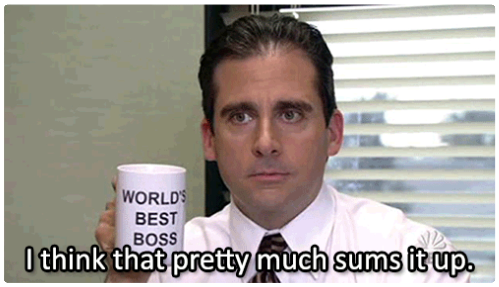
Reinforcement Learning: a comprehensive introduction [Part 1]

Jun 11, 2018 12:00 · 1875 words · 9 minute read

REINFORCEMENT LEARNING MACHINE LEARNING AI

Recap



In the previous post we introduced:

- states, $\{S_t\}_{t=1}^T$;
- actions, $\{A_t\}_{t=1}^T$;
- rewards, $\{R_{t+1}\}_{t=1}^T$.

We remarked that states and rewards are environment-related random variables: the agent has no way to interfere with the reward mechanism or modify the state transition resulting as a consequence of one of its actions.
Actions are the only domain entirely under the responsibility of the agent - specifying the probability distribution of A_t conditioned on all the possible values of $S_t,\ A_{t-1},\ \ldots, S_1$ for every $t \in \mathbb{N}$ is exactly equivalent to a full specification of the agent behaviour - we shall take a closer look at the issue in this post.

In other words, we are going to formally introduce the concept of **policy**.

A bit of notation

The history of our system is the sequence up to time t, h_t, is nothing else than the sequence of the so-far observed states and actions, i.e.

$$h_t := (s_1, a_1, s_2, a_2, \dots, a_{t-1}, s_t) \tag{1}$$

where $s_i \in \mathcal{S}$ and $a_i \in \mathcal{A}(s_i)$ for all $i \in \{1, \dots, t\}$.
We will use H_t (capital H) to denote history as a random variable.

To simplify we shall assume, from now on, that the action space does not depend on the current state of the system - it does not make a substantial difference, but it allows the use of a cleaner notation.
This implies that h_t belongs to

$$\mathcal{H}_1 := \mathcal{S} \tag{2}$$

$$\mathcal{H}_t := \mathcal{S} \times \mathcal{A} \times \dots \times \mathcal{S} = \mathcal{S} \times \prod_{t-1 \text{ times}} (\mathcal{A} \times \mathcal{S}) \qquad \text{if } t \in \{2, 3, \dots\} \tag{3}$$

or, using a little bit of recursion,

$$\begin{aligned} \mathcal{H}_1 &= \mathcal{S} \\ \mathcal{H}_t &= \mathcal{H}_{t-1} \times \mathcal{A} \times \mathcal{S} \qquad \text{if } t \in \{2, 3, \dots\} \end{aligned} \tag{4}$$

We can thus denote the history of our system up to time t as

$$h_t = (h_{t-1}, a_{t-1}, s_t) \tag{5}$$

This will turn out to be a useful notation in upcoming computations.

Decision rules

As we anticipated in the recap, we want to come up with a flexible way to specify (and study)

$$\mathbb{P}(A_t = a_t \mid S_t = s_t, A_{t-1} = a_{t-1}, \dots, S_1 = s_1) \tag{6}$$

Using our brand-new history notation we can rewrite that expression as

$$\mathbb{P}(A_t = a_t \mid H_t = h_t) \tag{7}$$

which is nothing else that the probability of our agent performing action a_t at time t knowing what has happened so far in terms of states and actions.
We can formalize it as a function d_t from $\mathcal{H}_t$ to $\mathcal{P}(\mathcal{A})$, the space of probability distributions over $\mathcal{A}$:

$$d_t : \mathcal{H}_t \to \mathcal{P}(\mathcal{A}) \tag{8}$$

We shall denote by $b_{d_t}(a \mid h_t)$ or $b_{d_t(h_t)}(a)$ the probability of taking action a if our system history up to time t is equal to h_t, i.e.

$$\mathbb{P}(A_t = a_t \mid H_t = h_t) = b_{d_t(h_t)}(a_t) \tag{9}$$

d_t is usually called **decision rule** because it models how the agent reacts according to its current simulation experience.

Policies

A **policy** is the *recipe* underlying the behaviour of our agent - what it is programmed to do under the set of circumstances it is currently experiencing. In other words, how the agent is going to act at each time instant $t \in \{1, \dots\}$.
Decision rules provide a quick and painless way to formally define it: a policy π is nothing more than a **sequence** of decision rules, i.e.

$$\pi = (d_1, d_2, d_3, \dots) \tag{10}$$

with $d_i : \mathcal{H}_i \to \mathcal{P}(\mathcal{A})$ for each $i \in \{1, 2, \dots\}$.

Now that we a proper mathematical definition of policy we can give a rigorous formulation of the **reward hypothesis**, which we have already stated in the previous post:

> That all of what we mean by goals and purposes can be well thought of as maximization of the expected value of the cumulative sum of a received scalar signal (called reward).

In particular, we can shed some light on the expression "*expected value of the cumulative sum of a received scalar signal*".

Returns

How do we measure the performance of a policy?
In a Reinforcement Learning context, every action is followed by a reward (positive or negative). To measure the overall performance of a policy we need to combine all these rewards together - this is usually called **return** in the literature.

The most common reward criteria is the one mentioned in the reward hypothesis: **cumulative return**.
If R_{t+1} denotes the reward obtained after action A_t we can define the **cumulative return** at time t, G_t, as

$$G_t := \sum_{k=t+1}^{T} R_k \tag{11}$$

where T is the random variable denoting the end time of our simulation (e.g. the total number of moves in a chess game).

This definition is perfectly fine if T is almost surely finite, but it becomes problematic for those simulations in which it makes sense to potentially have $T = +\infty$ - an exploration game, for example.
The issue can be promptly solved introducing a **discount rate** $\gamma \in [0, 1)$ which can be used to define the **discounted cumulative return**:

$$G_t := \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k \tag{12}$$

$\gamma \in [0, 1)$ allows us to associate a bigger weight to closer rewards as well as ensuring that the overall return G_t remains finite if our rewards are bounded:

$$|G_t| \leq \sum_{k=t+1}^{+\infty} \left|\gamma^{k-t-1} R_k\right| = \sum_{k=t+1}^{+\infty} \gamma^{k-t-1} \, |R_k| \leq M \sum_{k=t+1}^{+\infty} \gamma^{k-t-1} = \frac{M}{1-\gamma} < +\infty \tag{13}$$

assuming that there exists $M > 0$ such that $|R_t| \leq M$ for all t.
We will implicitly assume that if $\gamma = 1$ then $T \neq +\infty$, and vice versa.

State-value and history-value functions

Let:

- $\pi = (d_1, d_2, \ldots)$ be the policy followed by our agent;
- G_t be the discounted cumulative return from time t onwards;
- s_1 be the initial environment state.

Then “*expected value of the (discounted) cumulative sum of a received scalar signal*” is

$$v_\pi^1(s_1) := \mathbb{E}_\pi\left[G_1 \mid S_1 = s_1\right] \tag{14}$$

which is usually called **state-value function** for the policy π.
The fancy “E”, $\mathbb{E}$, stands for the probabilistic expectation of a random variable - look it up if this is the first time you encounter it (link).
The π subscript, instead, indicates that this expectation is computed under the hypothesis that the agent is following the policy π. What does this mean, mathematically?

Well, let’s unroll the definition!

Expected reward

First of all we’ll start by computing the expected value of the first reward, R_2, under policy π:

$$\mathbb{E}_\pi\left[R_2 \mid S_1 = s_1\right] \tag{15}$$

The expectation of a random variable is nothing more than the sum of all its possible values weighted by their probability of being observed. Formally:

$$\mathbb{E}_\pi\left[R_2 \mid S_1 = s_1\right] = \sum_{r\in\mathcal{R}} r\,\mathbb{P}(R_2 = r \mid S_1 = s_1) \tag{16}$$

where $\mathcal{R}$ is the set of possible rewards (which we assume to be finite or countable, in order to avoid integrals).

Using the law of total probability (link) we can decompose those probabilities with respect to the set of possible actions available to the agent ($\mathcal{A}$):

$$\mathbb{P}(R_2 = r \mid S_1 = s_1) = \sum_{a\in\mathcal{A}} \mathbb{P}(R_2 = r \mid A_1 = a, S_1 = s_1)\,\mathbb{P}(A_1 = a \mid S_1 = s_1) \tag{17}$$

But $\mathbb{P}(A_1 = a \mid S_1 = s_1)$ is fully determined by our agent’s first decision rule - d_1:

$$\mathbb{P}(A_1 = a \mid S_1 = s_1) = b_{d_1(s_1)}(a) \tag{18}$$

Thus, going back to equation 16, we get:

$$\mathbb{E}_\pi\left[R_2 \mid S_1 = s_1\right] = \sum_{r\in\mathcal{R}} r \sum_{a\in\mathcal{A}} \mathbb{P}(R_2 = r \mid A_1 = a, S_1 = s_1)\,b_{d_1(s_1)}(a) \tag{19}$$

which, with a simple reordering, becomes

$$\mathbb{E}_\pi\left[R_2 \mid S_1 = s_1\right] = \sum_{a\in\mathcal{A}} b_{d_1(s_1)}(a)\left(\sum_{r\in\mathcal{R}} r\,\mathbb{P}(R_2 = r \mid A_1 = a, S_1 = s_1)\right) \tag{20}$$

Using again the definition of mathematical expectation we get

$$\mathbb{E}_\pi\left[R_2 \mid S_1 = s_1\right] = \sum_{a\in\mathcal{A}} \underbrace{b_{d_1(s_1)}(a)}_{\text{agent}}\,\underbrace{\mathbb{E}(R_2 \mid A_1 = a, S_1 = s_1)}_{\text{environment}} \tag{21}$$

You can clearly see the different contributions in equation 21 - the policy determines the probability to choose a particular action while the environment is responsible for the expected return awaiting the agent if it were to choose that specific course of action. In fact there is no π subscript under the last expectation!

Expected return

We can now derive an explicit formula for the expected discounted return under policy π.

First of all, expectation is linear - i.e.

$$\mathbb{E}[\alpha A + \beta B] = \alpha\mathbb{E}[A] + \beta\mathbb{E}[B] \tag{22}$$

where A and B are random variables with finite expectation and α and β are real numbers.

Then:

$$\begin{aligned} v_\pi^1(s_1) &= \mathbb{E}_\pi\left[G_1 \mid S_1 = s_1\right] = && (23) \\ &= \mathbb{E}_\pi\left[R_2 + \gamma G_2 \mid S_1 = s_1\right] = && (24) \\ &= \underbrace{\mathbb{E}_\pi\left[R_2 \mid S_1 = s_1\right]}_{\triangle} + \gamma \underbrace{\mathbb{E}_\pi\left[G_2 \mid S_1 = s_1\right]}_{\square} && (25) \end{aligned}$$

We have already derived a formula for $\triangle$ - let's focus on $\square$.
Using again the law of total probability and the definition of expected value we get:

$$\mathbb{E}_\pi\left[G_2 \mid S_1 = s_1\right] = \sum_g g \sum_{a \in \mathcal{A}} \mathbb{P}(G_2 = g \mid A_1 = a, S_1 = s_1)\, b_{d_1(s_1)}(a) \tag{26}$$

But we can do it again - this time decomposing with respect to S_2:

$$\begin{aligned} \mathbb{P}(G_2 = g \mid A_1 = a, S_1 = s_1) &= \sum_{s \in \mathcal{S}} \mathbb{P}(G_2 = g \mid S_2 = s, A_1 = a, S_1 = s_1)\, \mathbb{P}(S_2 = s \mid A_1 = a, S_1 = s_1) = \qquad (2 \\ &= \sum_{s \in \mathcal{S}} \mathbb{P}\Big(G_2 = g \mid H_2 = (s_1, a, s)\Big)\, \mathbb{P}(S_2 = s \mid A_1 = a, S_1 = s_1) \qquad (2 \end{aligned}$$

Putting everything together we get:

$$\begin{aligned} \mathbb{E}_\pi\left[G_2 \mid S_1 = s_1\right] &= \sum_g g \sum_{a \in \mathcal{A}} \sum_{s \in \mathcal{S}} \mathbb{P}\Big(G_2 = g \mid H_2 = (s_1, a_1, s)\Big)\, b_{d_1(s_1)}(a)\, \mathbb{P}(S_2 = s \mid A_1 = a, S_1 = s_1) = \\ &= \sum_{a \in \mathcal{A}} \sum_{s \in \mathcal{S}} \left[\sum_g g\, \mathbb{P}\Big(G_2 = g \mid H_2 = (s_1, a_1, s)\Big)\right] b_{d_1(s_1)}(a)\, \mathbb{P}(S_2 = s \mid A_1 = a, S_1 = s_1) = \\ &= \sum_{a \in \mathcal{A}} \sum_{s \in \mathcal{S}} \underbrace{\mathbb{E}_\pi\Big[G_2 \mid H_2 = (s_1, a_1, s)\Big]}_{\star} \underbrace{b_{d_1(s_1)}(a)}_{\text{agent}} \underbrace{\mathbb{P}(S_2 = s \mid A_1 = a, S_1 = s_1)}_{\text{environment}} \end{aligned}$$

Let's take a step back and look at what we got here: $\star$ is not that different from the state-value function we have defined in equation 14: the only difference is that we are computing the expected discounted cumulative return starting from $t = 2$ instead of $t = 1$, as well as conditioning on H_2 instead of $S_1 = H_1$.

We can formalize this intuition introducing the **history-value function for policy π at time t**:

$$v_\pi^t(h_t) := \mathbb{E}_\pi\left[G_t \mid H_t = h_t\right] \tag{32}$$

Plugging this new object into our previous computations we obtain:

$$\mathbb{E}_\pi\left[G_2 \mid S_1 = s_1\right] = \sum_{a \in \mathcal{A}} \sum_{s \in \mathcal{S}} \underbrace{v_\pi^2((s_1, a, s))}_{\star} \underbrace{b_{d_1(s_1)}(a)}_{\text{agent}} \underbrace{\mathbb{P}(S_2 = s \mid A_1 = a, S_1 = s_1)}_{\text{environment}} \tag{33}$$

Thanks to equations 25, 21 and 33 we get our general formula for the expected cumulative discounted return under policy π:

$$v_\pi^1(s_1) = \sum_{a \in \mathcal{A}} \underbrace{b_{d_1(s_1)}(a)}_{\text{agent}} \left(\underbrace{\mathbb{E}[R_2 \mid A_1 = a, S_1 = s_1]}_{\text{environment}} + \gamma \sum_{s \in \mathcal{S}} v_\pi^2\Big((s_1, a, s)\Big) \underbrace{\mathbb{P}(S_2 = s \mid A_1 = a, S_1 = s_1)}_{\text{environment}} \right) \qquad (34$$

This is a generalized version of what is usually called **Bellman equation** in the framework of Markov Decision Processes.

Bellman equation

Using the same techniques we have employed in the previous two subsections we can prove that the following equation holds for every $t \in \{1, 2, \ldots\}$:

$$v_\pi^t(h_t) = \sum_{a\in\mathcal{A}} b_{d_t(h_t)}(a) \left(\mathbb{E}[R_{t+1} \mid A_t = a, H_t = h_t] + \gamma \sum_{s\in\mathcal{S}} v_\pi^{t+1}\left((h_t, a, s)\right) \mathbb{P}(S_{t+1} = s \mid A_t = a, H_t = h_t)\right)$$

You can clearly spot the issue here: to compute v_π^1 we need to know v_π^2, which in turn requires $v_\pi^3 \dots$ there is no end in sight, assuming that T is potentially infinite.
Things would simplify significantly if we were to assume that the environment and the agent are **markovian** and **stationary** - S_t and R_t depend only on A_{t-1} and S_{t-1}, while the agent bases its A_t choice only on the value of S_{t-1}, without taking into account the current time instant t - i.e. $\pi = (d_1, d_1, \dots)$.
With these further hypotheses it can be proved (and we will, in a later episode) that equation 34 becomes:

$$v_\pi^1(s_1) = \sum_{a\in\mathcal{A}} b_{d_1(s_1)}(a) \left(\mathbb{E}[R_2 \mid A_1 = a, S_1 = s_1] + \gamma \sum_{s\in\mathcal{S}} v_\pi^1(s_1)\, \mathbb{P}(S_2 = s \mid A_1 = a, S_1 = s_1)\right) \quad (36)$$

A closed relation!

We can actually craft an algorithm to find v_π^1 and we can use its value to compare different policies.
A policy value-function, in fact, is the expected cumulative discounted reward following that strategy: if $v_\tau^1(s) \geq v_\pi^1(s)$ for every $s \in \mathcal{S}$ we can generally conclude that τ is a **better** policy than π.
We might then ask ourselves if there exists an **optimal** policy - a policy whose expected cumulative discounted return is greater or equal than the one associated with any other policy. Does it exists? Can we find it in the subset of markovian stationary policy? Is there a practical algorithm to do so?

These, and many other related questions, will be the main topic of our next episode.

Reinforcement Learning series index

- Part 0
- Part 1 *(this post)*
- Part 2
- *Coming soon…*

tweet Share

0 Comments **lpalmieri** **Arefe**

Recommend Share Sort by Best

Start the discussion…

Be the first to comment.

Subscribe Add Disqus to your siteAdd DisqusAdd Disqus' Privacy PolicyPrivacy PolicyPrivacy

Powered by Hugo Theme By nodejh