Simulation of Discrete-Time Markov Chains

Peter J. Haas

CS590M: Simulation
Spring Semester 2020

DTMC Definition

Simplest model for dynamic stochastic system

- X_n = system state after nth transition
- $(X_n : n \geq 0)$ satisfies the Markov property

$$P\{X_{n+1} = x \mid X_n = x_n, X_{n-1} = x_{n-1}, \ldots, X_0 = x_0\}$$
$$= P\{X_{n+1} = x \mid X_n = x_n\}$$

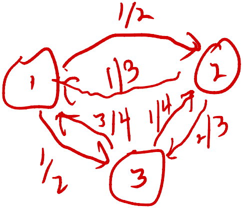
Time-homogeneous DTMC with state space S defined via

1. Transition matrix $P = (P(x, y) : x, y \in S)$, with
   $P(x, y) = P\{X_{n+1} = y \mid X_n = x\}$
2. Initial distribution $\mu = (\mu(x) : x \in S)$, with
   $\mu(x) = P\{X_0 = x\}$

Example: Markovian Jumping Frog

$X_n =$ **lily pad occupied by frog after nth jump**

- Frog starts in states 1 and 2 with equal probability



$$P = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{bmatrix} 0 & 1/2 & 1/2 \\ 1/3 & 0 & 2/3 \\ 3/4 & 1/4 & 0 \end{bmatrix} \end{array} \quad \text{and} \quad \mu = \begin{array}{c} 1 \\ 2 \\ 3 \end{array} \begin{bmatrix} 1/2 \\ 1/2 \\ 0 \end{bmatrix}$$

Computing Probabilities and Expectations

Example: $\theta = P\{\text{frog on pad 2 after } k\text{th jump}\}$

- Write as $\theta = E[f(X_k)]$, where $f(x) = \begin{cases} 1 & \text{if } x = 2 \\ 0 & \text{otherwise} \end{cases}$ $= P_k(x=2)$
- Sometimes write indicator function $f(x)$ as $I(x = 2)$
- Q: Why is this correct?

Numerical solution for arbitrary function f

- Q: What is probability distribution after first jump?
- Let $v_n(i) = P\{\text{frog on pad } i \text{ after } n\text{th jump}\}$ $v_n = \begin{bmatrix} v_n(1) \\ v_n(2) \\ v_n(3) \end{bmatrix}$
- Set $v_0 = \mu$ and $v_{m+1}^\top = v_m^\top P$ for $m \geq 0$
  I.e., $v_{m+1}(j) = \sum_{i=1}^{3} v_m(i)\, P(i,j)$
- Then $E[f(X_k)] = v_k^\top f$, where $f = [f(1), f(2), f(3)]^\top$
- Ex: For θ as above, take $f = (0, 1, 0)$ so that $v_k^\top f = v(2)$

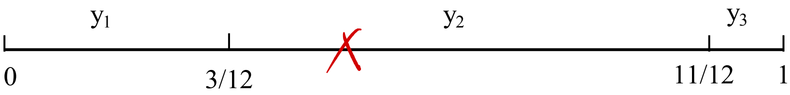
Simulation of DTMCs

Why simulate? Huge state space, complex estimand

Naive method for generating a discrete random variable

- Goal: Generate Y having pmf $p_i = P\{Y = y_i\}$ and cdf $c_i = P\{Y \le y_i\}$ for $1 \le i \le m$
- Example:

i	p_i	c_i
1	3/12	3/12
2	8/12	11/12
3	1/12	12/12



Ex: If $U = 0.27$, then return $Y = y_2$

- Q: How can we speed up this algorithm? sort in decreasing order of length $O(m)$

m = # of possible values

binary search $O(\log m)$

Simulation of DTMCs, Continued

Generating a sample path $X_0, X_1, \dots$

1. Generate X_0 from μ and set $m = 0$
2. Generate Y according to $P(X_m, \cdot)$ and set $X_{m+1} = Y$
3. Set $m \leftarrow m + 1$ and go to 2.

Estimating $\theta = E[f(X_k)]$

1. Generate $X_0, X_1, \dots, X_k$ and set $Z = f(X_k)$
2. Repeat n times to generate $Z_1, Z_2, \dots, Z_n$ (i.i.d.)
3. Compute point estimate $\theta_n = (1/n) \sum_{i=1}^{n} Z_i$

Can generalize to estimate $\theta = E[f(W)]$, where
$W = f(X_0, X_1, \dots, X_k)$

DTMCs: Recursive Definition

Proposition

- Let $U_1, U_2, \ldots$ be a sequence of i.i.d. random variables and X_0 a given random variable
- $(X_n : n \geq 0)$ is a time homogeneous DTMC $\Leftrightarrow$ $X_{n+1} = f(X_n, U_{n+1})$ for $n \geq 0$ and some function f

In $\Rightarrow$ direction, $U_1, U_2, \ldots$ can be taken as uniform

Can use to prove that a given process $(X_n : n \geq 0)$ is a DTMC

Q: What if $U_0, U_1, \ldots$ are independent but not identical?

you get a non-homogeneous DTMC

Q: Practical advantages of recursive definition?

- easier to code
- can handle large state spaces

Example: (s, S) Inventory System

The model

- X_n = inventory level at the end of period n
- D_n = demand in period n
- If (s, S) policy is followed then

$X_{n+1} = f(X_n, D_{n+1})$

$$X_{n+1} = \begin{cases} X_n - D_{n+1} & \text{if } X_n - D_{n+1} \geq s; \\ S & \text{if } X_n - D_{n+1} < s \end{cases}$$

Claim: If $(D_n : n \geq 1)$ is i.i.d. then $(X_n : n \geq 0)$ is a DTMC with state space $\{s, s+1, \ldots, S\}$ True because of recursive representation

Q: Critique of model—what might be missing?

non-iid demand item returns
delivery delays (backlogs)

Digression: Stationary Distribution of a DTMC

Definition

- *Informal*: π is a stationary distribution of the DTMC if $X_n \overset{D}{\sim} \pi$ implies $X_{n+1} \overset{D}{\sim} \pi$
- *Formal:*

$$\pi(j) = P(X_{n+1} = j) = \sum_i P(X_{n+1} = j \mid X_n = i)\, P(X_n = i)$$
$$= \sum_i P(i,j)\pi(i) \qquad \text{or } \pi^\top = \pi^\top P$$

- So if $X_0 \overset{D}{\sim} \pi$, then $X_n \overset{D}{\sim} \pi$ for $n \geq 1$

Under appropriate conditions, $\lim_{n\to\infty} P(X_n = i) = \pi(i)$

- Also written as $X_n \Rightarrow X$, where $X \overset{D}{\sim} \pi$

How to estimate $\theta = \sum_i f(i)\pi(i) = E[f(X)]$ **(where** $X \overset{D}{\sim} \pi$**)?**

General State Space Markov Chains: GSSMCs

Problem: With continuous state space,
$P\{X_{n+1} = x' \mid X_n = x\} = 0$!

- Solution: Use transition kernel
  $P(x, A) = P\{X_{n+1} \in A \mid X_n = x\}$
- In practice: Use recursive definition

Random walk:

$X_{n+1} = f(X_n, Y_{n+1})$
where $f(x,y) = x + y$

Example: Continuous (s, S) inventory system

Example: Random walk on the real line

- Let $Y_1, Y_2, \ldots$ be an i.i.d. sequence of continuous, real-valued random variables
- Set $X_0 = 0$ and $X_{n+1} = X_n + Y_{n+1}$ for $n \geq 0$
- Then $(X_n : n \geq 0)$ is a GSSMC with state space $\Re$
  (recursive representation given above)

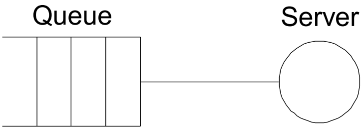
GSSMC Example: Waiting times in GI/G/1 Queue

one server

GI : general distribution for i.i.d. interarrival times
G : general service-time distribution

The GI/G/1 Queue

- Service center: single server, infinite-capacity waiting room
- Jobs arrive one at a time
- First-come, first served (FCFS) service discipline
- Successive interarrival times are i.i.d.
- Successive service times are i.i.d.



GI/G/1 Waiting Times, Continued

Notation

- W_n = the waiting time of the nth customer (excl. of service)
- A_n/D_n = arrival/departure time of the nth customer
- V_n = processing time of the nth customer

Recursion (Lindley Equation)

- $D_n = A_n + W_n + V_n$
- Thus

$$\begin{aligned} W_{n+1} &= [D_n - A_{n+1}]^+ = [A_n + W_n + V_n - A_{n+1}]^+ \\ &= [W_n + V_n - I_{n+1}]^+ \end{aligned}$$

  where $I_{n+1} = A_{n+1} - A_n$ is $(n+1)$st interarrival time and $[x]^+ = \max(x, 0)$

- Thus $(W_n : n \geq 0)$ is a GSSMC
- To simulate: generate the V_n's and I_n's and apply recursion