FPT Education

FPT UNIVERSITY

Monte-Carlo for Control

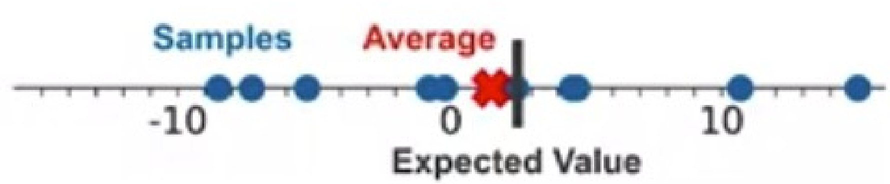
Objectives

- Estimate action-value functions using Monte-Carlo
- Understand the importance of maintaining exploration in Monte-Carlo algorithms
- Understand how to use Monte-Carlo methods to implement a GPI algorithm
- Apply Monte-Carlo with exploring starts to solve an MDP .

Monte-Carlo for Action Value

- Learning action values is almost exactly the same process as learning state values.
- We learned the value of a state by averaging sample returns from that state

$$v_\pi(s) \doteq \mathbb{E}_\pi \left[G_t \mid S_t = s \right]$$



Monte-Carlo for Action Value

- The same process works for action values.
- We collect returns following a policy from state-action pair and then take their average

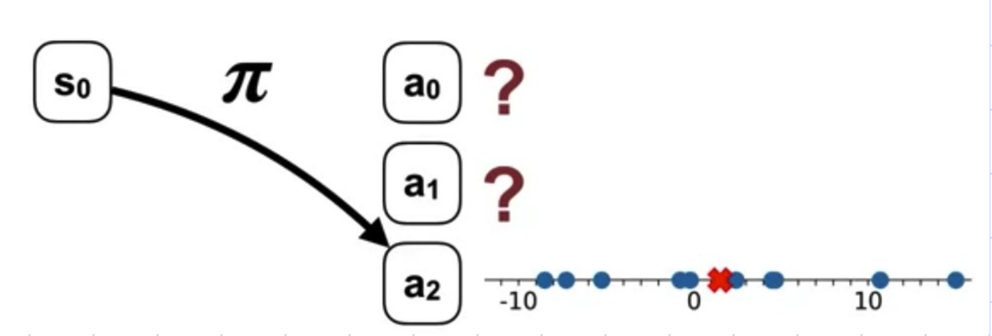
$$q_\pi(s,a) \doteq \mathbb{E}_\pi\left[G_t \mid S_t = s, A_t = a\right]$$

Monte-Carlo for Action Value

- Action values are useful for learning a policy
- They allow us to compare different actions in the same state. Then, we can switch to a better action if one is available.

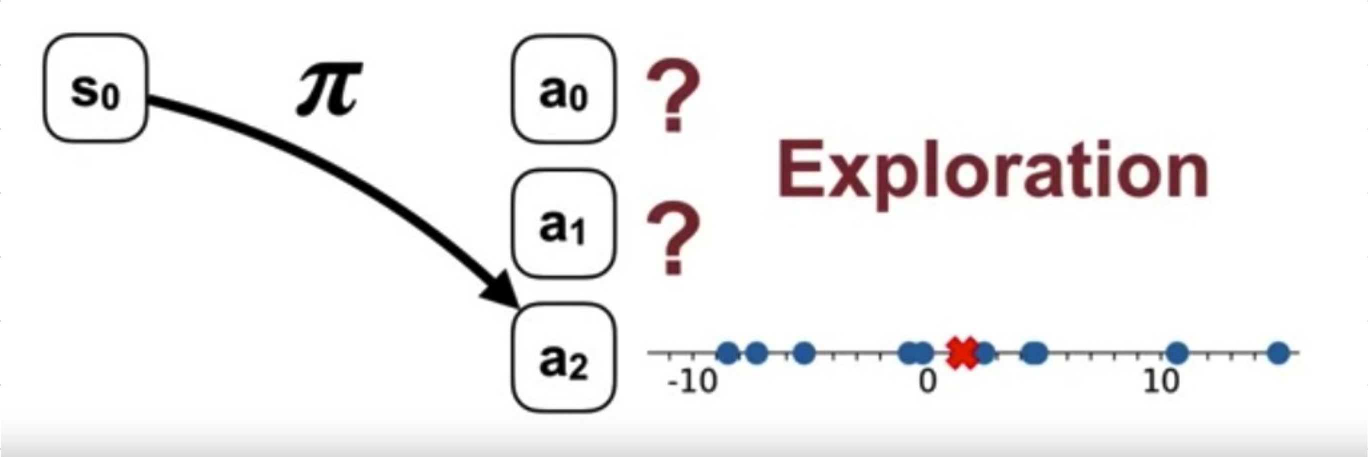
Monte-Carlo for Action Value

- Let's consider learning the action-value function for a deterministic policy.
- Imagine an action that is never selected by the policy.
- The agent will never observe returns corresponding to that action. We won't be able to form an accurate l



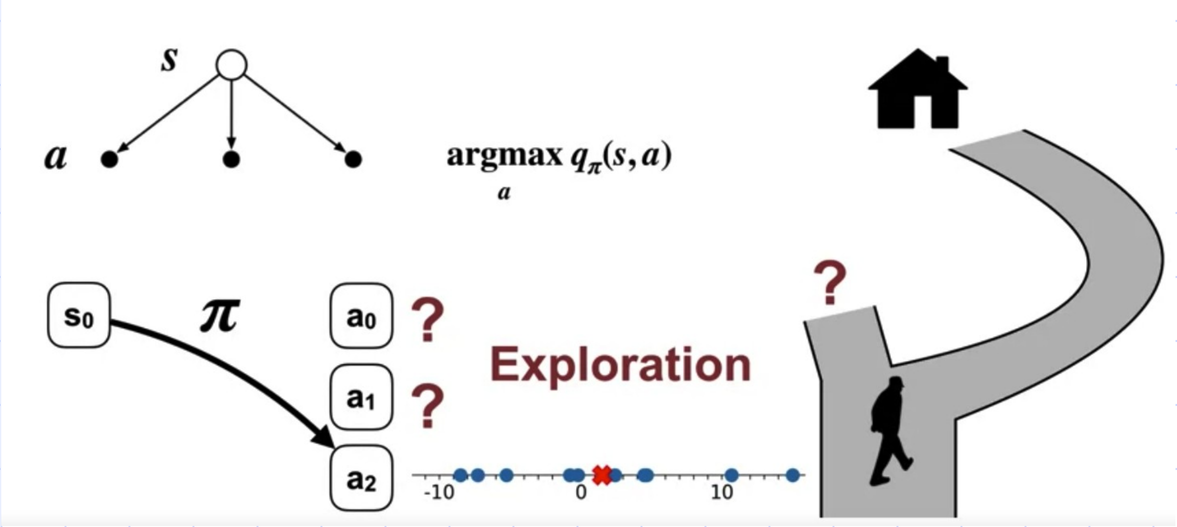
Monte-Carlo for Action Value

- The agent must try all the actions in each state in order to learn their values. This is the problem of maintaining exploration in reinforcement learning.



Monte-Carlo for Action Value

- Real-world example. Imagine walking home along the road you usually take. Recently, a new road was built nearby. If we don't ever try the new way, then we couldn't know if it was actually better



Monte-Carlo for Action Value

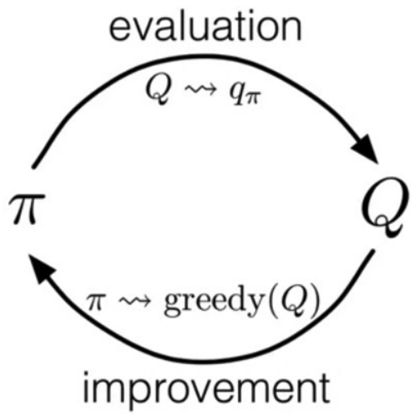
- Exploring starts.
  - This is a way to maintain exploration.
  - In exploring starts, we must guarantee that episodes start in every state-action pair.
  - Afterwards, the agent simply follows its policy.

$$\underbrace{s_0, a_0}_{\text{Random}}, \underbrace{s_1, a_1, s_2, a_2, \cdots}_{\text{From } \pi \text{ and } p}$$

Monte-Carlo for Generalized Policy Iteration

- GPI includes a policy evaluation and a policy improvement step.
- GPI algorithms produce sequences of policies that are at least as good as the policies before them

$$\pi_0 \rightarrow \pi_1 \rightarrow \pi_2 \rightarrow \dots$$



Monte-Carlo for Generalized Policy Iteration

- For the policy improvement step, we can make the policy greedy with respect to the agent's current action value estimates.
- For the policy evaluation step, we will use a Monte Carlo method to estimate the action values.

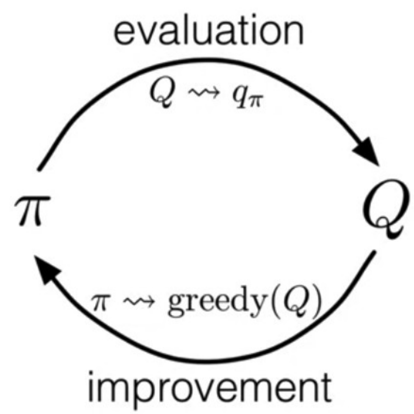
$$\pi_0 \rightarrow \pi_1 \rightarrow \pi_2 \rightarrow \cdots$$

Improvement:

$$\pi_{k+1}(s) \doteq \underset{a}{\operatorname{argmax}}\, q_{\pi_k}(s,a)$$

Evaluation:

Monte Carlo Prediction



Monte-Carlo for Generalized Policy Iteration

- Monte Carlo method for learning action values

Initialize:

$\pi(s) \in \mathcal{A}(s)$ (arbitrarily), for all $s \in \mathcal{S}$

$Q(s,a) \in \mathbb{R}$ (arbitrarily), for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$

$Returns(s,a) \leftarrow$ empty list, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$

Loop forever (for each episode):

Choose $S_0 \in \mathcal{S}$, $A_0 \in \mathcal{A}(S_0)$ randomly such that all pairs have probability > 0

Generate an episode from S_0, A_0, following π: $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$

$G \leftarrow 0$

Loop for each step of episode, $t = T-1, T-2, \ldots, 0$:

$G \leftarrow \gamma G + R_{t+1}$

Append G to $Returns(S_t, A_t)$

$Q(S_t, A_t) \leftarrow \text{average}(Returns(S_t, A_t))$

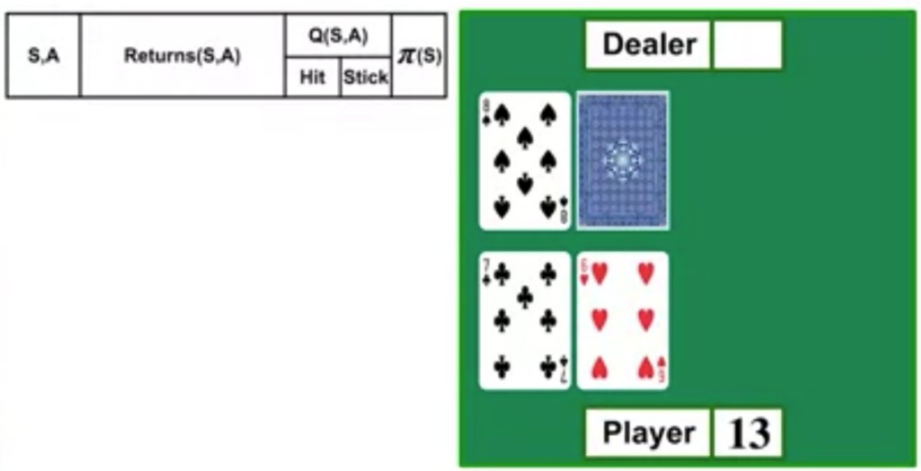
$\pi(S_t) \leftarrow \arg\max_a Q(S_t, a)$

Blackjack Example

- We will train our agent to play blackjack using Monte Carlo with Exploring Starts.
- Exploring starts requires that episodes begin with a random state and action.
- Our version of blackjack naturally starts in random states. Then, all we have to do is to randomly select the first action in each episode. --> the agent ignores what it thinks is the best action in the first state and randomly chooses to hit or stick.

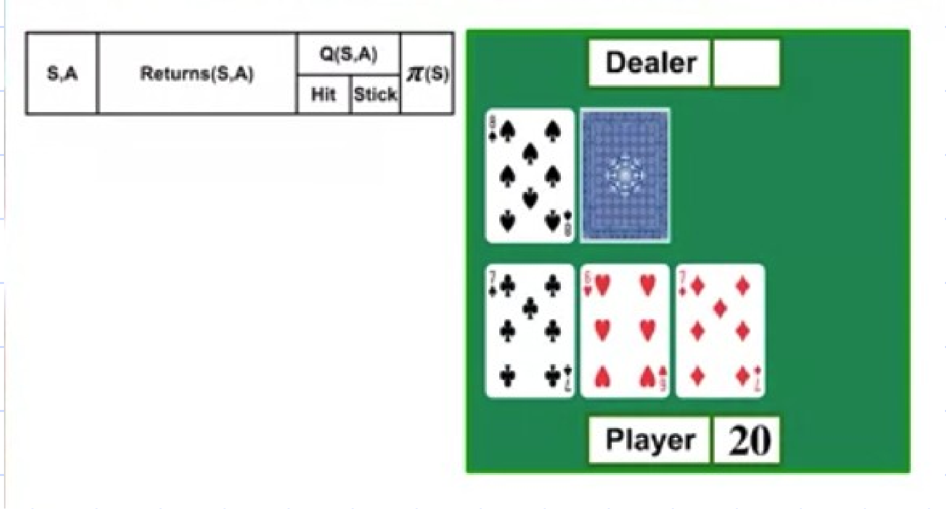
Blackjack Example

- The initial policy is hit, when the agent sum is less than 20, and to stick when the sum is 20 or 21.
- Suppose the agents cards show a total of 13 with no usable ace and the dealers visible card is an eight.



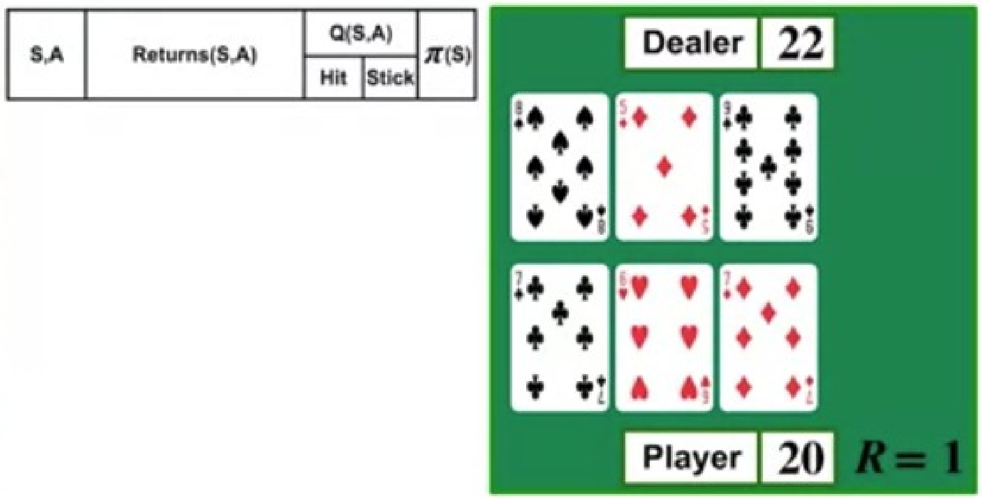
Blackjack Example

- According to the randomly sampled first action, the agent hits.
- The agent gets a seven moving the sum to 20.
- The next step, the agent chooses the action according to its



Blackjack Example

- The dealer draws a nine and goes over 21 losing the game and resulting in a plus one reward for the agent.



Blackjack Example

- In the last non-terminal state, the agent had a sum of 20, no usable ace, and the dealer had an eight.
- From that state, the agent chose to stick.
- The agent adds plus one to its list of returns corresponding to that state action pair.
- The estimated value for the action stick in this state is simply one.

S,A	Returns(S,A)	Q(S,A)		π(S)
		Hit	Stick	
X,Stick	Returns(X,Stick) = [1]	0	1	

X = (NoAce, 20, 8)

Blackjack Example

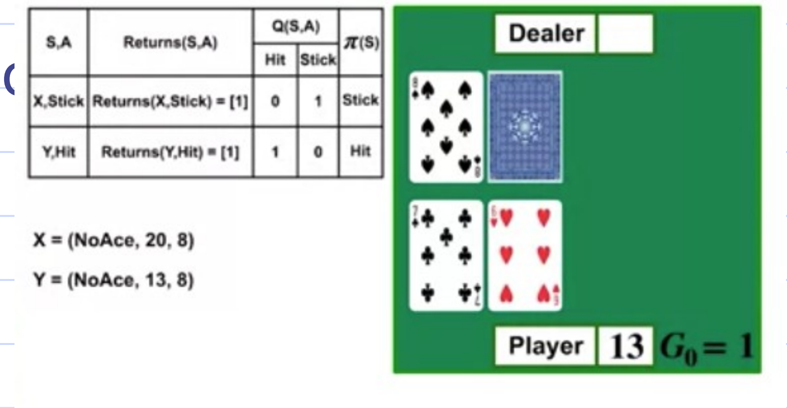
- Let's look at the value of the two actions in this state.
- The agent never tried to hit action, so its value is zero.
- The value of the stick action is one. So the greedy action with r… to stick.
- It has the hig…

Blackjack Example

- One more step to the start state.
- The agent had 13 with no usable ace, and the dealer had an eight.
- The randomly chosen action was to hit. → the agent adds plus one to the list of returns following that state action pair.
- The average of th f the action value.

Blackjack Example

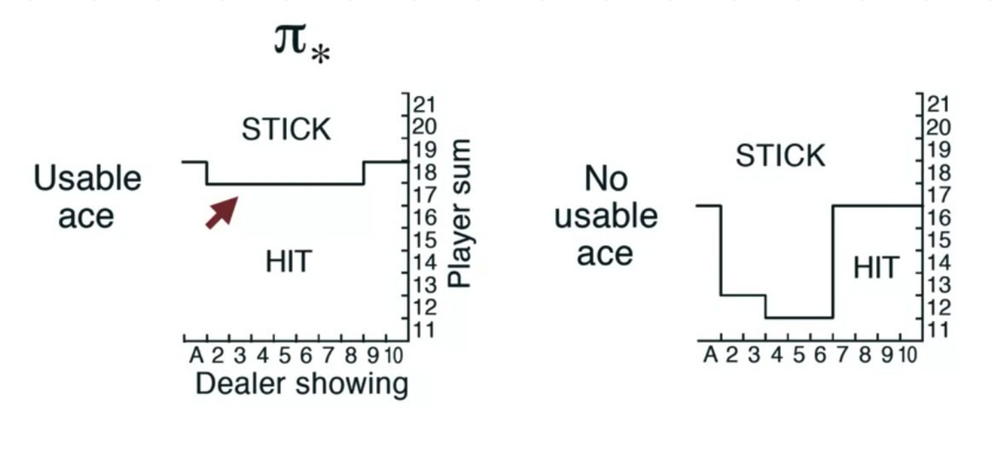
- Finally, the policy is updated in this state to be greedy with respect to the action value estimates.
- In this state, we had never tried to stick action and its value is zero.
- But the hit action resulted in a return of one.
- So the greedy action is to hit, and the policy is updated to reflec



S,A	Returns(S,A)	Q(S,A)		π(S)
		Hit	Stick	
X,Stick	Returns(X,Stick) = [1]	0	1	Stick
Y,Hit	Returns(Y,Hit) = [1]	1	0	Hit

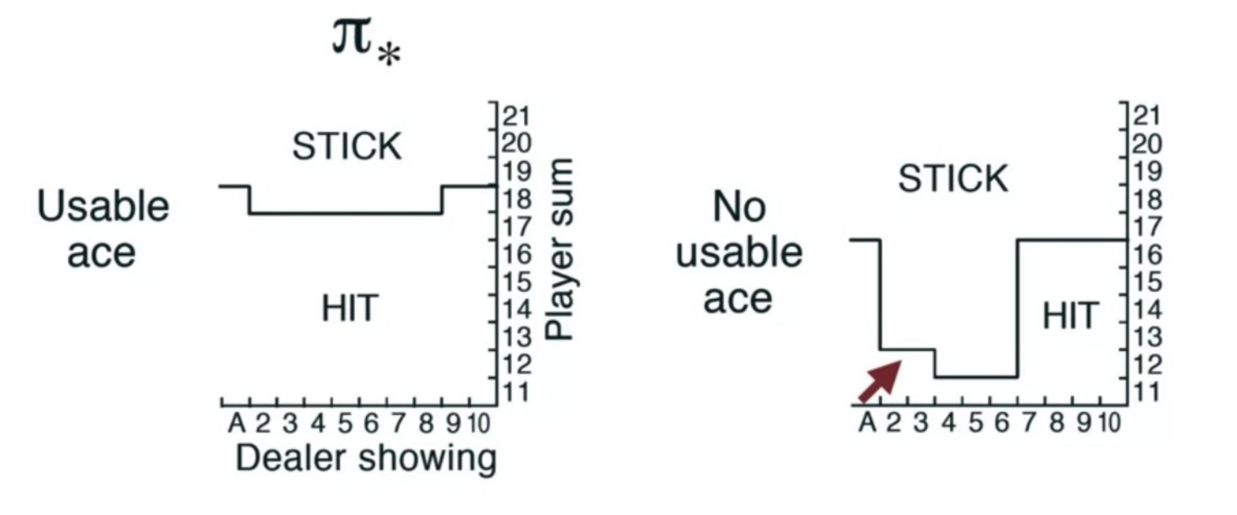
Blackjack Example

- The optimal policy that the agent found after we ran it for a really long time.
  - Notice how the agent plays when it has the usable ace.
  - For most dealer cards, the agent hits until it has the sum near 19.
  - With a usable ace, the agent has a lot more flexibility in calculating th
    more aggres



Blackjack Example

- Without a usable ace, the policy depends a lot more on the cards the dealer is showing.
- The agent sticks when its sum is 13 or greater and the dealer has a low card, like a two or three.



FPT Education
FPT UNIVERSITY

Summary

- Estimate action-value functions using Monte-Carlo
- Understand the importance of maintaining exploration in Monte-Carlo algorithms
- Understand how to use Monte-Carlo methods to implement a GPI algorithm
- Apply Monte-Carlo with exploring starts to solve an MDP .

Q & A