Pandemia: Default Health Model

James Thompson

October 2022

Introduction

Epidemic models typically represent health using discrete states. In a typical compartmental model, the population is partitioned into subsets, labelled ‘Susceptible’, ‘Infected’, ‘Recovered’ and ‘Dead’. These compartments can then be subdivided, and new compartments added, to produce increasingly complicated models.

This approach runs into difficulties once partial immunity is introduced, since the compartmental label ‘Recovered’ then becomes ambiguous. It runs into further difficulties once multiple strains of the pathogen are introduced, since then the label ‘Susceptible’ becomes ambiguous.

Pandemia therefore dispenses with the compartmental framework altogether, taking an entirely new approach to modelling individual health.

Attributes

In Pandemia, the health of an individual is described by five attributes:

- **Strain**
- **Disease**
- **Infectiousness**
- **Immunity (outer layer)** (determining whether or not an infection is blocked)
- **Immunity (inner layers)** (determining the outcome, for infections that are not blocked)

A key innovation of Pandemia is that, for each individual, these five attributes are stored as *functions*.

Variables

The value of an attribute at time `t` is given by evaluating the corresponding function at time `t`.

During a simulation, the following variables store the values of these functions at the current time:

- `current_strain`
- `current_disease`
- `current_infectiousness`
- `current_sigma_immunity_failure`
- `current_rho_immunity_failure`

The prefix `sigma_` refers to the outer layer, while the prefix `rho_` refers to the inner layers.

current_strain

For an individual n, the variable

$$\texttt{current\_strain[n]}$$

indicates whether or not individual n is infected, and if so with which strain. The variable is an integer, taking values in the range

$$\{\texttt{-1, 0, 1, 2, ..., S - 1}\}$$

where S denotes the number of strains. If the individual is not infected, then

$$\texttt{current\_strain[n] = -1.}$$

We assume that individuals can only be infected with one strain at a time.

current_disease

For an individual `n`, the variable

```
current_disease[n]
```

indicates the extent to which individual `n` is diseased. The variable is a float, taking values in the range `[0,1]`. If the individual has no disease, then

```
current_disease[n] = 0.
```

If the individual is dead, then

```
current_disease[n] = 1.
```

Values close to `1` represent severe illness, values close to `0` represent mild illness. Values above a threshold represent symptomatic illness, values below the threshold represent asymptomatic illness.

current_infectiousness

For an individual n, the variable

$$\texttt{current\_infectiousness[n]}$$

indicates the extent to which individual n is infectious. The variable is a float, taking non-negative values. If the individual is infected but not infectious, then

$$\texttt{current\_infectiousness[n]} = 0.$$

If the individual is infected, then increasing the value of this variable increases the probability that the individual transmits strain current_strain[n] to other individuals.

current_sigma_immunity_failure

For an individual `n` and strain `s`, the variable

`current_sigma_immunity_failure[n][s]`

represents the probability that the immune system of individual `n` *fails* to prevent an infection when confronted with strain `s`. Being a probability, this variable is therefore a float taking values in the range `[0,1]`. Since this variable stores a failure probability, values closer to `0` represent higher levels of protection.

current_rho_immunity_failure

If an infection has occurred, then sigma immunity has failed and the pathogen has made it past the outer layer of defence. It now confronts a number of internal layers.

Each layer has a probability of failing to stop the pathogen, with the outcome of the infection getting progressively worse the deeper the pathogen penetrates.

This binary tree structure allows the user to parametrize, for example, the efficacy of a vaccine against a range of possible outcomes (such as symptomatic illness, severe illness and death).

If the pathogen passes through all but the last internal layer, which is impenetrable, then the individual experiences the worst possible outcome.

For example, suppose the model is configured in such a way that there are two internal layers.

Then, for a given individual, infected with a given strain, there are two possible outcomes. If the first internal layer fails to stop the pathogen, the outcome will be the latter of the two outcomes. If the first internal layer is successful, then the outcome will be the former of the two outcomes.

These two outcomes could, for example, be configured to represent mild and severe illness.

current_rho_immunity_failure

More generally, suppose the model is configured in such a way that there are R internal layers (and therefore R possible outcomes).

Then, for an individual n and strain s, the variable

```
current_rho_immunity_failure[n][s]
```

gives an array of probabilities, of length R, corresponding to the *failure* probabilities of each layer. If a pathogen makes it past layer r, then it moves on to face layer r+1, else the individual gets outcome r. Since the last layer is impenetrable, the final entry in this array is always set equal to 0, meaning that the last layer never fails to stop the pathogen.

health_presets

While sigma immunity determines whether or not an infection occurs, with rho immunity determining the outcome of that infection, to understand the outcomes themselves we must discuss the `health_presets`.

Pandemia determines exactly how each individual will respond to an infection, for each strain, in advance of the simulation.

If an individual has been assigned the preset response `p`, then the object `p[r]` contains parameters, which determine updates for each of the five health attributes described above, corresponding to outcome `r`.

For example, suppose that our model features two strains and two internal immune layers. Then the object `p[r]` may look as follows:

```
rho_immunity_failure:
  [[[[-1, 5, 30], [[1.0, 0.0], [0.25, 0.0], [1.0, 0.0]]],
    [[-1, 5, 30], [[1.0, 0.0], [0.50, 0.0], [1.0, 0.0]]]],
   [[[-1, 6, 30], [[1.0, 0.0], [0.50, 0.0], [1.0, 0.0]]],
    [[-1, 6, 30], [[1.0, 0.0], [0.25, 0.0], [1.0, 0.0]]]]]
sigma_immunity_failure:
  [[[[-1, 5, 30], [1.0, 0.25, 1.0]],
    [[-1, 5, 30], [1.0, 0.50, 1.0]]],
   [[[-1, 6, 30], [1.0, 0.50, 1.0]],
    [[-1, 6, 30], [1.0, 0.25, 1.0]]]]
infectiousness:
  [[[-1, 0, 3, 5], [0.0, 0.0025, 0.0075, 0.0]],
   [[-1, 0, 3, 6], [0.0, 0.0055, 0.0085, 0.0]]]
disease:
  [[[-1, 0, 5], [0.0, 0.2, 0.0]],
   [[-1, 0, 6], [0.0, 0.3, 0.0]]]
strain:
  [[[-1, 0, 5], [-1, 0, -1]],
   [[-1, 0, 6], [-1, 1, -1]]]
```

The numbers in these arrays encode step functions.

health_presets

Suppose in this example that an individual `n` has just been infected with strain `0`, and that the infection has resulted in the above outcome.

Then the variable `current_strain[n]` will take the value `0` for the next `5` days, at which time it returns to the value `-1`, indicating that the individual is no longer infected.

The variable `current_disease[n]` will take the value `0.2` for the next `5` days, at which time it returns to the value `0.0`, indicating that the individual has recovered.

The variable `current_infectiousness[n]` will take the value `0.0025` for the next `3` days, followed by `0.0075` for `2` days, after which time it returns to the value `0.0`, indicating that the individual is no longer infectious.

Updates to the sigma and rho immunity variables are more complicated, as explained below.

The immunity variables are updated via the operation of function multiplication. Recall that the immunity functions store probabilities of failure, so the product of such functions give the probabilities of failure for overlapping immune responses, assuming independence.

For example, suppose in the above example that the individual has no immunity against either strain prior to the infection by strain `0`, which we assume occurred at time `t`. Then after updating their immunity functions, the component of their sigma immunity function corresponding to strain `0` will be given by the step function

```
[-1, t + 5, t + 30], [1.0, 0.25, 1.0],
```

while the component corresponding to strain `1` will be given by

```
[-1, t + 5, t + 30], [1.0, 0.5, 1.0].
```

In particular, for 25 days following the end of their infection, the probability that their immune system fails to protect against another infection by strain `0` is improved from `1.0` to `0.25`, after which it returns to `1.0`, representing a loss of immunity. The probability that their immune system fails to protect against an infection by strain `0` is improved from `1.0` to `0.5`, after which it returns to `1.0`, representing a loss of partial cross immunity.

Suppose now that at time `t + 10` individual `n` is again infected by strain `0`. Then between times `t + 10` and `t + 15`, the three variables describing their current strain, disease and infectiousness will be updated as before.

But the component of their sigma immunity function corresponding to strain 0 will now be subject to appropriate multiplication, after which it will be given by the step function

```
[-1, t + 5, t + 15, t + 30, t + 40],
```

```
[1.0, 0.25, 0.0625, 0.25, 1.0].
```

In particular, between times `t + 15` and `t + 30` there are overlapping immunity responses, so that in order for a third infection to occur by strain 0, the pathogen must defeat the immune response generated by the first infection *and* the immune response generated by the second infection.

Updates to rho immunity are similar, except these functions are now array-valued, these arrays corresponding to the failure probabilities for each internal layer. The functions are simply multiplied component-wise.

While the components of a preset outcome corresponding to strain, disease and infectiousness are vectors, each containing precisely $\mathtt{S}$ functions, where $\mathtt{S}$ is the number of strains, the components corresponding to rho and sigma immunity are matrices, containing precisely $\mathtt{S} \times \mathtt{S}$ functions, the additional dimension accounting for possibility of cross-immunity, as in the above example.