Reinforcement Learning

Solution Methods

Learning Objectives

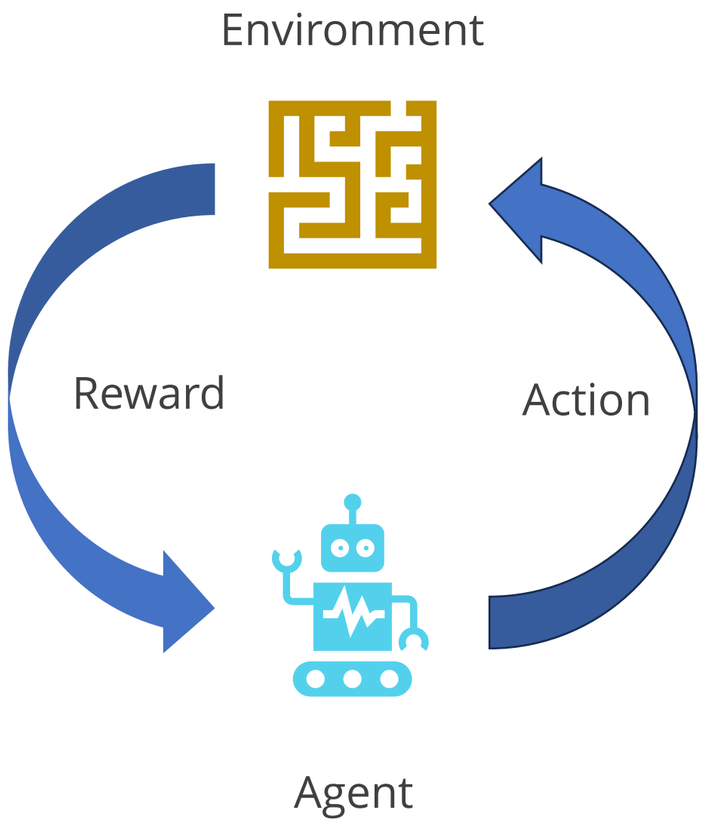
By the end of this lesson, you will be able to:

- Discuss Monte Carlo methods
- Describe temporal difference learning
- Compare the Monte Carlo methods and temporal learning difference in RL

Monte Carlo Methods

Monte Carlo Methods

Monte Carlo is a solution method to estimate the state values when there is no model to define the environment. In other words, there is no information available about transition probabilities and the rewards.



Monte Carlo methods in reinforcement learning, solve problems by learning from the complete sequences of states, actions, and rewards from episodes.

Monte Carlo Methods

Monte Carlo methods can learn directly from raw experience without a model of the environment's dynamics.

- This algorithm updates the estimates of state values based on the transitions and rewards that it observes.
- For a given policy, the value of each state is estimated by the average return following visits to that state over multiple episodes.

The more episodes and visits to the state, the more accurate the estimate.

Types of Monte Carlo Methods

Types of Monte Carlo Methods

In reinforcement learning and decision-making, different types of Monte Carlo methods are available. These include:

First Visit Monte Carlo

First-visit Monte Carlo is a method in reinforcement learning where value of a state is estimated by the average return following the first time the state was visited in each episode.

Let $\mathbf{S_t}$ be state at time t during an episode and $\mathbf{G_t}$ be the return following the first visit to $\mathbf{S_t}$

$$G_t = R_{t+1} + \gamma . R_{t+2} + \gamma^2 . R_{t+3} + \ldots\ldots\ldots + \gamma^{T-t-1} . R_T$$

Here,

- $\mathbf{R_{t+1}}$ is the reward received after taking an action at time $\boldsymbol{t}$,
- $\boldsymbol{\gamma}$ is the discount factor (between 0 and 1), and
- $\boldsymbol{T}$ is the time step of the final state in the episode.

First Visit Monte Carlo

The value of state ***s*** after n episodes is given by the equation below:

$$V(s) = \frac{1}{N(s)} \sum_{i=1}^{N(s)} G_i$$

Here,

- ***N(s)*** is the number of episodes where s is visited for the first time received after taking an action at time ***t.***
- G_i are the returns following the first visit to ***s***.

First Visit Monte Carlo: Example

Imagine a simple board game where an agent moves across different cells, and each cell has a reward or penalty.

First Visit Monte Carlo: Example

In this scenario, each game played is considered as an **episode**. In a first-visit Monte Carlo

If the agent visits a cell multiple times in one episode, only the return from the first visit to that cell is used to estimate its value.

Over many ***episodes***, the average of these first-visit returns is computed for each cell.

This computation process leads to an estimate of each cell's value under the current policy.

Every Visit Monte Carlo

Every-visit Monte Carlo is a reinforcement learning technique where value of a state is estimated by the average return following every visit to the state, not just the first.

In every visit Monte Carlo, the value of state ***s*** is the average of all the returns following visits to ***s*** over many episodes.

The update rule after episode ***i*** is:

$$V(s) = \frac{1}{N(s)} \sum_{j=1}^{N(s)} G_j$$

Here,

- ***N(s)*** is the total number of times state ***s*** has been visited across all episodes up to episode ***i***.
- G_j are the returns following each visit to ***s***.

Every Visit Monte Carlo: Example

Imagine a simple board game where an agent moves across different cells, and each cell has a reward or penalty.

Every Visit Monte Carlo: Example

In every-visit Monte Carlo, every time the agent visits a cell, the return from that visit is used to estimate the cell's value.

If the agent visits the same cell multiple times in an episode, each visit contributes to the average.

Over many ***episodes***, the average of these first-visit returns is computed for each cell.

This provides a more comprehensive estimate that accounts for the variability in returns from all visits to a state.

Temporal Difference

Temporal Difference Learning

Temporal difference (TD) learning is a type of reinforcement learning method that combines elements of both Monte Carlo methods and dynamic programming.

- RL problems with discrete actions can be represented as markov decision processes (mdps).
- Agents in these problems typically lack knowledge of transition probabilities, denoted as ***t(s, a, s')***.
- Similarly, agents are also unaware of the rewards they will receive, represented as ***r(s, a, s')***.

Temporal Difference Learning

Temporal difference (TD) learning algorithm closely resembles the value iteration algorithm.

- TD learning, however, incorporates a modification to account for the initial lack of knowledge about states and actions.
- Agents in TD learning start with limited information, not knowing all possible states and actions from the beginning.
- The algorithm adapts to the dynamic environment by updating its estimates based on observed experiences over time.

Temporal Difference Learning

The agent uses an exploration policy, and as it progress, the TD learning algorithm updates the estimates of state values based on the transitions and rewards that it observes.

- Temporal difference (TD) learning updates the value of a state s_t in the learning process.
- The update is determined by the temporal difference, which is the discrepancy between the estimated return and the actual return.
- This estimation is based on taking an action, observing the resulting state, and receiving the actual return from that transition.

The difference, is denoted as **TD error.**

Temporal Difference Learning

TD error is calculated as the temporal difference between the current estimate and the observed outcome.

- The TD learning algorithm adjusts its state value estimates using this TD error, allowing it to iteratively refine its understanding of the environment.
- This dynamic updating process helps the agent adapt to the changing and uncertain nature of the environment over time.

Temporal Difference Learning Algorithm

Like the value iteration algorithm, the temporal difference learning algorithm is represented as below:

$$V_{k+1}(s) \leftarrow (1-\alpha)V_k(s) + \alpha\left(r + \gamma . V_k(s')\right)$$

The above equation can also be represented as:

With,

$$V_{k+1}(s) \leftarrow V_k(s) + \alpha . \delta_k(s, r, s')$$

$$\delta_k(s, r, s') = r + \gamma . V_k(s') - V_k(s)$$

α is the learning rate(i.e., 0.01)

Temporal Difference Learning Algorithm

Like the value iteration algorithm, the temporal difference learning algorithm is represented as below:

$$V_{k+1}(s) \leftarrow (1-\alpha)V_k(s) + \alpha\left(r + \gamma . V_k(s')\right)$$

The above equation can also be represented as:

With,

$$V_{k+1}(s) \leftarrow V_k(s) + \alpha . \delta_k(s, r, s')$$

$$\delta_k(s, r, s') = r + \gamma . V_k(s') - V_k(s)$$

$r + \gamma . V_k(s')$ is called the TD target

Temporal Difference Learning Algorithm

Like the value iteration algorithm, the temporal difference learning algorithm is represented as below:

$$V_{k+1}(s) \leftarrow (1-\alpha)V_k(s) + \alpha\left(r + \gamma . V_k(s')\right)$$

The above equation can also be represented as:

With,

$$V_{k+1}(s) \leftarrow V_k(s) + \alpha . \delta_k(s, r, s')$$

$$\delta_k(s, r, s') = r + \gamma . V_k(s') - V_k(s)$$

δ_k (s, r, s^′) is called the TD error

Key Features of Temporal Difference Learning

The key features of temporal difference learning are:

Bootstrapping

- TD methods update estimates of the value function using existing estimates.
- In contrast to Monte Carlo methods, TD does not wait for the final outcome of the episode to update estimates.
- TD utilizes learned estimates throughout the episode, enabling continuous refinement.
- The updating process in TD is based on intermediate information, avoiding the need to wait for the episode's completion, as in Monte Carlo methods.

Key Features of Temporal Difference Learning

The key features of temporal difference learning are:

Learning from incomplete sequences

- TD learning has a notable advantage: it doesn't require the completion of entire episodes to learn.
- This method can learn from incomplete sequences of states and rewards, enhancing its adaptability.
- Its ability to learn from partial information makes TD learning efficient for ongoing tasks without defined endpoints.
- This characteristic is particularly beneficial in scenarios where episodic completion is not clearly defined, or continuous learning is essential.

Key Features of Temporal Difference Learning

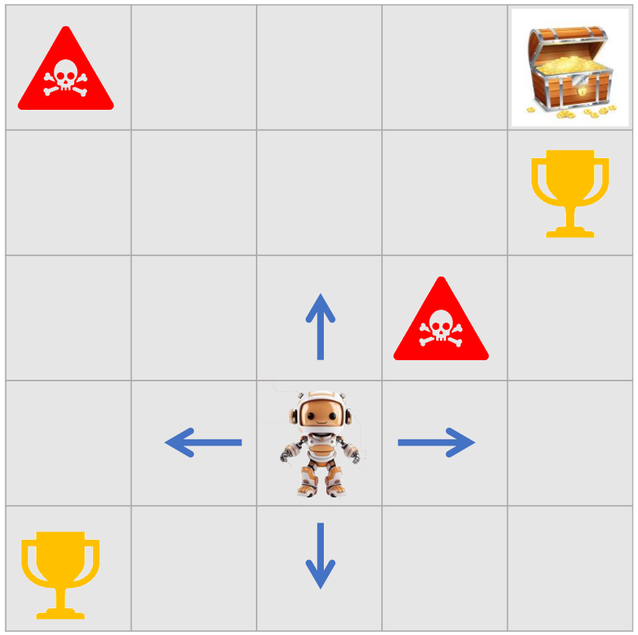
The key features of temporal difference learning are:

Updating estimates based on temporal differences

- The core of TD learning lies in the temporal difference error.
- This error signifies the difference between the predicted value of a state and the observed value of the next state plus reward.
- Utilizing the temporal difference error, TD learning updates the value of the current state.
- This updating process involves adjusting the state value estimate, gradually aligning it with the expected returns.

Temporal Difference Learning: Example

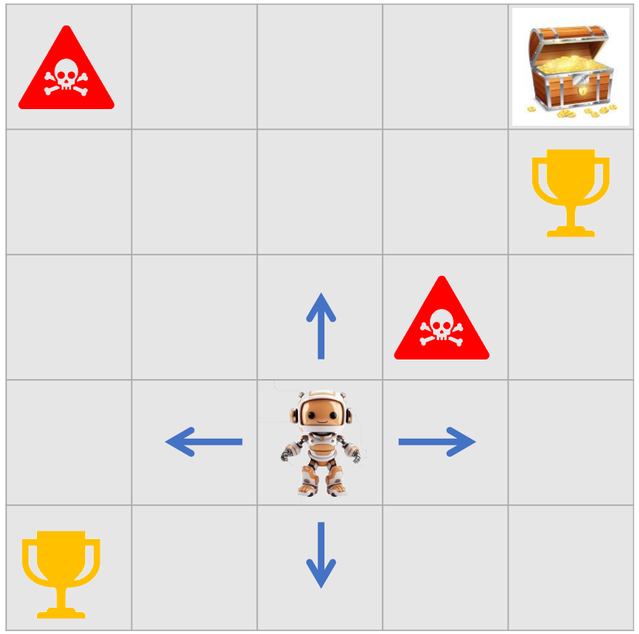
Consider a robot navigating through a grid to reach a goal.



Imagine a grid where a robot must find the shortest path from a starting position to a goal position, with various rewards or penalties scattered throughout.

Temporal Difference Learning: Example Solution

Setting up the grid world



States

Each cell in the grid represents a state.

Actions

At each state, the robot can move up, down, left, or right.

Temporal Difference Learning: Example Solution

Setting up the grid world

Rewards

- Most transitions carry a small negative reward, typically -0.01.
- This negative reward serves as an encouragement for the robot to prioritize finding the shortest path.
- Specific cells, such as a battery recharge station, may have positive rewards.
- Alternatively, certain cells, like those containing obstacles, might have larger negative rewards.

Temporal Difference Learning: Example Solution

Implementing TD learning

Initialization

- Initially, the robot lacks knowledge about the value of each state (cell).
- All non-terminal states can be initialized with a value of 0.
- The goal state, representing reaching the objective, may be assigned a positive value (e.g., +1).

Temporal Difference Learning: Example Solution

Implementing TD learning

Robot's movement

- The movement of the robot is referred to as an episode.
- The robot initiates the episode from the starting position.
- It proceeds by making a series of moves with the objective of reaching the goal.

Temporal Difference Learning: Example Solution

Implementing TD learning

Robot's movement

- TD updates occur after each move within the episode.
- The robot observes the immediate reward and the new state.
- The value of the previous state is updated using the TD error during the ongoing episode.

Temporal Difference Learning: Example Solution

Implementing TD learning

During an episode:

- Observe the immediate reward R, typically a small negative value for each step.
- Observe the new state S' after the move.
- Update the value of the previous state S using the TD formula:

$$\mathrm{V}(s) \leftarrow V(s) + \alpha\big(R + \gamma.V(s') - V(s)\big)$$

This update happens after every move, gradually refining the estimated values of each state to reflect the cost of reaching the goal from those states.

Temporal Difference Learning: Example Solution

Rewards

Intermediate rewards

- As the robot moves, it may encounter cells with varying rewards or penalties.
- Intermediate rewards obtained during movement are integrated into the value updates.
- This incorporation of intermediate rewards enables the robot to learn from its experiences.
- For instance, it can learn to navigate around obstacles or strategically pass through recharging stations.

Temporal Difference Learning: Example Solution

Rewards

Learning path to goal

- Over many episodes, the robot continuously updates the values of states to reflect the shortest path to the goal.
- This updating process considers both intermediate rewards and penalties encountered during movement.
- The policy, which represents the strategy for choosing actions, undergoes gradual improvement.
- As the learning progresses, the refined policy increasingly favors paths that either maximize rewards or minimize penalties.

Monte Carlo Vs Temporal Difference Learning

Monte Carlo vs Temporal Difference

The key comparison points between Monte Carlo and temporal difference learning are listed below:

Dependency on episodes

Monte Carlo

- In Monte Carlo updates are contingent upon the completion of entire episodes.
- The value function remains unchanged until the conclusion of an episode.

Temporal difference

- In TD estimates are updated at each step using the latest reward and the estimated value of the next state.
- The process doesn't mandate waiting for the episode to conclude; it can learn online from incomplete sequences.

Monte Carlo vs Temporal Difference

The key comparison points between Monte Carlo and temporal difference learning are listed below:

Bootstrapping

Monte Carlo

- Avoid bootstrapping; updates are deferred until the complete return is known.
- It utilize the actual complete return for the update process.

Temporal difference

- Involves bootstrapping, updating based on estimates of values from other states.
- Updates are not solely reliant on the final return.

Monte Carlo vs Temporal Difference

The key comparison points between Monte Carlo and temporal difference learning are listed below:

Update frequency

Monte Carlo

- Perform a single update to the value function at the conclusion of each episode.
- This update captures the entire sequence of events that transpired.

Temporal difference

- Updates the value function continuously after each step during an episode.
- Enables more frequent updates compared to a single update at the end.

Monte Carlo vs Temporal Difference

The key comparison points between Monte Carlo and temporal difference learning are listed below:

Variance and bias

Monte Carlo

- Exhibit high variance due to reliance on actual returns, which can vary significantly between episodes.
- Remain unbiased by using the true complete returns for updates.

Temporal difference

- TD learning algorithm typically have lower variance.
- Introduce bias by bootstrapping from other estimated values.

Monte Carlo vs Temporal Difference

The key comparison points between Monte Carlo and temporal difference learning are listed below:

Suitability for different types of tasks

Monte Carlo

- Especially suitable for tasks with guaranteed episode termination.
- Particularly effective in episodic scenarios.

Temporal difference

- Appropriate for both episodic and continuing tasks.
- Particularly beneficial in environments with no natural endpoint or very long episodes.

Monte Carlo vs Temporal Difference

The key comparison points between Monte Carlo and temporal difference learning are listed below:

Complexity and implementation

Monte Carlo

- Conceptually simpler and more straightforward.
- Averages the returns after visits to states.

Temporal difference

- Slightly more complex, requiring estimation and updates based on other estimates.
- Possesses power due to flexibility and efficiency.

Key Takeaways

- Monte Carlo methods learn from full episode sequences.
- Monte Carlo methods utilize complete states, actions, and rewards for learning.
- Various Monte Carlo methods exist in reinforcement learning for decision-making. These include first visit and every visit methods.
- Temporal difference (TD) learning is a type of reinforcement learning method that combines elements of both Monte Carlo methods and dynamic programming.

Knowledge Check

Knowledge Check

1

What does Monte Carlo learning primarily use for updates?

A. Immediate rewards

B. Intermediate states

C. Complete episode sequences

D. Random exploration

Knowledge Check

1

What does Monte Carlo learning primarily use for updates?

A. Immediate rewards

B. Intermediate states

C. Complete episode sequences

D. Random exploration

The correct answer is **C**

Monte Carlo methods use full episode information, including states, actions, and rewards, for updates.

Knowledge Check 2

When does temporal difference learning update its estimates?

A. After each move

B. Only at the end of the episode

C. At random time intervals

D. Before starting the episode

Knowledge Check 2

When does temporal difference learning update its estimates?

A. After each move

B. Only at the end of the episode

C. At random time intervals

D. Before starting the episode

The correct answer is **A**

TD learning updates estimates continuously after each time step within an episode.

Knowledge Check

3

What does bootstrapping refer to in TD learning?

A. Updating using complete returns

B. Waiting for the episode to finish

C. Continuous exploration

D. Using estimated values to update

Knowledge Check 3

What does bootstrapping refer to in TD learning?

A. Updating using complete returns

B. Waiting for the episode to finish

C. Continuous exploration

D. Using estimated values to update

The correct answer is **D**

Bootstrapping in TD involves updating values based on other estimated values.

Knowledge Check

4

What is a potential advantage of Temporal Difference over Monte Carlo?

A. Higher sample efficiency

B. Lower variance

C. Slower convergence

D. Only suitable for episodic tasks

Knowledge Check 4

What is a potential advantage of Temporal Difference over Monte Carlo?

A. Higher sample efficiency

B. Lower variance

C. Slower convergence

D. Only suitable for episodic tasks

The correct answer is **A**

TD methods are often more sample-efficient due to continuous updates.

Thank You