Exercise Sheet 2: Temporal-Difference learning

Younesse Kaddar

1. Temporal-Difference learning with discounting

In many instances, immediate rewards are worth more than those in the future. To take this observation into account, the value $V(s_t)$ of a particular state s_t is not the sum all of future rewards, but rather the sum of all future *discounted* rewards.

$$V(s_t) \overset{\text{def}}{=} \sum_{\tau=0}^{+\infty} \gamma^\tau r(s_{t+\tau})$$

where $0 < \gamma < 1$. Here, s_t is the state at time t, *i.e.* the state in which the agent is right now, s_{t+1} the state that the agent will move to next, and so on.

Following the derivation in the lecture, show that the temporal-difference-learning rule in this case is given by:

$$V(s_t) \to V(s_t) + \varepsilon\Big(r(s_t) + \gamma V(s_{t+1}) - V(s_t)\Big)$$

Reminder: with $\gamma = 1$

With the notations used in the lecture: in compliance with the δ-rule, we came up with the following TD-learning rule with $\gamma = 1$ (no discounting):

$$\hat{V}(s_t) \to \hat{V}(s_t) + \varepsilon\delta$$

where

- $\hat{V}(s_t)$ is the internal estimate of value (denoted by $V(s_t)$ in the problem statement, by abuse of notation)
- $0 < \varepsilon << 1$ is a learning rate
- δ is the prediction error

The most "natural" prediction error is the difference between the actual value the state $V(s_t)$ and the internal estimate thereof:

$$\delta_{natural} \stackrel{\text{def}}{=} V(s_t) - \hat{V}(s_t)$$

As explained in class: the problem is that the real value of the state $V(s_t)$ is *not* known, so one is unable to compute such a prediction error! The trick we used to cope with that was to notice that:

$$\begin{aligned}\delta_{natural} &\stackrel{\text{def}}{=} \overbrace{V(s_t)}^{= r(s_t)+V(s_{t+1})} - \hat{V}(s_t) \\ &= r(s_t) + V(s_{t+1}) - \hat{V}(s_t)\end{aligned}$$

so that we could use the current internal estimate $\hat{V}(s_{t+1})$ of $V(s_{t+1})$, as in *dynamic programming*, to approximate:

$$\delta_{natural} = r(s_t) + V(s_{t+1}) - \hat{V}(s_t) \simeq r(s_t) + \hat{V}(s_{t+1}) - \hat{V}(s_t)$$

And we set δ to be this approximate (and now computable!) value:

$$\delta \stackrel{\text{def}}{=} r(s_t) + \hat{V}(s_{t+1}) - \hat{V}(s_t)$$

For $0 < \gamma < 1$

In the discounting case, likewise:

$$\begin{aligned}\delta_{natural} &\stackrel{\text{def}}{=} V(s_t) - \hat{V}(s_t) \\ &= \sum_{\tau=0}^{+\infty} \gamma^\tau r(s_{t+\tau}) - \hat{V}(s_t) \\ &= r(s_t) + \sum_{\tau=1}^{+\infty} \gamma^\tau r(s_{t+\tau}) - \hat{V}(s_t) \\ &= r(s_t) + \sum_{\tau=0}^{+\infty} \gamma^{\tau+1} r(s_{t+\tau+1}) - \hat{V}(s_t) \\ &= r(s_t) + \gamma \underbrace{\sum_{\tau=0}^{+\infty} \gamma^\tau r(s_{(t+1)\tau})}_{\stackrel{\text{def}}{=} V(s_{t+1})} - \hat{V}(s_t) \\ &= r(s_t) + \gamma \underbrace{V(s_{t+1})}_{\simeq \hat{V}(s_{t+1})} - \hat{V}(s_t)\end{aligned}$$

And again, by approximating $V(s_{t+1})$ by $\hat{V}(s_{t+1})$, one sets δ to be:

$$\delta \stackrel{\text{def}}{=} r(s_t) + \gamma\hat{V}(s_{t+1}) - \hat{V}(s_t)$$

Therefore, the overall TD-learning rule is, in the discounting case:

$$\hat{V}(s_t) \to \hat{V}(s_t) + \varepsilon\delta = \hat{V}(s_t) + \varepsilon\Big(r(s_t) + \gamma\hat{V}(s_{t+1}) - \hat{V}(s_t)\Big)$$

or, with the abuse of notation ($\hat{V}$ is denoted by V as well) made in the problem statement:

$$V(s_t) \to V(s_t) + \varepsilon\Big(r(s_t) + \gamma V(s_{t+1}) - V(s_t)\Big)$$

2. Models for the value function

In the lecture, we talked about the necessity to introduce models for the value of a state, so that one could properly generalize to new, unseen situations. One very simple model is given by the value function $V(\mathbf{u}) = \mathbf{w} \cdot \mathbf{u}$ where $\mathbf{u}$ is a vector of stimuli that could either be present (1) or absent (0).

a)

Take the example of two stimuli, $\mathbf{u} = (u_1, u_2)$. Let's assume that the subject (agent) has already learned the value of a state in which the first (resp. second) stimulus is present: $V(\mathbf{u} = (1, 0)) \stackrel{\text{def}}{=} \alpha$ (resp. $V(\mathbf{u} = (0, 1)) \stackrel{\text{def}}{=} \beta$).

What are the values of the parameters $\mathbf{w} = (w_1, w_2)$ that the agent has learnt?

$$\begin{aligned} w_1 &= w_1 \times 1 + w_2 \times 0 \\ &= \mathbf{w} \cdot (1, 0) \\ &= V((1, 0)) \\ &= \alpha \end{aligned}$$

and analogously

$$\begin{aligned} w_2 &= w_1 \times 0 + w_2 \times 1 \\ &= \mathbf{w} \cdot (0, 1) \\ &= V((0, 1)) \\ &= \beta \end{aligned}$$

Therefore:

$$\mathbf{w} = (\alpha, \beta)$$

Assuming that the agent, for the very first time, runs into a state in which both stimuli are present: what is the value of this state?

If both stimuli are present, then $u_1 = u_2 = 1$, and:

$$V(\mathbf{u} = (1, 1)) = \mathbf{w} \cdot (1, 1) = w_1 \times 1 + w_2 \times 1 = \alpha + \beta$$

What if we now add some uncertainty: what would be the value of a state where the first stimulus has 50% chance of being present and the second stimulus has 10%?

In this case: $u_1 = 0.5$ and $u_2 = 0.1$, so that:

$$V(\mathbf{u} = (0.5, 0.1)) = \mathbf{w} \cdot (0.5, 0.1) = w_1 \times 0.5 + w_2 \times 0.1 = \frac{\alpha}{2} + \frac{\beta}{10}$$

In what situation do you think this sort of a more generalized model that you just came up with would not make much sense?

The model is more generalized in that the domain of V is no longer $\{0, 1\}^2$ but now $[0, 1]^2$.

But it remains a **linear model**, which is a rather significant constraint: the stimuli u_1 and u_2 constribute independently to the inner estimate of value, and they do so in a linear fashion.

A linear model is not suited for all situations: for example, the mere idea that there is a slight chance of having food might cause a dog to relish the prospect of getting food and make it salivate as much as if there were actually food.

b) *Advanced*: derive the temporal-difference learning rule for the parameter $\mathbf{w}$ that needs to be learned if the value function is $V(\mathbf{u}) = \mathbf{w} \cdot \mathbf{u}$. *Hint*: Start from a loss function - what would be a suitable choice?

Let's denote by $\mathbf{u}_t \stackrel{\text{def}}{=} (u_{t,1}, u_{t,2})$ the stimuli vector at time t, and again by $\hat{V}$ the inner estimate of value (as in the first answer).

As for the Rescola-Wagner rule, the squared loss seems to be a suitable choice in this case:

$$L_t \stackrel{\text{def}}{=} (V(\mathbf{u}_t) - \underbrace{\hat{V}(\mathbf{u}_t)}_{\stackrel{\text{def}}{=} \mathbf{w} \cdot \mathbf{u}_t})^2$$

Then, steps to update $\mathbf{w}$ are taken along the opposite of the gradient, i.e. toward a local minimum of the loss (gradient descent algorithm), which leads to the following learning rule:

$$\mathbf{w} \to \mathbf{w} - \varepsilon' \nabla_{\mathbf{w}} L_t$$

where the gradient

$$\nabla_{\mathbf{w}} L_t \stackrel{\text{def}}{=} -2(V(\mathbf{u}_t) - \mathbf{w} \cdot \mathbf{u}_t)\, \mathbf{u}_t$$

The problem is that the agent doesn't know V, but, as shown in question 1, we can approximate it as follows:

$$V(\mathbf{u}_t) \simeq r(\mathbf{u}_t) + \gamma \underbrace{\hat{V}(\mathbf{u}_{t+1})}_{\stackrel{\text{def}}{=} \mathbf{w} \cdot \mathbf{u}_{t+1}}$$

So that $\nabla_{\mathbf{w}} L_t$ becomes:

$$\nabla_{\mathbf{w}} L_t \simeq -2\big(r(\mathbf{u}_t) + \mathbf{w} \cdot (\gamma \mathbf{u}_{t+1} - \mathbf{u}_t)\big)\, \mathbf{u}_t$$

By setting $\varepsilon \stackrel{\text{def}}{=} \frac{\varepsilon'}{2}$, it follows that:

> *the temporal-difference learning rule for the parameter* $\mathbf{w}$ *if* $\hat{V}(\mathbf{u}) = \mathbf{w} \cdot \mathbf{u}$ *is:*
>
> $$\mathbf{w} \to \mathbf{w} + \varepsilon\big(r(\mathbf{u}_t) + \mathbf{w} \cdot (\gamma \mathbf{u}_{t+1} - \mathbf{u}_t)\big)\, \mathbf{u}_t$$

Can you derive a learning rule if the value function were given by $V(\mathbf{u}) = f(\mathbf{w} \cdot \mathbf{u})$, where f is a known (non-linear) function?

Assuming that f is differentiable: the only difference with the previous answer is that now:

$$L_t \stackrel{\text{def}}{=} (V(\mathbf{u}_t) - f(\mathbf{w} \cdot \mathbf{u}_t))^2$$

So that:

$$\nabla_{\mathbf{w}} L_t \stackrel{\text{def}}{=} -2(V(\mathbf{u}_t) - \mathbf{w} \cdot \mathbf{u}_t) f'(\mathbf{w} \cdot \mathbf{u}_t)\, \mathbf{u}_t$$

as result of which:

> *the temporal-difference learning rule for the parameter* $\mathbf{w}$ *if* $\hat{V}(\mathbf{u}) = f(\mathbf{w} \cdot \mathbf{u})$*, where* f *is differentiable, is:*
>
> $$\mathbf{w} \to \mathbf{w} + \varepsilon\big(r(\mathbf{u}_t) + \mathbf{w} \cdot (\gamma \mathbf{u}_{t+1} - \mathbf{u}_t)\big) f'(\mathbf{w} \cdot \mathbf{u}_t)\, \mathbf{u}_t$$