Temporal Difference Learning

Temporal difference learning is central to reinforcement learning, allowing for incremental updates to value estimates without waiting for the end of an episode.

The TD error, denoted as δ_t, represents the difference between the estimated value of a state and the value of the next state.

$$V(S_t) \leftarrow V(S_t) + \alpha[G_t - V(S_t)]$$
$$V(S_t) \leftarrow V(S_t) + \alpha[R_{t+1} + \gamma V(S_{t+1}) - V(S_t)] \leftarrow \delta_t$$

Reminder: $G_t \approx R_{t+1} + \gamma V(S_{t+1})$

Tabular TD(0) for estimating v_π

Input: the policy π to be evaluated
Algorithm parameter: step size $\alpha \in (0,1]$
Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(terminal) = 0$
Loop for each episode:
    Initialize S
    Loop for each step of the episode:
        $A \leftarrow$ action given by π for S
        Take action A, observe R, S'
        $V(S) \leftarrow V(S) + \alpha[R_{t+1} + \gamma V(S_{t+1}) - V(S_t)]$
        $S \leftarrow S'$
    Until S is terminal

The advantages of Temporal Difference

Temporal Difference combines key ideas from Monte Carlo methods and Dynamic programming together, making it able to update from the previous parameters and learn directly from experience at once.

Feature	**Monte Carlo (MC)**	**Dynamic Programming (DP)**	**Temporal Difference (TD)**
Model Requirement	Does not need a model	Requires full model	Does not need a model
When It Updates	Only after episode ends	Iteratively over full state space	Step-by-step, during episode

Use in Continuing Tasks	Not suitable	Suitable	Suitable
Works in Real-Time	No (waits for episode to finish)	No	Yes
Learning Efficiency	Slower (uses full returns)	Fast if model is accurate	Faster (bootstraps estimates)
Sample Efficiency	Less efficient	High (requires full sweeps)	More efficient
Bias/Variance	Low bias, high variance	Low variance, higher bias	Moderate bias and variance
Scalability	Moderate (with enough samples)	Poor in large state spaces	High (learns from samples)
Flexibility	Works with unknown environments	Only in known environments	Works with unknown environments
Responsiveness to Change	Slower to adapt	Depends on full environment update	Quickly adapts with each step