CSC 449 Advanced Topics in Artificial Intelligence

Deep Reinforcement Learning

Exam 2

Fall, 2022

Name: David Mathews ID#: 7472735 Score: 85

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 $y \neq 0$.

25 pts

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

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

Set Epsilon to some small value > 0.

use 1/n to guarantee convergence

Initialize matrices $Q(s,a)$ ~~and $n(s,a)$~~ to $0, \forall s,a$

This array shouldn't exist.

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

Uniform?

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*

$$y_t = 0$$

Assuming this is to show the value of a terminal state is 0.

ok

else

$$y_t = r_{t+1} + \lambda Q(s_{t+1}, a_{t+1})$$

$$\cancel{y_t = r_t + \max_a Q(s_{t+1}, a)}$$

Q Learning Algorithm, Should be SARSA Algorithm

end if
$\cancel{n(s_t,a_t) \leftarrow n(s_t,a_t)+1}$

Shouldn't need to update anything related to the number of episodes. This is just wrong.

Update Q function:

+

$$Q(s_{\cancel{t+1}}, a_{\cancel{t+1}}) \leftarrow Q(s_t, a_t) - \alpha_{n(s_t,a_t)}(y_t - Q(s_t,a_t))$$

Should update current S and A based on future S and A.

$t \leftarrow t+1$
end while
end for
end procedure

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

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

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

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.

$\{\pi_1, \pi_2, \pi_3, \ldots \pi_n\}$

As long as the policy learned is the policy used, Expected SARSA is on-policy.

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.

Yes, under a fixed policy, the Q values converge. Because the ratio of selecting a given action A does not change under a fixed policy, the Q values will eventually converge to some constant. Because the terminal state has no future states or actions to update from, every Q leading to a terminal state will limit and set the values for the converging Q's.

ok, but could be better.