Chapter 5: Monte Carlo Methods

Seungjae Ryan Lee

endtoend.ai

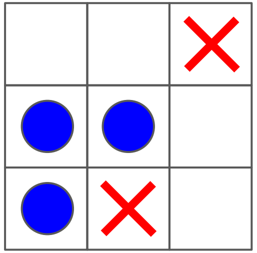
New method: Monte Carlo method

- Do not assume complete knowledge of environment
  - Only need *experience*
  - Can use *simulated* experience
- Average sample returns
- Use General Policy Iteration (GPI)
  - *Prediction*: compute value functions
  - *Policy Improvement*: improve policy from value functions
  - *Control*: discover optimal policy

Monte Carlo Prediction: v_π

- Estimate v_π from sample return
- **Converges as more returns are observed**

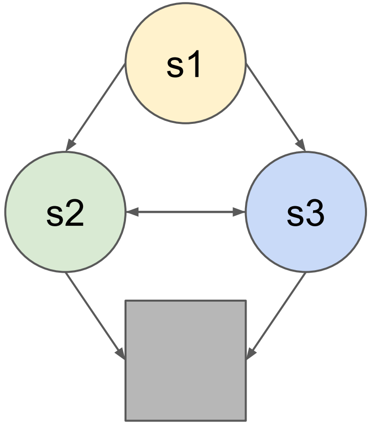
$s \longrightarrow$



Return 1 observed 10 times
Return 0 observed 2 times
Return -1 observed 0 times

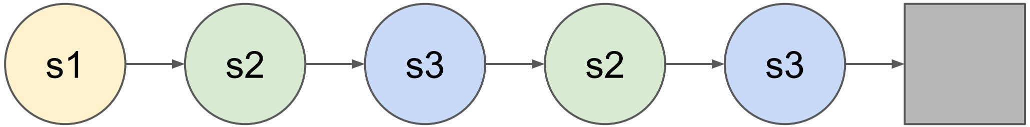
$$V(s) = \frac{10 \times 1 + 2 \times 0 + 3 \times 0}{10 + 2} = \frac{10}{12} \simeq 0.857$$

First-visit MC vs. Every-visit MC



- First-visit
  - Average of returns following **first** visits to states
  - Studied widely
  - Primary focus for this chapter
- Every-visit
  - Average returns following **all** visits to states
  - Extended naturally to function approximation (Ch. 9) and eligibility traces (Ch. 12)

Sample Trajectory:



endtoend.ai

First-visit MC prediction in Practice: v_π

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

    $G \leftarrow 0$

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

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

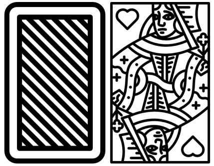
        Unless S_t appears in $S_0, S_1, \ldots, S_{t-1}$:

            Append G to $Returns(S_t)$

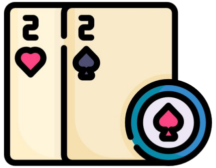
            $V(S_t) \leftarrow \text{average}(Returns(S_t))$

endtoend.ai

Blackjack Example



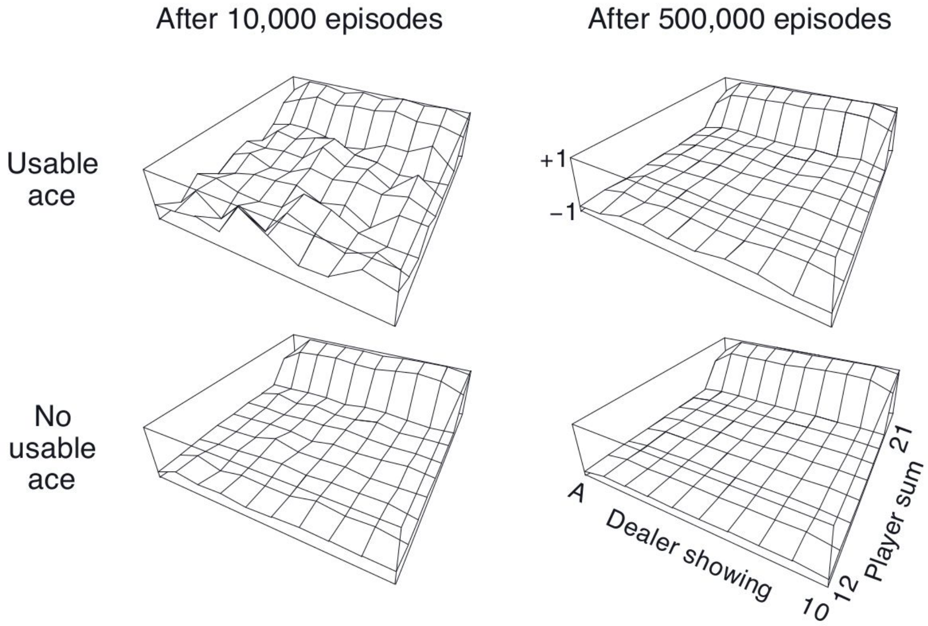
- **States**: (Sum of cards, Has usable ace, Dealer's card)
- **Action**: *Hit* (request card), *Stick* (stop)
- **Reward**: +1, 0, -1 for win, draw, loss
- **Policy**: request cards if and only if sum < 20



- Difficult to use DP although environment dynamics is known

Blackjack Example Results

- Less common experience have uncertain estimates
  - ex) States with usable ace

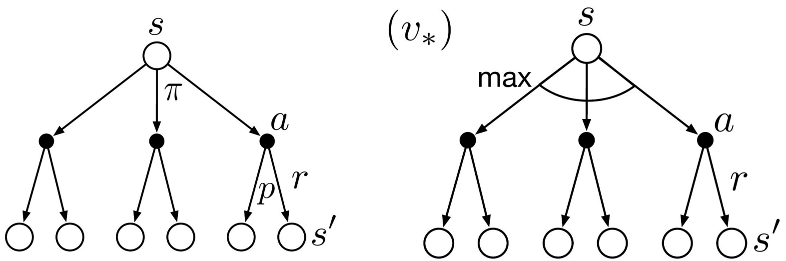


endtoend.ai

MC vs. DP

- No *bootstrapping*
- Estimates for each state are independent
- **Can estimate the value of a subset of all states**

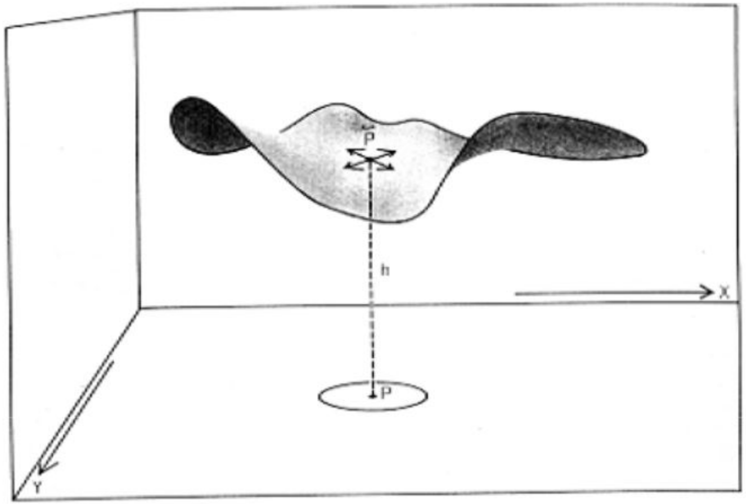
Monte Carlo



Dynamic Programming

Soap Bubble Example

- Compute shape of soap surface for a closed wire frame
- Height of surface is average of heights at neighboring points
- Surface must meet boundaries with the wire frame

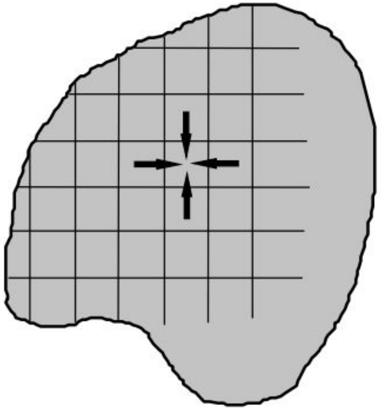


endtoend.ai

Soap Bubble Example: DP vs. MC

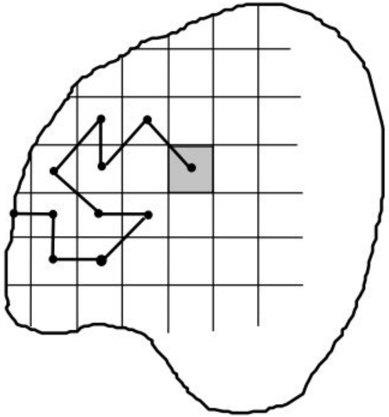
DP

- Update heights by its neighboring heights
- Iteratively sweep the grid



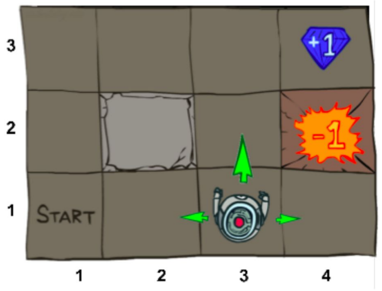
MC

- Take random walk until boundary is reached
- Average sampled boundary height

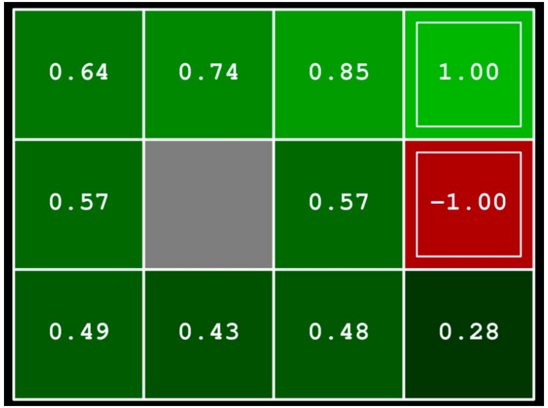


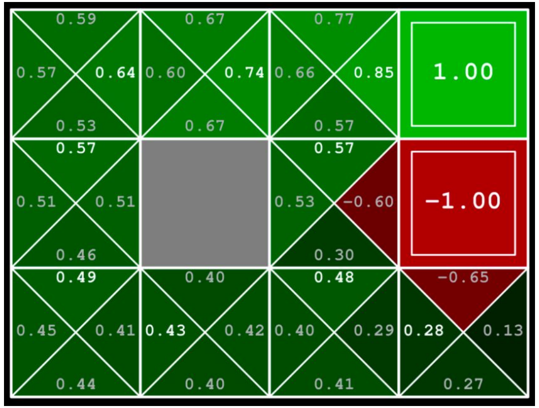
http://www-anw.cs.umass.edu/~barto/courses/cs687/Chapter%205.pdf

Monte Carlo Prediction: q_π



- More useful if model is not available
    - Can determine policy without model
- Converges quadratically to $N(s, a)$ when infinite samples
- **Need exploration**: all state-action pairs need to be visited infinitely





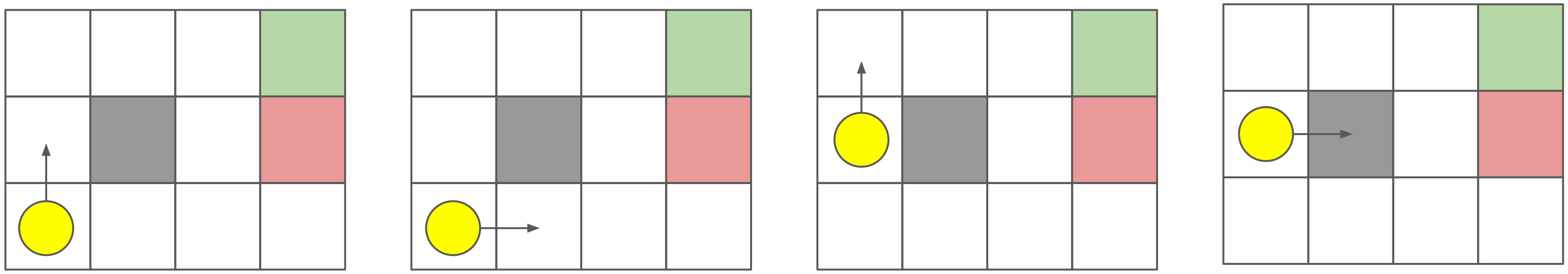
https://www.youtube.com/watch?v=qaMdN6LS9rA

endtoend.ai

Exploring Starts (ES)

- Specify state-action pair to start episode on
- Cannot be used when learning from actual interactions

$$(s_0\,,\,a_0)$$



endtoend.ai

Monte Carlo ES

- Control: approximate optimal policies
- Use Generalized Policy Iteration (GPI)
    - Maintain approximate policy and approximate value function
    - Policy evaluation: Monte Carlo Prediction for *one episode with start chosen by ES*
    - Policy Improvement: Greedy selection\
- No proof of convergence

$$\pi_0 \rightarrow q_{\pi_0} \rightarrow \pi_1 \rightarrow q_{\pi_1} \rightarrow \dots \rightarrow \pi_* \rightarrow q_*$$

Monte Carlo ES Pseudocode

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

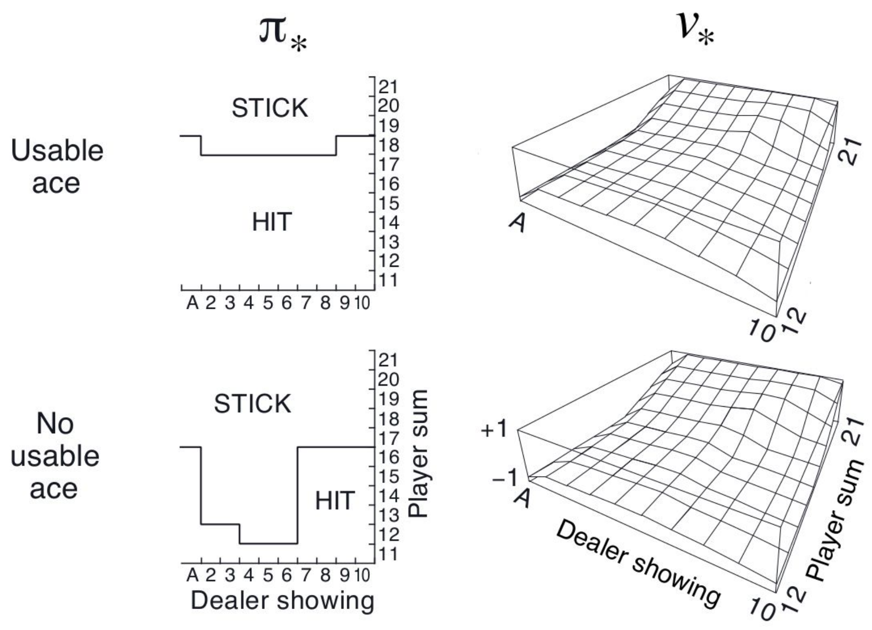
Initialize:
  $\pi(s) \in \mathcal{A}(s)$ (arbitrarily), for all $s \in \mathcal{S}$
  $Q(s, a) \in \mathbb{R}$ (arbitrarily), for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$
  $Returns(s, a) \leftarrow$ empty list, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$

Loop forever (for each episode):
  Choose $S_0 \in \mathcal{S}$, $A_0 \in \mathcal{A}(S_0)$ 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 $Returns(S_t, A_t)$
      $Q(S_t, A_t) \leftarrow \text{average}(Returns(S_t, A_t))$
      $\pi(S_t) \leftarrow \arg\max_a Q(S_t, a)$

endtoend.ai

Blackjack Example Revisited

- Prediction → Control



endtoend.ai

ε-soft Policy

- Avoid exploring starts → Add exploration to policy
- *Soft* policy: every action has nonzero probability of being selected

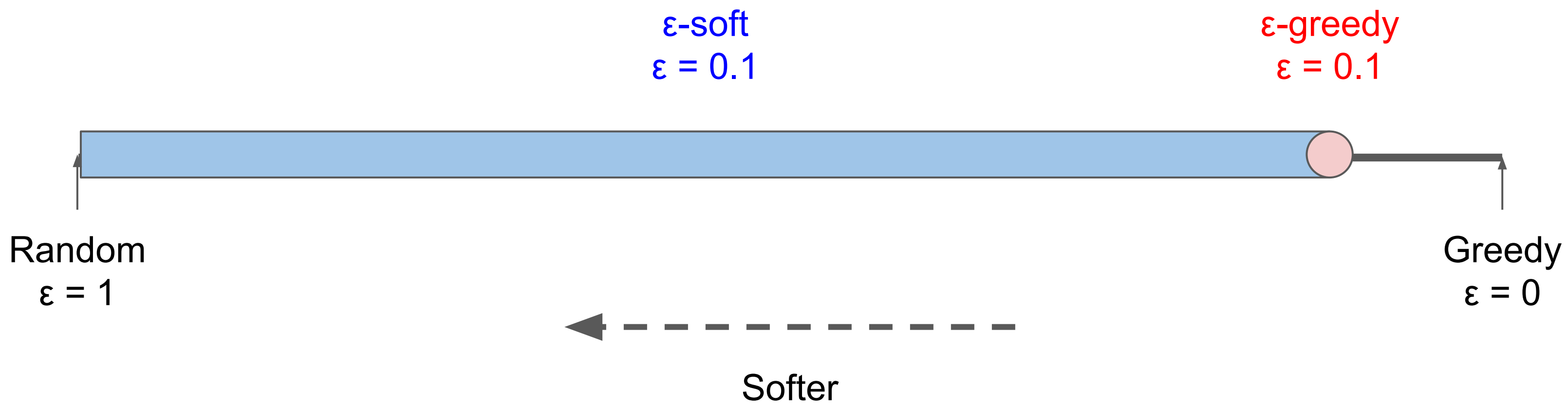
$$\pi(a \mid s) > 0$$

- *ε-soft* policy: every action has at least $\epsilon / |\mathcal{A}(s)|$ probability of being selected

$$\pi(a \mid s) \geq \frac{\epsilon}{|\mathcal{A}(s)|}$$

- ex) ε-greedy policy
  - Select greedily for $1 - \epsilon$ probability
  - Select randomly for ϵ probability (including greedy)

ε-soft vs ε-greedy



On-policy ε-soft MC control Pseudocode

- *On-policy*: Evaluate / improve policy that is used to make decisions

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

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

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

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

endtoend.ai

On-policy vs. Off-policy

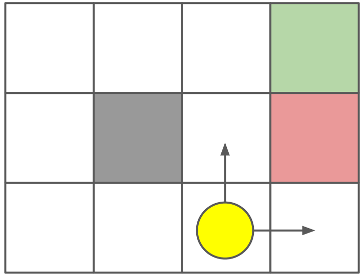
- *On-policy*: Evaluate / improve policy that is used to make decisions
  - Requires ε-soft policy: near optimal but never optimal
  - Simple, low variance
- *Off-policy*: Evaluate / improve policy different from that used to generate data
  - *Target policy* π : policy to evaluate
  - *Behavior policy* b : policy for taking actions
  - More powerful and general
  - High variance, slower convergence
  - Can learn from non-learning controller or human expert

Coverage assumption for off-policy learning

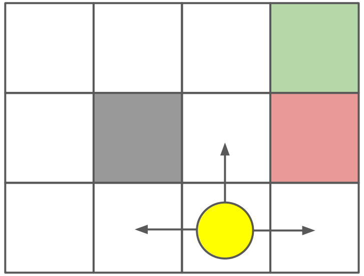
- To estimate values under π , all possible actions of π must be taken by b

$$\pi(a \mid s) > 0 \Rightarrow b(a \mid s) > 0$$

- b must be stochastic in states where $\pi(a \mid s) \neq b(a \mid s)$



π



b

endtoend.ai

Importance Sampling

- Trajectories have different probabilities under different policies
- Estimate expected value from one distribution given samples from another
- Weight returns by *importance sampling ratio*
  - Relative probability of trajectory occurring under the target and behavior policies

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

Ordinary Importance Sampling

- Zero bias but **unbounded variance**

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

- With single return:

$$V(s) = \rho_{t:T(t)-1} G$$

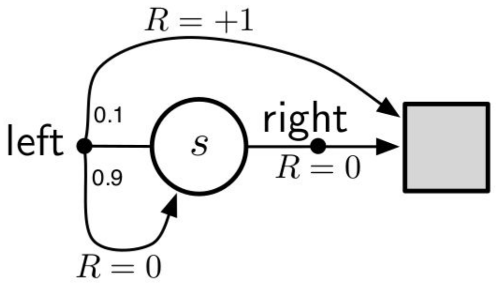
endtoend.ai

Ordinary Importance Sampling: Zero Bias

$$
\begin{aligned}
v_\pi(s) &= \mathbb{E}_\pi[G_t \mid S_t = s] \\
&= \sum \prod_{k=t}^{T-1} \pi(A_k \mid S_k) p(S_{k+1} \mid S_k, A_k) G_t \\
&= \rho_{t:T-1} \sum \prod_{k=t}^{T-1} b(A_k \mid S_k) p(S_{k+1} \mid S_k, A_k) G_t \\
&= \rho_{t:T-1} \mathbb{E}_b[G_t \mid S_t = s]
\end{aligned}
$$

Ordinary Importance Sampling: Unbounded Variance

- 1-state, 2-action undiscounted MDP
- Off-policy first-visit MC



$$\pi(\text{left}|s) = 1$$

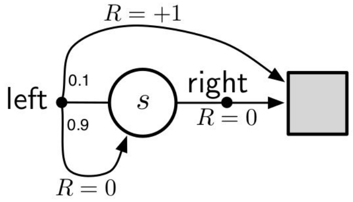
$$b(\text{left}|s) = \frac{1}{2}$$

- Variance of an estimator:

$$\mathrm{Var}[X] = \mathbb{E}[X^2] - \bar{X}^2 = \mathbb{E}[X^2] - 1$$

Ordinary Importance Sampling: Unbounded Variance

- Just consider all-**left** episodes with different lengths
  - Any trajectory with **right** has importance sampling ratio of 0
  - All-**left** trajectory have importance sampling ratio of 2^T



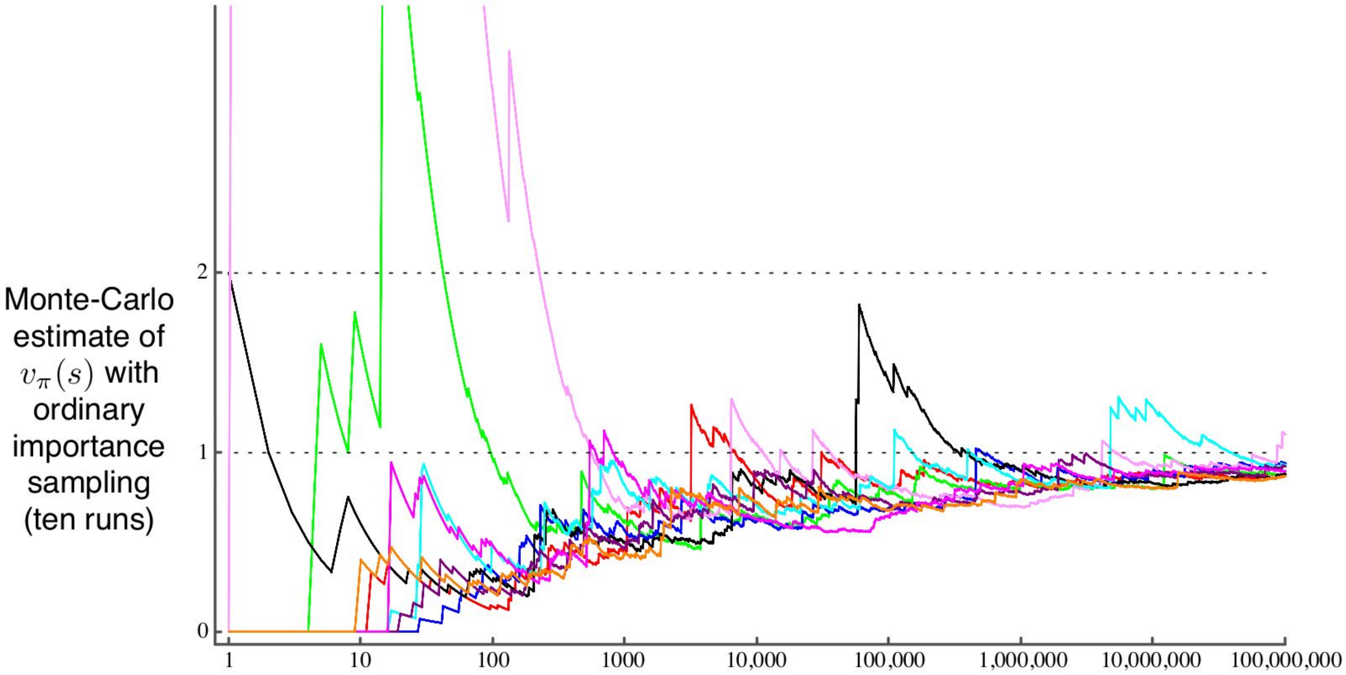
$$\pi(\text{left}|s) = 1$$

$$b(\text{left}|s) = \frac{1}{2}$$

$$\begin{aligned}
\mathbb{E}_b[(\rho G_0)^2] &= \mathbb{E}_b[\rho^2] \\
&= \sum_{T=1}^{\infty} \left(p_{trajectory} \rho^2 \right) \\
&= \sum_{T=1}^{\infty} \left(b(\text{left} \mid s)^T p(s \mid s, \text{left})^{T-1} p(t \mid s, \text{left}) \rho^2 \right) \\
&= \sum_{T=1}^{\infty} \left(\frac{1}{2^T} \times 0.9^{T-1} \times 0.1 \times 2^{2T} \right) \\
&= 0.1 \sum_{T=1}^{\infty} \left(0.9^{T-1} \times 2^T \right) \\
&= 0.2 \sum_{k=0}^{\infty} 1.8^k = \infty
\end{aligned}$$

endtoend.ai

Ordinary Importance Sampling: Unbounded Variance



endtoend.ai

Weighted Importance Sampling

- Has bias that converges asymptotically to zero
- Strongly preferred due to lower variance

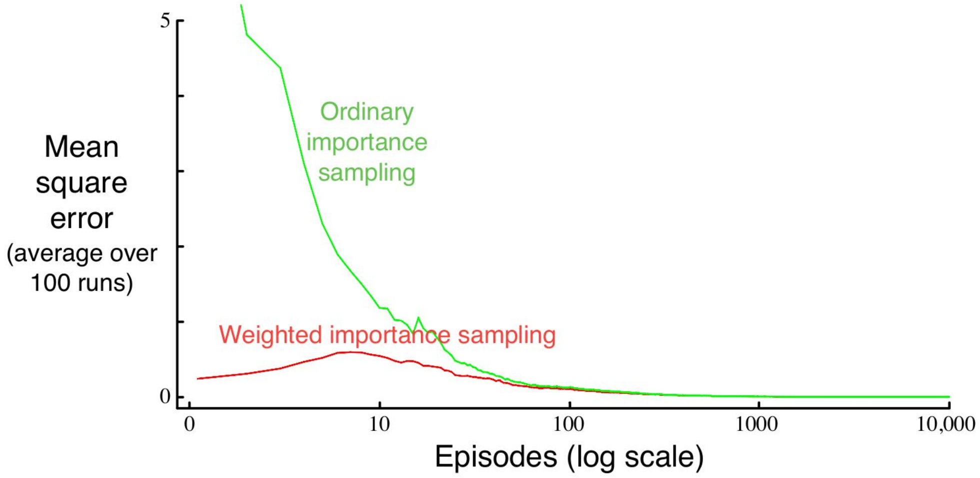
$$V(s) \doteq \frac{\sum_{t \in \mathfrak{T}(s)} \rho_{t:T(t)-1} G_t}{\sum_{t \in \mathfrak{T}(s)} \rho_{t:T(t)-1}},$$

- With single return:

$$V(s) = G$$

Blackjack example for Importance Sampling

- Evaluated for a single state
  - player's sum = 13, has usable ace, dealer's card = 2
  - Behavior policy: uniform random policy
  - Target policy: stick iff player's sum >= 20



endtoend.ai

Incremental Monte Carlo

- Update value without tracking all returns
- Ordinary importance sampling:

$$V_{n+1} = V_n + \frac{1}{n}[W_n G_n - V_n]$$

- Weighted importance sampling:

$$V_{n+1} = V_n + \frac{W_n}{C_n}[G_n - V_n] \text{ for } n \geq 1$$

$$C_{n+1} = C_n + W_{n+1}$$

endtoend.ai

Incremental Monte Carlo Pseudocode

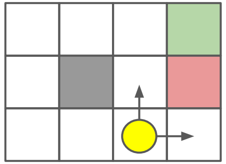
Off-policy MC prediction (policy evaluation) for estimating $Q \approx q_\pi$

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

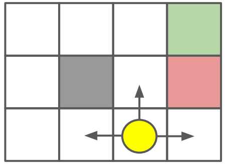
Loop forever (for each episode):
    $b \leftarrow$ any policy with coverage of π
    Generate an episode following b: $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$
    $G \leftarrow 0$
    $W \leftarrow 1$
    Loop for each step of episode, $t = T-1, T-2, \ldots, 0$:
        $G \leftarrow \gamma G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

Off-policy Monte Carlo Control

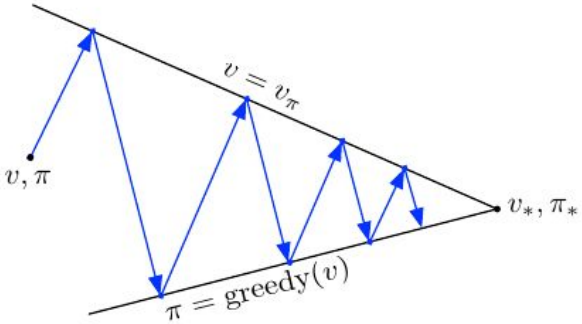
- *Off-policy*: target policy and behavior policy
- *Monte Carlo*: Learn from samples without bootstrapping
- *Control*: Find optimal policy through GPI



π



b



Off-policy Monte Carlo Control 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)
- $C(s,a) \leftarrow 0$
- $\pi(s) \leftarrow \operatorname{argmax}_a Q(s,a)$ (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, $t = T-1, T-2, \ldots, 0$:
  - $G \leftarrow \gamma G + R_{t+1}$
  - $C(S_t, A_t) \leftarrow C(S_t, A_t) + W$
  - $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \frac{W}{C(S_t, A_t)} [G - Q(S_t, A_t)]$
  - $\pi(S_t) \leftarrow \operatorname{argmax}_a Q(S_t, a)$ (with ties broken consistently)
  - If $A_t \neq \pi(S_t)$ then exit For loop
  - $W \leftarrow W \frac{1}{b(A_t|S_t)}$

endtoend.ai

Discounting-aware Importance Sampling: Intuition*

- Exploit return's internal structure to **reduce variance**
  - Return = Discounted sum of rewards
- Consider myopic discount $\gamma = 0$

$$\rho_{t:T-1} = \frac{\pi(A_0|S_0)}{b(A_0|S_0)} \frac{\pi(A_1|S_1)}{b(A_1|S_1)} \cdots \frac{\pi(A_{T-1}|S_{T-1})}{b(A_{T-1}|S_{T-1})}$$

Irrelevant to return: adds variance

Discounting as Partial Termination*

- Consider discount as *degree of partial termination*
  - If $\gamma = 0$, all episodes terminate after receiving first reward
  - If $0 \leq \gamma < 1$, episode could terminate after n steps with probability $(1-\gamma)\gamma^{h-1}$
  - Premature termination results in *partial returns*
- Full Return as *flat* (undiscounted) partial return $\bar{G}_{t:h} = R_{t+1} + \ldots + R_h$

$$
\begin{aligned}
G_t &= R_{t+1} + \ldots + \gamma^{T-t-1} R_T \\
&= (1-\gamma) \sum_{h=t+1}^{T-1} \gamma^{h-t-1} \bar{G}_{t:h} + \gamma^{T-t-1} \bar{G}_{t:T}
\end{aligned}
$$

endtoend.ai

Discounting-aware Ordinary Importance Sampling*

- Scale flat partial returns by a *truncated* importance sampling ratio
- Estimator for Ordinary importance sampling:

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

- Estimator for *Discounting-aware* ordinary importance sampling

$$V(s) \doteq \frac{\sum_{t \in \mathcal{T}(s)} \left((1-\gamma) \sum_{h=t+1}^{T(t)-1} \gamma^{h-t-1} \rho_{t:h-1} \bar{G}_{t:h} \; + \; \gamma^{T(t)-t-1} \rho_{t:T(t)-1} \bar{G}_{t:T(t)} \right)}{|\mathcal{T}(s)|},$$

endtoend.ai

Discounting-aware Weighted Importance Sampling*

- Scale flat partial returns by a *truncated* importance sampling ratio
- Estimator for Weighted importance sampling

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

- Estimator for *Discounting-aware* weighted importance sampling

$$V(s) \doteq \frac{\sum_{t \in \mathcal{T}(s)} \left((1-\gamma) \sum_{h=t+1}^{T(t)-1} \gamma^{h-t-1} \rho_{t:h-1} \bar{G}_{t:h} \; + \; \gamma^{T(t)-t-1} \rho_{t:T(t)-1} \bar{G}_{t:T(t)} \right)}{\sum_{t \in \mathcal{T}(s)} \left((1-\gamma) \sum_{h=t+1}^{T(t)-1} \gamma^{h-t-1} \rho_{t:h-1} \; + \; \gamma^{T(t)-t-1} \rho_{t:T(t)-1} \right)}.$$

Per-decision Importance Sampling: Intuition*

- Unroll returns as sum of rewards

$$\begin{aligned}\rho_{t:T-1}G_t &= \rho_{t:T-1}\left(R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{T-t-1}R_T\right) \\ &= \rho_{t:T-1}R_{t+1} + \gamma\rho_{t:T-1}R_{t+2} + \cdots + \gamma^{T-t-1}\rho_{t:T-1}R_T.\end{aligned}$$

- Can ignore trajectory after the reward since they are uncorrelated

$$\mathbb{E}\left[\frac{\pi(A_k \mid S_k)}{b(A_k \mid S_k)}\right] = \sum_a b(a \mid S_k)\frac{\pi(A_k \mid S_k)}{b(A_k \mid S_k)} = \sum_a \pi(a \mid S_k) = 1$$

Per-decision Importance Sampling: Process*

- Simplify expectation

$$\mathbb{E}_b[\rho_{t:T-1}R_{t+k}] = \mathbb{E}_b\left[\frac{\pi(A_t \mid S_t)}{b(A_t \mid S_t)}\frac{\pi(A_{t+1} \mid S_{t+1})}{b(A_{t+1} \mid S_{t+1})}\cdots\frac{\pi(A_{T-1} \mid S_{T-1})}{b(A_{T-1} \mid S_{T-1})}R_{t+1}\right]$$
$$= \mathbb{E}_b\left[\frac{\pi(A_t \mid S_t)}{b(A_t \mid S_t)}\cdots\frac{\pi(A_{t+k} \mid S_{t+k})}{b(A_{t+k} \mid S_{t+k})}R_{t+k}\right]\mathbb{E}_b\left[\frac{\pi(A_{t+k+1} \mid S_{t+k+1})}{b(A_{t+k+1} \mid S_{t+k+1})}\cdots\frac{\pi(A_{T-1} \mid S_{T-1})}{b(A_{T-1} \mid S_{T-1})}\right]$$
$$= \mathbb{E}_b\left[\frac{\pi(A_t \mid S_t)}{b(A_t \mid S_t)}\cdots\frac{\pi(A_{t+k} \mid S_{t+k})}{b(A_{t+k} \mid S_{t+k})}R_{t+k}\right]$$
$$= \mathbb{E}_b\left[\rho_{t:t+k-1}R_{t+k}\right]$$

- Equivalent expectation for return

$$\mathbb{E}\left[\rho_{t:T-1}G_t\right] = \mathbb{E}\left[\tilde{G}_t\right] = \mathbb{E}\left[\rho_{t:t}R_{t+1} + \gamma\rho_{t:t+1}R_{t+2} + \ldots + \gamma^{T-t-1}\rho_{t:T-1}R_T\right]$$

endtoend.ai

Per-decision Ordinary Importance Sampling*

- Estimator for Ordinary Importance Sampling:

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

- Estimator for Per-reward Ordinary Importance Sampling:

$$V(s) \doteq \frac{\sum_{t \in \mathcal{T}(s)} \tilde{G}_t}{|\mathcal{T}(s)|},$$

Per-decision Weighted Importance Sampling?*

- Unclear if per-reward *weighted* importance sampling is possible
- All proposed estimators are *inconsistent*
    - Do not converge asymptotically

Summary

- Learn from experience (sample episodes)
    - Learn directly from interaction without model
    - Can learn with simulation
    - Can focus to subset of states
    - No bootstrapping → less harmed by violation of Markov property
- Need to maintain exploration for Control
    - Exploring starts: unlikely in learning from real experience
    - On-policy: maintain exploration in policy
    - Off-policy: separate behavior and target policies
        - Importance Sampling
            - Ordinary importance sampling
            - Weighted importance sampling

Thank you!

Original content from

- Reinforcement Learning: An Introduction by Sutton and Barto

You can find more content in

- github.com/seungjaeryanlee
- www.endtoend.ai