5- Value-based Simulation

Melih Kandemir

University of Southern Denmark
Department of Mathematics and Computer Science (IMADA)
kandemir@imada.sdu.dk

Fall 2022

Simulation-based methods

$$J^{\mu}(i) = \sum_{j=0}^{n} p_{ij}(\mu(i))(g(i,u,j) + \alpha J(j))$$

Dynamic programming is not feasible when:

i) The state space is too large: $n \to \infty$

ii) Transition probabilities are not known: $p_{ij}(\mu(i)) = ?$

Approximate by **simulation**, i.e. collect samples from the environment:

$$i_0, i_1, \ldots, i_N$$

where

$$i_k \sim \mathrm{Cat}(p_{i1}(\mu(i)), p_{i2}(\mu(i)), \ldots, p_{in}(\mu(i))), \qquad k = 0, \ldots, N-1.$$

The symbol $\sim$ means to call the random number generator. Then simply evaluate

$$g(i_0, \mu(i_0), i_1), g(i_1, \mu(i_1), i_2), \ldots, g(i_{N-1}, \mu(i_{N-1}), i_N).$$

Monte Carlo (MC) simulation

Assume a sample set $v_1, \ldots, v_N \sim p(V)$. Then the sample mean is

$$M_N = \frac{1}{N} \sum_{k=1}^{N} v_k \approx \mathbb{E}[v] = \sum_{v \in S} \mathbb{P}(V = v) v.$$

Note that this quantity can be calculated **online**:

$$M_{N+1} = M_N + \frac{1}{N+1}(v_{N+1} - M_N).$$

If the sample set is i.i.d. and $\mathbb{E}[v] = m$, then

$$\mathbb{E}[M_N] = \frac{1}{N} \sum_{k=1}^{N} \mathbb{E}[v_k] = m.$$

If $m = \mathbb{E}[M_N]$ then M_N is an **unbiased** estimator of m. We also have

$$\mathrm{Var}(M_N) = \frac{1}{N^2} \sum_{k=1}^{N} \mathrm{Var}(v_k) = \frac{\sigma^2}{N}.$$

$\lim_{N \to \infty} \mathrm{Var}(M_N) = 0 \Rightarrow M_1, M_2, \ldots \to m$ w.p. 1 (law of large #s).

Monte Carlo RL

- (+) learns directly from episodes of experience.
- (+) is **model-free** (i.e. requires no knowledge of MDP transitions and rewards).
- (+) is based only on generated sample transitions, not complete distributions of all possible transitions.
- (-) works only for **episodic** tasks.
- (o) applies Monte Carlo integration to value approximation.

Wald's identity

When N is a random variable and we condition on it

$$\mathbb{E}[M_N] = \mathbb{E}\left[\mathbb{E}\left[\frac{1}{N}\sum_{k=1}^{N} v_k \middle| N\right]\right] = \mathbb{E}[m] = m$$

but $\mathbb{E}[M_N] \neq m$ hence M_N is a **biased** estimator of the marginal.
Suppose $v_1, v_2, \ldots$ have common mean and $\mathbb{E}[v_k|N \geq k] = \mathbb{E}[v_1]$, then

$$\begin{aligned}
\mathbb{E}\left[\sum_{k=1}^{N} v_k\right] &= \sum_{k=1}^{\infty} \mathbb{P}(N \geq k)\mathbb{E}[v_k|N \geq k] = \mathbb{E}[v_1]\sum_{k=1}^{\infty}\mathbb{P}(N \geq k) \\
&= \mathbb{E}[v_1]\sum_{k=1}^{\infty}\sum_{n=k}^{\infty}\mathbb{P}(N = n) = \mathbb{E}[v_1]\sum_{n=1}^{\infty} n\mathbb{P}(N = n) \\
&= \mathbb{E}[v_1]\mathbb{E}[N]
\end{aligned}$$

This is the **Wald's identity** very useful for convergence proofs in RL.

Policy evaluation with MC simulation

Simulate a trajectory until terminal state: $i_0, i_1, \ldots, i_N$ such that $i_N = 0$. This is called an **episode**. Denote $k_m(i)$ as the time step when a state i is encountered mth time. Then the observed cost-to-go is

$$c(i,m) = \sum_{k=k_m(i)}^{N-1} g(i_k, \mu(i_k), i_{k+1})$$

and the MC estimate of the true cost-to-go for M encounters is

$$J^\mu(i) = \mathbb{E}[c(i,m)] \approx \frac{1}{M} \sum_{m=1}^{M} c(i,m).$$

This is called the **every-visit** method. Start with $J(i) = 0, \forall i$ and update after each encounter

$$J(i_k) := J(i_k) + \gamma(i_k)(c(i_k, m_{i_k}) - J(i_k))$$

where $\gamma(i_k) = 1/m_{i_k}$ with m_{i_k} the count of visits to i_k until time step k. Possible to use other step sizes as long as the Robbins-Monro conditions are satisfied.

Every-visit estimator is biased but consistent

Denote $c(i,m,k)$ as $c(i,m)$ of kth of K simulated trajectories, K_i of which visit i. Then

$$\lim_{K\to\infty} \frac{\sum_{\{k|n_k\geq 1\}} \sum_{m=1}^{n_k} c(i,m,k)}{\sum_{\{k|n_k\geq 1\}} n_k}$$

$$= \lim_{K_i\to\infty} \frac{\frac{1}{K_i}\sum_{\{k|n_k\geq 1\}} \sum_{m=1}^{n_k} c(i,m,k)}{\frac{1}{K_i}\sum_{\{k|n_k\geq 1\}} n_k}$$

$$= \frac{\mathbb{E}\left[\sum_{m=1}^{n_k} c(i,m,k)\middle| n_k\geq 1\right]}{\mathbb{E}[n_k|n_k\geq 1]} = \frac{\mathbb{E}\left[\mathbb{E}\left[\sum_{m=1}^{n_k} c(i,m,k)\middle| n_k\geq m\right]\right]}{\mathbb{E}[n_k|n_k\geq 1]}$$

$$= \frac{\mathbb{E}[c(i,1,k)n_k]}{\mathbb{E}[n_k|n_k\geq 1]} = \frac{\mathbb{E}[c(i,1,k)|n_k\geq 1]\mathbb{E}[n_k|n_k\geq 1]}{\mathbb{E}[n_k|n_k\geq 1]} = J^\mu(i)$$

where $\mathbb{E}[c(i,m,k)|n_k\geq m] = J^\mu(i)$ due to Markov property. Estimator

$$\frac{\sum_{\{k|n_k\geq 1\}} c(i,1,k)}{K_i}$$

is for the **first-visit** method and it is also consistent.

MC policy evaluation with Temporal Difference

$$J(i_k) := J(i_k) + \gamma \left[\left(\sum_{m=k}^{N-1} g(i_m, i_{m+1}) \right) - J(i_k) \right]$$

$$= J(i_k) + \gamma \left[\left(\sum_{m=k}^{N-1} \underbrace{g(i_m, i_{m+1}) + J(i_{m+1}) - J(i_m)}_{d_m} \right) \right]$$

- $d_m = g(i_m, i_{m+1}) + J(i_{m+1}) - J(i_m)$ is called the **Temporal Difference (TD)**
- $g(i_m, i_{m+1}) + J(i_{m+1})$ and $J(i_m)$ estimate the same quantity. Backpropagate the mismatch as error, hence the name.
- Also possible to do sequential updates

$$J(i_k) := J(i_k) + \gamma d_m, \qquad m = 1, \ldots, N-1.$$

Multi-step TD

Denote by ∞ an unknown time step N_e with $i_{N_e} = 0$.

$$J^{\mu}(i_k) = \mathbb{E}\left[\sum_{m=0}^{\infty} g(i_{k+m}, i_{k+m+1})\right] = \mathbb{E}[g(i_k, i_{k+1}) + J^{\mu}(i_{k+1})]$$

The stochastic approximation of the latter is

$$J^{\mu}(i_k) := J(i_k) + \gamma(g(i_k, i_{k+1}) + J(i_{k+1}) - J(i_k))$$

One can also go with stochastic approximations for l steps and **bootstrap** after that point:

$$J^{\mu}(i_k) = \mathbb{E}\left[\sum_{m=0}^{l} g(i_{k+m}, i_{k+m+1}) + J^{\mu}(i_{k+l+1})\right]$$

The question is what l should be.

TD(λ)

Use domain knowledge if available or the answer below otherwise:

$$J^{\mu}(i_k) = (1-\lambda)\mathbb{E}\left[\sum_{l=0}^{\infty}\lambda^l\left(\sum_{m=0}^{l} g(i_{k+m}, i_{k+m+1}) + J^{\mu}(i_{k+l+1})\right)\right].$$

Interchanging the sum order and using $(1-\lambda)\sum_{l=m}^{\infty}\lambda^l = \lambda^m$ gives

$$\begin{aligned}
&J^{\mu}(i_k) \\
&\quad= \mathbb{E}\left[(1-\lambda)\sum_{m=0}^{\infty} g(i_{k+m}, i_{k+m+1})\sum_{l=m}^{\infty}\lambda^l + \sum_{l=0}^{\infty} J^{\mu}(i_{k+l+1})(\lambda^l - \lambda^{l+1})\right] \\
&\quad= \mathbb{E}\left[\sum_{m=0}^{\infty}\lambda^m\Big(g(i_{k+m}, i_{k+m+1}) + J^{\mu}(i_{k+m+1}) - J^{\mu}(i_{k+m})\Big)\right] + J^{\mu}(i_k) \\
&\quad= \mathbb{E}\left[\sum_{m=k}^{\infty}\lambda^{m-k} d_m\right] + J^{\mu}(i_k)
\end{aligned}$$

TD(λ)

The corresponding Robbins-Monro stochastic approximation is

$$J(i_k) := J(i_k) + \gamma \sum_{m=k}^{\infty} \lambda^{m-k} d_m$$

- $\lambda = 1 \Rightarrow$ MC policy evaluation algorithm, a.k.a. $TD(1)$.
- $\lambda = 0 \Rightarrow$ 1-step TD, a.k.a. $TD(0)$.
- $\lambda < 1$ discounts the effect of state transitions on the cost estimate of the current state. Different from the cost discount factor!
- **Every visit**

$$J(i) := J(i) + \gamma \sum_{j=1}^{M} \sum_{m=m_j}^{\infty} \lambda^{m-m_j} d_m$$

- **First visit**

$$J(i) := J(i) + \gamma \sum_{m=m_1}^{\infty} \lambda^{m-m_1} d_m$$

Online versus offline policy evaluation

Assume a simulated trajectory $i_0, i_1, \ldots, i_N$.

- Offline:

$$J(i_0) := J(i_0) + \gamma(\lambda^0 d_0 + \lambda^1 d_1 + \lambda^2 d_2 + \ldots)$$
$$J(i_1) := J(i_1) + \gamma(\lambda^0 d_1 + \lambda^1 d_2 + \ldots)$$

- Online:

$$J(i_0) := J(i_0) + \gamma\lambda^0 d_0 \qquad \text{after } (i_0, i_1)$$

$$J(i_0) := J(i_0) + \gamma\lambda^1 d_1 \qquad \text{after } (i_1, i_2)$$
$$J(i_1) := J(i_1) + \gamma\lambda^0 d_1$$

$$J(i_0) := J(i_0) + \gamma\lambda^2 d_2 \qquad \text{after } (i_2, i_3)$$
$$J(i_1) := J(i_1) + \gamma\lambda^1 d_2$$

Eligibility coefficients

$$J(i) := J(i) + \gamma \sum_{m=0}^{\infty} z_m(i) d_m$$

where $z_m(i)$ are called **eligibility coefficients.**

(a) $z_m(i) = \lambda^{m-m_1}, \quad m \geq m_1$ is first-visit TD(λ)

(b) $z_m(i) = \sum_{\{j|m_j \leq m\}} \lambda^{m-m_j},$ is every-visit TD(λ)

(c) $z_m(i) = \lambda^{m-m_j}, \quad m_j \leq m \leq m_{j+1}, \forall j$ is **restart** TD(λ)

- The restart variant resets $z_m(i)$ at every new visit to i, hence treats each trajectory between two visits as if they are separate.
- It is observed that that restart variant outperforms the every-visit variant.

Q-Factors

$$Q^{\mu}(i,u) = \sum_{j=0}^{n} p_{ij}(u)(g(i,u,j) + J^{\mu}(j))$$

Then policy improvement reads as

$$\bar{\mu}(i) = \arg \min_{u \in U(i)} Q^{\mu}(i,u), \qquad i = 1, \ldots, n.$$

- Any μ may tend to explore region R more than the rest. Hence $J^{\mu}(i)$ will have good quality if $i \in R$.
- If μ drives i to $\bar{R}$ with $\bar{R} \cap R = \emptyset$, then $J^{\mu}(i)$ will be poor for $i \in \bar{R}$. So decide the initial states well.
- One solution is **iterative resampling**: Do not update μ if previous simulation ends in $i \in \bar{R}$. Simulate few times with different $i \in \bar{R}$ using the old μ.

Optimistic policy iteration

- The actor uses μ for control and critic observes outcome to compute J^μ.
- In vanilla policy iteration, actor and critic communicate rarely, as it takes multiple steps to solve policy evaluation while a single step to do the policy update.
- It is in fact possible to update the policy before policy evaluation converges. This approach is called **optimistic policy iteration**.

repeat
  for $e = 1, \ldots, E$ **do** ▷ Episodes
    for $k = 0, \ldots, N_e - 1$ **do** ▷ Policy evaluation
      $i_{k+1} \sim \mathrm{Cat}(p_{i_k 0}^{\mu(i_k)}, \ldots, p_{i_k n}^{\mu(i_k)})$
      $J(i_k) := g(i_k, \mu(i_k), i_{k+1}) + \alpha J(i_{k+1})$
    end for
  end for
  $T_{\mu_{t+1}} J^{\mu_t} := T J^{\mu_t}$ ▷ Policy improvement
until $J^{\mu_{t+1}} = J^{\mu_t}$

When matrix inversion is infeasible

repeat
  for $e = 1, \ldots, E$ **do** ▷ Episodes
    for $k = 0, \ldots, N_e - 1$ **do** ▷ Policy evaluation
      $i_{k+1} \sim \mathrm{Cat}(p_{i_k 0}(\mu(i_k)), \ldots, p_{i_k n}(\mu(i_k)))$
      $J(i_k) := \sum_{j=1}^{n} p_{i_k j}(\mu(i_k)) \Big(g(i_k, \mu(i_k), j) + \alpha J(j) \Big)$
    end for
  end for
  $T_{\mu_{t+1}} J^{\mu_t} := T J^{\mu_t}$ ▷ Policy improvement
until $J^{\mu_{t+1}} = J^{\mu_t}$

This is a smart way of doing asynchronous updates, where the computation cost of the value of a state is proportional to its probability of occurrence.

When even one pass over states is infeasible

repeat
  for $e = 1, \ldots, E$ **do** ▷ Episodes
    for $k = 0, \ldots, N_e - 1$ **do** ▷ Policy evaluation
      $i_{k+1} \sim \mathrm{Cat}(p_{i_k 0}(\mu(i_k)), \ldots, p_{i_k n}(\mu(i_k)))$
      $J(i_k) := g(i_k, \mu(i_k), i_{k+1}) + \alpha J(i_{k+1})$
    end for
  end for
  $T_{\mu_{t+1}} J^{\mu_t} := T J^{\mu_t}$ ▷ Policy improvement
until $J^{\mu_{t+1}} = J^{\mu_t}$

Note that we actually do not need to know $p_{ij}(\mu(u))$ if we are in a real environment, as only taking action $u(i_k)$ would drive us to i_{k+1}.

Optimistic policy iteration with TD(λ)

repeat
  for $e = 1, \ldots, E$ **do** ▷ Episodes
    for $k = 0, \ldots, N_e - 1$ **do** ▷ Policy evaluation
      $i_{k+1} \sim \mathrm{Cat}(p_{i_k 0}(\mu(i_k)), \ldots, p_{i_k n}(\mu(i_k)))$
      $d_k := g(i_k, \mu(i_k), i_{k+1}) + J(i_{k+1}) - J(i_k)$
      $\mathcal{M}_{i_k} := \mathcal{M}_{i_k} \cup k$ ▷ Save visit time
    end for
    for $i = 0, \ldots, n$ **do** ▷ Offline variant
      $J(i) := J(i) + \gamma \sum_{m_j \in \mathcal{M}_i} \sum_{m=m_j}^{N_e} \lambda^{m-m_j} d_m$
    end for
  end for
  $T_{\mu_{t+1}} J^{\mu_t} := T J^{\mu_t}$ ▷ Policy improvement
until $J^{\mu_{t+1}} = J^{\mu_t}$

Optimism (and compute speed) at the extremes

repeat
  for $e = 1, \ldots, E$ **do** ▷ Episodes
    for $k = 0, \ldots, N_e - 1$ **do** ▷ Policy evaluation
      $i_{k+1} \sim \mathrm{Cat}(p_{i_k 0}(\mu(i_k)), \ldots, p_{i_k n}(\mu(i_k)))$
      $J(i_k) := g(i_k, \mu(i_k), i_{k+1}) + \alpha J(i_{k+1})$
      $T_{\mu_{t+1}} J^{\mu_t} := T J^{\mu_t}$ ▷ Policy improvement
    end for
  end for
until $J^{\mu_{t+1}} = J^{\mu_t}$

Value iteration with ultimate optimism

In terms of Q-factors, Bellman equation is expressed as

$$Q^*(i,u) = \sum_{j=0}^{n} p_{ij}(u)(g(i,u,j) + J^*(j)), \qquad i = 1, \ldots, n$$
$$= \sum_{j=0}^{n} p_{ij}(u)\left(g(i,u,j) + \min_{v \in U(i)} Q^*(i,v)\right)$$

and value iteration as the update rule

$$Q(i,u) := \sum_{j=0}^{n} p_{ij}(u)\left(g(i,u,j) + \min_{v \in U(i)} Q(i,v)\right)$$

and generally with step size $\gamma \in (0,1]$ as

$$Q(i,u) := (1-\gamma)Q(i,u) + \gamma \sum_{j=0}^{n} p_{ij}(u)\left(g(i,u,j) + \min_{v \in U(i)} Q(i,v)\right).$$

Q-Learning

Approximate optimistic value iteration by replacing the expectation on next state with a single sample.

for $e = 1, \dots, E$ **do** ▷ Episodes
  for $k = 0, \dots, N_e - 1$ **do** ▷ Value iteration
    $i_{k+1} \sim \mathrm{Cat}(p_{i_k 0}(\mu(i_k)), \dots, p_{i_k n}(\mu(i_k)))$
    $Q(i_k, \mu(i_k)) := (1-\gamma) Q(i_k, \mu(i_k))$
      $+\gamma \left[g(i_k, u, i_{k+1}) + \min_{v \in U(i)} Q(i_{k+1}, v) \right]$
  end for
end for

Here the **behavior policy** $\mu(i)$ can be chosen in multiple ways

i) $\epsilon-$**Greedy:** $\mathbb{P}(\mu(i) = u) = \mathbb{1}_{u=u^*} \left(1 - \epsilon + \frac{\epsilon}{|U(i)|}\right) + \mathbb{1}_{u \neq u^*} \frac{\epsilon}{|U(i)|}$
where $u^* = \arg\min_{v \in U(i)} Q(i, v)$. Greedy (also on-policy) if $\epsilon = 0$,

ii) **Temperature-scaled softmax:** For temperature parameter $T > 0$

$$\mathbb{P}(\mu(i) = u) = \frac{\exp(-Q(i,u)/T)}{\sum_{v \in U(i)} \exp(-Q(i,v)/T)}.$$

On-policy versus off-policy RL

- Suppose the policy is greedy and the MDP is deterministic, then the entire episode following i_0 is determined. **Nothing to average!**
- **Remedy:** Take random actions $\Rightarrow$ **Exploring Starts (ES)**.
- When the policy is arbitrarily random, it is hard to target important states in large state spaces.
- Classify RL methods into two:
  - **On-policy** methods generate data from the policy being learned.
  - **Off-policy** methods use different policies for learning and data generation.
- On-policy methods use **soft** policies for exploration, i.e. $\mathbb{P}(\mu(i) = u) > 0,\ \forall i, u$.
- Off-policy methods trade exploration and exploitation.

Off-policy methods

Solving RL requires solving two conflicting subtasks:

- **exploration:** learn as many states as possible
- **exploitation:** learn important states better

But how to know which state is more important without knowing the optimal policy? Use two policies instead of one:

- **target policy:** policy being learned (μ)
- **behavior policy:** policy that generates behavior (b)

Because $\mu \neq b$, we call this approach **off-policy** RL.

$\epsilon-$*greedy* policy improvement theorem

Definition

Denote $\mu(u|i) = \mathbb{P}(\mu(i) = u)$. A policy μ is called $\epsilon-$**soft** if $\mu(u|i) \geq \frac{\epsilon}{|U(i)|}$ for all $u \in U(i)$.

Theorem

For any $\epsilon-$soft policy μ, the $\epsilon-$greedy policy μ' wrt Q^{μ} is an improvement, i.e. $J_{\mu'} \leq J_{\mu}$.

Proof

$$
\begin{aligned}
J^{\mu'}(i) &= Q^{\mu}(i, \mu'(i)) = \sum_u \mu'(u|i) Q^{\mu}(i,u) \\
&= \frac{\epsilon}{|U(i)|} \sum_u Q^{\mu}(i,u) + (1-\epsilon) \min_u \; Q^{\mu}(i,u) \\
&\leq \frac{\epsilon}{|U(i)|} \sum_u Q^{\mu}(i,u) + (1-\epsilon) \sum_u \underbrace{\frac{\mu(u|i) - \frac{\epsilon}{|U(i)|}}{1-\epsilon}}_{\text{sums to 1}} \; Q^{\mu}(i,u) \\
&= \frac{\epsilon}{|U(i)|} \sum_u Q^{\mu}(i,u) + \sum_u \mu(u|i) \; Q^{\mu}(i,u) - \sum_u \frac{\epsilon}{|U(i)|} \; Q^{\mu}(i,u) \\
&= \sum_u \mu(u|i) Q_{\mu}(i,u) = J^{\mu}(i)
\end{aligned}
$$

■

Importance Sampling (IS)

Intuition: Sample from a different distribution from the one being integrated.

$$\mathbb{E}_{p(z)}\Big[f(z)\Big] = \sum_{z} f(z)p(z)$$
$$= \sum_{u} f(z)\frac{p(z)}{q(z)}q(z)$$

then do Monte Carlo integration

$$\mathbb{E}_{p(z)}\Big[f(z)\Big] \approx \frac{1}{N}\sum_{k=1}^{N} f(z^{(k)}) \times \underbrace{\frac{p(z^{(k)})}{q(z^{(k)})}}_{\text{Importance weight}}$$

for a set of $z^{(k)} \sim q(z)$.

IS applied to MC-RL

Assume $i_0, \ldots, i_N$ is the MC sequence used to update state $i_0 = i$

- $z = (u_0, \ldots, u_{N-1})$
- $f(z) = \sum_{k=0}^{N-1} g(i_k, u_k, i_{k+1})$
- $p(z) = \prod_{k=0}^{N-1} \mu(u_k|i_k) p_{i_k i_{k+1}}^{u_k}$
- $q(z) = \prod_{k=0}^{N-1} b(u_k|i_k) p_{i_k i_{k+1}}^{u_k}$

Then the importance weight is given as

$$w = \frac{\prod_{k=0}^{N-1} \mu(u_k|i_k) \cancel{p_{i_k i_{k+1}}^{u}}}{\prod_{k=0}^{N-1} b(u_k|i_k) \cancel{p_{i_k i_{k+1}}^{u}}},$$

which does not depend on transition probabilities! Note that we require

$$\mu(u|i) > 0 \Rightarrow b(u|i) > 0, \quad \forall (i, u)$$

which is called the **coverage** assumption.

Ordinary vs Weighted IS

Assume we sampled R sequences: $u_k^r \sim b(u|i_k^r), i_{k+1}^r \sim p_{i_k^r i_{k+1}^r}^{u_k^r} \forall k, r$

$$u_0^1,(i_2^1,u_1^1),\ldots,(i_N^1,u_N^1) \rightarrow \sum_{k=0}^{N-1} g(i_k^1,u_k^1,i_{k+1}^1) = C_1$$

$$\vdots \qquad\qquad \vdots$$

$$u_0^R,(i_2^R,u_1^R),\ldots,(i_N^R,u_N^R) \rightarrow \sum_{k=0}^{N-1} g(i_k^R,u_k^R,i_{k+1}^R) = C_R$$

Calculate an importance weight for each

$$w_r = \prod_{k=0}^{N-1} \mu(u_k|i_k^r)/b(u_k|i_k^r).$$

Then we can perform IS two ways

- **Ordinary IS**: $J(i) := \frac{1}{R}\sum_{r=1}^{R} w_r C_r$
- **Weighted IS**: $J(i) := \sum_{r=1}^{R}\left[\left(\frac{w_r}{\sum_{r=1}^{R} w_r}\right)C_r\right]$

Ordinary vs Weighted IS

- Ordinary IS is unbiased, but its variance is unbounded (due to the importance weight). Problematic for loopy trajectories.
- Weighted IS is biased, but its variance is bounded.
- Weighted IS is preferred more often.
- Bias of Weighted IS converges to zero. Hence, it is asymptotically unbiased.
- Ordinary IS has poor convergence properties.

Incremental IS

Given a set of sequences with corresponding observed costs $C_1, C_2, \cdots, C_N$, all starting with the same state and having the corresponding importance weights $w_1, w_2, \cdots, w_N$, we can do the online update for ordinary IS as

$$J(i) := J(i) + \frac{1}{r}\Big[w_k C_r - J(i)\Big],$$

and for weighted IS as follows. Define $\beta_r = \beta_{r-1} + w_r$ with $\beta_0 = 0$,

$$\begin{aligned}
\beta_r J_r(i) =& C_r w_r + \beta_{r-1} J_{r-1}(i) \\
=& C_r w_r + (\beta_r - w_r) J_{r-1}(i) \\
=& C_r w_r + \beta_r J_{r-1}(i) - w_r J_{r-1}(i) \\
\therefore \quad J(i) =& \frac{C_r w_r + \beta_r J_{r-1}(i) - w_r J_{r-1}(i)}{\beta_r} \\
\therefore \quad J(i) :=& J(i) + \frac{w_r}{\beta_r}\Big[C_r - J_{r-1}(i)\Big]
\end{aligned}$$

Off-policy MC control

Initialize for all $i \in \mathcal{S}, u \in U$:

$Q(i,u) := \text{arbitrary}, \quad \beta(i,u) := 0, \quad \mu(i) := \arg\min_u Q(i,u)$

repeat forever

$b :=$ any soft policy

Sample episode $\{(u_k, i_{k+1}) | u_k \sim b(u|i_k), i_{k+1} \sim p_{i_k i_{k+1}}^{u_k}\}$

$C := 0, \quad w := 1$

for $k := N-1 \to 0$:

$C := C + g(i_k, u_k, i_{k+1})$

$\beta(i_k, u_k) := \beta(i_k, u_k) + w$

$Q(i_k, u_k) := Q(i_k, u_k) + \frac{w}{\beta(i_k, u_k)}[C - Q(i_k, u_k)]$

$u^* := \arg\min_u \ Q(i_k, u)$

if $u_k \neq u^*$ **then break**

$w := w(1/b(u_k|i_k))$

end for

Pros and cons of off-policy RL

- Off-policy methods incur higher variance, hence converge slower than on-policy methods.
- Off-policy methods have on-policy methods as their special case, hence they are more general and powerful.
- Off-policy methods can learn from a non-learning controller (e.g. a human expert), on-policy methods cannot.