Reinforcement Learning and Control

Felipe José Bravo Márquez

January 4, 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.

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

Introduction

- In the reinforcement learning framework, we povide 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 os 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 probabilites. 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)

- P_{sa} are the state transition probabilites. 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.
- $R : S \to \mathcal{R}$ is a reward function.

Markov Decision Process (MDP)

The dynamics of an MDP proceeds as follows:

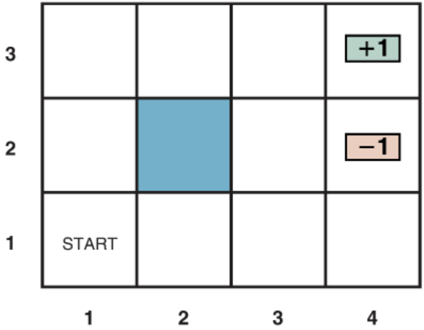
- We start in some state $s0$, and get to choose some action $a0 \in A$ to take in the MDP.
- As a result of our choice, the state of the MDP randomly transitions to some successor state $s1$, drawn according to $s1 \sim P_{s0a0}$.
- Then, we get to pick another action $a1$.
- As a result of this action, the state transitions again, now to some $s2 \sim P_{s1a1}$.
- We then pick $a2$, and so on. . . .

Pictorially, we can represent this process as follows:

$$so \xrightarrow{a0} s1 \xrightarrow{a1} s2 \xrightarrow{a2} s3 \xrightarrow{a3} ..$$

Example

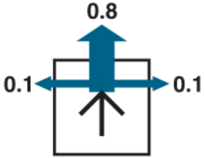
- Suppose that an agent is situated in the 4×3 environment shown in the Figure



- Beginning in the start state, it must choose an action at each time step.
- 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:



- A collision with a wall results in no movement.
- Transitions into the two terminal states have reward +1 and –1, respectively.
- 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.