Temporal Difference

Temporal Difference (TD) methods are a family of algorithms used in reinforcement learning to estimate the value function of a Markov Decision Process (MDP). Unlike Monte Carlo methods that require the complete episode to be observed, TD methods can learn online, updating the value function after each step. This makes them more suitable for tasks where episodes can be very long or even infinite.

How TD Methods Work:

1. **Initialize:** Start with an arbitrary value function for each state.
2. **Experience:** Interact with the environment, observing the current state, the action taken, the next state, and the reward received.
3. **Update Value Function:** Use the observed experience to update the value function estimate for the current state.
4. **Repeat:** Continue steps 2 and 3 until convergence or a desired number of steps.

The TD Update Rule:

The core idea of TD methods is to update the value function based on the difference between the current estimate and the estimated future reward. This difference is known as the TD error.

The TD update rule for a state `s` is:

$$V(s) < -V(s) + \alpha * (R(s,a) + \gamma * V(s') - V(s))$$

where:

- `V(s)` is the current estimate of the value function for state `s`.
- `α` is the learning rate, controlling how much the value function is updated based on the TD error.
- `R(s, a)` is the reward received for taking action `a` in state `s`.
- `γ` is the discount factor, determining the importance of future rewards.
- `V(s')` is the current estimate of the value function for the next state `s'`.

TD Methods:

There are several TD methods, each with its own characteristics:

- **TD(0):** The simplest TD method, using the one-step TD error.
- **TD(λ):** A family of methods that use a combination of one-step and multi-step TD errors, controlled by the eligibility trace parameter λ.
- **Q-learning:** A TD method that estimates the Q-value function, which represents the expected future reward for taking a particular action in a particular state.

Advantages of TD Methods:

- **Online learning:** TD methods can learn from experience as it is acquired, making them suitable for tasks with long or infinite episodes.
- **Lower variance:** Compared to Monte Carlo methods, TD methods can have lower variance in their estimates, especially for tasks with high-dimensional state spaces.
- **Efficient:** TD methods can be computationally efficient, making them suitable for large-scale problems.

Applications of TD Methods:

- **Game playing:** TD methods have been successfully applied to various games, including chess, backgammon, and Go.
- **Robotics:** TD methods can be used to learn control policies for robots.
- **Natural language processing:** TD methods can be used for tasks such as machine translation and dialogue systems.

DP VS MC VS TD

Dynamic Programming (DP):

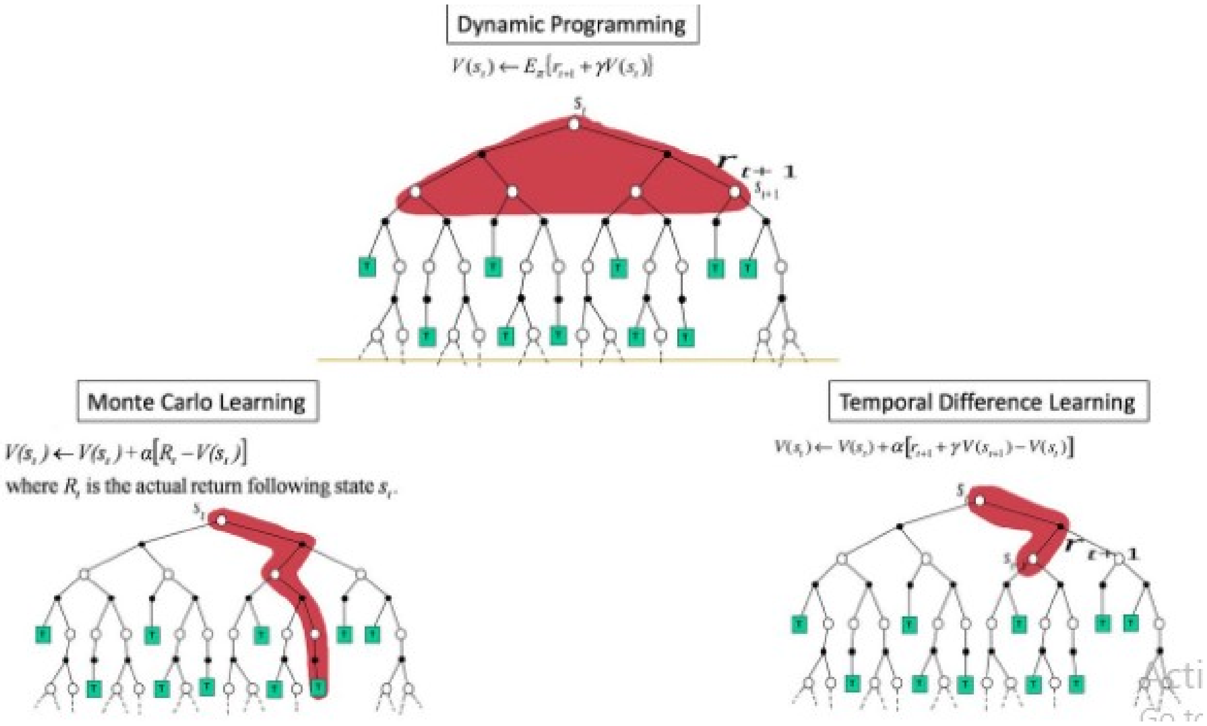
- **Core Idea:** DP assumes a complete model of the environment, including transition probabilities and rewards. It works by iteratively calculating the optimal value function for each state.

Monte Carlo (MC):

- **Core Idea:** MC methods learn from complete episodes of experience. They estimate the value function by averaging the returns obtained from multiple simulations.

Temporal Difference (TD):

- **Core Idea:** TD methods learn from incomplete episodes and update the value function after each step. They bootstrap the value estimate using the current estimate of the next state's value.



Dynamic Programming

$V(s_t) \leftarrow E_\pi[r_{t+1} + \gamma V(s_t)]$

Monte Carlo Learning

$V(s_t) \leftarrow V(s_t) + \alpha[R_t - V(s_t)]$

where R_t is the actual return following state s_t.

Temporal Difference Learning

$V(s_t) \leftarrow V(s_t) + \alpha[r_{t+1} + \gamma V(s_{t+1}) - V(s_t)]$