EEMB247/BMSE247 Notes on Stochastic Models

A stochastic model is formulated as a stochastic process with a collection of **random variables**.

Random variable: A ***random variable***, is a variable whose possible values are numerical outcomes of a random phenomenon.

Whereas the **solution** of a deterministic model is a function of time (or space, or whatever), and is generally uniquely dependent on the initial conditions, the solution of a stochastic model is a **probability distribution for each of the random variables**. One sample path over time (or space) is one **realization** from this distribution.

Markov Property: A stochastic process with a Markov property is one where the future state of the process depends only on the current state, and not on the past.
For a discrete-time stochastic process, for the random variable X(t), this can be expressed as:
$\mathrm{Prob}\{X(t+\Delta t)|X(t),X(t-\Delta t),\ldots,X(0)\} = \mathrm{Prob}\{X(t+\Delta t)|X(t)\}$
(where | means "given" (conditional probability))

Random Variables can take on discrete or continuous states.
At any particular time t, each random variable X(t) has a **probability distribution**:
Discrete: $\mathrm{Prob}\{X(t) = i\} = p_i(t)$ for i in {0,1,2,…}

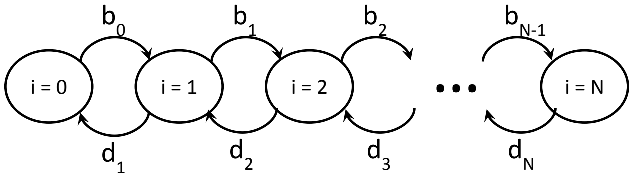
Continuous: $\mathrm{Prob}\{X(t) \in [a,b]\} = \int_a^b p(x,t)dx$

(where[a,b] means "is in the interval a to b")
Time can also be discrete or continuous.

Discrete-time Markov Chain:
X(t) is a discrete random variable: $X(t) \in \{0,1,2,\ldots\}$
Time occurs in discrete steps: $t \in \{0, \Delta t, 2\Delta t,\ldots\}$
(where ∈ means "is an element of")

Continuous-time Markov Chain:
X(t) is a discrete random variable: $X(t) \in \{0,1,2,\ldots\}$
Time is continuous: $t \in [0, \infty)$

Diffusion Process, Stochastic Differential Equation:
X(t) is a continuous random variable: $X(t) \in [0, N]$
Time is continuous: $t \in [0, \infty)$



One-step or birth-death process

Consider a continuous-time stochastic process in which X(t) is a discrete random variable $X(t) \in \{0,1,2,\dots\}$, with permissible jumps between adjacent integers only, and (time-dependent) transition rates d_i for i → i-1 and b_i for i → i+1:

For small Δt, the probability $p_i(t+\Delta t)$ that the system will be in state i at time t+Δt is given by:

(Equation 1) $p_i(t+\Delta t) = (d_{i+1}\Delta t)\, p_{i+1}(t) + (b_{i-1}\Delta t)\, p_{i-1}(t) + (1 - d_i\Delta t - b_i\Delta t)\, p_i(t) + o(\Delta t)$

(where o(Δt) means "of order Δt", which means $\lim_{\Delta t \to 0} \frac{o(\Delta t)}{\Delta t} = 0$
o(Δt) approaches zero faster than Δt.
generally this represents all of the bits of the equation that are negligible.)

subtracting $p_i(t)$, dividing by Δt, and letting Δt→0 gives us:
(Equation 2) $dp_i/dt = d_{i+1}\, p_{i+1} + b_{i-1}\, p_{i-1} - (d_i + b_i)\, p_i$
This is the **Master equation** of the process, which is an equation for the time evolution of the probability distributions. It is also known as the forward Kolmogorov differential equation. A solution requires the specification of the initial time values of the probabilities.

Example: **Homogeneous Poisson process:** (Poisson process with a constant rate r)
$d_i = 0$, $b_i = r$, with initial condition $p_i(0) = 1$ for i=0, $p_i(0) = 0$ for $i \neq 0$.
The Master equation is: $dp_i/dt = r\,(p_{i-1} - p_i)$
This system of equations can be solved recursively starting with the equation for p_0:
$dp_0/dt = - r\, p_0$, and then proceeding to the solution for p_1, and so on, ending up with the probability of being in state i at time t:
$p_i(t) = (rt)^i \exp(-rt)/i!$
This is the **Poisson distribution** with mean = rt and variance = rt.

Another example: "Birth and death" process: Production and decay of mRNA
Transcription of gene D into mRNA (M), and then subsequent translation into proteins (P):

$$D \xrightarrow{k_1} D + M$$
$$M \xrightarrow{k_2} M + P$$

with reaction rates k_1 and k_2. mRNA and Proteins are degraded through reactions:

$$M \xrightarrow{k_3} 0$$
$$P \xrightarrow{k_4} 0$$

with decay rates k_3 and k_4 (which means the half-lives are: $t_2 = \ln(2)/k_3$ and $t_4 = \ln(2)/k_4$.

If we assume that we are dealing with a large number of molecules, we could write this as a system of deterministic ordinary differential equations for the concentrations [M] of mRNA and [P] of protein P:
$d[M]/dt = k_1[D] - k_3[M]$
$d[P]/dt = k_2[M] - k_4[P]$

We assume that D is constant, and this system has a stable equilibrium at $[M] = (k_1/k_3)[D]$, and $[P] = (k_2/k_4)[M]$.
If instead, we assume that there are few copies of some of these molecules, we could consider the production and decay of mRNA as a stochastic, birth-death process:
Let $p_n(t)$ be the probability of having n molecules of mRNA at time t. Assume D = 1. The corresponding Master equation is:
(Equation 3) $dp_n/dt = k_1[p_{n-1} - p_n] + k_3[(n+1)p_{n+1} - np_n]$

The steady state solution, which satisfies $(n+1)p_{n+1} = np_n + (k_1/k_3)\,[\,p_n - p_{n-1}]$
can be found recursively:
$p_1 = (k_1/k_3)p_0$
$p_2 = (1/2)\,(k_1/k_3)^2\,p_0$
$p_n = (1/n!)\,(k_1/k_3)^n\,p_0$
with $p_{-1} = 0$
Normalization of this probability function (i.e. making sure that the sum of the probabilities adds up to 1) then gives $p_0 = \exp(-k_1/k_3)$
The steady state distribution of M thus follows a **Poisson distribution**, with average steady state value equal to (k_1/k_3).

These differential equations describe the rate of change of the **probability** that the system will be in a given state at any point in time.
How do you run one realization of this system, from a given starting condition?

Method 1: Approximate stochastic realization by dividing the system into discrete time steps:
One option would be to discretize time into small time steps Δt, and use the information that starting at a given number of mRNA molecules M = i:
the probability of increasing M from $i \rightarrow i + 1$ is: $k_1 {*} D {*} \Delta t$
the probability of decreasing M from $i \rightarrow i - 1$ is: $k_3 {*} i {*} \Delta t$
the probability of staying at M is: $(1 - k_1 {*} D {*} \Delta t - k_3 {*} i {*} \Delta t)$

The recipe for generating a single stochastic realization using Method 1 is:

1. Assign values to you parameters, k_1, k_3, and D (you can set D=1), and set your time increment Δt. (Note: $k_1 {*} D {*} \Delta t + k_3 {*} i {*} \Delta t$ has to be less than or equal to 1, so if k_1 or k_3 is large, then you'll have to use a small time step, Δt.)
2. Initialize M to some starting value. (If you want to save the values of M at each time step, for example to plot them out at the end, then set up some type of R structure in which to hold the results (e.g. a vector or an array, or just Mvalues<-numeric(0)).
3. Loop through time from a start time (e.g. 0) to a stop time in increments of Δt (perhaps using a for loop). In each iteration of the loop do the following:
    a. calculate the probability of M increasing by 1 (this actually doesn't change through time)
    b. calculate the probability of M decreasing by 1 (this depends on the current value of M)
    c. calculate the probability of M staying the same
    d. roll a 3-sided die and depending on the outcome either increase M, decrease M, or have M remain unchanged.

4. Plot your results through time.

Method 2: Exact stochastic realization using Gillespie's algorithm:
Although Method 1 is easy to understand and implement, it is just an approximation and assumes that Δt is small enough that the probability that two events (births and deaths) occur within Δt is negligible. The Gillespie algorithm (Gillespie 1977) is an exact alternative that instead treats time as continuous and uses two random numbers to determine (i) the time to the next event and (ii) which event occurs.

The general set-up for this Stochastic Simulation Algorithm (SSA) was first formulated for chemical systems, but it applies to ecological systems equally well (and has been used in a number of ecological models). The set-up is as follows (from: Li and Petzold, Bioinformatics 2005):
Consider a spatially homogeneous chemically reacting system with a fixed volume and at a constant temperature.

The system involves N molecular species: $\{S_1, \ldots, S_N\}$, represented by the dynamical state vector $X(t) = (X_1(t), \ldots, X_N(t))$, where $X_i(t)$ is the population of species S_i in the system at time t.

In the system being modeled, there are K chemical reactions, $\{R_1, \ldots R_K\}$. Each reaction R_j is characterized by:
a propensity function a_j, where $a_j(x)dt$ is the probability, given the state of the system at time t, that one R_j reaction will occur in the next infinitesimal time interval [t,t+dt), and
a state change vector $v_j = \{v_{1j}, \ldots, v_{Nj}\}$, in which v_{ij} is the change in the number of species S_i due to one R_j reaction.

For any given current state of the system, $X(t) = x_t$, the time τ to the next reaction is exponentially distributed with mean = $1/a_{tot}(x_t)$, where $a_{tot}(x_t)$ is the sum of the propensities of all of the possible reactions: $a_{tot}(x_t) = \sum_{j=1}^{K} a_j(x_t)$.

If u_1 is a uniformly distributed random number in the interval [0,1], then at any point in time, a value of τ (time to the next reaction) can be generated by: $\tau = \frac{1}{a_{tot}(x_t)} \ln\left(\frac{1}{u_1}\right)$.

The probability that the next reaction is of type j is: $P(j|x,t) = \frac{a_j(x_t)}{a_{tot}(x_t)}$, for each reaction (j = 1,…K).

If u_2 is a second uniformly distributed random number in the interval [0,1], then the type of the next reaction can be determined by rolling a K-sided die, with the probability of the die landing on each of the K sides being P(j|x,t) for j = 1,…K.

The recipe for generating a single stochastic realization of the production and decay of mRNA model using The Gillespie Algorithm is:

1. Assign values to you parameters, k_1, k_3, and D (you can set D=1).

2. Initialize the state of the system to some starting value. In this case we have only a single state variable X(t) = M(t).
   (Again, if you want R to save the values of M at each time step, for example to plot them out at the end, then set up some type of R structure in which to hold the results (e.g. a vector or an array, or just `Mvalues<-numeric(0)`)).
3. Initialize time to t = 0
   (If you want to plot your results through time, you'll also want to save the value of time at each time step, e.g. `tvalues<-numeric(0)`)
4. Loop through time, in this case because time is going to jump forward in uneven increments, perhaps use a `while` loop (e.g. `while (t<Tend)`, where `Tend` is whatever final time you choose for your simulations). In each iteration of the loop do the following:
   a. Calculate the current values of the propensity functions based on the current state of the system.
      For this example, we have 2 possible reactions (we are ignoring the reactions that don't affect the concentration of M):
      R_1 = 1: $D \xrightarrow{k_1} D + M$
      R_2 = 1: $M \xrightarrow{k_3} 0$
      The propensities of these two reactions are:
      $a_1 = k_1 * D$
      $a_2 = k_3 * M$
      $a_{tot} = a_1 + a_2$
   b. Generate two numbers u_1 and u_2 from a uniform distribution on the interval [0,1].
   c. Determine the time to the next event, $\tau = \frac{1}{a_{tot}(x_t)} \ln\left(\frac{1}{u_1}\right)$
   d. Determine which reaction occurs: if $u_2 < (a_1/a_{tot})$ then reaction 1 occurs, if $u_2 >= (a_1/a_{tot})$, then reaction 2 occurs.
   e. Update time: $t = t + \tau$
   f. Update the state of the system, depending on which reaction occurs. If reaction 1 occurs, M increases by 1. If reaction 2 occurs, M decreases by 1.
5. Plot your results through time.