Discrete-Time Markov Chains

Norman Matloff
(adapted from
Probability Modeling and Computer Simulation
N.S. Matloff, PWS, 1988)

One of the most commonly used stochastic models is that of a **Markov chain**. To motivate this discussion, we will start with a simple example: Consider a **random walk** on the set of integers between 1 and 5, moving randomly through that set, say one move per second, according to the following scheme. If we are currently at position i, then one time period later we will be at either i-1, i or i+1, according to the outcome of rolling a fair die—we move to i-1 if the die comes up 1 or 2, stay at i if the die comes up 3 or 4, and move to i+1 in the case of a 5 or 6. For the special cases of i = 1 and i = 5, we simply move to 2 or 4, respectively.

The integers 1 through 5 form the **state space** for this process; if we are currently at 4, for instance, we say we are in state 4. Letting X_t represent the position of the particle at time t, t = 0, 1,2,... which is called the **state** of the process at time t.

The random walk is a **Markov process**. The term *Markov* here has meaning similar to that of the term *memoryless* used for the exponential distribution, in that we can "forget the past":

$$P(X_{t+1} = s_{t+1} | X_t = s_t, X_{t-1} = s_{t-1}, \ldots, X_0 = s_0) = P(X_{t+1} = s_{t+1} | X_t = s_t) \tag{1}$$

Although this equation has a very complex look, it has a very simple meaning: The distribution of our next position, given our current position and all our past positions, is dependent only on the current position. It is clear that the random walk process above does have this property; for instance, if we are now at position 4, the probability that our next state will be 3 is 1/3—no matter where we were in the past.

Continuing this example, let p_{ij} denote the probability of going from position i to position j in one step. For example, $p_{21} = p_{23} = \frac{1}{3}$, while $p_{24} = 0$ (we can reach position 4 from position 2 in two steps, but not in one step). The numbers p_{ij} are called the **one-step transition probabilities** of the process. Denote by P the matrix whose entries are the p_{ij}.

In typical applications we are interested in the long-run distribution of the process, for example, the long-run proportion of the time that we are at position 4. For each state i, define

$$\pi_i = \lim_{t \to \infty} \frac{N_{it}}{t} \tag{2}$$

where N_{it} is the number of visits the process makes to state i among times 1, 2,..., t. In most practical cases, this proportion will exist and be independent of our initial position X_0. (There are mathematical conditions under which this is guaranteed to occur, but they will not be stated here.)

Intuitively, the existence of π_i implies that as t approaches infinity, the system approaches steady-state, in the sense that

$$\lim_{t \to \infty} P(X_t = i) = \pi_i \tag{3}$$

Though we will again avoid discussing mathematical conditions for this to occur, the point here is that this last equation suggests a way to calculate the values π_i, as follows.

First note that

$$P(X_{t+1} = i) = \sum_k P(X_t = k) p_{ki} \tag{4}$$

Then as $t \to \infty$ in this equation, intuitively we would have

$$\pi_i = \sum_k \pi_k p_{ki} \tag{5}$$

Letting π denote the row vector of the elements π_i, these equations (one for each i) then have the matrix form

$$\pi = \pi P \tag{6}$$

Note that there is also the constraint

$$\sum_i \pi_i = 1 \tag{7}$$

This can be used to calculate the π_i. For the random walk problem above, for instance, the solution is $[\frac{1}{11}, \frac{3}{11}, \frac{3}{11}, \frac{3}{11}, \frac{1}{11}]$. Thus in the long run we will spend 1/11 of our time at position 1, 3/11 of our time at position 2, and so on.

In the above example, the labels for the states consisted of single integers i. In some other examples, convenient labels may be r-tuples, for example 2-tuples (i,j).

Example:

Consider a serial communication line. Let $B_1, B_2, B_3, ...$ denote the sequence of bits transmitted on this line. It is reasonable to assume the B_i to be independent, and that $P(B_i = 0)$ and $P(B_i = 1)$ both being equal to 0.5.

Suppose that the receiver will eventually fail, with the type of failure being **stuck at 0**, meaning that after failure it will report all future received bits to be 0, regardless of their true value. Once failed, the receiver stays failed, and should be replaced. Eventually the new receiver will also fail, and we will replace it; we continue this process indefinitely.

However, the problem is that we will not know whether a receiver has failed (unless we test it once in a while, which we are not including in this example). If the receiver reports a long string of 0s, we should suspect that the receiver has failed, but of course we cannot be sure that it has; it is still possible that the message being transmitted just happened to contain a long string of 0s.

Suppose we adopt the policy that, if we receive k consecutive 0s, we will replace the receiver with a new unit. Here k is a design parameter; what value should we choose for it? If we use a very small value, then we will incur great expense, due to the fact that we will be replacing receiver units at a very high rate. On the other hand, if we make k too large, then we will often wait too long to replace the receiver, and the resulting error rate in received bits will be sizable. Resolution of this tradeoff between expense and accuracy depends on the relative importance of the two. (There are also other possibilities, involving the addition of redundant bits for error detection, such as parity bits. For simplicity, we will not consider such refinements here. However, the analysis of more complex systems would be similar to the one below.)

A natural state space in this example would be

$$\{(i,j) : i = 0, 1, ..., i-1; j = 0, 1; i+j \neq 0\} \tag{8}$$

where i represents the number of consecutive 0s that we have received so far, and j represents the state of the receiver (0 for failed, 1 for nonfailed). Suppose the lifetime of the receiver, that is, the time to failure, is geometrically distributed with "success" probability ρ, i.e. the probability of failing on receipt of the i-th bit after the receiver is installed is $(1-\rho)^{i-1}\rho$, for i = 1,2,3,...

Then calculation of the transition matrix P is straightforward. For example, suppose the current state is (2,1), and that we are investigating the expense and accuracy corresponding to a policy having k = 5. What can happen upon receipt of the next bit? The next bit will have a true value of either 0 or 1, with probability 0.5 each. The receiver will change from working to failed status with probability ρ. Thus our next state could be:

- (3,1), if a 0 arrives, and the receiver does not fail;
- (0,1), if a 1 arrives, and the receiver does not fail; or
- (3,0), if the receiver fails

The probabilities of these three transitions out of state (2,1) are:

$$p_{(2,1),(3,1)} = 0.5(1-\rho) \tag{9}$$

$$p_{(2,1),(0,1)} = 0.5(1-\rho) \tag{10}$$

$$p_{(2,1),(3,0)} = \rho \tag{11}$$

Other entries of the matrix P can be computed similarly.

Formally specifying the matrix P using the 2-tuple notation would be very cumbersome. In this case, it would be much easier to map to a one-dimensional labeling. For example, if k = 5, the nine states (1,0),...,(4,0),(0,1),(1,1),...,(4,1) could be renamed states 1,2,...,9. Then we could form P under this labeling, and the transition probabilities above would appear as

$$p_{78} = 0.5(1-\rho) \tag{12}$$

$$p_{75} = 0.5(1-\rho) \tag{13}$$

$$p_{73} = \rho \tag{14}$$

After the π_i are determined, we can find the error rate ϵ, and the mean time (i.e., the mean number of bit receptions) between receiver replacements, μ. We can find both ϵ and μ in terms of the π_i, in the following manner.

The quantity ϵ is the proportion of the time that the true value of the received bit is 1 but the receiver is down, which is 0.5 times the proportion of the time spent in states of the form (i,0):

$$\epsilon = 0.5(\pi_1 + \pi_2 + \pi_3 + \pi_4) \tag{15}$$

Now to get μ in terms of the π_i note that since μ is the mean number of bits between receiver replacements, it is then the reciprocal of the proportion of bits that result in replacements. For example, if 5% of the received bits result in replacement of the receiver, then (speaking on an intuitive level) the "average" set of 20 bits will contain one bit which makes us replace the receiver, and there will thus be an average of 20 bits between replacements. A replacement will occur only from states of the form (4,j), and even then only under the condition that the next reported bit is a 0. In other words, there are three possible ways in which replacement can occur:

(a) We are in state (4,0). Here, since the receiver has failed, the next reported bit will definitely be a 0, regardless of that bit's true value. We will then have a total of k = 5 consecutive received 0s, and therefore will replace the receiver.

(b) We are in the state (4,1), and the next bit to arrive is a true 0. It then will be reported as a 0, our fifth consecutive 0, and we will replace the receiver, as in (a).

(c) We are in the state (4,1), and the next bit to arrive is a true 1, but the receiver fails at that time, resulting in the reported value being a 0. Again we have five consecutive reported 0s, so we replace the receiver.

Therefore,

$$\mu^-1 = \pi_4 + \pi_9(0.5 + 0.5\rho) \tag{16}$$

This kind of analysis could be used as the core of a cost-benefit tradeoff investigation to determine a good value of k. (Note that the π_i are functions of k, and that the above equations for the case k = 5 must be modified for other values of k.)