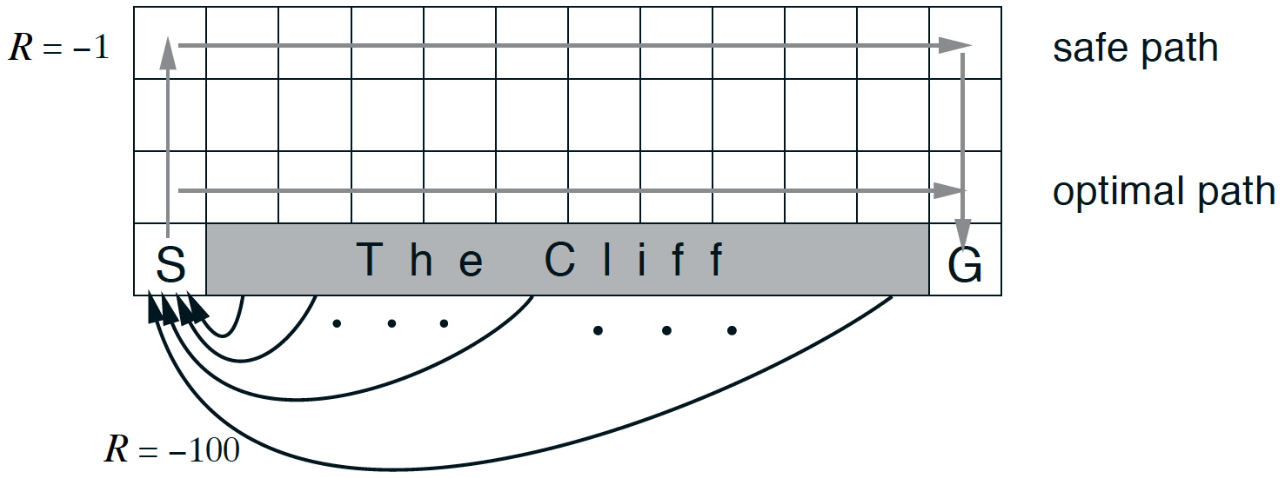
Summary



The cliff-walking task (Sutton and Barto, 2017)

TD Prediction: TD(0)

- Whereas Monte Carlo (MC) prediction methods must wait until the end of an episode to update the value function estimate, temporal-difference (TD) methods update the value function after every time step.
- For any fixed policy, **one-step TD** (or **TD(0)**) is guaranteed to converge to the true state-value function, as long as the step-size parameter α is sufficiently small.
- In practice, TD prediction converges faster than MC prediction.

Input: policy π, positive integer *num_episodes*
Output: value function V ($\approx v_\pi$ if *num_episodes* is large enough)
Initialize V arbitrarily (e.g., $V(s) = 0$ for all $s \in \mathcal{S}^+$)
for $i \leftarrow 1$ ***to*** *num_episodes* **do**
  Observe S_0
  $t \leftarrow 0$
  repeat
    Choose action A_t using policy π
    Take action A_t and observe R_{t+1}, S_{t+1}
    $V(S_t) \leftarrow V(S_t) + \alpha(R_{t+1} + \gamma V(S_{t+1}) - V(S_t))$
    $t \leftarrow t + 1$
  until S_t *is terminal*;
end
return V

TD Prediction: Action Values

- (In this concept, we discussed a TD prediction algorithm for estimating action values. Similar to TD(0), this algorithm is guaranteed to converge to the true action-value function, as long as the step-size parameter α is sufficiently small.)

TD Control: Sarsa(0)

- **Sarsa(0)** (or **Sarsa**) is an on-policy TD control method. It is guaranteed to converge to the optimal action-value function q_*, as long as the step-size parameter α is sufficiently small and ϵ is chosen to satisfy the **Greedy in the Limit with Infinite Exploration (GLIE)** conditions.

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
  Choose action A_0 using policy derived from Q (e.g., ϵ-greedy)
  $t \leftarrow 0$
  repeat
    Take action A_t and observe R_{t+1}, S_{t+1}
    Choose action A_{t+1} using policy derived from Q (e.g., ϵ-greedy)
    $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha(R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t))$
    $t \leftarrow t + 1$
  until S_t *is terminal*;
end
return Q

TD Control: Sarsamax

- **Sarsamax** (or **Q-Learning**) is an off-policy TD control method. It is guaranteed to converge to the optimal action value function q_*, under the same conditions that guarantee convergence of the Sarsa control algorithm.

TD Control: Sarsamax

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}
    $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha(R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t))$
    $t \leftarrow t + 1$
  until S_t *is terminal*;
end
return Q

TD Control: Expected Sarsa

- **Expected Sarsa** is an on-policy TD control method. It is guaranteed to converge to the optimal action value function q_*, under the same conditions that guarantee convergence of Sarsa and Sarsamax.

Summary

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 $Q(terminal\text{-}state, \cdot) = 0$)
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

Analyzing Performance

- On-policy TD control methods (like Expected Sarsa and Sarsa) have better online

NEXT