Reinforcement Learning

Kristian Wichmann

January 27, 2018

1 Basics of reinforcement learning

Reinforcement learning deals with an *agent* making *decisions* over time in dealing with some *environment*, to maximize some cumulative, scalar *reward*.

An example would be playing a video game. Here, the player is the agent. The control inputs are the decisions. The game and its internal logic is the environment. And the score is the reward.

Sometimes, a reward will be greater if it is postponed - it is not always advantagegous to reap any immediate reward. In other words, reinforcement learning algorithms will not benefit from being greedy.

1.1 The reward hypothesis

The basis of all reinforcement learning is the *reward hypothesis*, which states:

> All goals can be described by the maximization of some cumulative, expected reward.

1.2 Observation, action, and reward

At at given timestep t, a reinforcement learning agent gets some input; an *observation* O_t about the environment. It then takes an *action* A_t. And finally it gets a scalar *reward* R_t.

1.3 History and state

At any given time step t, the agent has a *history* H_t, which simply consist of all the observations, actions, and rewards that have happened so far:

$$H_t = O_1, A_1, R_1, O_2, A_2, R_2, \cdots, O_t, A_t, R_t, \tag{1.1}$$

A *state* is a way for to parse this history into a more meaningful form. Formally, the state representation is simply a function of the history:

$$S_t = f(H_t) \tag{1.2}$$

It is very important to note, that this is generally distinct from the *environment state* H_t^e. The environment state contains complete information about the environment, including data and mechanisms that may be hidden to the agent. Hence H_t^e can depend on other things than just the historySuch information is *private* to the environment.

The *agent state* S_t^a on the other hand is the internal representation the agent uses to decide on which actions to take. Once again, it can be any function of history:

$$S_t^a = f(H_t) \tag{1.3}$$

1.4 Markov states

A state S_{t+1} is called a *Markov state* or an *information state* if it only depends on the state of the previous time step. Expressed probabilistically:

$$\mathbb{P}[S_{t+1}|S_1, S_2, \cdots S_t] = \mathbb{P}[S_{t+1}|S_t] \tag{1.4}$$

In other words, we don't need the entire history to decide on an action: Knowing the present is enough. Or put another way:

> The future is independent of the past, given the present.

Or:

> The state is a *sufficient statistic* of the future.

The environment state is always Markov, as by definition it contains all information about what can happen next. Similarly, the state consisting of the entire history is trivially Markov as well.

1.5 Full observability

This is the case where, in fact, we can observe everything about the environment and its inner workings. So that:

$$O_t = S_t^a = S_t^e \tag{1.5}$$

Sometimes this is reasonable. Sometimes not. But it will be a useful theoretical situation. This is known as a *Markov decision process* or MDP for short.

When this condition is not fulfilled, we speak about *partial observability* or a *partially observable evironment*. Here the agent only indirectly observes the invironment. This situation is known as a *partially observable Markov decision process* or POMDP for short.

1.6 State example: Bayesian beliefs

This state representation can be seen as a current best bet at what the actual environment state is. In other words, it is represented by Bayesian probabilities, which may then be updated over time using Bayes' rule:

$$S_t^a = (\mathbb{P}[S_t^e = s_1], \mathbb{P}[S_t^e = s_2], \cdots, \mathbb{P}[S_t^e = s_n]) \tag{1.6}$$

1.7 State example: Recurrent neural net