ELTE | FACULTY OF INFORMATICS

MONTE CARLO METHODS

Deep Reinforcement Learning

Balázs Nagy, PhD

ELTE | IK

DEPARTMENT OF ARTIFICIAL INTELLIGENCE

Monte Carlo Methods

- Estimated value function
- No complete knowledge of the environment
- Experience – sample sequences of (s,a,r) from actual or simulated interactions
- Model is required, but not the complete probability distribution

Monte Carlo Methods

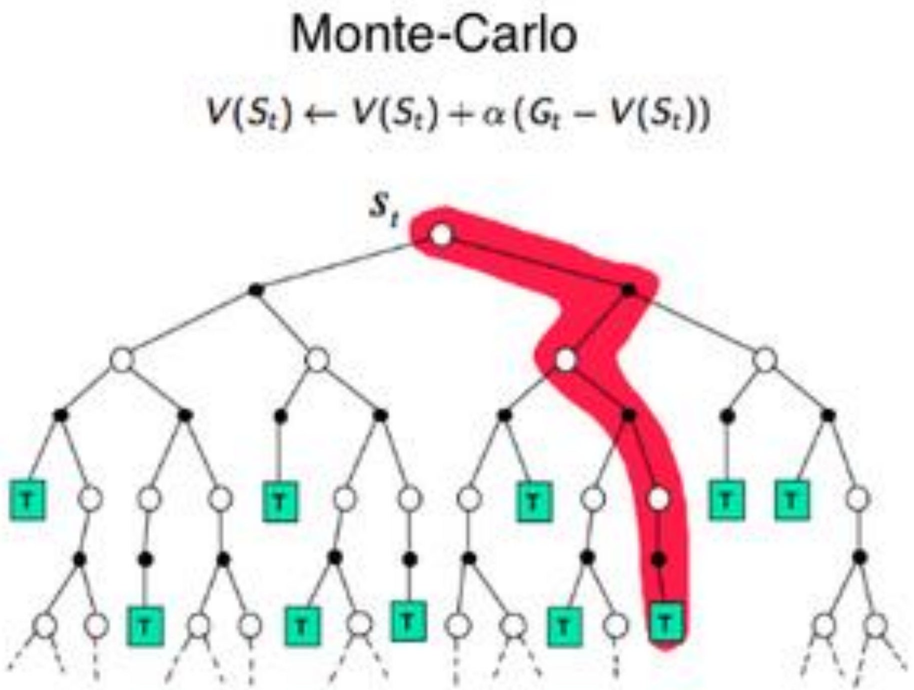
- Estimated value function
- No complete knowledge of the environment
- Experience – sample sequences of (s,a,r) from actual or simulated interactions
- Model is required, but not the complete probability distribution

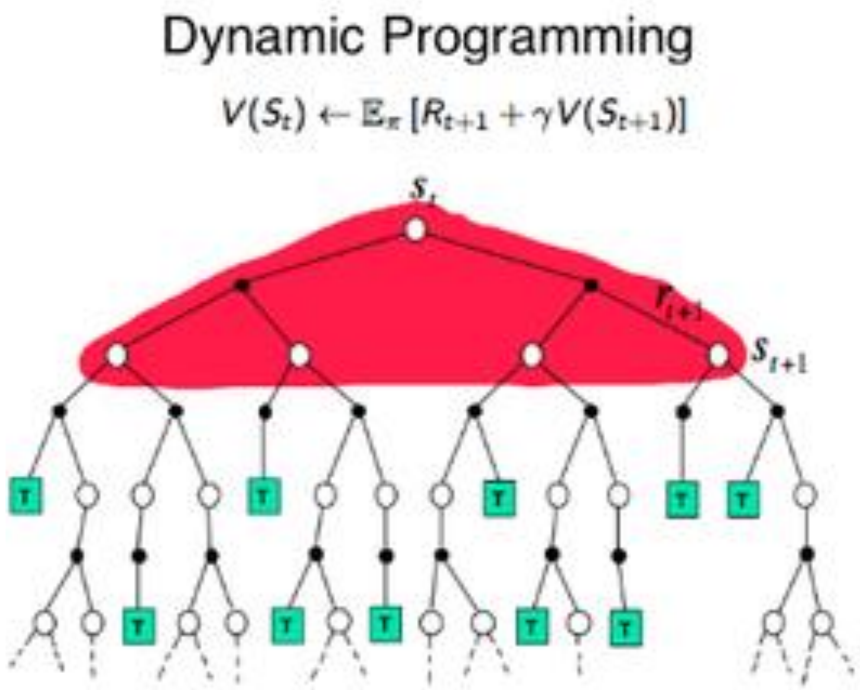
- Solving RL problem based on averaging sample returns
- For now: only episodic task
- Update only at the end of the episode

Monte Carlo Prediction

- MC for learning the state-value function for a given policy
- Value of a state = expected cumulative future discounted reward
- Estimated value of a state = average the returns observed after visits to that state (from experience)
  - First-visit MC – in each episode only the first visit to *s* counts
  - Every-visit MC – in each episode all of the visits to *s* count

MC vs DP





No bootstrap
does not build upon the
estimate of any other state

Pseudocode

First-visit MC prediction, for estimating $V \approx v_\pi$

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: $S_0, A_0, R_1, S_1, A_1, R_2, \dots, S_{T-1}, A_{T-1}, R_T$

  $G \leftarrow 0$

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

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

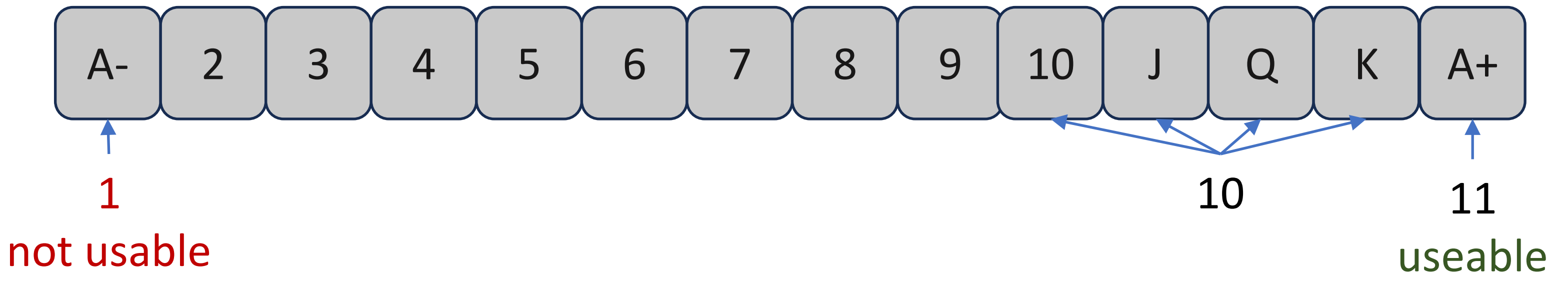
    Unless S_t appears in $S_0, S_1, \dots, S_{t-1}$:

      Append G to $Returns(S_t)$

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

Blackjack example

The cards:



The goal: obtain cards the sum of whose numerical values is as great as possible without exceeding 21 (Player's actions first, Dealer's actions after)

Starting state:

Player: Q 8

Dealer: 10 ?

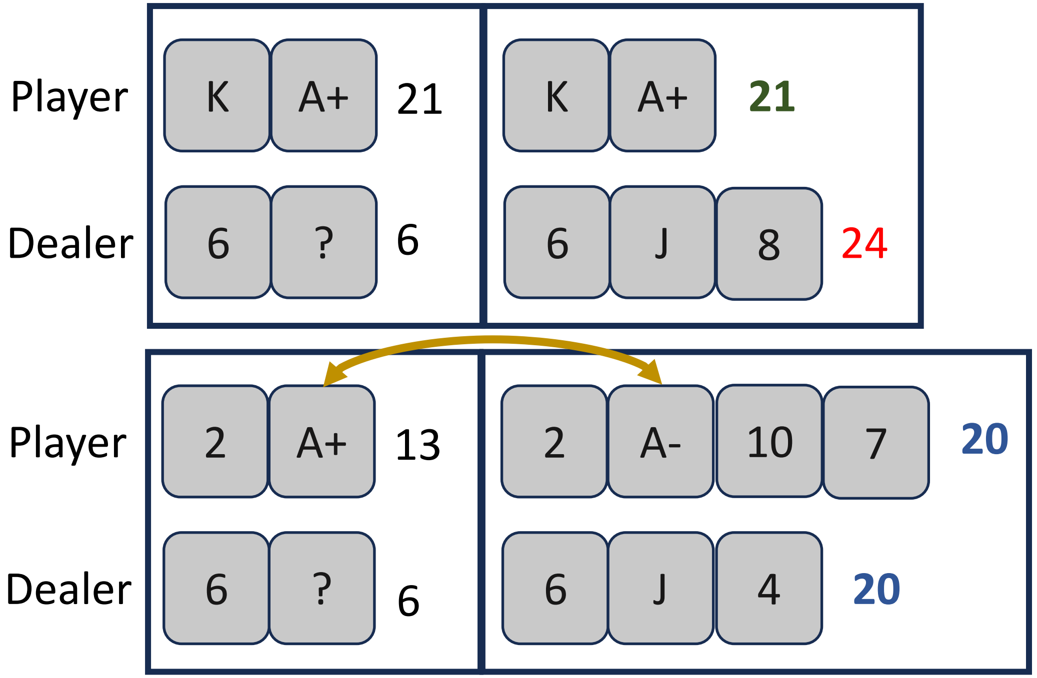
Actions:
- stick
- hit (get another card)

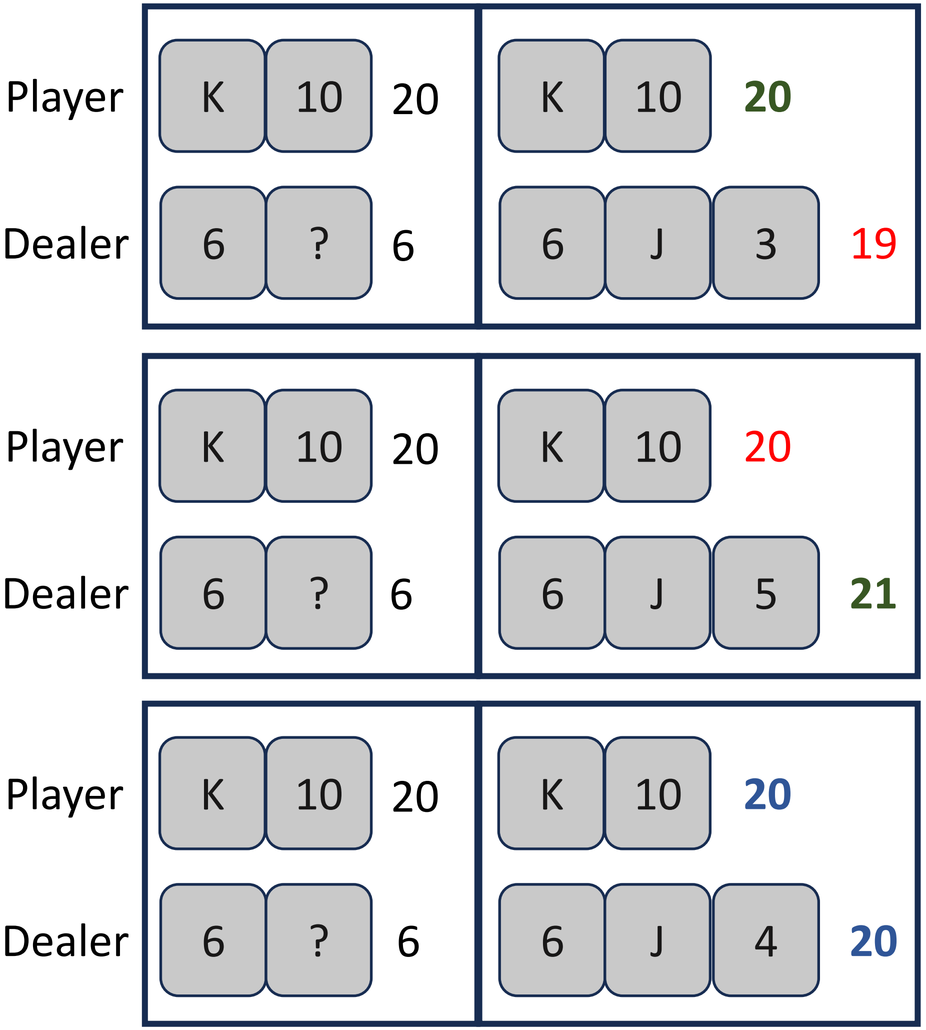
Dealer's strategy:
- stick is sum is 17 or greater
- hit otherwise

Blackjack as a natural MDP

- Episodic finite MDP
- Each game of Blackjack is an episode
- **Rewards**: +1 (win), -1 (lose), 0 (draw)
  all immediate rewards within a game is 0
  there is no discount ($\gamma = 1$)
- The **agent** is the player
- The player **actions** are hit or stick
- The **states** depend on the player's card (sum 12-21) and the dealer's showing card (ace-10) – approximately 200 states
- Infinite deck is assumed with replacement (no card counting)

Let's play some episodes



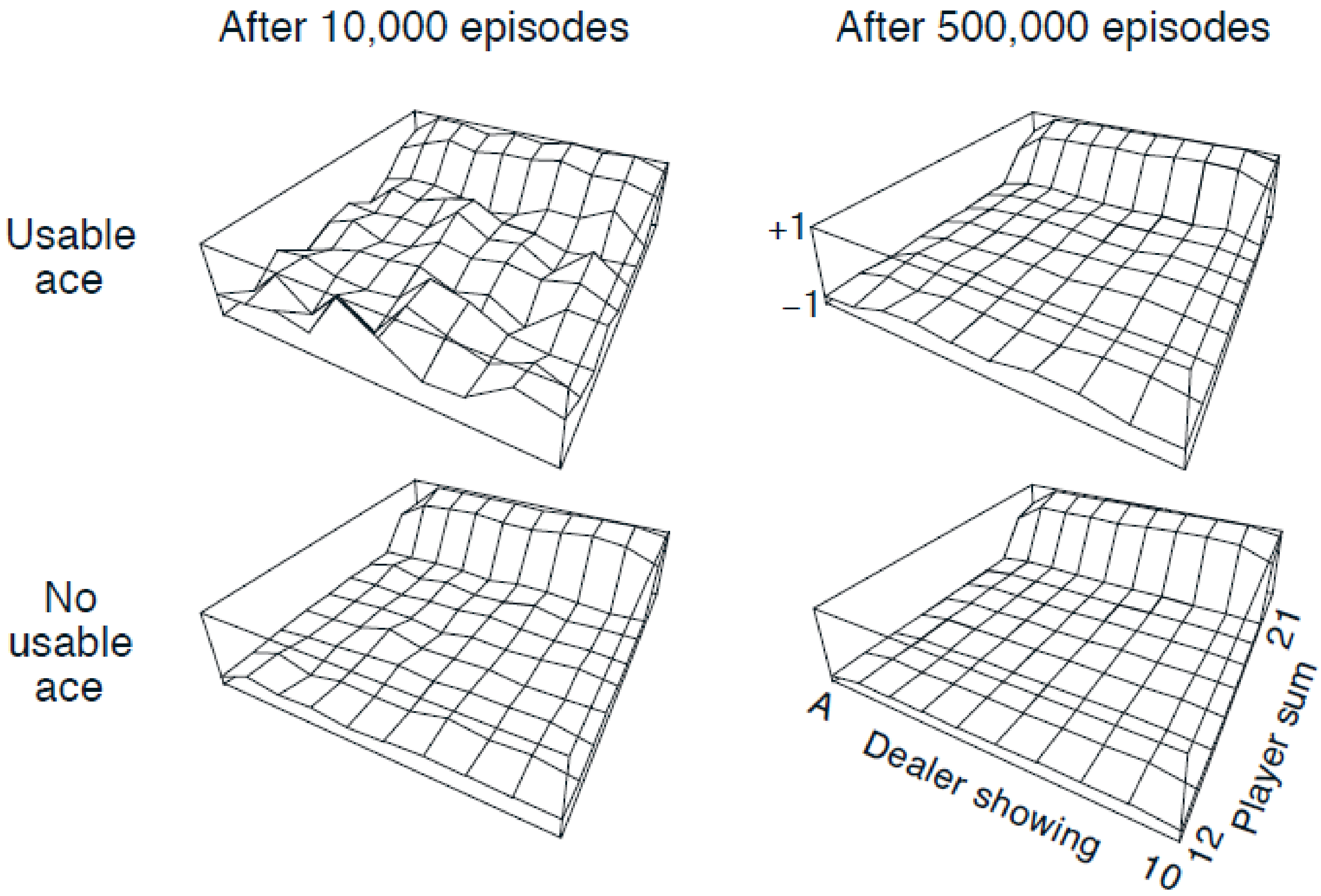


1, Starting state

2, Player's actions (stick if 20 or greater)

3, Dealer's actions (stick if 17 or greater)

Approximate state value functions



State

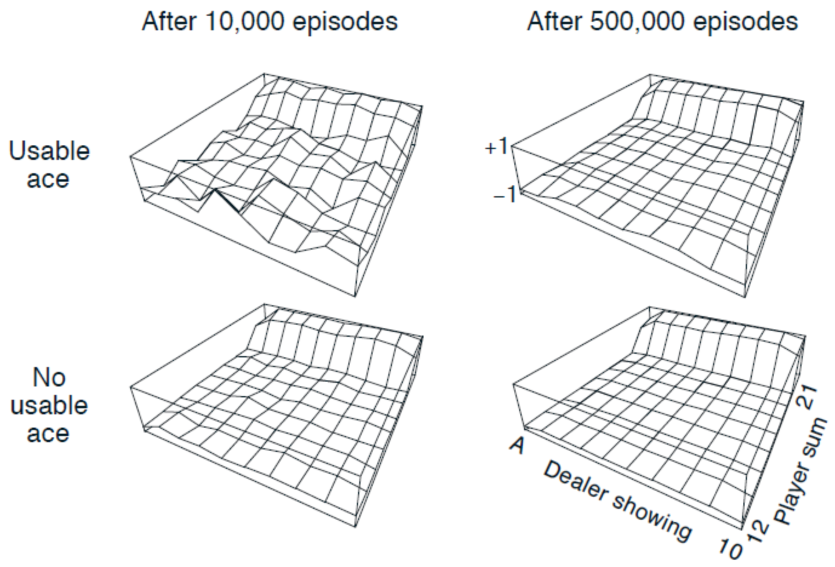
Player's sum: 18

Useable ace: Yes / No

Dealer's sum: 9

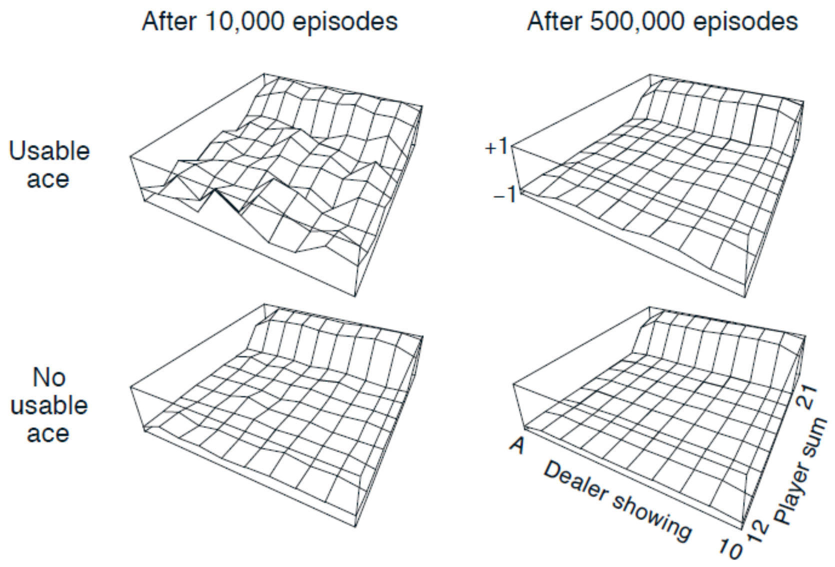
Blackjack questions

- Why does the estimated value function jump up for the last two?
- Why are the frontmost values higher in the upper diagrams than in the lower?



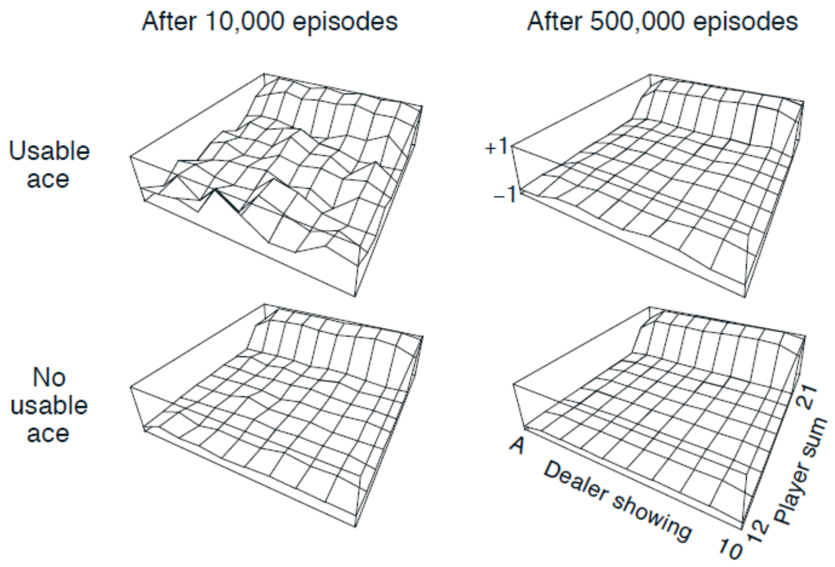
Blackjack questions

- Why does the estimated value function jump up for the last two?
  - The player sticks at 20 and 21, where it is unlikely the dealer beat him given the dealer's policy to hit for all hands lower than 17
- Why are the frontmost values higher in the upper diagrams than in the lower?



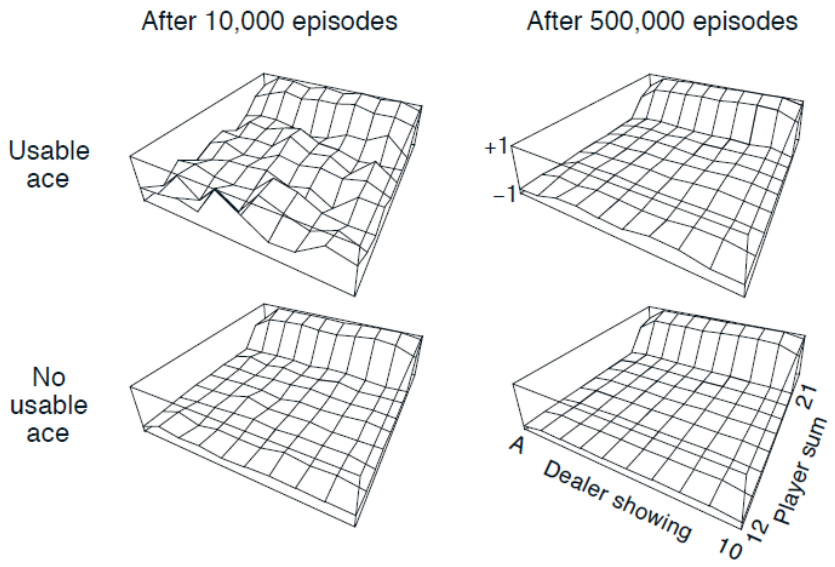
Blackjack questions

- Why does the estimated value function jump up for the last two?
  - The player sticks at 20 and 21, where it is unlikely the dealer beat him given the dealer's policy to hit for all hands lower than 17
- Why are the frontmost values higher in the upper diagrams than in the lower?
  - When an ace is usable, the value function is higher because the player can change the value of his ace from 11 to 1 if she is about to go bust



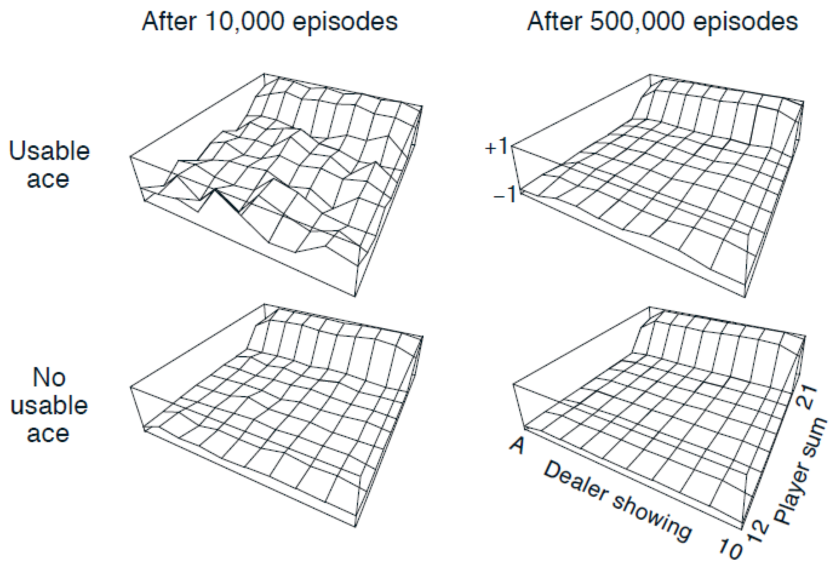
Blackjack questions

- Suppose every-visit MC was used instead of first-visit MC on the blackjack task. Would you expect the results to be very different? Why or why not?



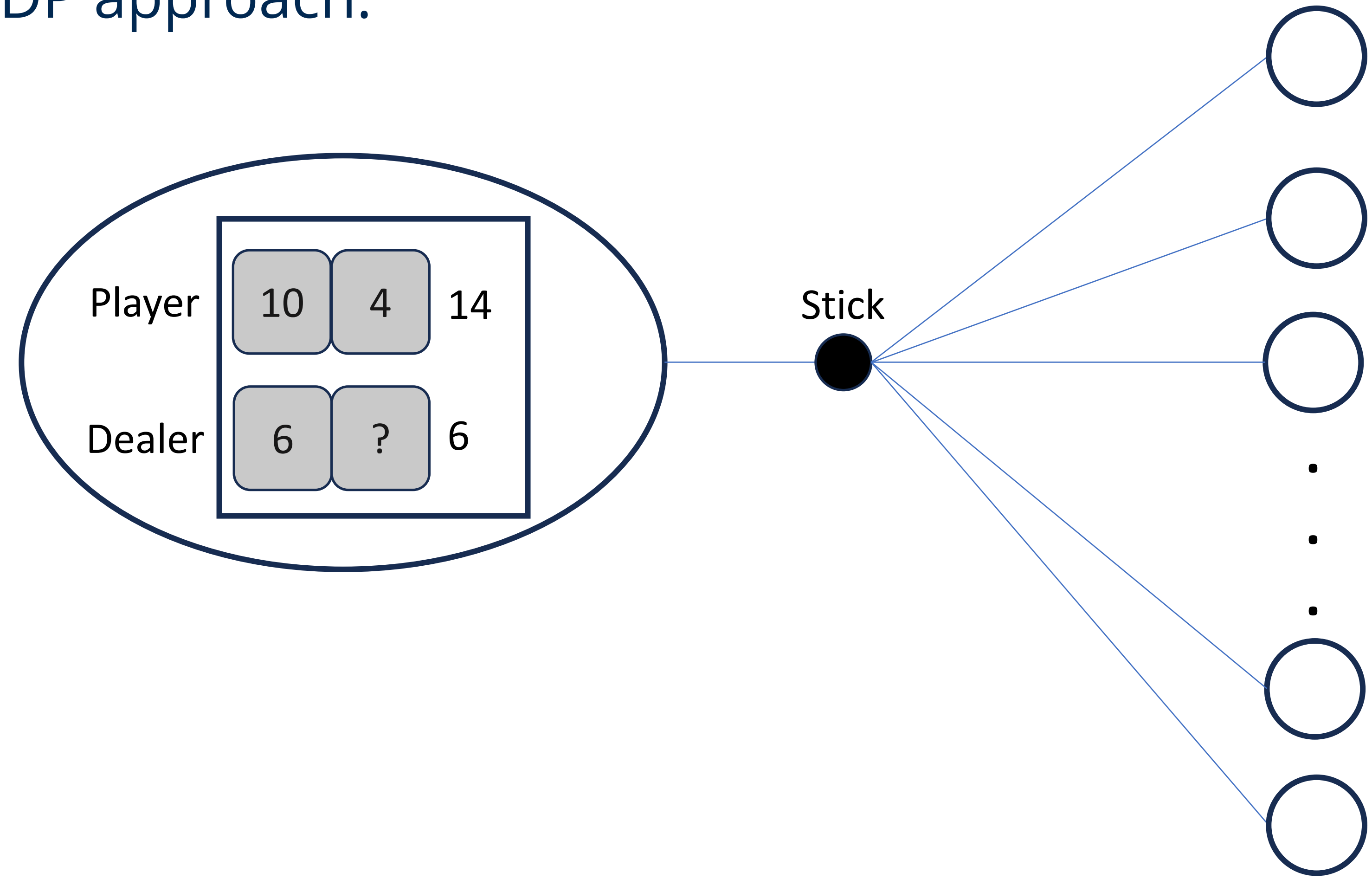
Blackjack questions

- Suppose every-visit MC was used instead of first-visit MC on the blackjack task. Would you expect the results to be very different? Why or why not?
  - The state is blackjack is monotonically increasing and memoryless (sampled with replacement), thus you can never revisit an old state in an episode once it has been first visited. Using every visit MC in this case would have no effect on the value function



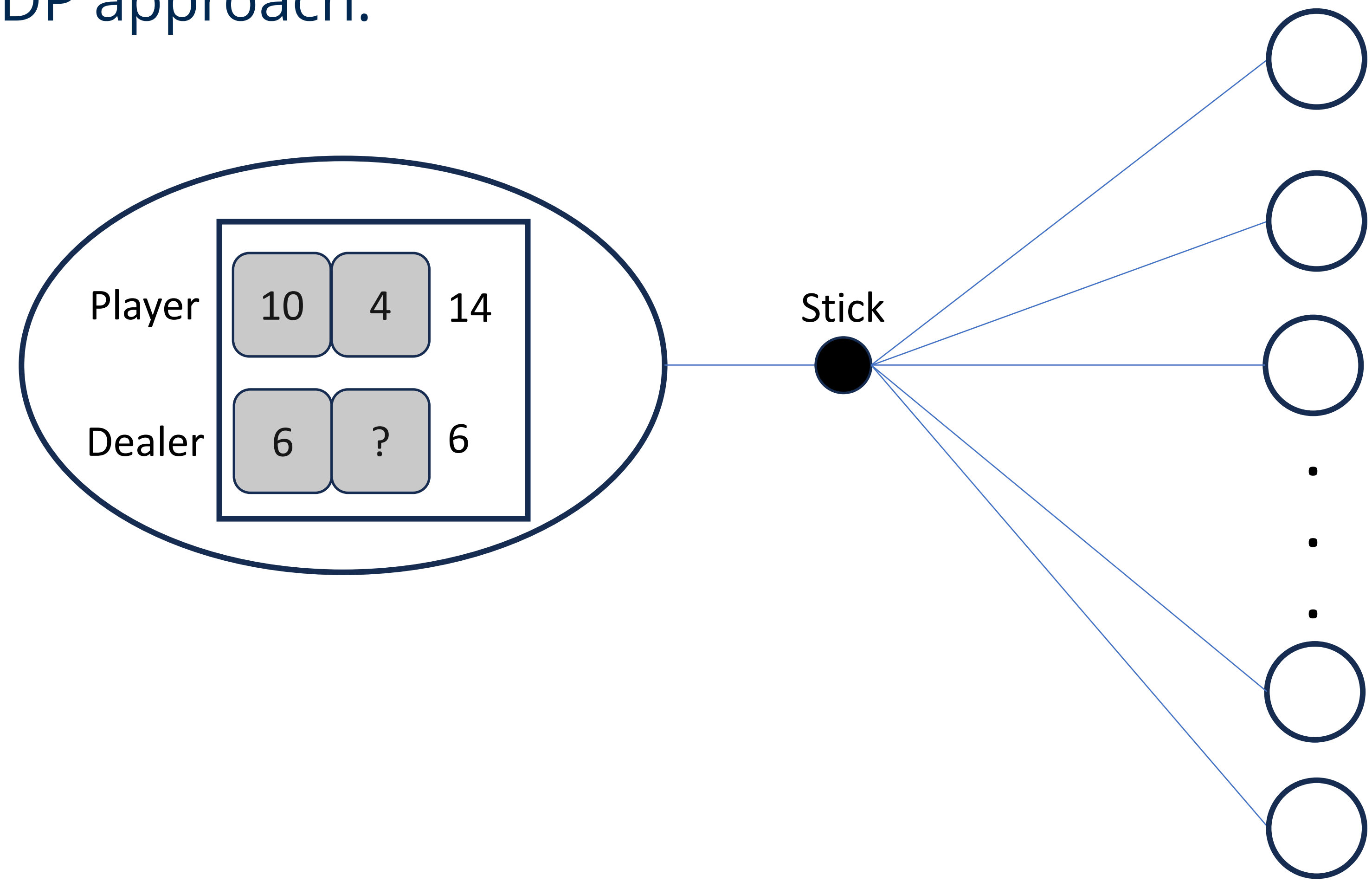
Using MC or DP

DP approach:



Using MC or DP

DP approach:



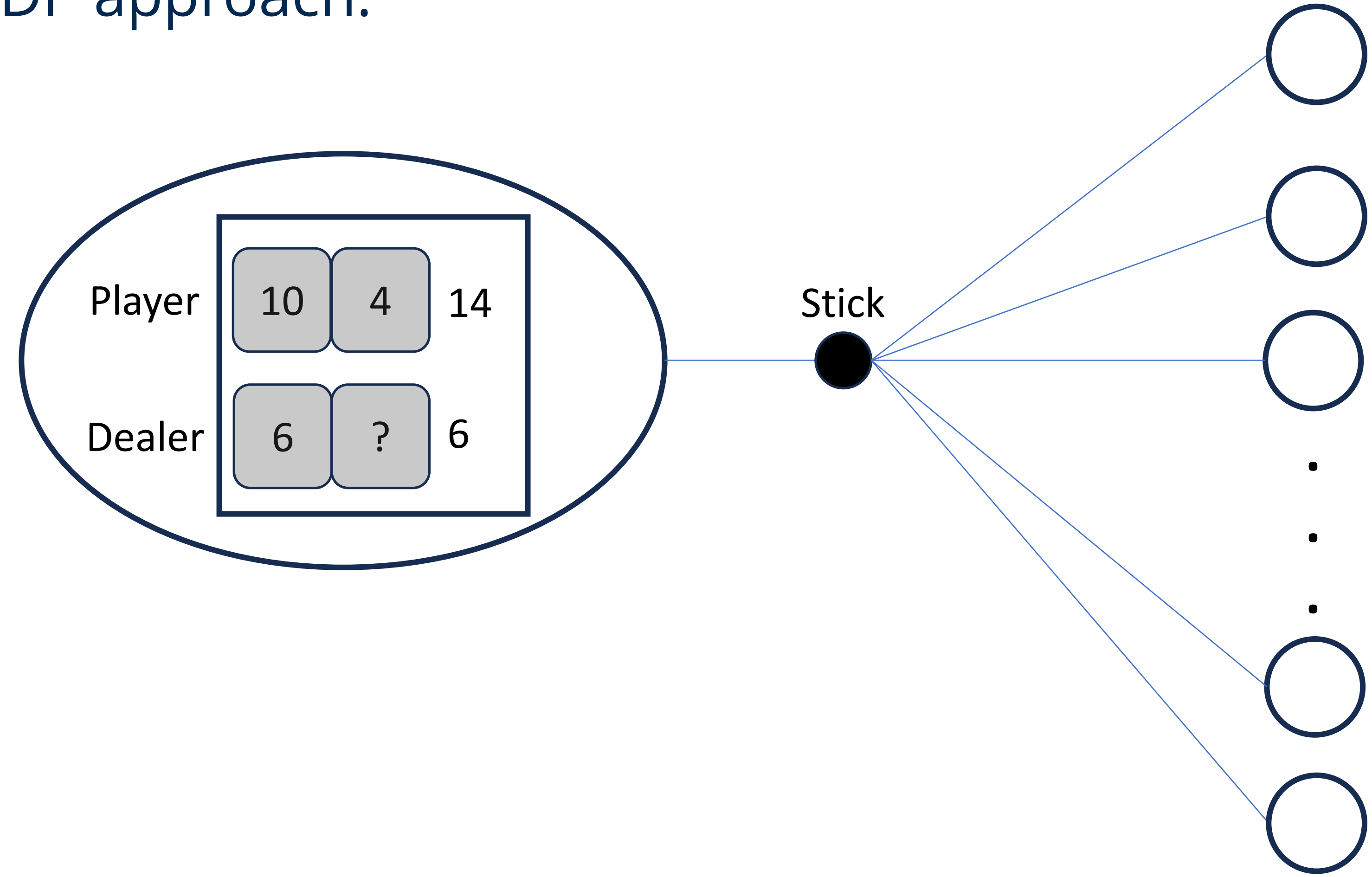
What is the probability of winning?

How many possible states there are?

Distribution of next events?

Using MC or DP

DP approach:



What is the probability of winning?

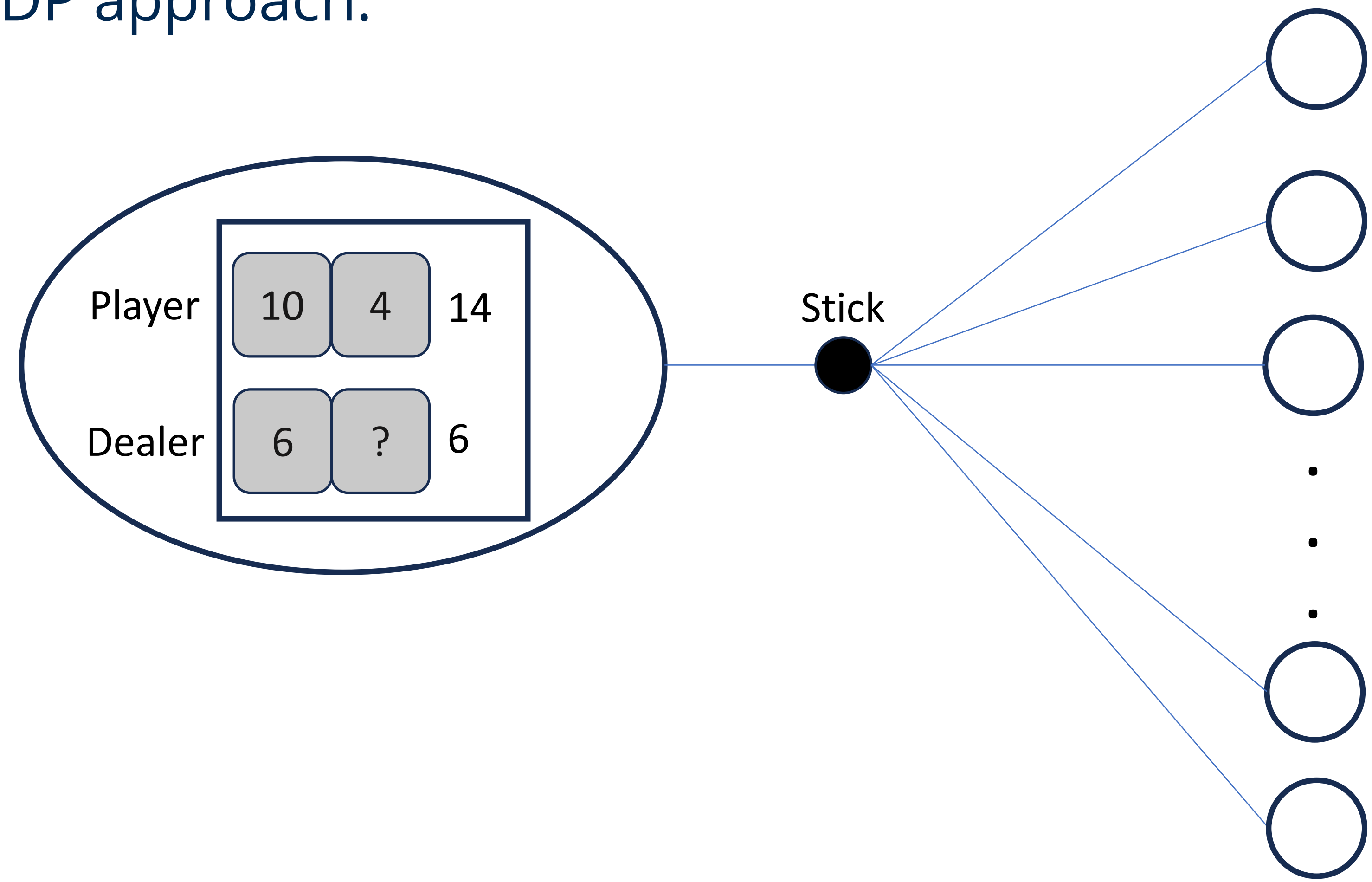
How many possible states there are?

Distribution of next events?

Lot of computation
Problematic calculation

Using MC or DP

DP approach:



What is the probability of winning?

How many possible states there are?

Distribution of next events?

Lot of computation
Problematic calculation

Using approximation-based MC, *even when the dynamics of the environment is known*

MC estimation of action values

- If model is known
  - State values can determine a policy
- If model is not known
  - State values only not sufficient to form a policy
  - Estimate of action values needed

MC estimation of action values

- If model is known
  - State values can determine a policy
- If model is not known
  - State values only not sufficient to form a policy
  - Estimate of action values needed

Problem: How to estimate $q_\pi(s,a)$?

MC estimation of action values

- If model is known
  - State values can determine a policy
- If model is not known
  - State values only not sufficient to form a policy
  - Estimate of action values needed

Problem: How to estimate $q_\pi(s,a)$?

Solution: Use MC to visit state-action pairs instead of states

MC estimation of action values

- Complication
  - Many state-action pair may never be visited
    (in case of deterministic policy)
  - General problem: How to maintain exploration?

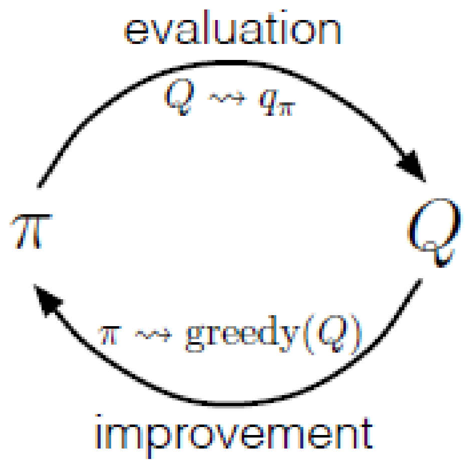
MC estimation of action values

- Complication
  - Many state-action pair may never be visited
    (in case of deterministic policy)
  - General problem: How to maintain exploration?

- Solution
  - Exploring starts: the episode start in a random state-action pair
    (every pair has a nonzero probability)
  - Useing stochastic policies
    (with nonzero probability of selecting all actions in each state)

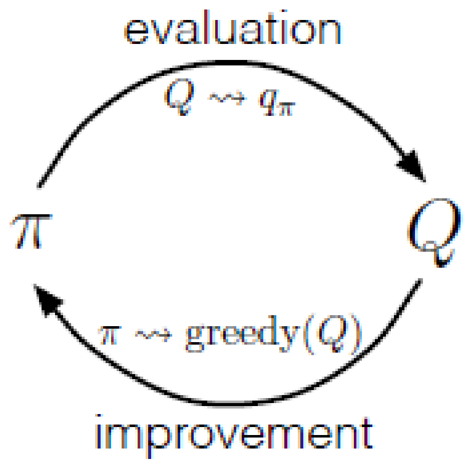
MC control

- Approximate optimal policies
- Idea of the GPI
- Policy and Value function creates a moving target for each other
- Together they approach optimality



MC control

- Approximate optimal policies
- Idea of the GPI
- Policy and Value function creates a moving target for each other
- Together they approach optimality



$$\pi_0 \xrightarrow{\mathrm{E}} q_{\pi_0} \xrightarrow{\mathrm{I}} \pi_1 \xrightarrow{\mathrm{E}} q_{\pi_1} \xrightarrow{\mathrm{I}} \pi_2 \xrightarrow{\mathrm{E}} \cdots \xrightarrow{\mathrm{I}} \pi_* \xrightarrow{\mathrm{E}} q_*$$

E: policy evaluation, many episodes experienced (theoretically infinite) with exploring starts

I: policy improvement, make the policy greedy to the current value function

$$\pi(s) \doteq \arg\max_a q(s,a)$$

Policy improvement theorem

$$\pi_0 \xrightarrow{\mathrm{E}} q_{\pi_0} \xrightarrow{\mathrm{I}} \pi_1 \xrightarrow{\mathrm{E}} q_{\pi_1} \xrightarrow{\mathrm{I}} \pi_2 \xrightarrow{\mathrm{E}} \cdots \xrightarrow{\mathrm{I}} \pi_* \xrightarrow{\mathrm{E}} q_*$$

π_{k+1} is better than π_k (or as good as)

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

MC control summary

- The overall process converges to the optimal policy and optimal value function
- MC methods can be used to find optimal policies given only sample episodes and no other knowledge of the environment's dynamics

MC control summary

- The overall process converges to the optimal policy and optimal value function
- MC methods can be used to find optimal policies given only sample episodes and no other knowledge of the environment's dynamics

Unlikely assumptions: (can not be in a practical algorithm)

- Exploring starts
  (random (s,a) could be dangerous when learning from actual interaction
- Infinite number of episodes during the policy evaluation

Number of Episodes in Policy Evaluation

- Infinite number of episodes
  - Converge to the true value
  - Impossible to obtain
- Finite number of episodes
  - Converge asymptotically to the true value
  - Still require many episodes in a complete policy evaluation step
- Incomplete policy evaluation
  - No complete policy evaluation before returning to policy improvement
  - Extreme case: only 1 iteration

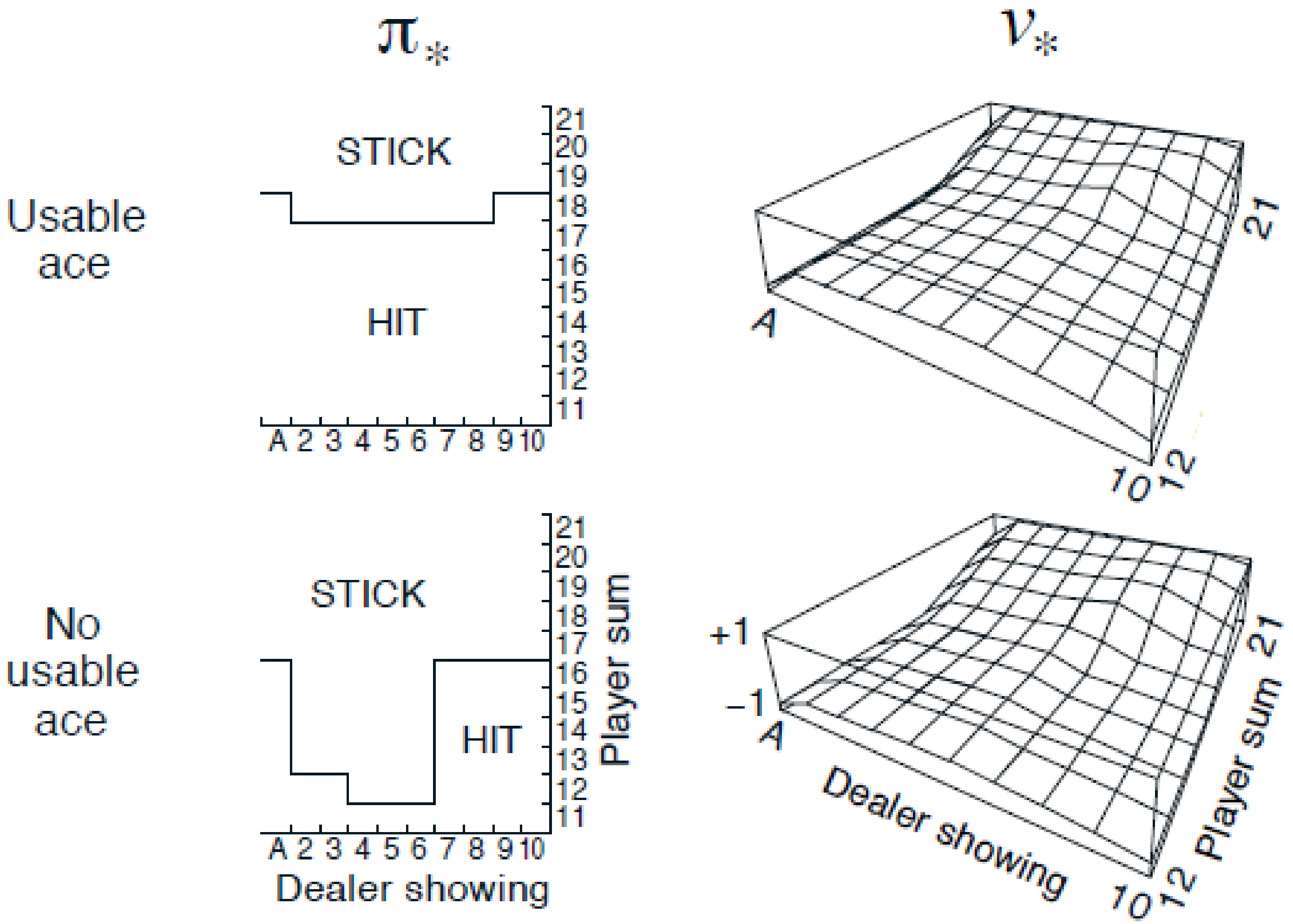
Pseudocode

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):
  Choose $S_0 \in \mathcal{S}$ and $A_0 \in \mathcal{A}(S_0)$ 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, $t = T-1, T-2, \ldots, 0$:
    $G \leftarrow G + R_{t+1}$
    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 $Returns(S_t, A_t)$
      $Q(S_t, A_t) \leftarrow \text{average}(Returns(S_t, A_t))$
      $\pi(S_t) \leftarrow \arg\max_a Q(S_t, a)$

Blackjack with MC ES



MC discussion

- MC policy evaluation and improvement on an episode-by-episode basis
- MC ES: all the returns for each state-action pair are accumulated and averaged, irrespective of what policy was in force when they were observed
- Stability is achieved only when both the policy and the value function are optimal
- Convergence to this optimal fixed point seems inevitable as the changes to the action-value function decrease over time, but has **not yet been formally proved**

MC discussion

Theory

You know everything, but nothing works

Practice

Everything works, but no one knows why

Definitions

- **On-policy methods**:
  evaluate or improve the policy that is used to make decisions
  (example: MC with ES)
  - Generally **soft policy**:
    has some, usually small but finite, probability of selecting any possible action $\pi(a|s)>0$
    (example: ε-greedy)

- **Off-policy methods**:
  evaluate or improve a policy different from that used to generate the data

ε-soft and ε-greedy

- An **ε−soft** policy is any policy where the probability of all actions given a state *s* is greater than some minimum value.

$$\pi(a|s) \geq \frac{\varepsilon}{|\mathcal{A}(s)|}, \forall a \in \mathcal{A}(s)$$

- An **ε−greedy** policy is a specific instance of ε−soft policy. Defined with respect to the action-value *Q*(*s*,*a*)

$$a^* = \arg\max_a Q(s,a)$$

$$\pi(a|s) = 1 - \varepsilon + \frac{\varepsilon}{|\mathcal{A}(s)|}, \quad a = a^*$$
$$\pi(a|s) = \frac{\varepsilon}{|\mathcal{A}(s)|}, \quad a \neq a^*$$

Pseudocode

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: $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$
    $G \leftarrow 0$
    Loop for each step of episode, $t = T-1, T-2, \ldots, 0$:
        $G \leftarrow G + R_{t+1}$
        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 $a \in \mathcal{A}(S_t)$:

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

No exploring starts (ES)

Policy improvement theorem

$$
\begin{aligned}
q_\pi(s, \pi'(s)) &= \sum_a \pi'(a|s) q_\pi(s,a) \\
&= \frac{\varepsilon}{|\mathcal{A}(s)|} \sum_a q_\pi(s,a) \;+\; (1-\varepsilon) \max_a q_\pi(s,a) \\
&\geq \frac{\varepsilon}{|\mathcal{A}(s)|} \sum_a q_\pi(s,a) \;+\; (1-\varepsilon) \sum_a \frac{\pi(a|s) - \frac{\varepsilon}{|\mathcal{A}(s)|}}{1-\varepsilon} q_\pi(s,a) \\
&= \frac{\varepsilon}{|\mathcal{A}(s)|} \sum_a q_\pi(s,a) \;-\; \frac{\varepsilon}{|\mathcal{A}(s)|} \sum_a q_\pi(s,a) \;+\; \sum_a \pi(a|s) q_\pi(s,a) \\
&= v_\pi(s)
\end{aligned}
$$

Policy improvement theorem

$$
\begin{aligned}
q_\pi(s, \boxed{\pi'(s)}) &= \sum_a \pi'(a|s) q_\pi(s,a) \\
&= \frac{\varepsilon}{|\mathcal{A}(s)|} \sum_a q_\pi(s,a) \; + \; (1-\varepsilon) \max_a q_\pi(s,a) \\
&\geq \frac{\varepsilon}{|\mathcal{A}(s)|} \sum_a q_\pi(s,a) \; + \; (1-\varepsilon) \sum_a \frac{\pi(a|s) - \frac{\varepsilon}{|\mathcal{A}(s)|}}{1-\varepsilon} q_\pi(s,a) \\
&= \frac{\varepsilon}{|\mathcal{A}(s)|} \sum_a q_\pi(s,a) \; - \; \frac{\varepsilon}{|\mathcal{A}(s)|} \sum_a q_\pi(s,a) \; + \; \sum_a \pi(a|s) q_\pi(s,a) \\
&= v_\pi(s)
\end{aligned}
$$

Action choice based on a greedy policy

Policy improvement theorem

$$
\begin{aligned}
q_\pi(s, \boxed{\pi'(s)}) &= \sum_a \pi'(a|s) q_\pi(s,a) \\
&= \boxed{\frac{\varepsilon}{|\mathcal{A}(s)|} \sum_a q_\pi(s,a)} + (1-\varepsilon) \max_a q_\pi(s,a) \\
&\geq \frac{\varepsilon}{|\mathcal{A}(s)|} \sum_a q_\pi(s,a) + (1-\varepsilon) \sum_a \frac{\pi(a|s) - \frac{\varepsilon}{|\mathcal{A}(s)|}}{1-\varepsilon} q_\pi(s,a) \\
&= \frac{\varepsilon}{|\mathcal{A}(s)|} \sum_a q_\pi(s,a) - \frac{\varepsilon}{|\mathcal{A}(s)|} \sum_a q_\pi(s,a) + \sum_a \pi(a|s) q_\pi(s,a) \\
&= v_\pi(s)
\end{aligned}
$$

Action choice based on a greedy policy

Soft policy (every action has a small probability)

Policy improvement theorem

$$
\begin{aligned}
q_\pi(s, \boxed{\pi'(s)}) &= \sum_a \pi'(a|s) q_\pi(s,a) \\
&= \boxed{\frac{\varepsilon}{|\mathcal{A}(s)|} \sum_a q_\pi(s,a)} + \boxed{(1-\varepsilon) \max_a q_\pi(s,a)} \\
&\geq \frac{\varepsilon}{|\mathcal{A}(s)|} \sum_a q_\pi(s,a) + (1-\varepsilon) \sum_a \frac{\pi(a|s) - \frac{\varepsilon}{|\mathcal{A}(s)|}}{1-\varepsilon} q_\pi(s,a) \\
&= \frac{\varepsilon}{|\mathcal{A}(s)|} \sum_a q_\pi(s,a) - \frac{\varepsilon}{|\mathcal{A}(s)|} \sum_a q_\pi(s,a) + \sum_a \pi(a|s) q_\pi(s,a) \\
&= v_\pi(s)
\end{aligned}
$$

Action choice based on a greedy policy

Soft policy (every action has a small probability)

Greedy action has a higher probability

Policy improvement theorem

$$
\begin{aligned}
q_\pi(s, \boxed{\pi'(s)}) &= \sum_a \pi'(a|s) q_\pi(s,a) \\
&= \boxed{\frac{\varepsilon}{|\mathcal{A}(s)|} \sum_a q_\pi(s,a)} + \boxed{(1-\varepsilon) \max_a q_\pi(s,a)} \\
&\boxed{\geq} \frac{\varepsilon}{|\mathcal{A}(s)|} \sum_a q_\pi(s,a) + (1-\varepsilon) \boxed{\sum_a \frac{\pi(a|s) - \frac{\varepsilon}{|\mathcal{A}(s)|}}{1-\varepsilon}} q_\pi(s,a) \\
&= \frac{\varepsilon}{|\mathcal{A}(s)|} \sum_a q_\pi(s,a) - \frac{\varepsilon}{|\mathcal{A}(s)|} \sum_a q_\pi(s,a) + \sum_a \pi(a|s) q_\pi(s,a) \\
&= v_\pi(s)
\end{aligned}
$$

Action choice based on a greedy policy

Soft policy (every action has a small probability)

Greedy action has a higher probability

the sum is a weighted average with nonnegative weights summing to 1, and as such it must be less than or equal to the largest number averaged

Policy improvement theorem

$$
\begin{aligned}
q_\pi(s, \boxed{\pi'(s)}) &= \boxed{\sum_a \pi'(a|s) q_\pi(s,a)} \\
&= \boxed{\frac{\varepsilon}{|\mathcal{A}(s)|} \sum_a q_\pi(s,a)} + \boxed{(1-\varepsilon) \max_a q_\pi(s,a)} \\
&\boxed{\geq} \frac{\varepsilon}{|\mathcal{A}(s)|} \sum_a q_\pi(s,a) + (1-\varepsilon) \boxed{\sum_a \frac{\pi(a|s) - \frac{\varepsilon}{|\mathcal{A}(s)|}}{1-\varepsilon}} q_\pi(s,a) \\
&= \cancel{\frac{\varepsilon}{|\mathcal{A}(s)|} \sum_a q_\pi(s,a)} - \cancel{\frac{\varepsilon}{|\mathcal{A}(s)|} \sum_a q_\pi(s,a)} + \boxed{\sum_a \pi(a|s) q_\pi(s,a)} \\
&= v_\pi(s)
\end{aligned}
$$

Action choice based on a greedy policy

Soft policy (every action has a small probability)

Greedy action has a higher probability

the sum is a weighted average with nonnegative weights summing to 1, and as such it must be less than or equal to the largest number averaged

One greedy step improved the performance

$\pi' \geq \pi$ (i.e., $v_{\pi'}(s) \geq v_\pi(s)$, for all $s \in \mathcal{S}$)

Policy prediction problem

Dilemma:

- Learn action values conditional on subsequent optimal behavior
- Behave non-optimally to explore all actions

Policy prediction problem

Dilemma:

- Learn action values conditional on subsequent optimal behavior
- Behave non-optimally to explore all actions

How to learn the optimal policy while behaving according to an exploratory policy?

Policy prediction problem

Dilemma:

- Learn action values conditional on subsequent optimal behavior
- Behave non-optimally to explore all actions

How to learn the optimal policy while behaving according to an exploratory policy?

- On-policy: using a near optimal policy
- Off-policy: using 2 policies (target and behavior)

Off-policy prediction

- **Target policy**:
  The policy is being learned about
- **Behavior policy**:
  A policy is used to generate behavior

The learning is from data ”off” the target policy
= off-policy learning

On-policy and Off-policy comparison

On-Policy	Off-Policy
Simpler	More complex
Faster convergence	Greater variance Slower convergence
	More general, more powerful When Target policy = Behavior policy it is an On-policy
	Can learn from data generated by a conventional non-learning controller or from a human

Off-policy prediction

Let's assume:

- Both target (π) and behavior (b) policies are fixed and given
- Goal is to estimate v_π and q_π
- Given: episodes following b ($b \neq \pi$)

Off-policy prediction

Let’s assume:

- Both target (π) and behavior (b) policies are fixed and given
- Goal is to estimate v_π and q_π
- Given: episodes following b ($b \neq \pi$)

Assumption of coverage:

To use episodes from b to estimate values for π, it is required that every action taken under π is also taken, at least occasionally, under b.

$$\pi(a|s) > 0 \text{ implies } b(a|s) > 0$$

Off-policy prediction

- The behavior policy b must be stochastic in states where it is not identical to π
- The target policy π may be deterministic

Off-policy prediction

- The behavior policy b must be stochastic in states where it is not identical to π
- The target policy π may be deterministic

deterministic greedy policy

ε-greedy policy

Off-policy prediction

- The behavior policy *b* must be stochastic in states where it is not identical to π
- The target policy π may be deterministic

Target policy	Behavior policy
deterministic greedy policy	ε-greedy policy

Definitions

- **Importance sampling:**
  General technique for estimating expected values under one distribution given samples from another distribution
- **Importance-sampling ratio:**
  Weighting returns according to the relative probability of their trajectories occurring under the target and behavior policies

Expected Value

- **Expected value:**
  is a weighted average of a function of Random Variable

$$\mathbb{E}_{p_\theta}[h(X)] = \sum_{i=1}^{\infty} h(x_i)p_\theta(x_i)$$

$$\mathbb{E}_{p_\theta}[h(X)] = \int_{\mathbb{R}} h(x)p_\theta(x)dx$$

https://www.youtube.com/watch?v=ivBtpzHcvpg

Expected Value

- **Expected value:**
  is a weighted average of a function of Random Variable
- **Law of Large Numbers:**
  The average of the results obtained from a large number of trials should be close to the expected value and will tend to become closer to the expected value as more trials are performed

$$\mathbb{E}_{p_\theta}[h(X)] = \sum_{i=1}^{\infty} h(x_i)p_\theta(x_i)$$

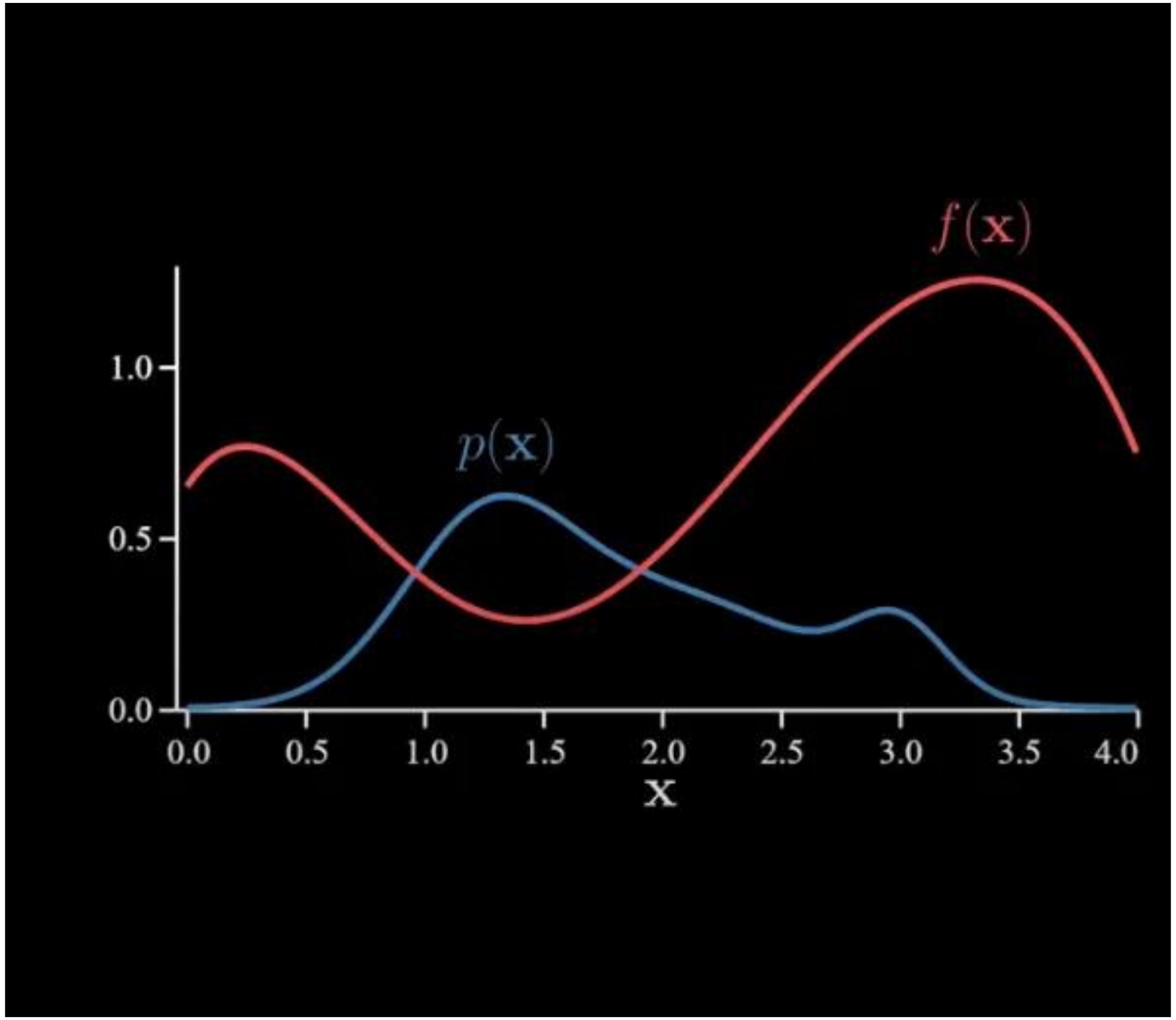
$$\mathbb{E}_{p_\theta}[h(X)] = \int_{\mathbb{R}} h(x)p_\theta(x)dx$$

$$\frac{1}{N}\sum_{i=1}^{N} h(x_i) \approx \mathbb{E}_{p_\theta}[(h(X)]$$

https://www.youtube.com/watch?v=ivBtpzHcvpg

MC method

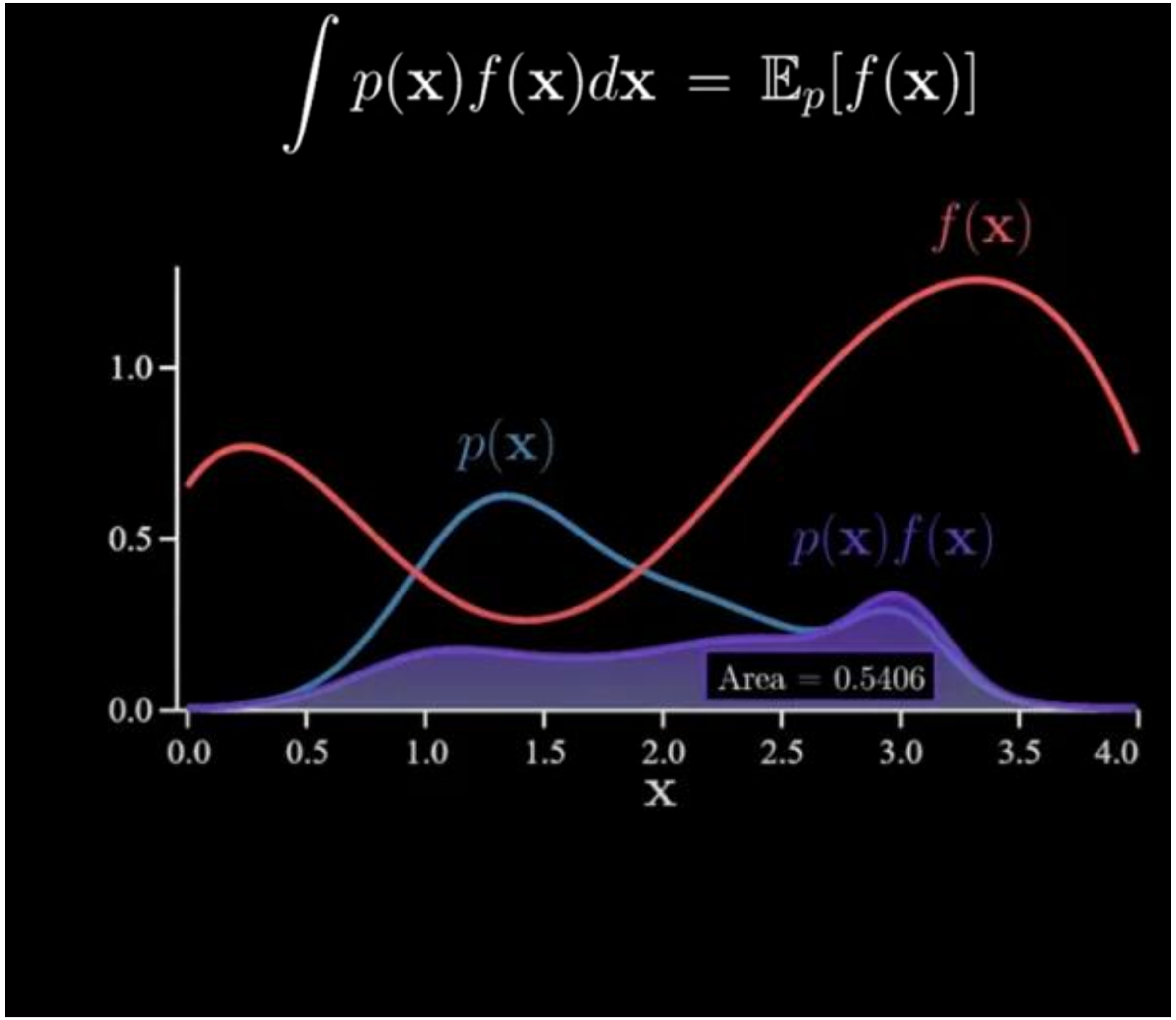
- *x* is a random variable
- *f*(*x*) is a scaler function of *x*
- *p*(*x*) is a probability density function of *x*



https://www.youtube.com/watch?v=C3p2wI4RAi8

MC method

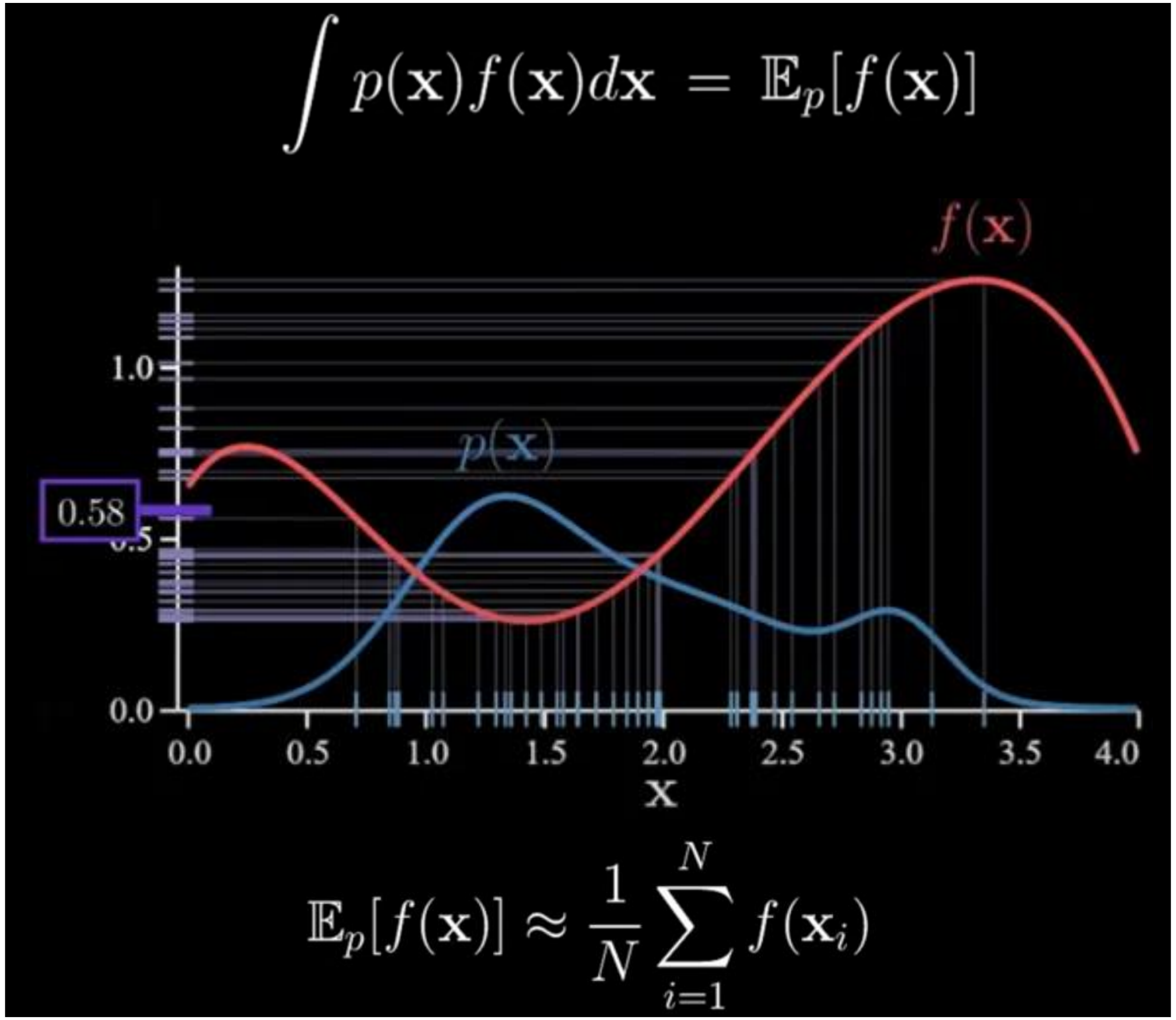
- *x* is a random variable
- *f*(*x*) is a scaler function of *x*
- *p*(*x*) is a probability density function of *x*
- Expected value is the probability weighted average



https://www.youtube.com/watch?v=C3p2wI4RAi8

MC method

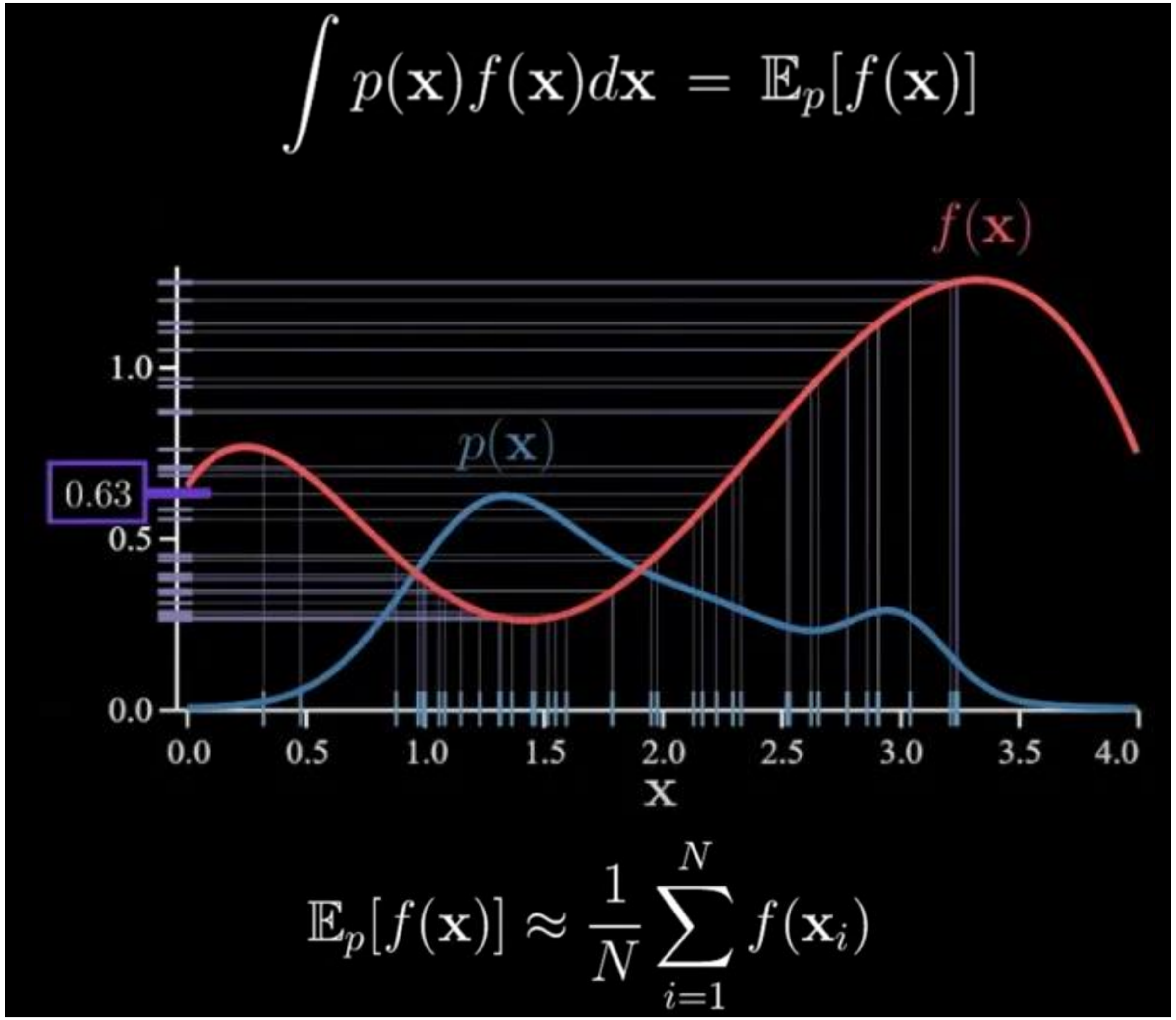
- *x* is a random variable
- *f*(*x*) is a scaler function of *x*
- *p*(*x*) is a probability density function of *x*
- Expected value is the probability weighted average
- Sample x according to *p*(*x*) The average approximates the expected value



https://www.youtube.com/watch?v=C3p2wI4RAi8

MC method

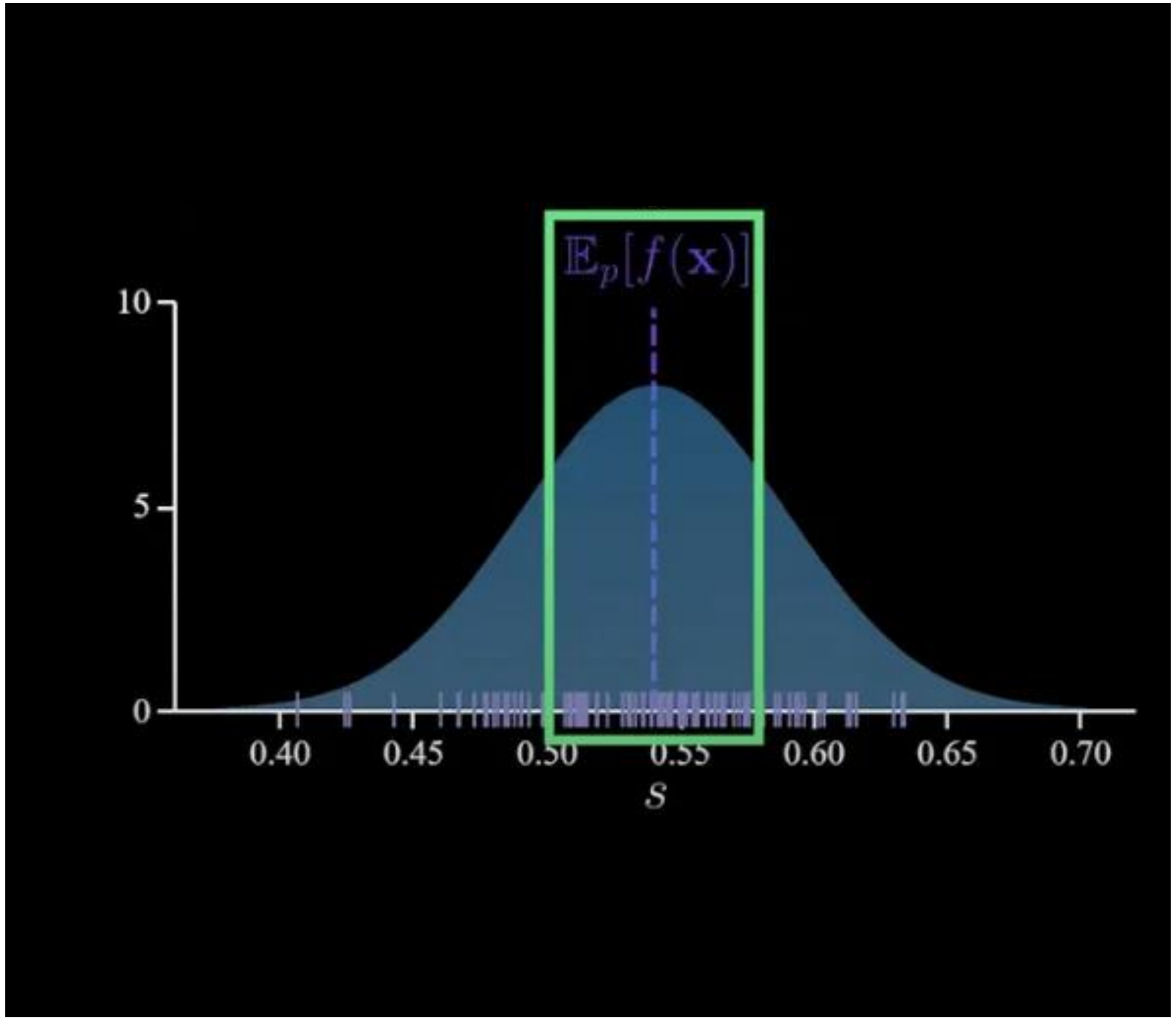
- *x* is a random variable
- *f*(*x*) is a scaler function of *x*
- *p*(*x*) is a probability density function of *x*
- Expected value is the probability weighted average
- Sample x according to *p*(*x*) The average approximates the expected value



https://www.youtube.com/watch?v=C3p2wI4RAi8

MC method

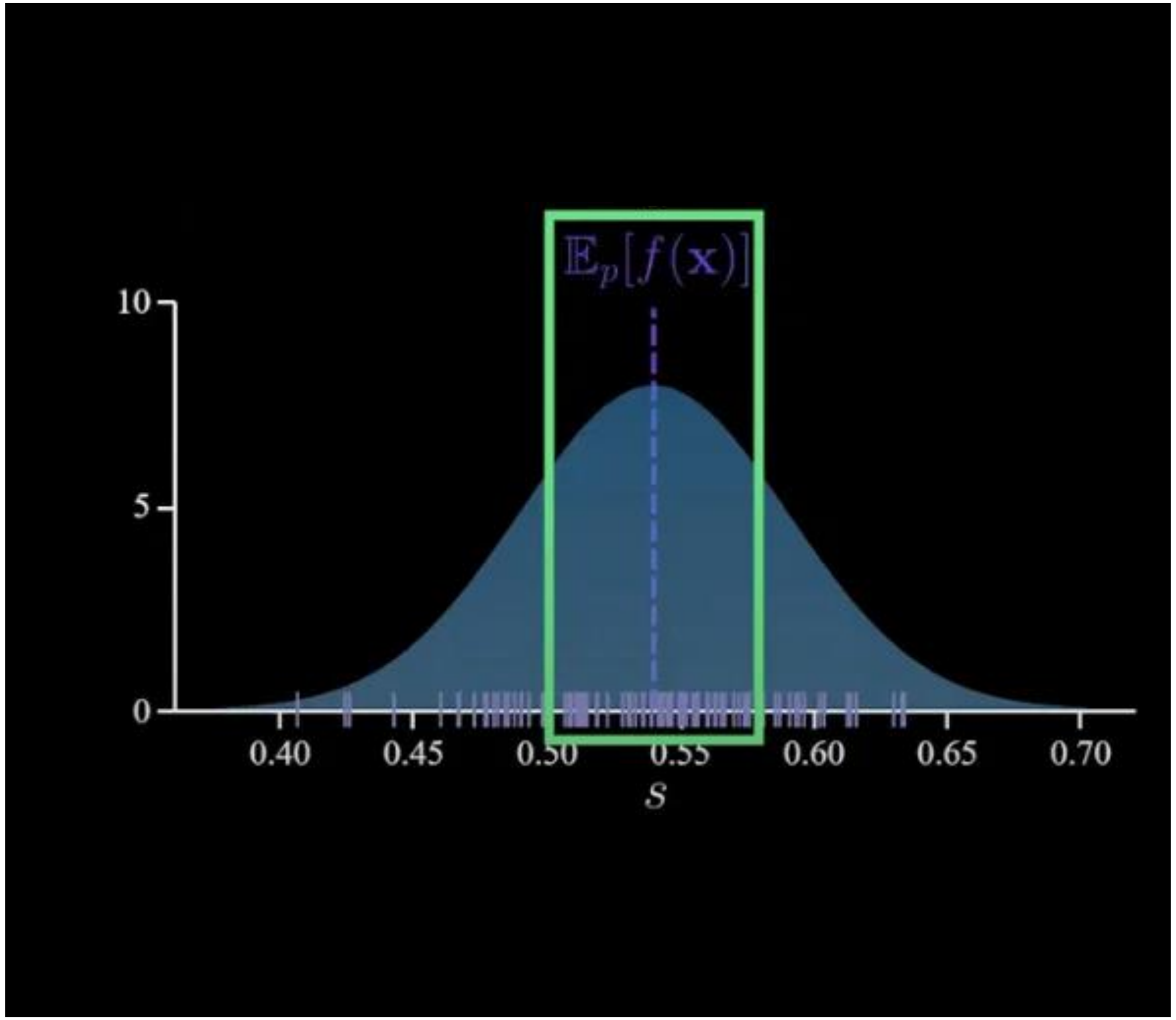
- *x* is a random variable
- *f*(*x*) is a scaler function of *x*
- *p*(*x*) is a probability density function of *x*
- Expected value is the probability weighted average
- Sample x according to *p*(*x*) The average approximates the expected value
- Sample average has its own distribution



https://www.youtube.com/watch?v=C3p2wI4RAi8

Central Limit Theorem

The distribution of a normalized version of the sample mean converges to a standard normal distribution even if the original variables themselves are not normally distributed



https://www.youtube.com/watch?v=C3p2wI4RAi8

Expected Value

Challenges:

- Can not sample from p_θ
- Inefficient to sample from p_θ
    - Large Variance
    - Rare event probabilities
- Not normalized p_θ

$$\mathbb{E}_{p_\theta}[h(X)] = \sum_{i=1}^{\infty} h(x_i)p_\theta(x_i)$$

$$\mathbb{E}_{p_\theta}[h(X)] = \int_{\mathbb{R}} h(x)p_\theta(x)dx$$

$$\frac{1}{N}\sum_{i=1}^{N} h(x_i) \approx \mathbb{E}_{p_\theta}[(h(X)]$$

https://www.youtube.com/watch?v=ivBtpzHcvpg

Importance sampling

- Use a sampling distribution q (proxy or proposal distribution)

$$\mathbb{E}_{p_\theta}[h(X)] = \int_{\mathbb{R}} h(x) p_\theta(x) dx$$

https://www.youtube.com/watch?v=ivBtpzHcvpg

Importance sampling

- Use a sampling distribution *q* (proxy or proposal distribution)
- Multiply by 1 (probabilistic one)

$$\mathbb{E}_{p_\theta}[h(X)] = \int_{\mathbb{R}} h(x) p_\theta(x) dx$$

$$\mathbb{E}_{p_\theta}[h(X)] = \int_{\mathbb{R}} h(x) \frac{q_\phi(x)}{q_\phi(x)} p_\theta(x) dx$$

https://www.youtube.com/watch?v=ivBtpzHcvpg

Importance sampling

- Use a sampling distribution *q* (proxy or proposal distribution)
- Multiply by 1 (probabilistic one)
- Switch distributions

$$\mathbb{E}_{p_\theta}[h(X)] = \int_{\mathbb{R}} h(x) p_\theta(x) dx$$

$$\mathbb{E}_{p_\theta}[h(X)] = \int_{\mathbb{R}} h(x) \frac{q_\phi(x)}{q_\phi(x)} p_\theta(x) dx$$

$$\mathbb{E}_{p_\theta}[h(X)] = \int_{\mathbb{R}} h(x) \frac{p_\theta(x)}{q_\phi(x)} q_\phi(x) dx$$

https://www.youtube.com/watch?v=ivBtpzHcvpg

Importance sampling

- Use a sampling distribution *q* (proxy or proposal distribution)
- Multiply by 1 (probabilistic one)
- Switch places
- Change expectation value to refer to the sampling PDF

$$\mathbb{E}_{p_\theta}[h(X)] = \int_{\mathbb{R}} h(x) p_\theta(x) dx$$

$$\mathbb{E}_{p_\theta}[h(X)] = \int_{\mathbb{R}} h(x) \frac{q_\phi(x)}{q_\phi(x)} p_\theta(x) dx$$

$$\mathbb{E}_{p_\theta}[h(X)] = \int_{\mathbb{R}} h(x) \frac{p_\theta(x)}{q_\phi(x)} q_\phi(x) dx$$

$$\mathbb{E}_{q_\phi}[h(X)] = \int_{\mathbb{R}} h(x) \frac{p_\theta(x)}{q_\phi(x)} q_\phi(x) dx$$

https://www.youtube.com/watch?v=ivBtpzHcvpg

Importance sampling

- Use a sampling distribution *q* (proxy or proposal distribution)
- Multiply by 1 (probabilistic one)
- Switch places
- Change expectation value to refer to the sampling PDF
- Change the function of random variable

$$\mathbb{E}_{p_\theta}[h(X)] = \int_{\mathbb{R}} h(x) p_\theta(x) dx$$

$$\mathbb{E}_{p_\theta}[h(X)] = \int_{\mathbb{R}} h(x) \frac{q_\phi(x)}{q_\phi(x)} p_\theta(x) dx$$

$$\mathbb{E}_{p_\theta}[h(X)] = \int_{\mathbb{R}} h(x) \frac{p_\theta(x)}{q_\phi(x)} q_\phi(x) dx$$

$$\mathbb{E}_{q_\phi}[h(X)] = \int_{\mathbb{R}} h(x) \frac{p_\theta(x)}{q_\phi(x)} q_\phi(x) dx$$

$$\mathbb{E}_{q_\phi}[h'(X)] = \int_{\mathbb{R}} h(x) \frac{p_\theta(x)}{q_\phi(x)} q_\phi(x) dx$$

https://www.youtube.com/watch?v=ivBtpzHcvpg

Importance sampling

- Use a sampling distribution *q* (proxy or proposal distribution)
- Multiply by 1 (probabilistic one)
- Switch places
- Change expectation value to refer to the sampling PDF
- Change the function of random variable
- Importance weight likelihood ration or **Importance-sampling ratio**

$$\mathbb{E}_{p_\theta}[h(X)] = \int_{\mathbb{R}} h(x) p_\theta(x) dx$$

$$\mathbb{E}_{p_\theta}[h(X)] = \int_{\mathbb{R}} h(x) \frac{q_\phi(x)}{q_\phi(x)} p_\theta(x) dx$$

$$\mathbb{E}_{p_\theta}[h(X)] = \int_{\mathbb{R}} h(x) \frac{p_\theta(x)}{q_\phi(x)} q_\phi(x) dx$$

$$\mathbb{E}_{q_\phi}[h(X)] = \int_{\mathbb{R}} h(x) \frac{p_\theta(x)}{q_\phi(x)} q_\phi(x) dx$$

$$\mathbb{E}_{q_\phi}[h'(X)] = \int_{\mathbb{R}} h(x) \boxed{\frac{p_\theta(x)}{q_\phi(x)}} q_\phi(x) dx$$

https://www.youtube.com/watch?v=ivBtpzHcvpg

Importance sampling

- Use a sampling distribution *q* (proxy or proposal distribution)
- Multiply by 1 (probabilistic one)
- Switch places
- Change expectation value to refer to the sampling PDF
- Change the function of random variable
- Importance weight likelihood ration or **Importance-sampling ratio**

$$\mathbb{E}_{q_\phi}[h'(X)] = \int_{\mathbb{R}} h(x) \frac{p_\theta(x)}{q_\phi(x)} q_\phi(x) dx$$

$$\approx \frac{1}{N} \sum_{i=1}^{N} h(x_i) \frac{p_\theta(x_i)}{q_\phi(x_i)}$$

https://www.youtube.com/watch?v=ivBtpzHcvpg

Importance sampling

- Use a sampling distribution *q* (proxy or proposal distribution)
- Multiply by 1 (probabilistic one)
- Switch places
- Change expectation value to refer to the sampling PDF
- Change the function of random variable
- Importance weight likelihood ration or **Importance-sampling ratio**

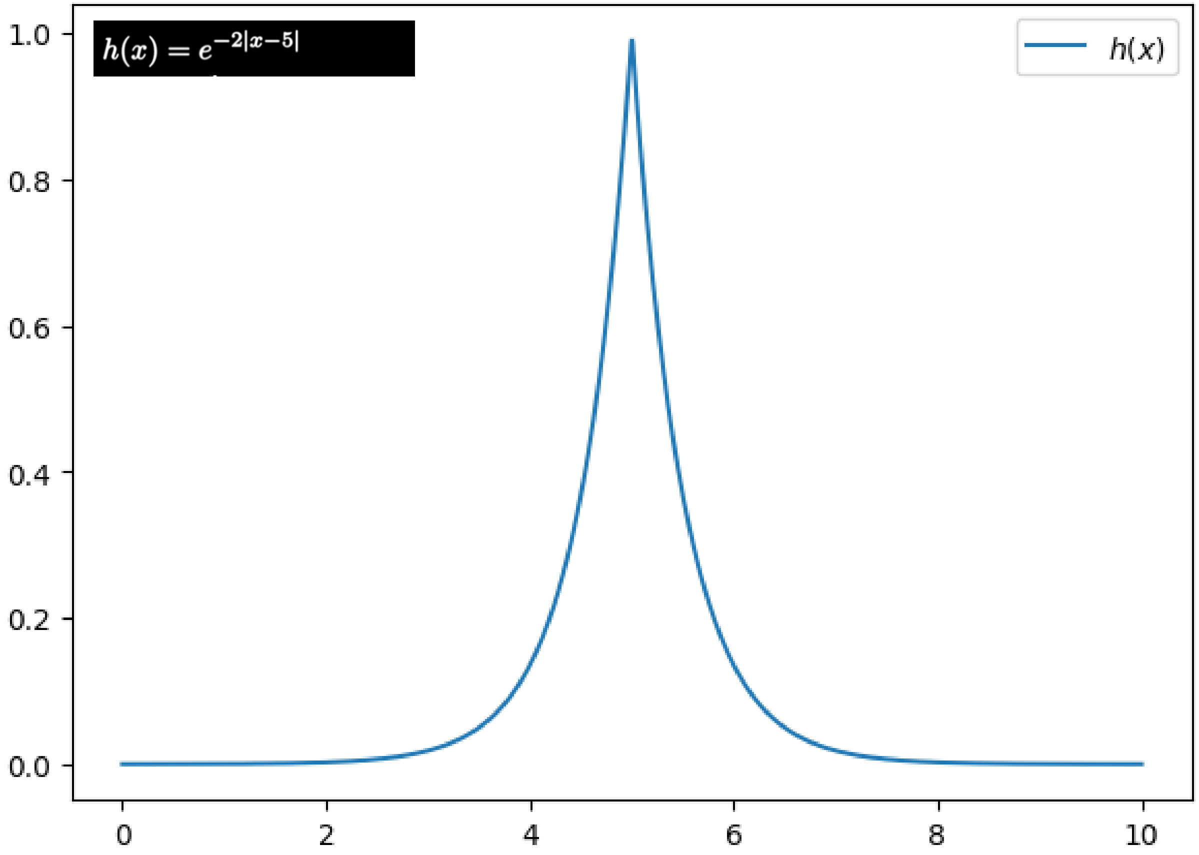
$$\mathbb{E}_{q_\phi}[h'(X)] = \int_{\mathbb{R}} h(x)\frac{p_\theta(x)}{q_\phi(x)}q_\phi(x)dx$$

$$\approx \frac{1}{N}\sum_{i=1}^{N} h(x_i)\frac{p_\theta(x_i)}{q_\phi(x_i)}$$

But how do I choose q_ϕ?

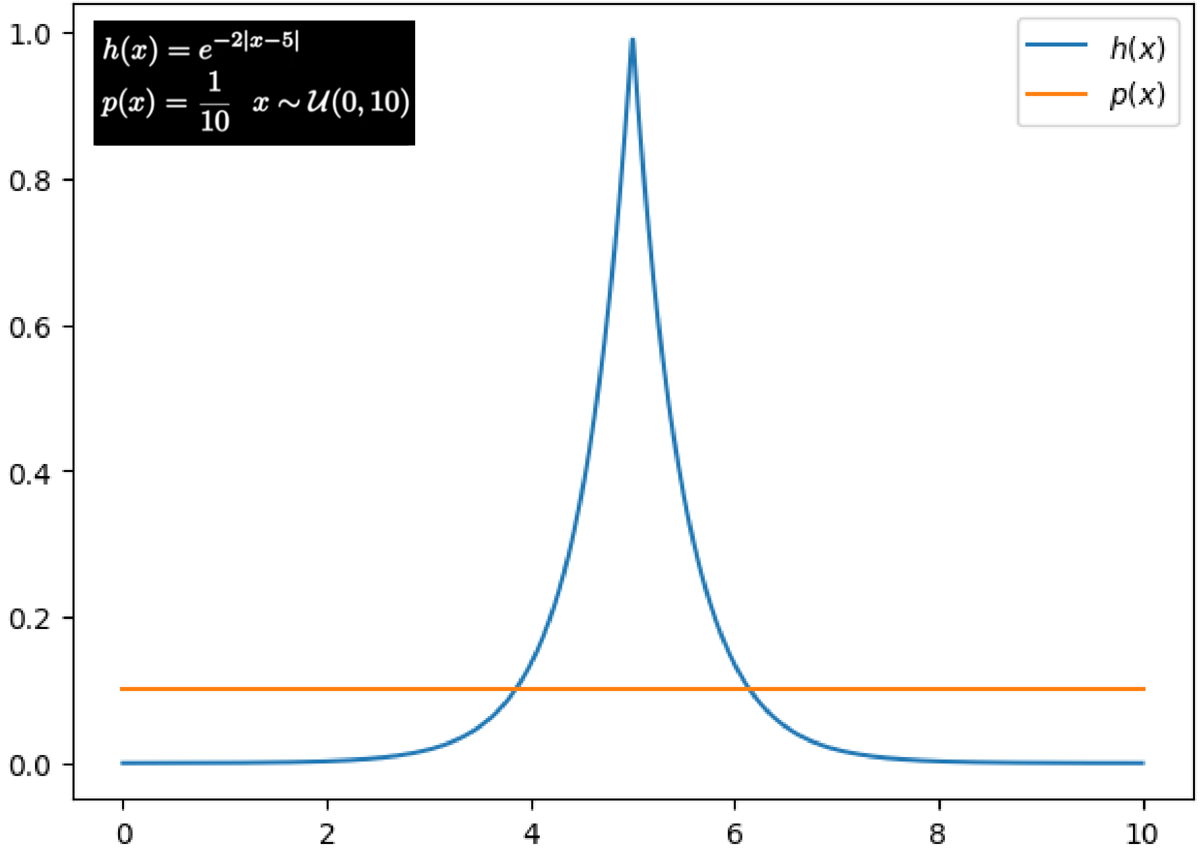
https://www.youtube.com/watch?v=ivBtpzHcvpg

Importance sampling example



$$\int_0^{10} e^{-2|x-5|}dx = 1 - \frac{1}{e^{10}} \approx 0.99995$$

Importance sampling example



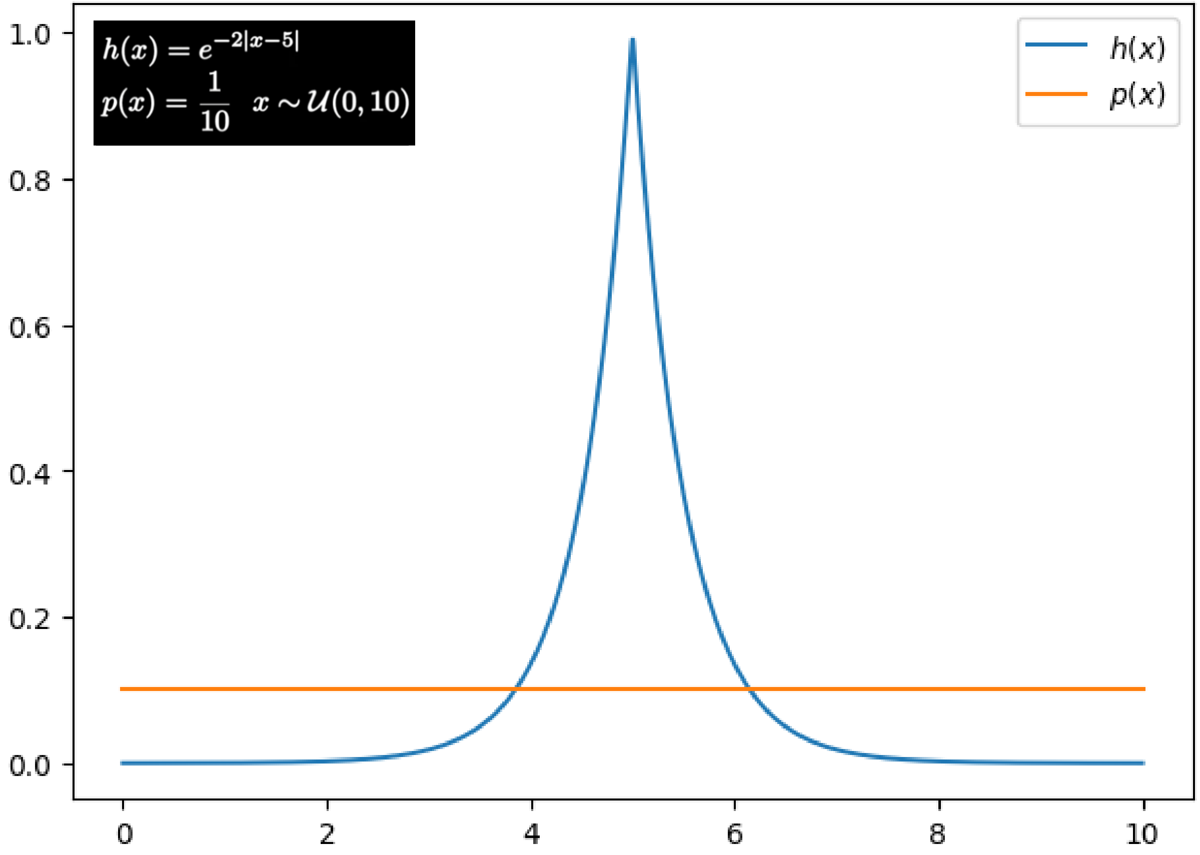
$$\int_0^{10} e^{-2|x-5|} dx = 1 - \frac{1}{e^{10}} \approx 0.99995$$

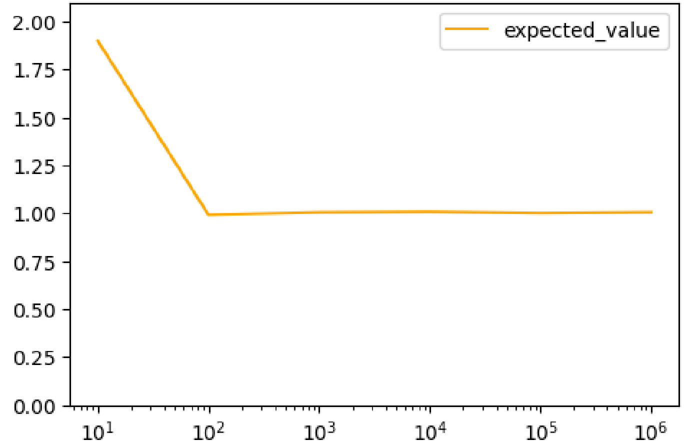
$$= 10 \int_0^{10} e^{-2|x-5|} \frac{1}{10} dx$$

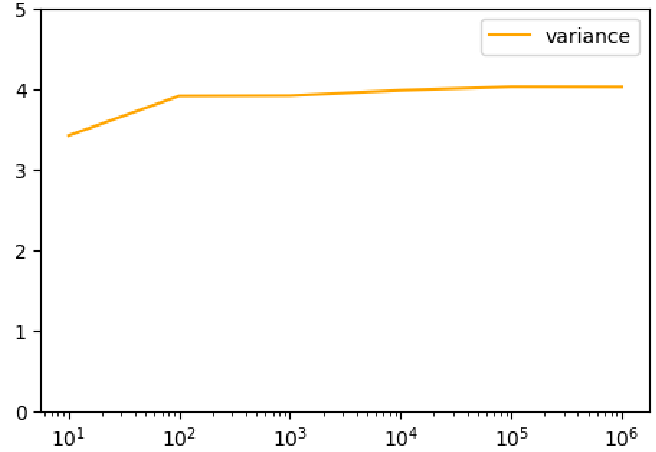
$$= 10 \int_0^{10} h(x)p(x)dx$$

$$\approx \frac{10}{N} \sum_{i=1}^{N} h(x_i) \quad x_i \sim \mathcal{U}(0,10)$$

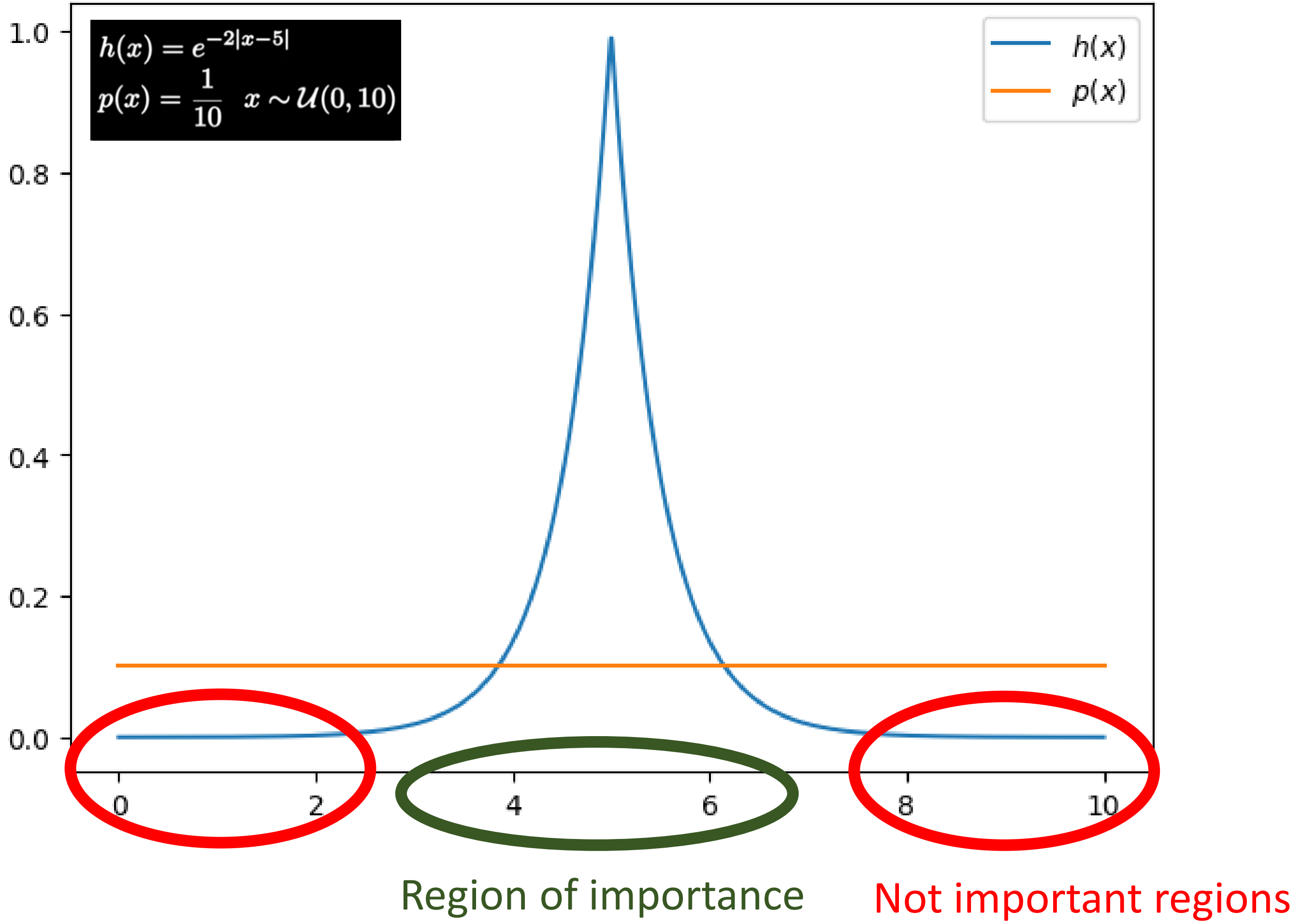
Importance sampling example

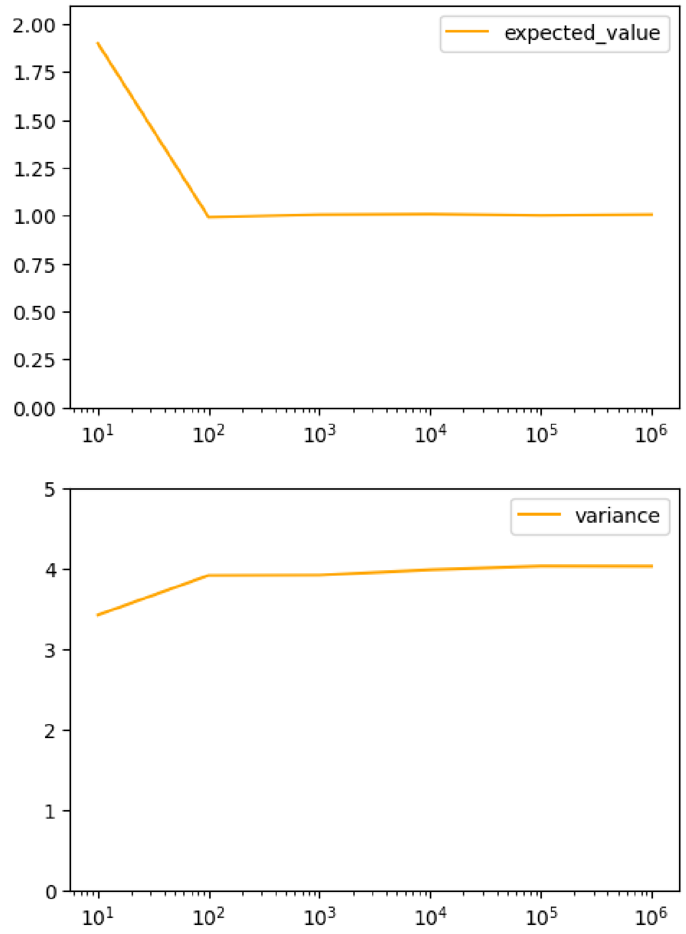




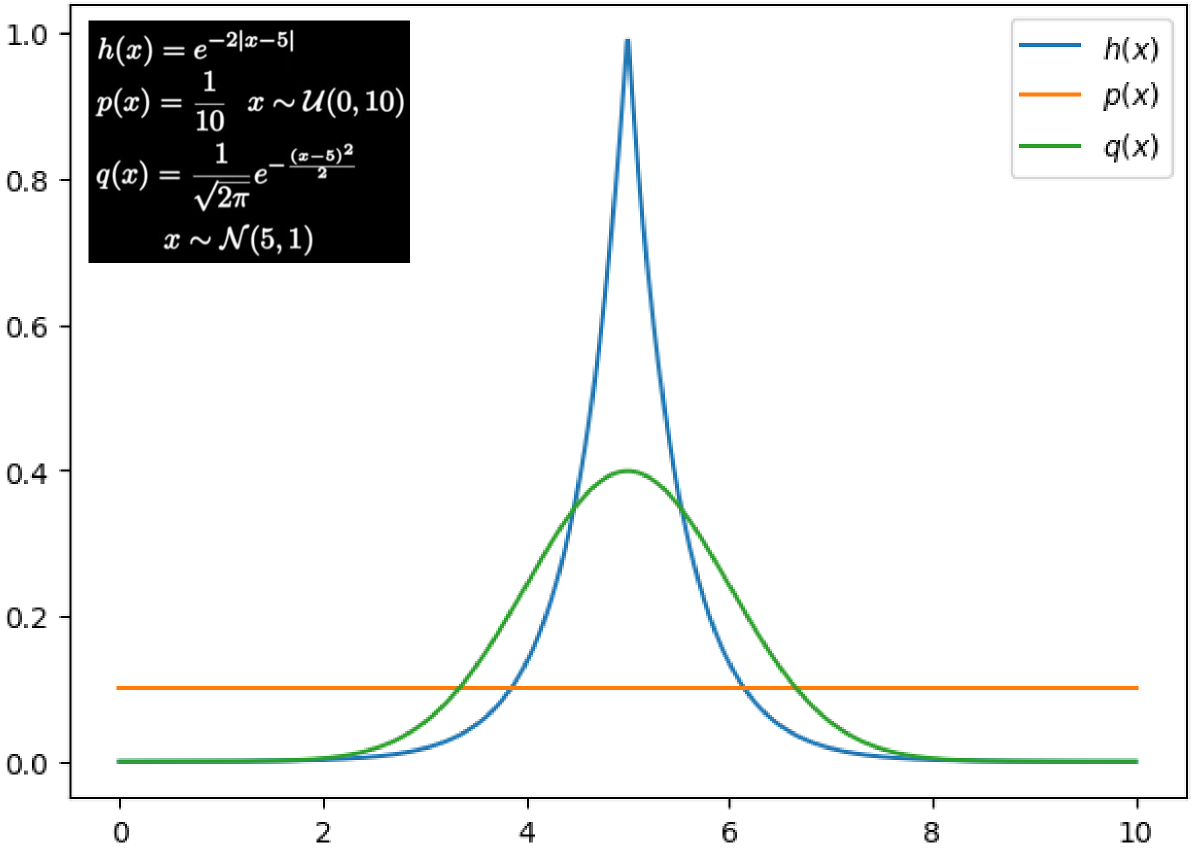


Importance sampling example





Importance sampling example



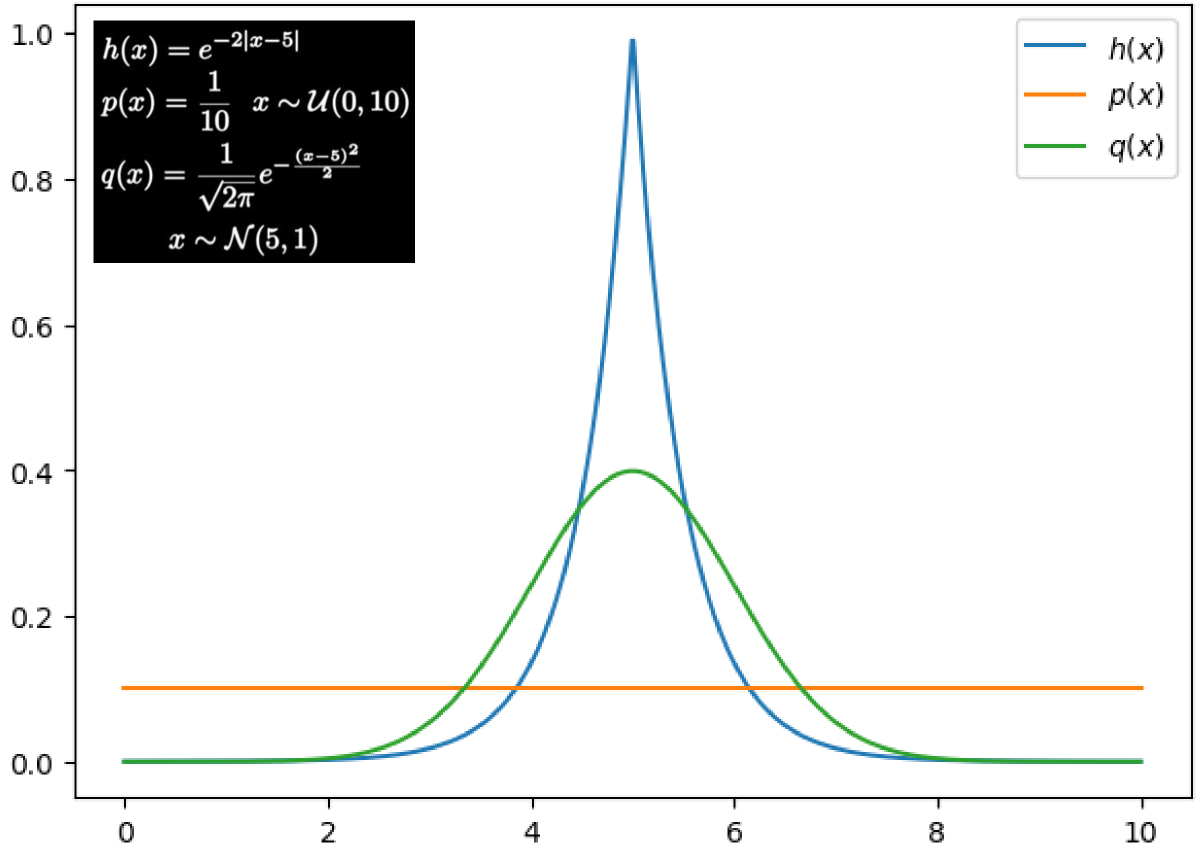
$$\int_0^{10} e^{-2|x-5|} dx = 1 - \frac{1}{e^{10}} \approx 0.99995$$

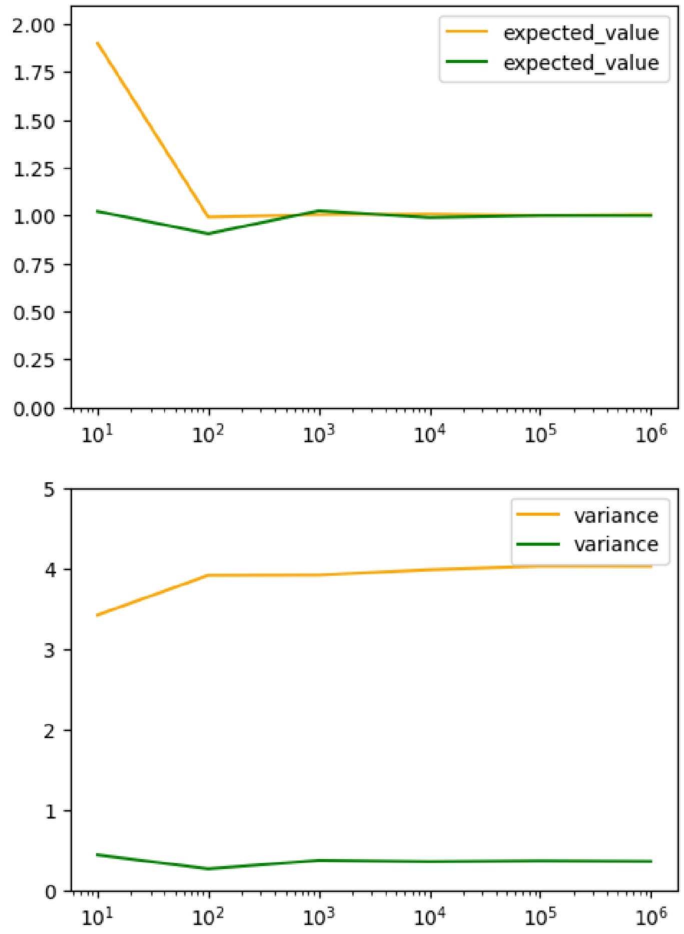
$$= 10 \int_0^{10} h(x) p(x) dx$$

$$= 10 \int_0^{10} h(x) \frac{p(x)}{q(x)} q(x) dx$$

$$\approx 10 \sum_{i=1}^{N} h(x_i) \frac{p(x_i)}{q(x_i)}$$

Importance sampling example





Prediction with Importance Sampling

- Almost all off-policy methods utilize importance sampling
- Apply importance sampling to off-policy learning by weighting returns according to the relative probability of their trajectories occurring under the target and behavior policies

Prediction with Importance Sampling

- Given a starting state S_t
- A subsequent state-action trajectory is: A_t, S_{t+1}, A_{t+1}, … S_T
- The probability this trajectory occurring under π

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

Prediction with Importance Sampling

- Given a starting state S_t
- A subsequent state-action trajectory is: A_t, S_{t+1}, A_{t+1}, … S_T
- The probability this trajectory occurring under π

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

- The relative probability of the trajectory under the target and behavior policies (the importance sampling ratio)

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

Prediction with Importance Sampling

- Given a starting state S_t
- A subsequent state-action trajectory is: $A_t, S_{t+1}, A_{t+1}, \ldots S_T$
- The probability this trajectory occurring under π

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

- The relative probability of the trajectory under the target and behavior policies (the importance sampling ratio)

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

Not dependent on the MDP

Prediction with Importance Sampling

- Goal is to estimate the expected returns (values) under the target policy

$$v_\pi(s)$$

Prediction with Importance Sampling

- Goal is to estimate the expected returns (values) under the target policy

$$v_\pi(s)$$

- Only have returns G_t due to the behavior policy and the expected value referring to the behavior policy

$$\mathbb{E}[G_t | S_t = s] = v_b(s)$$

Prediction with Importance Sampling

- Goal is to estimate the expected returns (values) under the target policy

$$v_\pi(s)$$

- Only have returns G_t due to the behavior policy and the expected value referring to the behavior policy

$$\mathbb{E}[G_t|S_t=s] = v_b(s)$$

- Using the importance-sampling ration to transform

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

Important sampling variants

- Using a batch of episodes following policy *b*
- Number time steps in a way that increases across episode boundaries $\mathcal{T}(s)$

- Ordinary importance sampling

$$V(s) \doteq \frac{\sum_{t \in \mathcal{T}(s)} \rho_{t:T(t)-1} G_t}{|\mathcal{T}(s)|}$$

Unbiased
Higher variance
(Can be extreme)

- Weighted importance sampling

$$V(s) \doteq \frac{\sum_{t \in \mathcal{T}(s)} \rho_{t:T(t)-1} G_t}{\sum_{t \in \mathcal{T}(s)} \rho_{t:T(t)-1}}$$

Biased
(the bias converges asymptotically to zero)
Lower variance

Incremental implementation

- In ordinary importance sampling, the returns are scaled by the importance sampling ratio then simply averaged
- In weighted importance sampling form a weighted average of the returns, and a slightly different incremental algorithm is required

Incremental implementation (weighted)

- Sequence of returns $G_1, G_2, \ldots, G_{n-1}$
- Corresponding random weight $W_i = \rho_{t:T(t)-1}$
- Want the estimate and keep it up to date

$$V_n \doteq \frac{\sum_{k=1}^{n-1} W_k G_k}{\sum_{k=1}^{n-1} W_k}, \qquad n \geq 2$$

Incremental implementation (weighted)

- Sequence of returns $G_1, G_2, \ldots, G_{n-1}$
- Corresponding random weight $W_i = \rho_{t:T(t)-1}$
- Want the estimate and keep it up to date

$$V_n \doteq \frac{\sum_{k=1}^{n-1} W_k G_k}{\sum_{k=1}^{n-1} W_k}, \qquad n \geq 2$$

- With the cumulative sum C_n of the weights given to the first n returns the update rule for V_n

$$V_{n+1} \doteq V_n + \frac{W_n}{C_n}\Big[G_n - V_n\Big], \qquad n \geq 1$$

$$C_{n+1} \doteq C_n + W_{n+1}$$

$$C_0 \doteq 0$$

Pseudocode

Off-policy MC control, for estimating $\pi \approx \pi_*$

Initialize, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$:
    $Q(s,a) \in \mathbb{R}$ (arbitrarily)
    $C(s,a) \leftarrow 0$
    $\pi(s) \leftarrow \arg\max_a Q(s,a)$ (with ties broken consistently)

Loop forever (for each episode):
    $b \leftarrow$ any soft policy
    Generate an episode using 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, $t = T-1, T-2, \ldots, 0$:
        $G \leftarrow \gamma G + R_{t+1}$
        $C(S_t, A_t) \leftarrow C(S_t, A_t) + W$
        $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \frac{W}{C(S_t, A_t)} [G - Q(S_t, A_t)]$
        $\pi(S_t) \leftarrow \arg\max_a Q(S_t, a)$ (with ties broken consistently)
        If $A_t \neq \pi(S_t)$ then exit For loop
        $W \leftarrow W \frac{1}{b(A_t|S_t)}$

MC summary

- learn optimal behavior directly from interaction with the environment, with no model of the environment's dynamics
- can be used with simulation or sample models
- easy and efficient to focus Monte Carlo methods on a small subset of the states
- may be less harmed by violations of the Markov property (do not update their value estimates on the basis of the value estimates of successor states)

DP vs MC

DP	MC
Bootstrap	No Bootstrap
Model is required	Operate on sample experience No model is required

ELTE | FACULTY OF INFORMATICS

Thank you for your attention!