Simulation of CTMC

Monday, November 23, 2015
1:55 PM

Homework 4 will be posted by tomorrow morning, due Friday, December 11 at 5 PM.

Let's solve the Kolmogorov forward equation for the Poisson counting process via Laplace transform.

$$\tilde{\phi}_j(z) = \int_0^\infty e^{-zt}\,\phi(t)\,dt$$

for $\operatorname{Re} z > 0.$

Laplace transforms of derivatives behave nicely; they convert differentiation to multiplication:

$$\int_0^\infty e^{-zt}\frac{d\phi}{dt}\,dt = -\int_0^\infty -ze^{-zt}\phi(t)\,dt + e^{-zt}\phi(t)\Big|_0^\infty$$

int by parts

$u = e^{-zt}$ $dv = \frac{d\phi}{dt}\,dt$

$du = -ze^{-zt}$ $v(t) = \phi$

$$= z\int_0^\infty e^{-zt}\phi(t)\,dt + (0 - \phi(0)$$

$$= z\,\tilde{\phi}(z) - \phi(0)$$

Apply this to our differential equations for $j \geq 1$.

$$\frac{d\phi_j}{dt} = -\lambda\phi_j + \lambda\phi_{j-1}$$

$$z\tilde{\phi}_j(z) - \phi_j(0) = -\lambda\tilde{\phi}_j(z) + \lambda\tilde{\phi}_{j-1}(z)$$

This is now an algebraic equation that can be solved simply:

$$\tilde{\phi}_j(z) = \frac{\overset{0}{\cancel{\phi_j(0)}} + \lambda\tilde{\phi}_{j-1}(z)}{\lambda + z}$$

But $\phi_j(0) = 0$ for $j \geq 1$ because $X(0) = 0$ for

Poisson process.

$$\hat{\phi}_j(z) = \hat{\phi}_{j-1}(z)\left(\frac{\lambda}{\lambda+z}\right)$$

$$\Rightarrow \hat{\phi}_j(z) = \hat{\phi}_0(z)\left(\frac{\lambda}{\lambda+z}\right)^j$$

$$\hat{\phi}_0(z) = \int_0^\infty e^{-zt}\,\phi_0(t)\,dt$$

$$= \int_0^\infty e^{-zt}\,e^{-\lambda t}\,dt$$

$$= \int_0^\infty e^{-(z+\lambda)t}\,dt = \frac{1}{\lambda+z}$$

$$\therefore \quad \hat{\phi}_j(z) = \frac{\lambda^j}{(z+\lambda)^{j+1}}$$

How do we recover the solution with respect to the original time variable? Some general purpose procedures:

- Look up tables.
- Contour integration over the Bromwich contour.
- Or sometimes you can use tricks to relate the Laplace transform you are considering to a known Laplace transform.

Note that $\frac{1}{z+\lambda}$ is the Laplace transform of $e^{-\lambda t}$.

Then observe that:

$$\hat{\phi}_j(z) = \frac{(-\lambda)^j}{j!}\left(\frac{d}{dz}\right)^j \frac{1}{z+\lambda}$$

$$= \frac{(-\lambda)^j}{j!}\left(\frac{d}{dz}\right)^j \int^\infty e^{-zt}\,e^{-\lambda t}\,dt$$

$$= \frac{(-\lambda)^j}{} \int^\infty (-t)^j\,e^{-zt}\,e^{-\lambda t}\,dt$$

$$= \frac{(-\lambda)^j}{j!} \int_0^\infty (-t)^j e^{-zt} e^{-\lambda t}\, dt$$

$$\hat{\phi}_j(z) = \int_0^\infty e^{-zt} \left(\frac{\lambda^j t^j}{j!} e^{-\lambda t} \right) dt$$

$$= \int_0^\infty e^{-zt} \phi_j(t)\, dt$$

Since the Laplace transform is invertible.

$\phi_j(t) = \frac{\lambda^j t^j}{j!} e^{-\lambda t}$ for $j \geq 1$ in fact also for $j \geq 0$.

More calculations involving the solution of probability distributions of CTMCs by solving Kolmogorov forward equations can be found in Karlin and Taylor Sec. 4.1.

- Another common trick for countable state CTMCs with relatively simple structure is to take the probability generating function w.r.t. to the state (rather than Laplace transform in time) to get a function $G(s,t) = \mathbb{E}s^{X_j(t)}$; for some models this will satisfy a linear PDE which you can solve (sometimes) by PDE techniques.

The exact solution techniques only work for sufficiently simple CTMCs; in general one needs to employ computational linear algebra to extract eigenvalues of the transition rate matrix or solve linear systems in order to represent the solution, and/or solve the various absorption probability, etc. problems associated to CTMCs.

However, the results from our Poisson counting process calculation have broader implication.

What do the results say?

- We see that the probability distribution for the state of the Poisson counting process is Poisson distributed with mean λt.
- The probability to still be in state 0 after a time t is exponential. In other words, the amount of time that the Poisson counting process spends in state 0 (Which is where it started) is exponentially distributed with mean $\frac{1}{\lambda}$.

$$e^{-\lambda t} = \phi_0(t) = P(X(t) = 0)$$

$$= P(T_1 > t)$$

where $T = \min\{t > 0 : X(t) \neq X(0)\}$

PDF for τ_1

$$P(\tau_1 > t) = \int_t^\infty p_{\tau_1}(t')\,dt'$$

$$\frac{d}{dt}: \quad e^{-\lambda t} = \int_t^\infty p_{\tau_1}(t')\,dt'$$

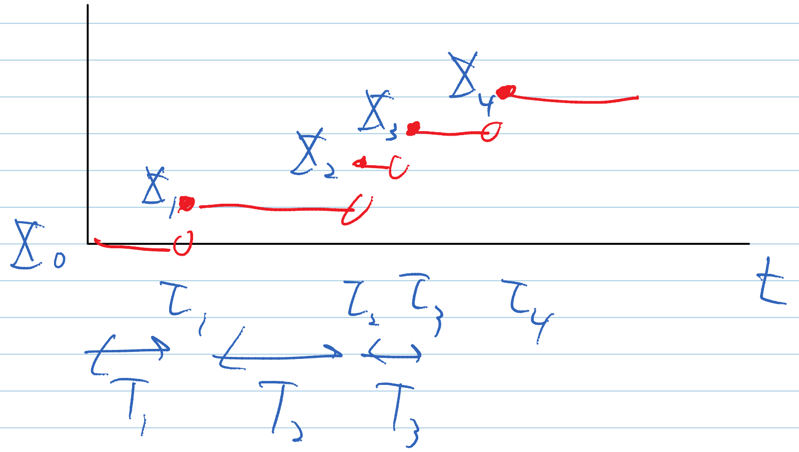
$$-\lambda e^{-\lambda t} = -p_{\tau_1}(t) \qquad \text{for any } t > 0$$

$$p_{\tau_1}(t) = \begin{cases} \lambda e^{-\lambda t} & \text{for } t > 0 \\ 0 & \text{else} \end{cases}$$

- The previous result can be generalized starting from any state $j \neq 0$ by either redoing the calculation with $X(s) = j$ for some $s > 0$, or observing the symmetry among states, i.e., that the dynamics of the Poisson counting process does not explicitly depend on the value of the state it starts in. Either argument, together with the strong Markov property, yields the following result:

  Let $T_j = \tau_j - \tau_{j-1}$ where $\tau_j = \min\{t > \tau_{j-1} : X(t) \neq X(\tau_{j-1})\}$
  $\tau_0 = 0$.

  Then T_j measures the length of the time interval during which $X(t)$ remains in state $X_{j-1} = X(\tau_{j-1})$.



$T_j \sim Exp\left(\frac{1}{\lambda}\right)$ (exponentially distributed with mean $1/\lambda$), and the

T_j are independent random variables.

So first of all, we see that the Poisson counting process precisely counts the number of incidents generated by a Poisson point process which we defined previously. (Poisson point process generates a sequence of "incidents" along the time axis such that the time between incidents is exponentially distributed with a prescribed mean. It is the continuum limit of a Bernoulli process.)

Secondly, the fact that we obtained exponential distributions for the amount of time spent in a state of the CTMC is not accidental -- it is the only waiting time probability distribution in continuous time consistent with the memoryless property of the CTMC (in particular Poisson counting process). One can define a more general point process called a renewal process which has non-exponential waiting times between incidents, but the renewal process only refreshes its memory with each incident, not with each time. (Resnick Ch. 3).

Now what can we carry over from this Poisson counting process example to more general CTMCs?

1. By the memoryless argument, the amount of time T_n spent in any given state must be exponentially distributed.
2. Also by the memoryless argument, the time variable T_j is conditionally independent of previous time variable $\{T_1, T_2, \ldots, T_{n-1}\}$, given X_{n-1}, the current state value of the CTMC. Actually by a deeper analysis, one can show the $\{T_n\}$ are unconditionally independent, though this isn't particularly helpful fact in practice.
3. The mean of the time T_n spent in state X_{n-1} is $(\lambda_{X_{n-1}})^{-1}$, the inverse of the total transition rate out of state X_{n-1}. Can show this by repeating the Kolmogorov forward equation calculation starting from a given state i, making all other states absorbing.

$$\frac{d\phi_i}{dt} = -\overline{\lambda}_i \phi_i \qquad \Rightarrow \phi_i(t) = e^{-\overline{\lambda}_i t}$$

$$\phi_i(0) = 1$$

4. The probability distribution for X_n, given X_{n-1} is given by

$$P(X_n = j | X_{n-1} = i) = \frac{A_{ij}}{\lambda_i}.$$

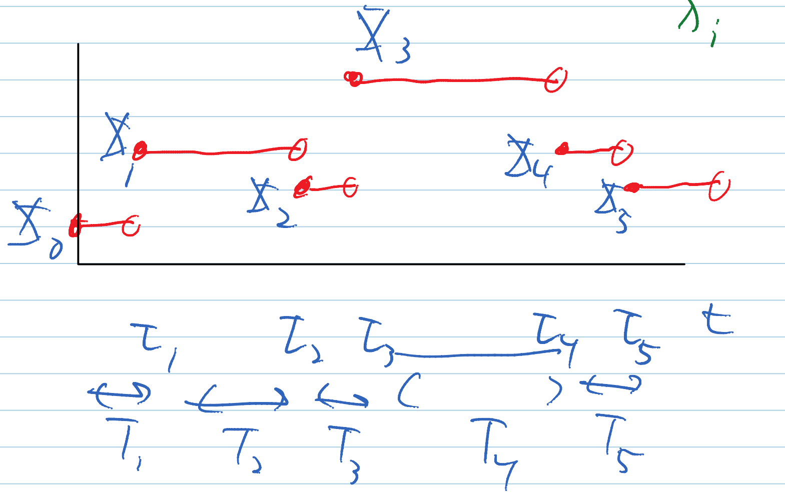
This also will result from the FKE calculation in the previous step.

$j \neq i \qquad \frac{d\phi_j}{dt} = A_{ij}\phi_i(t)$

$$\phi_j(t) = A_{ij}\int_0^t \phi_i(t')\,dt' = \lambda_{ij}\int_0^t e^{-\lambda_i t}$$

$$= A_{ij}\left(\frac{1-e^{-\lambda_i t}}{\lambda_i}\right)$$

$$P(X_n=j \,|\, X_{n-1}=i) = \lim_{t\to\infty}\phi_j(t) = \frac{A_{ij}}{\lambda_i}$$



These observations allow us to transfer information about the transition rate matrix (the entries A_{ij}) to the stochastic dynamics of the embedded DTMC with timing: $\{(X_0,\tau_0),(X_1,\tau_1),\dots,(X_n,\tau_n),\dots\}$. And this description is well-suited for doing event-based (Gillespie, kinetic Monte Carlo) simulation of CTMCs. Event-based approaches just update the CTMC at transitions, not at some regular time interval Δt.

Two key variations on event-based simulation for CTMCs:

1. First event simulation method

    Start with $\tau_0 = 0$, X_0 is prescribed (possibly random) by initial conditions.

    Then use the following procedure to update from (X_n,τ_n) to (X_{n+1},τ_{n+1}):

    - If $X_n = i$ (current state), generate a $T_{n+1} \sim Exp((\lambda_i)^{-1})$
    - $\tau_{n+1} = \tau_n + T_{n+1}$
    - X_{n+1} is to be sampled from the conditional probability distribution:
        - $P(X_{n+1} = j | X_n = i) = \dfrac{A_{ij}}{\lambda_i}$
    - Note that given X_n, the random variables T_{n+1} and X_{n+1} are independent of each other (can be shown by doing conditional calculation with the forward FKE with absorption, at finite time t, indicated above).

2. Next event simulation method

Start with $\tau_0 = 0$, X_0 is prescribed (possibly random) by initial conditions.

Then use the following procedure to update from (X_n, τ_n) to (X_{n+1}, τ_{n+1}):

- If $X_n = i$ (current state), then for every possible next state j (meaning every j for which $A_{ij} \neq 0$), generate $T^{(j)} \sim Exp((A_{ij})^{-1})$
- $T_{n+1} = \min_{j:A_{ij}\neq 0} T^{(j)}, \tau_{n+1} = \tau_n + T_{n+1}$
- $X_{n+1} = \arg\min_{\mathrm{j:A_{ij}\neq 0}} T^{(j)}$

Proving that the first event and next event simulation methods are equivalent amounts to an elementary probability calculation on the minimum of a collection of independent exponentially distributed random variables with different means; Lawler Ch. 3 does this.

Relative advantages of the two event-based approaches?

- first event simulation method seems more efficient because it only generates two random variables per update
- However, the first event simulation approach requires the simulation of a discrete random variable whereas the next event simulation approach only requires the simulation of exponential random variables
  - Simulating an exponential random variable with mean μ can be done simply by taking a uniform random number U on $(0,1)$ (which is supplied by any basic pseudorandom number generator) and taking $Y = -\mu \ln U$. The justification of this approach is the Inverse Transform Method.
- For simple systems, first event simulation method is probably faster but for more complex systems, people use both approaches
  - The expense of generating the probability distribution for the next state and sampling from it (as in first event simulation) can become awkward for large CTMCs (reaction networks).
  - One can more easily parallelize the next event simulation approach.
  - Also, one can often recycle the alarm clocks, especially as part of a more sophisticated parallelization of the reaction network.