CSC 449 Advanced Topics in Artificial Intelligence

Deep Reinforcement Learning
Exam 2
Fall, 2022

Alex Hanson

1. Consider the following pseudo-code for a faulty SARSA algorithm:

This seems to be an attempt at SARSA(λ) there is a few errors present.

- Remove the if statement checking for "if The current state is terminal". Change it to "if The current state is NOT terminal". Move the $y_t =$ up to new if statement. This is because setting the value of y to zero doesn't appear to do anything.
- Change value y_t is set to. Change "$max_a Q(s_{t+1}, a)$" to "$Q(s', a')$" doing this to make sure the Q update later includes both s' and a' otherwise missing key elements of the SARSA algorithm.
- Change "Update Q function:" to "For all s,a". We want to be running the Q update over the entire state space. This is to take advantage of saving the traces.
- Because we are saving the traces there needs to be a trace decay which is missing. We can add this trace decay in the "For all s,a" loop. The decay will look something like: $n(s_t, a_t) \leftarrow \gamma\lambda n(s_t, a_t)$. Since there does not seem to be a γ present in the algorithm we can assume that it is intended to be set to one meaning we can drop the γ term from the trace decay update.
- Lastly, the Q update needs to be adjusted. The SARSA algorithm has 5 parts that need to be included s, a, r, s', a'. The Current Q update does not include s' and a' in the update. A more standard SARSA Q update looks like: $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)]$. In the Q update we have it is close but we need to adjust where the update is taking place. Want to update the state the we came from not the state we are arriving in. Updating the state we are arriving using info about where we came from does not help with getting to s' again. The Q update also does not take advantage of the traces we are saving. To take advantage of the traces we simply add a multiply of the trace

value to the end of the current Q update. One more thing to change with the Q update is to fix the subtraction to an addition.

Updated algorithm

procedure SARSA (number of episodes $N \in \mathbb{N}$, discount factor $\lambda \in (0, 1]$, learning rate $\alpha_n = \frac{1}{\log(n+1)}$) may not converge. Use 1/n

> Initialize matrices $Q(s, a)$ and $n(s, a)$ to 0
> **for** episode $k \in 1, 2, 3, \ldots, n$ **do**
>
> > $t \leftarrow 1$
> > Initialize s_1
> > Choose a_1 from a uniform distribution over the actions
> > **while** Episode k is not finished **do**
> >
> > > Take action a_t and observe r_t and s_{t+1}
> > > **if** The current state is not terminal **then**
> > >
> > > > $y_t = r_t + Q(s', a')$
>
> > > **end if**
> > > $n(s_t, a_t) \leftarrow n(s_t, a_t) + 1$
> > > **for** all s, a **do**
> > >
> > > > $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha_{n(s_t, a_t)}(y_t - Q(s_t, a_t))n(s_t, a_t)$
> > > > $n(s_t, a_t) \leftarrow \lambda n(s_t, a_t)$
>
> > > **end for**
>
> > **end while**
>
> **end for**

end procedure

---

2. Variant of SARSA:
    1. What sequence of policies (π_t) should you choose so that the corresponding variant of SARSA is on-policy? This variant is called Expected SARSA.

       To meet the requirements of being on-policy both the behavior policy and the target policy need to be the same. This means that when selecting actions using the policy it should be something like ϵ greedy with respect to Q. Even when calculating the expected value the policy needs to be the same policy.
    2. Consider an off-policy variant of SARSA corresponding to a stationary policy $\pi = \pi_t \forall t$. Under this algorithm, do the Q values converge? If so, what are the limiting Q values? Justify your answer.

       This seems like one of the trick questions because its a yes and no answer. It really depends on how the policy π is interpreted. For q_π under a given policy π to converge it needs to be able to visit every state and take every action an infinite number of times. Assuming policy π allows for continual exploration then $q_\pi \approx q_*$ over infinite episodes. Meaning it will converge. On the other hand if π is deterministic then following π does not allow for continual exploration meaning that q_π may converge on a policy but there is no guaranties that it will converge on q_*.