CSC 449 Advanced Topics in Artificial Intelligence

Deep Reinforcement Learning

Exam 2

Fall, 2022

100

Name: Jonathan Mathews ID#: 7110031 Score: _____

Your solutions to these problems should be uploaded to D2L as a single pdf file by the deadline. You may turn in the solution up to two days late, with a penalty of 10% per day, and you should only upload one version of your solutions.

This exam is individual and open book. You may consult any reference work. If you make specific use of a reference outside those on the course web page in solving a problem, include a citation to that reference.

You may discuss the course material in general with other students, but you must work on the solutions to the problems on your own.

It is difficult to write questions in which every possibility is taken into account. As a result, there may sometimes be "trick" answers that are simple and avoid addressing the intended problem. Such trick answers will not receive credit. As an example, suppose we said, use the chain rule to compute $\frac{\partial z}{\partial x}$ with $z = \frac{7}{y}$ and $y = x^2$. A trick answer would be to say that the partial deriviative is not well defined because y might equal 0. A correct answer might note this, but would then give the correct partial derivative when $y \neq 0$.

1. (40 pts) Consider the following pseudo-code for a faulty SARSA algorithm:

procedure SARSA(number of episodes ~~$N \in \mathbb{N}$~~ $n \in \mathbb{N}$
discount factor ~~$\lambda \in (0,1]$~~ $\gamma \in (0,1]$
learning rate $\alpha_n = \frac{1}{\log(n+1)}$)

$\alpha_1 = 1.44$ which is greater than 1. $\alpha_n, n > 1$ is okay

Initialize matrices $Q(s,a)$ and $n(s,a)$ to $0, \forall s, a$
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 ← exploring starts. There are other ways too.
while Episode k is not finished **do**
Take action a_t: observe reward r_t and next state s_{t+1}
Choose a_{t+1} from s_{t+1} using μ_t: an ε-greedy policy with respect to Q
if The current state is terminal **then** ▷ *Compute target value*
$y_t = 0$
else
$y_t = r_t + \gamma Q(s_{t+1}, a_{t+1})$
~~$y_t = r_t + \max_a Q(s_{t+1}, a)$~~
end if
$n(s_t, a_t) \leftarrow n(s_t, a_t) + 1$
Update Q function:
$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha_{n(s_t,a_t)}(y_t - Q(s_t, a_t))$
~~$Q(s_{t+1}, a_{t+1}) \leftarrow Q(s_t, a_t) - \alpha_{n(s_t,a_t)}(y_t - Q(s_t, a_t))$~~
$t \leftarrow t + 1$
end while
end for
end procedure

Find all of the mistakes in the algorithm. Explain why they are mistakes, and correct them.

See following pages

2. (60 pts) Your friend found a variant of SARSA which is defined through a sequence of policies π_t (where $t \geq 1$), and consists of just changing (in the previous algorithm **after corrections**) the way the target is computed. The target becomes

$$y_t = r_t + \gamma \sum_a \pi_t(a|s_{t+1}) Q(S_{t+1}, a),$$

where $\pi_t(a|s)$ is the probability that a is selected in state s under policy π_t.

a) 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.

See following pages

b) 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.

See following pages

Question 1

Here are my comments on the pseudo-code to accompany the marked up page of pseudo-code

1. The discount factor should be γ, not λ.
2. The learning rate is greater than 1 for $n < 1.72$ which means that the first update will be weighted very heavily. In general, a learning rate greater than 1 is not okay, but in this case it might be okay because it's less than 1 for every step after the first step.
3. The learning rate is unique for every state-action pair, and decays every time a state is visited. This is a very strange way to do this and I can't find any references that say it's a good idea. That being said, I can't prove that it won't work either so I haven't marked it as a mistake.
4. The algorithm uses exploring starts, which is not the only way to pick an initial action, but should be fine.
5. The policy is indicated with μ_t but usually we use π_t
6. The equation for y_t should be $y_t = r_t + \gamma Q(s_{t+1}, a_{t+1})$ instead of $y_t = r_t + \max_a Q(s_{t+1}, a)$ because we are doing SARSA, not Q-learning.
7. The update equation should be updating $Q(s_t, a_t)$ not $Q(s_{t+1}, a_{t+1})$.
8. The update equation should have a plus sign, not a minus sign.

Question 2

Your friend found a variant of SARSA which is defined through a sequence of policies π_t (where $t \geq 1$), and consists of just changing (in the previous algorithm **after corrections**) the way the target is computed. The target becomes:

$$y_t = r_t + \gamma \sum_a \pi_t(a|s_{t+1}) Q(s_{t+1}, a)$$

where $\pi_t(a|s)$ is the probability that a is selected in state s under policy π_t.

a) Question

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.

a) Answer

In order to have the variant be on-policy, we must choose a sequence of policies such that the behavioral policy is the same as the target policy at each time step, t. The behavioral policy is the policy used to select the next action, a'. The target policy is the policy used to select the action used to update $Q(s, a)$. The target policy can be taken from the equation given above:

$$y_t = r_t + \gamma \sum_a \pi_t(a|s_{t+1}) Q(s_{t+1}, a)$$

The target policy is thus a stochastic policy, $\pi_t(a|s)$. In order for this to be on-policy, the behavioral policy must also be $\pi_t(a|s)$. The question states that the only thing changed from the previous algorithm was "the way the target is computed" which implies that the behavioral policy is still ϵ-greedy from the previous question. ϵ-greedy is a stochastic policy, so it would work.

Each time step, $Q_t(s, a)$ is updated to $Q_{t+1}(s, a)$ and the next policy in the sequence, π_{t+1} is computed from the updated Q. Thus the sequence of policies we should choose is the sequence of epsilon greedy policies created from the sequence of changing values of Q.

b) Question

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.

b) Answer

The Q values should converge to something, but they may not converge to the optimum. The Q values will be limited to those states explored by the stationary policy. If the stationary policy doesn't try the optimum action in each state, then there's no way for the Q values to converge to the optimum.

I'm assuming that the stationary policy defines all the probabilities of all the actions in all states, rather than being something like a greedy or ϵ-greedy policy. Those policies can't be stationary because they change depending on Q.

In the case where the target policy is greedy, and the stationary policy explores every state, the described algorithm is Q-learning, which does converge to the optimum.