Reinforcement Learning - Lecture 3

Finite Horizon MDPs and Backward Induction

ENS M2

October 1, 2025

Contents

1 Finite Horizon Markov Decision Processes

Definition

Finite Horizon MDP

- The horizon T is a finite positive integer
- The state space $\mathcal{S} = (\mathcal{S}_t)_{t=1}^{T}$ is finite with cardinality S
- The action space $\mathcal{A} = (\mathcal{A}_t)_{t=1}^{T-1}$ is finite with cardinality A

Dynamical system defined by:

- An initial state $S_1 \in \mathcal{S}_1$
- $\forall t \geq 1, S_{t+1} = \phi(S_t, A_t, u_t)$ where s_t is the current state, A_t is the action taken at time t, and u_t is a random variable with known distribution (extended randomization)
- The reward is defined by $R_t = r(S_t, A_t, S_{t+1})$
- Cumulated reward: $W_T = \sum_{t=1}^{T-1} R_t$

MDP view: kernel $P_t(s, a, s') = \mathbb{P}(S_{t+1} = s' | S_t = s, A_t = a)$

2 Policy Definitions

Definition

Types of Policies

- **Decision rule**: π_t: A mapping from $\mathcal{S}_t$ to $\mathcal{A}_t$
- **Policy**: $\pi = (\pi_t)_{t=1}^{T}$
- **Randomized decision rule**: π_t: A mapping from $\mathcal{S}_t$ to $\mathcal{M}_1(\mathcal{A}_t)$ where $\mathcal{M}_1(\mathcal{A}_t)$ is the set of probability distributions on $\mathcal{A}_t$
- **Randomized policy**: $\pi = (\pi_t)_{t=1}^{T}$
- **Strategy** ψ: for each time t, maps a history $h_t \in (\mathcal{S}_1 \times \mathcal{A}_1) \times \ldots \times (\mathcal{S}_t)$ to a decision rule π_t

$$A_t = \psi(S_1, A_1, \ldots, S_t)$$

Tip

Markov Property
For a fixed policy π:

$$S_{t+1} = \phi(S_t, \pi_t(S_t), u_t)$$

The law of S_{t+1} is given by S_t (and $A_t = \pi_t(S_t)$)

$\Rightarrow (S_t)$ is a **Markov Chain**
This means that the law of S_{t+1} given $(S_1, \dots, S_t)$ is the same as the law of S_{t+1} given S_t.

3 Classic Examples

3.1 Riverswim

Example

Riverswim
We consider a riverswim with $N+1$ boxes indexed from 0 to N.
$r(s, a, s') = 1$ if $s' = N$ and 0 otherwise.
Action 1:

$$k(s, 1, s') = \mathbb{P}(S_{t+1} = s' | A_t = 1, S_t = s) = \begin{cases} p & \text{if } s' = s+1 \\ 1-p & \text{if } s' = s-1 \\ 0 & \text{otherwise} \end{cases}$$

Action 2:

$$k(s, 2, s') = \mathbb{P}(S_{t+1} = s' | A_t = 2, S_t = s) = \begin{cases} q & \text{if } s' = s+1 \\ 1-q-r & \text{if } s' = s \\ r & \text{if } s' = s-1 \end{cases}$$

Remember we need to clip the values of s to stay in the interval $[0, N]$.

3.2 Secretary Problem

Example

Secretary Problem
Let $U_1, \dots, U_T$ be i.i.d. random variables with uniform distribution on $[0, 1]$.
For $t = 1, \dots, T$:

- You observe U_t
- You decide to keep U_t or reject it forever

If you did not stop before, you keep U_T.
Goal: Maximize the probability to keep the maximum of $U_1, \dots, U_T$.
Formalization as MDP:

- State space: $\mathcal{S}_t = [0, 1] \times \mathbb{N}$ where $\mathbb{N}$ means 0 if we did not stop and k if we stopped at time k
- Action space: $\mathcal{A}_t = \{0, 1\}$ where 0 means stop and 1 means continue
- $r(s_t, a_t, s_{t+1}) = 0$ except $r(s_{T-1}, a_{T-1}, s_T) = 1$ if denoting $s_T = ((u_0, \dots, u_T), k)$ we have $u_k = \max(u_1, \dots, u_T)$
- $k(s, a, s') \to s' = (s, u)$ with u uniform on $[0, 1]$

- $\phi(s, a, u) = (s, u)$

4 Policy Evaluation

4.1 Value Function Definition

A policy π is fixed.

We take an initial state $s \in \mathcal{S}_1$.

We consider the random variable:

$$V^\pi(s) = \mathbb{E}^\pi \left[\sum_{t=1}^{T} R_t \middle| S_1 = s \right]$$

where $\mathbb{E}^\pi$ means that the expectation is taken with respect to the law of the process when we apply policy π.

The objective is to maximize $\mathbb{E}[V^\pi(S_1)]$.

4.2 Monte-Carlo Evaluation

Monte-Carlo Algorithm:

1. Take N (large integer)
2. For $k = 1, \dots, N$:
   - $s = s_1$, rew $= 0$
   - For $t = 1, \dots, T$:
     - Sample s' from $k(s, \pi_t(s), \cdot)$
     - rew $=$ rew $+ r(s, \pi_t(s), s')$
     - $s = s'$
   - results.append(rew)
3. Return mean(results)

Complexity: $O(NT \cdot |\mathcal{S}|)$

4.3 Choice of N

If $0 \leq r(s, a, s') \leq 1$ then $0 \leq \text{rew} \leq T$.

By Hoeffding's inequality:

$$\mathbb{P}(|\bar{Y}_k - \mathbb{E}[Y]| \geq \epsilon) \leq 2 \exp\left(-\frac{2N\epsilon^2}{T^2} \right) \leq \delta$$

We obtain: $N \geq \frac{T^2}{2\epsilon^2} \log\left(\frac{2}{\delta}\right)$

In the end, the complexity is $O\left(\frac{NT^3}{\epsilon^2} \log\left(\frac{1}{\delta}\right)\right)$

5 Backward Induction

5.1 Principle

Grid with time on the abscissa and state on the ordinate. We can fill the grid backwards.

Definition

Temporal Value Function

$$V_t^\pi(s) = \mathbb{E}^\pi \left[\sum_{k=t}^{T} R_k \middle| S_t = s \right] \text{ for } t = 1, \dots, T$$

$$V^\pi(s) = V_1^\pi(s)$$

For $t = T$, we know that:

$$V_T^\pi(s) = \mathbb{E}^\pi[R_T | S_T = s] = \mathbb{E}[r(s, \pi_T(s), S_{T+1})]$$

Recurrence relation:

$$V_t^\pi(s) = \mathbb{E}^\pi[r(S_t, \pi(S_t), S_{t+1}) + \sum_{k=t+1}^{T} R_k | S_t = s]$$

$$= \sum_{s' \in \mathcal{S}} p(s, \pi(s), s')[r(s, \pi(s), s') + V_{t+1}^\pi(s')]$$

5.2 Example: Photo-Booth Problem

Example

Photo-Booth Problem
Each shot has a value given by the law:

$$\sum_{k=1}^{3} p_k \delta_{v_k}$$

where $(p_1, p_2, p_3) \in \mathcal{M}_1(\{1, 2, 3\})$ and $v = (v_1, v_2, v_3) \in \mathbb{R}^3$.
The shots are independent.
After each shot, either stop and keep the shot or continue.
Strategy 1: Stop only if the shot has value v_3.
Grid of values with $pV = p_1 v_1 + p_2 v_2 + p_3 v_3$.

Complexity of backward induction: $O(T \cdot |\mathcal{S}|^2)$

This is better if $|\mathcal{S}|$ is small.

6 Finding an Optimal Policy

6.1 Definitions

Definition

Dominance and Optimality

- A policy π dominates a policy π' ($\pi \succ \pi'$) if $\forall s \in \mathcal{S}, V^{\pi}(s) \geq V^{\pi'}(s)$
- A policy π is optimal if $\forall \pi', \pi \succ \pi'$

Definition

Optimal Value Function

$$v^*(s) = \max_{\pi} V^{\pi}(s)$$

$v^*(s)$ is called the value function of the MDP.

Question: Is there always an optimal policy π^* such that $V^{\pi^*} = V^*$?

Answer: Yes.

6.2 Optimal Backward Induction Algorithm

Definition

Optimal Temporal Value Functions

$$V_t^*(s) = \max_{\pi} V_t^{\pi}(s) \text{ for } t = 1, \dots, T$$

With backward induction:
At $t = T$:

$$V_T^*(s) = \max_{a \in \mathcal{A}} \mathbb{E}^{\pi}[r(s, a, S_{T+1}) | S_T = s] = \max_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}} p(s, a, s') r(s, a, s')$$

$$\pi_T^*(s) = \arg\max_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}} p(s, a, s') r(s, a, s')$$

For $t = T - 1, \dots, 1$:

$$V_t^*(s) = \max_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}} p(s, a, s')[r(s, a, s') + V_{t+1}^*(s')]$$

Theorem

Backward Induction Algorithm

$$\begin{cases} Q(s, a, t) = \sum_{s' \in \mathcal{S}} k(s, a, s')[r(s, a, s') + V_{t+1}^*(s')] \\ V_t^*(s) = \max_{a \in \mathcal{A}} Q(s, a, t) \\ \pi_t^*(s) = \arg\max_{a \in \mathcal{A}} Q(s, a, t) \end{cases}$$

We can fill the grid backwards and choose the optimal action at each state.

7 Application to the Secretary Problem

We deduce that the optimal policy is to observe until a certain step then choose the first one that is better than all the previous ones.

Let's compute the value of step r such that:

- Never stop before $t = r$
- Stop for $t > r$ as soon as the current secretary is the current best

Backward induction shows that the optimal policy is of this form:

$$V(r) = \sum_{t=r}^{T} \frac{1}{T} \cdot \left(1 - \frac{1}{r}\right) \cdot \left(1 - \frac{1}{r+1}\right) \cdots \left(1 - \frac{1}{t-1}\right)$$

$$= \sum_{t=r}^{T} \frac{1}{T} \cdot \frac{r-1}{r} \cdot \frac{r}{r+1} \cdots \frac{t-2}{t-1}$$

$$= \sum_{t=r}^{T} \frac{1}{T} \cdot \frac{r-1}{t-1}$$

$$= \frac{r-1}{T} \sum_{t=r}^{T} \frac{1}{t-1} \approx \frac{r-1}{T}(\log(T) - \log(r))$$

for T large enough. This is roughly equal to $x \ln\left(\frac{1}{x}\right)$ with $x = \frac{r}{T}$.

The maximum is reached for $x = \frac{1}{e}$ thus $r = \frac{T}{e}$.

$$v(r^*) = \frac{1}{e}$$

This example characterizes optimal stopping problems.