Advanced Topics on Artificial Intelligence

Alban Grastien

The Australian National University

Second Semester, 2020

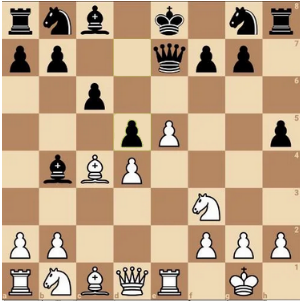
What is a Markov Decision Process?

- Dynamic system
- Uncertainty on the actions' effects
  - Uncontrollable, stochastic effect of the environment
  - Uncertainty linked to simplification

  but the **probability distribution** of the effects is known
- Complete observability

- **Importantly**: the (non-deterministic) effects of the actions are completely determined by the current state.
- In other words, the current state contains all the information about the past.

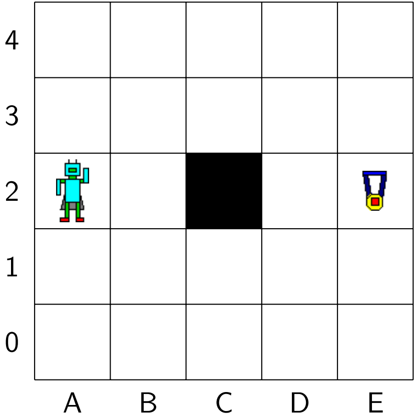
Are these Markov Decision Processes?



White to play

White to play

Our Running Example: Little Robot



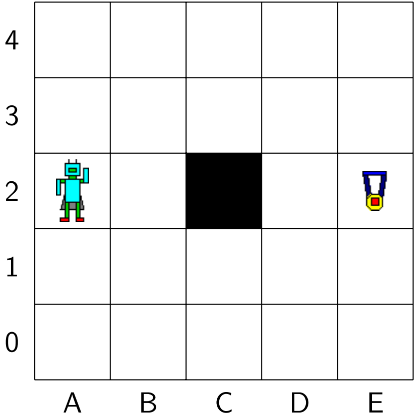
- Goal: get robot to as fast as possible
- Can move in all four directions: Up, Down, Left, Right or Stay.
- Moving right has 50% chance of going down too (if able)

Definition of Markov Decision Process

Markov Decision Process: $\langle S, A, P, R \rangle$

- S: (finite) set of states
- A: (finite) set of actions
- $P : S \times A \to Prob(S)$, the partial probabilistic transition function
  - $P(s, a, s')$ is the probability of reaching s' if you execute a in s.
  - A_s is the set of actions applicable in s
  - If a is applicable in sum, then $\Sigma_{s' \in S} \; P(s, a, s') = 1$, otherwise 0
- $R : A \to \mathbb{Q}$: reward function

Little Robot



- State: location in the grid $\langle x, y\rangle$
- Actions: Up, Down, Left, Right, Stay
- Some transition probabilities:
  - $P(A2, U, A3) = 1$
  - $P(A2, R, B2) = 0.5$
  - $P(A2, R, B1) = 0.5$
  - $P(A2, R, A1) = 0$
- Rewards:
  - $R(E2) = 100$
  - $R(s) = -1$ if $s \neq E2$

History

k-long history:

- A sequence of $k+1$ states and k actions:

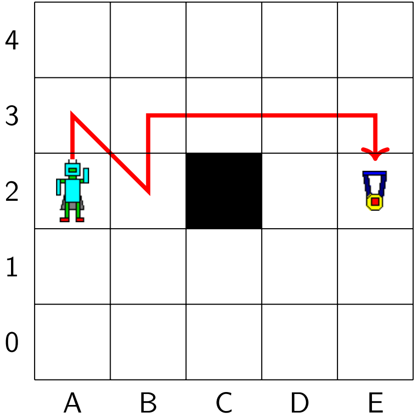
$$s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} \dots \xrightarrow{a_{k-1}} s_{k-1} \xrightarrow{a_k} s_k$$

  such that

  - a_i is applicable in state s_{i-1} and
  - $P(s_{i-1}, a_i, s_i) \neq 0$

- We write $\mathcal{H}$ the set of possible histories

Little Robot



A 7-long history:

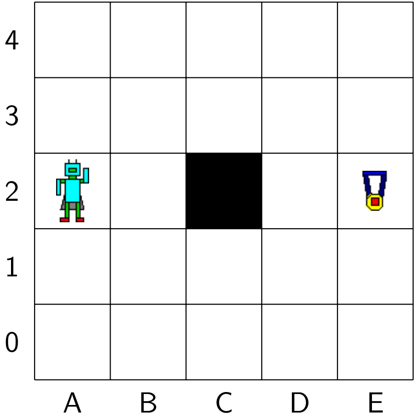
- $A2 \xrightarrow{U} A3 \xrightarrow{R} B2 \xrightarrow{U} B3 \xrightarrow{R} C3 \xrightarrow{R} D3 \xrightarrow{R} E3 \xrightarrow{D} E2$

Policy

Policy:

- A function $\pi : \mathcal{H} \to Prob(A)$ that, given a history h, return a probability distribution $\pi(h)$ over the actions
- Constraint:
  - $\pi(s_0, a_1, s_1, \ldots a_k, s_k)(a) > 0$ only if a is applicable in s_k

Little Robot



A policy:

- If the last move was Right–Down, then do Up.
- Else if $x < C$ and $y = 2$, then
    - 50% chance Up
    - 50% chance Down
- Else if $x < E$, Right
- Else if $y < 2$, Up
- Else if $y > 2$, Down
- Else Stay.

History Evaluation

How to evaluate a history?

- Essentially add up the rewards during the history.
- But doing so would mean, in the limit, the value would be $+\infty$ or $-\infty$,
  $\Rightarrow$ hard to compare two policies with infinite payoff
- Also the reward could oscillate (think reward $+1$, -1, $+1$, etc.)
- Also, immediate rewards are generally better (other things being equal)

History Evaluation

Discount factor $\gamma \in [0, 1)$

- Value of finite history $h_k = s_0 \xrightarrow{a_1} s_1 \dots s_{k-1} \xrightarrow{a_k} s_k$:

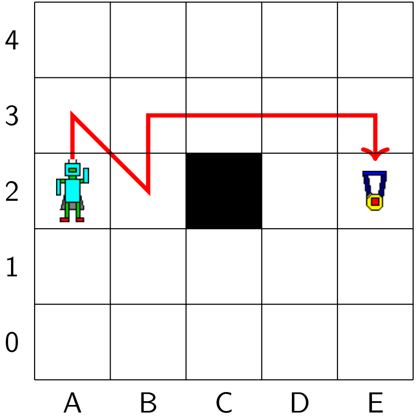
$$\begin{aligned} V(h_k) &\stackrel{def}{=} R(a_1) + (\gamma \times R(a_2)) + (\gamma^2 \times R(a_3)) + \dots \\ &\stackrel{def}{=} \Sigma_{i \in \{1,\dots,k\}} \left(\gamma^{i-1} \times R(a_i)\right). \end{aligned}$$

- Value of infinite history $h = s_0 \xrightarrow{a_1} s_1 \dots$:

$$\begin{aligned} V(h) &\stackrel{def}{=} \lim_{k \to \infty} V(h_k) \\ &= \Sigma_{i \in \{1,\dots,\infty\}} \left(\gamma^{i-1} \times R(a_i)\right) \end{aligned}$$

where h_k is the prefix of length k of h.

Little Robot



Set $\gamma := 0.9$ (goal = 100, move = -1)

- $V(h_7) = -1 + (\gamma^1 \times -1) + \cdots + (\gamma^6 \times -1) + (\gamma^7 \times 100) = 42.612$

Let $h_k = h_7 \xrightarrow{S} \ldots \xrightarrow{S} E2$

- $V(h_k) = \Sigma_{i \in \{7,\ldots,k\}} \gamma^i$
- $\lim_{k \to \infty} V(h_k) = \frac{\gamma^7}{1-\gamma} = 42.612$

Notice the Strong Assumption!

- It is assumed that the rewards add-up.
- A very large reward + a very large penalty is roughly the same as a zero reward.
- Is ethics just about accounting?
  - Trolley problem: kill one person to save five.

Value of Policy V_π

Probability of k-long history $h_k = s_0 \xrightarrow{a_1} s_1 \dots s_{k-1} \xrightarrow{a_k} s_k$ starting in s_0:

$$\begin{aligned} P(h_k) &= \pi(h_0)(a_1) \times P(s_0, a_1, s_1) \times \pi(h_1)(a_2) \times P(s_1, a_2, s_2) \times \dots \\ &\stackrel{def}{=} \left[\times_{i \in \{1,\dots,k\}} \ \pi(h_{i-1})(a_i) \right] \times \left[\times_{i \in \{1,\dots,k} P(s_{i-1}, a_i, s_i) \right] \end{aligned}$$

Value of Policy V_π

Value of policy at depth k:

$$V_{\pi,k}(s) \stackrel{def}{=} \Sigma_{h_k \in \mathcal{H}_k(s)} \Big(V(h_k) \times P(h_k) \Big)$$

We also write $\mathbb{E}_{h_k \sim \pi, M} V(h_k)$, i.e., the expected* value of $V(h_k)$ where h_k is randomly drawn according to the policy π and the MDP M.

* expected $=$ "averaged over probability distribution"

Notice the assumption: we want to maximise the **expected** reward.
What would you choose:

- 50% chance of gaining \$3M or nothing
- vs 100% chance of gaining \$1M?

Value of Policy V_π

Value of state s for policy π: value of the policy in the infinite horizon

$$V_\pi(s) \stackrel{def}{=} \lim_{k \to \infty} V_{\pi,k}(s).$$

Converges if $\gamma < 1$

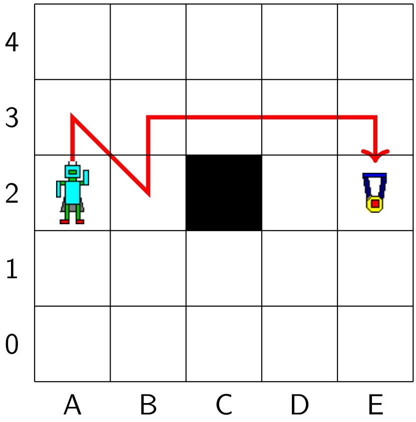
Comparing Policies

- Policy π is $\boxed{\text{better than or as good as}}$ π' if its value is greater from each state

$$\forall s \in S.\ V_\pi(s) \geq V_{\pi'}(s).$$

- A policy is $\boxed{\text{optimal}}$ if it is better than or as good as any other policy.

Little Robot



π' is better than π

Comparing two policies π and π' nearly identical:

- If the last move was Right–Down, then do Up.
- Else if $x < C$ and $y = 2$, then
  - 50% chance Up
  - 50% chance Down

  (except for π': always Up)
- Else if $x < E$, Right
- Else if $y < 2$, Up
- Else if $y > 2$, Down
- Else Stay.

Optimal Policy: theorems

Theorem

There is at least one optimal policy.

- If π and π' are different and neither is better than the other, then π is better on some states and π' is better on other states.
- The optimal policy is better on **all** states.

Markov Policy

- A policy is $\boxed{\text{Markov}}$ if it only depends on the current state
- Formally for any two histories
  - $h = s_0 \xrightarrow{a_1} s_1 \dots s_{k-1} \xrightarrow{a_k} s_k$ and
  - $h' = s'_0 \xrightarrow{a'_1} s'_1 \dots s'_{k'-1} \xrightarrow{a'_{k'}} s'_{k'}$

  then
  - $s_k = s'_{k'} \;\Rightarrow \pi(h) = \pi(h')$.

Example of non-Markov policy

In Blocks-World, the goal is to pile up blocks in a given order

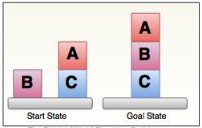


Fig: Blocks-World Planning Problem

A simple solution is:

- Unstack all blocks
- Stack blocks according to the goal

Why is this non-Markov?

Optimal Policy: theorems

Theorem

One of the optimal policies is Markov, i.e., it depends only on the current state.

Optimal Policy: theorems

- A policy π is **deterministic** if for any history h, there exists an action a such that $\pi(h)(a) = 1$.

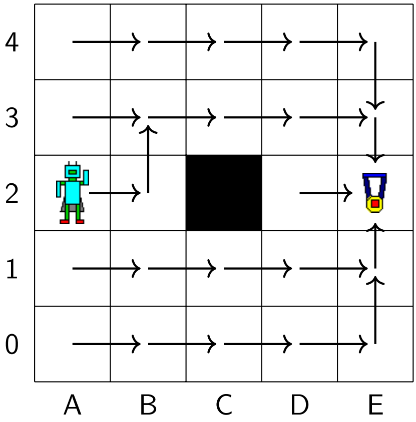
Theorem

One of the Markov optimal policy is deterministic.

- Consequently, we only consider Markov deterministic policies, and we rewrite $\pi : S \to A$.
- $\pi(s)$ is the action applied in state s.

What about bluffing? What about poker?

Little Robot



Optimal policy

Stochastic Shortest Path

Let's forget MDPs for a moment

Slightly different problem with the following differences:

- The problem ends when you reach a "goal" $\rightarrow$ indefinite problem.
- The objective is to minimise the cost (not maximise the reward).
- There is no discount factor.

Stochastic Shortest Path (SSP)

Tuple $\langle S, G, A, P, C \rangle$ where

- S is a set of states.
- $G \subseteq S$ is the set of goal states.
- A is a set of actions.
- $P : S \times A \to Prob(S)$ is the function that indicates the probability $P(s, a, s')$ of reaching s' when applying a in s.
- $C : A \to \mathbb{Q}^+$ is the cost of applying action a.

Similar semantics with undiscounted cost. Stops in the goal state.

From MDP to SSP

It is possible to translate an MDP into an SSP:

- Add a single goal state g.
- For any transition $s \xrightarrow{a} s'$, replace the probability with $(1-\gamma)P(s,a,s')$.
- For any state s, for any applicable action a, create a transition $s \xrightarrow{a} g$ with probability $P(s,a,g)=\gamma$.

Three $\times$ two different interpretations

1. Problems with a bounded length (example: Chess)
2. Problems with a discount: indefinite
3. Problems with finite horizon: do not care about the rewards after this horizon

Also,

1. Maximise rewards
2. Minimise costs

We will move back and forth between these definitions in an inconsistent manner: stay open-minded!

How to find the optimal policy?

Bellmann Equations

Those equations determine the optimal action in each state, as well as the expected value in each state.

It is not necessary to look at all histories!

- The value of state s is the value of the **best** action in this state
- The value of an action in a state is the **expected sum***:
  - the reward this action will provide
  - the (discounted) value from the next state

Bellmann Equations

Optimal state value:

$$V^*(s) \quad \stackrel{def}{=} \quad V_{\pi^*}(s).$$

Characterising V^*:

$$\begin{aligned} V^*(s) &= \begin{cases} 0 & \text{if } s \in G \\ \min_{a \in A(s)} \; Q^*(s,a) & \end{cases} \\ Q^*(s,a) &= \Sigma_{s' \in S} \Big(P(s,a,s') \cdot (C(s,a,s') + \gamma V^*(s')) \Big) \end{aligned}$$

(Here, we minimise the cost)

Bellmann Equations

Optimal state value:

$$V^*(s) \quad \stackrel{def}{=} \quad V_{\pi^*}(s).$$

Characterising V^*:

$$\begin{aligned}
V^*(s) \quad &= \quad \begin{cases} 0 & \text{if } s \in G \\ \min_{a \in A(s)} \; Q^*(s,a) & \end{cases} \\
Q^*(s,a) \quad &= \quad \Sigma_{s' \in S} \Big(P(s,a,s') \cdot (C(s,a,s') + \gamma V^*(s')) \Big)
\end{aligned}$$

(Here, we minimise the cost)

The next action in state s should be:

$$\arg \min_{a \in A(s)} Q^*(s,a)$$

- The value of a state is often written $V(\cdot)$
- The value of a state-action pair $Q(\cdot,\cdot)$