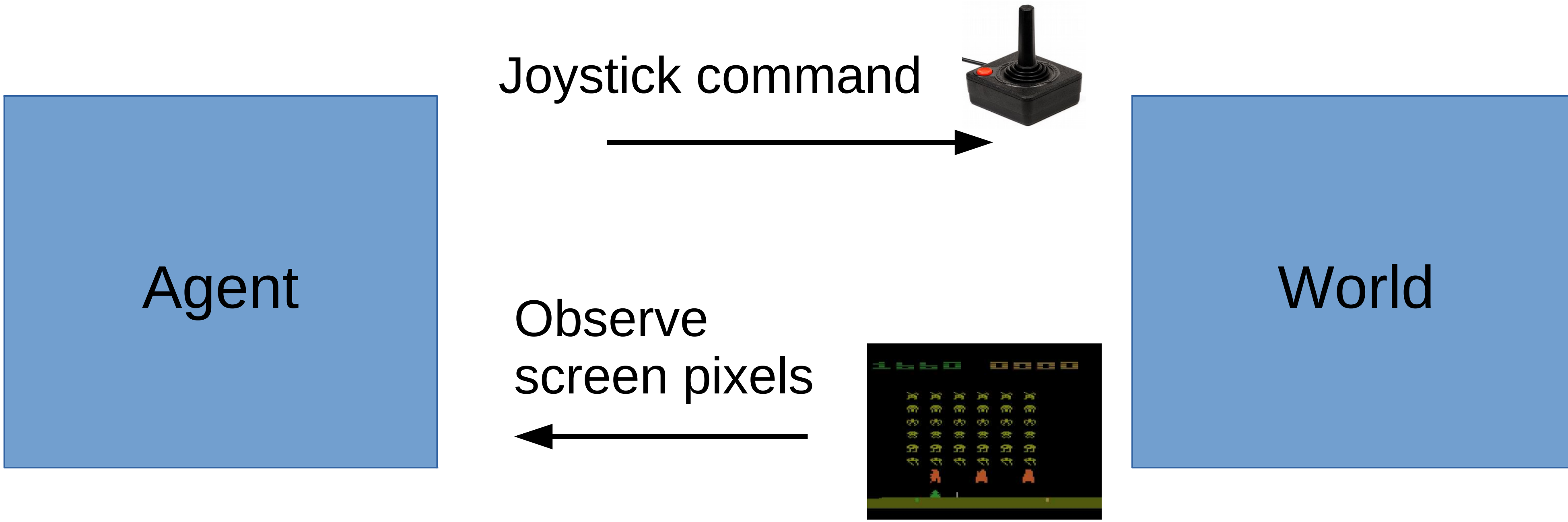
Monte Carlo Approaches to Reinforcement Learning

Robert Platt (w/ Marcus Gualtieri's edits)
Northeastern University

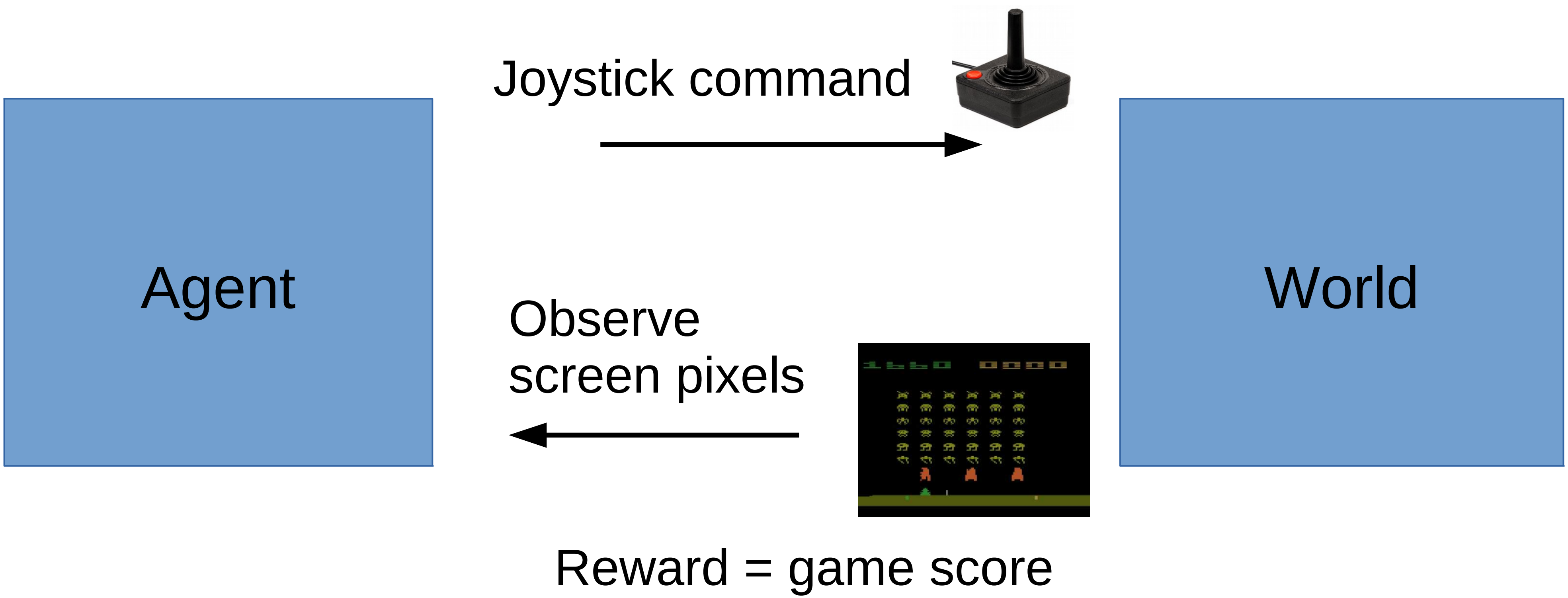
Model Free Reinforcement Learning



Reward = game score

Goal: learn a value function through trial-and-error experience

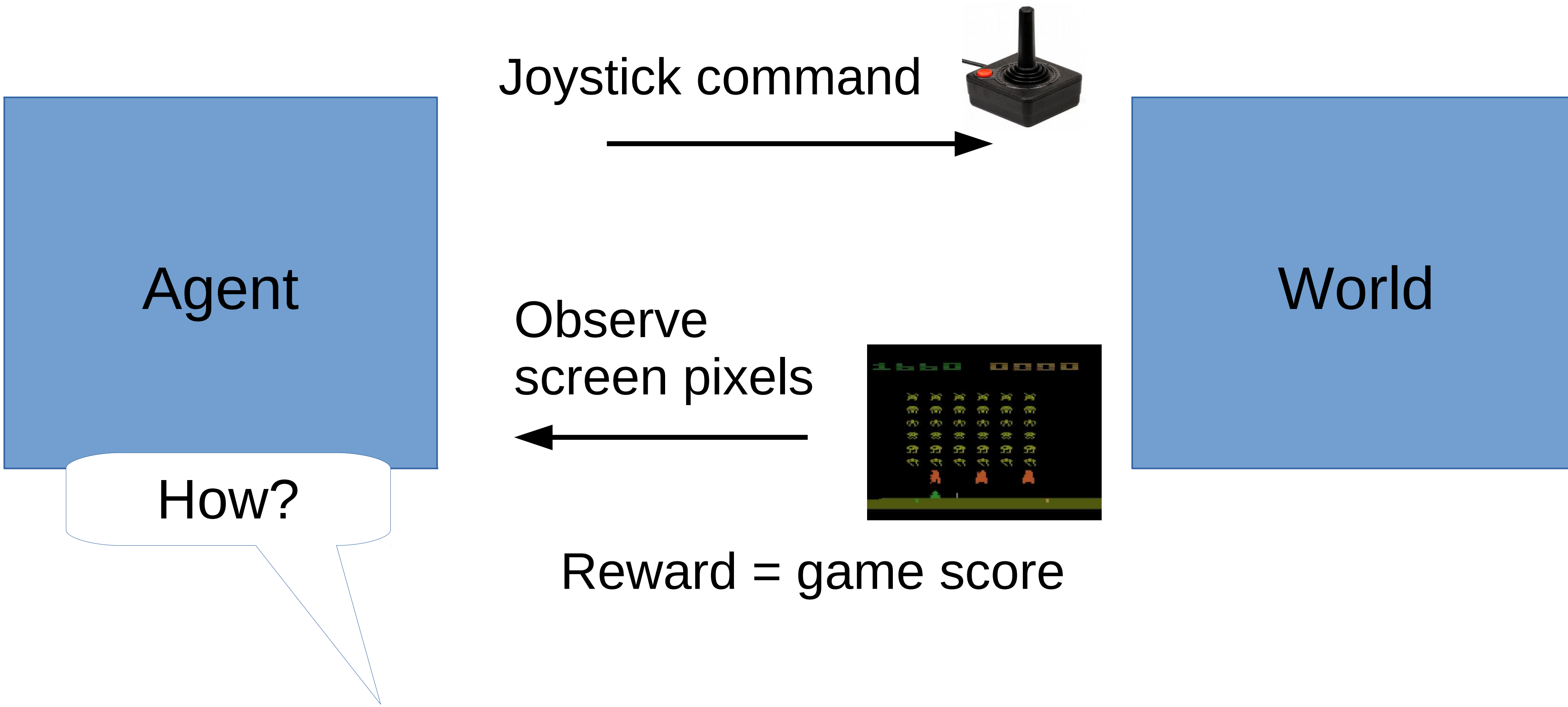
Model Free Reinforcement Learning



Goal: learn a value function through trial-and-error experience

Recall: $V^{\pi}(s_t = s) \equiv$ Value of state s when acting according to policy π

Model Free Reinforcement Learning



Goal: $\boxed{\text{learn a value function through trial-and-error experience}}$

Recall: $V^{\pi}(s_t = s) \equiv$ Value of state s when acting according to policy π

Model Free Reinforcement Learning

Simplest solution: average all outcomes from previous experiences in a given state

– this is called a *Monte Carlo* method

Agent

een pixels

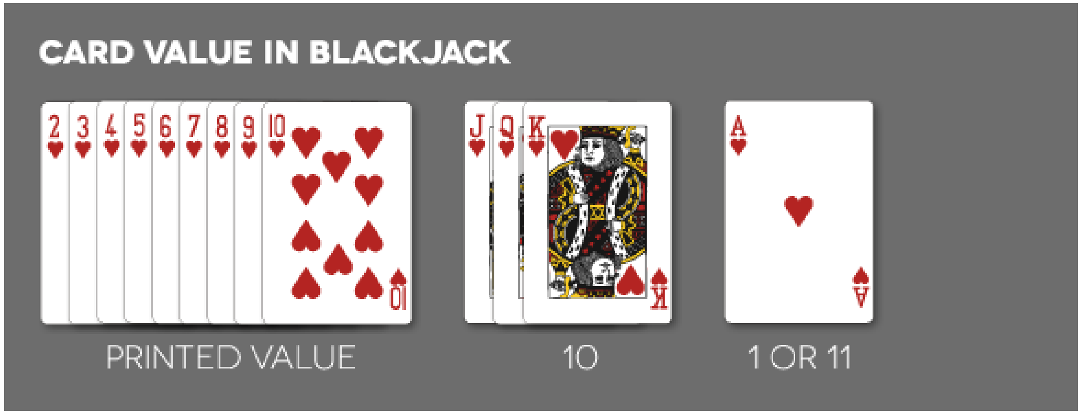
Reward = game score

How?

Goal: learn a value function through trial-and-error experience

Recall: $V^{\pi}(s_t = s) \equiv$ Value of state S when acting according to policy π

Running Example: Blackjack



State: sum of cards in agent's hand + dealer's showing card + does agent have usable ace?

Actions: hit, stick

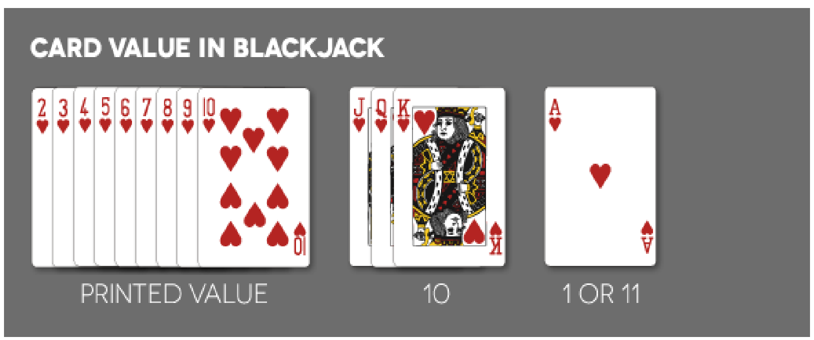
Objective: Have agent's card sum be greater than the dealer's without exceeding 21

Reward: +1 for winning, 0 for a draw, -1 for losing

Discounting: $\gamma = 1$

Dealer policy: draw until sum at least 17

Running Example: Blackjack



Dealer's Up Card

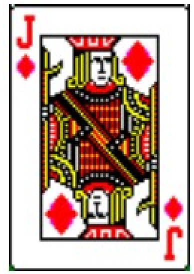
Your Hand	2	3	4	5	6	7	8	9	10	A
17+	S	S	S	S	S	S	S	S	S	S
16	S	S	S	S	S	H	H	H	H	H
15	S	S	S	S	S	H	H	H	H	H
14	S	S	S	S	S	H	H	H	H	H
13	S	S	S	S	S	H	H	H	H	H
12	H	H	S	S	S	H	H	H	H	H
11	D	D	D	D	D	D	D	D	D	H
10	D	D	D	D	D	D	D	D	H	H
9	H	D	D	D	D	H	H	H	H	H
5 - 8	H	H	H	H	H	H	H	H	H	H
A 8 - 10	S	S	S	S	S	S	S	S	S	S
A, 7	S	D	D	D	D	S	S	H	H	H
A, 6	H	D	D	D	D	H	H	H	H	H
A, 5	H	H	D	D	D	H	H	H	H	H
A, 4	H	H	D	D	D	H	H	H	H	H
A, 3	H	H	H	D	D	H	H	H	H	H
A, 2	H	H	H	D	D	H	H	H	H	H
A,A 8,8	SP	SP	SP	SP	SP	SP	SP	SP	SP	SP
10, 10	S	S	S	S	S	S	S	S	S	S
9, 9	SP	SP	SP	SP	SP	S	SP	SP	S	S
7, 7	SP	SP	SP	SP	SP	SP	H	H	H	H
6, 6	SP	SP	SP	SP	SP	H	H	H	H	H
5, 5	D	D	D	D	D	D	D	D	H	H
4, 4	H	H	H	SP	SP	H	H	H	H	H
3, 3	SP	SP	SP	SP	SP	SP	H	H	H	H
2, 2	SP	SP	SP	SP	SP	SP	H	H	H	H
	2	3	4	5	6	7	8	9	10	A

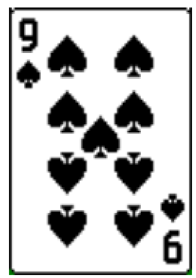
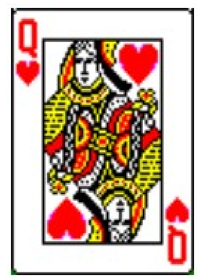
If doubling down after splitting is not allowed, then just hit the following:
2,2 and 3,3 vs. 2 and 3   4,4 vs. 5 and 6   6,6 vs. 2

HIT | STAND | DOUBLE DOWN | SPLIT

Blackjack "Basic Strategy" is a set of rules for play so as to maximize return
- well known in the gambling community
- how might an RL agent *learn* the Basic Strategy?

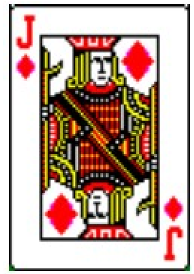
Monte Carlo Policy Evaluation: Example

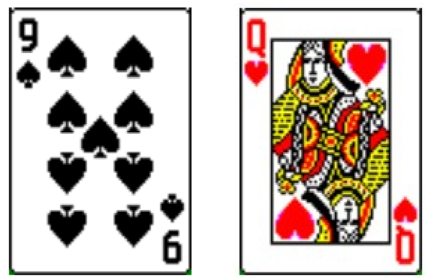
Dealer card: 

Agent's hand:  

State	Action	Next State	Reward

Monte Carlo Policy Evaluation: Example

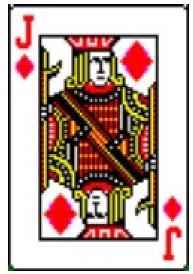
Dealer card: 

Agent’s hand: 

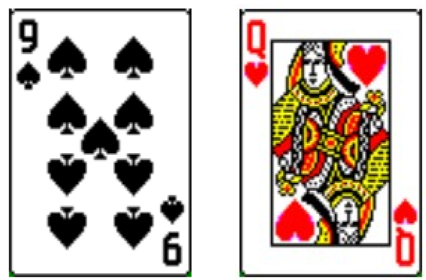
State	Action	Next State	Reward
19, 10, no			

Monte Carlo Policy Evaluation: Example

Dealer card:



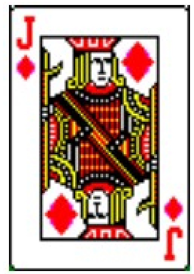
Agent's hand:

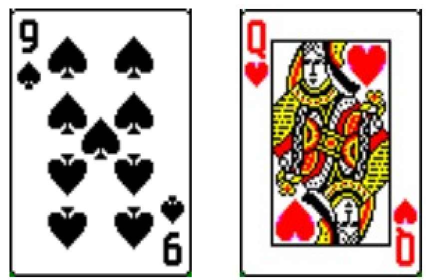


Agent sum, dealer's card, ace?

State	Action	Next State	Reward
19, 10, no			

Monte Carlo Policy Evaluation: Example

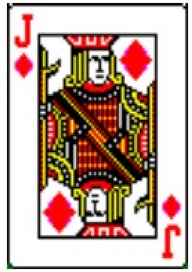
Dealer card: 

Agent's hand: 

Agent sum, dealer's card, ace?

State	Action	Next State	Reward
19, 10, no	HIT		

Monte Carlo Policy Evaluation: Example

Dealer card: 

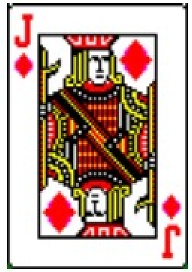
Agent's hand: 

Agent sum, dealer's card, ace?

State	Action	Next State	Reward
19, 10, no	HIT	22, 10, no	-1

Monte Carlo Policy Evaluation: Example

Dealer card:



Agent’s hand:

Agent sum, dealer’s card, ace?

Bust!
(reward = -1)

State	Action	Next State	Reward
19, 10, no	HIT	22, 10, no	-1

Monte Carlo Policy Evaluation: Example

State	Action	Next State	Reward
19, 10, no	HIT	22, 10, no	-1

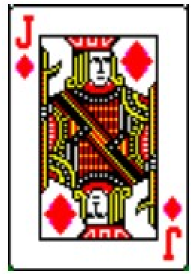
Upon episode termination, make the following value function updates:

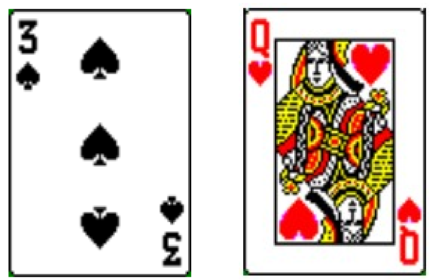
$$V((19, 10, no)) \leftarrow -1$$

Monte Carlo Policy Evaluation: Example

Next episode...

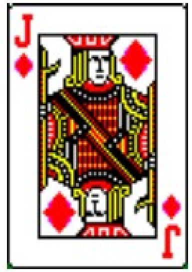
Monte Carlo Policy Evaluation: Example

Dealer card: 

Agent's hand: 

State	Action	Next State	Reward
13, 10, no			

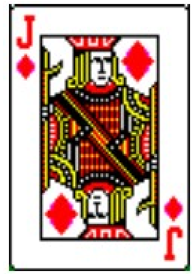
Monte Carlo Policy Evaluation: Example

Dealer card: 

Agent's hand: 

State	Action	Next State	Reward
13, 10, no	HIT	16, 10, no	0

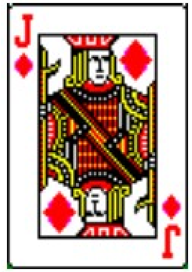
Monte Carlo Policy Evaluation: Example

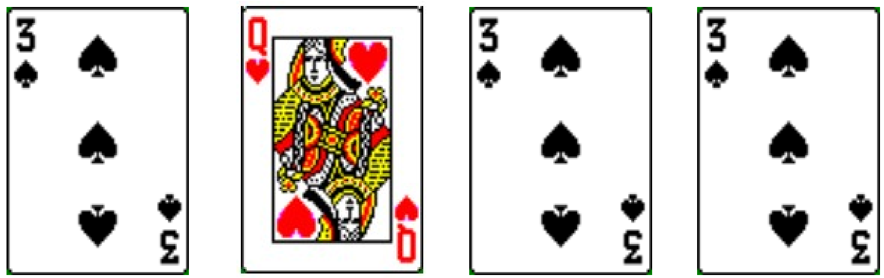
Dealer card: 

Agent's hand: 

State	Action	Next State	Reward
13, 10, no	HIT	16, 10, no	0
13, 10, no			

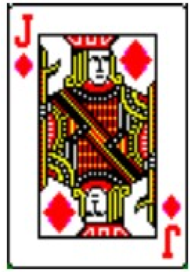
Monte Carlo Policy Evaluation: Example

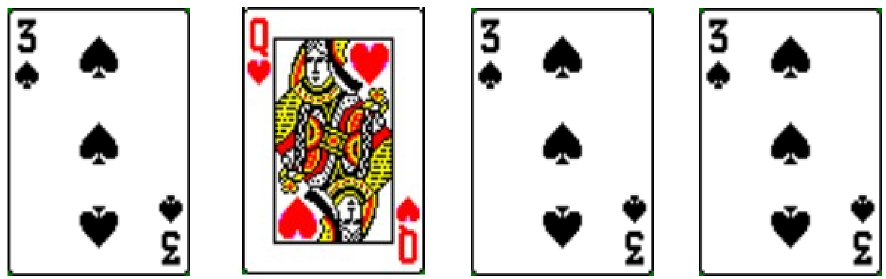
Dealer card: 

Agent's hand: 

State	Action	Next State	Reward
13, 10, no	HIT	16, 10, no	0
13, 10, no	HIT	19, 10, no	0

Monte Carlo Policy Evaluation: Example

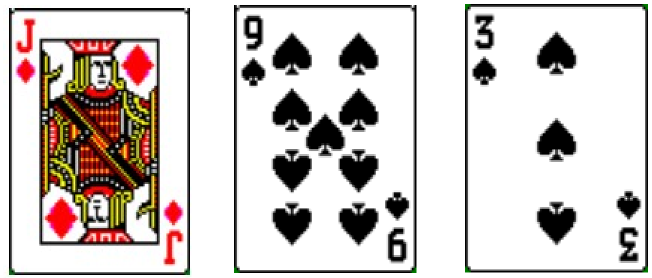
Dealer card: 

Agent's hand: 

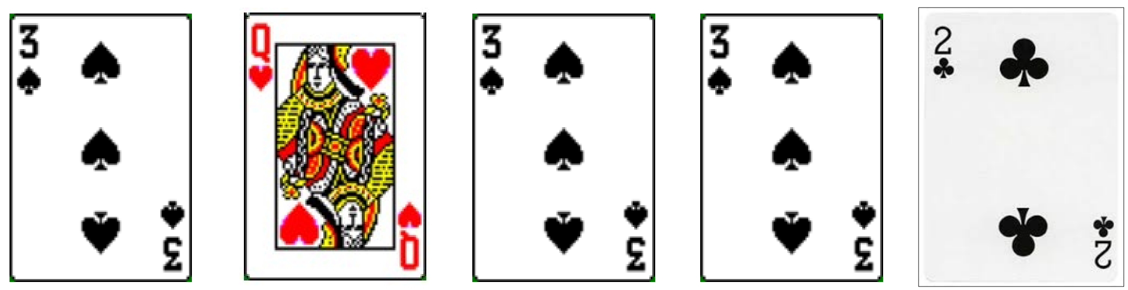
State	Action	Next State	Reward
13, 10, no	HIT	16, 10, no	0
13, 10, no	HIT	19, 10, no	0
19, 10, no			

Monte Carlo Policy Evaluation: Example

Dealer card:



Agent's hand:



State	Action	Next State	Reward
13, 10, no	HIT	16, 10, no	0
13, 10, no	HIT	19, 10, no	0
19, 10, no	HIT	21, 22, no	1

Monte Carlo Policy Evaluation: Example

State	Action	Next State	Reward
13, 10, no	HIT	16, 10, no	0
16, 10, no	HIT	19, 10, no	0
19, 10, no	HIT	21, 22, no	1

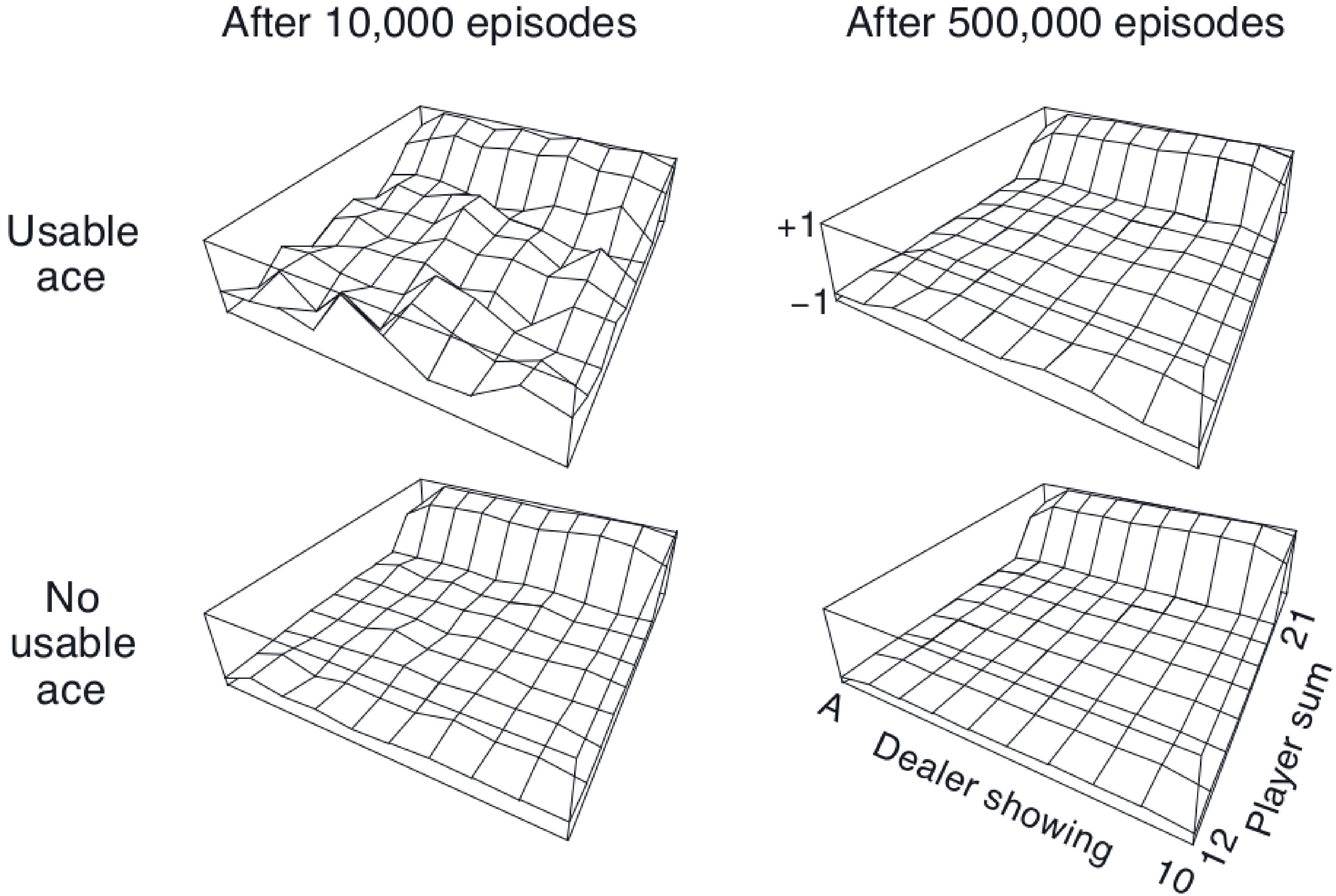
Upon episode termination, make the following value function updates:

$$V((13, 10, no)) \leftarrow 1$$

$$V((16, 10, no)) \leftarrow 1$$

$$V((19, 10, no)) \leftarrow (-1 + 1)/2$$

Monte Carlo Policy Evaluation: Example



Value function learned for “hit everything except for 20 and 21” policy.

Monte Carlo Policy Evaluation

Given a policy, π , estimate the value function, $V(s)$, for all states, $s \in \mathcal{S}$

Monte Carlo Policy Evaluation

Given a policy, π, estimate the value function, $V(s)$, for all states, $s \in \mathcal{S}$

Monte Carlo Policy Evaluation (first visit):

Input: a policy π to be evaluated

Initialize:

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

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

Loop forever (for each episode):

Generate an episode following π: $S_0, A_0, R_1, S_1, A_1, R_2, \ldots, S_{T-1}, A_{T-1}, R_T$

$G \leftarrow 0$

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

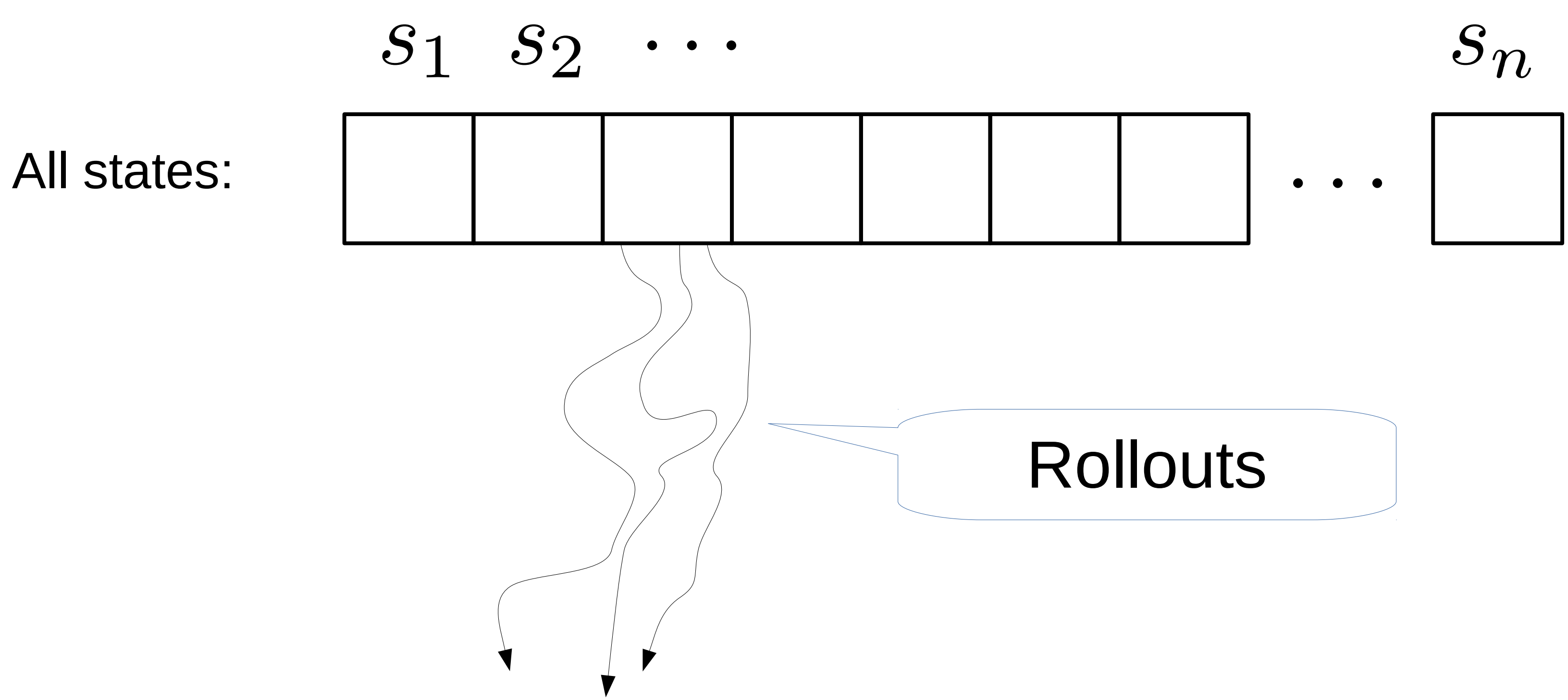
$G \leftarrow \gamma G + R_{t+1}$

Unless S_t appears in $S_0, S_1, \ldots, S_{t-1}$:

Append G to $Returns(S_t)$

$V(S_t) \leftarrow \text{average}(Returns(S_t))$

Monte Carlo Policy Evaluation



To get an accurate estimate of the value function, every state has to be visited many times.

Think-pair-share: frozenlake env

	0	1	2	3
0	S	F	F	F
1	F	H	F	H
2	F	F	F	H
3	H	F	F	G

States: grid world coordinates

Actions: L, R, U, D

Reward: 0 except at G

Think-pair-share: frozenlake env

```
  0123
0 SFFF
1 FHFH
2 FFFH
3 HFFG
```

States: grid world coordinates

Actions: L, R, U, D

Reward: 0 except at G where r=1

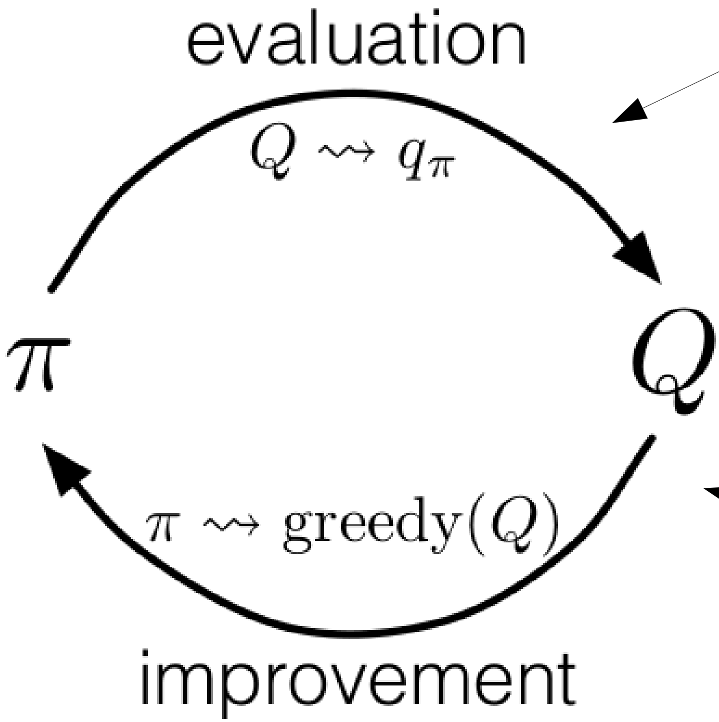
Given: three episodes as shown

Calculate: values of states on top row as calculated by MC

Monte Carlo Control

So far, we're only talking about policy *evaluation*

… but RL requires us to find a policy, not just evaluate it… How?



Estimate $Q_\pi(s_t = s, a_t = a)$ via rollouts

$$\pi(s) \leftarrow \arg\max_{a \in \mathcal{A}} Q_\pi(s_t = s, a_t = a)$$

Key idea: evaluate/improve policy iteratively...

Monte Carlo Control

Monte Carlo, Exploring Starts

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, $t = T-1, T-2, \ldots, 0$:

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

Unless the pair S_t, A_t appears in $S_0, A_0, S_1, A_1 \ldots, S_{t-1}, A_{t-1}$:

Append G to $Returns(S_t, A_t)$

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

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

Monte Carlo Control

Monte Carlo, Exploring Starts

Exploring starts:
– each episode starts with a random action taken from a random state

Initialize:
  $\pi(s) \in \mathcal{A}(s)$ (arbitrarily), for all s
  $Q(s,a) \in \mathbb{R}$ (arbitrarily), for all s
  $Returns(s,a) \leftarrow$ empty list, for al

Loop forever (for each episode):
  $\boxed{\text{Choose } S_0 \in \mathcal{S},\ A_0 \in \mathcal{A}(S_0) \text{ randomly such that all pairs have probability} > 0}$
  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, $t = T-1, T-2, \ldots, 0$:
    $G \leftarrow \gamma G + R_{t+1}$
    Unless the pair S_t, A_t appears in $S_0, A_0, S_1, A_1 \ldots, S_{t-1}, A_{t-1}$:
      Append G to $Returns(S_t, A_t)$
      $Q(S_t, A_t) \leftarrow \text{average}(Returns(S_t, A_t))$
      $\pi(S_t) \leftarrow \arg\max_a Q(S_t, a)$

Monte Carlo Control

Monte Carlo, Exploring Starts

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, $t = T-1, T-2, \ldots, 0$:

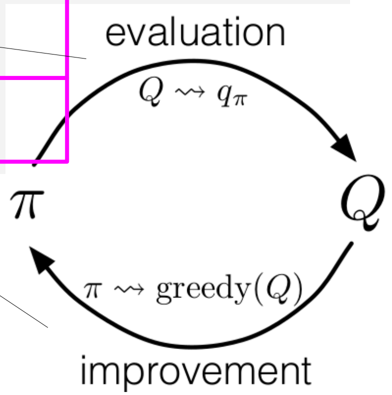
    $G \leftarrow \gamma G + R_{t+1}$

    Unless the pair S_t, A_t appears in $S_0, A_0, S_1, A_1 \ldots, S_{t-1}, A_{t-1}$:

      Append G to $Returns(S_t, A_t)$

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

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



Monte Carlo Control

Monte Carlo, Exploring Starts

Initialize:

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

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

$Returns(s,a) \leftarrow$ empty list, for a…

Loop forever (for each episode):

Choose $S_0 \in \mathcal{S}$, $A_0 \in \mathcal{A}(S_0)$ randomly such that all pairs ha… …robability > 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, $t = T-1, T-2, \ldots, 0$:

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

Unless the pair S_t, A_t appears in $S_0, A_0, S_1, A_1 \ldots, S_{t-1}, A_{t-1}$:

Append G to $Returns(S_t, A_t)$

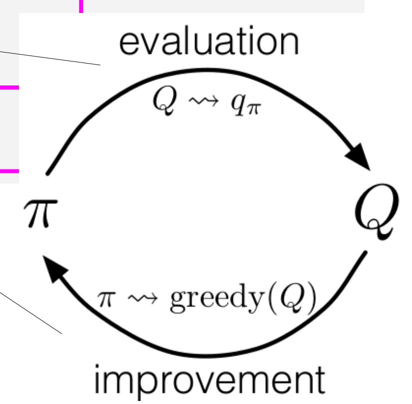
$Q(S_t, A_t) \leftarrow \text{average}(Returns(S_t, A_t))$

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

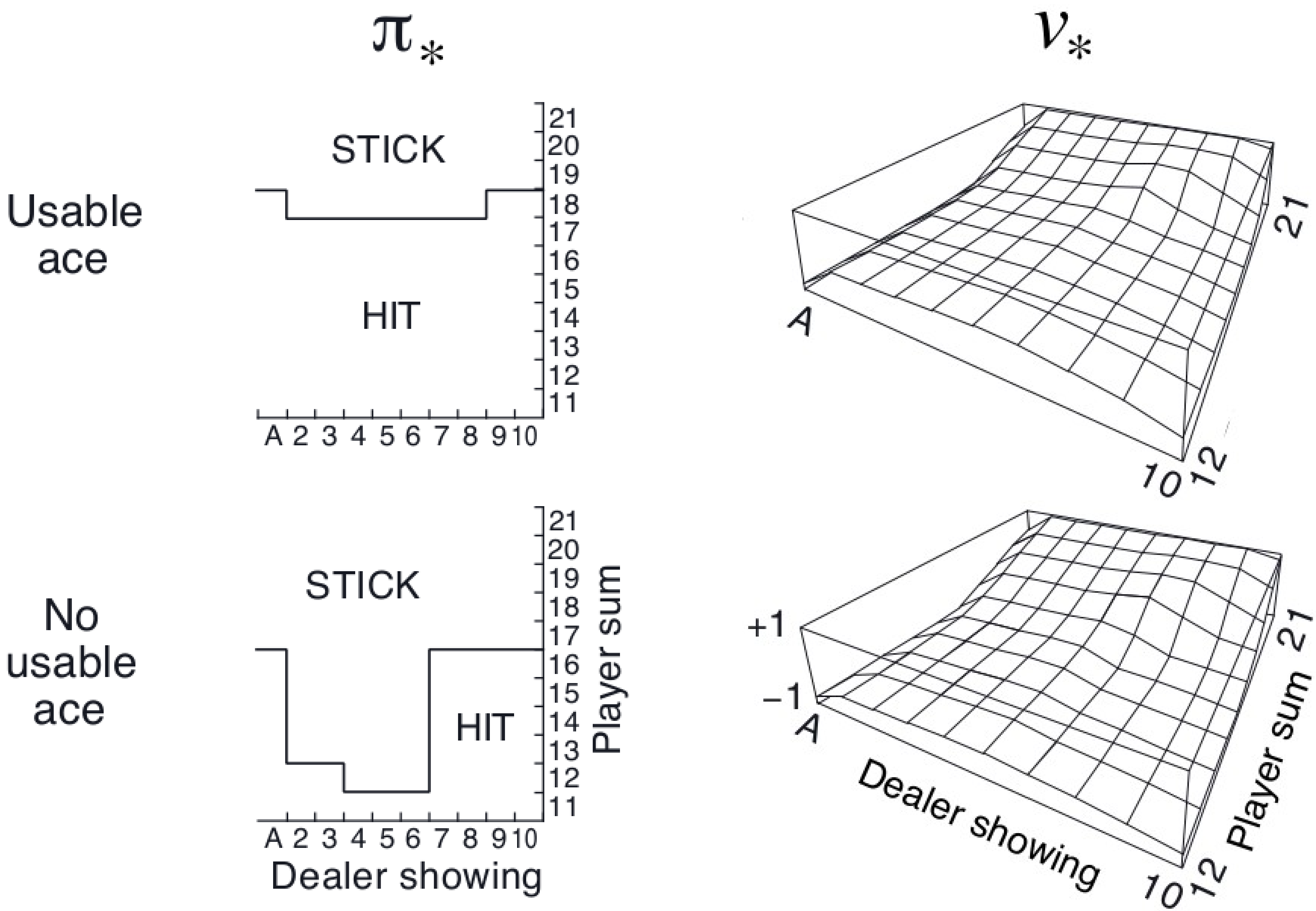
Notice there is only one step of policy evaluation

– that's okay.

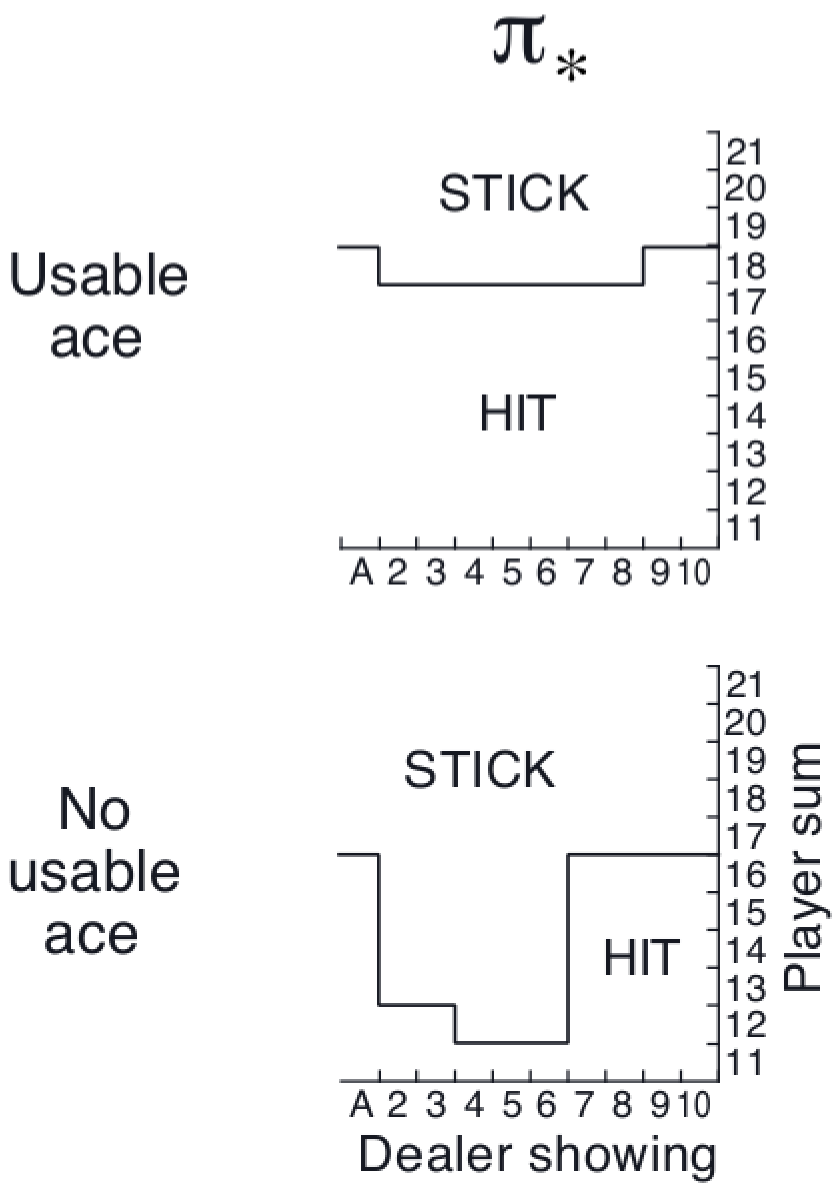
– each evaluation iter moves value fn toward its optimal value. Good enough to improve policy.



Monte Carlo Control



Monte Carlo Control



What the MC agent learned

Dealer's Up Card

Your Hand	2	3	4	5	6	7	8	9	10	A
17+	S	S	S	S	S	S	S	S	S	S
16	S	S	S	S	S	H	H	H	H	H
15	S	S	S	S	S	H	H	H	H	H
14	S	S	S	S	S	H	H	H	H	H
13	S	S	S	S	S	H	H	H	H	H
12	H	H	S	S	S	H	H	H	H	H
11	D	D	D	D	D	D	D	D	D	H
10	D	D	D	D	D	D	D	D	H	H
9	H	D	D	D	D	H	H	H	H	H
5 - 8	H	H	H	H	H	H	H	H	H	H
A 8 - 10	S	S	S	S	S	S	S	S	S	S
A, 7	S	D	D	D	D	S	S	H	H	H
A, 6	H	D	D	D	D	H	H	H	H	H
A, 5	H	H	D	D	D	H	H	H	H	H
A, 4	H	H	D	D	D	H	H	H	H	H
A, 3	H	H	H	D	D	H	H	H	H	H
A, 2	H	H	H	D	D	H	H	H	H	H
A,A 8,8	SP	SP	SP	SP	SP	SP	SP	SP	SP	SP
10, 10	S	S	S	S	S	S	S	S	S	S
9, 9	SP	SP	SP	SP	SP	S	SP	SP	S	S
7, 7	SP	SP	SP	SP	SP	SP	H	H	H	H
6, 6	SP	SP	SP	SP	SP	H	H	H	H	H
5, 5	D	D	D	D	D	D	D	D	H	H
4, 4	H	H	H	SP	SP	H	H	H	H	H
3, 3	SP	SP	SP	SP	SP	SP	H	H	H	H
2, 2	SP	SP	SP	SP	SP	SP	H	H	H	H
	2	3	4	5	6	7	8	9	10	A

If doubling down after splitting is not allowed, then just hit the following:
2,2 and 3,3 vs. 2 and 3 4,4 vs. 5 and 6 6,6 vs. 2

HIT	STAND	DOUBLE DOWN	SPLIT

The official "basic strategy"

Monte Carlo Control: Convergence

- Greedified policy meets the conditions for policy improvement:

$$\begin{aligned} q_{\pi_k}(s, \pi_{k+1}(s)) &= q_{\pi_k}(s, \arg\max_a q_{\pi_k}(s,a)) \\ &= \max_a q_{\pi_k}(s,a) \\ &\geq q_{\pi_k}(s, \pi_k(s)) \\ &\geq v_{\pi_k}(s). \end{aligned}$$

- And thus must be ≥ π_k by the policy improvement theorem
- This assumes exploring starts and infinite number of episodes for MC policy evaluation
- To solve the latter:
  - update only to a given level of performance
  - alternate between evaluation and improvement per episode

Monte Carlo Control: Convergence

- Greedified policy meets the conditions for policy improvement:

$$
\begin{aligned}
q_{\pi_k}(s, \pi_{k+1}(s)) &= q_{\pi_k}(s, \arg\max_a q_{\pi_k}(s,a)) \\
&= \max_a q_{\pi_k}(s,a) \\
&\geq q_{\pi_k}(s, \pi_k(s)) \\
&\geq v_{\pi_k}(s).
\end{aligned}
$$

- And thus must be $\geq \pi_k$ by the $\boxed{\text{policy improvement theorem}}$
- This assumes exploring starts and in ... er of episodes for
- To
  -
  - a ... pisode

If $q_\pi(s, \pi'(s)) \geq v_\pi(s), \forall s \in S,$

then $v_{\pi'}(s) \geq v_\pi(s)$ i.e. π' is better than π

Policy Improvement Theorem: Proof (Sketch)

$$
\begin{aligned}
v_\pi(s) &\le q_\pi(s, \pi'(s)) \\
&= \mathbb{E}[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s, A_t = \pi'(s)] && \text{(by (4.6))} \\
&= \mathbb{E}_{\pi'}[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s] \\
&\le \mathbb{E}_{\pi'}[R_{t+1} + \gamma q_\pi(S_{t+1}, \pi'(S_{t+1})) \mid S_t = s] && \text{(by (4.7))} \\
&= \mathbb{E}_{\pi'}[R_{t+1} + \gamma \mathbb{E}_{\pi'}[R_{t+2} + \gamma v_\pi(S_{t+2}) | S_{t+1}, A_{t+1} = \pi'(S_{t+1})] \mid S_t = s] \\
&= \mathbb{E}_{\pi'}\big[R_{t+1} + \gamma R_{t+2} + \gamma^2 v_\pi(S_{t+2}) \,\big|\, S_t = s\big] \\
&\le \mathbb{E}_{\pi'}\big[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 v_\pi(S_{t+3}) \,\big|\, S_t = s\big] \\
&\;\;\vdots \\
&\le \mathbb{E}_{\pi'}\big[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \cdots \,\big|\, S_t = s\big] \\
&= v_{\pi'}(s).
\end{aligned}
$$

E-Greedy Exploration

Without exploring starts, we are not guaranteed to explore the state/action space
– why is this a problem?
– what happens if we never experience certain transitions?

Monte Carlo, Exploring Starts:

Initialize:
  $\pi(s) \in \mathcal{A}(s)$ (arbitrarily), for all s
  $Q(s,a) \in \mathbb{R}$ (arbitrarily), for all s
  $Returns(s,a) \leftarrow$ empty list, for all

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, $t = T-1, T-2, \ldots, 0$:
    $G \leftarrow \gamma G + R_{t+1}$
    Unless the pair S_t, A_t appears in $S_0, A_0, S_1, A_1 \ldots, S_{t-1}, A_{t-1}$:
      Append G to $Returns(S_t, A_t)$
      $Q(S_t, A_t) \leftarrow \text{average}(Returns(S_t, A_t))$
      $\pi(S_t) \leftarrow \text{argmax}_a\, Q(S_t, a)$

E-Greedy Exploration

Monte Carlo, Exploring Starts:

Initialize:

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

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

$Returns(s,a) \leftarrow$ empty list, for all s

Loop forever (for each episode):

Choose $S_0 \in \mathcal{S}$, $A_0 \in \mathcal{A}(S_0)$ randomly suc

Generate an episode from S_0, A_0, followin

$G \leftarrow 0$

Loop for each step of episode, $t = T-1, T$

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

Unless the pair S_t, A_t appears in $S_0, A_0, S_1, A_1 \ldots, S_{t-1}, A_{t-1}$:

Append G to $Returns(S_t, A_t)$

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

$\pi(S_t) \leftarrow \text{argmax}_a Q(S_t, a)$

Without exploring starts, we are not guaranteed to explore the state/action space

– why is this a problem?

– what happens if we never experience certain transitions?

Can we accomplish this without exploring starts?

E-Greedy Exploration

Monte Carlo, Exploring Starts:

Initialize:
  $\pi(s) \in \mathcal{A}(s)$ (arbitrarily), for all s
  $Q(s,a) \in \mathbb{R}$ (arbitrarily), for all s
  $Returns(s,a) \leftarrow$ empty list, for al

Loop forever (for each episode):
  Choose $S_0 \in \mathcal{S}$, $A_0 \in \mathcal{A}(S_0)$ randomly such th
  Generate an episode from S_0, A_0, followin
  $G \leftarrow 0$
  Loop for each step of episode, $t = T-1, T$
    $G \leftarrow \gamma G + R_{t+1}$
    Unless the pair S_t, A_t appears in S_0
      Append G to $Returns(S_t, A_t)$
      $Q(S_t, A_t) \leftarrow \text{average}(Returns(S_t, A_t))$
      $\pi(S_t) \leftarrow \text{argmax}_a\, Q(S_t, a)$

Without exploring starts, we are not guaranteed to explore the state/action space
– why is this a problem?
– what happens if we never experience certain transitions?

Can we accomplish this without exploring starts?

Yes: create a stochastic (e-greedy) policy

E-Greedy Exploration

Greedy policy:

$$\pi(a|s) = \begin{cases} 1 & \text{if } a = a^* \\ 0 & \text{otherwise} \end{cases}$$

$$a^* = \arg\max_{a \in \mathcal{A}} Q(s, a)$$

E-Greedy policy:

$$\pi(a|s) = \begin{cases} 1 - \epsilon + \frac{\epsilon}{|\mathcal{A}|} & \text{if } a = a^* \\ \frac{\epsilon}{|\mathcal{A}|} & \text{otherwise} \end{cases}$$

E-Greedy Exploration

Greedy policy:

$$\pi(a|s) = \begin{cases} 1 & \text{if } a = a^* \\ 0 & \text{otherwise} \end{cases}$$

$$a^* = \arg\max_{a \in \mathcal{A}} Q(s, a)$$

E-Greedy policy:

$$\pi(a|s) = \begin{cases} 1 - \epsilon + \frac{\epsilon}{|\mathcal{A}|} & \text{if } a = a^* \\ \frac{\epsilon}{|\mathcal{A}|} & \text{otherwise} \end{cases}$$

Action drawn uniformly from $\mathcal{A}$

E-Greedy Exploration

Greedy policy:

$$\pi(a|s) = \begin{cases} 1 \\ 0 \end{cases}$$

Guarantees every state/action will be visited infinitely often

E-Greedy policy:

$$\pi(a|s) = \begin{cases} 1 - \epsilon + \frac{\epsilon}{|\mathcal{A}|} & \text{if } a = a^* \\ \frac{\epsilon}{|\mathcal{A}|} & \text{otherwise} \end{cases}$$

– Notice that this is a stochastic policy (not deterministic).
– This is an example of an *soft* policy
– *soft policy*: all actions in all states have non-zero probability

E-Greedy Exploration

Monte Carlo, ε-greedy exploration:

Algorithm parameter: small $\varepsilon > 0$

Initialize:

$\pi \leftarrow$ an arbitrary ε-soft policy

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

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

Repeat forever (for each episode):

Generate an episode 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, $t = T-1, T-2, \ldots, 0$:

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

Unless the pair S_t, A_t appears in S_0, A_0, S_1, A_1

Append G to $Returns(S_t, A_t)$

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

E-greedy exploration

$A^* \leftarrow \arg\max_a Q(S_t, a)$ (with ties broken arbitrarily)

For all $a \in \mathcal{A}(S_t)$:

$$\pi(a|S_t) \leftarrow \begin{cases} 1 - \varepsilon + \varepsilon/|\mathcal{A}(S_t)| & \text{if } a = A^* \\ \varepsilon/|\mathcal{A}(S_t)| & \text{if } a \neq A^* \end{cases}$$

Off-Policy Methods

- *On-policy* methods evaluate or improve the policy that is used to make decisions.
- *Off-policy* methods evaluate or improve a policy different from that used to generate the data.

- The *target policy* is the policy (π) we wish to evaluate/improve.
- The *behavior policy* is the policy (b) used to generate experiences.

- Coverage:

$$\forall s, a \left[\pi(a|s) > 0 \implies b(a|s) > 0\right]$$

MC Summary

MC methods estimate value function by doing rollouts

Can estimate either the state value function, $V(s)$, or the action value function, $Q(s,a)$

MC Control alternates between policy evaluation and policy improvement

E-greedy exploration explores all possible actions while preferring greedy actions

Off-policy methods update a policy other than the one used to generate experience