

Reinforcement Learning and Control

Felipe José Bravo Márquez

January 26, 2023

Introduction

- In supervised learning, algorithms try to make their outputs mimic the labels y given in the training set ¹.
- The labels give an unambiguous “right answer” for each of the inputs x .
- In contrast, for many sequential decision making and control problems, it is very difficult to provide this type of explicit supervision to a learning algorithm.
- For example, if we have just built a four-legged robot and are trying to program it to walk.
- Initially we have no idea what the “correct” actions to take are to make it walk
- Hence, we don't know how to provide explicit supervision for a learning algorithm to try to mimic.

¹These slides are based on [Ng, 2000].

Introduction

- In the reinforcement learning framework, we provide our algorithms only a reward function.
- This function indicates to the learning agent when it is doing well, and when it is doing poorly.
- In the four-legged walking example, the reward function might give the robot positive rewards for moving forwards, and negative rewards for either moving backwards or falling over.
- It will then be the learning algorithm's job to figure out how to choose actions over time so as to obtain large rewards.

Introduction

- Reinforcement learning has been successful in applications as diverse as:
 - 1 autonomous helicopter flight
 - 2 robot legged locomotion
 - 3 cell-phone network routing
 - 4 marketing strategy selection
 - 5 factory control
 - 6 efficient web-page indexing
- Our study of reinforcement learning will begin with a definition of the Markov decision processes (MDPs).
- MDPs provide the formalism in which RL problems are usually posed.

Markov Decision Process (MDP)

- A Markov Decision Process is a tuple:

$$(S, A, \{P_{SA}\}, \gamma, R)$$

where

- S is a set of states. (For example, in autonomous helicopter flight, S might be the set of all possible positions and orientations of the helicopter.)
- A is a set of actions. (For example, the set of all possible directions in which you can push the helicopter's control sticks.)

Markov Decision Process (MDP)

- P_{sa} are the state transition probabilities. For each state $s \in S$ and action $a \in A$, P_{sa} is a distribution over the state space, i.e., it gives the distribution over what states we will transition to if we take action a in state s .

$$\sum_{s'} P_{sa}(s') = 1, \quad P_{sa}(s') \geq 0$$

- $\gamma \in [0, 1)$ is a discount factor.
- $R : S \rightarrow \mathcal{R}$ is a reward function.

Markov Decision Process (MDP)

The dynamics of an MDP proceeds as follows:

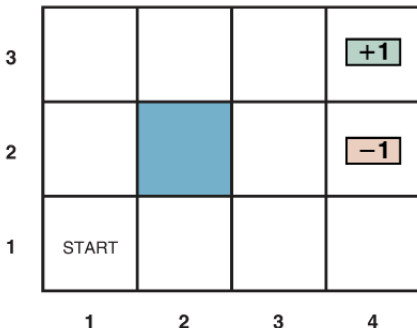
- We start in some state s_0 , and get to choose some action $a_0 \in A$ to take in the MDP.
- As a result of our choice, the state of the MDP randomly transitions to some successor state s_1 , drawn according to $s_1 \sim P_{s_0 a_0}$.
- Then, we get to pick another action a_1 .
- As a result of this action, the state transitions again, now to some $s_2 \sim P_{s_1 a_1}$.
- We then pick a_2 , and so on. . . .

Pictorially, we can represent this process as follows:

$$s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} s_2 \xrightarrow{a_2} s_3 \xrightarrow{a_3} \dots$$

Example

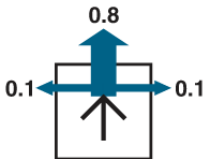
- Suppose that an agent is situated in the 4×3 environment as shown in the Figure below:



- Beginning in the start state, it must choose an action at each time step.
- There 11 states (S) and four actions $A = \{N, S, E, W\}$.
- The interaction with the environment terminates when the agent reaches one of the goal states, marked +1 or -1.

Example

- The “intended” outcome occurs with probability 0.8, but with probability 0.2 the agent moves at right angles to the intended direction:



- For example:

$$P_{(3,1)\mathcal{N}}((3,2)) = 0.8$$

$$P_{(3,1)\mathcal{N}}((4,1)) = 0.1$$

$$P_{(3,1)\mathcal{N}}((2,1)) = 0.1$$

$$P_{(3,1)\mathcal{N}}((3,3)) = 0$$

... etc

Total Payoff

- Upon visiting the sequence of states s_0, s_1, \dots with actions a_0, a_1, \dots , our **total payoff** is given by:

$$R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \dots$$

$$0 \leq \gamma < 1$$

- Our goal in reinforcement learning is to choose actions over time so as to maximize the expected value of the total payoff:

$$E[R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \dots]$$

- Note that the reward at timestep t is discounted by a factor of γ^t .
- Thus, to make this expectation large, we would like to accrue positive rewards as soon as possible (and postpone negative rewards as long as possible).
- In economic applications where $R(\cdot)$ is the amount of money made, γ also has a natural interpretation in terms of the interest rate (where a dollar today is worth more than a dollar tomorrow).

Example

- A collision with a wall results in no movement.
- Transitions into the two terminal states have reward +1 and -1, respectively.
 $R((4, 3)) = +1$
 $R((4, 2)) = -1$
 $R(S) = -0.02$ for all other states.
- All other transitions have a reward of -0.02 (to avoid the robot wasting time).

References I



Ng, A. (2000).

Cs229 lecture notes.

CS229 Lecture notes, 1(1):1–3.