Workshop Day 2 - Introduction and motivations

Why do we need other methods?

Gillespie ♡

Why do we need other methods?

Gillespie ♡

1. Time-dependent rates (Lewis thinning)
2. Large population sizes (τ-leaping)
3. Applications
4. Exercises

Time-dependent rates

- If rates are time-dependent (e.g. environmental variation), propensities are $a_i(\boldsymbol{x}, t)$

Time-dependent rates

- If rates are time-dependent (e.g. environmental variation), propensities are $a_i(\boldsymbol{x}, t)$
  - For example, birth rate vary through time: $a_1(x, t) = b(t)x(t)$

Time-dependent rates

- If rates are time-dependent (e.g. environmental variation), propensities are $a_i(\boldsymbol{x}, t)$
  - For example, birth rate vary through time: $a_1(x, t) = b(t)x(t)$
- The Gillespie algorithm is not suitable because the rates can change during the waiting time.

Time-dependent rates

- If rates are time-dependent (e.g. environmental variation), propensities are $a_i(\boldsymbol{x}, t)$
  - For example, birth rate vary through time: $a_1(x, t) = b(t)x(t)$
- The Gillespie algorithm is not suitable because the rates can change during the waiting time.
- But is it still possible to derive an exact method?

Temporal Gillespie algorithm

$$p(\tau, i)dt = \text{nothing happens in } (t, t+\tau)$$
$$\textbf{and} \text{ i occurs in } (t+\tau, t+\tau+dt)$$

$$p(\tau, i)dt = \text{nothing happens in } (t, t+\tau)$$

$$\textbf{and} \text{ i occurs in } (t+\tau, t+\tau+dt)$$

$$p(\tau, i) = e^{-a_0} \times a_i$$

Temporal Gillespie algorithm

$$p(\tau, i)\,dt = \text{nothing happens in } (t, t+\tau)$$
$$\textbf{and} \text{ i occurs in } (t+\tau, t+\tau+dt)$$

$$p(\tau, i) = \prod_i e^{-\int_t^{t+\tau} a_i(s)ds} \times a_i(t+\tau)$$

Temporal Gillespie algorithm

$$p(\tau, i)\,dt = \text{nothing happens in } (t, t+\tau)$$
$$\textbf{and} \text{ i occurs in } (t+\tau, t+\tau+dt)$$
$$p(\tau, i) = e^{-\sum_i \int_t^{t+\tau} a_i(s)ds} \times a_i(t+\tau)$$

Temporal Gillespie algorithm

$$p(\tau, i)\,dt = \text{nothing happens in } (t, t+\tau)$$
$$\quad\textbf{and}\text{ i occurs in } (t+\tau, t+\tau+dt)$$
$$p(\tau, i) = e^{-\int_t^{t+\tau} \sum_i a_i(s)\,ds} \times a_i(t+\tau)$$

Temporal Gillespie algorithm

$$
\begin{aligned}
p(\tau, i)\,dt &= \text{nothing happens in } (t, t+\tau) \\
&\quad \textbf{and } \text{i occurs in } (t+\tau, t+\tau+dt)
\end{aligned}
$$

$$p(\tau, i) = e^{-\int_t^{t+\tau} a_0(s)ds} \times a_i(t+\tau)$$

Temporal Gillespie algorithm

$$p(\tau,i)dt = \text{nothing happens in } (t, t+\tau)$$
$$\qquad \textbf{and} \text{ i occurs in } (t+\tau, t+\tau+dt)$$

$$p(\tau,i) = e^{-\int_t^{t+\tau} a_0(s)ds} \times a_i(t+\tau)$$

$$e^{-\int_t^{t+\tau} a_0(s)ds} = P(\tau > u)$$

Temporal Gillespie algorithm

$$p(\tau, i)dt = \text{nothing happens in } (t, t+\tau)$$
$$\qquad \textbf{and} \text{ i occurs in } (t+\tau, t+\tau+dt)$$

$$p(\tau, i) = e^{-\int_t^{t+\tau} a_0(s)ds} \times a_i(t+\tau)$$

$$e^{-\int_t^{t+\tau} a_0(s)ds} = 1 - P(\tau < u)$$

Temporal Gillespie algorithm

$$p(\tau, i)\,dt = \text{nothing happens in } (t, t+\tau)$$

$$\textbf{and} \text{ i occurs in } (t+\tau, t+\tau+dt)$$

$$p(\tau, i) = e^{-\int_t^{t+\tau} a_0(s)ds} \times a_i(t+\tau)$$

$$e^{-\int_t^{t+\tau} a_0(s)ds} = 1 - F(u)$$

Temporal Gillespie algorithm

$$p(\tau, i)\,dt = \text{nothing happens in } (t, t+\tau)$$
$$\textbf{and} \text{ i occurs in } (t+\tau, t+\tau+dt)$$

$$p(\tau, i) = e^{-\int_t^{t+\tau} a_0(s)ds} \times a_i(t+\tau)$$

$$1 - e^{-\int_t^{t+\tau} a_0(s)ds} = F(u)$$

Temporal Gillespie algorithm

$$p(\tau, i)dt = \text{nothing happens in } (t, t+\tau)$$
$$\textbf{and} \text{ i occurs in } (t+\tau, t+\tau+dt)$$
$$p(\tau, i) = e^{-\int_t^{t+\tau} a_0(s)ds} \times a_i(t+\tau)$$

$$r \sim \mathcal{U}(0,1) \quad ; \quad F(\tau) = r \iff 1 - e^{-\int_t^{t+\tau} a_0(s)ds} = r$$

Temporal Gillespie algorithm

$$p(\tau, i)\,dt = \text{nothing happens in } (t, t+\tau)$$
$$\textbf{and} \text{ i occurs in } (t+\tau, t+\tau+dt)$$
$$p(\tau, i) = e^{-\int_t^{t+\tau} a_0(s)ds} \times a_i(t+\tau)$$

$$r \sim \mathcal{U}(0,1) \quad ; \quad F(\tau) = r \iff \int_t^{t+\tau} a_0(s)ds = \ln\left(\frac{1}{r}\right)$$

Temporal Gillespie algorithm

$$p(\tau, i)\,dt = \text{nothing happens in } (t, t+\tau)$$
$$\textbf{and} \text{ i occurs in } (t+\tau, t+\tau+dt)$$
$$p(\tau, i) = e^{-\int_t^{t+\tau} a_0(s)ds} \times a_i(t+\tau)$$

$$r \sim \mathcal{U}(0,1) \quad ; \quad F(\tau) = r \iff \int_t^{t+\tau} a_0(s)ds = \ln\left(\frac{1}{r}\right)$$

- If $a_0(t)$ is integrable, or if one has a numerical expression, then it is possible to generalize the Gillespie algorithm. But it can be extremely time consuming. And the time increases with the number of reactions.

Temporal Gillespie algorithm

$$p(\tau, i)dt = \text{nothing happens in } (t, t+\tau)$$
$$\textbf{and} \text{ i occurs in } (t+\tau, t+\tau+dt)$$
$$p(\tau, i) = e^{-\int_t^{t+\tau} a_0(s)ds} \times a_i(t+\tau)$$

$$r \sim \mathcal{U}(0,1) \quad ; \quad F(\tau) = r \iff \int_t^{t+\tau} a_0(s)ds = \ln\left(\frac{1}{r}\right)$$

- If $a_0(t)$ is integrable, or if one has a numerical expression, then it is possible to generalize the Gillespie algorithm. But it can be extremely time consuming. And the time increases with the number of reactions.
- It is also possible to construct exact time-dependent methods by modifying the First-Reaction Method.

Summary: The Temporal Gillespie algorithm

1. Initialize the time $t = t_0$ and the system's state $\boldsymbol{x} = \boldsymbol{x_0}$

Summary: The Temporal Gillespie algorithm

1. Initialize the time $t = t_0$ and the system's state $\boldsymbol{x} = \boldsymbol{x_0}$
2. With the system in state $\boldsymbol{x}$ at time t, generate a random value $r \sim \mathcal{U}(0,1)$ and compute τ using $\int_t^{t+\tau} a_0(s)\,ds = \ln\left(\frac{1}{r}\right)$

Summary: The Temporal Gillespie algorithm

1. Initialize the time $t = t_0$ and the system's state $\boldsymbol{x} = \boldsymbol{x_0}$
2. With the system in state $\boldsymbol{x}$ at time t, generate a random value $r \sim \mathcal{U}(0,1)$ and compute τ using $\int_t^{t+\tau} a_0(s)\,ds = \ln\left(\frac{1}{r}\right)$
3. Generate a random value for i using the distribution $\frac{a_i(\boldsymbol{x},t+\tau)}{a_0(\boldsymbol{x},t+\tau)}$

Summary: The Temporal Gillespie algorithm

1. Initialize the time $t = t_0$ and the system's state $\boldsymbol{x} = \boldsymbol{x_0}$
2. With the system in state $\boldsymbol{x}$ at time t, generate a random value $r \sim \mathscr{U}(0,1)$ and compute τ using $\int_t^{t+\tau} a_0(s)\,ds = \ln\left(\frac{1}{r}\right)$
3. Generate a random value for i using the distribution $\frac{a_i(\boldsymbol{x},t+\tau)}{a_0(\boldsymbol{x},t+\tau)}$
4. Update $t \leftarrow t + \tau$ and $\boldsymbol{x}$ according to i

Summary: The Temporal Gillespie algorithm

1. Initialize the time $t = t_0$ and the system's state $\boldsymbol{x} = \boldsymbol{x_0}$
2. With the system in state $\boldsymbol{x}$ at time t, generate a random value $r \sim \mathcal{U}(0,1)$ and compute τ using $\int_t^{t+\tau} a_0(s)\,ds = \ln\left(\frac{1}{r}\right)$
3. Generate a random value for i using the distribution $\frac{a_i(\boldsymbol{x},t+\tau)}{a_0(\boldsymbol{x},t+\tau)}$
4. Update $t \leftarrow t+\tau$ and $\boldsymbol{x}$ according to i
5. Save $(\boldsymbol{x}, t)$ as desired and return to Step 2, or else end the simulation

Lewis thinning algorithm (1978)

SIMULATION OF NONHOMOGENEOUS POISSON PROCESSES BY THINNING

P. A. W. Lewis*

Naval Postgraduate School
Monterey, California

G. S. Shedler

IBM Research Laboratory
San Jose, California

Lewis thinning algorithm

- The main problem of the temporal Gillespie algorithm is the difficulty to integrate $\int_t^{t+\tau} a_0(s)ds$

Lewis thinning algorithm

- The main problem of the temporal Gillespie algorithm is the difficulty to integrate $\int_t^{t+\tau} a_0(s)ds$
- The trick here is to find an upper bound of $a_0(s)$:
  $\bar{a}_0 \geq a_0(s) \quad \forall s \geq t$
  and sample the waiting time via $\tau \sim \text{Exp}(\bar{a}_0)$

Lewis thinning algorithm

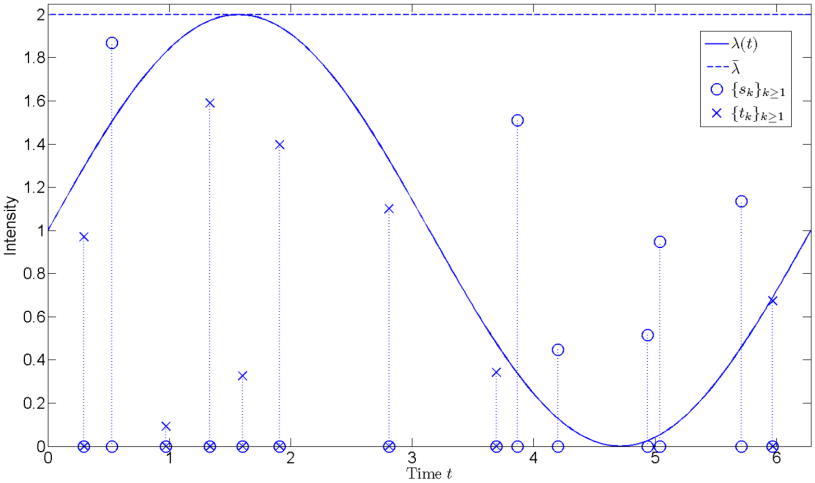
- The main problem of the temporal Gillespie algorithm is the difficulty to integrate $\int_t^{t+\tau} a_0(s)ds$
- The trick here is to find an upper bound of $a_0(s)$:
  $\bar{a}_0 \geq a_0(s) \quad \forall s \geq t$
  and sample the waiting time via $\tau \sim \text{Exp}(\bar{a}_0)$
- It means that we are updating time more frequently than we should: **sometimes, no reaction should occur**

Lewis thinning algorithm

- The main problem of the temporal Gillespie algorithm is the difficulty to integrate $\int_t^{t+\tau} a_0(s)ds$
- The trick here is to find an upper bound of $a_0(s)$:
  $\bar{a}_0 \geq a_0(s) \quad \forall s \geq t$
  and sample the waiting time via $\tau \sim \text{Exp}(\bar{a}_0)$
- It means that we are updating time more frequently than we should: **sometimes, no reaction should occur**
- One can prove that if we discard every update with probability $1 - \frac{a_0(t+\tau)}{\bar{a}_0}$, then this method actually sample from the joint distribution $p(\tau, i)$

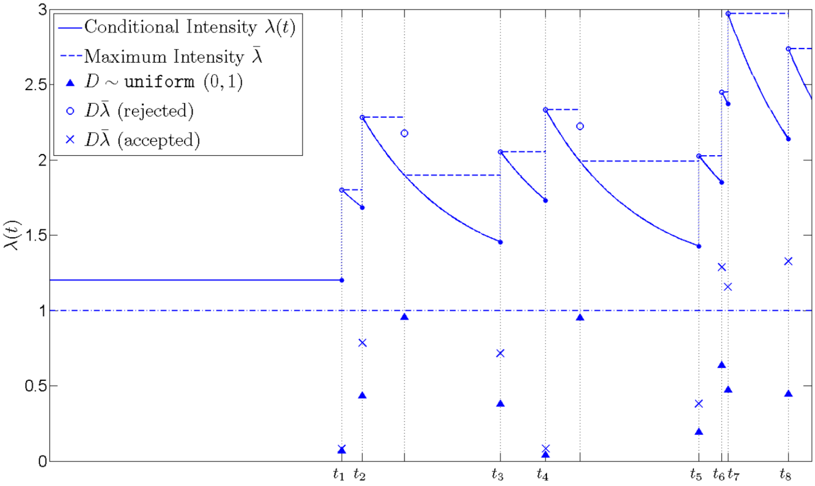
An example with a single reaction

Here we assume that the propensity $\lambda(t) = 1 + \sin(t)$ does not depend on the system state $\boldsymbol{x}$



An example with a single reaction

Now, the propensity $\lambda(t)$ depends on the system state $\boldsymbol{x}$



Summary: The Lewis thinning algorithm

1. Initialize the time $t = t_0$ and the system's state $\boldsymbol{x} = \boldsymbol{x_0}$.

Summary: The Lewis thinning algorithm

1. Initialize the time $t = t_0$ and the system's state $\boldsymbol{x} = \boldsymbol{x_0}$.
2. With the system in state $\boldsymbol{x}$ at time t, for each reaction, find a constant which is always larger than its propensity
   $\bar{a}_i \geq a_i(\boldsymbol{x}, s) \quad \forall s \geq t$

Summary: The Lewis thinning algorithm

1. Initialize the time $t = t_0$ and the system's state $\boldsymbol{x} = \boldsymbol{x_0}$.
2. With the system in state $\boldsymbol{x}$ at time t, for each reaction, find a constant which is always larger than its propensity
   $\bar{a}_i \geq a_i(\boldsymbol{x}, s) \quad \forall s \geq t$
3. Compute the sum $\bar{a}_0 = \sum_i \bar{a}_i$

Summary: The Lewis thinning algorithm

1. Initialize the time $t = t_0$ and the system's state $\boldsymbol{x} = \boldsymbol{x_0}$.
2. With the system in state $\boldsymbol{x}$ at time t, for each reaction, find a constant which is always larger than its propensity $\bar{a}_i \geq a_i(\boldsymbol{x}, s) \quad \forall s \geq t$
3. Compute the sum $\bar{a}_0 = \sum_i \bar{a}_i$
4. Generate a random value for τ using an exponential distribution with parameter $\bar{a}_0$

Summary: The Lewis thinning algorithm

1. Initialize the time $t = t_0$ and the system's state $\boldsymbol{x} = \boldsymbol{x_0}$.
2. With the system in state $\boldsymbol{x}$ at time t, for each reaction, find a constant which is always larger than its propensity $\bar{a}_i \geq a_i(\boldsymbol{x}, s) \quad \forall s \geq t$
3. Compute the sum $\bar{a}_0 = \sum_i \bar{a}_i$
4. Generate a random value for τ using an exponential distribution with parameter $\bar{a}_0$
5. Evaluate all the propensities at time $(t+\tau)$: $a_i(\boldsymbol{x}, t+\tau)$

Summary: The Lewis thinning algorithm

1. Initialize the time $t = t_0$ and the system's state $\boldsymbol{x} = \boldsymbol{x_0}$.
2. With the system in state $\boldsymbol{x}$ at time t, for each reaction, find a constant which is always larger than its propensity
   $\bar{a}_i \geq a_i(\boldsymbol{x}, s) \quad \forall s \geq t$
3. Compute the sum $\bar{a}_0 = \sum_i \bar{a}_i$
4. Generate a random value for τ using an exponential distribution with parameter $\bar{a}_0$
5. Evaluate all the propensities at time $(t+\tau)$: $a_i(\boldsymbol{x}, t+\tau)$
6. Generate a random value for i using the distribution $\frac{a_i(\boldsymbol{x}, t+\tau)}{\bar{a}_0}$

Summary: The Lewis thinning algorithm

1. Initialize the time $t = t_0$ and the system's state $\boldsymbol{x} = \boldsymbol{x_0}$.
2. With the system in state $\boldsymbol{x}$ at time t, for each reaction, find a constant which is always larger than its propensity $\bar{a}_i \geq a_i(\boldsymbol{x}, s) \quad \forall s \geq t$
3. Compute the sum $\bar{a}_0 = \sum_i \bar{a}_i$
4. Generate a random value for τ using an exponential distribution with parameter $\bar{a}_0$
5. Evaluate all the propensities at time $(t+\tau)$: $a_i(\boldsymbol{x}, t+\tau)$
6. Generate a random value for i using the distribution $\frac{a_i(\boldsymbol{x}, t+\tau)}{\bar{a}_0}$
7. Update $t \leftarrow t+\tau$ and $\boldsymbol{x}$ according to i

Summary: The Lewis thinning algorithm

1. Initialize the time $t = t_0$ and the system's state $\boldsymbol{x} = \boldsymbol{x_0}$.
2. With the system in state $\boldsymbol{x}$ at time t, for each reaction, find a constant which is always larger than its propensity
   $\bar{a}_i \geq a_i(\boldsymbol{x}, s) \quad \forall s \geq t$
3. Compute the sum $\bar{a}_0 = \sum_i \bar{a}_i$
4. Generate a random value for τ using an exponential distribution with parameter $\bar{a}_0$
5. Evaluate all the propensities at time $(t+\tau)$: $a_i(\boldsymbol{x}, t+\tau)$
6. Generate a random value for i using the distribution $\frac{a_i(\boldsymbol{x}, t+\tau)}{\bar{a}_0}$
7. Update $t \leftarrow t + \tau$ and $\boldsymbol{x}$ according to i
8. Save $(\boldsymbol{x}, t)$ as desired and return to Step 2, or else end the simulation

- Alternatively, one can use the τ-leaping method: consider that in a small time interval of size τ, the rates are constant.

τ-leaping method

- Alternatively, one can use the τ-leaping method: consider that in a small time interval of size τ, the rates are constant.
- ⚠Not the same τ

τ-leaping method

- Alternatively, one can use the τ-leaping method: consider that in a small time interval of size τ, the rates are constant.
- ⚠Not the same τ
- τ-leaping has other interesting properties, especially for large populations.

Exact vs inexact methods

- A method is exact if it is derived **without approximation** from the fundamental premise of stochastic kinetics $a_i dt$.

Exact vs inexact methods

- A method is exact if it is derived **without approximation** from the fundamental premise of stochastic kinetics $a_i dt$.
- The probability of generating a given trajectory with an exact method is exactly the probability that would come out of the solution of the master / Kolmogorov forward equation.

Exact vs inexact methods

- A method is exact if it is derived **without approximation** from the fundamental premise of stochastic kinetics $a_i dt$.
- The probability of generating a given trajectory with an exact method is exactly the probability that would come out of the solution of the master / Kolmogorov forward equation.
- However, exact methods are usually slow for large dimensions and/or when the transitions occur very often (large population sizes for example). One transition at a time.

Logical structure of stochastic kinetics