Block 2.3
Stochastic Simulation Algorithms

NDMC Measles and Rubella Transmission Modelling Worshop

5-8 February 2024

The SIR Model

$$\frac{dS}{dt} = -\beta S \frac{I}{N}$$

$$\frac{dI}{dt} = \beta S \frac{I}{N} - \gamma I$$

$$\frac{dR}{dt} = \gamma I$$

Recall that:

Rate of event is proportional to time until event

Here, rates depend on current states (number of S/I/R)

So, rates are constant ***until*** the states change

When the states change, the rates change

The SIR Model

$$\frac{dS}{dt} = -\beta S \frac{I}{N}$$

$$\frac{dI}{dt} = \beta S \frac{I}{N} - \gamma I$$

$$\frac{dR}{dt} = \gamma I$$

Recall that:

For constant rate processes, with rate θ, the time until the event occurs can be reasonably modeled as an exponentially distributed random variable with mean 1/ θ

The SIR Model

$$\frac{dS}{dt} = -\beta S \frac{I}{N}$$

$$\frac{dI}{dt} = \beta S \frac{I}{N} - \gamma I$$

$$\frac{dR}{dt} = \gamma I$$

For this simple SIR model:

Two kinds of events can happen

1. An infection can occur, which leads to
   S -> S – 1 and
   I -> I + 1
   This occurs at a rate $\beta S \frac{I}{N}$

2. An infected individual recovers, which leads to
   I -> I – 1 and
   R -> R + 1
   This occurs at a rate γI

The Gillespie Algorithm

Conditional on current numbers of S, I, R and current time T

1. Take a random draws from Exponential distribution for all possible state transitions

   X = random draw from exponential distribution with mean $1/(\beta S \frac{I}{N})$

   Y = random draw from exponential distribution with mean γI

2. Update States: IF

   X <= Y, then transmission occurs first: S -> S - 1, I -> I + 1, increment time T + X

   ELSE, recovery occurs first: I -> I - 1, R -> R + 1, increment time T + Y

3. Return to step 1

Interactive session 1: Run code

Questions:

What was R_0?

What did you notice about trajectories?

Gillespie Algorithm

Benefits

- Exact translation of stochastic ODE

Costs

- Computation scales with the number of state transitions (arrows in the SIR model) AND with the population size
- For this example, must draw ~2 random variables for every individual in the simulated population

Tau-leaping Algorithm

$$\frac{dS}{dt} = -\beta S \frac{I}{N}$$

$$\frac{dI}{dt} = \beta S \frac{I}{N} - \gamma I$$

$$\frac{dR}{dt} = \gamma I$$

Recall that:

IF an event occurs at a constant rate θ, THEN the number of events that will occur in a time interval ΔT will be distributed as:

Poisson(θ* ΔT)

i.e. number of events will increase with the rate AND the duration of the interval

Tau-leaping Algorithm

$$\frac{dS}{dt} = -\beta S \frac{I}{N}$$

$$\frac{dI}{dt} = \beta S \frac{I}{N} - \gamma I$$

$$\frac{dR}{dt} = \gamma I$$

So, for relatively small time steps, ΔT:

We can assume that the number of new infections that will occur will be distributed as:

$$\text{Poisson}(\beta S \frac{I}{N} * \Delta T)$$

And the number of recoveries that will occur will be distributed as:

$$\text{Poisson}(\gamma I * \Delta T)$$

Tau-leaping Algorithm

Conditional on current numbers of S, I, R and time step ΔT

1. Make random draws:

   New infections, dSI, are random draw from Poisson($\beta S \frac{I}{N}$* ΔT)

   New recoveries, dIR, are random draw from Poisson(γI* ΔT)

2. Update states

   S_{new} = S - dSI

   I_{new} = I + dSI – dIR

   R_{new} = R + dIR

   Update time by ΔT

3. Return to 1

Tau-leaping Algorithm

Conditional on current numbers of S, I, R and time step ΔT

1. Make random draws:

   New infections, dSI, are random draw from Poisson($\beta S \frac{I}{N}$* ΔT)

   New recoveries, dIR, are random draw from Poisson(γI* ΔT)

2. Update states

   S_{new} = S - dSI

   I_{new} = I + dSI – dIR

   R_{new} = R + dIR

   Update time by ΔT

3. Return to 1

Recall the definitions of **Incidence** and **Prevalence**,

Which of the terms at the left is the **Incidence**?

Which of the terms at the left is the **Prevalence**?

Is the number of children newly diagnosed for measles at clinic **Incidence** or **Prevalence**?

Tau-leaping Algorithm

Benefits

- Computation (number of random draws) is independent of population size

  (Note: while this makes a small difference in this code, it makes a ***large*** difference when simulating with births and deaths)

Costs

- Inexact. Requires assumption that rate is constant over ΔT, even though we know states are changing.

Interactive session 2: Run code

Questions:

How well do projections of Gillespie and Tau-leaping match?

Tau-leaping Algorithm

Benefits

- Computation (number of random draws) is independent of population size

  (Note: while this makes a small difference in this code, it makes a ***large*** difference when simulating with births and deaths)

Costs

- Did anyone get “warning” errors?

There were 50 or more warnings (use warnings() to see the first 50)

Tau-leaping Algorithm

Conditional on current numbers of S, I, R and time step ΔT

1. Make random draws:

   New infections, dSI, are random draw from Poisson($\beta S \frac{I}{N}$* ΔT)

   New recoveries, dIR, are random draw from Poisson(γI* ΔT)

2. Update states

   S_{new} = S - dSI

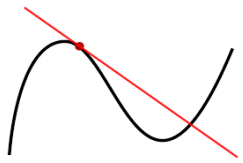
   I_{new} = I + dSI – dIR

   R_{new} = R + dIR

   Update time by ΔT

3. Return to 1

Tangent



It is possible to make a random draw of dIR that is greater than I, which makes I_{new}<0

This can be addressed by disallowing negative outcomes in code, or using binomial draws

Tau-leaping Algorithm

$$\frac{dS}{dt} = -\beta S \frac{I}{N}$$

$$\frac{dI}{dt} = \beta S \frac{I}{N} - \gamma I$$

$$\frac{dR}{dt} = \gamma I$$

ALSO, for relatively small time steps, ΔT:

Rate in Poisson process can be translated to probability of event and modeled as a Binomial random variable.

The number of new infections that will occur will be:

Binomial(S, 1-exp($-\beta \frac{I}{N}$* ΔT))

And the number of recoveries that will occur will be:

Binomial(I,1-exp(-γ * ΔT))

Tau-leaping Algorithm

$$\frac{dS}{dt} = -\beta S \frac{I}{N}$$

$$\frac{dI}{dt} = \beta S \frac{I}{N} - \gamma I$$

$$\frac{dR}{dt} = \gamma I$$

ALSO, for relatively small time steps, ΔT:

Rate in Poisson process can be translated to probability of event and modeled as a Binomial random variable.

The number of new infections that will occur will be:

Binomial(S, 1-exp($-\beta \frac{I}{N}$* ΔT))

And the number of recoveries that will occur will be:

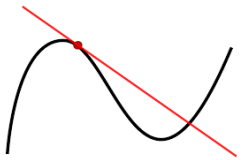
Binomial(I, 1-exp(-γ * ΔT))

Recall that this expression turns **rate** ($\beta S \frac{I}{N}$ or γ) measured over time into **risk,** which is measured per population.

ODEs, Gillespie, Tau-leaping

- An implicit assumption of the standard ODE representation the SIR model, the difference equation representation, AND both the Gillespie and Tau-leaping algorithms is that the rate of recovery (γ) is constant across time for each infected individual.
- This means that an individual is just as likely to recover after 1 day as after 10 days or 20 day (if they haven't already recovered)

Tangent



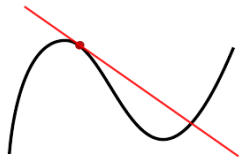
Some elegant approaches to overcome this:
http://alun.math.ncsu.edu/static/alun/tpb_01.pdf

ODEs, Gillespie, Tau-leaping

- An implicit assumption of the standard ODE representation the SIR model, the difference equation representation, AND both the Gillespie and Tau-leaping algorithms is that the rate of recovery (γ) is constant across time for each infected individual.
- This means that an individual is just as likely to recover after 1 day as after 10 days or 20 day (if they haven't already recovered)

One simple approach to overcome this is to use a time step, ΔT, that is equal to the average infectious period, $1/\gamma$

Tangent



Some elegant approaches to overcome this:
http://alun.math.ncsu.edu/static/alun/tpb_01.pdf

Chain Binomial Algorithm

Conditional on current numbers of S, I, R and time step equal to $1/\gamma$

1. I_{new} is a binomial draw from all current susceptibles

   I_{new} ~binomial(S, 1 – exp(- $\beta \frac{I}{N}$))

2. $S_{new} = S - I_{new}$
3. Return to 1

Note that magnitude of β will be different than for Gillespie and Tau-leaping algorithms.

Expected value of binomial(N,p) is N*p.
Here, the expected value of I_{new} is
$E[I_{new}] = S * 1\text{-}\exp(-\beta \frac{I}{N}) \cong \beta * S * \frac{I}{N}$

Chain Binomial Algorithm

Conditional on current numbers of S, I, R and time step equal to $1/\gamma$

1. I_{new} is a binomial draw from all current susceptibles

   I_{new} ~binomial(S, $1 - \exp(-\beta \frac{I}{N})$)

2. $S_{new} = S - I_{new}$
3. Return to 1

Note that magnitude of β will be different than for Gillespie and Tau-leaping algorithms.

Expected value of binomial(N,p) is N*p.
Here, the expected value of I_{new} when I=1 and S=N is

$E[I_{new}] = S * 1\text{-}\exp(-\beta \frac{I}{N}) \cong \beta * S * \frac{I}{N}$

$= N * 1\text{-}\exp(-\beta \frac{1}{N}) = N * 1\text{-}\exp(-\frac{\beta}{N})$

$\cong N * \frac{\beta}{N}$

$\cong \beta$ -> $\beta = R_0$

Chain Binomial Algorithm

Benefits

- Computation (number of random draws) is independent of population size
- Only need 1 random draw per time step
- Natural translation to methods for inference …

Costs

- Inexact. Requires assumption that rate is constant over ΔT, even though we know states are changing.

Interactive session 3: Run code

Questions:

How well do projections of Chain Binomial match Gillespie and Tau-leaping?

Chain Binomial Algorithm

Conditional on current numbers of S, I, R and time step equal to $1/\gamma$

1. I_{new} is a binomial draw from all current susceptibles

   I_{new} ~binomial(S, 1 – exp(- $\beta \frac{I}{N}$))

2. $S_{new} = S - I_{new}$
3. Return to 1

Note that magnitude of β will be different than for Gillespie and Tau-leaping algorithms.

Expected value of binomial(N,p) is N*p.
Here, the expected value of I_{new} is

$E[I_{new}] = S * 1\text{-}\exp(-\beta \frac{I}{N}) \cong \frac{\beta}{N} * S * I = \hat{\beta} * S * I$

Chain Binomial Algorithm – with births

Conditional on current numbers of S, I, R and time step equal to $1/\gamma$

1. I_{new} is a binomial draw from all current susceptibles

   I_{new} ~binomial(S, 1 – exp(- $\beta \frac{I}{N}$))

2. $S_{new} = B + S – I_{new}$
3. Return to 1

Note that magnitude of β will be different than for Gillespie and Tau-leaping algorithms.

Expected value of binomial(N,p) is N*p.
Here, the expected value of I_{new} is
$E[I_{new}] = S * 1\text{-}\exp(-\beta \frac{I}{N}) \cong \frac{\beta}{N} * S * I$

B = birth in each time step