CSC 449 Advanced Topics in Artificial Intelligence

Deep Reinforcement Learning

Exam 2

Fall, 2022

Name: Noah Zikmund ID#: 7469562 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 $y \neq 0$.

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 $\alpha_n = \frac{1}{\log(n+1)}$)

$\frac{1}{\log(n+1)}$ will be greater than 1 for $n = 1 \ldots$ & should be $\frac{1}{\log_2(n+1)}$ where there is a case for α if $n = 0$ defined.

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
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

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

end if
$n(s_t,a_t) \leftarrow n(s_t,a_t) + 1$
Update Q function:

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

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

Assuming we want our SARSA to be on-policy, we should not choose actions from a uniform distribution because it would result in an off-policy SARSA. Instead a_1 should be chosen based off μ_t, a ε-greedy policy with respect to Q

missed 3 errors

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

The algorithm above resembles a SARSA(λ). Assuming that it is, the eligibility trace update is missing with lambda:

$$n(s_t, a_t) \leftarrow \lambda n(s_t, a_t)$$

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.

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.

a) To ensure on-policy SARSA, the action that is selected by π_t should be the highest value action in the state

So the sequence of policies (π_t) should be greedy in the sense that the highest-value action is always selected.

b) Given a stationary policy, the answer to "Do the Q-values converge?" is it depends. If the stationary policy does NOT bring us to the terminal state, the Q-values will not converge and approach $-\infty$. If the stationary policy DOES reach the terminal state, the Q-values will converge. The Q-values that limit the convergence of the other Q-values is the terminal state. The terminal state is a constant, and all other Q-value's values are dependent upon the terminal state's value.

Note: Because the policy is stationary, there will be Q-values that are never updated because their states are never visited by the policy. Their Q-values remain the same as they were at initialization.