

Lecture 20: Markov Decision Processes

Agenda:

- Markov Decision Process (MDP)
- Value Functions

Reading:

- No textbook pages

Lecture 20: Markov Decision Processes

An MDP is defined by:

- S , a set of states,
- A , a set of actions,
- $P(s, a, s')$, the probability of reaching s' after taking action a from state s (transition function).

$$\forall a \in A, s \in S, \quad \sum_{s' \in S} P(s, a, s') = 1$$

- $R(s)$, the reward reaching state s .

An Example: Football

- $S = \{5, 10, 15, 0, \text{END}\}$: yards to go for the first down.
- $A = \{\text{RunInside}, \text{RunOutside}, \text{Pass}\}$: plays that can be called.
- Each play has a probability distribution over its outcomes.

Play	-5	0	+5	+10	+15
RI	0.0	0.3	0.5	0.2	0.0
RO	0.1	0.3	0.2	0.3	0.1
P	0.2	0.3	0.1	0.1	0.3

E.g., $P(10, P, 15) = 0.2$ and $P(10, P, 0) = 0.4$

- $\forall a \quad P(0, a, \text{END}) = 1$
- $\forall a \quad P(\text{END}, a, \text{END}) = 1$
- $R(0) = 1$, otherwise $R(s) = 0$.
 - The END state prevents us from getting the reward multiple times.

Lecture 20: Markov Decision Processes

Other Examples:

- Elevator Control
 - States are $\{\text{elevator position}\}^k \times \{\text{button states}\}$.
 - Actions: $\{\text{up, down}\}^k$
 - Transition function determined by action and likely stochastic changes to the button states.
 - Reward of -1 for every floor with waiting passengers.
 - Crites and Barto (1996): High performance down-peak elevator controller.
- Backgammon (against a fixed opponent)
 - States are $\{\text{checker positions}\} \times \{\text{dice roll}\}$
 - Transition function takes player's turn, opponent's turn, and rolls the dice.
 - Reward of 1 for winning, -1 for losing.
 - Tesauro (1992): World's best backgammon program.
- Inventory Management
- Walking Robots

Lecture 20: Markov Decision Processes

Computing a Value Function:

- Given an MDP, a state s , and a horizon time T , find the maximum expected sum of rewards that could be received over T steps.
- Call this $V(s, T)$.
- Optimal substructure: If we know $V(s', T - 1)$ for all s' , we can compute $V(s, T)$. How?
 - Suppose we know the action to take that would maximize our sum of rewards.

$$V(s, T) = R(s) + \sum_{s' \in S} (P(s, a, s') V(s', T - 1))$$

- If we don't know the action?

$$V(s, T) = \max_{a \in A} \left(R(s) + \sum_{s' \in S} (P(s, a, s') V(s', T - 1)) \right)$$

- Lastly, $V(s, 0) = R(s)$.

Lecture 20: Markov Decision Processes

Computing a Value Function:

- Assume $S = \{1, \dots, n\}$ and $A = \{1, \dots, m\}$,
- Pseudocode

MDP-Value(n, m, P, R, s, T)

```

for  $i \leftarrow 1$  to  $n$  do
   $V[i, 0] \leftarrow R[i]$ 
  for  $t \leftarrow 1$  to  $T$  do
    for  $i \leftarrow 1$  to  $n$  do
       $V[i, t] \leftarrow -\infty$ 
      for  $j \leftarrow 1$  to  $m$  do
         $v \leftarrow R[i] + \sum_{k=1}^n (P[i, j, k]V[j, t-1])$ 
        if  $v > V[i, t]$  then
           $V[i, t] \leftarrow v$ 
      return  $V[s, T]$ 

```

- Running Time: $\Theta(Tn^2m)$
- Example: Football ($T = 2, s = 10$)

State	Time		
	0	1	2
END	0	0	0
0	1	1	1
5	0	0.7	0.91
10	0	0.4	0.69
15	0	0.3	0.56

– Consider $V[10, 2]$.

$$\text{RI: } 0.2V[0, 1] + 0.5V[5, 1] + 0.3V[10, 1] = 0.67$$

$$\text{RO: } 0.4V[0, 1] + 0.2V[5, 1] + 0.3V[10, 1] + 0.1V[15, 1] = 0.69$$

$$\text{P: } 0.4V[0, 1] + 0.1V[5, 1] + 0.3V[10, 1] + 0.2V[15, 1] = 0.65$$

Lecture 20: Markov Decision Processes

Using a Value Function:

- What action should we take at state s with t time left?
- Compute from the value function.

$$\pi(s, t) = \operatorname{argmax}_{a \in A} \left(R(s) + \sum_{s'} P(s, a, s') V(s') \right)$$

- Running Time: $\Theta(nm)$
- We could also store this while we compute it.

MDP-Value(n, m, P, R, s, T)

```
for  $i \leftarrow 1$  to  $n$  do
   $V[i, 0] \leftarrow R[i]$ 
  for  $t \leftarrow 1$  to  $T$  do
    for  $i \leftarrow 1$  to  $n$  do
       $V[i, t] \leftarrow -\infty$ 
      for  $j \leftarrow 1$  to  $m$  do
         $v \leftarrow R[i] + \sum_{k=1}^n (P[i, j, k] V[j, t-1])$ 
        if  $v > V[i, t]$  then
           $V[i, t] \leftarrow v$ 
           $\pi[i, t] \leftarrow j$ 
return  $V[s, T]$ 
```

Lecture 20: Markov Decision Processes

Midterm Review:

- Heaps
- Heapsort
- Quicksort
- Sorting Lower-Bound
- Dynamic Programming
 - Principles
 - Knapsack Problem
 - Matrix-Chain Multiplication
 - Common Subsequences

Lecture 20: Markov Decision Processes

Have you understood the lecture contents?

well	ok	not-at-all	topic
☐	☐	☐	Markov decision processes
☐	☐	☐	value functions
☐	☐	☐	finding a policy