Markov Decision Processes

Self-Adaptive & Learning Systems

Milan R. Rapaić

Chair of Automatic Control
Computing and Control Department
Faculty of Technical Sciences
University of Novi Sad
Novi Sad • Serbia

October 18, 2023

Outline

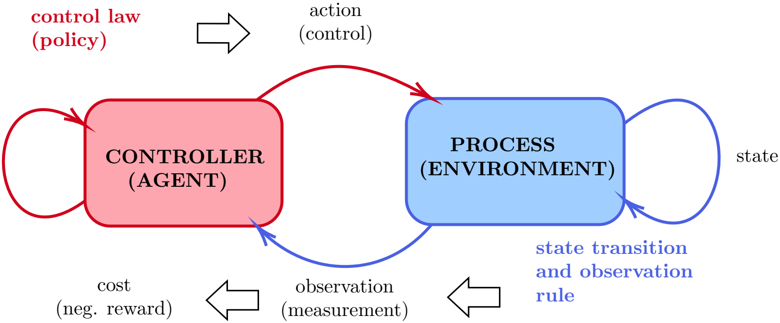
1. Markov Decision Processes – The Model

2. Decision Making using State and Action Values

3. Exploration vs. Exploitation

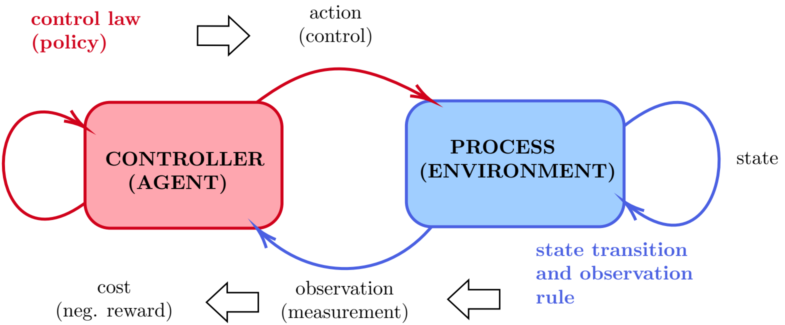
What is Reinforcement Learning?

RL is a framework for sequential decision making.



What is Reinforcement Learning?

RL is a framework for sequential decision making.



Reinforcement Learning is

a model-free framework for solving optimal control problems stated as **Markov Decision Processes** (MDPs).

Markov Decision Processes – The Model

1. Markov Decision Processes – The Model
2. Decision Making using State and Action Values
3. Exploration vs. Exploitation

Markov Decission Processes

MDP is a model of the environment (process). It tells us ...

- ... how the state of the process is changing in reaction to the applied (control) actions
- ... what observations the agent (controller) may receive

Based on the received observation, the agent (controller) evaluates assesses the immediate (short-term) effects of the applied action. In the AI community, it is often said that the agent evaluates a reward, while in the control community one often speaks of a penalty. The two positions are philosophically different, but essentially equivalent

Deterministic MDP

$$s^+ = f(s, a)$$

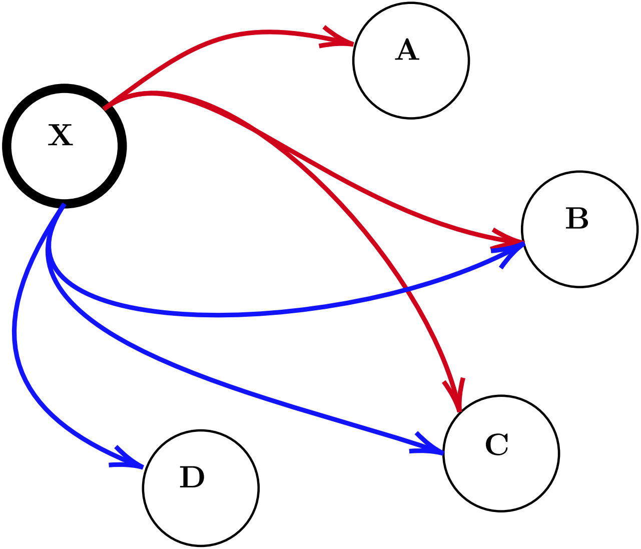
$$r = h(a, u)$$

Stochastic MDP

$$p(s^+, r|s, a) = \mathbb{P}\left\{S^+ = s^+, R = r\middle|S = s, A = a\right\}$$

MDP as a graph

... a conceptual representation



Gain

The goal is never to optimize short-term returns. We are always interested in optimizing measures of long-term success: maximizing long-term gains (or minimizing long-term losses).

Let r_0, r_1, ... be a sequence of rewards obtained in subsequent time instances. The gain is defines as

Deterministic MDP

$$g = \sum_{k=0}^{T} \gamma^k r_k$$

Stochastic MDP

$$g = \mathbb{E}\left\{ \sum_{k=0}^{T} \gamma^k R_k \right\}$$

where the discount factor γ (typically chosen between 0 and 1) accounts for the fact that high rewards expected in the far future are often favored less than substantially smaller rewards to be received immediately.

Decision Policy

... a.k.a. Control Law

Decision policy describes actions of the controller: it tells us how controller will act *in each possible observed state* of the environment.

Deterministic Policy

$$a = \pi(s)$$

Stochastic Policy

$$\pi(a|s) = \mathbb{P}\left\{A = a \middle| S = s\right\}$$

One can apply both deterministic and stochastic policies regardless of the nature of the environment (to both stochastic and deterministic MDPs).

Gain revisited

as a function of the initial state and the decision policy

Given an MDP in a certain initial state s_0 controlled by an agent applying policy π, one can observer a sequence of applied actions, environment states and rewards.

$$\begin{aligned} \textbf{deterministic case}: &\quad s_0 \to a_0 \to r_0, s_1 \to a_1 \to r_1, s_2 \to ... \\ \textbf{stochastic case}: &\quad s_0 \to A_0 \to R_0, S_1 \to A_1 \to R_1, S_2 \to ... \end{aligned}$$

Once the initial state is fixed, together with the decision policy, these sequences are fixed as well, and so is the gain! Therefore, it is possible to associate a map between initial state and decision policies with the resulting gain for the given MDP.

$$g_\pi(s) = \mathbb{E}_\pi \left\{ \sum_{k=0}^{T} \gamma^k R_k \Big| S_0 = s \right\}$$

Decision Making using State and Action Values

1 Markov Decision Processes – The Model

2 Decision Making using State and Action Values

3 Exploration vs. Exploitation

The Value

of a state, or of an action in state, for the given policy

State-Value Function for policy

The value of a state s for the policy π is the gain that a controller following decision policy π attains when the environment starts from the initial state s.

$$v_\pi(s) = g_\pi(s)$$

Action-Value Function for policy

The value of an action a in state s for the policy π is the gain that a controller attains when the environment starts from state s when the initial action a and the policy π is followed afterward

$$q_\pi(s,a) = g \text{ when } s_0 = s, a_0 = a \text{ and policy } \pi \text{ is used for } k \geq 1$$

The Optimal Values

of a state, or of an action in state

Let $\mathscr{P}$ denote the set of all possible policies (regardless if they are deterministic or stochastic).

Optimal State-Value

The optimal value of a state is the maximal value of that state under any policy:

$$v^*(s) = \max_{\pi \in \mathscr{P}} v_\pi(s)$$

Optimal State-Action Value

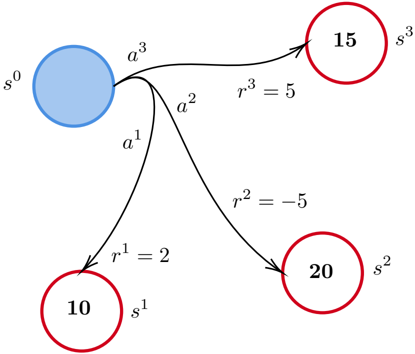
The optimal value of an action within a state is the maximal value of that action within that state under any policy:

$$q^*(s, a) = \max_{\pi \in \mathscr{P}} q_\pi(s, a)$$

Decision Making using Optimal State Values

Assume that I know values of all states, and assuming that I know that the environment is in state s_0,

- How to choose the best action?
- What else do I need to know in order to be able to choose?



$\gamma = 0.9$

a	s^+	$v(s^+)$	r	g
a^1	s^1	10	2	11
a^2	s^2	20	-5	13
a^3	s^3	15	5	18.5

$$a^* = \pi^*(s) \in \underset{a\in\mathcal{A}}{\operatorname{argmax}} \{h(s,a) + \gamma v^*(f(s,a))\}$$

Decision Making using Optimal State-Action Values

Assume that I know values of all actions in all states, and assuming that I know that the environment is in state s_0,

- How to choose the best action?
- What else do I need to know in order to be able to choose?

$\gamma = 0.9$

a	$q(s,a)$
a^1	11
a^2	13
a^3	18.5

$$a^* = \pi^*(s) \in \operatorname*{argmax}_{a \in \mathcal{A}} q^*(s,a)$$

Decision Making using Values

Concluding remarks...

Arguably, state values are easier to understand in comparison to the state-action values, however state-action values (q-values) are more convenient for decision making purposes.

- q-values store all relevant information for decision making. The explicit model is not needed!
- It is not possible to decide based on the state-values alone. In this case, an explicit model is also necessary.
- Storing q-values is more expensive than storing v-values, however in most practical cases the states are abundant while the actions are relatively few, so the practical difference is actually not that significant.

Exploration vs. Exploitation

1 Markov Decision Processes – The Model

2 Decision Making using State and Action Values

3 Exploration vs. Exploitation

Greedy and Optimal Policy

Given an *arbitrary* (not necessarily optimal) state value function v, or state action value function q, the greedy policy is defined as

$$a^{\textsf{greedy}} = \pi_v^{\textsf{greedy}}(s) \in \underset{a \in \mathcal{A}}{\operatorname{argmax}} \{h(s,a) + \gamma v(f(s,a))\}$$

$$a^{\textsf{greedy}} = \pi_q^{\textsf{greedy}}(s) \in \underset{a \in \mathcal{A}}{\operatorname{argmax}} \, q(s,a)$$

Optimal Policy

The greedy policy with respect to the optimal value functions v^* and q^* is the optimal policy.

Exploration vs. Exploitation

If the estimated values of v and q are not optimal, the greedy policy with respect to them need not be good, nor even reasonable.

Even worse, by consistently following a non-optimal greedy policy we keep revisiting the same combinations of states and actions over-and-over again. We continue to exploit existing incomplete knowledge of the system, without any attempt to explore, to experiment, to advance our understanding, and possibly to find better actions in certain situations.

In situations when we are learning (because we do not know the system sufficiently, or we know it but anticipate that it will change in the future) **completely exploitative (greedy) policy should be avoided**, and explorative component must be added to the decision-making process.

Random Policy

... as an example of a completely explorative policy

Random policy is suitable in situations where the rewards (and gain) are completely neglected, and the goal is not to maximize the gain but to learn as much as possible about the environment (i.e. about the controlled system).

$$\pi^{\text{random}}(s) = \text{choose } a \text{ for } \mathcal{A} \text{ randomly with uniform probability}$$

ε-greedy Policy

... as an example of a balanced policy

In reality, one is interested in **simultaneously** maximizing the gain **and** exploring the environment. This is a multi-criteria optimization problem, and it is necessary to establish a tradeoff between two **conflicting** goals.

$$\pi^{\varepsilon}(s) = \begin{cases} \pi^{\text{random}}(s) \text{ with probability } \varepsilon \\ \pi^{\text{greedy}}(s) \text{ with probability } 1 - \varepsilon \end{cases}$$

Further Reading

For further reading please consult [Sutton and Barto, 2018]. Especially pay attention to **Chapter 3** in which you may find several illustrative examples illustrating various components of MDP models.

Sutton, R. S. and Barto, A. G. (2018).
Reinforcement Learning: An Introduction.
The MIT Press, second edition.