

Markov Chain Models and Basic Computations

Thursday, September 11, 2008
11:50 AM

Homework 1 is posted, due Monday, September 22.

Two more examples.

Inventory Model (Karlin and Taylor, Sec. 2.3)

Suppose that a store has a maximum capacity M for a given product.

Reordering policy: Any time the stock falls to a critical value s , then the store orders enough products to refill inventory up to its maximum value M .

Suppose that the reordering is done at the end of each business day, new stock arrives early the next morning.

Each day n , the customers demand a certain amount of products D_n , which are immediately fulfilled if the products are available.

Questions:

- Average inventory level in the long run?
- How much demand is unfulfilled per day?
- How many days were there unfulfilled demand?

These questions can be answered using Markov chain computation techniques. Let's for now just set up the model.

X_n = amount of stock in inventory at the end of day n

Will assume that the demand each day are independent, identically distributed random variables.

$$\{D_n\}_{n \geq 0}$$

Probability transition matrix a little unwieldy; let's write down the stochastic update rule:

$$X_{n+1} = \begin{cases} (M - D_{n+1})_+ & \text{if } X_n \leq s \\ (X_n - D_{n+1})_+ & \text{if } X_n > s \end{cases}$$

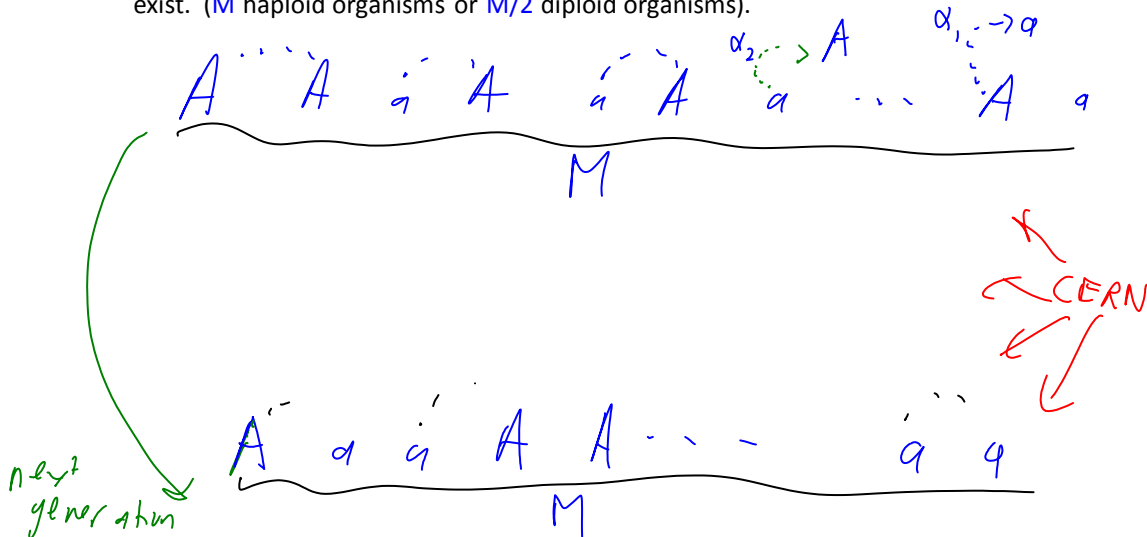
Remark: Suppose that instead the reordered products took a few days to arrive. Then clearly the stochastic update rule would have to keep track of variables from the past few days (how many products were ordered). This seems to lose the Markov property but here, as in many cases, it can be regained by simply formulating a Markov chain model with a larger state space, here it would something like the current inventory level and the number of products ordered on each of the last few days.

$$\begin{pmatrix} \sum_{n=1}^{\infty} \\ 0_n \\ \vdots \\ 0_{n-d} \end{pmatrix} \rightarrow \begin{pmatrix} \sum_{n=1}^{\infty} \\ 0_{n+1} \\ 0_n \\ \vdots \\ 0_{n-d+1} \end{pmatrix}$$

In fact, in some abstract way, any stochastic process can be expressed as a Markov process in some suitably large state space (Skorohod embedding theorem).

Wright-Fisher Model for Genetic Drift (Karlin and Taylor Sec. 2.2G)

Let's consider a single gene which has two alleles **A** and **a**. Suppose that we have a population in which an even number **M** copies of the gene exist. (**M** haploid organisms or **M/2** diploid organisms).



Drastic assumption: each generation has exactly **M** genes in the gene pool but this allows us to make a simple finite-state Markov chain model (which is actually used as a baseline model by geneticists).

How is the gene pool in the next generation determined? Choose randomly, *with replacement*, from the gene pool in the preceding generation, and each choice is independent of the others.

State space variable: X_n is the number of copies of the **A** gene in the **n**th generation.

Probability transition matrix:

$$P_{ij} = P(X_{n+1} = j | X_n = i) = \binom{M}{j} p_A^j p_a^{M-j}$$

Binomial distribution gives the probability of **j** "successes" in a sequence of **M** independent trials, each of which have the same given success probability (here p_A).

$$p_a = 1 - p_A$$

$$P_A = \frac{i}{M}$$

$$P_{ij} = \binom{M}{j} \left(\frac{i}{M}\right)^j \left(\frac{M-i}{M}\right)^{M-j}$$

Stochastic update rule? Think of writing a code that chooses each element of the new gene pool by generating M random numbers and encoding this as a math equation.

Suppose now we take the possibility of mutation into account. Let's suppose that the mutation happens in the parent before it is inherited by the offspring; in other words, the Markov chain state space variable is taking a census of gene pool immediately after reproduction.

$$P_A = P(\text{choose an } A \text{ that isn't mutated or choose an } a \text{ that did mutate})$$

$$= P(\text{choose an } A \text{ that isn't mutated}) + P(\text{choose an } a \text{ that did mutate})$$

(mutually exclusive events)

$$= P(\text{choose an } A) P(\text{the chosen } A \text{ isn't mutated})$$

$$+ P(\text{choose an } a) P(\text{the chosen } a \text{ does mutate})$$

(choice of gene is independent of the mutation of the gene)

$$P_A = \frac{i}{M} (1 - \alpha_1) + \frac{M-i}{M} \alpha_2$$

$$P_a = \frac{i}{M} \alpha_1 + \frac{M-i}{M} (1 - \alpha_2)$$

What about effects of selection (preference for one or the other allele)? Here we will focus on haploid organisms.

One easy way to model the positive selection for, say, the A allele is to add a weight to the choice for each A gene.

$$\begin{array}{ccccccc}
 A & A & a & A & a & a & A & A \\
 |s & & | & & | & & |s &
 \end{array}$$

$$P_A = \frac{i(1+s)}{i(1+s) + (M-i)} = \frac{\text{weight of } A\text{'s}}{\text{"weight" of whole pool}}$$

What is this model used for? Basic idea is to study the drift in the frequencies of the alleles in the population.

- Does one allele take over the whole gene pool?
- How long does it take when it does happen?

Basic computations with Markov chains

Once we have defined a Markov chain model, how do we use it to answer questions of interest from the real world?

Easy to simulate finite state, discrete-time Markov chains numerically

- o Either the stochastic update rule or the probability transition matrix tells you that to go from one epoch n to the next epoch $n+1$, you have to simulate one discrete random variable and combine it with the previous state X_n in some prescribed way. Of course also one needs to simulate the initial data from the initial probability distribution.

This is often called "Monte Carlo" simulation. So what's wrong with this? If one wants to study practical questions by simulating Markov chains, one needs to run the Markov chain many times to get a good sample of possible outcomes. Fundamental limitation though is that Monte Carlo simulations behave like a half-order accurate numerical method. That is, the accuracy of the results only increase with the Square root of the computational effort.

Rough idea: Monte Carlo simulations are good for rough answers, but usually not for precise answers.

Also, if we are concerned about what happens to the Markov chain at long times, then of course the numerical simulations would have to be run a long time.

The point of the theory in subsequent lectures is to provide equations for certain important statistics which are deterministic, and therefore are not subject to the slowly converging sampling error of Monte Carlo simulations.

Let's begin with a basic tool, which is how to compute the statistics of anything involving a Markov chain over a finite time horizon.

Any event that involves the state of the Markov chain over a fixed time horizon can be expressed in terms of disjoint unions of elementary outcomes of the form

$$\left\{ \begin{array}{l} X_{n_0} = \dot{j}_0, X_{n_1} = \dot{i}_1, \dots, X_{n_k} = \dot{j}_k \\ 0 \leq n_0 < n_1 < \dots < n_k \end{array} \right\}$$

$\dot{j}_0 \quad \dot{i}_1 \quad \dot{i}_2 \quad \dot{i}_3 \quad \dots \quad \dot{j}_k$

$$P(X_{n_0} = \dot{j}_0, X_{n_1} = \dot{i}_1, \dots, X_{n_{k-1}} = \dot{j}_{k-1})$$

$$= P(X_{n_k} = \dot{j}_k | X_{n_0} = \dot{j}_0, X_{n_1} = \dot{i}_1, \dots, X_{n_{k-1}} = \dot{j}_{k-1})$$

$$\begin{aligned}
 & \times P(X_{n_0} = j_0, X_{n_1} = j_1, \dots, X_{n_{k-1}} = j_{k-1}) \\
 P(A \text{ and } B) & \geq P(A|B) P(B) \\
 & = P(X_{n_k} = j_k | X_{n_{k-1}} = j_{k-1}) \quad (\text{Markov property}) \\
 & \quad \times P(X_{n_0} = j_0, X_{n_1} = j_1, \dots, X_{n_{k-1}} = j_{k-1})
 \end{aligned}$$

By induction,

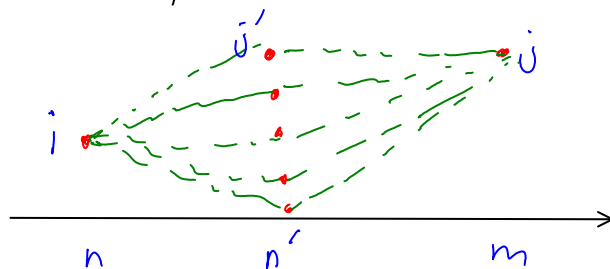
$$\begin{aligned}
 & P(X_{n_0} = j_0, X_{n_1} = j_1, \dots, X_{n_k} = j_k) \\
 & = P(X_{n_k} = j_k | X_{n_{k-1}} = j_{k-1}) \\
 \star & \quad \times P(X_{n_{k-1}} = j_{k-1} | X_{n_{k-2}} = j_{k-2}) \\
 & \quad \vdots \\
 & \quad \times P(X_{n_1} = j_1 | X_{n_0} = j_0) \\
 & \quad \times P(X_{n_0} = j_0)
 \end{aligned}$$

So this has reduced the computation of any "finite-dimensional distribution" to the computation of a single point observation (at n_0) and conditional probabilities for the next observation given the previous observation (but not necessarily at the immediately next epoch).

To compute the conditional probabilities, we use the

Chapman-Kolmogorov equation

$$\begin{aligned}
 & P(X_m = j | X_n = i) \\
 & = \sum_{j' \in S} P(X_m = j | X_{n'} = j') P(X_{n'} = j' | X_n = i) \\
 & \text{for any } n < n' < m
 \end{aligned}$$



We'll prove this next time, but notice that if we apply it with $n'=m-1$ then

$$P(X_{m+1} = j | X_m = i) = \sum_{j'} P(X_{m+1} = j | X_m = j') P(X_m = j' | X_{m-1} = i)$$

then

$$P(X_m = j \mid X_n = i) = \sum_{j' \in S} P(X_m = j \mid X_{m-1} = j') P(X_{m-1} = j' \mid X_n = i)$$

By induction,

$$P(X_m = j \mid X_n = i) = P_{j \hat{=} j}^{m-n} P_{i \hat{=} j}^{m-n}$$

for $m-n \geq 0$