Lecture-04

Dynamic Programming :-

access to the dynamics of the world

$p(s', r \mid s, a)$

- Policy Iteration
- Value Iteration.

Policy evaluation : Given a fixed policy, find v_π

Policy Improvement : Given v_π, find π'

$\pi' \geq \pi$

$$\pi_0 \xrightarrow{PE} v_{\pi_0} \xrightarrow{PI} \pi_1 \xrightarrow{PE} v_{\pi_1} \xrightarrow{PI} \pi_2 \rightarrow \cdots \rightarrow \pi_* \rightarrow v_{\pi_*} \rightarrow \pi_*$$

Policy Iteration (using iterative policy evaluation) for estimating $\pi \approx \pi_*$

1. Initialization
   $V(s) \in \mathbb{R}$ and $\pi(s) \in \mathcal{A}(s)$ arbitrarily for all $s \in \mathcal{S}$

2. Policy Evaluation
   Loop:
      $\Delta \leftarrow 0$
      Loop for each $s \in \mathcal{S}$:
         $v \leftarrow V(s)$
         $V(s) \leftarrow \sum_{s',r} p(s', r \mid s, \pi(s)) \big[r + \gamma V(s')\big]$
         $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
   until $\Delta < \theta$ (a small positive number determining the accuracy of estimation)

3. Policy Improvement
   policy-stable $\leftarrow$ *true*
   For each $s \in \mathcal{S}$:
      old-action $\leftarrow \pi(s)$
      $\pi(s) \leftarrow \arg\max_a \sum_{s',r} p(s', r \mid s, a) \big[r + \gamma V(s')\big]$
      If *old-action* $\neq \pi(s)$, then *policy-stable* $\leftarrow$ *false*
   If *policy-stable*, then stop and return $V \approx v_*$ and $\pi \approx \pi_*$; else go to 2

Value Iteration, for estimating $\pi \approx \pi_*$

Algorithm parameter: a small threshold $\theta > 0$ determining accuracy of estimation
Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(terminal) = 0$

Loop:
  $\Delta \leftarrow 0$
  Loop for each $s \in \mathcal{S}$:
    $v \leftarrow V(s)$
    $V(s) \leftarrow \max_a \sum_{s',r} p(s', r \mid s, a)\big[r + \gamma V(s')\big]$
    $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
until $\Delta < \theta$

Output a deterministic policy, $\pi \approx \pi_*$, such that
  $\pi(s) = \text{argmax}_a \sum_{s',r} p(s', r \mid s, a)\big[r + \gamma V(s')\big]$

PE

DP

[ ]

Backgammon 10^{20} states.

Asynchronous DP :-

[ ]

Using agents experience in the MDP to update values.

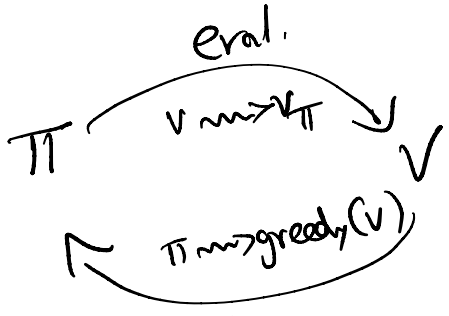
Agent's experience → used to decide what state to update.

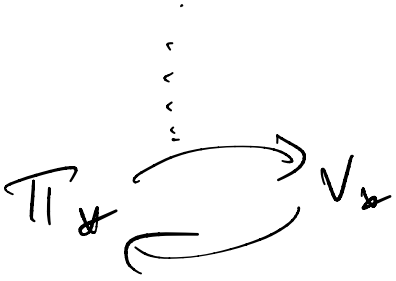
latest value & Policy info from DP → Can guide the agent's decision making.

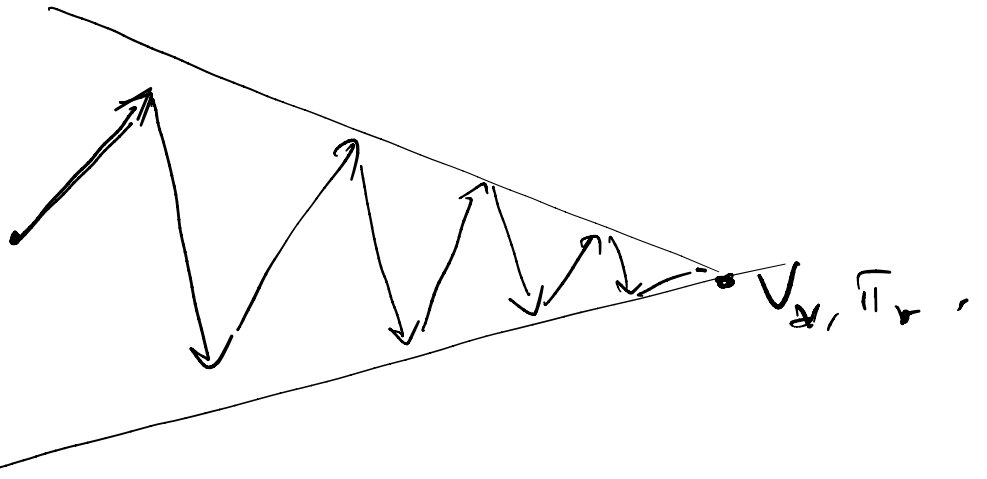
Generalized Policy iteration (GPI):

→ P.E (make the value fn. Consistent with current policy)

→ P.I (make policy greedy wrt Curr. Val. fn)







Efficiency of DP:-

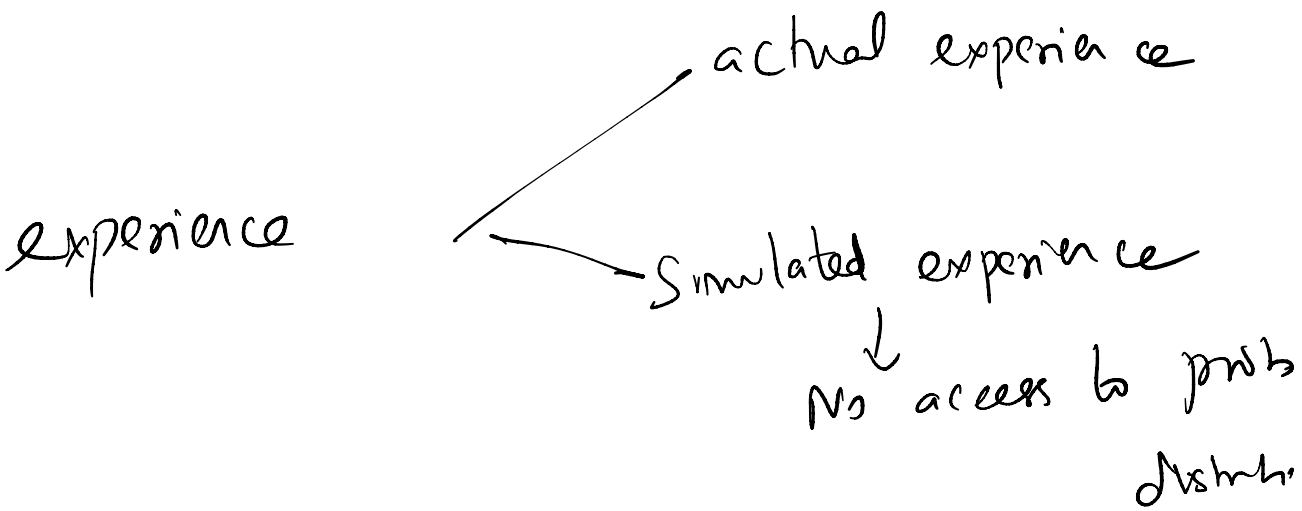
n = # of states

k = # of actions

k^n det. policies.

Monte Carlo Methods.

Can we directly learn the value fns. from experience ?



Monte Carlo ~~methods~~:

— Solving RL based on averaging sample returns

Episodic setting:

Similar to reward estimation in bandits.

bandits

- rewards.
- no states

MC. for RL

- returns.
- Many states.

$S_1 \xrightarrow{a_1} S_2 r_2 \xrightarrow{a_2} S_3 r_3 \xrightarrow{a_3} S_4 r_4 \cdots$

interrelated bandit problem

$q(S, a)$

non-stationary

target.

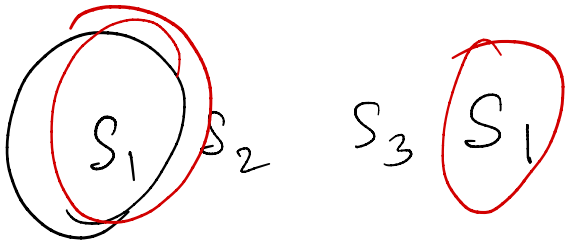
Monte Carlo Prediction:-

Given π, estimate $V_\pi(s)$

$V_\pi(s)$ — Value of a state 's' under policy π, given a set of episodes obtained by following π and passing through s.

Each occurrence of a state 's' in an episode → "visit" to 's'.

First-visit MC → estimates $V_\pi(s)$ as the avg of returns following first visit to 's'.



Every-visit MC → averages return following all visits to s.

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: $S_0, A_0, R_1, S_1, A_1, R_2, \ldots, S_{T-1}, A_{T-1}, R_T$

$G \leftarrow 0$

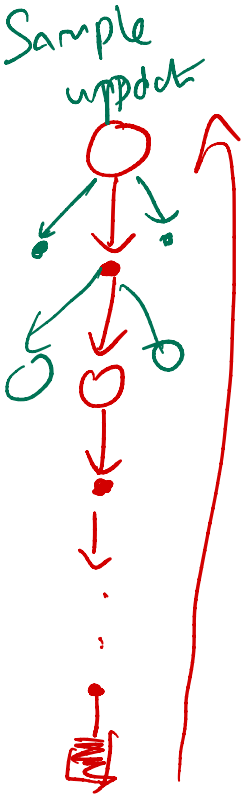
Loop for each step of episode, $t = T-1, T-2, \ldots, 0$:

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

Unless S_t appears in $S_0, S_1, \ldots, S_{t-1}$:

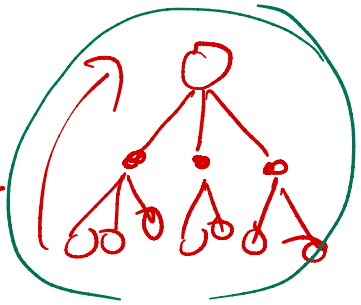
Append G to $Returns(S_t)$

$V(S_t) \leftarrow$ average$(Returns(S_t))$



$$G_t = R_t + \gamma G_{t+1}$$

Reward	1	2	3	4	5
return	15	14	12	9	5



$\gamma = 1$

DP — expected update

MC — Sample update

Note 1:

DP → bootstrapping.

MC → estimate of each state are independent.

Note 2: If you want to estimate value of one state.

MC estimation of action values:-

$$\pi(a|s) = \underset{a}{\operatorname{argmax}}\, q_\pi(s,a)$$

$q_\pi(s,a)$ — expected return when starting in state 's' and taking action 'a' and thereafter following policy π.

"Exploring Starts" assumption:

the episodes start in a state-action pair and that every (S, a) has non-zero prob. of selected as start.

Monte Carlo Control:-

$$\pi_0 \xrightarrow{E} q_{\pi_0} \xrightarrow{I} \pi_1 \xrightarrow{E} q_{\pi_1} \cdots \rightarrow \pi_* \xrightarrow{E} q_{\pi_*}$$

Policy improvement:

$$\pi_{k+1}(s) = \underset{a}{\operatorname{argmax}}\, q_{\pi_k}(s,a)$$

$$q_{\pi_k}(s, \pi_{k+1}(s)) = q_{\pi_k}(s, \underset{a}{\operatorname{argmax}}\, q_{\pi_k}(s,a))$$

$$= \max_a q_{\pi_k}(s,a)$$

$$\geq q_{\pi_k}(s, \pi_k(s)) \geq v_{\pi_k}(s)$$

— 2 strong assumptions.

(1) Episodes have exploring starts.

(2). Policy ev. could be done with infinite # of episodes.

First visit

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 $Returns(S_t, A_t)$
    - $Q(S_t, A_t) \leftarrow \text{average}(Returns(S_t, A_t))$
    - $\pi(S_t) \leftarrow \text{argmax}_a\, Q(S_t, a)$

ES.

Monte-Carlo Control without ES :-

Stochastic / Soft policies.

$\pi(a|s) > 0 \quad \forall s \in S, \; a \in A(s)$

ϵ-greedy poly.

non-greedy action : $\frac{\epsilon}{|A(s)|}$

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

ϵ-soft Policies :-

$$\pi(a|s) \geq \frac{\epsilon}{|A(s)|} \qquad \forall s, a, \; \epsilon > 0.$$

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 $Returns(S_t, A_t)$

$Q(S_t, A_t) \leftarrow$ average$(Returns(S_t, A_t))$

$A^* \leftarrow \arg\max_a Q(S_t, a)$ (with ties broken arbitrarily)

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

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

Any ϵ-greedy policy w.r.t. to q_π is an improvement over any ϵ-soft policy π.

π' = ϵ-greedy policy.

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

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

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

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

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

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

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

$\pi' \geq \pi$

On-policy / Off-policy

- "behavior policy" to explore
- "target policy" being learned

data generated from "off" the target policy.

Off-Policy prediction via Importance Sampling:-

π – target policy } fixed.
b – behavior policy }

"Coverage" assumption:

$$\pi(a|s) > 0 \implies b(a|s) > 0$$

π can be deterministic.

'b' $\quad V_b(s)$

$V_\pi(s)$

Importance Sampling!

Given a starting state S_t,

the prob. of subseq. action traj.

under any policy π is

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

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

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

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

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

$$\mathbb{E}[G_t \mid S_t = s] = V_b(s)$$

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

$\mathcal{T}(s)$ — set of all time steps when state 's' is visited

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

Ordinary Important Sampling (OIS)

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

Weighted importance Sampling (WIS)

first visit off-policy MC with OIS → WIS.

OIS

estimate $V_\pi(s)$ all the time.

No bias

high variance

WIS

estimate V_b for first interaction.

biased.

less variance

Incremental implementation of WIS :-

$G_1, G_2 \cdots G_{n-1}$

$W_1, W_2 \cdots W_{n-1}$

$$V_n = \frac{\sum_{k=1}^{n-1} W_k G_k}{\sum_{k=1}^{n-1} W_k} \qquad n \geq 2.$$

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

and $C_{n+1} = C_n + W_{n+1}$

$C_0 = 0.$

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 MC control, for estimating $\pi \approx \pi_*$

Initialize, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$:
$Q(s,a) \in \mathbb{R}$ (arbitrarily)
$C(s,a) \leftarrow 0$
$\pi(s) \leftarrow \arg\max_a Q(s,a)$ (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, $t = T-1, T-2, \ldots, 0$:
$G \leftarrow \gamma G + R_{t+1}$
$C(S_t, A_t) \leftarrow C(S_t, A_t) + W$
$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \frac{W}{C(S_t, A_t)} [G - Q(S_t, A_t)]$
$\pi(S_t) \leftarrow \arg\max_a Q(S_t, a)$ (with ties broken consistently)
If $A_t \neq \pi(S_t)$ then exit inner Loop (proceed to next episode)
$W \leftarrow W \frac{1}{b(A_t|S_t)}$

bootstrapping → D.P.

→ Monte Carlo
- Monte-Carlo ES
- Monte-Carlo ε-Soft policy
- } on Policy
- off policy Monte Carlo ⇒ OIS / WIS } off Policy.

entire trajectory.

→ Temporal difference (TD) learning.