ($\mathfrak{A}, \mathfrak{B}, \mathfrak{C}$) will denote set of sets
- Lowercase Greek alphabet ($\alpha, \beta, \gamma, \ldots$) will denote most functions (except for some conventional notation).
- Uppercase Greek alphabet ($\Gamma, \Delta, \Theta, \ldots$) will denote various special structures
- The power set of a set A is denoted by $\mathcal{P}(A)$
- The cardinality of a set A is denoted by $|A|$
- The natural numbers $\mathbb{N}$ include 0
- The set of positive integers are denoted by $\mathbb{Z}_+$
- The set of positive real numbers is denoted by $\mathbb{R}_+$
- The unit interval $[0, 1]$ is denoted by $\mathcal{I}$
- Bern(p) denotes the Bernoulli distribution with parameter p
- Exp(λ) denotes the exponential distribution with parameter λ
- $\mathcal{N}(\mu, \sigma)$ denotes the normal distribution with mean μ and standard deviation σ
- For closed intervals $S = [a, b]$, $\text{clamp}_S(x) = \max(a, \min(x, b))$

1 The Model

The model used to simulate is a structure that is based on a set equipped with a special partition and function.

1.1 Compartmentation Space

Definition 1 (Compartmentation)**.** *A n-compartmentation of a set S is an ordered collection of subsets of S, $(C_1, C_2, \dots C_n) \in \mathcal{P}(S)^n, n \in \mathbb{Z}_+$, such that:*

1. $C_i \cap C_j = \varnothing$, *for all* $1 \leq i, j \leq n, i \neq j$
2. $\bigcup_{i=1}^{n} C_i = S$

Note that Definition 1 is similar to a partition of a set but has a fixed size and is ordered. Therefore the definition enforces that every element of S must always be in one and only one subset in the compartmentation at a time. Also note that throughout this document, the compartmentation subset might written in vector/stacked form:

$$\begin{pmatrix} C_1 \\ C_2 \\ \vdots \\ C_n \end{pmatrix}$$

Now the n-compartmentation of a set is used in the following definition that makes up the whole simulation model.

Definition 2 (Transitional Compartmentation Space)**.** *For a fixed n, a transitional n-compartmentation space is a tuple (S, τ), such that*

- *S is a set*
- *$K = \{(T, \Gamma) \mid T \subseteq S,\ \Gamma$ is a n-compartmentation of $T\}$.*
- *$\tau : K \to K$ is a function called the transition function*

That is really it for the model. Most of the work goes to strictly defining the set Transitional Compartmentation Space of size n, which will be abbreviated as TCS-n.

1.2 An Example

This section considers an example to establish more understanding before moving on to the implementation. Let the set $S = \mathbb{N}$. Let (S, K, τ) be a TCS-2, for a compartmentation labeled (O, E). Let the transition function τ be:

$$\tau(k) = \begin{cases} (T, \begin{pmatrix} O \backslash \{\min(P')\} \\ E \cup \{\min(P')\} \end{pmatrix}) & \text{if } \min(O) \equiv 0 \mod 2 \\ \\ (T, \begin{pmatrix} O \\ E \end{pmatrix}) & \text{otherwise} \end{cases}$$

Consider $T = \{0, 2, 4, 5, 6\}$ and let $O_0 = T$ making $E_0 = \varnothing$. Let us apply τ on $k_0 = (T, \begin{pmatrix} O_0 \\ E_0 \end{pmatrix})$ a couple of times.

1. $k_1 = T(k_0) = (T, \begin{pmatrix} \{2, 4, 5, 6\} \\ \{0\} \end{pmatrix})$
2. $k_2 = T(k_1) = (T, \begin{pmatrix} \{4, 5, 6\} \\ \{0, 2\} \end{pmatrix})$
3. $k_3 = T(k_2) = (T, \begin{pmatrix} \{5, 6\} \\ \{0, 2, 4\} \end{pmatrix})$
4. $k_4 = T(k_3) = (T, \begin{pmatrix} \{5, 6\} \\ \{0, 2, 4\} \end{pmatrix})$
5. It continues to be the same value afterwards...

This example is simple. But the full potential of a TCS-n shines when T is changed during transitions as well.

2 Implementation

This section will explain how the definition under Section 1 are implemented in the simulation.

2.1 Definitions

In this section we will define the parts that build up the simulation.

2.1.1 States and Transition Functions

This simulation will use 5 states: *Susceptible*, *Exposed*, *Contagious*, *Recovered* and *Deceased*. Their transition graph is shown in Figure 1.

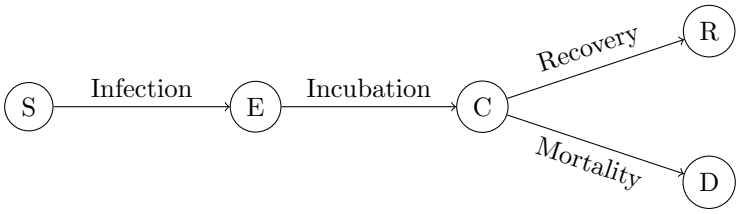

Figure 1: Graph of transitions between states

2.1.2 Population

We define the *population* to be a set P whose elements are called *persons*. Additionally, we let $\Gamma = (P, \tau)$ be a TCS-5 on P with the compartmentation

labeled (S, E, C, R, D) after the states defined in Section 2.1.1.

We will say that the simulation starts with the initial 5-compartmentation of an initial subset of P denoted as $K_0 = (S_0, E_0, C_0, R_0, D_0)$. Then for n applications (which we assume will be done daily) of τ unto K_0 gives K_n. That is $\tau^n(K_0) = K_n = (S_n, E_n, C_n, R_n, D_n)$.

Any function $\phi : A \to B$ that maps a subset A of the population P into any set B is called a *trait*.

2.1.3 Relationship

Let $\rho : P^2 \to \mathcal{I}$ be a function that takes two persons $p, q \in P$ and returns the relationship strength $\rho(p, q)$ of person p onto q.

We can assume that $\rho(p, q) = \rho(q, p)$ (which we will do during programming), but that does not affect the implementation theory. We will also assume that $\rho(p, p) = 0$ for all $p \in P$.

2.1.4 Susceptible to Exposed

Every time unit (which we will assume to be days), a person p has a risk of being infected based on the persons p is related to. The probability of p being infected will be calculated using the relationship strength defined in Section 2.1.3.

First, let $\tilde{\pi} : P^2 \to \mathcal{I}$ be a function such that for persons $p, q \in P$, $\tilde{\pi}(p, q)$ is the probability that person p gets infected by person q. We will set $\tilde{\pi}(p, q) = s\rho(p, q)(1 - \alpha(p))(1 - \alpha(q))1_{C_n}(q)$, where $s \in \mathcal{I}$ is a constant representing the strength of disease spread, $\alpha : P \to \mathcal{I}$ is a trait representing a persons *protectiveness* or *hygiencity* and $1_C(q)$ is the indicator function that returns 1 if $q \in C_n$ (q is contagious at day n) and 0 otherwise.

Now a person p is considered infected if

$$1 = I_p = \max\{I_{p,q} : q \in P, I_{p,q} \in \text{Bern}(\tilde{\pi}(p, q))\}$$

That is, p getting infected by any person q is Bernoulli distributed with parameter $\tilde{\pi}(p, q)$. And p gets infected if any of those Bernoulli trials succeed.

Finding the distribution of I_p is simple, if we assume that the I_ps are independent. Note that I_p is 0 if all $I_{p,q}$ are 0. Therefore it would easier to find

$\mathbb{P}(I_p = 0)$ to find the distribution and is done thusly:

$$\begin{aligned}\mathbb{P}(I_p = 0) &= \mathbb{P}(\max\{I_{p,q} : q \in P\} = 0) = \\ &= \mathbb{P}\left(\bigwedge_{q\in P} I_{p,q} = 0\right) = \\ &= \prod_{q\in P} \mathbb{P}(I_{p,q} = 0) = \\ &= \prod_{q\in P} (1 - \tilde{\pi}(p, q))\end{aligned}$$

Now because I_p is either 0 or 1, $\mathbb{P}(I_p = 1) = 1 - \mathbb{P}(I_p = 0) = 1 - \prod_{q\in P}(1 - \tilde{\pi}(p, q))$. This result gives that I_p is also Bernoulli distributed with the parameter $1 - \prod_{q\in P}(1 - \tilde{\pi}(p, q))$ which we will denote by $\pi(p)$, i.e. $I_p \in \text{Bern}(\pi(p))$.

2.1.5 Exposed to Contagious

For every exposed person $p \in E_n$ we define the trait $\delta_{E_n} : E_n \to \mathbb{N}$ that counts the amount of days an exposed person has been exposed thusly:

$$\delta_{E_n}(p) = \sum_{i=0}^{n-1} 1_{E_i}(p)$$

That is, it adds up 1's for every previous compartmentations where p was in E (exposed). Note that programmatically we will implement this very differently, by simply having a field in a struct that is incremented by one each iteration.

For every exposed person $p \in E_n$ we will assign an incubation time value given by $C_p\tilde{\iota}(p)$, where $C_p \in \text{Exp}(\lambda)$, λ is some constant that corresponds to *disease incubation time* and $\iota : P \to \mathbb{R}_+$ is a trait that corresponds to some *personal incubation time scaling factor* (depending on e.g. age or health status).

Now we will say that a person p has transitioned from Exposed to Contagious at day n if $\delta_{E_n}(p) > C_p\iota(p)$.

2.1.6 Contagious to Recovered or Deceased

Similarly to Section 2.1.5, we define the day-counting trait $\delta_{C_n} : C_n \to \mathbb{N}$ but now for Contagious persons thus:

$$\delta_{C_n}(p) = \sum_{i=0}^{n-1} 1_{C_i}(p)$$

We also define (again similar to Section 2.1.5) two Exp-distributed random variables that describe a person's time as Contagious before getting Recovered or Deceased.

For every Contagious person $p \in C_n$ we will assign $K_p\kappa(p)$ for $K_p \in \text{Exp}(\lambda_R)$, $\lambda_R \in \mathbb{R}_+$ corresponding to *average time to get cured* and $\kappa(p)$ (of course $\kappa : P \to \mathbb{R}_+$) as a *personal scaling factor for time to get cured.*

Again, for every contagious person we give $M_p\mu(p)$ where $M_p \in \text{Exp}(\lambda_D)$, λ_D is constant corresponding to *average time before mortality* and $\mu(p)$ (same codomain as κ) is *personal scaling factor for time to reach the deceased state.*

Now to summarise the transitions from the Contagious state at day n we define this function:

$$\beta_n(p) = \begin{cases} 1 & \text{if } K_p\kappa(p) \leq M_p\mu(p) \text{ and } \delta_{C_n}(p) \geq K_p\kappa(p) \\ -1 & \text{if } K_p\kappa(p) > M_p\mu(p) \text{ and } \delta_{C_n}(p) \geq M_p\mu(p) \\ 0 & \text{otherwise} \end{cases}$$

The function is equal to 1 if the time to get cured is less than or equal the time to become deceased and the amount days a person has been contagious is greater than the time to get cured. If, however, the time to become deceased is less than the time to get cured and the amount of contagious-days for a person is greater than the deceased-days, then β_n evaluates to -1.

Now we day a person p, at day n has transitioned from Contagious to Recovered if $\beta_n(p) = 1$; and from Contagious to Deceased if $\beta_n(p) = -1$; otherwise no transitions happened for person p.

2.2 Putting the Pieces Together

Finally, we can define the transition τ. Assume that at day n the compartmentation of P is $(S_n, E_n, C_n, R_n, D_n)$. The transition to day $n+1$ is done thusly:

$$\begin{aligned} S_{n+1} &= \{p \in S_n : I_p = 0\} \\ E_{n+1} &= \{p \in E_n : \delta_n(p) \leq C_p\iota(p)\} \cup \{p \in S_n : I_p = 1\} \\ C_{n+1} &= \{p \in C_n : \beta_n(p) = 0\} \cup \{p \in E_n : \delta_n(p) > \iota(p)\} \\ R_{n+1} &= R_n \cup \{p \in C_n : \beta_n(p) = 1\} \\ D_{n+1} &= D_n \cup \{p \in C_n : \beta_n(p) = -1\} \end{aligned}$$

Or put in words, The Susceptible persons at day $n+1$ are those from day n that are not infected. The Exposed persons at day $n+1$ are persons who are still in their incubation period together with those Susceptible people at day n that have been infected. At day $n+1$ the Contagious are those at day n that did not Recover or Decease together with those that have exceeded their incubation period. Lastly the newly Recovered and Deceased at day $n+1$ are the Recovered and Deceased from the previous day together with those that Recovered or Deceased respectively.

2.3 Extensions: Disease Testing

There is one last thing I will add to the model before programming it and that is a testing. It will be quite messy to model with mathematical notation so I will describe its implementation informally.

We will give the simulation a list of tuples (t, p, r, d) corresponding to a testing "plan", such that t is the number of samples we will take from the population to test if they are Exposed or Contagious. If the number of positive tests is at least p then we apply the restrictionthe restriction is applied by having r as a factor in the $\tilde{\pi}$ function for everyone. This restriction will apply for d days (iterations). If there are multiple applicable restrictions, we will choose the "strongest" restriction, i.e. the smallest r amongst applicable tests.

Another testing plan which we can give is by the tuple (t, r, d), where we test