Reinforcement Learning Exam 2

100

Submitted by: Anoushka Mathews

Answer 1:

These are following errors with the pseudo-code SARSA algorithm.

a) Discount factor is usually denoted by the letter γ, gamma, instead of λ, lambda. λ is mostly used to denote eligibility trace constant.
b) Learning rate, α_n, should have values bounded within $(0, 1]$ to cope with policy stochasticity. The function $1/(\log(n+1))$ is not confined within the bounds. The learning rate, α_n, can be variable for different episodes. It is a good idea to have larger learning rate for slower frequencies and smaller learning rate for faster frequencies. Putting a hard bound on α could work.
c) We need to assign a value to $\varepsilon > 0$ for ε-greedy policy.
d) $\mathrm{n}(\mathrm{s}, \mathrm{a})$ seems to be unnecessary.
e) $t \leftarrow 1$ seems to be unnecessary.
f) After initializing s_1 or s_t, we choose a_1 or a_t using policy derived from Q (e.g., ε-greedy) instead of uniform distribution over the actions. SARSA, in its pure form, is an on-policy TD control method. This means that the target policy and the behavior policy should both be the same. Uniform distribution over the action is a bad target policy because it very likely will not converge. Since we should not use uniform distribution over the actions as a target policy, we are to not use it as a behavior policy either.
g) We should terminate the while loop only when s_1 is the terminal state. The while loop statement says, "while episode K is not finished," which is not quite accurate because it doesn't define what finishes an episode. Saying "while s is not terminal state" is clearer in that way.
h) The if statement for calculating y_t should be eliminated. It makes no difference.
i) The y_t calculation should be done using the formula
$$y_t = r_t + \gamma Q(s_{t+1}, a_{t+1})$$
Instead of using $\underset{a}{max} Q(s_{t+1}, a)$. Q-learning, off policy TD method uses the max approach.
j) Using and updating $n(s_t, a_t)$ is unnecessary.
k) Q update formula is wrong. First, we are to update the value of (s_t, a_t) and not (s_{t+1}, a_{t+1}). Second, we are to add the update and not subtract it. So, the Q update should be done using the following formula
$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha_n(s_t, a_t)[y_t - Q(s_t, a_t)]$$
l) Using and updating t is unnecessary.
m) In the end we need to update the current state and action to the next state and action.
$$s_t \leftarrow s_{t+1}$$
$$a_t \leftarrow a_{t+1}$$

Finally, the algorithm should look like this:

Procedure SARSA(number of episodes N, discount factor γ, learning rate $\alpha_n = bound(\frac{1}{\log(n+1)})$)

Initialize $Q(s, a) => 0$ for all s and all a.

Initialize small ε-greedy > 0

For each Episode $K \in [1, 2, \ldots n]$:

Choose a_t using policy derived from Q (e.g., ε-greedy) from s_t

While s_t is not terminal state

Take action a_t, get reward r_t and state s_{t+1}

Choose a_{t+1} using policy derived from Q (e.g., ε-greedy) from s_{t+1}

$y_t = r_t + \gamma Q(s_{t+1}, a_{t+1})$

$Q(s_t, a_t) = Q(s_t, a_t) + \alpha_n(s_t, a_t)[y_t - Q(s_t, a_t)]$

$s_t \leftarrow s_{t+1}$

$a_t \leftarrow a_{t+1}$

End while

End For

End Procedure

Answer 2:

Expected SARSA has the following update equation:

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

a) For an on-policy method, the target policy or estimation policy (policy that gets used in the update step) is the same as the behavior policy (policy that picks actions to take). For expected SARSA to work in an on-policy way, the target policy and the behavior policy will have to be the same policies. For instance, if the policy we choose is ε-greedy (as in the above question), then we must pick the same policy and the same value of ε for both behavior policy and target policy.

This policy, probability distribution over actions in every state, will change over time because Q will change over time.

Let's say that at time, $t = 1$, we used π_1 as our behavior policy, derived from $Q1$, to get action, a_1. Then we will use π_1 in our update step to change Q_1 to Q_2. The behavior policy will now be π_2 derived by Q_2, at timestep, $t = 2$. Then, we will use π_2 as our behavior policy and so on. This sequence looks like this:

$$Q_1 \rightarrow \pi_1(behavior) \rightarrow \pi_1(target) \rightarrow Q_2 \rightarrow \pi_2(behavior) \rightarrow \pi_2(target) \rightarrow \ldots$$

b) For an off-policy SARSA variant corresponding to a stationary policy $\pi_t = \pi \quad \forall\, t$, the Q values will converge if π, the behavior policy, allows infinite exploration. π can allow infinite exploration if it is stochastic and all the transitional probabilities are non-zero.

Expected SARSA is guaranteed to converge to the optimal value function under the following conditions (Seijen):

1) S and A are finite,
2) $\alpha_t(s_t, a_t) \in [0,1]$, $\sum_t \alpha_t(s_t, a_t) = \infty$, $\sum_t (\alpha_t(s_t, a_t))^2 < \infty$ $\forall (s, a) \neq (s_t, a_t): \alpha_t(s, a) = 0$,
3) **The policy is greedy in the limit with infinite exploration**,
4) The reward function is bounded.

So, the Q values will converge to the optimal value function, Q* given the infinite exploration condition posed on the stationary policy. Expected SARSA can be thought of as a more generalized version of Q-learning, an off-policy TD method or off-policy variant of SARSA. Q-learning converges to an optimal value function, Q*, independent of the policy being followed as long as the policy explores the optimal value function.

Citations

Seijen, Harm van, et al. *A Theoretical and Empirical Analysis of Expected Sarsa*, https://www.cs.ox.ac.uk/people/shimon.whiteson/pubs/vanseijenadprl09.pdf.