Monte Carlo Methods

- learning methods without complete knowledge, only experience
+ Simulated experience is powerful
+ still require a model, but for simpler transition generation
* still often leads to optimal policy without defining distributions explicitly
- MCM's (Monte Carlo methods) solve RL by averaging sample returns

+ here, defined only for episodic tasks
* learning is done between episodes, not online
* learning on complete returns, not partial returns, discussed later
- Going back to bandits, they averaged reward to understand optimal action
+ now avg return, because rewards are interrelated

- nonstationary from POV of previous states
+ to handle nonstationarity, adapt GPI
* don't compute from known value function, instead learn value function from sample returns of MDP

- will use same approach as DP
+ understand prediction problem v_π, q_π for policy π (arbitrary)
+ then policy improvement
+ then control problem and solution via GPI

Monte Carlo Prediction

- easiest way to evaluate return of state is to average all returns experienced by state
+ should converge to expected value
- estimate $v_\pi(s)$
+ s can be visited by an agent more than once in an episode!
* focus on first visit to s
* every visit to s similar, but different theoretical properties and application

- First visit MC prediction

```
$V(s) \in \mathbb{R}$ arbitrarily for $\forall s \in S$
$Returns(s) = $ empty list $\forall s \in S$

While true:
    Generate full episode using $\pi$: $S_0, A_0, R_0, S_1, A_1, R_1, \dots, S_{T-1}, A_{T-1}, R_T$
    $G = 0$
    for each $t$ in $T-1, T-2, \dots, 0$:
        $G = \gamma G + R_{t+1}$
        Unless $S_t$ in $S_0, S_1 \dots S_{t-1}$:
            Append $G$ to $Returns(S_t)$
            $V(S_t) = avg(Returns(S_t))$
```

- Both first visit MC and every visit MC converge to $v_\pi(s)$ as visits to $s \to \infty$

+ each return is estimate of v_π

+ finite variance (s.d: $1/\sqrt{n}$ for n returns)

* every-visit converges quadratically

EXAMPLE 5.1: BLACKJACK

finite MDP

full game = 1 ep

rewards = +1 for win, -1 for loss, 0 for draw

No discounting

actions = hit, stick

states → dealer's showing hand, players card sum, player's usable ace

total of 200 states

What is value function if policy is to always hit unless player has 20 or 21?

EXERCISE 5.1:

- Estimated value function spikes at 20/21 because the likelihood of winning is extremely high
- If the dealer is showing an ace, they simply have a higher chance of hitting Blackjack no matter what, decreasing the value of all those states
- usable ace raises value of all states for the player

EXERCISE 5.2: No, I would expect no difference because no state is re-reachable in this problem.

- Simply infeasible to apply DP here due to compute for complete knowledge
- Does a backup diagram work?

+ MCM backup diagrams show sampled states, from beginning to end of one episode.

* fundamentally diff from single step all state transitions from a single state

- MCM does not bootstrap

+ does not rely on previous estimates

+ each estimate independent

+ computationally inexpensive to gather return of a single state or set of states

EXAMPLE 5.2: Soap bubble with varying height to frame. What does the surface look like?

- iterative solution: fixed grid points at frame, next point heigh is average of heights of 4 points next to it
+ iterate over and over, converges on final solution
- MCM, stand on surface and take a random walk, equal probability
+ any points height is the average of the boundary heights starting at said point.
- might prefer MCM if only need one point or small set of points

Monte Carlo Estimation of Action Values

- with no model, prefer estimating $\boxed{\text{action values}}$ over state values
- model make it easy => just pick whatever action leads to next best state

+ no model, must find best policy by finding q_*, which requires action-value policy evaluation.

- identical to state estimation for every visit and first visit except for a state action pair

+ converges quadratically

+ issue: some (s, a) may never be visited

× can't then estimate actual return

× general problem of maintaining exploration

* one way to address is to specify episodes start in a state-action pair with every pair have a nonzero probability of being selected to start in

* called exploring starts

* not a useful long term strategy, but stuck to for this chapter

Monte Carlo Control

- control - approximate optimal policies
- same idea as general GPI, contrasting policy iteration with value iteration
- $v_{\pi_0} \xrightarrow{E} q_{\pi_0} \xrightarrow[I]{} v_{\pi_1} \xrightarrow[E]{} q_{\pi_1} \rightarrow \cdots \rightarrow v_{\pi_*} \xrightarrow{E} q_*$

- we assume infinite eps and exploring starts
+ each q_{π_k} will be computed for π_k
- policy → greedy w.r.t action-value function

$$\pi(s) \doteq \underset{a}{\arg\max}\ q(s,a)$$

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

- this assumes π_{k+1} is always uniformly better than π_k for convergence
- now we unravel the 2 assumptions of infinite eps and exploring starts
+ first remove infinite eps in 2 ways

x one: keep estimation bounds on small, guaranteeing correctness to some approximation

* only practical on small problems

x do not worry about total evaluation

* treat like 'fast' GPI in DP, but only do a single step of value eval before improving

x eval ↔ iterate between episodes

Monte Carlo ES (Exploring Starts)

```
Init:
    $\pi(s) \in \mathcal{A}(s)$ arbitrarily $\forall s \in \mathcal{S}$
    $Q(s, a) \in \mathbb{R}$ arbitrarily $\forall s \in \mathcal{S}, a \in \mathcal{A}(s)$
    $G(s, a)$ = empty $\forall s \in \mathcal{S}\ a \in \mathcal{A}(s)$

While true:
    $S_0$ = random($\mathcal{S}$)          s.t. $pr(S_0, A_0) > 0$
    $A_0$ = random($\mathcal{A}(s)$)
    generate an episode from $S_0, A_0$ following $\pi$.
    $G_{ep} = 0$
    For $t$ in $(T-1, 0)$:
        $G_{ep} = R_{t+1} + \gamma G_{ep}$
        Unless $S_t, A_t$ in episode $t=0 \ldots t-1$:
            $G(S_t, A_t) \ll G_{ep}$
            $Q(S_t, A_t) = avg(G(S_t, A_t))$
            $\pi(S_t) = \arg\max_a Q(S_t, a)$
```

- cannot be suboptimal because if policy were, q would change, forcing π to change
- inevitably to converge to optimal functions for stability but not yet proven, open theoretical questions in RL

EXERCISE 5.4: Adjust to optimize

```
init $N(s,a)$
in loop
    $N(s,a) \mathrel{+}= 1$
    $Q(s,a) = Q(s,a) + \frac{1}{N(s,a)} (G - Q(s,a))$
```

Monte Carlo without Exploring Starts

- how to guarantee all actions chosen infinitely often

+ 2 approaches: on-policy & off-policy

x on-policy: improve policy that is used to make decision on action

x off-policy: improve policy that is not the one used to simulate an episode

* MCES is on policy

- on-policy control generally $\boxed{\text{soft}}$: $\pi(a|s) > 0$ $\forall s \in \mathcal{S}$ $\forall a \in \mathcal{A}$

+ shifts slowly to deterministic policy

+ use ε-greedy policy

x nongreedy action taken with pr: $\frac{\varepsilon}{|\mathcal{A}(s)|}$

greedy action pr: $1 - \varepsilon + \frac{\varepsilon}{|\mathcal{A}(s)|}$

+ ε-greedy policies are examples of $\boxed{\varepsilon\text{-soft}}$

- On Policy MC control is still GPI.

+ as long as policy π moves towards a greedy one convergence assumed

On Policy first-visit MC control (for ε-soft policies), $\pi \approx \pi_*$

```
Init:
    $\pi$ = arbitrary $\varepsilon$-soft policy
    $Q(s,a) \in \mathbb{R}$ arbitrary $\forall s \in \mathcal{S}, a \in \mathcal{A}(s)$
    Returns$(s,a)$ = empty list $\forall s \in \mathcal{S}, a \in \mathcal{A}(s)$

while true:
    Generated episode using $\pi$: $S_0, A_0, R_1 \ldots S_{T-1}, A_{T-1}, R_T$
    $G = 0$
    for $t$ in $(T-1, 0)$:
        $G = \gamma G + R_{t+1}$
        Unless pair $S_t, A_t$ in $S_0, A_0, S_1, A_1 \ldots S_{t-1}, A_{t-1}$
            Returns$(S_t, A_t)$ << $G$
            $Q(S_t, A_t)$ = avg(Returns$(S_t, A_t)$)
            $A^* = \underset{a}{\operatorname{argmax}}\, Q(S_t, a)$

            for $a$ in $\mathcal{A}(S_t)$:
                $\pi(a|S_t) = \begin{cases} 1-\varepsilon+\varepsilon/|\mathcal{A}(S_t)| & a = A^* \\ \varepsilon/|\mathcal{A}(S_t)| & a \neq A^* \end{cases}$
```

- is ε-greedy guaranteed improvement ova ε-soft?
+ yes via policy improvement theorem

× ε-soft $\to \pi$, ε-greedy $\to \pi'$

$$q_\pi(s, \pi'(s)) = \sum_a \pi'(a|s)\, q_\pi(s,a)$$

$$= \frac{\varepsilon}{|\mathcal{A}(s)|} \sum_a q_\pi(s,a) + 1 - \varepsilon \max_a q_\pi(s,a)$$

$$\geq \frac{\varepsilon}{|\mathcal{A}(s)|} \sum_a q_\pi(s,a) + (1-\varepsilon) \sum_a \frac{\pi(a|s) - \frac{\varepsilon}{|\mathcal{A}(s)|}}{1-\varepsilon}\, q_\pi(s,a)$$

$$\geq \cancel{\frac{\varepsilon}{|\mathcal{A}(s)|} \sum_a} q_\pi(s,a) - \cancel{\frac{\varepsilon}{|\mathcal{A}(s)|} \sum_a} q_\pi(s,a) + \sum_a \pi(a|s)\, q_\pi(s,a)$$

$$\geq v_\pi(s)$$

× thus $\pi' > \pi$ ($v_{\pi'}(s) > v_\pi(s)\ \forall s \in \mathcal{S}$)

- this only holds when π' & π are optimal among the ε-soft policies

+ assume new env that is the same as old env except:

1) if in (s,a) with pr$(1-\varepsilon)$ do exactly as old env
2) else pr(ε) picks random action, then behaves like old env

× **Best** policy is same as best ε-soft policy

$\tilde{v}_*, \tilde{q}_*$ optimal new env function

- ε-soft policy π only optimal if $v_\pi = \tilde{v}_*$

$$\tilde{v}_*(s) = \max_a \sum_{s',r} \left[(1-\varepsilon) p(s',r|s,a) + \sum_{a'} \frac{\varepsilon}{|\mathcal{A}(s)|} p(s',r|s,a') \right]$$

$$\times \left[r + \gamma \tilde{v}_*(s') \right]$$

$$= (1-\varepsilon) \max_a \sum_{s',r} p(s',r|s,a) \left[r + \gamma \tilde{v}_*(s') \right] +$$

$$\frac{\varepsilon}{|\mathcal{A}(s)|} \sum_{a'} \sum_{s',r} p(s',r|s,a) \left[r + \gamma v_*(s') \right]$$

①

- when ε-soft no longer improves

$$v_\pi(s) = (1-\varepsilon) \max_a q_\pi(s,a) + \frac{\varepsilon}{|\mathcal{A}(s)|} \sum_a q_\pi(s,a)$$

$$= (1-\varepsilon) \max_a \sum_{s',r} p(s',r|s,a) \left[r + \gamma v_\pi(s') \right] +$$

$$\frac{\varepsilon}{|\mathcal{A}(s)|} \sum_a \sum_{s',r} p(s',r|s,a) \left[r + \gamma v_\pi(s') \right]$$

②

× ① = ② because $\tilde{v}_*$ is unique

- ε-soft policy using greedy approach guarantees improvement without exploring starts
+ doesn't matter how action values are computed, but they must be computed exactly

Off Policy Prediction via Importance Sampling

- remember exploration vs exploitation dilemma
+ similar: learning controls est action values based on optimal subsequent action, but must behave non-optimally to explore actions (and discover optimal actions)
- on-policy compromises this approach
- off-policy uses 2: target-policy & behavior-policy

+ target-policy → policy being learned about
+ behavior policy → policy used to generate policy
- learning is being done from data "off" target policy, thus "off-policy learning"

- Rest of book, always consider both types of policies
+ on-policy always simpler
+ off-policy takes a lot of notation to track
x off-policy way more powerful
* target = behavior → on-policy, special case
* non conventional behavior-policy (aka learning controller) like human expert
* seen as key to multistep predictive models of world's dynamics
- start with the prediction problem for a fixed behavior & target policy
+ stated differently: want v_π or q_π but we only have episodes of policy b where $b \neq \pi$
x π → target, b → behavior, both fixed &

given

- require that $\pi(a|s)$ implies $b(a|s)$
because we have seen enough of a from s in b.
+ called assumption of $\boxed{\text{coverage}}$
x b must be stochastic when different from π
x π may be deterministic, ideally for control apps, and b remains stochastic to explore
* for now, don't worry, π remains fixed + given
- off-policy uses $\boxed{\text{importance sampling}}$
+ standard technique for estimating expected value of one dist. given samples from another
- weight sample by likelihood of trajectory of said return in π & b.
+ $\boxed{\text{importance sampling ratio}}$
x given S_t, likelihood of trajectory $A_t, S_{t+1}, A_{t+1}, \ldots S_T$ under a policy π is

$$\Pr\{A_t, S_{t+1}, A_{t+1}, \ldots, S_T \mid S_t, A_{t:T-1} \sim \pi\}$$

$$= \pi(A_t|S_t)\, p(S_{t+1}|S_t, A_t)\, \pi(A_{t+1}|S_{t+1}) \cdots p(S_T \mid S_{T-1}, A_{T-1})$$

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

× thus, the ratio $\rho_{t:T-1}$ is:

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

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

- trajectory depends on unknown MDP transition probabilities, ratio cancels them out

+ ratio only needs both policies & the sequence

- use ρ ratio to expected value for π, not b

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

- time to estimate $v_\pi(s)$ from a batch of observed episodes from b, Monte Carlo alg.

+ definitions / terms:

× time increases between episodes without stopping

* first ep ends at t=100, next starts at T=101

× $\mathcal{T}(s)$ → set of all time steps in which state s was visited

* this is for every visit; first visit would be only for first visit to s

× $T(t)$ → first time of termination following time t

× G_t → return from $t \to T(t)$

+ Now, $\{G_t\}_{t \in \mathcal{J}(s)}$ = returns for state s

+ $\{\rho_{t:T-1}\}_{t \in \mathcal{J}(s)}$ are corresponding sampling ratios

– $v_\pi(s)$ is scaling the returns by ratios & averaging

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

+ known as $\boxed{\text{ordinary importance sampling}}$ if using simple average

+ weighted importance sampling with weighted avg:

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

or 0 if denominator = 0

- intuition for difference:
+ weighted average has ratio cancel out upon first visit

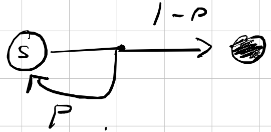
x reasonable est., but $V_b(s)$, not $V_\pi(s)$

+ ordinary average will be $v_\pi(s)$ always, by extremely biased
- diff expressed in bias & variance
+ ordinary average: unbiased, weighted: biased
+ Ordinary avg: variance unbounded by ratio
  Weighted avg: bounded by weight

x weighted avg variance converges to 0

- in practice weighted avg is stronger and preferred
+ ordinary easier to extend & learn from
- above applied to first visit
+ every-visit, all biased, but falls to 0 as samples increase.
- every visit preferred: easier to track states, easier to extend approximations

EXERCISE 5.5



$r = +1$

$\gamma = 1$

1 episode, 10 steps

first-visit & every-visit value of s?

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

Assume no diff in behavior/target, $\rho = 1$

$\mathcal{T}(s) = 0 \quad G_0 = 10$

$V(s) = \frac{10}{1} = \boxed{10}$ first-visit

Every visit

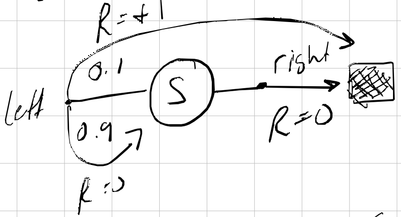
$\mathcal{T}(s) = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9$

$G_{t \in \mathcal{T}(s)} = 10, 9, 8, 7, 6, 5, 4, 3, 2, 1$

$$V(s) = \frac{(G_0 + \dots + G_9)}{10} = \frac{10(11)}{2} = \boxed{5.5}$$

EXAMPLE 5.5: Infinite Variance

ordinary importance sampling = infinite variance if returns are ∞ in variance



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

$b(\text{left} \mid s) = 1/2$

REFER TO FIG 5.4 FOR UNSATISFACTORY CONVERGENCE

Why ∞?

$$\mathrm{Var}[X] \doteq \mathbb{E}[(X-\bar{X})^2]$$

$$= \mathbb{E}[X^2 - 2X\bar{X} + \bar{X}^2]$$

$$= \mathbb{E}[X^2] - \mathbb{E}[2X\bar{X}] + \mathbb{E}[\bar{X}^2]$$

$$= \mathbb{E}[X^2] - 2\bar{X}\,\mathbb{E}[X] + \mathbb{E}[\bar{X}^2]$$

$$= \mathbb{E}[X^2] - 2\bar{X}^2 + X^2$$

$$= \mathbb{E}[X^2] - \bar{X}^2 = \mathbb{E}[X^2] - (\mathbb{E}[X])^2$$

mean finit, so var ∞ iff $\mathbb{E}[X^2] \to \infty$

$$\mathbb{E}\left[\left(\prod_{t=0}^{T-1} \frac{\pi(A_t|S_t)}{b(A_t|S_t)} G_0\right)^2\right]$$

which when expanded by ep length $\to$ ∞

EXERCISE 5.6 What is the analogous $Q(s,a)$ for the $V(s)$

Account for every sample where s,a occurs, not just s!

Thus, redefine $\mathcal{T}$ as $\mathcal{T}(s,a)$ - all timesteps where state a was observed & then action a was taken

EXERCISE 5.7: Why does the weighted average sampling ratio spike before coming down?

Because heavy importance is placed initially on a choice made $b(a|s)$ even if it does not align with the expected reward when experienced first. It quickly peters out as more samples are obtained that match the expected reward

Incremental Updates

- Same as Ch2, but use returns not rewards, cuz of episodes
- for on-policy MC methods, all is identical
- off-policy needs special treatment based on weighted or ordinary importance sampling
- ordinary importance is still simple, because scaled return is still averaged
- weighted sampling:

+ $W_i = \rho_{t_i : T(t_i) - 1}$

$$V_n \doteq \frac{\sum_{k=1}^{n-1} W_k G_k}{\sum_{k=1}^{n-1} W_k}$$

, but incrementally for a G_n

* maintain $C_n = \sum_{k=1}^{n-1} W_i$ (sum up to not including n)

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

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

$C_0 = 0$

Off policy MC prediction (policy eval) for $Q \approx q_\pi$

```
input = arbitrary policy $\pi$
init $\forall s \in \mathcal{S}, a \in \mathcal{A}(s)$
    $Q(s,a) \in \mathbb{R}$ (arbitrarily)
    $C(s,a) = 0$

While true:
    b = any policy doing at least $\pi$
    Generate episode on b: $S_0, A_0, R_1, \dots, R_T$
    $G = 0$
    $W = 1$
    for each t in $(T-1, 0)$ while $W \neq 0$:
        $G = \gamma G + R_{t+1}$
        $C(S_t, A_t) = C(S_t, A_t) + W$
        $Q(S_t, A_t) = Q(S_t, A_t) + \frac{W}{C(S_t, A_t)} [G - Q(S_t, A_t)]$
        $W = W \frac{\pi(A_t | S_t)}{b(A_t | S_t)}$
```

- code is simpler for on policy eval

< $W = 1$

Off Policy Monte Carlo Control

- on-policy: eval target policy while using it for control
- off-policy: behavior can be anything as long as it covers target
- slow learning due to tail of episode backwards could occur

+ solution may exist for $\gamma < 1$ OR temporal-difference learning, discussed in next ch.

EXERCIS 5.11:

Because if $\pi(A_t | S_t) \neq 1$ if statement would have broken out. Only in this problem because π is greedy

Discounting-aware Importance Sampling

- current impl doesn't know about internal structure of reward

+ can we do better?? to reduce variance in off-policy estimates

* e.g. $\gamma = 0$, $t = 0 \dots 100$

therefore $G_0 = R_1$, all else unnecessary

yet $\pi(A_1 | S_1) \dots \pi(A_{99} | S_{99})$ contribute to variance

- degree of partial termination uses discount rate to estimate further return

+ ends on t with $pr = 1 - \gamma$

$t+1 \quad pr = (1-\gamma)\gamma$

$t+2 \quad pr = (1-\gamma)\gamma^2$

+ flat partial return

$$\bar{G}_{t:h} \doteq R_{t+1} + R_{t+2} + \dots + R_h$$

$$0 \le t \le h \le T$$

flat = no discount
h = horizon

- G_t is now sum of flat partial returns

$$G_t = R_{t+1} + \gamma R_{t+2} + \dots + \gamma^{T-t+1} R_T$$

$$= (1-\gamma) R_{t+1} + (1-\gamma)\gamma (R_{t+1} + R_{t+2}) +$$
$$(1-\gamma)\gamma^2 (R_{t+1} + R_{t+2} + R_{t+3}) \dots$$

$$= (1-\gamma) \sum_{h=t+1}^{T-1} \gamma^{h-t-1} \bar{G}_{t:h} + \gamma^{T-t-1} \bar{G}_{t:T}$$

- now, scale importance sampling ratio + ordinary:

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

+ weighted: denominator same as numerator without G_T

Per-decision Importance Sampling

- another way to reduce variance if $\gamma = 1$
- first, redefine numerator:

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

- then, for $\rho_{t:T-1} R_{t+1}$

$$\rho_{t:T-1} R_{t+1} = \frac{\prod^{T-1} \pi(A_t | S_t)}{\prod^{T-1} b(A_t | S_t)} R_{t+1}$$

by expected values

$$\mathbb{E}[\rho_{t:T-1} R_{t+1}] = \mathbb{E}[\rho_{t:t} R_{t+1}]$$

$$\mathbb{E}[\rho_{t:T-1} R_{..+k}] = \mathbb{E}[\rho_{t:t+k-1} R_{t+k}]$$

- so $\mathbb{E}[\rho_{t:T-1} G_t] = \mathbb{E}[\tilde{G}_t]$

$$\tilde{G}_t = \rho_{t:t} R_{t+1} + \gamma \rho_{t:t+1} R_{t+2} + \dots +$$

$$\gamma^{T-t+1} \rho_{t:T-1} R_t$$

- $\boxed{\underline{\text{per decision}}}$

- in the ordinary importance case it works with lower variance

+ have not proven convergence in weighted case

Summary

- MC solves 3 problems of DP
  1) learn from interaction with env, no model
  2) use simulation, or sample models
  3) focus MC on subset of states for eff. learning

  * 4) less harmed by violating Markov Prop by not bootstrapping
- on-policy method tries to learn optimal policy while exploring
- off-policy learns deterministic policy while using a separate policy for the world