outline

Caden Hewlett

2024-02-29

An Introduction to Q-Learning.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \big[r_{t+1} + \gamma \max_{a \in A} Q(s_{t+1}, a) - Q(s_t, a_t) \big]$$

Where R_{t+1} is the reward at state $t+1$.

It can be considered as the negative loss function.

$$R_{t+1} = R(a_t, s_t) = -\mathcal{L}(a_t, s_t)$$

Notably, we have hyper-parameters α and γ. These are set by the user; $\alpha \in [0, 1]$ is the **learning rate**, essentially to what extent newly acquired information overrides old information.

Secondly, we have $\gamma \in [0, 1]$, which is known as the **discount factor**. A γ of 0 makes the agent short-sighted by only considering current rewards, while a γ closer to 1 will make it strive for long-term high rewards.

Then, there are what are called **action selection** methods. There are three that I am familiar with, but I would likely investigate two to start.

The first is ε-greedy. In this method, the agent's decision at time t is to either "exploit" or "explore." When the agent exploits, it selects the action corresponding to the state-action pair at time t that maximizes the reward based on the current information. In other words, under exploitation,

$$a_t \mid s_t = \underset{a \in A}{\operatorname{argmax}} \{ Q(s_t, a) \}$$

And, under exploration, the agent randomly selects an option available to it.

$$a_t \mid s_t = a_t \sim \text{unif}\big(a_1, a_2, \dots, a_{|A|} \}\big)$$

This is notably *independent* of s_t, however, one could have an adjusted model (for maze exploration as an example) where only legal actions are selected. In most problems (such as the multi-armed bandit) this isn't strictly necessary.

Then, at each step t, a Bernoulli trial is conducted to see if the agent will explore or exploit. Let d_t be the decision at time t in this regard.

$$d_t \sim \text{bern}(\varepsilon)$$

However, ε is often a user-defined hyper-parameter. So I would like to implement a Bayesian model where it is Beta-Distributed random variable.

Further, both α and γ are hyper-parameters in $[0, 1]$ so I want to investigate the potential of a Beta prior on them, too.

So the full model would look something like this:

$$\begin{aligned}
\alpha &\sim \text{beta}(\mu_\alpha, s_\alpha) \\
\gamma &\sim \text{beta}(\mu_\gamma, s_\gamma) \\
\varepsilon &\sim \text{beta}(\mu_\varepsilon, s_\varepsilon) \\
d_t &\sim \text{bern}(\varepsilon) \\
Q(s_t, a_t) &\leftarrow Q(s_t, a_t) + \alpha\big[r_{t+1} + \gamma \max_{a \in A} Q(s_{t+1}, a) - Q(s_t, a_t)\big] \\
a_{\,t+1} \mid \{s_t, d_t\} &\sim \begin{cases} \text{unif}\big(a_1, a_2, \ldots, a_{|A|}\}\big), & d_t = 1 \\ \underset{a \in A}{\text{argmax}}\{Q(s_{\,t+1}, a)\}, & d_t = 0 \end{cases}
\end{aligned}$$

And we'd do a similar implementation for SoftMax where temperature $\tau \in \mathbb{R}^+$ likely takes an exponential prior.

The final object out of the training episodes is the Q table.

I'd like to see how more (or less) performant this is than a situation where we train on constant hyper-parameters. I'd do in-sample metrics (convergence time, consistency, etc.) and out-of-sample (i.e. transferability of $\alpha, \epsilon, \gamma, Q$ to other similar problem domains.)