Module 4, Lecture 2: Markov Decision Processes

M. Vidyasagar

Distinguished Professor, IIT Hyderabad
Email: m.vidyasagar@iith.ac.in
Website: www.iith.ac.in/∼m_vidyasagar/

Outline

Outline

Definition of a Markov Process

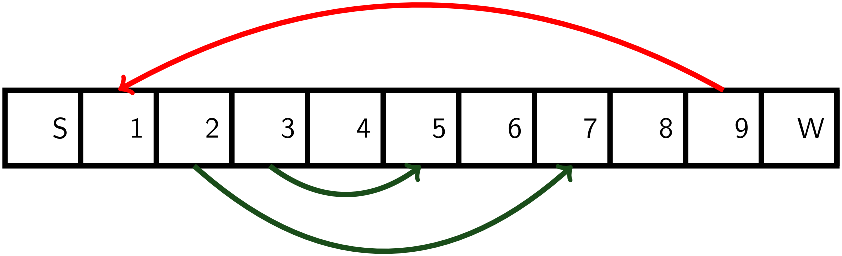
A **Markov process** (for present purposes) is a sequence of random variables $\{X_t\}_{t \geq 0}$, where each X_t takes values in a *finite* set $\mathbb{N}$ called the **state space**.

The key property is this: The conditional probability *does not depend on the past history*. Thus

$$\Pr\{X_{t+1}|X_0^t\} = \Pr\{X_{t+1}|X_t\},$$

where $X_0^t = (X_0, \ldots, X_t)$.

Example: Toy Snakes and Ladders Game



Rules of the Game

- Initial state is S.
- A four-sided, fair die is thrown at each stage.
- Player must land exactly on W to win.
- If implementing a move causes crossing of W, then the move is not implemented.

Why is This a Markov Process?

Because once a player is in state i, what happens next does not depend on how the player got to state i.

State Transition Matrix

Suppose there are n states in $\mathbb{N}$, call them $\{1, \ldots, n\}$. The matrix $A \in [0,1]^{n \times n}$ defined by

$$a_{ij} = \Pr\{X_{t+1} = j | X_t = i\}$$

is called the **state transition matrix**.

Note:

$$\sum_{j=1}^{n} a_{ij} = 1, \ \forall i.$$

So $\lambda = 1$ is an eigenvalue of A with *column eigenvector* $v = \mathbf{1}_n$. The corresponding *row eigenvector* is called a **stationary probability distribution**.

Transition Probabilities of Snakes and Ladders Game

- There are eleven possible states in all: S, 1, . . . , 9 , W.
- However, 2, 3, 9 can be omitted, leaving eight states, namely S, 1, 4, 5, 6, 7, 8, W.
- At each step, there are at most four possible outcomes.
- For example, from the state S, the four outcomes are 1, 7, 5, 4.
- From state 6, the four outcomes are 7, 8, 1, and W.
- From state 7, the four outcomes are 8, 1, W, 7.
- From state 8, there are only three outcomes: 1, W with probability $1/4$ each, and 8 with probability $1/2$.

State Transition Matrix of Snakes and Ladders Game

$$\Pr\{X(t+1) = j | X(t) = i\}.$$

	S	1	4	5	6	7	8	W
S	0	0.25	0.25	0.25	0	0.25	0	0
1	0	0	0.25	0.50	0	0.25	0	0
4	0	0	0	0.25	0.25	0.25	0.25	0
5	0	0.25	0	0	0.25	0.25	0.25	0
6	0	0.25	0	0	0	0.25	0.25	0.25
7	0	0.25	0	0	0	0.25	0.25	0.25
8	0	0.25	0	0	0	0	0.50	0.25
W	0	0	0	0	0	0	0	1

Transient, Recurring and Absorbing States

From the example, it is obvious that the state X_t can assume values other than W only for a while before eventually reaching W. So states S, $1, \ldots, 8$ are **transient** states, while the state W is a **recurrent** state.

W is also an **absorbing** state, because once the process hits the state W, it stays there.

Computing the Hitting Time

How long does the "average" game last from any current state? How long does it take "on average" to hit the absorbing state W?

Form $A_{7 \times 7}$ to consist of the first 7×7 submatrix of A, and define $v = \mathbf{1}_y$. Then the vector of stopping times is given by

$$\boldsymbol{\tau} = (I - A_{7 \times 7})^{-1} v.$$

The formula can be modified for any other Markov process.

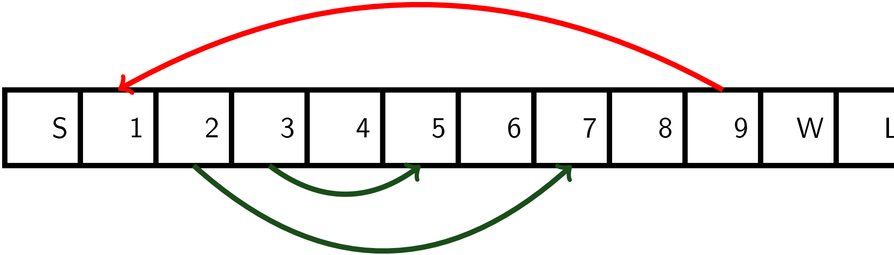
Computing the Hitting Times for Snakes and Ladders Game

Applying the above formula gives

$$\boldsymbol{\tau} = \begin{bmatrix} 8.3636 \\ 8.2182 \\ 7.4909 \\ 7.6364 \\ 6.1091 \\ 6.1091 \\ 6.1091 \end{bmatrix}.$$

Toy Snakes and Ladders Game (Modified)

Now we modify by adding a Lose (L) state in addition to the Win (W) state.



Modified State Transition Matrix

	S	1	4	5	6	7	8	W	L
S	0	0.25	0.25	0.25	0	0.25	0	0	0
1	0	0	0.25	0.50	0	0.25	0	0	0
4	0	0	0	0.25	0.25	0.25	0.25	0	0
5	0	0.25	0	0	0.25	0.25	0.25	0	0
6	0	0.25	0	0	0	0.25	0.25	0.25	0
7	0	0.25	0	0	0	0	0.25	0.25	0.25
8	0	0.25	0	0	0	0	0.25	0.25	0.25
W	0	0	0	0	0	0	0	1	0
L	0	0	0	0	0	0	0	0	1

Hitting Probabilities

What is the probability of hitting W or L from the current state?

Define $\mathbf{y}_W, \mathbf{y}_L$ to be the eighth and ninth columns (first seven rows only) of A. Define

$$\mathbf{h}_W = (I - A_{7\times 7})^{-1}\mathbf{y}_W, \mathbf{h}_L = (I - A_{7\times 7})^{-1}\mathbf{y}_L.$$

Hitting Probabilities for the Modified S&L Game

$$[\mathbf{h}_W \ \ \mathbf{h}_L] = \begin{bmatrix} 0.5433 & 0.4567 \\ 0.5457 & 0.4543 \\ 0.5574 & 0.4426 \\ 0.5550 & 0.4450 \\ 0.6440 & 0.3560 \\ 0.5152 & 0.4848 \\ 0.5152 & 0.4848 \end{bmatrix}.$$

Outline

Estimating the State Transition Matrix From Observations

Suppose we observe a sequence of states $\{x_0, x_1, \ldots, X_T\}$.

The *maximum likelihood estimate* (that is, the estimate that best fits the data) of the state transition matrix A is given by

$$\hat{A}_{ij} = \frac{\#\{(x_t, x_{t+1}) = ij\}}{\#\{x_t = i\}}.$$

How many times is $x_t = i$? Out of those, how many times is $x_{t+1} = j$? The ratio is the estimate $\hat{A}_{ij}$.

Outline

IIT Hyderabad

Markov Decision Processes

In a Markov decision process (MDP), in addition to the state space $\mathbb{N}$, there is also a control (or action) space U.

Instead of just one fixed state transition matrix A, there is one state transition matrix A^u for each $u \in U$.

At time t, if the current state is $x(t)$, and we decide on an action $u \in U$, this causes two things:

- It results in an immediate "reward" $R(x, u)$.
- The next state x_{t+1} is determined according to the matrix A^u. It is of course random.

Objective in a Markov Decision Process

There is a specified "discount factor" $\gamma < 1$. The objective is to choose an "optimal control policy" $\pi : \mathbb{N} \to U$ so as to maximize the **expected discounted reward**

$$E \left(\sum_{t=0}^{\infty} \gamma^t R(x(t), u(t)) \right) .$$

Bellman's Optimality Equation

Define $V^*(x)$ to be the **Highest possible reward** that can be achieved by any "policy" (method of selecting actions for each state), when starting at state x. So $V^* : \mathbb{N} \to \mathbb{R}$.

Then V^* satisfies

$$V^*(x) = \max_{u \in U} \left(R(x,u) + \gamma \sum_{y \in \mathbb{N}} A^u_{x,y} V^*(y) \right).$$

Methods for solving Bellman's equation exist but won't be discussed.

Outline

Reinforcement Learning

Suppose we have an *unknown* Markov Decision Process. So we know the state space $\mathbb{N}$, the action set U, and the discount factor γ. We can also *observe* what happens when we choose any particular action.

How can we choose an optimal policy?

Rough Answer: By watching the state transitions, we can estimate the state transition matrices A^u for each $u \in U$. (Obviously we would have to watch for a long, long time!)

We can also form an estimate of the reward function $R(x, u)$.

Based on our current estimate, we solve Bellman's equation with estimated quantities instead of the true quantities.

IIT Hyderabad