Compartmental models as stochastic processes

Stochastic processes

A Stochastic Process is a collection of random variables, $X(t)$. In the case of a compartmental SIR model **?@sec-ref**, the state of the model at time t can be represented by the number of individuals in the Susceptible and Infected compartments, i.e.

$$X(t) = S(t) + I(t)$$

Where $S(t), I(t) \in \{0, 1, 2, \ldots, N\}$, $S(t) + I(t) \leq N$, and $t \in [0, \infty)$

We can represent the probability of a transitioning one particular state $x = (s, i)$[1] to another $x = (s + k, i + j)$ (i.e. due to infection or recovery) after a time $\Delta t > 0$, as

$$P(X(t + \Delta t) = (s + k, i + j) \mid (S(t), I(t)) = (s, i))$$

We will make two assumptions about the probability of states, $P(X(t))$:

1. Time Homogeneity

   $$P\left(X(t + \Delta t) = j \mid X(t) = i\right) = P\left(X(t) = j \mid X(0) = i\right)$$

   Which states that the transition probabilities depend on the time between events, Δt, but not on the specific time t.

2. Memorylessness (Markov Property)

   $$P(X(t + \Delta t) = j \mid X(t) = i, X(u) = k, 0 \leq u < t) = P(X(t + \Delta t) = j \mid X(t) = i)$$

   Which states that the probability of transitioning from a state i at time t to a state j at time $(t + \Delta t)$ depends on the state at t and not on any previous state.

[1] lower case letters (e.g. s, i) represent specific values of the random variables ($S(t)$ and $I(t)$)

Representing our compartmental model as such a ‘Time Homogeneous Markov Chain’ means that we can succinctly describe the probabilities of possible state transitions (‘events’) because they only depend on the intervening time (Δt) and the current state $x_t = (s_t, i_t)$.

Table 1: Transition events in a SIR time homogeneous markov chain

Event	Change (ΔS , ΔI)	Probability
Infection	$(-1, +1)$	$\beta i \frac{s}{N} \Delta t + o(\Delta t)$
Recovery	$(0, -1)$	$\gamma i \Delta t + o(\Delta t)$

Time homogeneity and the markov property allow us to define the ‘transition function’ $p_{ij}(t)$, which gives the probabilty of transitioning from state i to state j after time t

$$p_{ij} = P(X(t) = P(X(t) = j \mid X(0) = i)$$

and represent the possible state transition probabilities for our SIR model as

$$p_{(s,i),(s+k,i+j)}(\Delta t) = \begin{cases} \beta i \frac{s}{N} \Delta t + o(\Delta t), & (k, j) = (-1, +1) \\ \gamma i \Delta t + o(\Delta t), & (k, j) = (0, -1) \\ 1 - \left(\beta i \frac{s}{N} + \gamma i\right) \Delta t & \\ +o(\Delta t), & (k, j) = (0, 0) \\ 0(\Delta t), & \text{otherwise.} \end{cases} \tag{1}$$

Moreover, if we consider the state space of $X(t)$, (e.g. containing all possible comninations of numbers of susceptible and infected individuals (s, i),

$$\mathbb{S} = \{s, i : 0 \leq s, i; s + i \leq N\}.$$

For a particular ordering of this space[2] (e.g. $(s, i) \in \{(N, 0), (N - 1, 1), \dots, (0, 0)\}$), $P(t)$ can be represented as a matrix specifying the probability of every state transition at time t. Our bivariate SIR state space contains $(N + 1)(N + 2)/2$ elements, and therefore P(t) corresponds to a $(N + 1)(N + 2)/2 \times (N + 1)(N + 2)/2$ matrix.

We can now consider the derivative of the transition function, $p'_{i,j}(t)$, which has two forms known as the forward (Equation 2) and backward (Equation 3) Kolmogorov equations

$$\frac{dp_{i,j}(t)}{dt} = \sum_{k \neq i} p_{i,k}(t) q_{k,j} - q_{i,i} p_{i,j}(t) \tag{2}$$

[2] Note that while the ordering of state space is arbitrary, it is not without consequence @black2015

$$\frac{dp_{i,j}(t)}{dt} = \sum_{k\neq i} q_{i,k}p_{k,j}(t) - q_{i,i}p_{i,j}(t) \tag{3}$$

where $q_{k,j}$,$q_{i,i}$, and $q_{i,k}$ are the transition *rates* defined in Equation 1, i.e. for our SIR model, the forward Kolmogorov equations are

$$\frac{dp_{a,(s,i)}(t)}{dt} = p_{a,(s+1,i-1)}(t)\frac{\beta}{N}(s+1)(i-1) + p_{a,(s,i+1)}(t)\gamma(i+1) - p_{a,(s,i)}(t)\left[\frac{\beta}{N}si + \gamma i\right].$$

and the backward Kolmogorov equations are

$$\frac{dp_{(s,i),b}(t)}{dt} = \frac{\beta si}{N}p_{(s-1,i+1),b}(t) + \gamma i p_{(s,i-1),b}(t) - \left[\frac{\beta}{N}si + \gamma i\right]p_{(s,i),b}(t).$$

We can simplify (Equation 2) and (Equation 3), by considering the matrix Q specifying the rates of all transitions q_{ij} for $i, j \in \mathbb{S}$.Similarly to the transition probability matrix $P(t)$, Q has dimension $(N+1)(N+2)/2 \times (N+1)(N+2)/2$.

The system of equations Equation 2 and Equation 3, can now be written as $dP(t)/dt = P(t)Q$ and $dP(t)/dt = QP(t)$, respectively. Considering$p'(t) = qp(t)$, where p is a differentiable function and q is a constant, when $p(0) = 1$, there is a unique solution $p(t) = e^{tq}$, we can similarly solve the Kolmogorov Equation 2 and Equation 3 as as $P(t) = e^{Qt}$ and $P(t) = e^{tQ}$ (when $P(0) = I$), respectively.

Thus, the theoretically and empirically derived event rates presented in Equation 1, along with the assumptions of time homogeneity and memorylessness, allow us to compute the probability of any state transition at time t. This technique can be used, for example, to derive important statistics about an epidemic, such as its mean final size and the effects of vaccination programs @teo2017, @teo2021. However, “computing the matrix exponential is often numerically challenging. Finding accurate and eficient algorithms is still a topic of current research” @dobrow

Stochastic simulation

Instead of analytically deriving the marginal probability of a particular series of events by the method outlined above, we can simulate instantiations of the continuous time markov chain (“sample paths”) numerically. There are several stochastic simulation algorithms @simoni2019, but in this thesis, we will use only the original ‘Direct-method’ algorithm put forward by @gillespie1976.

Originally a method for simulating chemical reaction networks, the Gillespie direct algorithm is applicable to many domains of stochastic simulation with discrete state spaces, continuous time, and known rate equations.

In general, the core of the Gillespie algorithm takes in the possible state changes in a system, and a 'propensity function' for each state change. Random number generation is used to select a particular state change, u and a time interval, τ, which can be used to update the state of the system $X(t) = i \rightarrow X(t + \tau) = i + u$ (Algorithm 1).

Algorithm 1 Gillespie Direct Method

```
Input: $v$, $a$
Output: $\mu$, $\tau$
set $a_{net} = 0$
for all j do
    compute $a_j$
    update $a_{net} = a_{net} + a_j$
end for
generate two random numbers $r_1$, $r_2$ in $\mathcal{U}(0, 1)$
select $\mu$ such that $\sum_{j=1}^{\mu} a_j \leq r_1 a_0$
compute $\tau \leftarrow \frac{1}{a_0} \ln(1/r_2)$
update $X \leftarrow X + v_\mu$
set $t \leftarrow t + \tau$
```

In particular, two random numbers r_1 and r_2 are used to compute mu and τ, respectively

- r_1 is used to calculate mu by cumulatively adding the values of the propensity function to each other. mu gets the largest value which is smaller than r_1
- r_2 is used to calculate τ by taking advantage of the fact that in a Continuous time markov proces $\Delta t \sim \lambda e^{-\lambda t}$, where lambda is the sum of the rates for all possible events.

In the case of our SIR model, the state change vector is given by Table 1 column two, and the propensity function vector is given in column 3.We can include the core stochastic event step in a loop which updates the state and time of our SIR model simulate the progression of an epidemic @alg-SIR_CTMC.

As an example, **?@fig-GMelb_SIR_Sim** 100 sample trajectories of an SIR CTMC comprising N = 4976157, the population of the Greater Melbourne region. In this example, $I_0 = 10$, $\beta = 1.4$, and $\gamma = 1.0$, though clearly variation in these parameter values can be explored.

The Greater Melbourne Region presents large geographical region which highlights the important assumption of homogeneous mixing, which we will explore relaxing in the next chapter.

Algorithm 2 Stochastic simulation of SIR Compartmental Model CTMC

```
Input:$N$, $I_0$, $\beta$, $\gamma$ Output:
Initialise time $t \leftarrow 0.0$ and $S \leftarrow N - I_0$, $I \leftarrow I_0$
state change vector $v = [(-1, 1), (0, -1)]$
propensity vector $a = [(\beta I \frac{S}{N}), (\gamma I)]$
while $I > 0$ do
    for all j do
        compute $a_j$
        update $a_{net} = a_{net} + a_j$
    end for
    generate two random numbers $r_1, r_2 \sim \mathcal{U}(0, 1)$
    select $\mu$ such that $\sum_{j=1}^{\mu} a_j \leq r_1 a_0$
    compute $\tau \leftarrow \frac{1}{a_0} \ln(1/r_2)$
    update $X \leftarrow X + v_\mu$
    set $t \leftarrow t + \tau$
end while
```