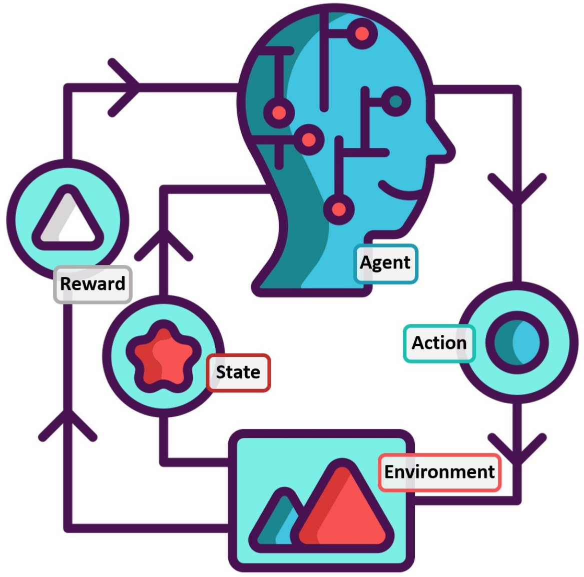
Reinforcement Learning 2025/2026

ARTIFICIAL INTELLIGENCE, MACHINE LEARNING AND CONTROL RESEARCH GROUP

RL2 Reinforcement Learning Research Lab @ UniPD

Lecture #07 Monte Carlo for Control & On-policy vs Off-policy

Gian Antonio Susto



Recap

- Last lecture we finally considered the ‘full’ RL problem: we don’t have the MDP (or the MDP is intractable)

$\mathcal{P}, \mathcal{R}$

known

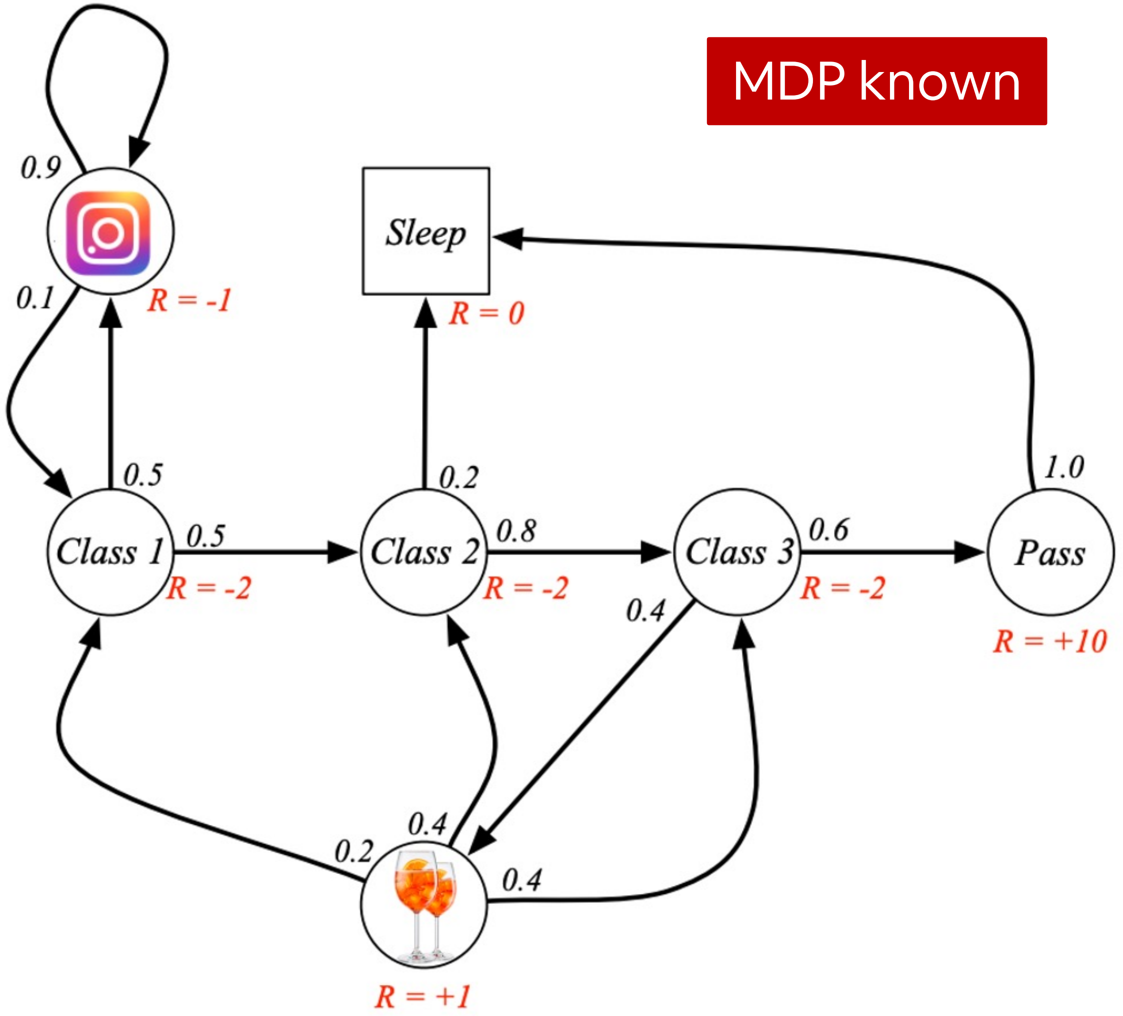
Recap

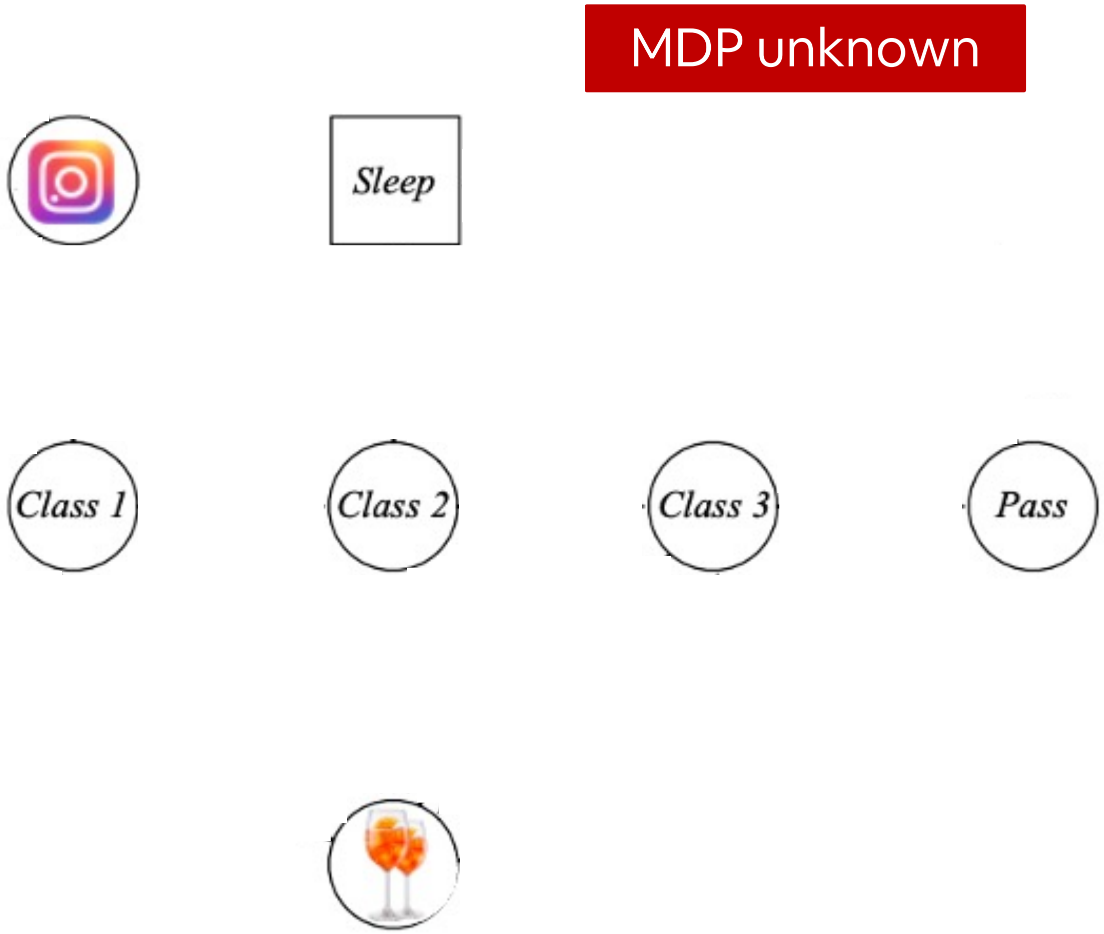
- Last lecture we finally considered the 'full' RL problem: we don't have the MDP (or the MDP is intractable) and we need to resort to data and experience

$\mathcal{P}, \mathcal{R}$ known $\Rightarrow$ Data, experience

$\mathcal{P}, \mathcal{R}$ known

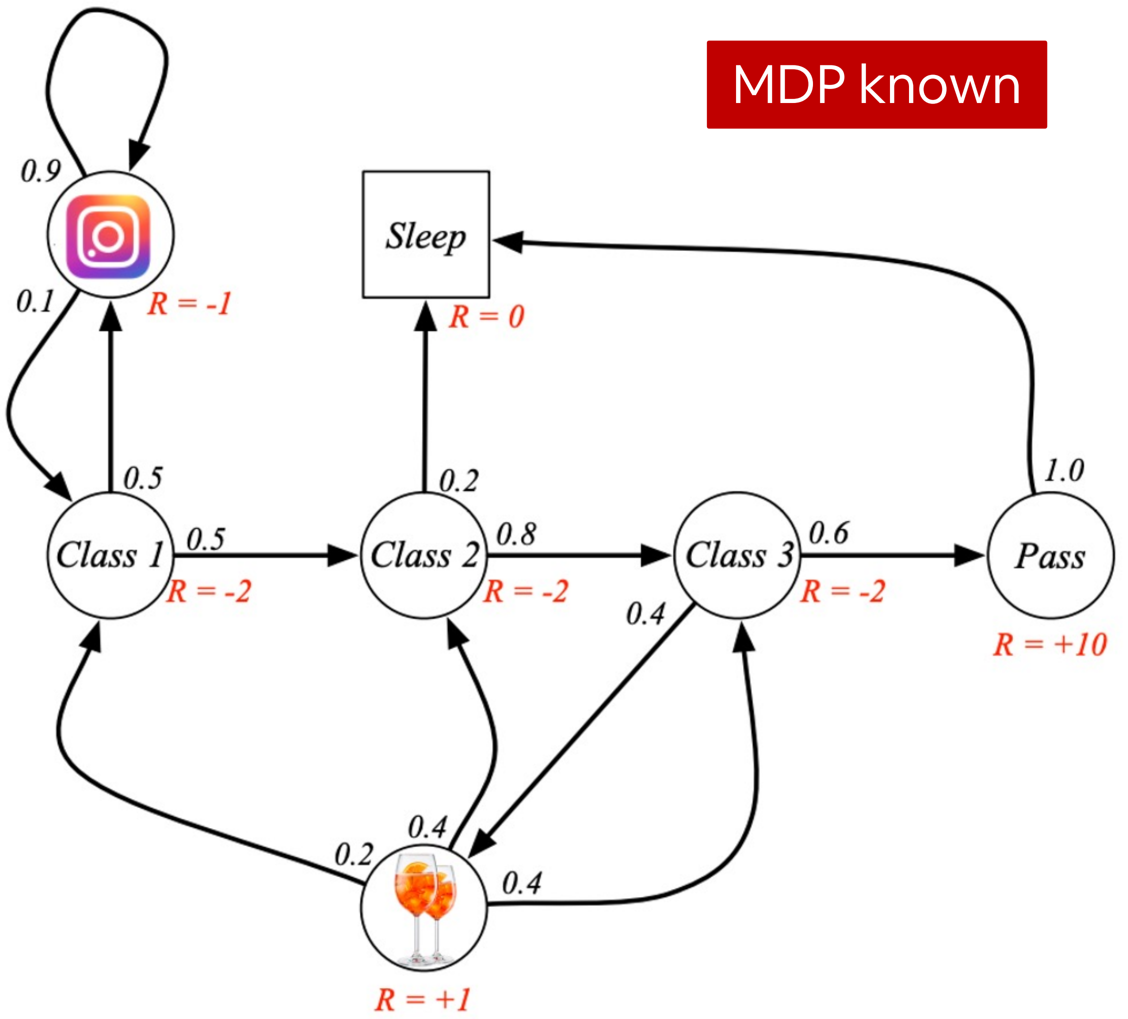
Data, experience

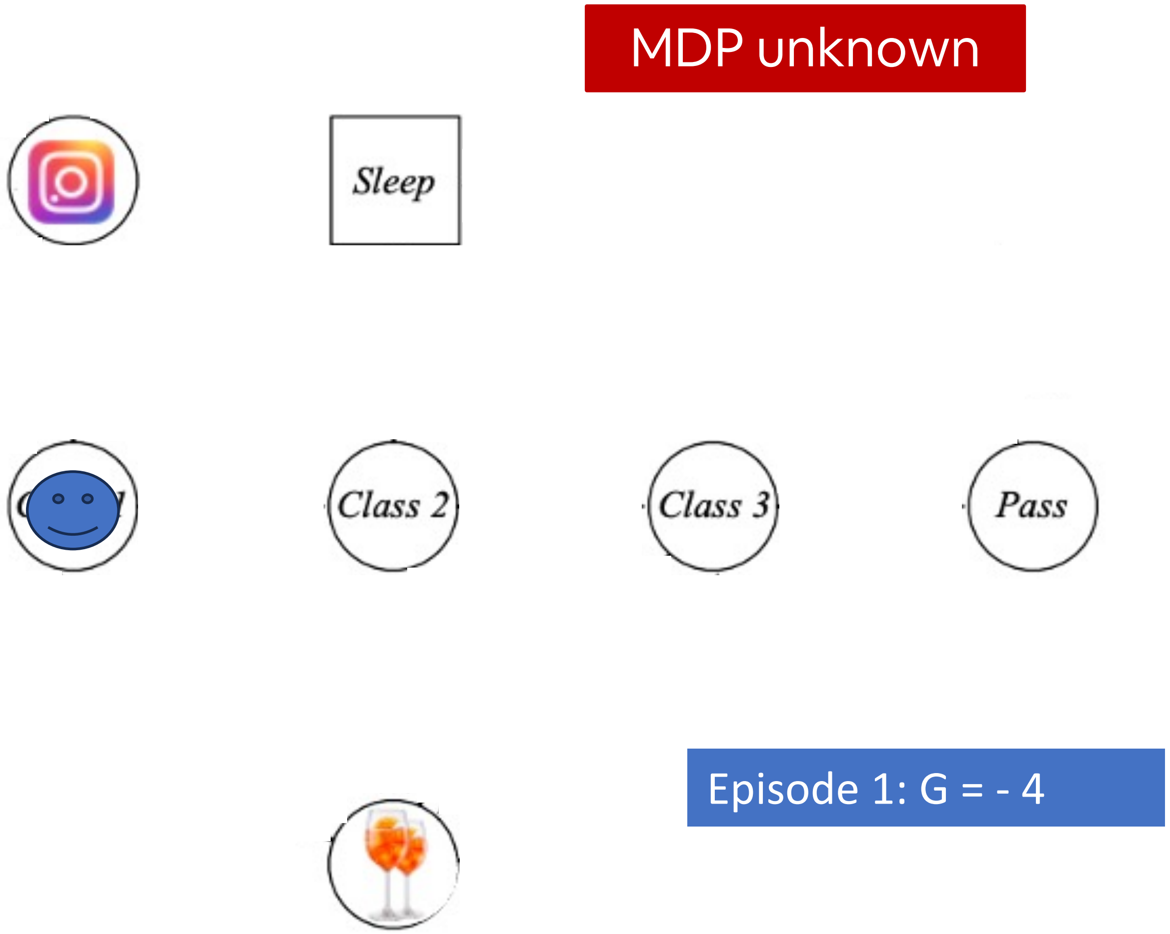




$\mathcal{P}, \mathcal{R}$ known

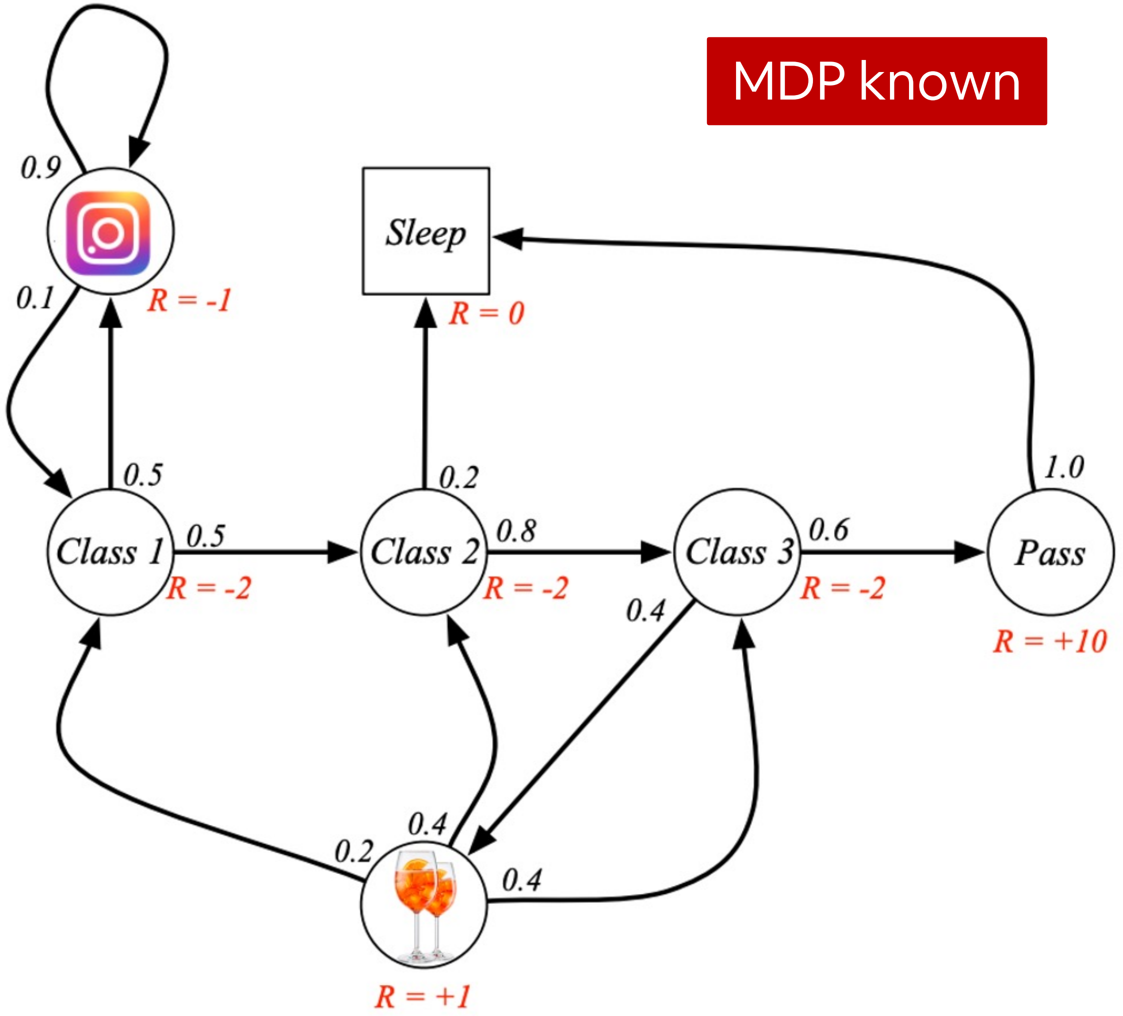
Data, experience





$\mathcal{P}, \mathcal{R}$ known → Data, experience

MDP known



MDP unknown

Episode 1: G = - 4
Episode 2: G = - 6

$\mathcal{P}, \mathcal{R}$
known

Data,
experience

True
expectations

Sample Averages based
on collected data

Recap

- Last lecture we finally considered the ‘full’ RL problem: we don’t have the MDP (or the MDP is intractable) and we need to resort to data and experience

$\mathcal{P}, \mathcal{R}$ known → Data, experience

Pay attention! Even if $\mathcal{P}$ and $\mathcal{R}$ are not known, we will not use data to estimate such quantities!

For efficiency, we will resort to ‘model-free’ RL approaches*, that will estimate instead v and q (later directly π)

*There are also ‘model-based’ RL approaches, we will see – probably – something during last lecture of the course

Recap

- Last lecture we finally considered the 'full' RL problem: we don't have the MDP (or the MDP is intractable) and we need to resort to data and experience

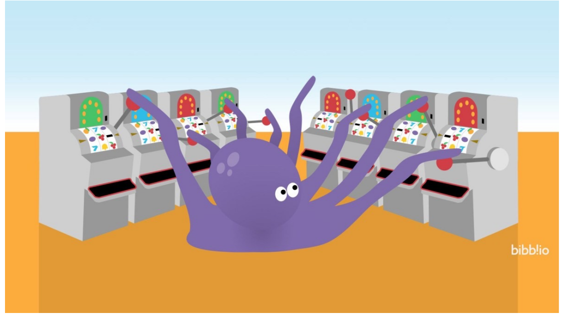
(also in bandits we did not care about learning the true distribution of the rewards!)

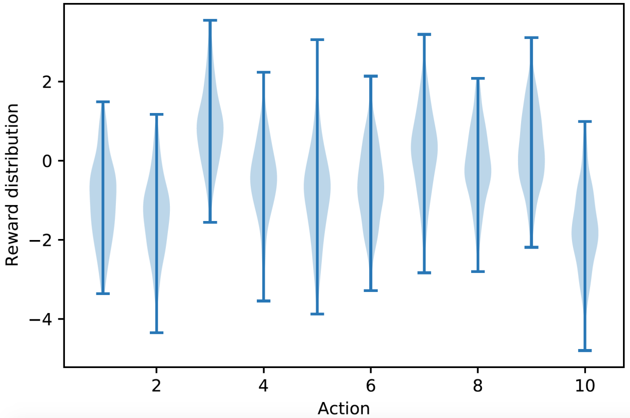
$\mathcal{P}, \mathcal{R}$ known → Data, experience

Pay attention! Even if $\mathcal{P}$ and $\mathcal{R}$ are not known, we will not use data to estimate such quantities!

For efficiency, we will resort to 'model-free' RL approaches*, that will estimate instead v and q (later directly π)

*There are also 'model-based' RL approaches, we will see – probably – something during last lecture of the course

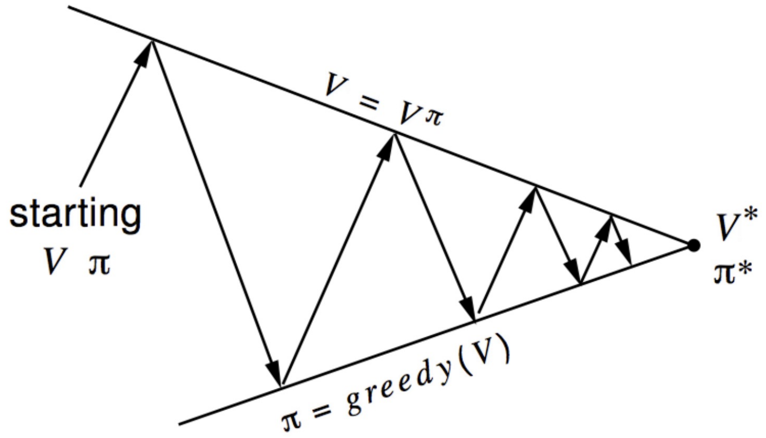




Recap

- Last lecture we finally considered the ‘full’ RL problem: we don’t have the MDP (or the MDP is intractable) and we need to resort to data and experience
- MC methods exploits the fact that expectations can be approximated with averages (work well when many samples are available)
- We have seen prediction, today we’ll see control

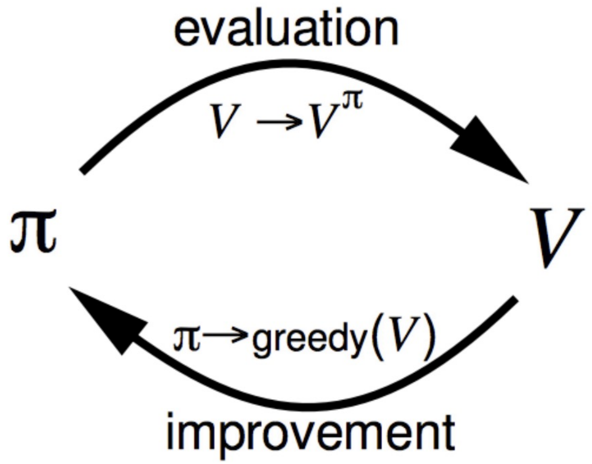
$\mathcal{P}, \mathcal{R}$ known ⇒ Data, experience

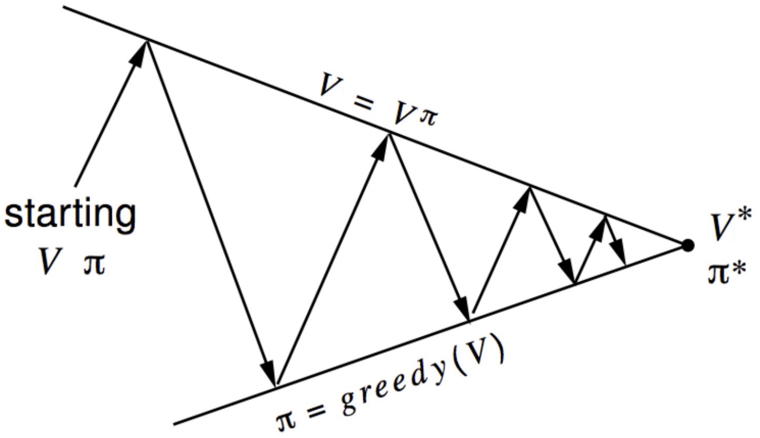


$$v_\pi(s) = \mathbb{E}_\pi[G_t | S_t = s]$$

Control: Model-free Generalized Policy Iteration (GPI)

- We have seen that the GPI approach (iterations between prediction and improvement) can be applied for solving control
- We know how to evaluate v_π with Monte Carlo approaches, so we can try to apply GPI... will that work?

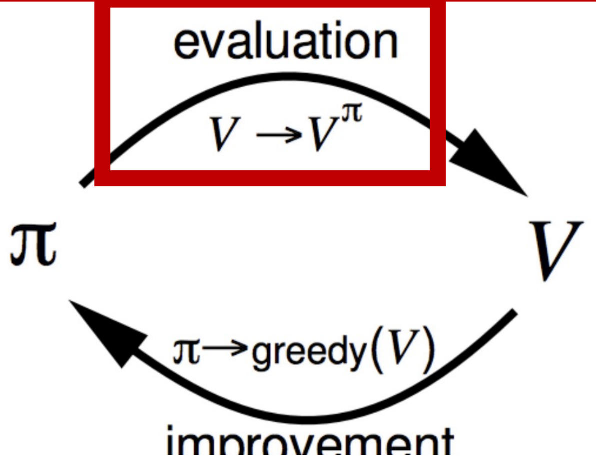


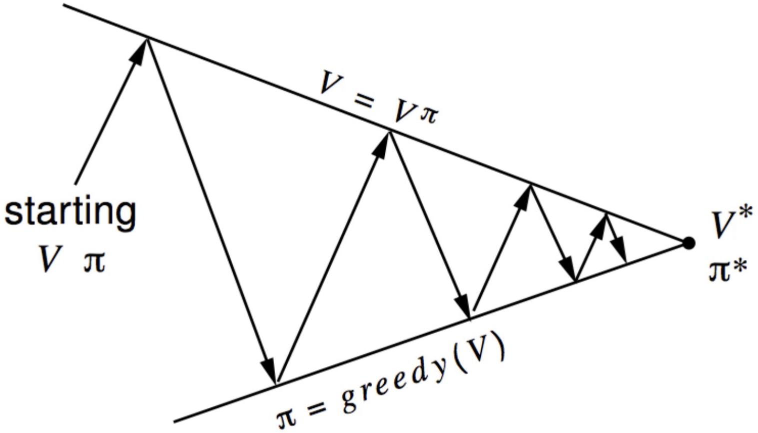


$$v_\pi(s) = \mathbb{E}_\pi[G_t | S_t = s]$$

Control: Model-free Generalized Policy Iteration (GPI)

Hint: we saw this on the previous lecture:





- We have seen that the GPI approach (iterations between prediction and improvement) can be applied for solving control
- We know how to evaluate v_{π} with Monte Carlo approaches, so we can try to apply GPI… will that work?

$$v_{\pi}(s) = \mathbb{E}_{\pi}[G_t | S_t = s]$$

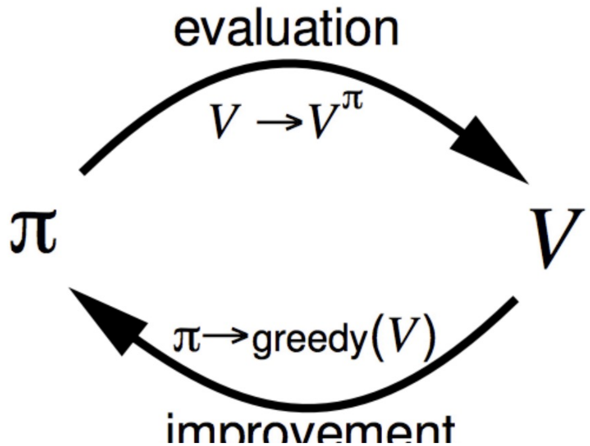
Control: Model-free Generalized Policy Iteration (GPI)

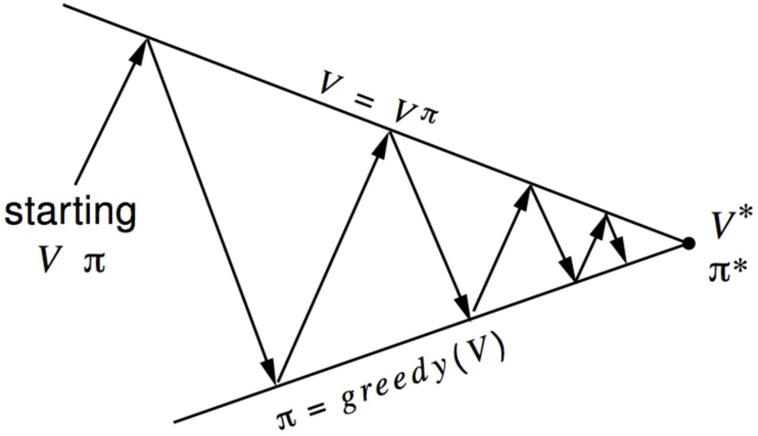
- We have seen that the GPI approach (iterations between prediction and improvement) can be applied for solving control
- We know how to evaluate v_π with Monte Carlo approaches, so we can try to apply GPI... will that work?
- Greedy policy improvement over v_π requires model of MDP:

$$\pi'(s) = \underset{a \in \mathcal{A}}{\text{argmax}}\ \mathcal{R}_s^a + \mathcal{P}_{ss'}^a V(s')$$

- Other ideas?

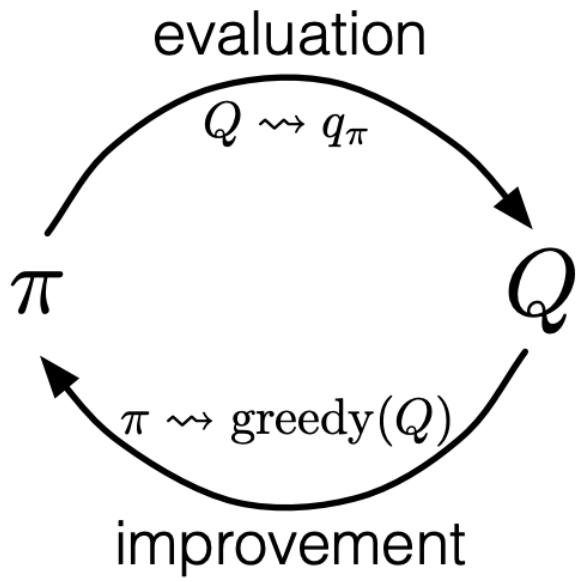
non abbiamo ne R ne P quindi non possiamo usare questa formulazione, possiamo stimare q invece di v





$$v_\pi(s) = \mathbb{E}_\pi[G_t | S_t = s]$$

Control: Model-free Generalized Policy Iteration (GPI)

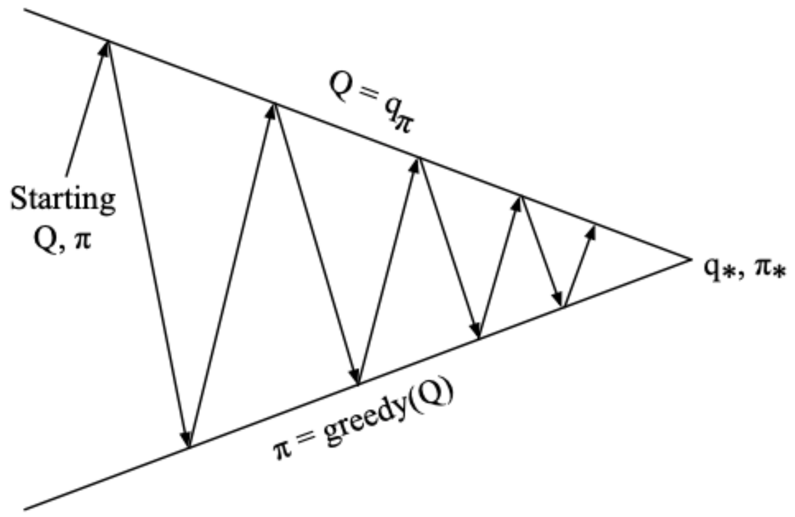


- We can use greedy policy improvement over $Q(s, a)$ instead:

$$\pi'(s) = \underset{a \in \mathcal{A}}{\operatorname{argmax}}\ Q(s, a)$$

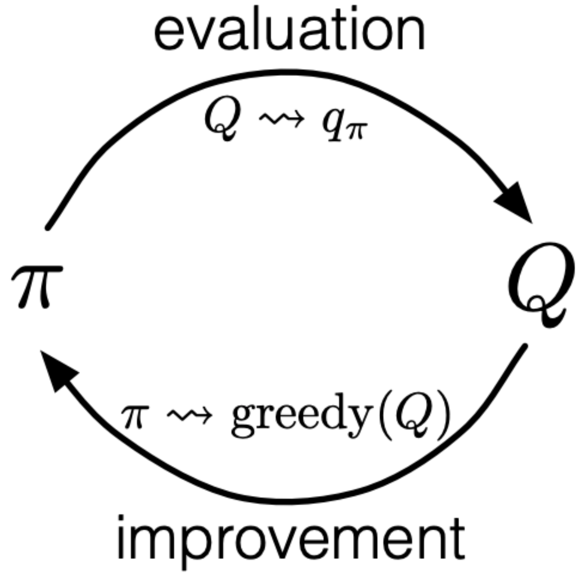
- Greedy policy improvement over $Q(s, a)$ is model-free, while greedy policy improvement over $V(s)$ requires model of MDP:

$$\pi'(s) = \underset{a \in \mathcal{A}}{\operatorname{argmax}}\ \mathcal{R}_s^a + \mathcal{P}_{ss'}^a V(s')$$



$$q_\pi(s, a) = \mathbb{E}_\pi[G_t | S_t = s, A_t =$$

Control: Model-free Generalized Policy Iteration (GPI)



- We can use greedy policy improvement over $Q(s,a)$ instead:

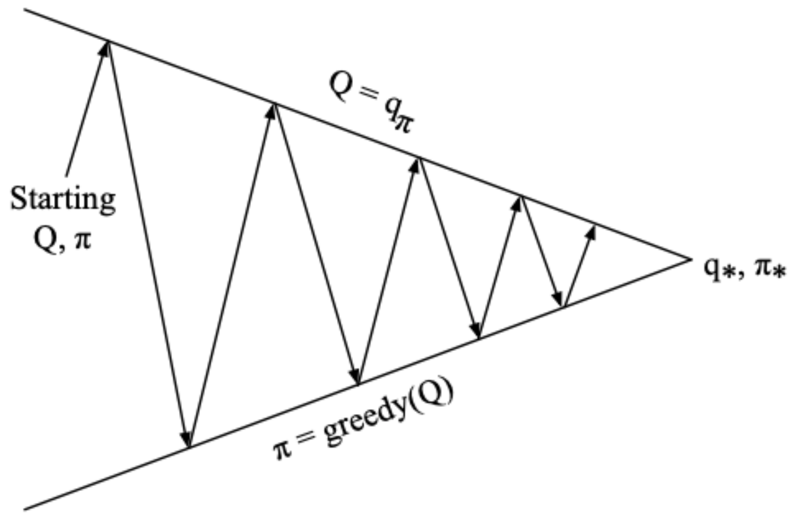
$$\boxed{\pi'(s) = \underset{a \in \mathcal{A}}{\operatorname{argmax}}\ Q(s,a)}$$

Feasible in the full RL problem

- Greedy policy improvement over $Q(s,a)$ is model-free, while greedy policy improvement over $V(s)$ requires model of MDP:

$$\boxed{\pi'(s) = \underset{a \in \mathcal{A}}{\operatorname{argmax}}\ \mathcal{R}_s^a + \mathcal{P}_{ss'}^a V(s')}$$

Not Feasible in the full RL problem

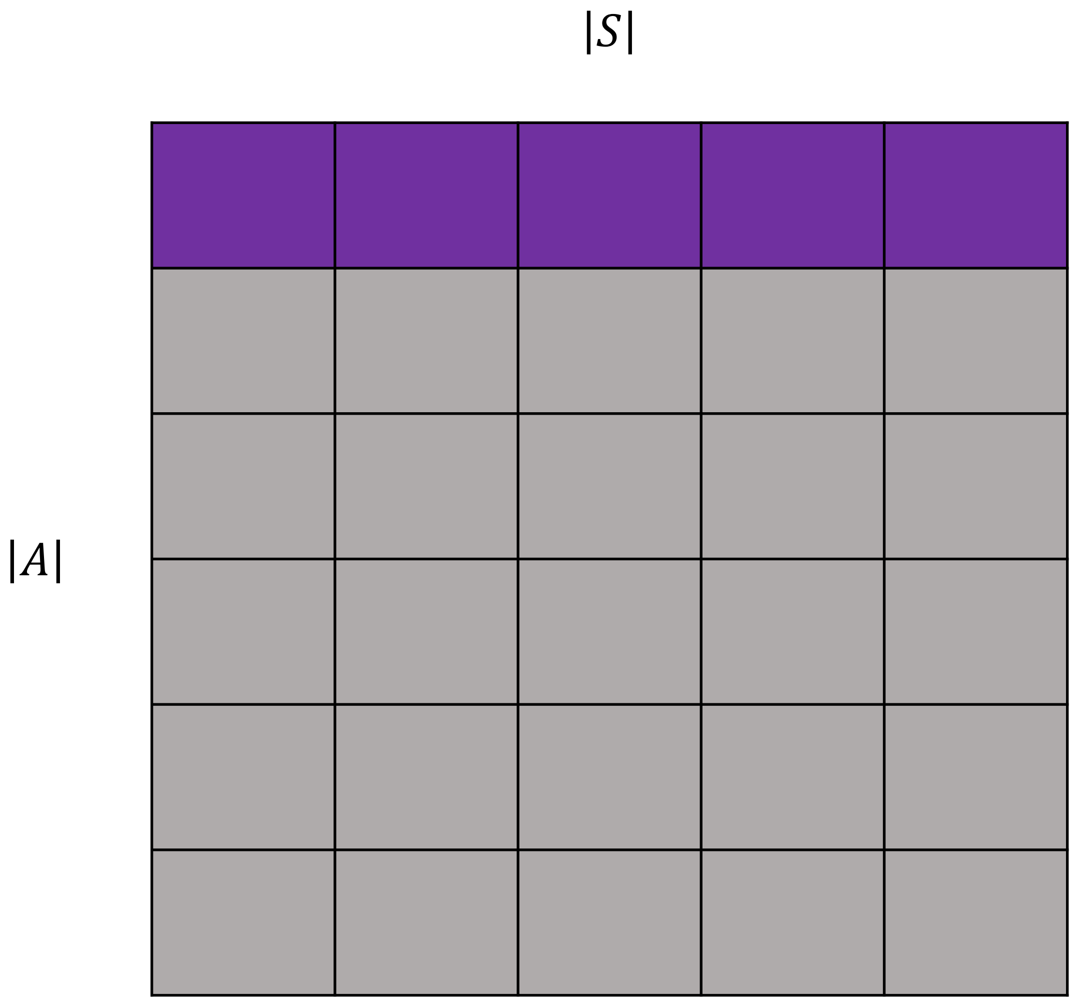


$$q_\pi(s,a) = \mathbb{E}_\pi[G_t | S_t = s, A_t = a$$

Control: Model-free GPI and Exploration

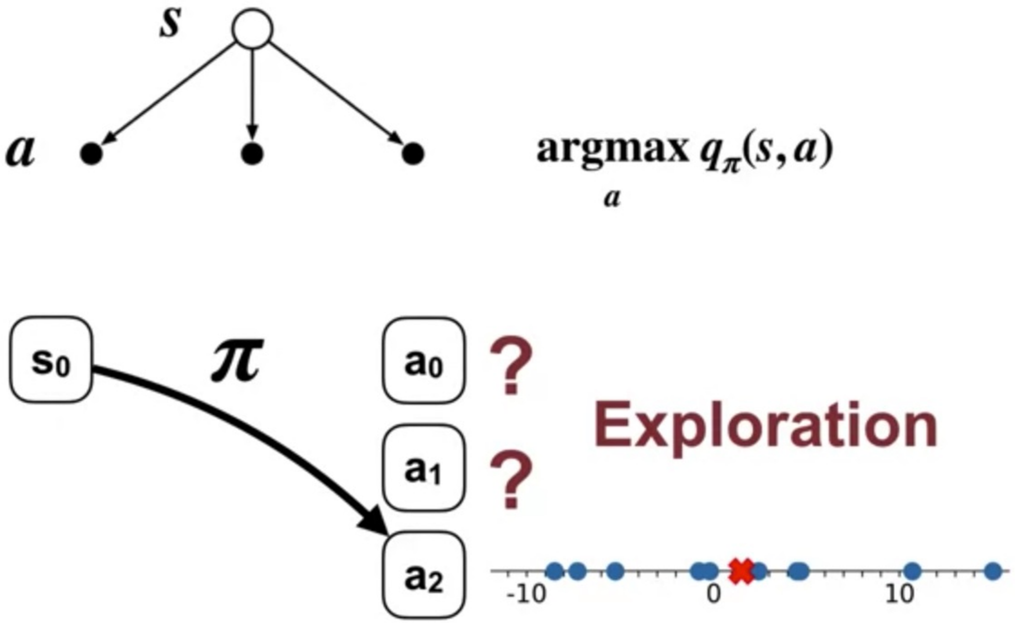
- The (state, action) space is much larger than the state space and we may have several (state, action) pairs unexplored
- In MC methods we need to collect data for each (state, action) pair to ensure reaching optimality, however this is not guaranteed for example with deterministic policies
- We need to adopt strategies that guarantees a certain level of exploration*

* This is true also when estimating v_π, but it is more critical for q_π



Control: Model-free GPI and Exploration

- The (state, action) space is much larger than the state space and we may have several (state, action) pairs unexplored
- In MC methods we need to collect data for each (state, action) pair to ensure reaching optimality, however this is not guaranteed for example with deterministic policies
- We need to adopt strategies that guarantees a certain level of exploration*



* This is true also when estimating v_π, but it is more critical for q_π

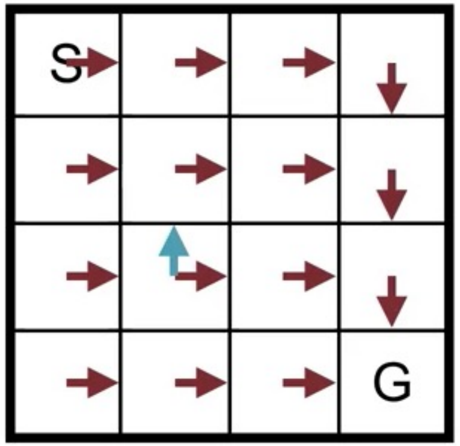
A. White, M. White 'Sample-based Learning Methods'

Control: Exploring Starts (ES)

- A simple strategy to ensure exploration is Exploring Starts (ES)
- With ES we ensure that episodes start in every state-action pair and after the agent follow the current policy
- We can combine ES in GPI to derive our first model-free control algorithm

$s_0, a_0, s_1, a_1, s_2, a_2, \ldots$

Random — From π and p



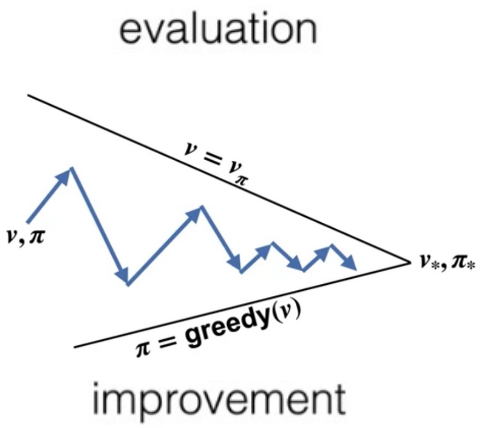
$$\pi_0 \rightarrow \pi_1 \rightarrow \pi_2 \rightarrow \ldots$$

Improvement:

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

Evaluation:

Monte Carlo Prediction



A. White, M. White 'Sample-based Learning Methods'

Monte Carlo ES (Exploring Starts), for estimating $\pi \approx \pi_*$

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 $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$

Initialization

Loop forever (for each episode):

Set of feasible actions in state S0

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 ES (Exploring Starts), for estimating $\pi \approx \pi_*$

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): Pay attention!

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

Set of feasible actions in state S0

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 ES (Exploring Starts), for estimating $\pi \approx \pi_*$

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 $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$

Exploring Starts

Loop forever (for each episode): A(S0) is the set of feasible actions in state S0

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 $Returns(S_t, A_t)$

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

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

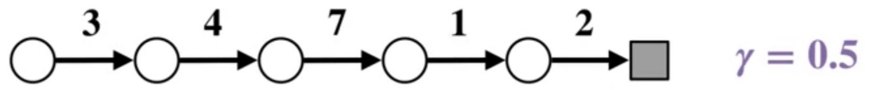
Monte Carlo ES (Exploring Starts), for estimating $\pi \approx \pi_*$

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 $s \in \mathcal{S}$, $a \in \mathcal{A}($



$G_0 = R_1 + \gamma G_1$

$G_1 = R_2 + \gamma G_2$

$G_2 = R_3 + \gamma G_3$

$G_3 = R_4 + \gamma G_4$

$G_4 = R_5 + \gamma G_5$

$G_5 = 0$

Loop forever (for each episode):

Set of feasible actions in state S0

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

Generate an episode from S_0, A_0, following π: S

$G \leftarrow 0$

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

For efficiency

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

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 ES (Exploring Starts), for estimating $\pi \approx \pi_*$

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):

Set of feasible actions in state S0

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

First-visit version!

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

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 ES (Exploring Starts), for estimating $\pi \approx \pi_*$

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):

  Set of feasible actions in state S0

  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 $Returns(S_t, A_t)$

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

      Policy-evaluation

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

Monte Carlo ES (Exploring Starts), for estimating $\pi \approx \pi_*$

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):

Set of feasible actions in state S0

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 $Returns(S_t, A_t)$

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

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

Policy-improvement

Monte Carlo ES (Exploring Starts), for estimating $\pi \approx \pi_*$

Initialize:

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

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

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

Loop forever (for each episode):

Set of feasible actions in state S0

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 $Returns(S_t, A_t)$

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

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

This step could have been implemented incrementally!

Control: Model-free GPI, will it work?

... It will! Following the policy improvement theorem (seen in Dynamic Programming)

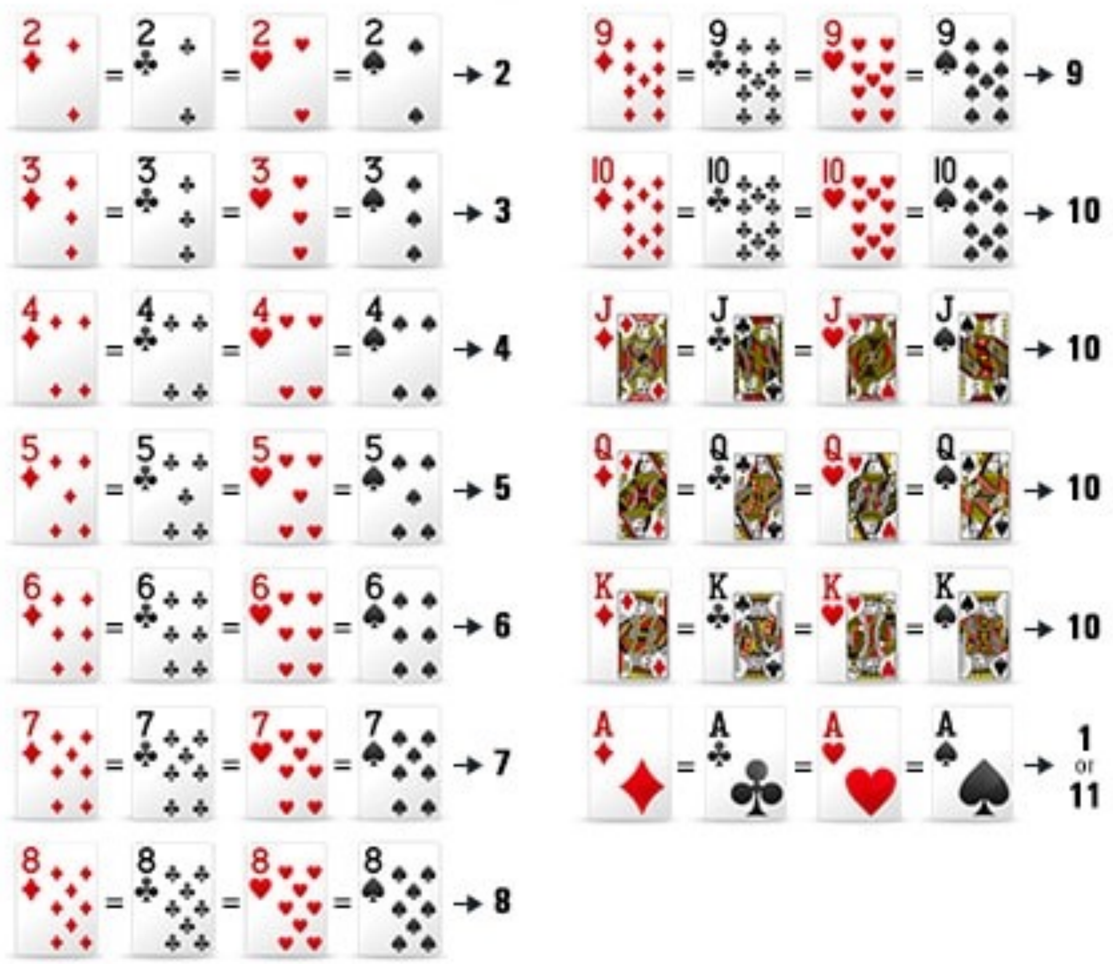
... It will! Following the policy impro
in Dynamic Programming)

For two consecutive policies (π_k and π_{k+1}):

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

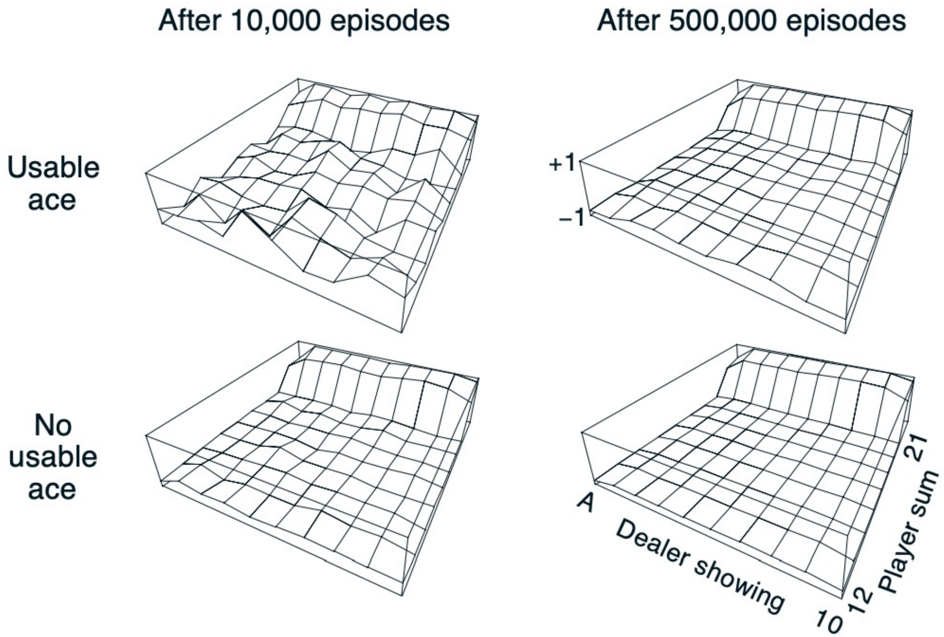
For all states, the next policy will always be better or equal than the previous one.

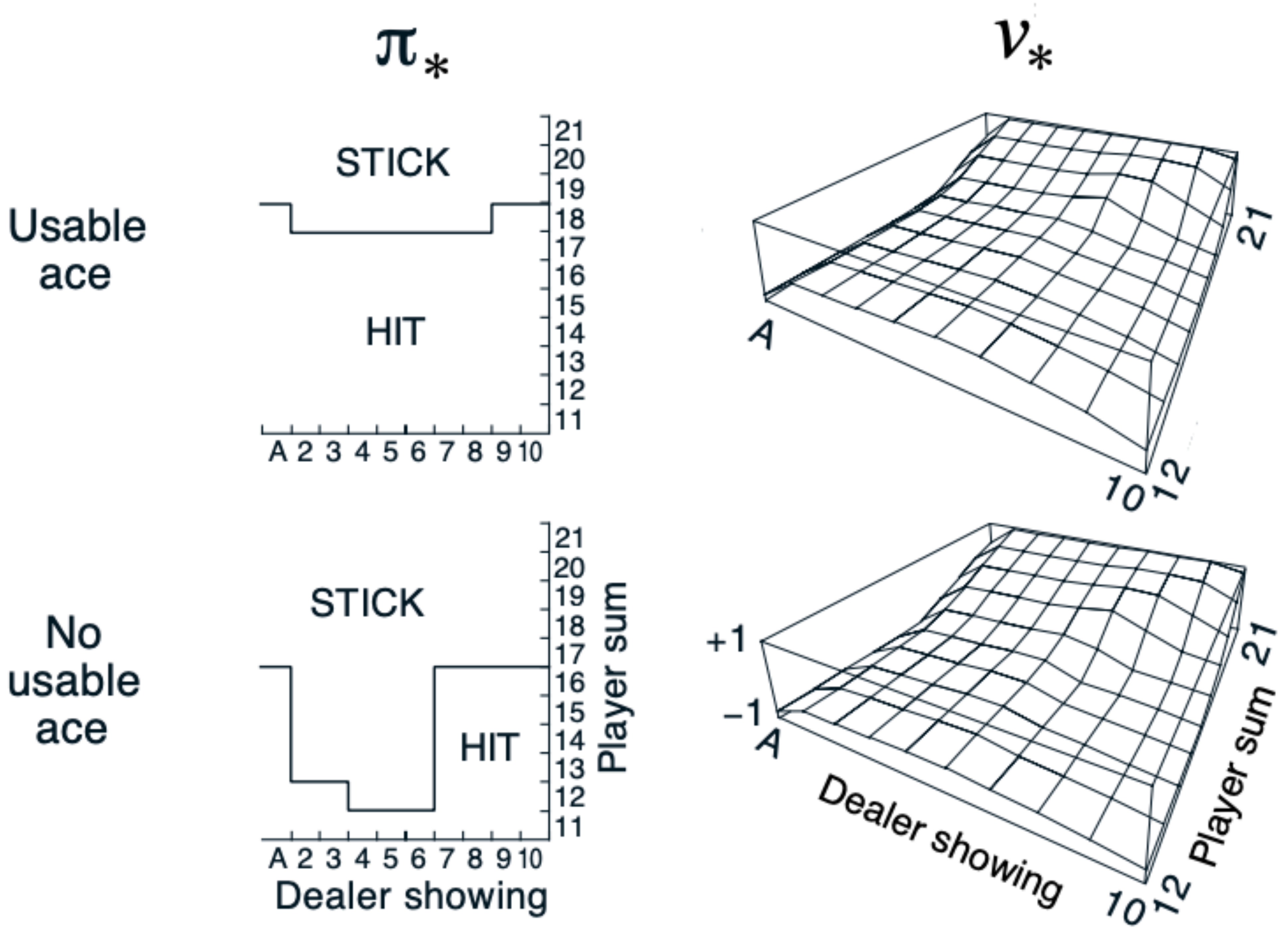
Control: MC prediction - Blackjack



Control: MC prediction - Blackjack

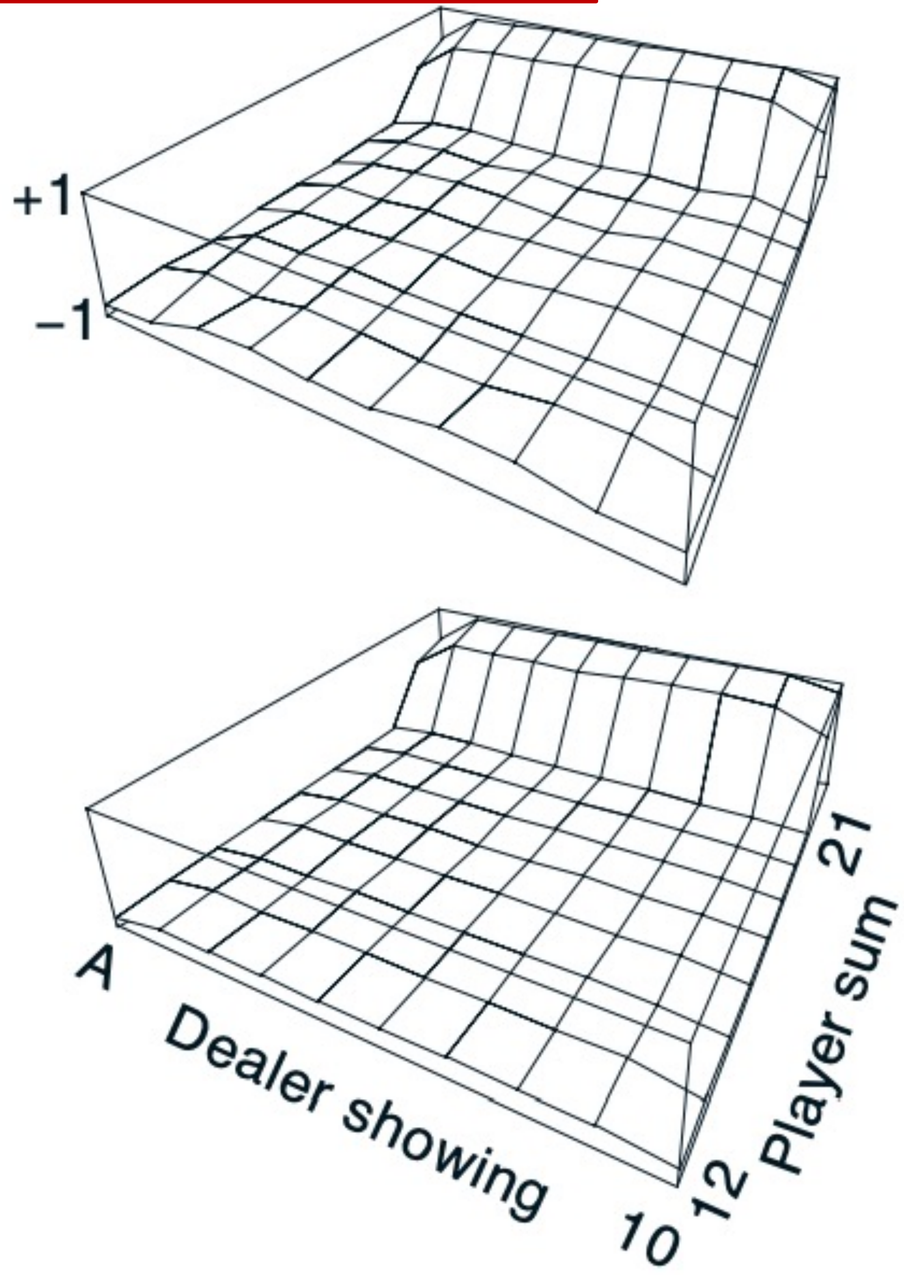
- Same settings (same dealer strategy and deck with replacement)
- Last lecture we have evaluated a fixed deterministic policy (player stick if he/she has 20/21, hit otherwise)
- In Blackjack the initial state is already random, however we need to ensure exploration in the action space



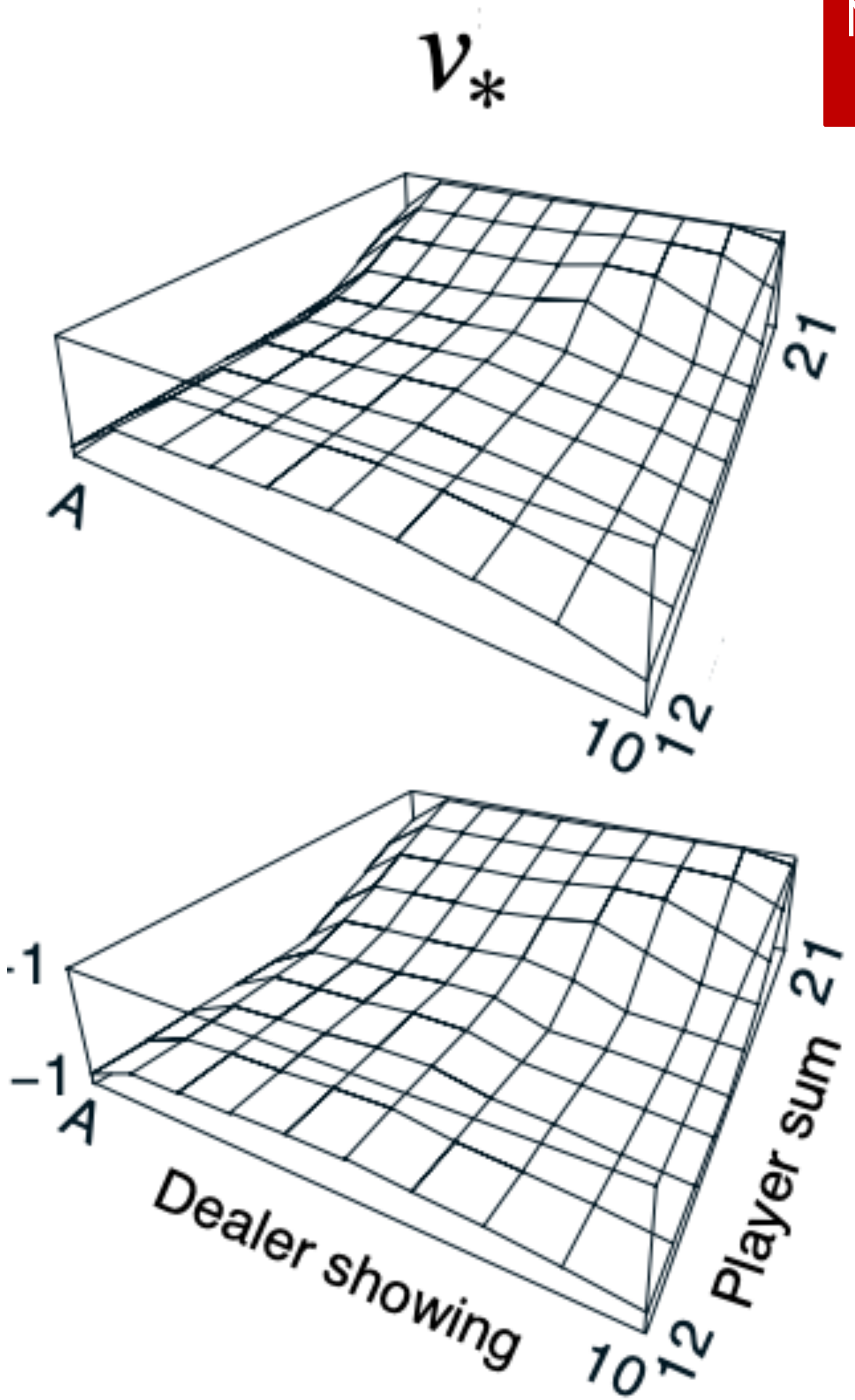


π_*
v_*
Usable ace
STICK
HIT
No usable ace
STICK
HIT
21
20
19
18
17
16
15
14
13
12
11
A 2 3 4 5 6 7 8 9 10
Player sum
Dealer showing
+1
−1
A
10
12
21
Dealer showing
Player sum

Previous lecture policy (hit if below 20, otherwise stick)



Monte Carlo ES policy



Control: is Exploring Starts (ES) always feasible?

Control: is Exploring Starts (ES) always feasible?

- In many scenarios it is impossible to start an episode with all the possible (state, action) pairs
- We will see different approaches to ensure exploration without ES
- Ideas?

Control: ε-greedy and ε-soft policies

- We can resort to stochastic policies like the ε-Greedy Policy that we have seen in k-armed bandits:

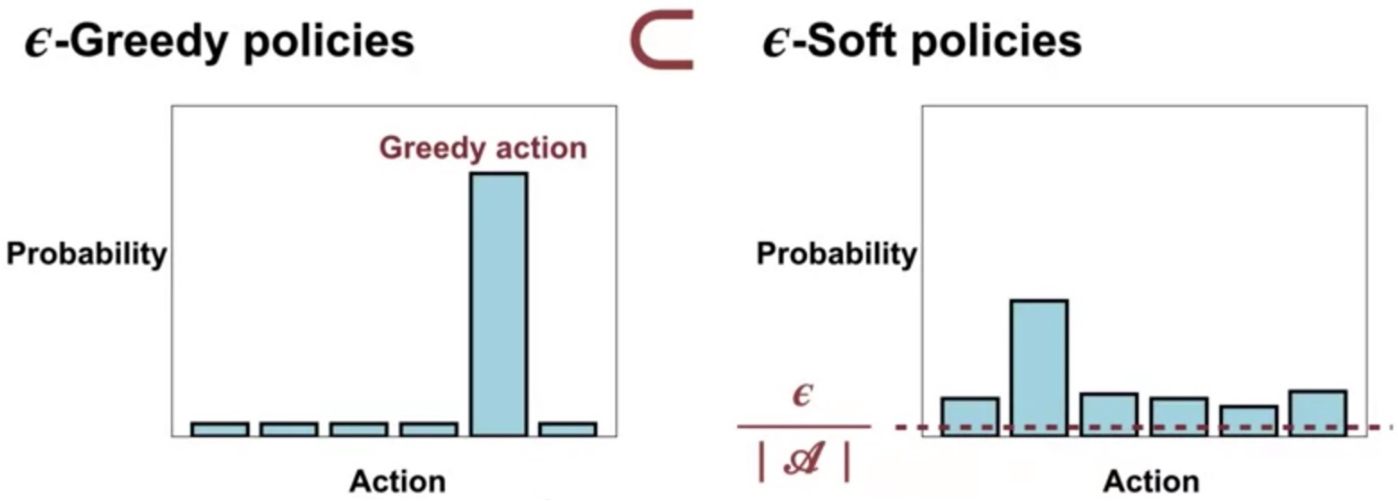
$$A_t \stackrel{\text{def}}{=} \begin{cases} \text{greedy action with probability } 1 - \ \varepsilon \\ \text{non greedy action with probability } \varepsilon \end{cases}$$

Control: ε-greedy and ε-soft policies

- We can resort to stochastic policies like the ε-Greedy Policy that we have seen in k-armed bandits:

$$A_t \stackrel{\text{def}}{=} \begin{cases} \text{greedy action with probability } 1 - \varepsilon \\ \text{non greedy action with probability } \varepsilon \end{cases}$$

- More in general we can resort to soft policies, ie. stochastic policies that have $\pi(a|s) > 0$ for all a and s.
- We define ε-soft policies the soft policies that have $\pi(a|s) \geq \frac{\varepsilon}{|A(s)|}$ for all a and s, for some ε > 0. ε-greedy is the 'greediest' ε-soft policy



A. White, M. White 'Sample-based Learning Methods'

On-policy first-visit MC control (for ε-soft policies), estimates $\pi \approx \pi_*$

Differences w.r.t. ES MC Control

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 $S_0, A_0, S_1, A_1 \ldots, S_{t-1}, A_{t-1}$:

Append G to $Returns(S_t, A_t)$

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

$A^* \leftarrow \text{argmax}_a Q(S_t, a)$ (with ties broken arbitrarily)

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

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

On-policy first-visit MC control (for ε-soft policies), estimates $\pi \approx \pi_*$

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 $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$

Differences w.r.t. ES MC Control

Initialization
ε governs the trade-off between exploration and exploitation

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 $S_0, A_0, S_1, A_1 \ldots, S_{t-1}, A_{t-1}$:

Append G to $Returns(S_t, A_t)$

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

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

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

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

On-policy first-visit MC control (for ε-soft policies), estimates $\pi \approx \pi_*$

Differences w.r.t. ES MC Control

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: $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$

We don't need Exploring Starts

  $G \leftarrow 0$

  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 $Returns(S_t, A_t)$

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

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

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

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

On-policy first-visit MC control (for ε-soft policies), estimates $\pi \approx \pi_*$

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: $S_0, A_0, R_1, \ldots, S_{T}$

$G \leftarrow 0$

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 $S_0, A_0, S_1, A_1 \ldots, S_{t-1}, A_{t-1}$:

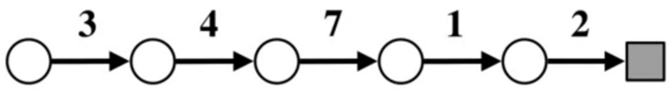
Append G to $Returns(S_t, A_t)$

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

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

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

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



$\gamma = 0.5$

$G_0 = R_1 + \gamma G_1$

$G_1 = R_2 + \gamma G_2$

$G_2 = R_3 + \gamma G_3$

$G_3 = R_4 + \gamma G_4$

$G_4 = R_5 + \gamma G_5$

$G_5 = 0$

For efficiency

On-policy first-visit MC control (for ε-soft policies), estimates $\pi \approx \pi_*$

Differences w.r.t. ES MC Control

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

First-visit version!

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

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 $Returns(S_t, A_t)$

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

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

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

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

On-policy first-visit MC control (for ε-soft policies), estimates $\pi \approx \pi_*$

Differences w.r.t. ES MC Control

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 $S_0, A_0, S_1, A_1 \ldots, S_{t-1}, A_{t-1}$:

Append G to $Returns(S_t, A_t)$

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

Policy-evaluation

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

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

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

On-policy first-visit MC control (for ε-soft policies), estimates $\pi \approx \pi_*$

Differences w.r.t. ES MC Control

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 $S_0, A_0, S_1, A_1 \ldots, S_{t-1}, A_{t-1}$:

Append G to $Returns(S_t, A_t)$

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

Policy-improvement

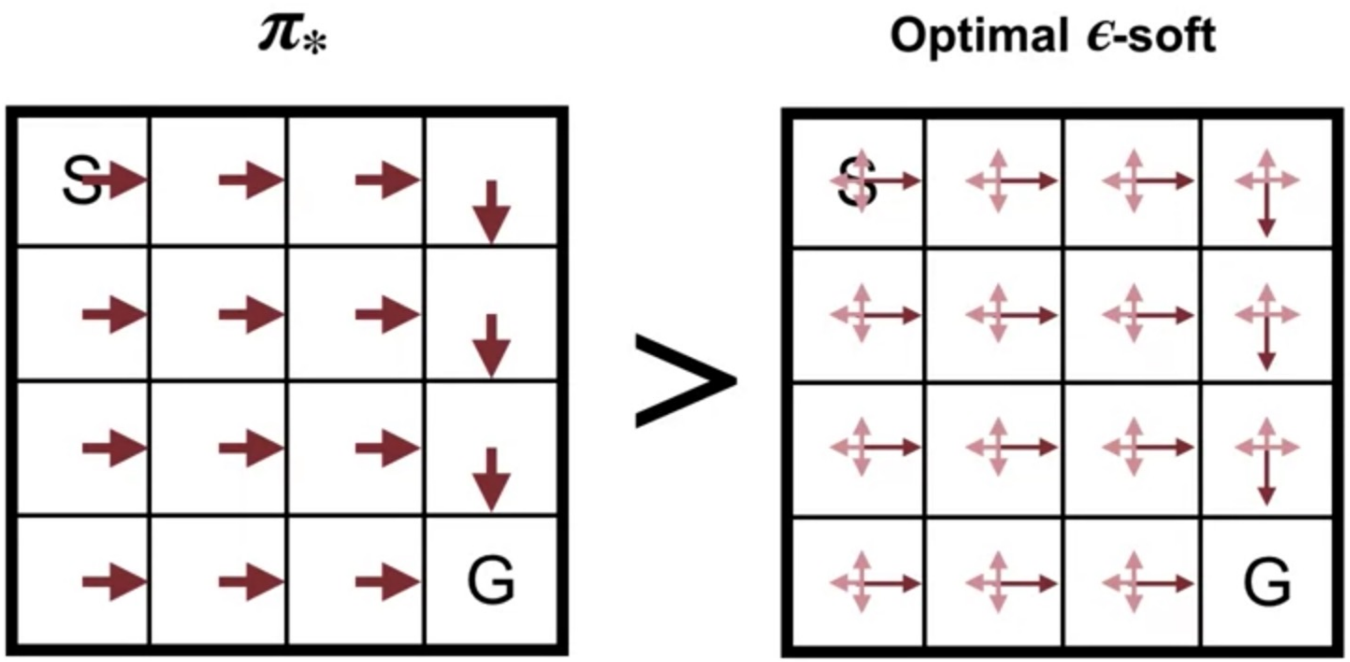
$A^* \leftarrow \text{argmax}_a Q(S_t, a)$ (with ties broken arbitrarily)

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

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

Control: ε-soft policies are not optimal!

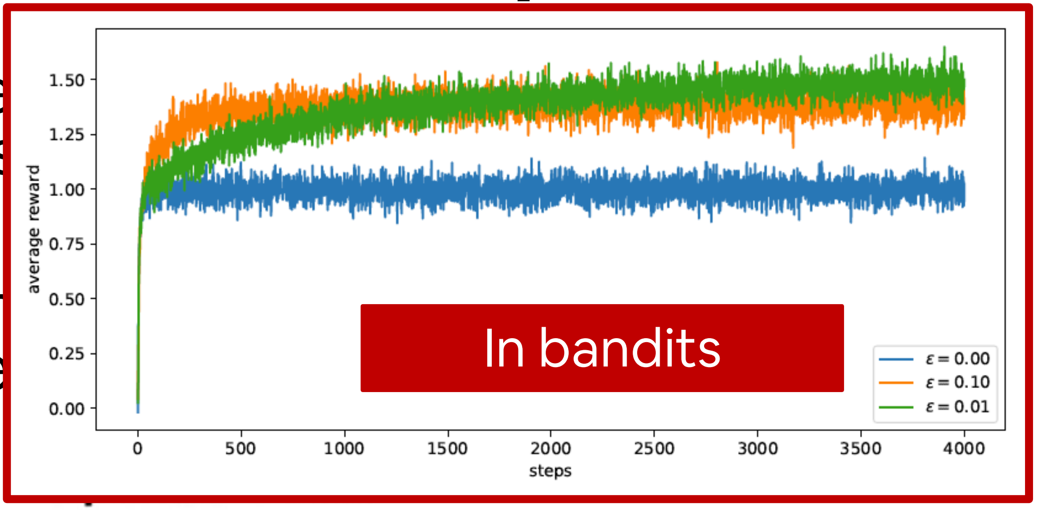
- It can be shown (policy improvement theorem) that the previous algorithm allow improvement (optional, see the book)
- ε-soft policies are not optimal!
- However, they may work quite well and may represent a feasible option when ES cannot be applied (which is true in many cases!)

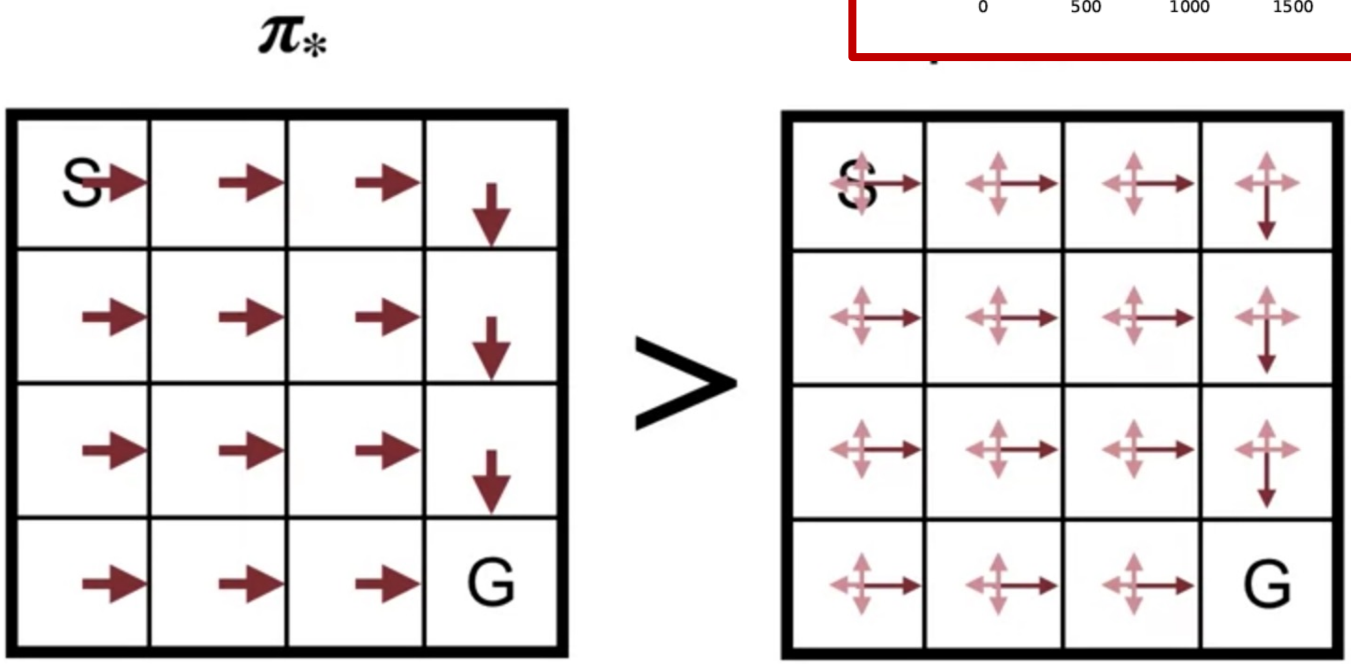


A. White, M. White 'Sample-based Learning Methods'

Control: ε-soft policies are not optimal!

- It can be shown (policy improvement the
  algorithm allow improvement (optional, s
- ε-soft policies are not optimal!
- However, they may work quite well and r
  when ES cannot be applied (which is true

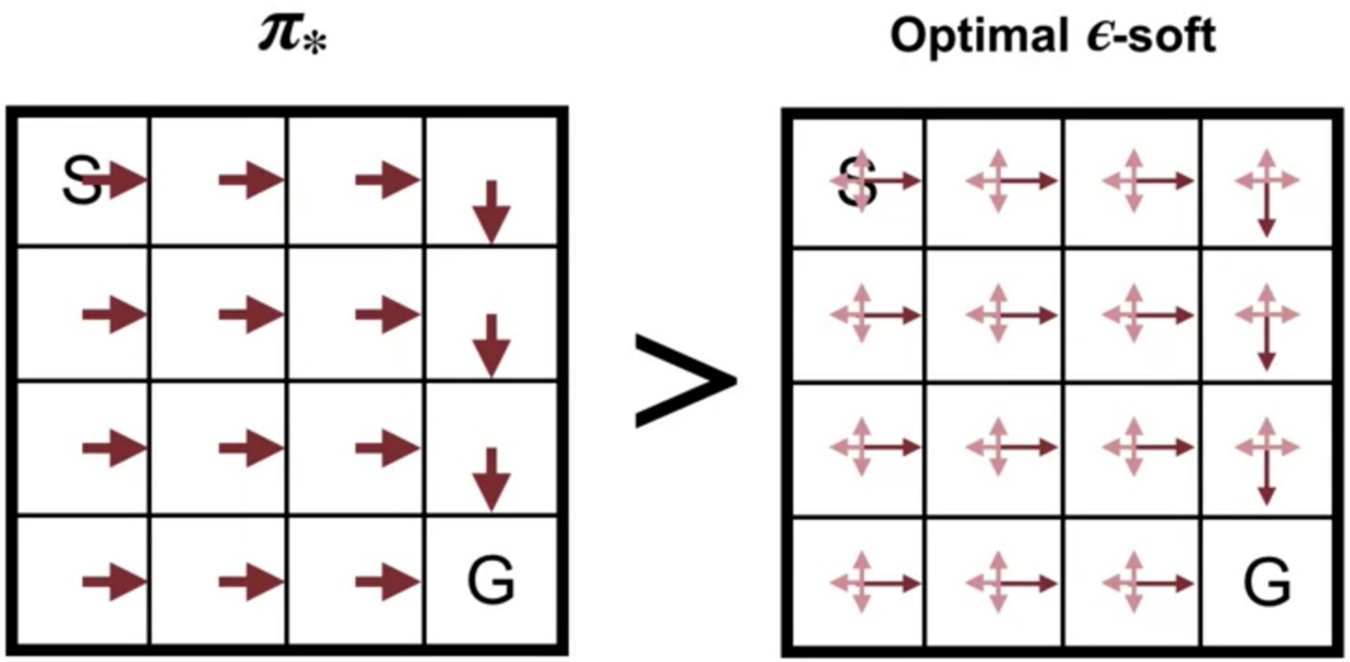




A. White, M. White 'Sample-based Learning Methods'

Control: ε-soft policies are not optimal!

- It can be shown (policy improvement theorem) that the previous algorithm allow improvement (optional, see the book)
- ε-soft policies are not optimal!
- However, they may work quite well and may represent a feasible option when ES cannot be applied (which is true in many cases!)

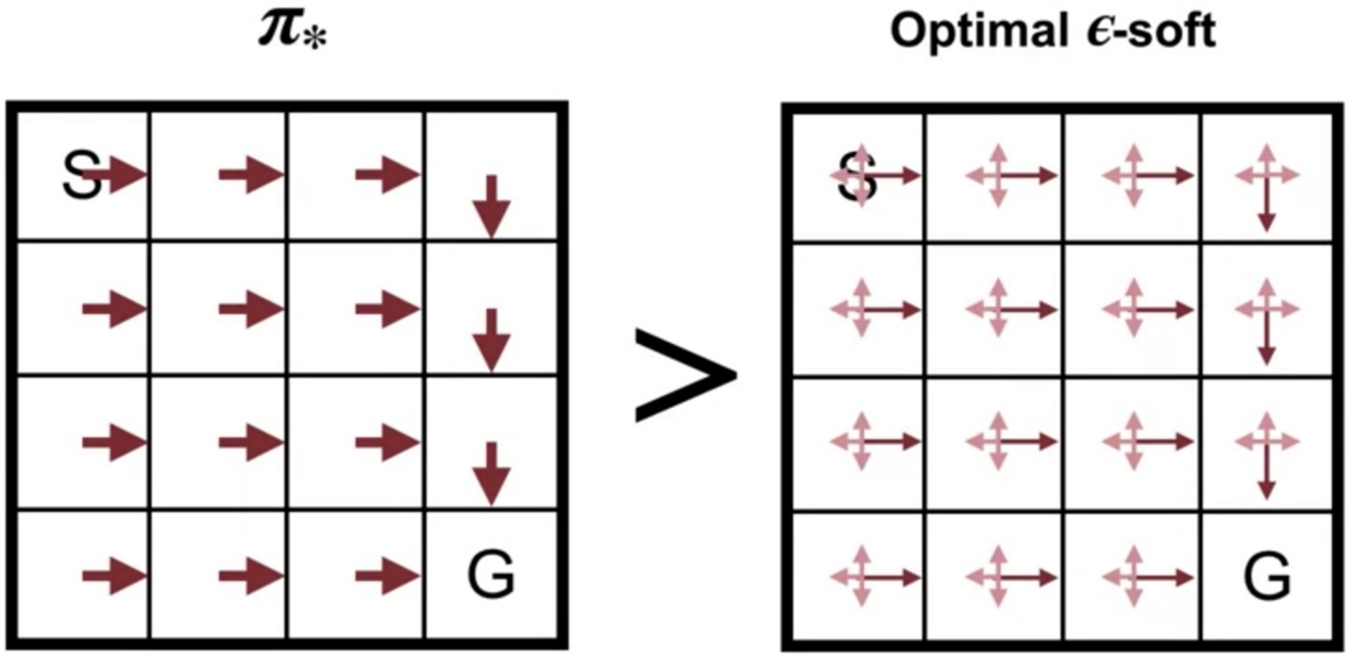


Can we do better?

A. White, M. White 'Sample-based Learning Methods'

Control: ε-soft policies are not optimal!

- It can be shown (policy improvement theorem) that the previous algorithm allow improvement (optional, see the book)
- ε-soft policies are not optimal!
- However, they may work quite well and may represent a feasible option when ES cannot be applied (which is true in many cases!)



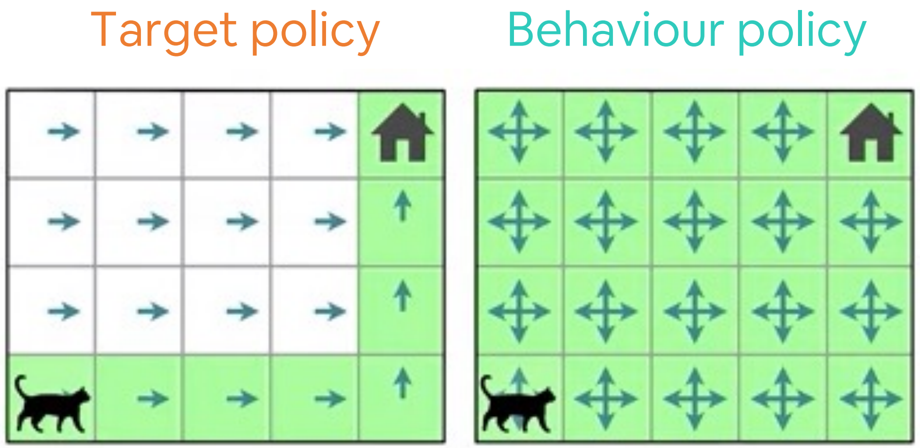
Can we do better?

Disclaimer: the following approaches will work especially with TD-learning (chapter 6)

A. White, M. White 'Sample-based Learning Methods'

Control: On-policy vs. Off-policy learning

- What we have seen so far are on-policy learning
- On-policy ('learning on the job'): learn about policy π from experience sampled from policy π
- Off-policy ('look over someone's shoulder'): learn about policy π (target policy) from experience sampled from policy b (behaviour policy)
- In off-policy we use the behaviour policy to collect data from the environment in order to evalute an optimal policy



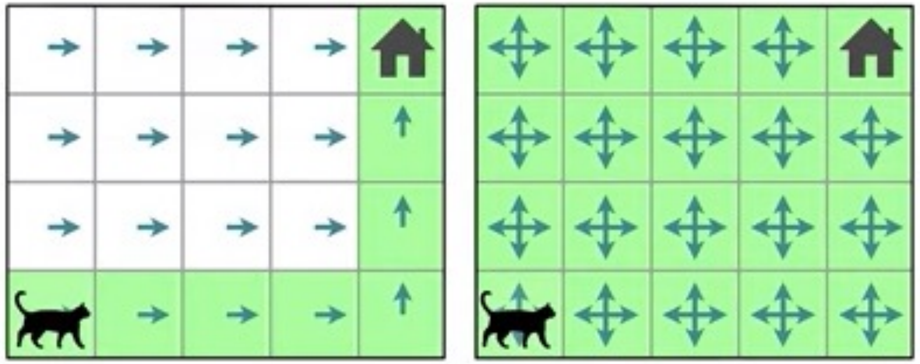
A. White, M. White 'Sample-based Learning Methods'

Control: On-policy vs. Off-policy learning

- What we have seen so far are on-policy learning
- On-policy ('learning on the job'): learn about policy π from experience sampled from policy π
- Off-policy ('look over someone's shoulder'): learn about policy π (target policy) from experience sampled from policy b (behaviour policy)
- In off-policy we use the behaviour policy to collect data from the environment in order to evalute an optimal policy

Why using a behaviour policy?

Target policy Behaviour policy

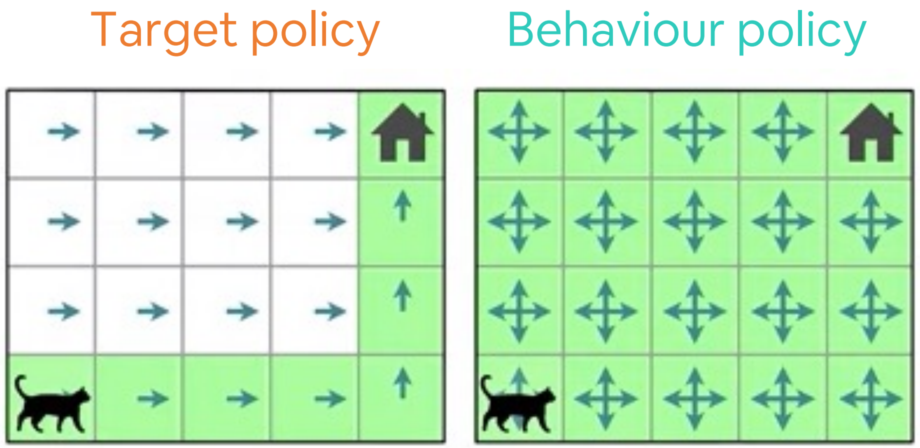


A. White, M. White 'Sample-based Learning Methods'

Control: On-policy vs. Off-policy learning

- What we have seen so far are on-policy learning
- On-policy ('learning on the job'): learn about policy π from experience sampled from policy π
- Off-policy ('look over someone's shoulder'): learn about policy π (target policy) from experience sampled from policy b (behaviour policy)
- In off-policy we use the behaviour policy to collect data from the environment in order to evalute an optimal policy

Why using a behaviour policy?
- We can leave the exploration to the behaviour policy and reach optimal policies!

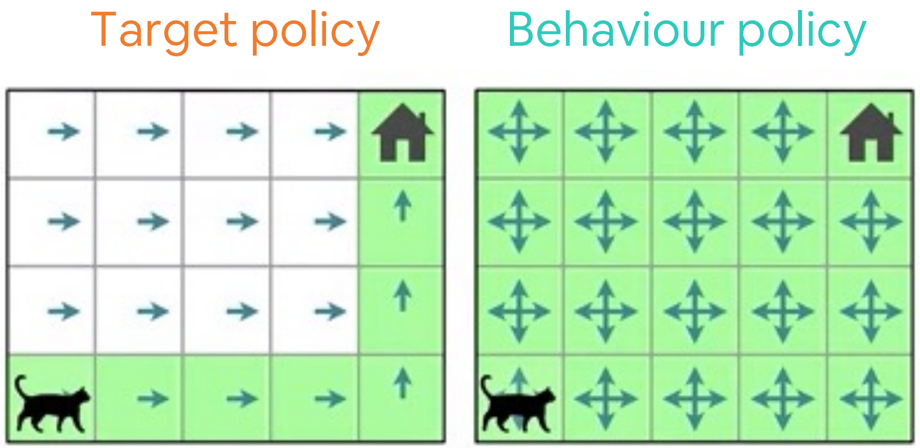


A. White, M. White 'Sample-based Learning Methods'

Control: On-policy vs. Off-policy learning

- What we have seen so far are on-policy learning
- On-policy ('learning on the job'): learn about policy π from experience sampled from policy π
- Off-policy ('look over someone's shoulder'): learn about policy π (target policy) from experience sampled from policy b (behaviour policy)
- In off-policy we use the behaviour policy to collect data from the environment in order to evalute an optimal policy

Why using a behaviour policy?
- We can leave the exploration to the behaviour policy and reach optimal policies!
- We may want to gain experience from observing other agents (or humans)

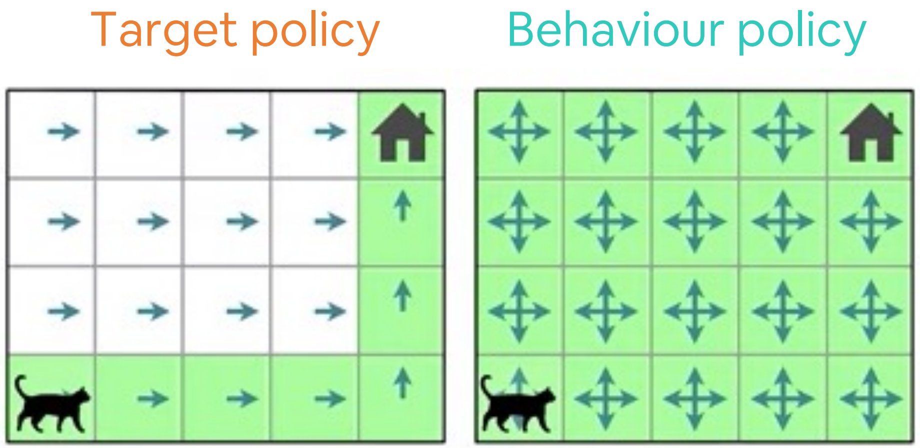


A. White, M. White 'Sample-based Learning Methods'

Control: On-policy vs. Off-policy learning

- What we have seen so far are on-policy learning
- On-policy ('learning on the job'): learn about policy π from experience sampled from policy π
- Off-policy ('look over someone's shoulder'): learn about policy π (target policy) from experience sampled from policy b (behaviour policy)
- In off-policy we use the behaviour policy to collect data from the environment in order to evalute an optimal policy

Why using a behaviour policy?
- We can leave the exploration to the behaviour policy and reach optimal policies!
- We may want to gain experience from observing other agents (or humans)
- Re-use experience generated from old policies $\pi_1, \pi_2, \pi_3, \ldots, \pi_{t-1}$



A. White, M. White 'Sample-based Learning Methods'

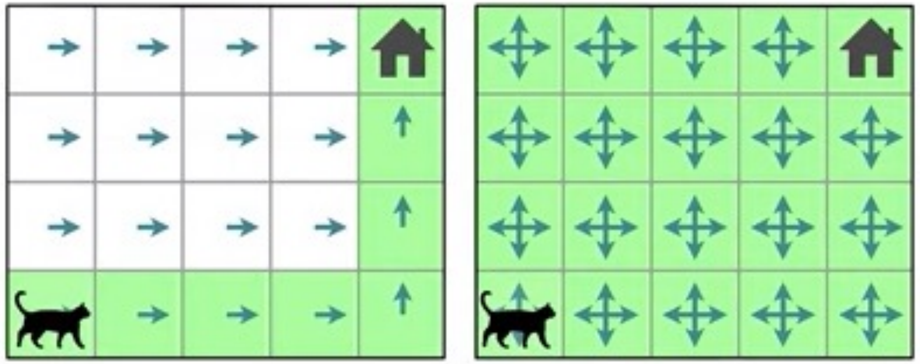
Control: On-policy vs. Off-policy learning

- What we have seen so far are on-policy learning
- On-policy ('learning on the job'): learn about policy π from experience sampled from policy π
- Off-policy ('look over someone's shoulder'): learn about policy π (target policy) from experience sampled from policy b (behaviour policy)
- In off-policy we use the behaviour policy to collect data from the environment in order to evalute an optimal policy

Why using a behaviour policy?

- We can leave the exploration to the behaviour policy and reach optimal policies!
- We may want to gain experience from observing other agents (or humans)
- Re-use experience generated from old policies $\pi_1, \pi_2, \pi_3, \ldots, \pi_{t-1}$
- We can learn multiple policies while following one policy

Target policy Behaviour policy



A. White, M. White 'Sample-based Learning Methods'

Prediction/Control: Off-policy learning - Coverage

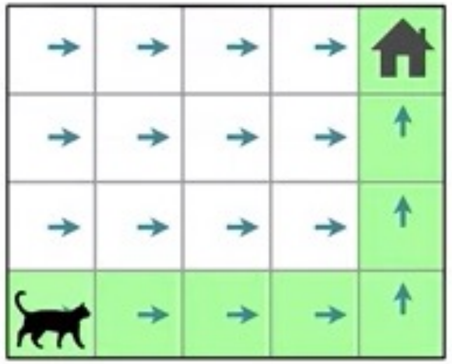
- If we resort to off-policy learning, we must ensure

(i) Knowledge of $b(a|s)$

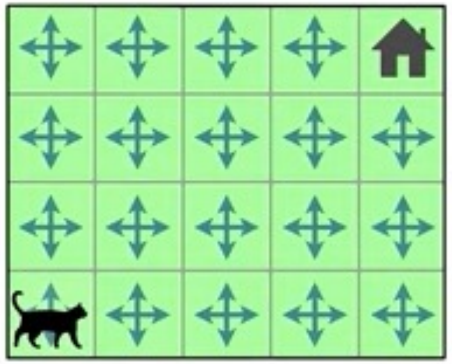
(ii) Coverage: if $\pi(a|s) > 0 \Longrightarrow b(a|s) > 0$

- Coverage means that we must ensure that the behaviour policy take explore the (state, action) pair that have non-zero probabilty in the target policy
- If this do not happen, the agent cannot learn the correct action value for that state because it never observed samples of what would happen.

Target policy



Behaviour policy



Prediction/Control: Off-policy learning – Importance Sampling

- Almost all off-policy methods utilize importance sampling: a general technique for estimating expected values under one distribution given samples from another.

DEFINITION. The *expectation* of a discrete random variable X taking the values $a_1, a_2, \ldots$ and with probability mass function p is the number

$$\mathrm{E}[X] = \sum_i a_i \mathrm{P}(X = a_i) = \sum_i a_i p(a_i).$$

F.M. Dekking et al. 'A Modern Introduction to Probability and Statistics'

- In off-policy we have:

Sample: $x \sim b$

Estimate: $\mathbb{E}_\pi[X] = \sum_{x \in X} x\pi(x) = \sum_{x \in X} x\pi(x)\frac{b(x)}{b(x)} = \sum_{x \in X} x\rho(x)b(x)$

Prediction/Control: Off-policy learning – Importance Sampling

- Almost all off-policy methods utilize importance sampling: a general technique for estimating expected values under one distribution given samples from another.

- Almost all off-policy methods utilize importance sampling: a gene technique for estimating expected values under one distribution given samples from another.

- In off-policy we have:

Sample: $x \sim b$

Estimate: $\mathbb{E}_\pi[X] = \sum_{x \in X} x\pi(x) = \sum_{x \in X} x\pi(x) \frac{b(x)}{b(x)} = \sum_{x \in X} x\rho(x) b(x)$

Importance Sampling Ratio:

$$\rho(x) = \frac{\pi(x)}{b(x)}$$

F.M. Dekking et al. 'A Modern Introduction to Probability and Statistics'

- In off-policy we have:

Sample: $x \sim b$

Estimate: $\mathbb{E}_\pi[X] = \sum_{x \in X} x\pi(x) = \sum_{x \in X} x\pi(x) \frac{b(x)}{b(x)} = \sum_{x \in X} x\rho(x) b(x)$

Prediction/Control: Off-policy learning – Importance Sampling

DEFINITION. The *expectation* of a discrete random variable X taking the values $a_1, a_2, \ldots$ and with probability mass function p is the number

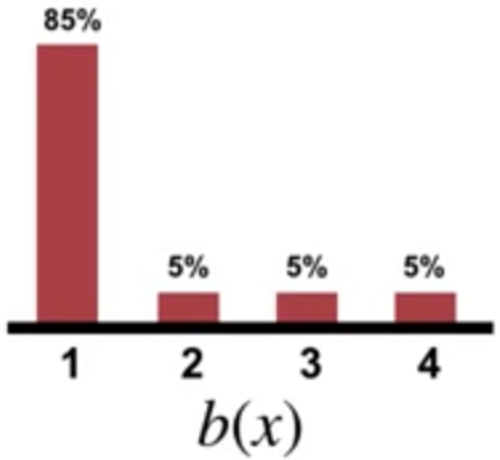
$$\mathrm{E}[X] = \sum_i a_i \mathrm{P}(X = a_i) = \sum_i a_i p(a_i).$$

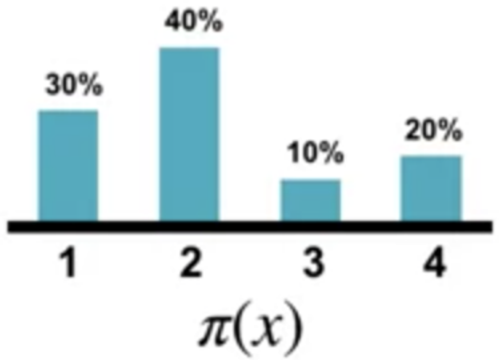
We can consider this as a new random variable

$$\mathbb{E}_\pi[X] = \sum_{x \in X} x\pi(x) = \sum_{x \in X} \boxed{x\rho(x)}\, b(x) = \mathbb{E}_b[X\rho(x)]$$

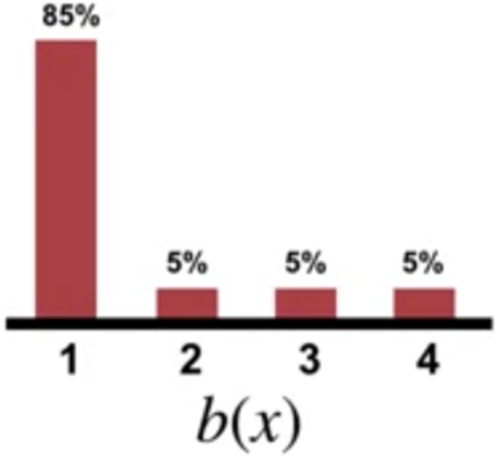
- We have a way to 'correct' the expectation if we draw samples from a different policy
- To actually use this with data, we exploit MC (a weighted sampled average)

$$\mathbb{E}_\pi[X] \sim \frac{1}{n} \sum_{i=1,\ldots,n} x_i \rho(x_i)$$

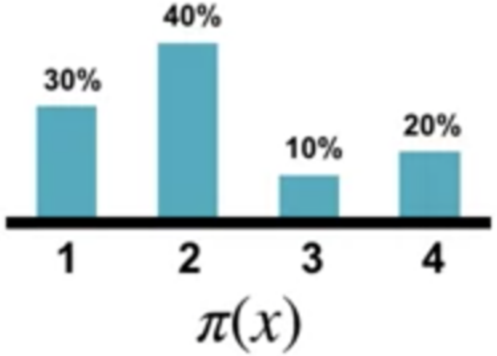




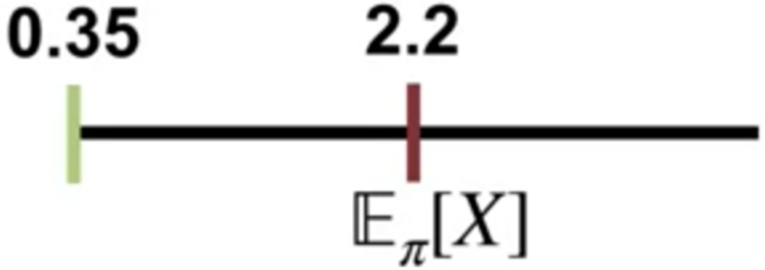
A. White, M. White 'Sample-based Learning Methods'



$$\longrightarrow \quad \begin{gathered} x = 1 \\ b(x) = .85 \end{gathered} \quad \longrightarrow \quad \mathbf{x = [\ 1\ ]}$$

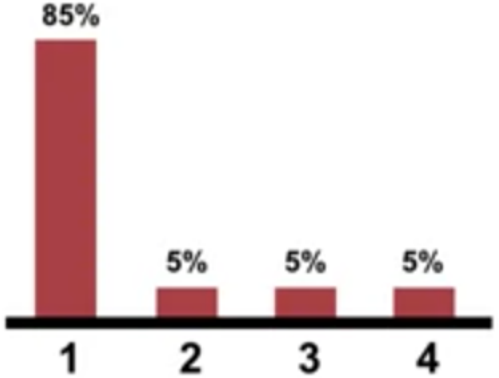


$$\longrightarrow \quad \pi(x) = .3 \quad \longrightarrow$$



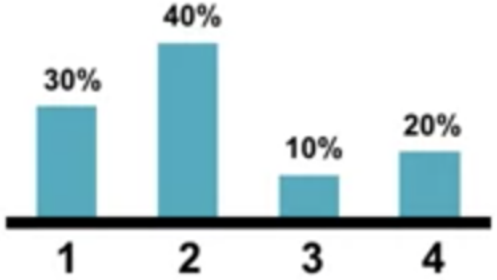
$$\frac{1}{n}\sum_{1}^{n} x\rho(x) \longrightarrow 1 \times \frac{.3}{.85} = 0.35$$

A. White, M. White 'Sample-based Learning Methods'



$b(x)$

$\longrightarrow$ $x = 3$, $b(x) = .05$ $\longrightarrow$ **x = [1, 3]**

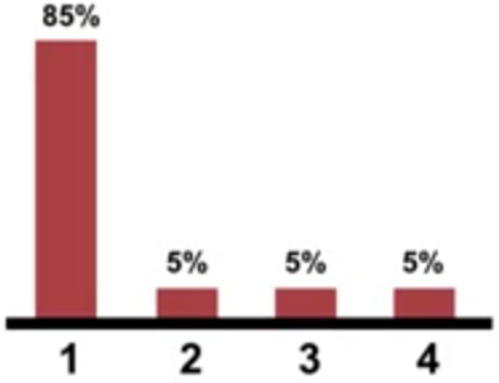


$\pi(x)$

$\longrightarrow$ $\pi(x) = .1$ $\longrightarrow$

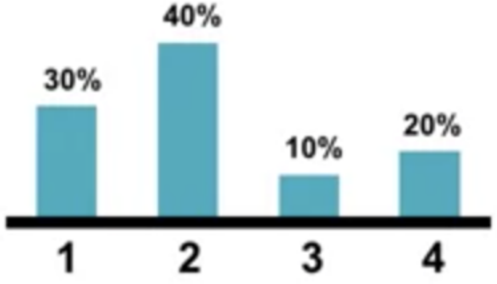
$$\frac{1}{n}\sum_{1}^{n} x\rho(x) \longrightarrow \frac{(1 \times \frac{.3}{.85}) + (3 \times \frac{.1}{.05})}{2} = 3.18$$

A. White, M. White 'Sample-based Learning Methods'



$b(x)$

$$\longrightarrow \quad \begin{gathered} x = 1 \\ b(x) = .85 \end{gathered} \quad \longrightarrow \quad \mathbf{x = [\ 1, 3, 1\ ]}$$



$\pi(x)$

$$\longrightarrow \quad \pi(x) = .3 \quad \longrightarrow$$

$$\frac{1}{n}\sum_{1}^{n} x\rho(x) \longrightarrow \frac{(1 \times \frac{.3}{.85}) + (3 \times \frac{.1}{.05}) + (1 \times \frac{.3}{.85})}{3} = 2.24$$

A. White, M. White 'Sample-based Learning Methods'

Prediction: Off-policy learning

- We use returns generated from b to evaluate π, we cannot directly compute v_π as the average of the returns seen!
- We have to correct each return thanks to importance sampling
- Given a starting state S_t the probability of the subsequent state-action trajectory under policy π is

$$\begin{aligned}
&\Pr\{A_t, S_{t+1}, A_{t+1}, \ldots, S_T \mid S_t, A_{t:T-1} \sim \pi\} \\
&\quad = \pi(A_t|S_t)p(S_{t+1}|S_t, A_t)\pi(A_{t+1}|S_{t+1}) \cdots p(S_T|S_{T-1}, A_{T-1}) \\
&\quad = \prod_{k=t}^{T-1} \pi(A_k|S_k)p(S_{k+1}|S_k, A_k),
\end{aligned}$$

Prediction: Off-policy learning

- The relative probability of the trajectory under the target and behaviour policy (the importance-sampling ratio) is:

$$\rho_{t:T-1} \doteq \frac{\prod_{k=t}^{T-1} \pi(A_k|S_k)p(S_{k+1}|S_k, A_k)}{\prod_{k=t}^{T-1} b(A_k|S_k)p(S_{k+1}|S_k, A_k)} = \prod_{k=t}^{T-1} \frac{\pi(A_k|S_k)}{b(A_k|S_k)}$$

- We can now exploit returns obtain with b to estimate v_π with MC

$$\mathbb{E}[\rho_{t:T-1} G_t \mid S_t = s] = v_\pi(s)$$

- We can consider importance-sampling ratios as 'weights'

Off-policy MC prediction (policy evaluation) for estimating $Q \approx q_\pi$

Differences w.r.t. on-policy

Input: an arbitrary target policy π

Initialize, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$:

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

$\quad C(s, a) \leftarrow 0$

Loop forever (for each episode):

$\quad \boxed{b \leftarrow \text{any policy with coverage of } \pi}$

$\quad$ Generate an episode following $\boxed{b\text{: } S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T}$

$\quad G \leftarrow 0$

$\quad \boxed{W \leftarrow 1}$

$\quad$ Loop for each step of episode, $t = T-1, T-2, \ldots, 0$, $\boxed{\text{while } W \neq 0\text{:}}$

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

$\quad\quad \boxed{C(S_t, A_t) \leftarrow C(S_t, A_t) + W}$

$\quad\quad Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \boxed{\frac{W}{C(S_t, A_t)}} [G - Q(S_t, A_t)]$

$\quad\quad \boxed{W \leftarrow W \frac{\pi(A_t|S_t)}{b(A_t|S_t)}}$

Off-policy MC prediction (policy evaluation) for estimating $Q \approx q_\pi$

Input: an arbitrary target policy π

Initialize, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$:

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

$C(s, a) \leftarrow 0$

Loop forever (for each episode):

$b \leftarrow$ any policy with coverage of π

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

$G \leftarrow 0$

$W \leftarrow 1$

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

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

$C(S_t, A_t) \leftarrow C(S_t, A_t) + W$

$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \frac{W}{C(S_t, A_t)}[G - Q(S_t, A_t)]$

$W \leftarrow W \frac{\pi(A_t|S_t)}{b(A_t|S_t)}$

Differences w.r.t. on-policy

Initialization

We are considering $C(s, a)$, a matrix that will contain the cumulative sum of the 'weights'

Off-policy MC prediction (policy evaluation) for estimating $Q \approx q_\pi$

Input: an arbitrary target policy π

Initialize, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$:

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

$C(s,a) \leftarrow 0$

Loop forever (for each episode):

$b \leftarrow$ any policy with coverage of π

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

$G \leftarrow 0$

$W \leftarrow 1$

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

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

$C(S_t, A_t) \leftarrow C(S_t, A_t) + W$

$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \frac{W}{C(S_t, A_t)}[G - Q(S_t, A_t)]$

$W \leftarrow W \frac{\pi(A_t|S_t)}{b(A_t|S_t)}$

Differences w.r.t. on-policy

We are using a different policy for data generation

Off-policy MC prediction (policy evaluation) for estimating $Q \approx q_\pi$

Differences w.r.t. on-policy

Input: an arbitrary target policy π
Initialize, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$:
  $Q(s,a) \in \mathbb{R}$ (arbitrarily)
  $C(s,a) \leftarrow 0$

Loop forever (for each episode):
  $b \leftarrow$ any policy with coverage of π
  Generate an episode following b: $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$
  $G \leftarrow 0$
  $W \leftarrow 1$
  Loop for each step of episode, $t = T-1, T-2, \ldots, 0$, while $W \neq 0$:
    $G \leftarrow \gamma G + R_{t+1}$
    $C(S_t, A_t) \leftarrow C(S_t, A_t) + W$
    $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \frac{W}{C(S_t, A_t)}[G - Q(S_t, A_t)]$
    $W \leftarrow W \frac{\pi(A_t|S_t)}{b(A_t|S_t)}$

We are correcting for the weights given by the Importance Sampling

Off-policy MC prediction (policy evaluation)

Input: an arbitrary target policy π

Initialize, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$:

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

$C(s, a) \leftarrow 0$

Loop forever (for each episode):

$b \leftarrow$ any policy with coverage of π

Generate an episode following b: $S_0, A_0, R_1, \ldots$

$G \leftarrow 0$

$W \leftarrow 1$

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

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

$C(S_t, A_t) \leftarrow C(S_t, A_t) + W$

$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \frac{W}{C(S_t, A_t)}[G - Q(S_t, A_t)]$

$W \leftarrow W \frac{\pi(A_t|S_t)}{b(A_t|S_t)}$

We are computing the 'weights' incrementally

$$\rho_{t:T-1} \doteq \prod_{k=t}^{T-1} \frac{\pi(A_k \mid S_k)}{b(A_k \mid S_k)}$$

$$= \rho_t \rho_{t+1} \rho_{t+2} \cdots \rho_{T-2} \rho_{T-1}$$

$$\longleftarrow$$

$$W_1 \leftarrow \rho_{T-1}$$

$$W_2 \leftarrow \rho_{T-1}\rho_{T-2}$$

$$W_3 \leftarrow \rho_{T-1}\rho_{T-2}\rho_{T-3}$$

We are correcting for the weights given by the Importance Sampling

Off-policy MC prediction (policy evaluation) for estimating $Q \approx q_\pi$

Differences w.r.t. on-policy

Input: an arbitrary target policy π
Initialize, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$:
    $Q(s,a) \in \mathbb{R}$ (arbitrarily)
    $C(s,a) \leftarrow 0$

Loop forever (for each episode):
    $\boxed{b \leftarrow \text{any policy with coverage of } \pi}$
    Generate an episode following $\boxed{b\text{: } S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T}$
    $G \leftarrow 0$
    $\boxed{W \leftarrow 1}$
    Loop for each step of episode, $t = T-1, T-2, \ldots, 0$, $\boxed{\text{while } W \neq 0\text{:}}$
        $G \leftarrow \gamma G + R_{t+1}$
        $\boxed{C(S_t, A_t) \leftarrow C(S_t, A_t) + W}$
        $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \boxed{\frac{W}{C(S_t, A_t)}}[G - Q(S_t, A_t)]$
        $\boxed{W \leftarrow W \frac{\pi(A_t|S_t)}{b(A_t|S_t)}}$

Pay attention: the book decided to report the every-visit case!

Monte Carlo methods: Exam

- All the content of Chapter 5 are Exam material, beside:

i. it can be considered optional the policy improvement theorem for ε-soft policies (in section 5.4)

ii. Sections 5.7, 5.8 and 5.9 can be skipped

- In particular, Section 5.7 talks about off-policy MC control: in practice it is useless! Over many steps, off-policy estimations are really high-variance and slow to converge! However, with Temporal Difference learning, we'll make off-policy work also on practice!
- Pay particular attention to the algorithms!

Credits

- Image of the course is taken from C. Mahoney ‘Reinforcement Learning’ https://towardsdatascience.com/reinforcement-learning-fda8ff535bb6
- Many concepts and material for examples have been taken from A. White, M. White ‘Sample-based Learning Methods’

Thank you!
Questions?

Gian Antonio Susto
Ruggero Carli