Implementation: Expected Sarsa

The pseudocode for Expected Sarsa can be found below.

TD Control: Expected Sarsa

Input: policy π, positive integer *num_episodes*, small positive fraction α
Output: value function Q ($\approx q_\pi$ if *num_episodes* is large enough)
Initialize Q arbitrarily (e.g., $Q(s,a) = 0$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}(s)$, and $Q(\textit{terminal-state}, \cdot) = 0$)
for $i \leftarrow 1$ ***to*** *num_episodes* **do**
  $\epsilon \leftarrow \frac{1}{i}$
  Observe S_0
  $t \leftarrow 0$
  repeat
    Choose action A_t using policy derived from Q (e.g., ϵ-greedy)
    Take action A_t and observe R_{t+1}, S_{t+1}
    $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha(R_{t+1} + \gamma \sum_a \pi(a|S_{t+1})Q(S_{t+1}, a) - Q(S_t, A_t))$
    $t \leftarrow t + 1$
  until S_t *is terminal*;
end
return Q

Expected Sarsa is **guaranteed to converge** under the same conditions that guarantee convergence of Sarsa and Sarsamax.

Remember that ***theoretically***, the as long as the step-size parameter α is sufficiently small, and the **Greedy in the Limit with Infinite Exploration (GLIE)** conditions are met, the agent is guaranteed to eventually discover the optimal action-value function (and an associated optimal policy). However, ***in practice***, for all of the algorithms we have discussed, it is common to completely ignore these conditions and still discover an optimal policy. You can see an example of this in the solution notebook.

Please use the next concept to complete **Part 4: TD Control: Expected Sarsa** of `Temporal_Difference.ipynb` . Remember to save your work!

If you'd like to reference the pseudocode while working on the notebook, you are encouraged to open this sheet in a new window.

Implementation

NEXT