Implementation: TD(0)

The pseudocode for TD(0) (or one-step TD) can be found below.

TD Prediction: TD(0)

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(0) is **guaranteed to converge** to the true state-value function, as long as the step-size parameter α is sufficiently small. If you recall, this was also the case for constant-α MC prediction. However, TD(0) has some nice advantages:

- Whereas MC prediction must wait until the end of an episode to update the value function estimate, TD prediction methods update the value function after every time step. Similarly, TD prediction methods work for continuous and episodic tasks, while MC prediction can only be applied to episodic tasks.
- In practice, TD prediction converges faster than MC prediction. (*That said, no one has yet been able to prove this, and it remains an open problem.*) You are encouraged to take the time to check this for yourself in your implementations! For an example of how to run this kind of analysis, check out Example 6.2 in the textbook.

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.

Feel free to check your solution by looking at the corresponding sections in `Temporal_Difference_Solution.ipynb`.

NEXT