Data-driven and Learning-based Control

Markov Decision Processes

Erica Salvato

Table of Contents

Optimal control

1 A brief recap

- Formal definition of an optimal control problem in terms of
  - performance index (e.g., cost or reward)
  - physical constraints
- Observability and reachability properties of a dynamical system
  - special case of linear dynamical systems
- Towards the solution of an optimal control problem
  - Bellman's principle of optimality
  - Dynamic programming (bottom-up approach)
  - Value function
  - Bellman's optimality equations
    - value iteration (forward approach)
    - policy iteration (forward approach)

Optimal control

1 A brief recap

- Linear quadratic regulator (LQR)
  - infinite horizon
  - finite horizon
  - time-varying
  - tracking
- LQR for non-linear system
  - linearization around an equilibrium point
  - iterative LQR

Table of Contents

What are we able to do so far?

We are able to solve optimal control problems in the case of dynamics that are:

- deterministic (linear or non-linear)
- known.

What are we able to do so far?

We are able to solve optimal control problems in the case of dynamics that are:

- deterministic (linear or non-linear)
- known.

What is a deterministic system?

What are we able to do so far?

We are able to solve optimal control problems in the case of dynamics that are:

- deterministic (linear or non-linear)
- known.

Deterministic system

A deterministic system is dynamical system in which the future behavior is completely determined by its initial conditions $x^{(0)}$ and the governing laws $f\left(x^{(k)}, u^{(k)}\right)$.
In simpler terms, if we know $x^{(0)}$ and $f(\cdot, \cdot)$ we can predict its future state with certainty.

$$x^{(k+1)} = f\left(x^{(k)}, u^{(k)}\right)$$

What are we able to do so far?

We are able to solve optimal control problems in the case of dynamics that are:

- deterministic (linear or non-linear)
- known.

What we mean by known system?

What are we able to do so far?

We are able to solve optimal control problems in the case of dynamics that are:

- deterministic (linear or non-linear)
- known.

Known deterministic system

A known deterministic system refers to a deterministic system in which $f(\cdot, \cdot)$ is perfectly known, i.e., we have no inherent randomness or uncertainty about how the system will evolve over time.
We can compute

$$x^{(k+1)} = f\left(x^{(k)}, u^{(k)}\right)$$

What are we able to do so far?

We are able to solve optimal control problems in the case of dynamics that are:

- deterministic (linear or non-linear)
- known.

1. What if we are not dealing with a deterministic environment?
2. What if we don't know the model?

What are we able to do so far?

We are able to solve optimal control problems in the case of dynamics that are:

- deterministic (linear or non-linear)
- known.

1. What if we are not dealing with a deterministic environment? $\rightarrow$ **Markov Decision Process**
2. What if we don't know the model? $\rightarrow$ **Learning**

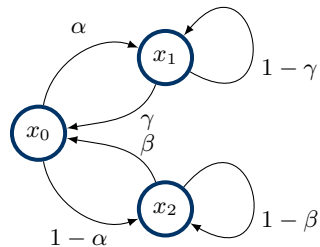
Table of Contents

Markov Decision Processes

A Markov Decision Process (MDP) serves as a robust mathematical framework for modeling decision-making under uncertainty.

Components of MDPs:

- State $\rightarrow x^{(k)} \in \mathcal{X} \subseteq \mathbb{R}^n$
- Control input $\rightarrow u^{(k)} \in \mathcal{U} \subseteq \mathbb{R}^m$
- Transition probability $\rightarrow T : \mathcal{X} \times \mathcal{U} \times \mathcal{X} \rightarrow [0, 1]$
- Reward function $\rightarrow h : \mathcal{X} \times \mathcal{U} \times \mathcal{X} \rightarrow \mathbb{R}$
- Markov property



Markov property

The Markov property states that the probability of observing future states $x^{(k+1)}$, given the past, depends only on the most recent state $x^{(k)}$:

$$Pr\left(x^{(k+1)}|x^{(k)}, u^{(k)}\right) = T\left(x^{(k)}, u^{(k)}\right)$$

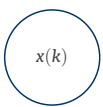
Therefore, $x^{(k)}$ is sufficient to determine the transition probability in $x^{(k+1)}$

Markov property

The Markov property states that the probability of observing future states $x^{(k+1)}$, given the past, depends only on the most recent state $x^{(k)}$:

$$Pr\left(x^{(k+1)}|x^{(k)}, u^{(k)}\right) = T\left(x^{(k)}, u^{(k)}\right)$$

Therefore, $x^{(k)}$ is sufficient to determine the transition probability in $x^{(k+1)}$.

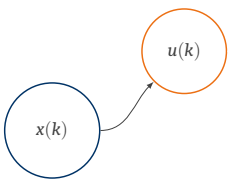


Markov property

The Markov property states that the probability of observing future states $x^{(k+1)}$, given the past, depends only on the most recent state $x^{(k)}$:

$$Pr\left(x^{(k+1)}|x^{(k)}, u^{(k)}\right) = T\left(x^{(k)}, u^{(k)}\right)$$

Therefore, $x^{(k)}$ is sufficient to determine the transition probability in $x^{(k+1)}$.

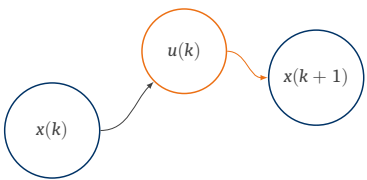


Markov property

The Markov property states that the probability of observing future states $x^{(k+1)}$, given the past, depends only on the most recent state $x^{(k)}$:

$$Pr\left(x^{(k+1)}|x^{(k)}, u^{(k)}\right) = T\left(x^{(k)}, u^{(k)}\right)$$

Therefore, $x^{(k)}$ is sufficient to determine the transition probability in $x^{(k+1)}$.

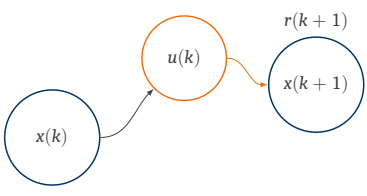


Markov property

The Markov property states that the probability of observing future states $x^{(k+1)}$, given the past, depends only on the most recent state $x^{(k)}$:

$$Pr\left(x^{(k+1)}|x^{(k)}, u^{(k)}\right) = T\left(x^{(k)}, u^{(k)}\right)$$

Therefore, $x^{(k)}$ is sufficient to determine the transition probability in $x^{(k+1)}$.

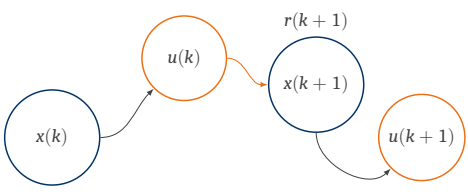


Markov property

The Markov property states that the probability of observing future states $x^{(k+1)}$, given the past, depends only on the most recent state $x^{(k)}$:

$$Pr\left(x^{(k+1)}|x^{(k)}, u^{(k)}\right) = T\left(x^{(k)}, u^{(k)}\right)$$

Therefore, $x^{(k)}$ is sufficient to determine the transition probability in $x^{(k+1)}$.

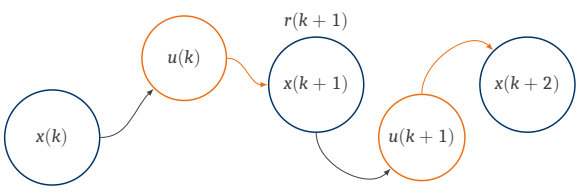


Markov property

The Markov property states that the probability of observing future states $x^{(k+1)}$, given the past, depends only on the most recent state $x^{(k)}$:

$$Pr\left(x^{(k+1)}|x^{(k)}, u^{(k)}\right) = T\left(x^{(k)}, u^{(k)}\right)$$

Therefore, $x^{(k)}$ is sufficient to determine the transition probability in $x^{(k+1)}$.

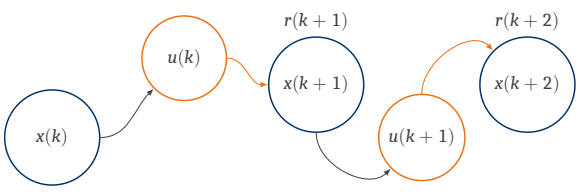


Markov property

The Markov property states that the probability of observing future states $x^{(k+1)}$, given the past, depends only on the most recent state $x^{(k)}$:

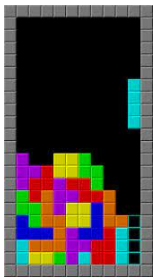
$$Pr\left(x^{(k+1)}|x^{(k)}, u^{(k)}\right) = T\left(x^{(k)}, u^{(k)}\right)$$

Therefore, $x^{(k)}$ is sufficient to determine the transition probability in $x^{(k+1)}$.



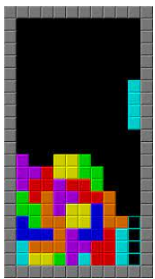
Tetris example

Tetris is a game where randomly falling pieces must be placed on the game board. Each horizontal line completed is cleared from the board and scores points for the player. The game terminates when the board fills up. The game of Tetris can be modeled as a MDP:



Tetris example

Tetris is a game where randomly falling pieces must be placed on the game board. Each horizontal line completed is cleared from the board and scores points for the player. The game terminates when the board fills up. The game of Tetris can be modeled as a MDP:



- **States:** Board configuration (each cell can be filled/not filled)
- **Control inputs:** Columns and from up to 4 possible orientations of the falling pieces
- **Transition probability:** A deterministic update of the board plus the selection of a random piece for the next time-step.

Table of Contents

4 Markov Decision Problem

Markov Decision Problems

4 Markov Decision Problem

The Markov Decision Problem is instead an optimal control problem whose objective is to find an optimal controller π^* that dictates the actions in each state of a given MDP in order to:

$$\pi^* = \arg\max_{\pi} \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k \boldsymbol{h}\left(\boldsymbol{x}^{(k)}, \pi\left(\boldsymbol{x}^{(k)}\right), \boldsymbol{x}^{(k+1)}\right)\right]$$

Markov Decision Problems

4 Markov Decision Problem

The Markov Decision Problem is instead an optimal control problem whose objective is to find an optimal controller π^* that dictates the actions in each state of a given MDP in order to:

$$\pi^* = \arg\max_{\pi} \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k \boldsymbol{r}^{(k+1)}\right]$$

expected discounted cumulative reward

where $\gamma \in [0, 1)$ is a discount factor.

Why discounted rewards?

$$\sum_{k=0}^{\infty} \gamma^k \boldsymbol{r}^{(k+1)}$$

Markov Decision Problems

4 Markov Decision Problem

Why discounted rewards?

$$\sum_{k=0}^{\infty} \gamma^k \boldsymbol{r}^{(k+1)}$$

- **Intuitive reasoning:** allows the reward to be weighted over time and thus dictate how far ahead the controller is in assessing the reward

$$\boldsymbol{r}^{(1)} + \gamma \boldsymbol{r}^{(2)} + \gamma^2 \boldsymbol{r}^{(3)} + \gamma^3 \boldsymbol{r}^{(4)} + \dots$$

Markov Decision Problems

4 Markov Decision Problem

Why discounted rewards?

$$\sum_{k=0}^{\infty} \gamma^k \boldsymbol{r}^{(k+1)}$$

- **Intuitive reasoning:** allows the reward to be weighted over time and thus dictate how far ahead the controller is in assessing the reward

$$\boldsymbol{r}^{(1)} + \gamma \boldsymbol{r}^{(2)} + \gamma^2 \boldsymbol{r}^{(3)} + \gamma^3 \boldsymbol{r}^{(4)} + \dots$$

- **Mathematical reasoning:** the discount factor $\gamma \in [0, 1)$ guarantees the convergence of the geometric series

How to choose γ?

- There is no general rule
- It should be chosen large enough so that the reward for reaching the target is detectable in value by the controller
- But not too large otherwise convergence slows down

How to choose γ?

- There is no general rule
- It should be chosen large enough so that the reward for reaching the target is detectable in value by the controller
- But not too large otherwise convergence slows down

The typical choices of γ are 0.9 and 0.95. But this is not a fixed rule

Markov Decision Problem

4 Markov Decision Problem

Define:

- $x^{(k)} \in \mathcal{X} \subseteq \mathbb{R}^n$ the n-dimensional state
- $u^{(k)} \in \mathcal{U} \subseteq \mathbb{R}^m$ the m-dimensional input
- $k \in \mathbb{Z}_0^+$ the time-step index
- $T\left(x^{(k)}, u^{(k)}\right)$ the transition function
- $h : \mathcal{X} \times \mathcal{U} \times \mathcal{X} \to \mathbb{R}$ the reward function
- $\gamma \in [0, 1)$ the discount factor

Markov Decision Problem

Define:

- $x^{(k)} \in \mathcal{X} \subseteq \mathbb{R}^n$ the n-dimensional state
- $u^{(k)} \in \mathcal{U} \subseteq \mathbb{R}^m$ the m-dimensional input
- $k \in \mathbb{Z}_0^+$ the time-step index
- $T\left(x^{(k)}, u^{(k)}\right)$ the transition function
- $h : \mathcal{X} \times \mathcal{U} \times \mathcal{X} \to \mathbb{R}$ the reward function
- $\gamma \in [0, 1)$ the discount factor

MDPs an Markov Decision Problem

An MDP is a tuple $(\mathcal{X}, \mathcal{U}, T, h)$, and the Markov decision problem is the optimal control problem of finding the optimal controller $\pi^*\left(x^{(k)}\right)$ such that

$$\pi^* = \arg\max_{\pi} \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k h\left(x^{(k)}, \pi\left(x^{(k)}\right), x^{(k+1)}\right)\right]$$

$$\text{s.t. } Pr\left(x^{(k+1)} | x^{(k)}, u^{(k)}\right) = T\left(x^{(k)}, u^{(k)}\right)$$

From Markov Decision Problem to Optimal Control Problem

Note that any Markov decision problem can be transformed into a classical optimal control problem that we have already studied if:

- $Pr\left(x^{(k+1)}|x^{(k)}, u^{(k)}\right) = T\left(x^{(k)}, u^{(k)}\right) = 1$

From Markov Decision Problem to Optimal Control Problem

4 Markov Decision Problem

Note that any Markov decision problem can be transformed into a classical optimal control problem that we have already studied if:

- $Pr\left(x^{(k+1)}|x^{(k)},u^{(k)}\right)=T\left(x^{(k)},u^{(k)}\right)=1 \implies x^{(k+1)}=f\left(x^{(k)},u^{(k)}\right)$

From Markov Decision Problem to Optimal Control Problem

4 Markov Decision Problem

Note that any Markov decision problem can be transformed into a classical optimal control problem that we have already studied if:

- $Pr\left(x^{(k+1)}|x^{(k)},u^{(k)}\right)=T\left(x^{(k)},u^{(k)}\right)=1 \implies x^{(k+1)}=f\left(x^{(k)},u^{(k)}\right)$
- $\gamma=1$

From Markov Decision Problem to Optimal Control Problem

Note that any Markov decision problem can be transformed into a classical optimal control problem that we have already studied if:

- $Pr\left(x^{(k+1)}|x^{(k)},u^{(k)}\right)=T\left(x^{(k)},u^{(k)}\right)=1 \implies x^{(k+1)}=f\left(x^{(k)},u^{(k)}\right)$
- $\gamma=1$

and recalling that

$$\max h(\cdot)=-\min -h(\cdot)$$

From Markov Decision Problem to Optimal Control Problem

Note that any Markov decision problem can be transformed into a classical optimal control problem that we have already studied if:

- $Pr\left(x^{(k+1)}|x^{(k)},u^{(k)}\right)=T\left(x^{(k)},u^{(k)}\right)=1 \implies x^{(k+1)}=f\left(x^{(k)},u^{(k)}\right)$
- $\gamma=1$

and recalling that

$$\max h(\cdot)=-\min -h(\cdot)$$

Therefore, what we have solved so far is a **deterministic MDP problem**.

Value function and Bellman equation

Everything we have defined for the classical optimal control problem applies, with some appropriate modification, to the Markov decision problems:

Value function and Bellman equation

Everything we have defined for the classical optimal control problem applies, with some appropriate modification, to the Markov decision problems:

- **Value function:** is a function used to measure the expected discounted sum of rewards from following a specific policy π from state $x^{(k)}$.

$$V_\pi\left(x^{(k)}\right) = \lim_{H\to\infty} \mathbb{E}_{x^{(k+1)}\sim T\left(x^{(k)},\pi\left(x^{(k)}\right)\right)} \left\{ \sum_{k=0}^{K} \gamma^k h\left(x^{(k)}, \pi\left(x^{(k)}\right), x^{(k+1)}\right) \right\}$$

which can also be written as follows:

$$V_\pi\left(x^{(k)}\right) = \mathbb{E}_{x^{(k+1)}\sim T\left(x^{(k)},\pi\left(x^{(k)}\right)\right)} \left\{ h\left(x^{(k)}, \pi\left(x^{(k)}\right), x^{(k+1)}\right) + \gamma V_\pi\left(x^{(k+1)}\right) \right\}$$

Value function and Bellman equation

- **Optimal value function:** is the value function of the optimal controller π^*

$$V_\pi^* = \max_\pi V_\pi \left(\boldsymbol{x}^{(k)} \right)$$

which can also be written as follows:

$$V_\pi^* \left(\boldsymbol{x}^{(k)} \right) = \max_\pi \mathbb{E}_{\boldsymbol{x}^{(k+1)} \sim T\left(\boldsymbol{x}^{(k)}, \pi\left(\boldsymbol{x}^{(k)} \right) \right)} \left\{ h \left(\boldsymbol{x}^{(k)}, \pi \left(\boldsymbol{x}^{(k)} \right), \boldsymbol{x}^{(k+1)} \right) + \gamma V_\pi^* \left(\boldsymbol{x}^{(k+1)} \right) \right\}$$

Value function and Bellman equation

- **Optimal value function:** is the value function of the optimal controller π^*

$$V_\pi^* = \max_\pi V_\pi\left(x^{(k)}\right)$$

which can also be written as follows:

$$V_\pi^*\left(x^{(k)}\right) = \max_\pi \mathbb{E}_{x^{(k+1)}\sim T\left(x^{(k)},\pi\left(x^{(k)}\right)\right)}\left\{h\left(x^{(k)},\pi\left(x^{(k)}\right),x^{(k+1)}\right)+\gamma V_\pi^*\left(x^{(k+1)}\right)\right\}$$

- **Optimal controller:** is the controller that yields the highest value for each state $x^{(k)}$

$$\pi^*\left(x^{(k)}\right) = \arg\max_\pi \mathbb{E}_{x^{(k+1)}\sim T\left(x^{(k)},\pi\left(x^{(k)}\right)\right)}\left\{h\left(x^{(k)},\pi\left(x^{(k)}\right),x^{(k+1)}\right)+\gamma V_\pi^*\left(x^{(k+1)}\right)\right\}$$

Value function and Bellman equation

- **Optimal value function:** is the value function of the optimal controller π^*

$$V_\pi^* = \max_\pi V_\pi \left(x^{(k)} \right)$$

which can also be written as follows:

$$V_\pi^* \left(x^{(k)} \right) = \max_\pi \mathbb{E}_{x^{(k+1)} \sim T\left(x^{(k)}, \pi\left(x^{(k)}\right)\right)} \left\{ h \left(x^{(k)}, \pi \left(x^{(k)} \right), x^{(k+1)} \right) + \gamma V_\pi^* \left(x^{(k+1)} \right) \right\}$$

- **Optimal controller:** is the controller that yields the highest value for each state $x^{(k)}$

$$\pi^* \left(x^{(k)} \right) = \arg\max_\pi \mathbb{E}_{x^{(k+1)} \sim T\left(x^{(k)}, \pi\left(x^{(k)}\right)\right)} \left\{ h \left(x^{(k)}, \pi \left(x^{(k)} \right), x^{(k+1)} \right) + \gamma V_\pi^* \left(x^{(k+1)} \right) \right\}$$

BELLMAN'S OPTIMALITY EQUATION

Action-value function

Value functions only describe the quality of the states. In order to infer the quality of transitions a model of the MDP is required.

- **Action-value function:** is a function that gives the reward obtained when starting from a certain state, a certain action is applied and then π is followed:

$$Q_\pi\left(x^{(k)}, u^{(k)}\right) = \mathbb{E}\left\{h\left(x^{(k)}, u^{(k)}, x^{(k+1)}\right) + \gamma Q_\pi\left(x^{(k+1)}, \pi\left(x^{(k+1)}\right)\right)\right\}$$

Action-value function

Value functions only describe the quality of the states. In order to infer the quality of transitions a model of the MDP is required.

- **Action-value function:** is a function that gives the reward obtained when starting from a certain state, a certain action is applied and then π is followed:

$$Q_\pi\left(x^{(k)}, u^{(k)}\right) = \mathbb{E}\left\{h\left(x^{(k)}, u^{(k)}, x^{(k+1)}\right) + \gamma Q_\pi\left(x^{(k+1)}, \pi\left(x^{(k+1)}\right)\right)\right\}$$

- **Optimal action-value function:** is a function that gives the reward obtained when starting from a certain state, a certain action is applied and then π is followed:

$$Q_\pi^*\left(x^{(k)}, u^{(k)}\right) = \mathbb{E}\left\{h\left(x^{(k)}, u^{(k)}, x^{(k+1)}\right) + \gamma \max_{u^{(k+1)}} Q_\pi\left(x^{(k+1)}, u^{(k+1)}\right)\right\}$$

Action-value function

- **Optimal controller:** is the controller that yields the highest value for each state $x^{(k)}$

$$\pi^* \left(x^{(k)} \right) = \arg\max_u Q_\pi^* \left(x^{(k)}, u \right)$$

Action-value function

- **Optimal controller:** is the controller that yields the highest value for each state $x^{(k)}$

$$\pi^* \left(x^{(k)} \right) = \arg\max_u Q_\pi^* \left(x^{(k)}, u \right)$$

We can also compute $V_\pi \left(x^{(k)} \right)$ in terms of $Q_\pi \left(x^{(k)}, u^{(k)} \right)$:

$$V_\pi \left(x^{(k)} \right) = Q_\pi \left(x^{(k)}, \pi \left(x^{(k)} \right) \right)$$

Action-value function

- **Optimal controller:** is the controller that yields the highest value for each state $x^{(k)}$

$$\pi^* \left(x^{(k)} \right) = \arg\max_u Q_\pi^* \left(x^{(k)}, u \right)$$

We can also compute $V_\pi \left(x^{(k)} \right)$ in terms of $Q_\pi \left(x^{(k)}, u^{(k)} \right)$:

$$V_\pi \left(x^{(k)} \right) = Q_\pi \left(x^{(k)}, \pi \left(x^{(k)} \right) \right)$$

We can also compute: $V_\pi \left(x^{(k)} \right)$ in terms of $Q_\pi \left(x^{(k)}, u^{(k)} \right)$:

$$V_\pi^* \left(x^{(k)} \right) = Q_\pi^* \left(x^{(k)}, \pi^* \left(x^{(k)} \right) \right)$$

Solving model-based Markov decision problems

Similar to classical optimal control, two ways to solve Markov decision problems, in the case of a known MDP model, are:

- Q-iteration
- Policy iteration

Q-iteration

4 Markov Decision Problem

Consider the optimal control problem of finding:

$$\pi^* \left(\boldsymbol{x}^{(k)} \right) = \arg\max_{u} \; Q_\pi^* \left(\boldsymbol{x}^{(k)}, u \right)$$

$$\text{s.t.} \qquad (\mathcal{X}, \mathcal{U}, T, h)$$

Q-iteration searches for the **optimal action-value function**.

The optimal value function is then used to compute an optimal policy.

Q-iteration

The algorithm steps are:

Initialization. Select a guess $Q_i = Q_0$

Value improvement. $\forall x \in \mathcal{X}$ and $u \in \mathcal{U}$

$$Q_{i+1}(x,u) = \sum_{x^{(k+1)}} T\left(x^{(k)}, u^{(k)}\right)\left[h\left(x^{(k)}, u^{(k)}, x^{(k+1)}\right) + \gamma \max_{u^{(k+1)}} Q\left(x^{(k+1)}, u^{(k+1)}\right)\right]$$

Terminal condition. Value improvement is repeated until $Q_{i+1} = Q_i$.

The stopping criterion can only be satisfied asymptotically.

Q-iteration

It is also important to guarantee the performance when the algorithm is stopped after a finite number of iterations. However, we can guarantee that $||Q_\pi - Q^*|| \leq \xi$ if the value improvement is repeated for a total of L steps:

$$L = \left\lceil \log_\gamma \frac{\xi (1-\gamma)^2}{2||h||_\infty} \right\rceil$$

where $\lceil \cdot \rceil$ is the smallest integer larger than or equal to the argument

Q-iteration

It is also important to guarantee the performance when the algorithm is stopped after a finite number of iterations. However, we can guarantee that $||Q_\pi - Q^*|| \leq \xi$ if the value improvement is repeated for a total of L steps:

$$L = \left\lceil \log_\gamma \frac{\xi (1-\gamma)^2}{2||h||_\infty} \right\rceil$$

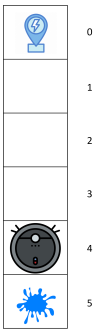
where $\lceil \cdot \rceil$ is the smallest integer larger than or equal to the argument

An **optimal policy** can be computed from Q_π^* applying

$$\pi^* = \arg\max_\pi Q_\pi^*$$

Example

Consider a robot vacuum cleaner that needs to clean a patch on the floor and also needs to recharge the batteries. Set $\gamma = 0.5$.



- **State:** $x \in \mathcal{X} = \{0, 1, 2, 3, 4, 5\}$
- **Control input:** $u \in \mathcal{U} = \{-1, 1\}$
- **State transition function:**
  $Pr\left(x^{k+1} = x^k + u^k | x^k, u^k\right) = 0.8$
  $Pr\left(x^{k+1} = x^k | x^k, u^k\right) = 0.15$
  $Pr\left(x^{k+1} = x^k - u^k | x^k, u^k\right) = 0.05$
  if $x^k = 0$ or $x^k = 5$ $x^{k+1} = x^k$
  (exercise: create a transition matrix)
- **Reward function:**
  5 if $x^{(k)} \neq 5$ and $x^{(k+1)} = 5$
  1 if $x^{(k)} \neq 0$ and $x^{(k+1)} = 0$
  0 otherwise

Example

We would like to solve the problem with Q-iteration algoritm.

Initialization.

$Q_0 =$

0	**1**	**2**	**3**	**4**	**5**
$0;0$	$0;0$	$0;0$	$0;0$	$0;0$	$0;0$

Example

4 Markov Decision Problem

We would like to solve the problem with Q-iteration algoritm.

Initialization.

$$Q_0 =$$

0	**1**	**2**	**3**	**4**	**5**
0; 0	0; 0	0; 0	0; 0	0; 0	0; 0

Value improvement. $\forall x \in \mathcal{X}$ and $u \in \mathcal{U}$

$$Q_{i+1}(x,u) = \sum_{x^{(k+1)}} T\left(x^{(k)}, u^{(k)}\right)\left[h\left(x^{(k)}, u^{(k)}, x^{(k+1)}\right) + \gamma \max_{u^{(k+1)}} Q\left(x^{(k+1)}, u^{(k+1)}\right)\right]$$

Try to do it!

Example

4 Markov Decision Problem

We would like to solve the problem with Q-iteration algoritm.

Initialization.

$Q_0 =$

0	**1**	**2**	**3**	**4**	**5**
0; 0	0; 0	0; 0	0; 0	0; 0	0; 0

Value improvement. $\forall x \in \mathcal{X}$ and $u \in \mathcal{U}$

$$Q_{i+1}(x,u) = \sum_{x^{(k+1)}} T\left(x^{(k)}, u^{(k)}\right)\left[h\left(x^{(k)}, u^{(k)}, x^{(k+1)}\right) + \gamma \max_{u^{(k+1)}} Q\left(x^{(k+1)}, u^{(k+1)}\right)\right]$$

Try to do it!

Terminal condition $\rightarrow \quad Q_{22} = Q_\pi^*$

Policy iteration

Consider the optimal control problem of finding:

$$\pi^* \left(\boldsymbol{x}^{(k)} \right) = \arg\max_{u} \; Q_\pi^* \left(\boldsymbol{x}^{(k)}, u \right)$$

$$\text{s.t.} \qquad (\mathcal{X}, \mathcal{U}, T, h)$$

Policy iteration evaluates **controllers** by constructing their **action-value functions** (instead of the optimal action-value function), and **uses these action-value functions to find new, improved controllers**.

Policy iteration

The algorithm steps are:

Initialization. Select a guess $\pi_i = \pi_0$

Policy evaluation (PE). Determine the value of the current policy

- Select a guess $Q_j = Q_0$
- Repeat until $Q_{j+1} = Q_j$

  $\forall \boldsymbol{x}^{(k)} \in \mathcal{X}$

  $$Q_{j+1}\left(\boldsymbol{x}^{(k)}, \pi_i\left(\boldsymbol{x}^{(k)}\right)\right) = \sum_{\boldsymbol{x}^{(k+1)}} T\left(\boldsymbol{x}^{(k)}, \pi_i\left(\boldsymbol{x}^{(k)}\right)\right)\left[h\left(\boldsymbol{x}^{(k)}, \pi_i\left(\boldsymbol{x}^{(k)}\right), \boldsymbol{x}^{(k+1)}\right) + \gamma \max_{u^{(k+1)}} Q_j\left(\boldsymbol{x}^{(k+1)}, \pi_i\left(\boldsymbol{x}^{(k+1)}\right)\right)\right]$$
- $Q_{\pi_i} = Q_{j+1} = Q_j$

Policy improvement (PI). Determine an improved policy

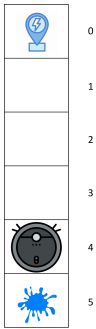
$$\pi_{i+1} = \arg\max_{\pi} Q_{\pi_i}$$

Terminal condition. PE and PI are repeated until $\pi_{i+1} = \pi_i$.

Example

4 Markov Decision Problem

Try to solve the previously defined control problem involving a robot vacuum cleaner with policy iteration.



Home excercise

Questions' time!

UNIVERSITÀ
DEGLI STUDI
DI TRIESTE