Analyzing Performance

All of the TD control algorithms we have examined (Sarsa, Sarsamax, Expected Sarsa) converge to the optimal action-value function q_* (and so yield the optimal policy π_*) if (1) the value of ϵ decays in accordance with the GLIE conditions, and (2) the step-size parameter α is sufficiently small.

The differences between these algorithms are summarized below:

- Sarsa and Expected Sarsa are both **on-policy** TD control algorithms. In this case, the same (ϵ-greedy) policy that is evaluated and improved is also used to select actions.
- Sarsamax is an **off-policy** method, where the (greedy) policy that is evaluated and improved is different from the (ϵ-greedy) policy that is used to select actions.
- On-policy TD control methods (like Expected Sarsa and Sarsa) have better online performance than off-policy TD control methods (like Sarsamax).
- Expected Sarsa generally achieves better performance than Sarsa.

If you would like to learn more, you are encouraged to read Chapter 6 of the textbook (especially sections 6.4-6.6).

As an optional exercise to deepen your understanding, you are encouraged to reproduce Figure 6.4. (Note that this exercise is optional!)

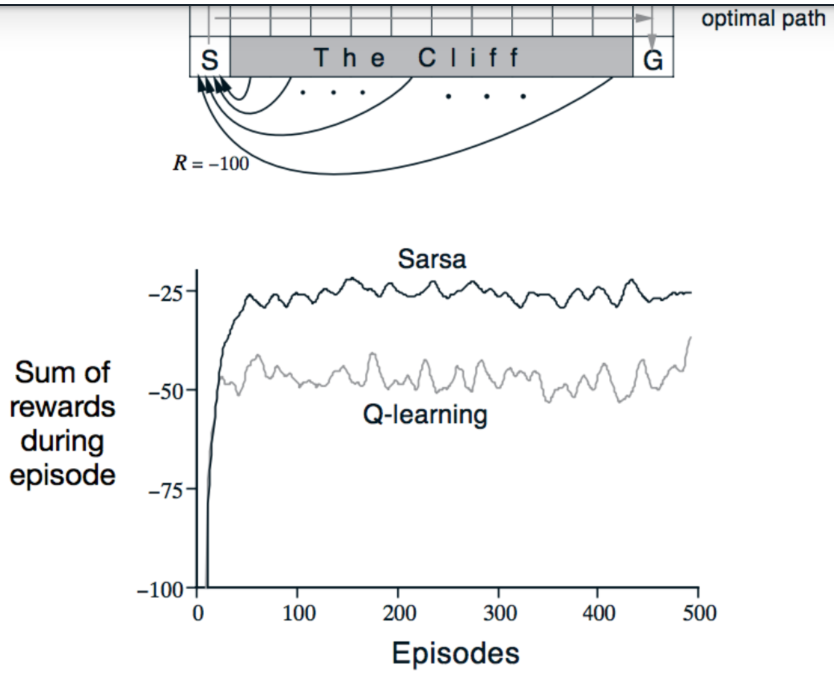

Figure 6.4: The cliff-walking task. The results are from a single run, but smoothed by averaging the reward sums from 10 successive episodes. ■

The figure shows the performance of Sarsa and Q-learning on the cliff walking environment for constant $\epsilon = 0.1$. As described in the textbook, in this case,

- Q-learning achieves worse online performance (where the agent collects less reward on average in each episode), but learns the optimal policy, and
- Sarsa achieves better online performance, but learns a sub-optimal "safe" policy.

You should be able to reproduce the figure by making only small modifications to your existing code.

NEXT