

Markov jump processes

Russ Harmer

A random variable $X(t)$ is *memoryless* iff $\Pr(X > t_1 + t_2 \mid X > t_1) = \Pr(X > t_2)$. Equivalently, $\Pr(X > t_1 + t_2) = \Pr(X > t_1) \cdot \Pr(X > t_2)$ which makes it clear that $\Pr(X > t_2)$ is independent of t_1 .

What form can such an X take? Well, $\Pr(X > t + dt)$ can be written in the form $\Pr(X > t) \cdot (1 - \alpha \cdot dt)$, for some positive α , and so

$$\frac{\Pr(X > t + dt) - \Pr(X > t)}{dt} = -\alpha \cdot \Pr(X > t)$$

and therefore $\Pr(X > t) = e^{-\alpha t}$. The *exponential* random variable $\exp(\alpha)$, where $\alpha \in [0, \infty)$, is therefore the only memoryless (continuous) random variable. Its pdf is $\alpha \cdot e^{-\alpha t}$ and cdf is $\Pr(\exp(\alpha) \leq t) = 1 - e^{-\alpha t}$.

For our purposes, a Markov jump process is just a finite, directed (simple) graph with positive real number weights on its edges. A *state* of the process is thus characterized by a finite set of outgoing *transitions* τ_j . The fact that each *transition* is equipped with a *rate* α_j necessarily means that that rate is time-independent, *i.e.* memoryless, and so our process is *time-homogeneous*.

The choice of *next state* s' and of *time advance* δt is decided by the ‘race’ between the τ_j s to make the first transition: given n exponential random variables $T_j \sim \exp(\alpha_j)$, the probability that ‘ T_i has the smallest value and that this is in $[t_i, t_i + dt)$ ’ is

$$\begin{aligned} \prod_{j \neq i} \Pr(T_j > t_i) \cdot \Pr(T_i \in [t_i, t_i + dt)) &= \prod_{j \neq i} e^{-\alpha_j t_i} \cdot \alpha_i e^{-\alpha_i t_i} dt \\ &= \alpha_i \prod_{j \neq i} e^{-\alpha_j t_i} dt \\ &= \alpha_i e^{-\alpha t_i} dt \end{aligned}$$

where $\alpha := \sum_j \alpha_j$.

The probability that ‘ T_i has the smallest value’ is therefore

$$\alpha_i \int_0^\infty e^{-\alpha t} dt = \frac{\alpha_i}{\alpha} \tag{1}$$

while the probability that ‘the smallest value is in $[t, t + dt)$ ’ is

$$\sum_j \alpha_j e^{-\alpha t} dt = \alpha e^{-\alpha t} dt. \tag{2}$$

Equations (1) and (2) tell us that we do not strictly need to generate the n individual samples t_j . It is enough (i) to sample $[0, \alpha]$ uniformly to select the ‘winning’ i ; and (ii) to sample $\exp(\alpha)$ to generate the time advance δt which is, morally, the phantom sample t_i of the ‘winning’ T_i . However, the net effect would be the same as the pedestrian approach of generating all t_j s at each step and simply selecting the ‘winning’ t_i .

An alternative alternative to the pedestrian approach is to try to keep the *same* t_j s (for all $j \neq i$; we have no choice but to generate a new t_i) after the transition to the new state s' . However, we also need to take into account any changes to our parameters, *i.e.* those j s for which $\alpha_j \neq \alpha'_j$. [This means that transitions must be labeled by the j s as well.] For any such j , the *memoryless*-ness of T_j means that it is distributed according to

$$\begin{aligned} \Pr(T_j > t_i + (t - t_i) \mid T_j > t_i) &= \Pr(T_j > t - t_i) \\ &= e^{-\alpha_j(t-t_i)} \end{aligned} \quad (3)$$

and so, by (a shift and) *rescaling*, $t'_j := \frac{\alpha_j}{\alpha'_j} \cdot (t_j - t_i)$ is distributed according to

$$e^{-\alpha_j t / \frac{\alpha_j}{\alpha'_j}} = e^{-\alpha'_j t} \quad (4)$$

and the final new absolute timer $t'_j := t'_j + t_i$ is distributed according to $e^{-\alpha'_j(t-t_i)}$ as required.

Chemical kinetics

Consider a system of *reactions* $R = r_1, \dots, r_n$ acting on *molecular species* $S = s_1, \dots, s_m$.

A *state* of the system is a function $s : S \rightarrow \mathbb{Z}_0^+$. Each r_i has (i) a *rate constant* $\gamma_i \in \mathbb{R}^+$; (ii) a *propensity function* $\pi_i : (S \rightarrow \mathbb{Z}_0^+) \rightarrow \mathbb{Z}_0^+$; and an *update vector* $\nu_i : S \rightarrow \mathbb{Z}$.

The propensity function π_i tells us how many instances of r_i exist in a state. Its definition is obvious in the case of mono-molecular reactions $A \rightarrow \dots$ or hetero-bi-molecular reactions $A + B \rightarrow \dots$: in the first case, we set $\pi(s) := s(A)$ whereas, in the second case, we set $\pi(s) := s(A) \cdot s(B)$.

What about the case of homo-bi-molecular reactions $A + A \rightarrow \dots$? The answer depends on whether we wish to assert a *symmetric* or an *asymmetric* reaction mechanism: in the first case, each *unordered* pair of A s gives rise to an instance of the reaction; whereas, in the second case, each *ordered* pair gives rise to an instance. We have no way to *represent* this difference; it must be encoded in the numerical value of the rate constant once we have chosen between $\pi(s) := s(A)(s(A) - 1)$ and $\pi(s) := \frac{1}{2}s(A)(s(A) - 1)$.

We transform state s by firing reaction r_i by setting $s'(s_j) := s(s_j) + \nu_i(s_j)$. This defines the *transition* $\tau_{s,i}$ iff $s'(s_j) \in \mathbb{Z}_0^+$ for all $s_j \in S$, *i.e.* s' is a state; we assign $\tau_{s,i}$ the *rate* $\alpha_i(s) := \gamma_i \cdot \pi_i(s)$.

Gillespie's 'first reaction' method

Requires n random numbers per iteration:

- the *timer* t_i in state s for each r_i is distributed according to $\exp(\alpha_i(s))$;
- the next reaction to fire is the r_i with the smallest timer.

Gillespie's 'direct' method

Requires two random numbers per iteration:

- the *time advance* in state s is distributed according to $\exp(\alpha(s))$ where $\alpha(s) := \sum_i \alpha_i(s)$;
- the *next reaction* to fire is r_i with probability $\frac{\alpha_i(s)}{\alpha(s)}$.

This is equivalent to the 'first reaction' method by (2) and (1) above.

Gibson-Bruck's 'next reaction' method

Uses absolute timers. Requires n random numbers for initialization; then one per iteration:

- the next reaction to fire is the r_i with the smallest timer;
- for $j \neq i$, update $t'_j := \frac{\pi_j(s)}{\pi_j(s')} (t_j - t_i) + t_i$;
- generate a new timer t'_i for r_i according to $\exp(\alpha_i(s'))$.

This is equivalent to the 'first reaction' method by (3) and (4) above. In practice, only those t_j s for which $\pi_j(s) \neq \pi_j(s')$ need be updated; identifying these j s is the role of Gibson-Bruck's 'dependency graph'.

The 'next event' method

Exploits the decomposition of $\alpha_j(s)$ as $\gamma_j \cdot \pi_j(s)$ to consider each *instance* of a reaction (or *event*) as a transition in its own right with rate γ_j .

Requires $\pi(\iota) := \sum_j \pi_j(\iota)$ random numbers for initialization (where ι is the initial state); and a few more at each iteration.

- the next event to fire is the one with the smallest timer;
- remove (uniformly at random) the appropriate number of *conflicting* events;
- add the appropriate number of *caused* events with their appropriate timers.

This is equivalent to the ‘next reaction’ method:

In state s , each r_j has $\pi_j(s)$ events whose ‘winner’ has timer t_j therefore distributed according to $\exp(\gamma_j \cdot \pi_j(s))$. After an r_i event with overall winning timer t_i taking the system to state s' , we now have $\pi_j(s')$ events whose winner has timer t'_j : this could be the same winner as before—if there is no conflict between r_i and r_j or, more generally, if there is conflict but that winning event was lucky enough not to be removed—and could even be a particularly fast newly-created r_j event—if r_i causes r_j events.

In any case, $\pi_j(s') = p + c$ where p is the number of *preserved* events and c is the number of *newly-created* events. The winner of the preserved events has timer $t_p \sim \exp(\gamma_j \cdot p) =: T_p$ and the winner of the newly-created events has timer $t_c \sim \exp(\gamma_j \cdot c)$. Since T_p is memoryless, $t'_j \sim \exp(\gamma_j \cdot (p + c))$.

We see here that the rescaling step in the ‘next *reaction*’ method is an artifact of coalescing all r_j events into a single timer; the ‘correction’ effected by the factor $\frac{\pi_j(s)}{\pi_j(s')}$ occurs in an immanent fashion in the ‘next *event*’ method by the overall loss or gain of r_j events after an r_i transition.