Chapter 5: Monte Carlo Methods

Seungjae Ryan Lee
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: $\mathcal{V}_\pi$

- Estimate $\mathcal{V}_\pi$ from sample return
- Converges as more returns are observed

\[
V(s) = \frac{10 \times 1 + 2 \times 0 + 3 \times 0}{10 + 2} = \frac{10}{12} \approx 0.857
\]
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: s1 → s2 → s3 → s2 → s3
First-visit MC prediction in Practice: $\mathcal{V}_\pi$

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

Input: a policy $\pi$ to be evaluated

Initialize:

$V(s) \in \mathbb{R}$, arbitrarily, for all $s \in S$

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

Loop forever (for each episode):

- Generate an episode following $\pi$: $S_0, A_0, R_1, S_1, A_1, R_2, \ldots, S_{T-1}, A_{T-1}, R_T$
- $G \leftarrow 0$
- 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 $\text{Returns}(S_t)$
    - $V(S_t) \leftarrow \text{average}(\text{Returns}(S_t))$
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
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
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](http://www-anw.cs.umass.edu/~barto/courses/cs687/Chapter%205.pdf)
Monte Carlo Prediction: $q_\pi$

- 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
Exploring Starts (ES)

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

\[(s_0, a_0)\]
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

\[ \pi_0 \rightarrow q_{\pi_0} \rightarrow \pi_1 \rightarrow q_{\pi_1} \rightarrow \ldots \rightarrow \pi^* \rightarrow q^* \]
Monte Carlo ES Pseudocode

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

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

Loop forever (for each episode):

- Choose $S_0 \in S$, $A_0 \in \mathcal{A}(S_0)$ randomly such that all pairs have probability > 0
- Generate an episode from $S_0, A_0$, following $\pi$: $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$
- $G \leftarrow 0$
- 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 $\text{Returns}(S_t, A_t)$
    - $Q(S_t, A_t) \leftarrow \text{average}(\text{Returns}(S_t, A_t))$
    - $\pi(S_t) \leftarrow \text{argmax}_a Q(S_t, a)$
Blackjack Example Revisited

- Prediction → Control
\( \varepsilon \)-soft Policy

- Avoid exploring starts \( \rightarrow \) Add exploration to policy
- **Soft** policy: every action has nonzero probability of being selected
  \[
  \pi(a \mid s) > 0
  \]
- **\( \varepsilon \)-soft** policy: every action has at least \( \varepsilon / |A(s)| \) probability of being selected
  \[
  \pi(a \mid s) \geq \frac{\varepsilon}{|A(s)|}
  \]
- ex) **\( \varepsilon \)-greedy policy**
  - Select greedily for \( 1 - \varepsilon \) probability
  - Select randomly for \( \varepsilon \) probability (including greedy)
\( \varepsilon \)-soft vs \( \varepsilon \)-greedy

- Random \( \varepsilon = 1 \)
- \( \varepsilon \)-soft \( \varepsilon = 0.1 \)
- \( \varepsilon \)-greedy \( \varepsilon = 0.1 \)
- Greedy \( \varepsilon = 0 \)
### On-policy $\varepsilon$-soft MC control Pseudocode

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

<table>
<thead>
<tr>
<th>On-policy first-visit MC control (for $\varepsilon$-soft policies), estimates $\pi \approx \pi^*$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithm parameter: small $\varepsilon &gt; 0$</td>
</tr>
<tr>
<td>Initialize:</td>
</tr>
<tr>
<td>$\pi \leftarrow$ an arbitrary $\varepsilon$-soft policy</td>
</tr>
<tr>
<td>$Q(s, a) \in \mathbb{R}$ (arbitrarily), for all $s \in S$, $a \in A(s)$</td>
</tr>
<tr>
<td>$\text{Returns}(s, a) \leftarrow$ empty list, for all $s \in S$, $a \in A(s)$</td>
</tr>
<tr>
<td>Repeat forever (for each episode):</td>
</tr>
<tr>
<td>Generate an episode following $\pi$: $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$</td>
</tr>
<tr>
<td>$G \leftarrow 0$</td>
</tr>
<tr>
<td>Loop for each step of episode, $t = T-1, T-2, \ldots, 0$:</td>
</tr>
<tr>
<td>$G \leftarrow \gamma G + R_{t+1}$</td>
</tr>
<tr>
<td>Unless the pair $S_t, A_t$ appears in $S_0, A_0, S_1, A_1 \ldots, S_{t-1}, A_{t-1}$:</td>
</tr>
<tr>
<td>Append $G$ to $\text{Returns}(S_t, A_t)$</td>
</tr>
<tr>
<td>$Q(S_t, A_t) \leftarrow \text{average}(\text{Returns}(S_t, A_t))$</td>
</tr>
<tr>
<td>$A^* \leftarrow \text{argmax}_a Q(S_t, a)$ (with ties broken arbitrarily)</td>
</tr>
<tr>
<td>For all $a \in A(S_t)$:</td>
</tr>
<tr>
<td>$\pi(a</td>
</tr>
</tbody>
</table>
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 $\pi$: 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 $\pi$, all possible actions of $\pi$ must be taken by $b$

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

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

\[
\rho_{t:T-1} = \frac{\prod_{k=t}^{T-1} \pi(A_k | S_k)p(S_{k+1} | S_k, A_k)}{\prod_{k=t}^{T-1} b(A_k | S_k)p(S_{k+1} | S_k, A_k)} = \prod_{k=t}^{T-1} \frac{\pi(A_k | S_k)}{b(A_k | S_k)}. 
\]
Ordinary Importance Sampling

- Zero bias but **unbounded variance**

\[ V(s) = \frac{\sum_{t \in I(s)} \rho_{t:T(t)-1} G_t}{|I(s)|} \]

- With single return:

\[ V(s) = \rho_{t:T(t)-1} G \]
Ordinary Importance Sampling: Zero Bias

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

\[ = \sum_{k=t}^{T-1} \prod_{k=t}^{T-1} \pi(A_k \mid S_k) p(S_{k+1} \mid S_k, A_k) G_t \]

\[ = \rho_{t:T-1} \sum_{k=t}^{T-1} \prod_{k=t}^{T-1} b(A_k \mid S_k) p(S_{k+1} \mid S_k, A_k) G_t \]

\[ = \rho_{t:T-1} \mathbb{E}_b [G_t \mid S_t = s] \]
Ordinary Importance Sampling: Unbounded Variance

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

Variance of an estimator:

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

\[ \pi(\text{left}|s) = 1 \]
\[ b(\text{left}|s) = \frac{1}{2} \]
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 $2^T$

$$
\begin{align*}
E_b[(\rho G_0)^2] &= E_b[\rho^2] \\
&= \sum_{T=1}^{\infty} \left( p_{\text{trajectory}} \rho^2 \right) \\
&= \sum_{T=1}^{\infty} \left( b(\text{left} | s)^T p(s | s, \text{left})^{T-1} p(t | s, \text{left}) \rho^2 \right) \\
&= \sum_{T=1}^{\infty} \left( \frac{1}{2^T} \times 0.9^{T-1} \times 0.1 \times 2^{2T} \right) \\
&= 0.1 \sum_{T=1}^{\infty} \left( 0.9^{T-1} \times 2^T \right) \\
&= 0.2 \sum_{k=0}^{\infty} 1.8^k = \infty
\end{align*}
$$
Ordinary Importance Sampling: Unbounded Variance

Monte-Carlo estimate of $v_\pi(s)$ with ordinary importance sampling (ten runs)
Weighted Importance Sampling

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

\[ V(s) = \frac{\sum_{t \in T(s)} \rho_{t:T(t)-1} G_t}{\sum_{t \in T(s)} \rho_{t:T(t)-1}} \]

- 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

![Graph showing mean square error over episodes for ordinary and weighted importance sampling.](chart.png)
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:
  \[ V_{n+1} = V_n + \frac{W_n}{C_n} [G_n - V_n] \text{ for } n \geq 1 \]
  \[ C_{n+1} = C_n + W_{n+1} \]
# Incremental Monte Carlo Pseudocode

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

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

Loop forever (for each episode):
- $b \leftarrow$ any policy with coverage of $\pi$
- 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, $t = T-1, T-2, \ldots, 0$:
- $G \leftarrow \gamma G + R_{t+1}$
- $C(S_t, A_t) \leftarrow C(S_t, A_t) + W$
- $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \frac{W}{C(S_t, A_t)} \left[ G - Q(S_t, A_t) \right]$
- $W \leftarrow W \pi(A_t | S_t) \frac{b(A_t | S_t)}{\pi(b | S_t)}$
- 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

\[ \pi \quad \quad b \]
Off-policy Monte Carlo Control Pseudocode

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

Initialize, for all $s \in S$, $a \in A(s)$:
- $Q(s, a) \in \mathbb{R}$ (arbitrarily)
- $C(s, a) \leftarrow 0$
- $\pi(s) \leftarrow \text{argmax}_a Q(s, a)$ (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, $t = T-1, T-2, \ldots, 0$:
- $G \leftarrow \gamma G + R_{t+1}$
- $C(S_t, A_t) \leftarrow C(S_t, A_t) + W$
- $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \frac{W}{\bar{C}(S_t, A_t)} [G - Q(S_t, A_t)]$
- $\pi(S_t) \leftarrow \text{argmax}_a Q(S_t, a)$ (with ties broken consistently)

If $A_t \neq \pi(S_t)$ then exit For loop
- $W \leftarrow W \frac{1}{b(A_t|S_t)}$
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 $(1 - \gamma)^{h-1}$
  - Premature termination results in *partial returns*
- Full Return as *flat* (undiscounted) partial return $\bar{G}_{t:h} = R_{t+1} + \ldots + R_h$

$$G_t = R_{t+1} + \ldots + \gamma^{T-t-1} R_T$$

$$= (1 - \gamma) \sum_{h=t+1}^{T-1} \gamma^{h-t-1} \bar{G}_{t:h} + \gamma^{T-t-1} \bar{G}_{t:T}$$
Discounting-aware Ordinary Importance Sampling*

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

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

- Estimator for *Discounting-aware* ordinary importance sampling

\[
V(s) = \sum_{t \in \mathcal{T}(s)} \left( (1 - \gamma) \sum_{h=t+1}^{T(t)-1} \gamma^{h-t-1} \rho_{t:h-1} \bar{G}_{t:h} + \gamma^{T(t)-t-1} \rho_{t:T(t)-1} \bar{G}_{t:T(t)} \right) \left/ |\mathcal{T}(s)| \right.,
\]
Discounting-aware Weighted Importance Sampling*

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

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

- Estimator for *Discounting-aware* weighted importance sampling

\[
V(s) = \frac{\sum_{t \in \mathcal{T}(s)} \left[ (1 - \gamma) \sum_{h=t+1}^{T(t)-1} \gamma^{h-t-1} \rho_{t:h-1} \tilde{G}_{t:h} + \gamma^{T(t)-t-1} \rho_{t:T(t)-1} \tilde{G}_{t:T(t)} \right]}{\sum_{t \in \mathcal{T}(s)} \left[ (1 - \gamma) \sum_{h=t+1}^{T(t)-1} \gamma^{h-t-1} \rho_{t:h-1} + \gamma^{T(t)-t-1} \rho_{t:T(t)-1} \right]}.
\]
Per-decision Importance Sampling: Intuition*

- Unroll returns as sum of rewards

\[ \rho_{t:T-1}G_t = \rho_{t:T-1}(R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{T-t-1} R_T) \]

\[ = \rho_{t:T-1}R_{t+1} + \gamma \rho_{t:T-1}R_{t+2} + \cdots + \gamma^{T-t-1} \rho_{t:T-1}R_T. \]

- 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

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

- Equivalent expectation for return

\[
\mathbb{E} [\rho_{t:T-1} G_t] = \mathbb{E} [\tilde{G}_t] = \mathbb{E} [\rho_{t:t} R_{t+1} + \gamma \rho_{t:t+1} R_{t+2} + \cdots + \gamma^{T-t-1} \rho_{t:T-1} R_T]
\]
Per-decision Ordinary Importance Sampling*

- Estimator for Ordinary Importance Sampling:

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

- Estimator for Per-reward Ordinary Importance Sampling:

\[ V(s) = \frac{\sum_{t \in \mathcal{T}(s)} \tilde{G}_t}{|\mathcal{T}(s)|} , \]
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