Monte Carlo Methods Overview

- Learning from sample experience (simulation or sample models) requiring no prior knowledge of the environment's dynamics (as required for DP), yet can still attain optimal behaviour
- Based on averaging sample returns so only for episodic tasks (and not online step-by-step)
- Based on general policy iteration (GPI) to learn value functions from sample returns

Monte Carlo Prediction

- Prediction = learning state-value function $v_\pi(s)$ (expected return starting from state) for a given policy
- Estimate by averaging the returns observed after visits to that state: first visit and every-visit
- Converges as number of visits tends to infinity
- Estimates for each state are independent (do not bootstrap)
- Computational expense of estimating the value of a single state is independent of the number of states

Backup diagram →

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 $Returns(S_t)$

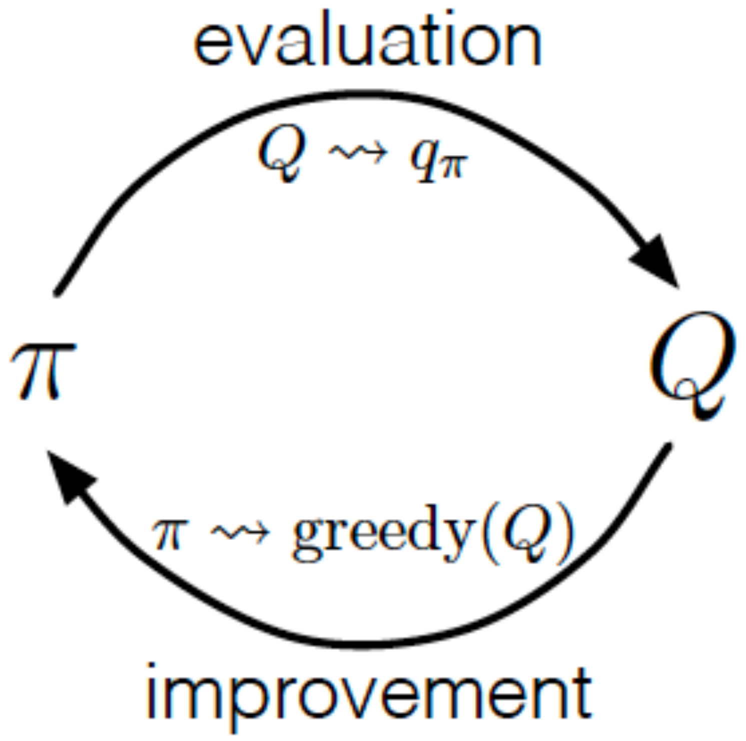
$V(S_t) \leftarrow \text{average}(Returns(S_t))$

Monte Carlo Estimation of Action Values

- Estimate action values $q_{\pi}(s, a)$: the expected return when starting in state s, taking action a, and thereafter following policy π.
- Exploring starts: to ensure convergence all state-action pairs must be evaluated so specifying that every pair has a nonzero probability of being selected as the start (works for simulation, not real experience)
- Not applicable for learning directly from interaction so must consider only policies that are stochastic with a nonzero probability of selecting all actions in each state

Monte Carlo Control

- GPI: value function is repeatedly altered to more closely approximate the value function for the current policy, and the policy is repeatedly improved with respect to the current value function
- Policy iteration: alternate between evaluation and improvement (acting greedy) on an episode-by-episode basis.
- First visit: averages the returns following the first time in each episode that the state was visited and the action was selected.



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

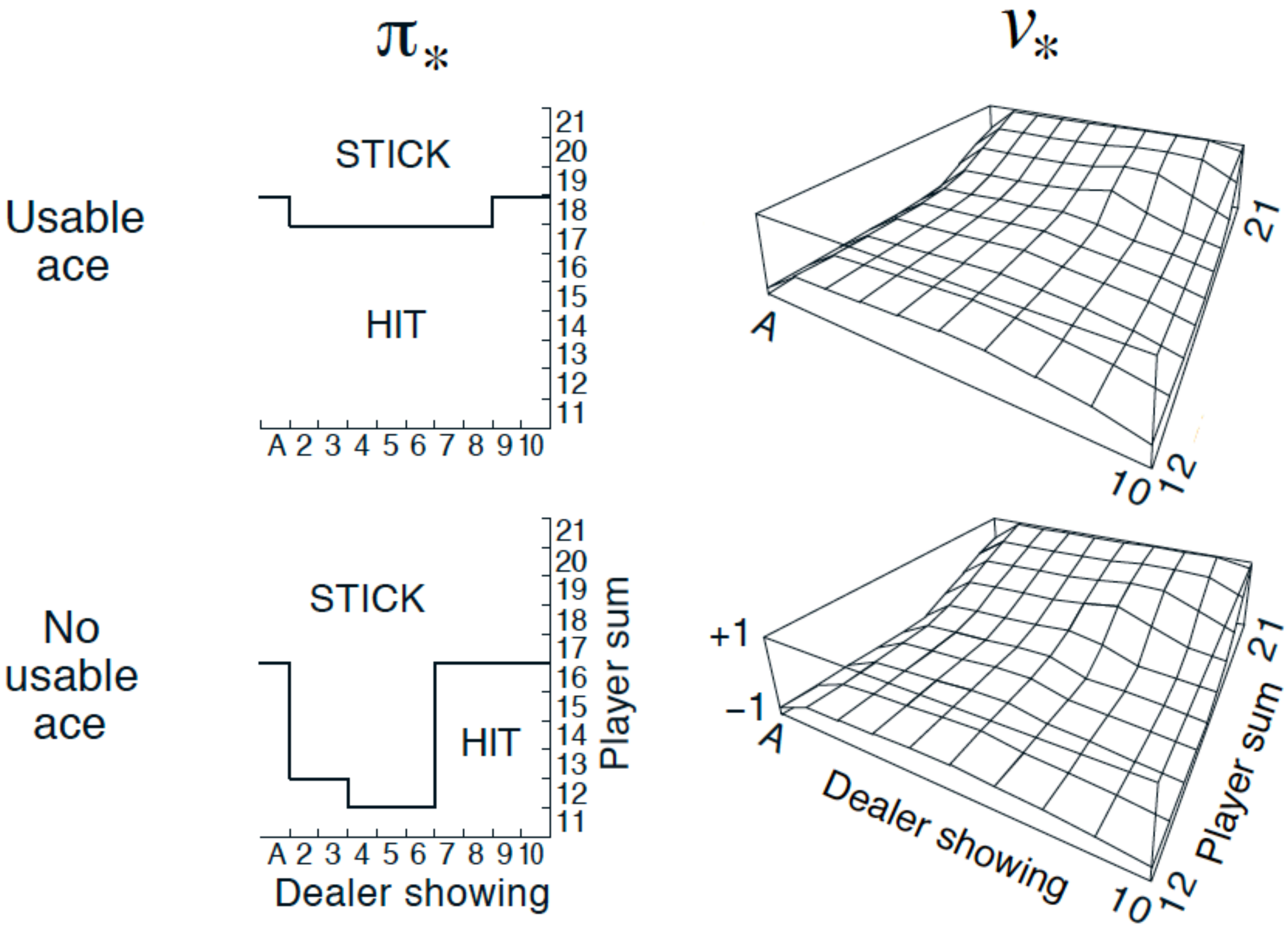
Append G to $Returns(S_t, A_t)$

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

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

Blackjack example - Monte Carlo ES

- The dealer follows a fixed strategy without choice: he sticks on any sum of 17 or greater, and hits otherwise.



Monte Carlo Control without Exploring Starts

- Soft policy: $\pi(a|s) > 0$ for all states and action but gradually shifted closer to a deterministic optimal policy
  - ε-greedy
- Policy improvement is guaranteed with convergence to the optimal policy among ε-soft policies (near-optimal policy)

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

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

Off-policy Prediction via Importance Sampling

- Off-policy: learn the value function of a target policy π from data generated from a behaviour policy b.
- Often greater variance and are slower to converge but allow learning from alternative data sources (e.g. human experts)
- Coverage: $\pi(a|s) > 0$ implies b$(a|s) > 0$
  - b has nonzero probability of selecting all actions that might be selected by π
- Importance sampling ratio ρ to weight returns relative to probability of trajectories occurring under the target and behaviour policies

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

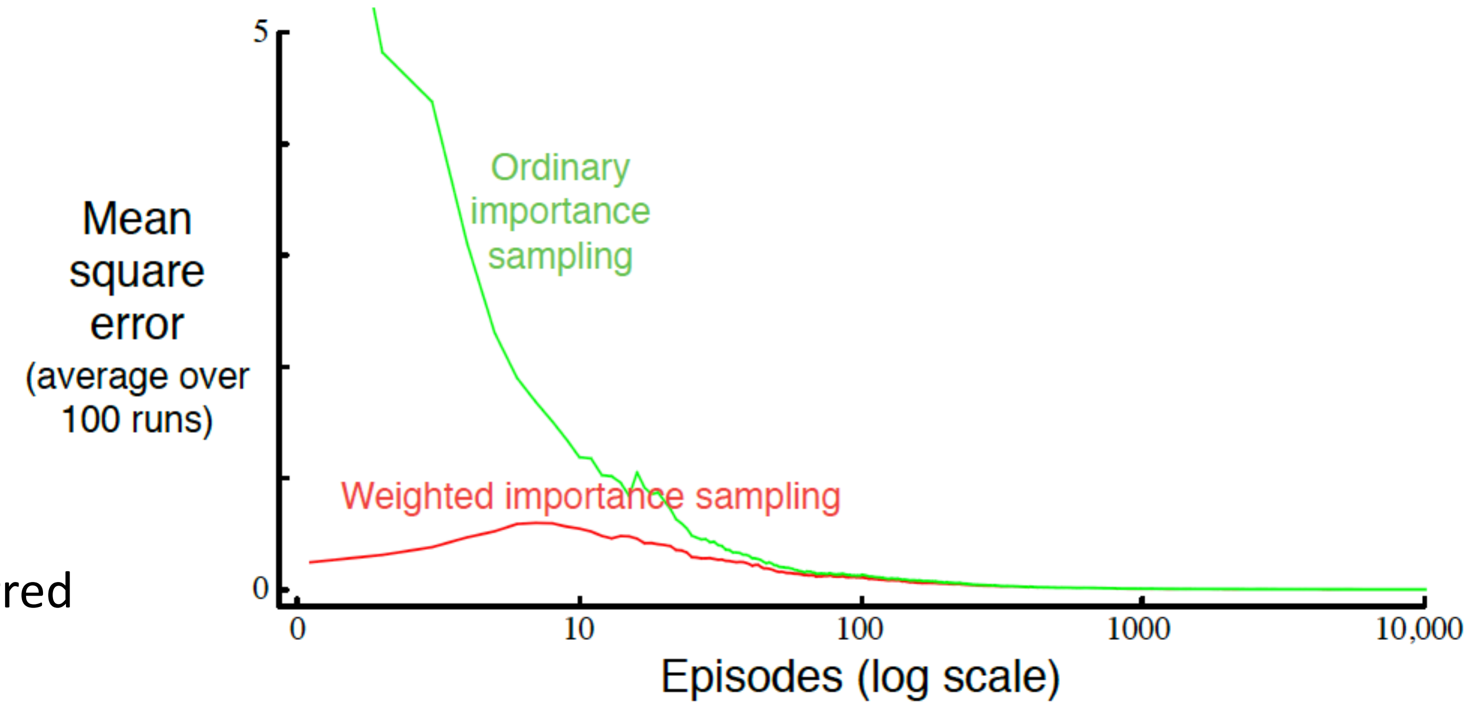
Off-policy Prediction via Importance Sampling

- Expected returns $\mathbb{E}[\rho_{t:T-1} G_t \mid S_t = s] = v_\pi(s).$
  - Ordinary important sampling
    - Average of weighted returns
    - Unbounded variance

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

  - Weighted importance sample
    - Weighted average
    - Bias but lower variance so preferred

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



Incremental implementation of weighted importance sampling

- Weighted average of the returns update rule:

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

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

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 $W \neq 0$:

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

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

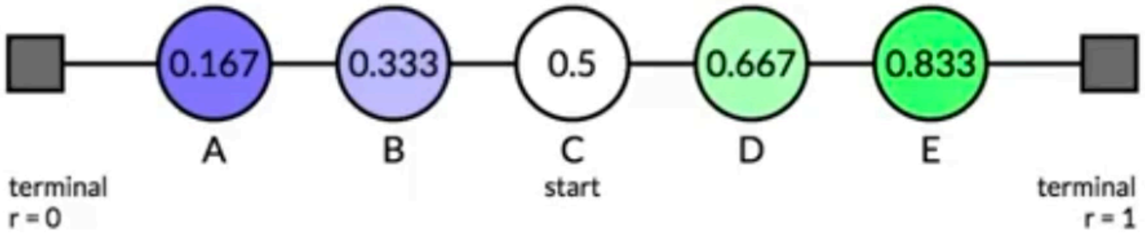
    $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \frac{W}{C(S_t, A_t)} [G - Q(S_t, A_t)]$

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

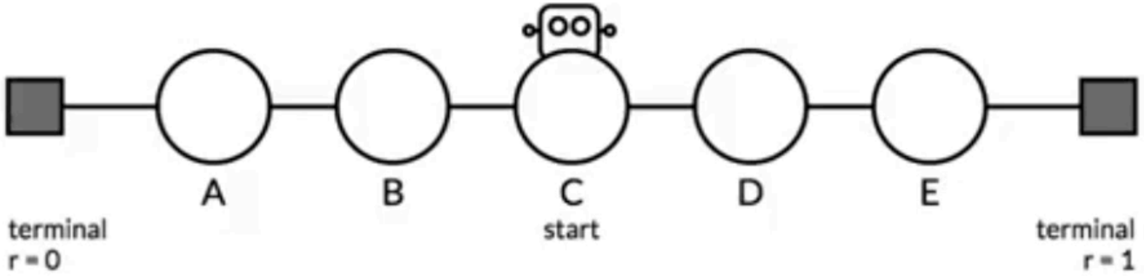
Off-policy Monte Carlo Control

- Target policy may be deterministic (e.g. greedy), while the behaviour policy can continue to sample all possible actions (e.g. ε-soft).
- Only learns from the tails of episodes when all of the remaining actions in the episode are greedy.
- Slow learning which is addressed by temporal difference (TD) learning

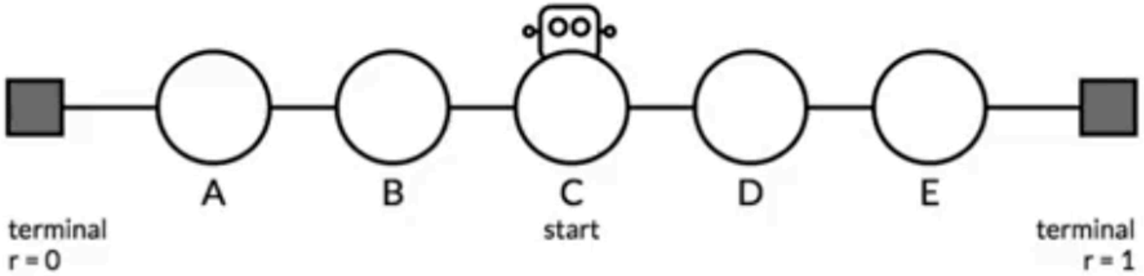
Target / Exact Values



Updates using TD Learning



Updates using Monte Carlo



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

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

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

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

$\quad \pi(s) \leftarrow \arg\max_a Q(s,a)$ (with ties broken consistently)

Loop forever (for each episode):

$\quad b \leftarrow$ any soft policy

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

$\quad G \leftarrow 0$

$\quad W \leftarrow 1$

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

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

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

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

$\quad\quad \pi(S_t) \leftarrow \arg\max_a Q(S_t, a)$ (with ties broken consistently)

$\quad\quad$ If $A_t \neq \pi(S_t)$ then exit inner Loop (proceed to next episode)

$\quad\quad W \leftarrow W \frac{1}{b(A_t|S_t)}$

Importance sampling extensions

- Discounting-aware Importance Sampling
- Per-decision Importance Sampling