Reinforcement learning

Linus Falk

March 21, 2023

Contents

1 Introduction

Machine learning can typically be separated into three branches:

- Supervised learning
- Unsupervised learning
- Reinforcement learning

Reinforcement learning is different from the two other branches in the way that it is not supervised with labeled data, but it doesn't find patterns in data completely on its own as in unsupervised learning. Instead there is a reward signal, and the feedback can be delayed, so time come in as a factor as well. The actions of the so called **agent** affect the subsequent data it receives.

The agent-environment interface

In our reinforcement learning setup we have the **agent** which is the decision maker and the **environment**, which can be seen as everything else in the setup. The goal is to maximize the cumulative reward. The agents task is to learn this from experience.

The agent observes the environments state and takes some sort of action, the agent then receives a rewards from the action and the environments state changes. The agent must now observe the new state and take a new action, and so on. Note: in reinforcement learning the environment, agent and/or reward may be stochastic.

Probability theory - review

Lets consider two random variables $X \in \chi$ and $Y \in \gamma$

- Probability: the probability that we will observe $x \in \chi$ is written as:

$$\Pr\{X = x\} \tag{1}$$

- Conditional probability: If we have already observed $y \in \gamma$ then the probability that we will observe $x \in \chi$ is written:

$$\Pr\{X = x|Y = y\} \tag{2}$$

- Probability functions:

$$\mathrm{p(x)} = \Pr\{X = x\}, \mathrm{p}(x|y) = \Pr\{X = x|Y = y\} \tag{3}$$

- Probabilities sum to 1:

$$\sum_{x\in\chi} p(x) = 1 \text{ and } \sum_{x\in\chi} p(x|y) = 1 \tag{4}$$

- The expected value:

$$\mathrm{E}[X] = \sum_{x\in\chi} xp(x), E[X|Y = y] = \sum_{x\in\chi} xp(x|y) \tag{5}$$

What is a state?

The state S_t is a representation of the environment at time t. The state space $\mathcal{S}$ is a set of all possible states. S_t will be dependent on what happened in the past, before time t. S_t contains all information that is relevant for the prediction of the future at time t.

The Markov property

$$p(S_{t+1}|S_0, A_0, S_1, A_1, \ldots, S_t, A_t) = p(S_{t+1}|S_t, A_t) \tag{6}$$

Example: The taxi environment

- Taxi: 25 different positions possible
- Passenger: 5 positions including picked up
- Destinations: 4 options
- In total 25 x 5 x 4 = 500 configurations

We can enumerate all possible configurations in some way, let $\mathcal{S} = \{0, 1, \ldots, 499\}$. This is a finite state space.

Example: Inverted pendulum

When balancing a stick, the control signal or action is the torque at the bottom. The state?

- $S_t = \theta_t$

No! we don't have any information about the direction the stick is moving in.

- $S_t = \begin{bmatrix} \theta_t \\ \frac{d\theta_t}{dt} \end{bmatrix} \in \mathcal{S} \subset \mathbb{R}^2$

This is an example of a continuous state space, Infinitely many possible states.

What is the reward

The reward R_t is a scalar signal that tells how it is doing at that time step t. The goal is the maximize the cumulative reward over **long-time**.

In the taxi example could we for example see a reward of -10 if a illegal pick-up or drop-off occurred, a successful drop-off would score +20 and all other actions would score -1 in reward. So in other words, to maximize the total reward we should deliver passenger in as few steps as possible.

In the inverted pendulum example could the goal be to keep the angle as close to zero as possible, perhaps: $R_{t+1} = -\theta_t^2$ and if we also want to use a low torque we could try: $R_{t+1} = -c_1\theta_t^2 - c_2 a_t^2$. With $\theta_t = 0$ and $a_t = 0$ we would get the maximum reward $R_{t+1} = 0$

Sequential decision making

The goal is to select actions that maximize the future reward. So actions may have long term consequences and reward may be delayed. It can also be the case that sacrificing immediate reward can gain more long-term reward. Examples:

- A financial investment,
- Fueling a car, prevent fuel stop later

Designing the reward functions

In this course we will often receive the reward functions. However it is important to note that the design of the reward function is important since it will determine what the agent tries to achieve. In some cases the agent might also find unintended ways to increase the reward and the reward influences **how** the agent learns.

an example would be the mountain car with the goal of reaching the top/the flag with as few steps as possible. The reward is -1 for each step until the flag is reached. This is a sparse rewards, all action looks equally bad until the flag is reached the first time. It could be possible to use a more informative reward function, but it is important to make sure that the agent still optimize the correct thing we want.

What is an action

An action A_t is the way an agent can affect the state a time t and $\mathcal{A}$ is the set of all possible actions. In the taxi example would an action be for example: go north, go west and pick-up and in the pendulum example could it be: torque.

Stochastic environments

A deterministic environment is one which the outcome of an action is determined by the current state and the actions taken. While in the stochastic environment there is some randomness and the outcome of an action is not completely determined by the current state. This means that the same action in the same state might have different outcomes at different times. This means that it is harder for the agent to learn an optimal policy.

The power of feedback

If no feedback is given we can pre-compute all actions $A_0, A_1, ...$ if the first state S_0 is given. While with feedback we decide the action A_t after we have observed the state at time t, S_t. Example could be to walk with or without blindfold. So instead of trying to find good actions we want to try to find a good policy.

What is a policy?

Policy is a distribution over actions given states:

$$\pi(a|s) = \Pr\{A_t = a|S_t = s\} \tag{7}$$

A policy defines how the agent will behave in different states. If we have a deterministic policy we can sometimes write:

$$a = \pi(s) \tag{8}$$

Example: The linear quadratic regulator

$$S_t = FS_t + GA_t + W_t \tag{9}$$

If we want to maximize the expected value

$$E\left[\sum_{t=0}^{\infty}(-c_1 S_t^T S_t - c_2 A_t^T A_t)\right] \tag{10}$$

Then the **Optimal policy:** when we observe s_t, choose the action:

$$a_t = \pi(s_t) = -Ls_t \tag{11}$$

for some fixed L, that we can find if F and G are know.

Terminology

Reinforcement learning	**Optimal control**
Environment	System / plant
State, S_t	State, x(t)
Action, A_t	Input, u(t)
Reward	Cost, c(t)
Policy	Controller
Maximize reward	Minimize cost
Learn policy from experience	Use model to find controller (LQC or MPC)

Table 1: example

Exploration vs exploitation

The agent should be able to learn a good policy without losing to much reward. **Exploration** is used to learn more about the environment and **exploitation** is the use of information to maximize the reward. We usually have both explore and exploit.

Model vs model-free

In model based reinforcement learning we learn a model from experience and uses the model to find a good policy and/or predictions. While in model-free learning we learn a policy and/or predictions without first learning the model. In this course the main focus will be on model-free reinforcement learning.