Homework 1 is now posted, due Monday, September 22 at 12 PM.

Let's see why the stochastic update rule and the classical characterization of Markov chains by probability transition matrix are equivalent.

Let's first take the stochastic update rule as specified and show that it implies the Markov property. (see Resnick Section 2.1)

\[
X_{n+1} = f_n (X_n, Z_n)
\]

\[
p(X_{n+1} = j \mid X_0 = i_0, X_1 = i_1, \ldots, X_n = i_n)
\]

\[
= p(f_n(X_n, Z_n) = j \mid X_0 = i_0, X_1 = i_1, \ldots, X_n = i_n)
\]

\[
= p(f_n(i_0, Z_n) = j \mid X_0 = i_0, X_1 = i_1, \ldots, X_n = i_n) \text{ (provided that } g \text{ is continuous)}
\]

\[
= p(f_n(i_0, Z_n) = j \mid X_n = i_n) \text{ (justification: } p(A \mid B, B_2, \ldots, B_n, C_2, C_3, \ldots, C_m) = p(A \mid B, B_2, B_n) \text{ whenever } C_2, \ldots, C_m \text{ is independent })
\]

\[
= p(f_n(X_n, Z_n) = j \mid X_n = i_n)
\]

\[
P(X_{n+1} = j \mid X_n = i) \text{ Markov property}
\]

Now let's show that given a classically defined Markov chain in terms of
a probability transition matrix, we can construct a stochastic update rule.

Given \( p_{ij} = P(X_{n+1} = j \mid X_n = i) \)

To construct the stochastic update rule we recognize that for each given value of the previous state \( i \), we essentially need to simulate a discrete random variable with prescribed probability distribution to determine the next state \( j \).

\[
P = \begin{pmatrix} p_{i1} & p_{i2} & \cdots & p_{in} \end{pmatrix}
\]

How does one simulate a random variable \( Y \) with finite state space \((y_1, y_2, \ldots, y_M)\) with probabilities

\[
P \left( \sum Y = y_j \right) = p_{j1}, \sum p_{ij} = 1
\]

The basic "pseudorandom" variable which a computer simulates is a uniformly distributed random variable on the unit interval

\[
U \sim \mathcal{U}(0,1)
\]

To get specific recommendations for hi-fi random number generators, see recent review articles and website of P. Lecuyer (Montreal).

In mathematical notation, this random variable simulation algorithm can be written:

\[
\sum = \begin{cases} y_1, & 0 < U < p_1 \\
y_2, & p_1 \leq U < p_1 + p_2 \\
\vdots & \ddots \\
y_M, & p_{M-1} \leq U < 1
\end{cases}
\]
In mathematical notation, this random variable simulation algorithm can be written:

\[
Y = \begin{cases} \gamma_1, & 0 \leq U < p_1 \\ \gamma_2, & p_1 \leq U < p_1 + p_2 \\ \vdots \\ \gamma_M, & p_1 + \ldots + p_{M-1} \leq U \leq 1 \\ \end{cases}
\]

where \( \gamma_j \) are independent identically distributed random variables uniformly distributed on the unit interval:

\[
\mathbb{I} \{ A \} = \begin{cases} 1 & \text{if } A \text{ is true} \\ 0 & \text{if } A \text{ is false} \end{cases}
\]

More precisely, recall that all random variables and events can be though of in terms of underlying probability space \( \Omega \), with elementary outcomes \( \omega \in \Omega \).

\[
\mathbb{I} \{ A \} (\omega) = \begin{cases} 1 & \omega \in A \\ 0 & \omega \in A^c \end{cases}
\]

So here is how one can write a stochastic update rule corresponding to an arbitrary probability transition matrix:

Let \( \{Z_n\} \) be a sequence of independent identically distributed random variables uniformly distributed on the unit interval:

\[
Z_n = U_n^i \sim U(0,1)
\]

Then

\[
X_{n+1} = f \left( X_n, U_n \right)
\]

will be a stochastic update rule consistent with the probability transition matrix. We are just using the rule for simulating a random variable on a finite state space from a uniform random number, using bins determined by the \( i \)th row of the probability transition matrix.

We've shown that the probability transition matrix and the stochastic update rule are equivalent means for defining a finite-state, discrete-time
Markov chain. In practice, as we'll see, sometimes one is more convenient than the other.

Specific Examples (Karlin and Taylor, Sections 2.1-2.3)

1) Two-state on/off system
   State 1: off/unbound/free
   State 2: on/bound/busy

   When in the off state, probability $p$ to switch to the on state in the next epoch.
   When in the on state, probability $q$ to switch to the off state in the next epoch.

   Probability transition matrix:
   \[
   \begin{pmatrix}
   1-p & p \\
   q & 1-q \\
   \end{pmatrix}
   \]

   And as always need to specify the initial probability distribution
   \[
   \begin{pmatrix}
   \phi_0 \\
   \phi_1 \\
   \end{pmatrix}
   \]

   Stochastic update rule: here annoying and pointless.

2) Random walk on finite graph (Lawler Ch. 1)

   A random walker proceeds among the 8 nodes. At each epoch, he has a probability 1/2 to move, and when he moves, he moves to a neighboring node along an arc, with probability proportional to the number of arcs connecting the nodes.
Stochastic update rule...a real pain here.

One thing to keep in mind with Markov chain models is that sometimes different choices of epochs are useful. Here, perhaps we chose epochs as equally spaced moments in time.

In the present case, it might be useful to consider instead epochs to be defined by moments at which a random walker visits a new state. This would change the probability transition matrix (and the Markov chain).
3) Queueing models with maximum capacity $M$

(Karlin and Taylor Section 2.2.C)

Server can handle one request/demand at a time.

State space for the queue will be the number of requests currently waiting or being served.

$$S = \{0, 1, \ldots, M\}$$

Time domain: at least two natural choices for epochs:

- Equally spaced time intervals, hopefully short enough that at most one arrival or departure occurs per time interval
- Event-based epoch: various possibilities (wait til anything happens, etc.), but we'll focus on marking epochs by when a service is completed.

Using the first strategy with epochs, equally spaced in time. Between each epoch, the following can happen:

- Arrival of a new request with probability $p$
- Completion of a service with probability $q$
- Nothing happens (probability $1-p-q$)
Stochastic update rule

Idea:

\[ X_{n+1} = X_n + Z_n \]

\[ Z_n = \begin{cases} 
-1 & \text{w/ prob } p \\
\phantom{-1} 0 & \text{w/ prob } 1-p-q \\
\phantom{0} 1 & \text{w/ prob } q 
\end{cases} \]

But we have to patch this up for the states 0, M.

\[ X_{n+1} = \min \left( \left( X_n - Z_n \right)_+, M \right) \]

where

\[ (y)_+ = \max(y, 0) \]

Consider now the case where we take epochs to be defined by the times at which services are completed.

○ Probability distribution for the number of arrivals during a service period.

\[ P_j = P\left( j \text{ requests arrive during a service period} \right) \]

\[ \sum_{j=0}^{\infty} P_j = 1 \]

\[ P_0 \quad P_1 \quad P_2 \quad \cdots \quad P_M \]

\[ \begin{pmatrix} P_0 & P_1 & P_2 & \cdots & P_{M-1} & \cdots & P_M \\
0 & P_0 & P_1 & \cdots & P_{M-1} & \cdots & P_M \\
1 & 0 & P_0 & P_1 & \cdots & P_{M-1} & \cdots & P_M \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \]