Theory and Methods for Reinforcement Learning

Prof. Volkan Cevher
volkan.cevher@epfl.ch

Lecture 3: Monte Carlo Methods

Laboratory for Information and Inference Systems (LIONS)
École Polytechnique Fédérale de Lausanne (EPFL)

EE-618 (Spring 2020)

License Information for Reinforcement Learning Slides

- This work is released under a Creative Commons License with the following terms:
- **Attribution**
  - The licensor permits others to copy, distribute, display, and perform the work. In return, licensees must give the original authors credit.
- **Non-Commercial**
  - The licensor permits others to copy, distribute, display, and perform the work. In return, licensees may not use the work for commercial purposes – unless they get the licensor's permission.
- **Share Alike**
  - The licensor permits others to distribute derivative works only under a license identical to the one that governs the licensor's work.
- Full Text of the License

Outline

- This class:
  1. Monte Carlo Prediction
  2. Monte Carlo Control
- Next class:
  1. Temporal Difference Learning

Recommended reading

- Chapter 5 in S. Sutton, and G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, 2018.

Motivation

Motivation

In the previous lecture, we studied the planning problem. But if we do not have complete knowledge of the environment, what can we do? Learning!

Monte Carlo Methods for RL

• Monte Carlo methods are model-free: knowledge of the environment (transition dynamics) is not required.

• Monte Carlo methods require only experience – sample sequences of states, actions, and rewards from actual or simulated interaction with an environment.

• Monte Carlo methods are ways of solving the RL problem based on averaging sample returns.

• Here we consider only episodic tasks.

• Monte Carlo methods are incremental in an episode-by-episode sense, but not in a step-by-step (online) sense.

Monte Carlo Estimation

Definition

Given a function $f : \mathcal{X} \to \mathbb{R}$ and a distribution P over $\mathcal{X}$, define

$$\mu \;:=\; \mathop{\mathbb{E}}_{X \sim P}[f(X)] \;=\; \int_{\mathcal{X}} f(x)P(x)dx.$$

Then we sample $X_1, \cdots X_n \overset{iid}{\sim} P$, and define

$$\hat{\mu} \;:=\; \frac{1}{n}\sum_{i=1}^{n} f(X_i).$$

- law of large numbers: $\Pr\left(\lim_{n\to\infty} \hat{\mu} \to \mu\right) = 1.$
- central limit theorem: as $n \to \infty$, $\frac{\hat{\mu}-\mu}{\sigma/\sqrt{n}} \overset{d}{\sim} \mathcal{N}(0,1)$, where $\sigma^2 = \mathbb{V}[f(X)] < \infty$.

Monte Carlo Policy Evaluation (Prediction)

Monte Carlo Prediction

Given a set of episodes obtained by following π and passing through s:

$$S_1 \xrightarrow{A_1 \sim \pi} R_2, S_2 \cdots, S_t \xrightarrow{A_t \sim \pi} R_{t+1}, S_{t+1} \cdots$$

estimate the state-value function:

$$v_\pi(s) \;:=\; \mathbb{E}_\pi[G_t \mid S_t = s], \text{ where } G_t := \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k.$$

- Each occurrence of state s in an episode is called a *visit* to s.
- The first time state s is visited in an episode is called the *first visit* to s.

Monte Carlo Policy Evaluation (Prediction)

First-visit MC method

- estimates $v_\pi(s)$ as the average of the returns following first visits to s.
- converges to $v_\pi(s)$ as the number of first visits to s goes to infinity (by the law of large numbers).

Every-visit MC method

- estimates $v_\pi(s)$ as the average of the returns following all visits to s.
- converges to $v_\pi(s)$ as the number of visits to s goes to infinity [2].

First-visit MC Prediction Algorithm

First-visit MC prediction

Input: the policy π to be evaluated

Initialize:

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

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

Loop forever (for each episode):

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

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 $\text{Returns}(S_t)$

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

Output: $V \approx v_\pi$

Incremental Updates

- Given a sequence of $\{X_t\}_t$, define

$$\mu_n \;:=\; \frac{1}{n}\sum_{t=1}^{n} X_t.$$

- The incremental update rule for μ_n is:

$$\begin{aligned}
\mu_n &= \frac{1}{n}\sum_{t=1}^{n} X_t \\
&= \frac{1}{n}\left(X_n + (n-1)\frac{1}{n-1}\sum_{t=1}^{n-1} X_t\right) \\
&= \frac{1}{n}\left(X_n + (n-1)\mu_{n-1}\right) \\
&= \mu_{n-1} + \frac{1}{n}\left(X_n - \mu_{n-1}\right).
\end{aligned}$$

- Can generalize to weighted average.

Incremental Monte Carlo Updates

- Update V(s) incrementally, on an episode-by-episode basis.
- For each state S_t with return G_t, update counter $N(S_t)$ and $V(S_t)$:

$$\begin{aligned} N(S_t) &\leftarrow N(S_t) + 1 \\ V(S_t) &\leftarrow V(S_t) + \frac{1}{N(S_t)} \left(G_t - V(S_t)\right). \end{aligned}$$

- In non-stationary problems, it can be useful to track a running mean, i.e., forget old episodes.

$$V(S_t) \leftarrow V(S_t) + \alpha \left(G_t - V(S_t)\right).$$

Example: Blackjack

- Goal: obtain cards the sum of whose numerical values is as great as possible without exceeding 21.
- States (total of 200 states):
  - current sum (12–21)
  - the dealer's one showing card (ace–10)
  - whether or not holding a usable ace (yes-no)
- Actions:
  - stick (stop receiving cards and terminate)
  - hit (request another card, no replacement)
- Reward for stick:
  - $+1$ if current sum $>$ dealer's sum
  - 0 if current sum $=$ dealer's sum
  - -1 if current sum $<$ dealer's sum
- Reward for hit:
  - -1 if current sum > 21 (and terminate)
  - 0 otherwise

Example: Blackjack

- The dealer's strategy: stick on any sum of 17 or greater, and hit otherwise.
- The target policy π: stick if the current sum is 20 or 21, and hit otherwise.
- To find $v_\pi(s)$ by a Monte Carlo approach, one simulates many blackjack games using the policy π and averages the returns following each state.

Example: Blackjack

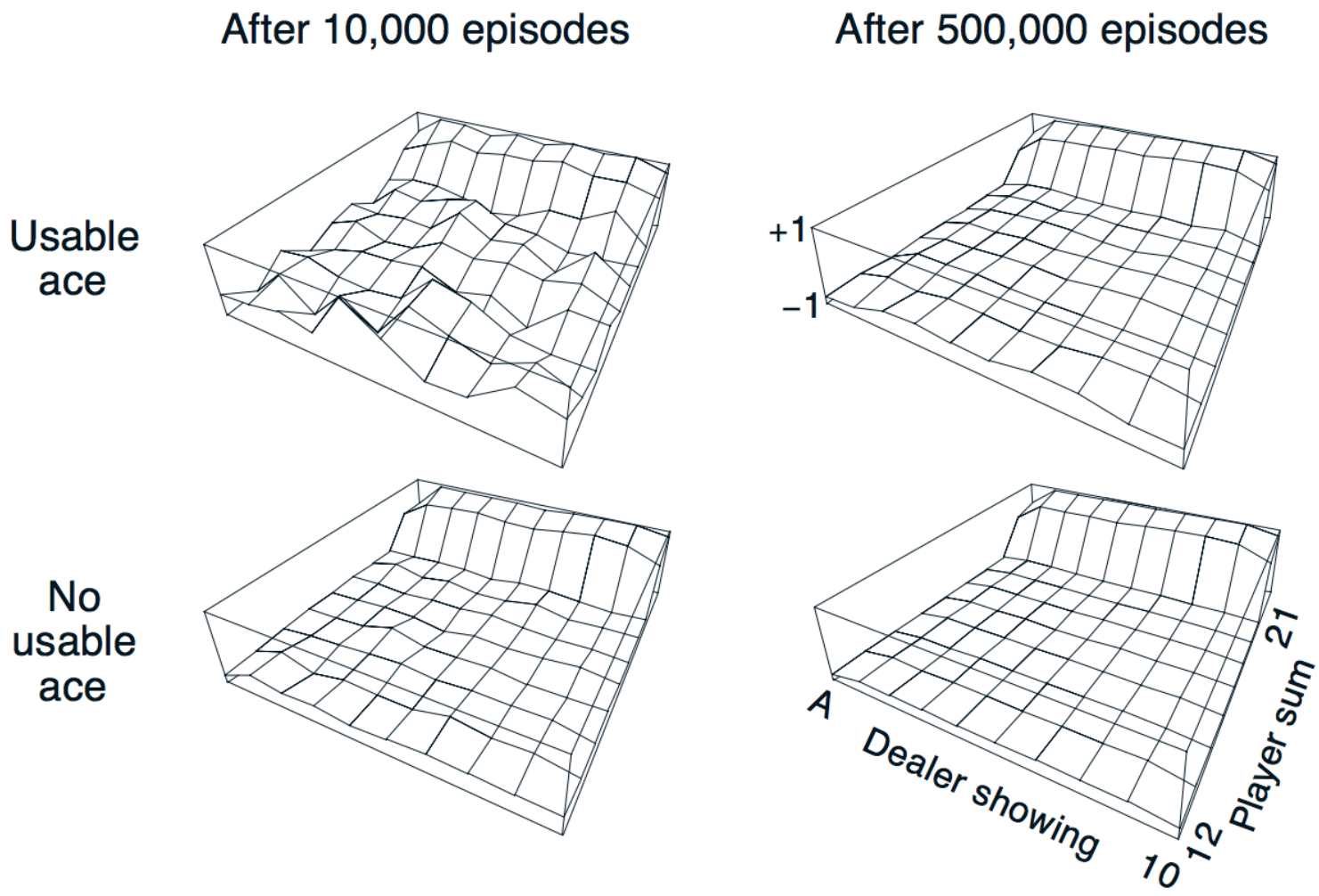


Figure: Approximate state-value functions $v_\pi(s)$ computed by Monte Carlo prediction.

Dynamic Programming vs. Monte Carlo

- DP methods:
  - update estimates of the values of states based on estimates of the values of successor states.
  - updating estimates on the basis of other estimates is called *bootstrapping*.
- MC methods:
  - the estimates for each state are independent.
  - the estimate for one state does not build upon the estimate of any other state, i.e., MC methods do not bootstrap.
  - the computational expense of estimating the value of a single state is independent of the number of states.

Monte Carlo Estimation of Action Values

- Without a model, state values alone are not sufficient to determine (optimal) policy.
- Thus, in MC methods we estimate action values $q_\pi(s,a) = \mathbb{E}_\pi[G_t | S_t = s, A_t = a]$.
- MC methods for state value estimation can be extended to action value estimation.
- Maintaining exploration problem: many state-action pairs may never be visited (e.g. consider a deterministic policy). For policy evaluation to work for action values, we must assure continual exploration.

Exploring starts assumption

- every state-action pair has a nonzero probability of being selected as the start of an episode.
- this guarantees that all state-action pairs will be visited an infinite number of times in the limit of an infinite number of episodes.

- Alternative option: consider only policies that are stochastic with a nonzero probability of selecting all actions in each state.

Generalized Policy Iteration

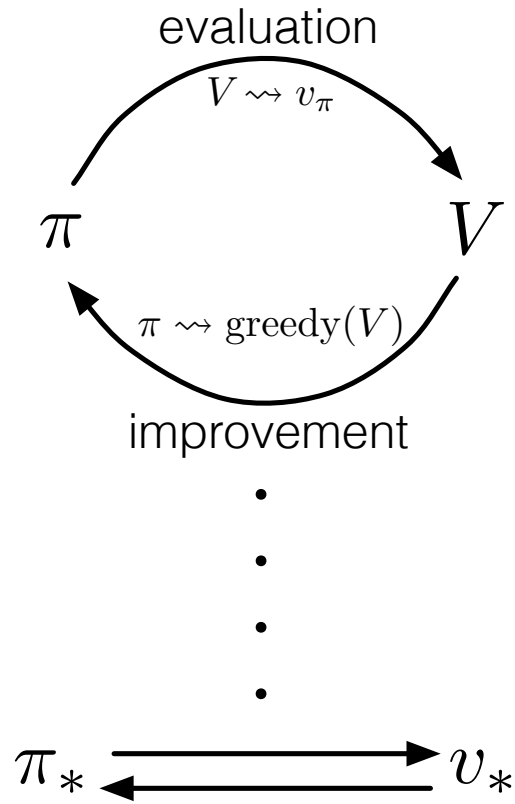
• **Policy evaluation**: estimate v_π
e.g., iterative policy iteration:

$$V(s) = \sum_{s',r} p(s', r|s, \pi(s)) \Big(r + \gamma V(s') \Big),$$

until convergence.

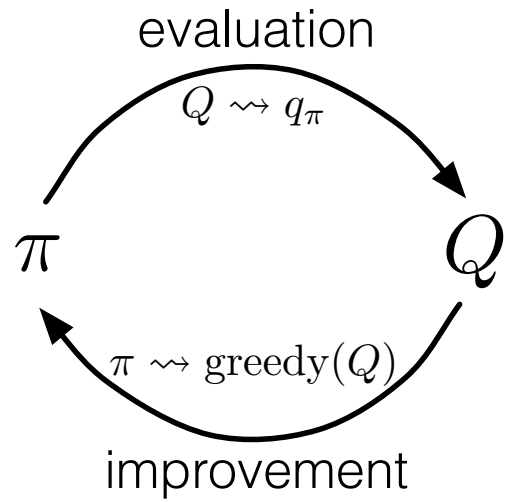
• **Policy improvement**: generate $\pi' \geq \pi$
e.g., greedy policy improvement:

$$\pi(s) \leftarrow \arg\max_a \sum_{s',r} p(s', r|s, a) \Big(r + \gamma V(s') \Big)$$



Greedy policy improvement requires the knowledge of MDP.

Monte Carlo Policy Iteration (Control)



• Alternating complete steps of policy evaluation and policy improvement.

• Beginning with an arbitrary policy π_0 and ending with the optimal policy and optimal action-value function:

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

- $\xrightarrow{E}$: complete MC policy evaluation
- $\xrightarrow{I}$: complete greedy policy improvement

Monte Carlo Policy Iteration (Control)

Policy evaluation

- Compute each q_{π_k} exactly, for arbitrary π_k using MC methods.
- Assumptions for exact policy evaluation:
  1. observe an infinite number of episodes.
  2. the episodes are generated with exploring starts.

Policy improvement

- Construct each π_{k+1} as the greedy policy with respect to the current action-value function q_{π_k}:

$$\pi_{k+1}(s) \;:=\; \arg\max_a q_{\pi_k}(s,a).$$

- No model is needed to construct the greedy policy.

Monte Carlo Policy Iteration (Control)

Theorem (Convergence for the MC Policy Iteration)

Under the two assumptions for exact policy evaluation, overall process (MC Policy Iteration) converges to the optimal policy and optimal value function.

Proof.

Suppose we observe an infinite number of episodes (with exploring starts), and thus q_{π_k} is computed exactly:

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

Therefore by the policy improvement theorem: $\pi_{k+1} \geq \pi_k$. This in turn assures us that the overall process converges to the optimal policy and optimal value function. □

Motivation

Key question

How can we remove the two unlikely assumptions in order to guarantee the convergence of the MC policy iteration?

MC Control w/o infinite number of episodes

• Give up trying to complete policy evaluation before returning to policy improvement.

• On each evaluation step, move the value function toward q_{π_k}, but do not expect to actually get close.

• Alternate between evaluation and improvement on an episode-by-episode basis.

• After each episode, the observed returns are used for policy evaluation, and then the policy is improved at all the states visited in the episode.

MC Control with Exploring Starts Algorithm

MC Control with Exploring Starts (ES)

Initialize:

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

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

$\text{Returns}(s,a) \leftarrow$ an empty list, $\forall s \in \mathcal{S}$

Loop forever (for each episode):

Choose $S_0 \in \mathcal{S}, A_0 \in \mathcal{A}(S_0)$ randomly s.t. 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 pair S_t, A_t appears in $S_0, A_0, \ldots, S_{t-1}, A_{t-1}$:

Append G to $\text{Returns}(S_t, A_t)$

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

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

Output: $\pi \approx \pi^*$

MC Control with Exploring Starts Algorithm

• In MC Control with 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.

Example: Blackjack

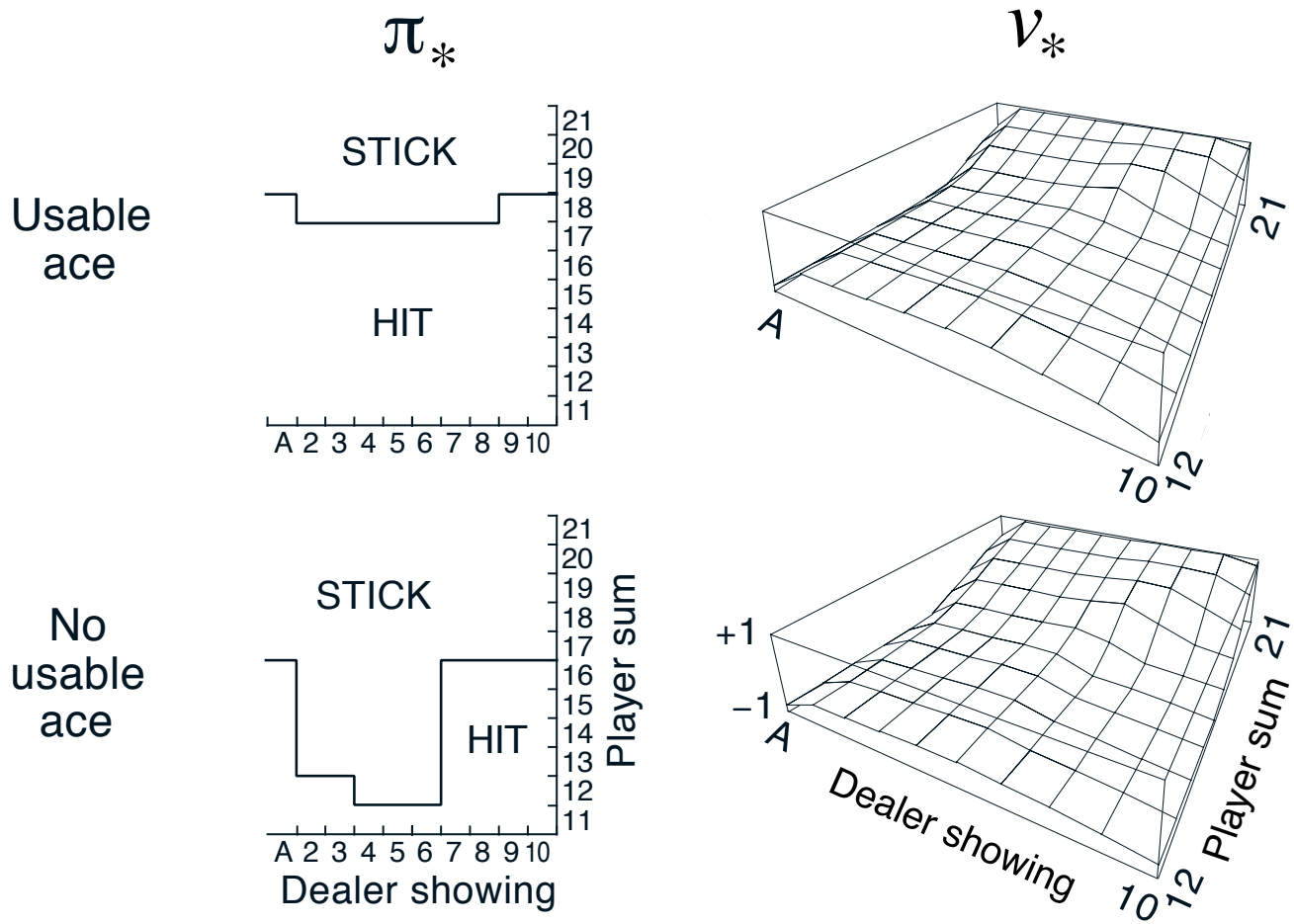


Figure: The optimal policy and state-value function for blackjack, found by Monte Carlo ES.

MC Control w/o Exploring Starts

• The only general way to ensure that all actions are selected infinitely often is for the agent to continue to select them.

On-policy vs. Off-policy

- on-policy methods: attempt to evaluate or improve the policy that is used to make decisions.
- off-policy methods: evaluate or improve a policy different from that used to generate the data.

On-policy MC Control w/o Exploring Starts

Definition (Soft policies)

- ϵ-soft policy:

$$\pi(a \mid s) \geq \epsilon, \quad \forall s \in \mathcal{S}, a \in \mathcal{A}(s).$$

- ϵ-greedy policy:

$$\pi'(a|s) \leftarrow \begin{cases} 1 - \epsilon + \frac{\epsilon}{|\mathcal{A}(s)|}, & \text{if } a \in \arg\max_a q_\pi(s,a) \\ \frac{\epsilon}{|\mathcal{A}(s)|}, & \text{otherwise} \end{cases}$$

- the ϵ-greedy policies are examples of ϵ-soft policies.
- among ϵ-soft policies, ϵ-greedy policies are in some sense those that are closest to greedy policy.

• The overall idea of on-policy MC control is still that of GPI.

• GPI does not require that the policy be taken all the way to a greedy policy, only that it be moved towards a greedy policy.

• We will move it only to an ϵ-greedy policy.

On-policy MC Control w/o Exploring Starts

Theorem (Soft policy improvement)

For any ϵ-soft policy π, any ϵ-greedy policy π' with respect to q_π is guaranteed to be better than or equal to π, i.e. $\pi' \geq \pi$.

Proof.

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

Applying the policy improvement theorem, one has $\pi' \geq \pi$. The equality can hold only when both π and π' are optimal among the ϵ-soft policies (see book for the proof). □

On-policy MC Control w/o Exploring Starts

On-policy first-visit MC control (for ϵ-soft policies)

Algorithm parameter: small $\epsilon > 0$

Initialize:

$\pi(s) \leftarrow$ an arbitrary ϵ-soft policy

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

$\text{Returns}(s,a) \leftarrow$ an empty list, $\forall s \in \mathcal{S}$

Loop 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 pair S_t, A_t appears in $S_0, A_0, \ldots, S_{t-1}, A_{t-1}$:

Append G to $\text{Returns}(S_t, A_t)$

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

Let $A^* \leftarrow \arg\max_a Q(S_t, a)$, and for all $a \in \mathcal{A}(S_t)$:

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

Output: $\pi \approx \pi^*$

Off-policy Methods

- Use two different policies:
  1. target policy - one that is learned about and that becomes the optimal policy.
  2. behavior policy - one that is more exploratory and is used to generate behavior.

- Sample efficient compared to on-policy methods.

- They have greater variance and are slower to converge (as the data is due to a different policy).

- They are more powerful and general, e.g., on-policy method is a special case in which the target and behavior policies are the same.

Off-policy MC Prediction

Off-policy Prediction

- estimate v_π or q_π, given episodes following another policy b, where $b \neq \pi$:

$$S_1 \xrightarrow{A_1 \sim b} R_2, S_2 \cdots, S_t \xrightarrow{A_t \sim b} R_{t+1}, S_{t+1} \cdots$$

- π is the target policy
- b is the behavior policy

Assumption of coverage

$$\pi(a|s) > 0 \Rightarrow b(a|s) > 0, \quad \forall a \in \mathcal{A}(s)$$

- it follows from coverage that b must be stochastic in states where it is not identical to π.
- target policy π, on the other hand, may be deterministic.

Importance Sampling (IS)

Importance sampling

Given a function $f : \mathcal{X} \to \mathbb{R}$ and distributions P, Q over $\mathcal{X}$, we have:

$$\begin{aligned}\mu \;:=\; \mathop{\mathbb{E}}_{X\sim P}[f(X)] \;&=\; \int_{\mathcal{X}} f(x)P(x)dx \\ &=\; \int_{\mathcal{X}} f(x)\frac{P(x)}{Q(x)}Q(x)dx \\ &=\; \mathop{\mathbb{E}}_{X\sim Q}\left[f(X)\frac{P(X)}{Q(X)}\right]\end{aligned}$$

Given $X_1, \cdots, X_n \overset{iid}{\sim} Q$, we estimate $\mathbb{E}_{X\sim P}[f(X)]$ by

$$\hat{\mu}_{\mathrm{IS}} \;:=\; \frac{1}{n}\sum_{i=1}^{n}\frac{f(X_i)P(X_i)}{Q(X_i)}$$

Importance Sampling (IS) Ratio

Probability of the trajectory

Given a state S_t, the probability of a subsequent state-action trajectory $A_t, S_{t+1}, A_{t+1}, ..., S_T$ occurring under policy π is

$$\Pr\{A_t, S_{t+1}, A_{t+1}, \cdots, S_T | S_t, A_{t:T-1} \sim \pi\} = \prod_{k=t}^{T-1} \pi(A_k|S_k) p(S_{k+1}|S_k, A_k).$$

Importance sampling ratio

The relative probability of the trajectory under the target and behavior policies:

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

- Returns G_t are due to the behavior policy:

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

Off-policy MC Prediction via Importance Sampling

- t: increases across episode boundaries

- $\mathcal{T}(s)$: the set of all time steps in which state s is visited (for an every-visit method); or time steps that were first visits to s within their episodes (for a first-visit method).

- $T(t)$: the first time of termination following time t.

- G_t: the return after t up through $T(t)$.

- $\{G_t\}_{t\in\mathcal{T}(s)}$: the returns that pertain to state s.

- $\{\rho_{t:T(t)-1}\}_{t\in\mathcal{T}(s)}$: the corresponding importance-sampling ratios.

Off-policy MC Prediction via Importance Sampling

Ordinary Importance Sampling (OIS) estimator for $v_\pi(s)$

- scale the returns by the ratios and average the results:

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

- unbiased (first-visit MC) estimator of $v_\pi(s)$
- the variance of OIS is in general unbounded because the variance of the ratios can be unbounded (see Example 5.5 from the book).

Off-policy MC Prediction via Importance Sampling

Weighted Importance Sampling (WIS) estimator for $v_\pi(s)$

- uses a weighted average:

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

  or zero if the denominator is zero.

- biased estimator (though the bias converges asymptotically to zero).
- in WIS the largest weight on any single return is one.
- assuming bounded returns, the variance of the WIS estimator converges to zero even if the variance of the ratios themselves is infinite [1].

Incremental Implementation of MC Methods

• For on-policy methods, and off-policy methods with OIS: incremental implementation is straight forward.

• For off-policy methods with WIS: given a sequence of $\{G_k\}_k$ and a corresponding weight sequence $\{W_k\}_k$, define

$$V_n \;:=\; \frac{\sum_{k=1}^{n-1} W_k G_k}{\sum_{k=1}^{n-1} W_k}, \quad n \geq 2$$

• The incremental update rule for V_n is:

$$\begin{aligned} V_{n+1} &= V_n + \frac{W_n}{C_n}\left[G_n - V_n\right], \quad n \geq 1 \\ C_{n+1} &= C_n + W_{n+1} \\ C_0 &= 0 \\ V_1 &\leftarrow \text{arbitrary} \end{aligned}$$

Off-policy MC Prediction via WIS

Off-policy (every-visit) MC prediction via WIS

Input: an arbitrary target policy π

Initialize:

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

$C(s,a) \leftarrow 0$, $\forall s \in \mathcal{S}, a \in \mathcal{A}(s)$

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 $W \neq 0$:

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

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

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

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

Output: $Q \approx q_\pi$

Example: Blackjack

- Off-policy estimation of a Blackjack state value:
  - state: current sum =13, dealer is showing a deuce; usable ace = yes
  - the behavior policy: choosing to hit or stick at random with equal probability.
  - the target policy: stick only on a sum of 20 or 21 (same as before)
  - the value of this state under the target policy is ≈ -0.27726

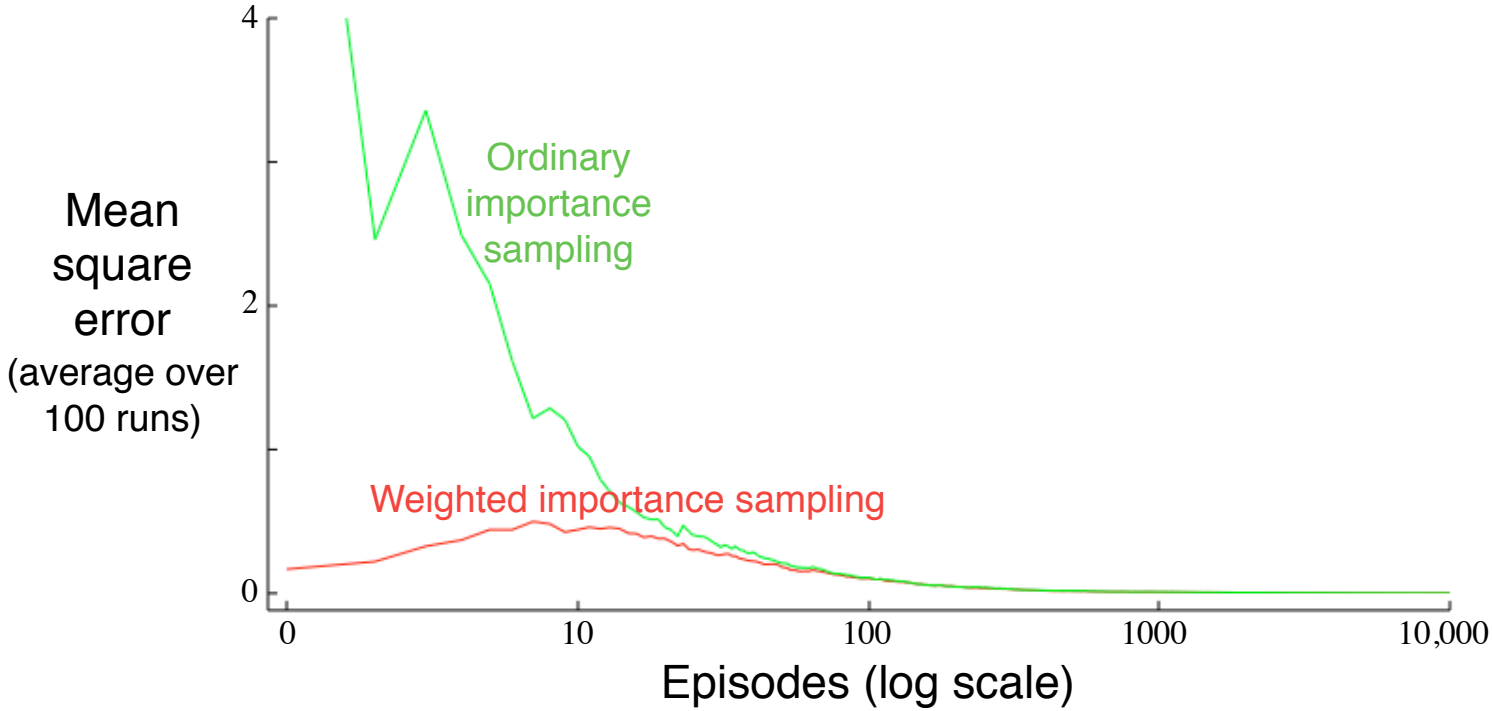


Figure: WIS produces lower error estimates of the value of a single blackjack state from off-policy episodes.

Off-policy MC Control

- Off-policy MC control method is based on GPI and WIS, for estimating π_* and q_*
- Target policy $\pi \approx \pi_*$ is the greedy policy w.r.t. Q, which is an estimate of q_π
- Behavior policy b is ϵ-soft

Off-policy MC Control

Off-policy (every-visit) MC Control

Initialize:

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

$C(s,a) \leftarrow 0$, $\forall s \in \mathcal{S}, a \in \mathcal{A}(s)$

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

Loop forever (for each episode):

$b \leftarrow$ any soft policy

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

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

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

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

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

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

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

Output: $\pi \approx \pi^*$

References

[1] Doina Precup, Richard S Sutton, and Sanjoy Dasgupta.
Off-policy temporal-difference learning with function approximation.
In *ICML*, pages 417–424, 2001.

[2] Satinder P Singh and Richard S Sutton.
Reinforcement learning with replacing eligibility traces.
Machine learning, 22(1-3):123–158, 1996.