13th Oct

Mercedes-Benz R & D India

Policy Iteration

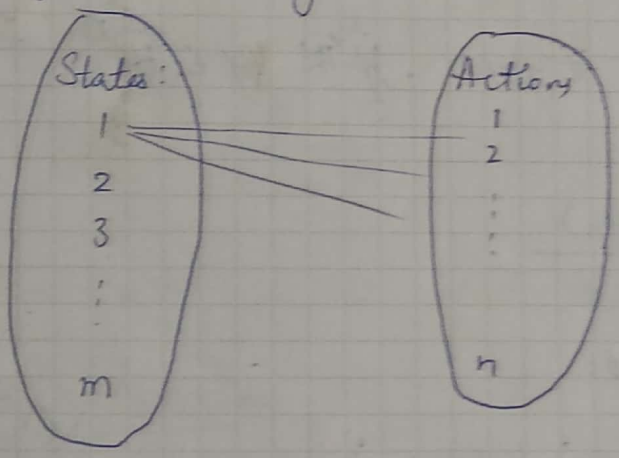
- Stard with inital foliy π_0
- Evaluate foliy π_0 [find V_{π_0}]
- Imfrovement of foliy π_0 to anothe foliy π_1
- Evaluate foliy π_1 find V_{π_1}

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

$\xrightarrow{E}$: foliy evaluation

$\xrightarrow{I}$: foliy improvement

2) After how many iterations, do we get V_{π_*}



$$\underbrace{m \times m \times m \times \cdots m}_{n} \Rightarrow m^n$$

Max no of ~~foli~~ functions that map state space to action space $= m^n$

Functions denote folicies

$\therefore$ Max no of foliyies $= m^n < \infty$

$\pi : S \longrightarrow A$

policy improvement : $\exists$ at least one state $s \in S$

$(\pi_i \to \pi_{i+1})$ s.t $V_{\pi_{i+1}}(s) > V_{\pi_i}(s)$ ~~(does not mean there is degradation~~

& the procedure has converged

but not imply policy have converged. Since there can be multiple policy that are optimal.

$$q_{\pi_0}(s, b) = \{ E_{\pi_0}[r + \gamma V_{\pi_0}(s') \mid x_0 = s, a_0 = s] \}$$

implies, pick intial rewards due to 'b', next pick actions acording to π_0 in future.

$$\pi_1(s) = \underset{b \in A(s)}{\text{argmax}} \; q_{\pi_0}(s, b) \quad \forall s \in S$$

$V_{\pi_1}(s) \geq V_{\pi_0}(s) \quad \forall s \in S$

Either $V_{\pi_1}(s) = V_{\pi_0}(s) \; \forall s \in S$ &

or $\exists \bar{s} \in S$ s.t $V_{\pi_1}(\bar{s}) > V_{\pi_0}(\bar{s})$

If $\forall V_{\pi_1} = V_{\pi_0} \; \forall s$ means π_1 & π_0 are converged.

In the above method, policies are

$\pi_1, \pi_2 \cdots \pi_{m^n}$ the search space is finite.

$V_{\pi_1} < V_{\pi_2} < V_{\pi_3} \cdots V_{\pi_{m^n}}$

$\Rightarrow$ The procedure will converge in a finite number of policy improvement steps.

Unlike randomize policy

$$\pi(a|s) = \begin{matrix} a_1 & w.p\ p_1 \\ a_2 & w.p\ p_2 \\ \\ a_m & w.p\ p_m \end{matrix}$$

$s.t\ p_1 + p_2 + \dots p_m = 1.$

this method has a distribution of policies hence might require to use gradient search. This method can have local max, but in previous method you got global max due to finite search space

~~PI procedure~~

Policy Improvement procedure:

1. Initialization : $V(s) \in \mathbb{R}$ and $\pi(s)\ A(s)$ arbitrarily $\forall s \in S$
2. Policy Eval : Loop

$\Delta \leftarrow 0$

Loop for each $s \in S$

$v \leftarrow V(s)$

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

$\Delta \leftarrow \max(\Delta, |v - V(s)|)$

until $\Delta < \theta$ (small +ve no)

3) Policy Improve:

for each $s \in S$

old action $\leftarrow \pi(s)$

$\pi(s) \leftarrow \underset{a}{\text{argmax}} \sum_{s',r} p(s',s|s,a)[r+\gamma V(s')]$

- If old-action $= \pi(s)$ $\forall s$
  then stop & return $V \equiv V^*$ & $\pi \equiv \pi^*$
- If old action $\neq \pi(s)$ for some (s),
  go back to step 2 & perform P.E for policy π
- If $V_{\pi_k}(s) = V_{old}(s)$ $\forall s$, then
  $V_\pi = V^*$ & π & old-policy are both optimal.

Ex: Jack's Car Rental. - Implement policy iteration.

Value Iteration:

$$V_{k+1}(s) = \max_a E[R_{t+1} + \gamma V_k(S_{t+1}) \mid S_t = s, A_t = a]$$

$$= \max_a \sum p(s', r|s, a)[r + \gamma V_k(s')] \quad k \geq 0$$

$$V_0 = (V_0(s), s \in S)$$

Note:
combines both evaluation & iteration

eg: at $t=0$.
$V_1(s)$ is calculated, later $\max_a$ is taken
~~this~~ first step is eval & second is improvement

Similarity between PI & VI:

Suppose in PI, we run iterative PE step exactly for one iteration & then do improvement.

$\Rightarrow$ Same as VI

Modified Policy Iteration (MPI)

Suppose $m_1, m_2, m_3 \ldots$ are a sequence of integers (all ≥ 1) s.t iterative P.E in PI procedure is run exactly m_1 steps, the first time, m_2 steps the second time etc..

MPI is guarenteed to converge to optimal policy-value function tuple.

Policy evaluated

$$(T_\pi V)(s) = \sum_{s',r} p(s', r | s, \pi(s)) [r + \gamma V_\pi(s')] \quad \forall s \in S$$

Define $(TV)(s) = \max_{a \in A(s)} \sum_{s',r} p(s', r | s, a) [r + \gamma V(s')]$

$\boxed{V = TV}$ — Bellmans equation.

T is a contraction.

$V, TV, T^2V \ldots \to V^*$ as $n \to \infty$ — Value iteration

$V_{n+1} = TV_n \quad \forall n$

V^* is the unique sol^n to bellman eqn. $\boxed{V = TV}$

Value Iteration:

Intialize : $V(s)\ \forall s \in S$

Loop :

$\Delta \leftarrow 0$

loop $\forall s \in S$

$v \leftarrow V(s)$

$$V(s) \leftarrow \max_a \sum_{s',r} p(s', r | s, a)\,[r + V(s')]$$

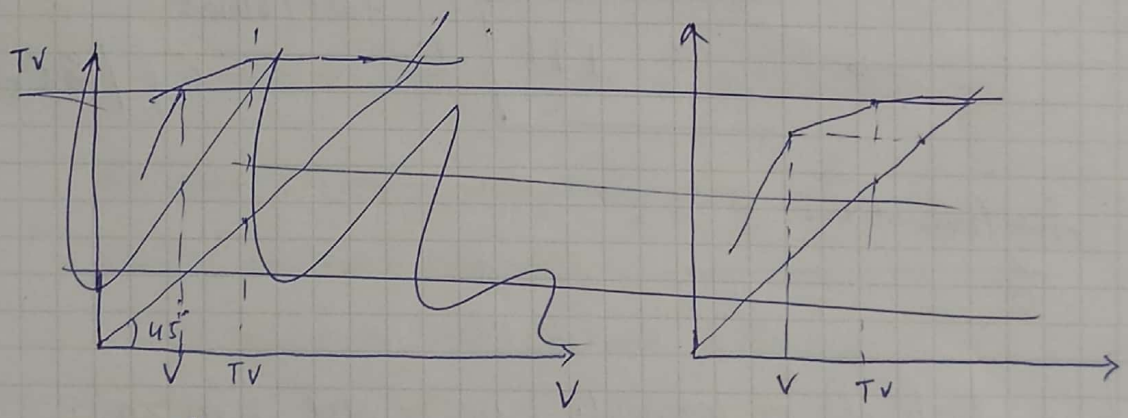
$\Delta \leftarrow \max(\Delta, |v - V(s)|)$

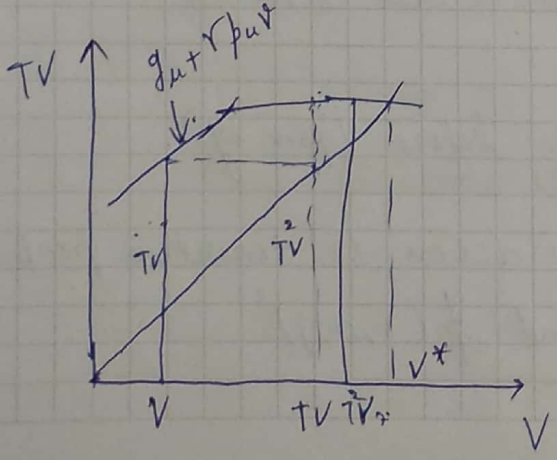
until $\Delta < \theta$

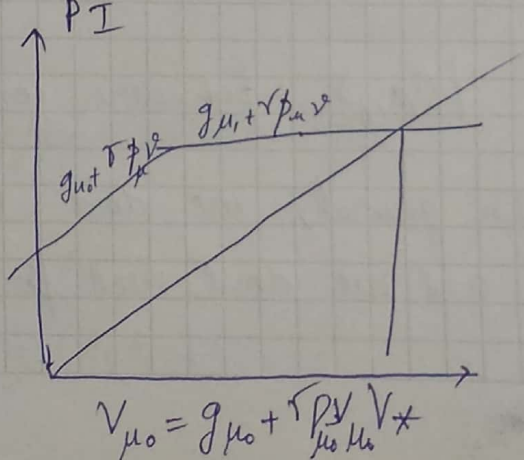
Output $V(s)\ \forall s$ and deterministic policy π

s:t $\pi(s) = \operatorname{argmax}_a \sum_{s',r} p(s', r | s, a)\,[r + \gamma V(s')]$

VI vs PI







5 Monte Carlo Methods:

Learning Theory: "Model of System is unknown."

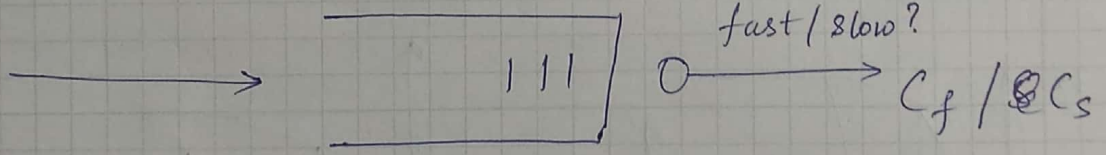
i.e., $p(i,a,j)$ is unknown

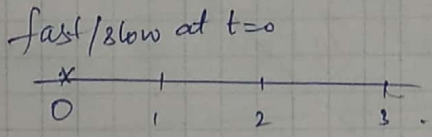
$r(i,a,j)$ is unknown.

but it is possible to Observe next state given current state and action.

Observe immediate random rewards. r_n

We have access to either real data or data from a simulator





p_m = prob of m arrivals in a slot, $m \geq 0$

$p(i,a,j)$ — prob were computed before (prev eg)

In general, we don't have access to transition prob and we dont need p to find $p(i,a,j)$'s

Monte - Carlo Prediction.

Aim is to learn value function for a given policy from sample rewards.

We first consider episodic tasks — ∃ a transition state

two procedures

- first visit MC method
- every visit MC method.

First visit MC prediction

Input : a policy π to be evaluated.

Initialize : $V(s) \in \mathbb{R} \quad \forall s \in S$

$Returns(s) \leftarrow$ an empty list $\forall s \in S$

loop (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 $t = T-1, T-2 \ldots 0$

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

Append G to $Returns(S_t)$

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

Note : T is a random time.

In the first pass, $G = R_T$

Next: $G := \gamma R_T + R_{T-1}$

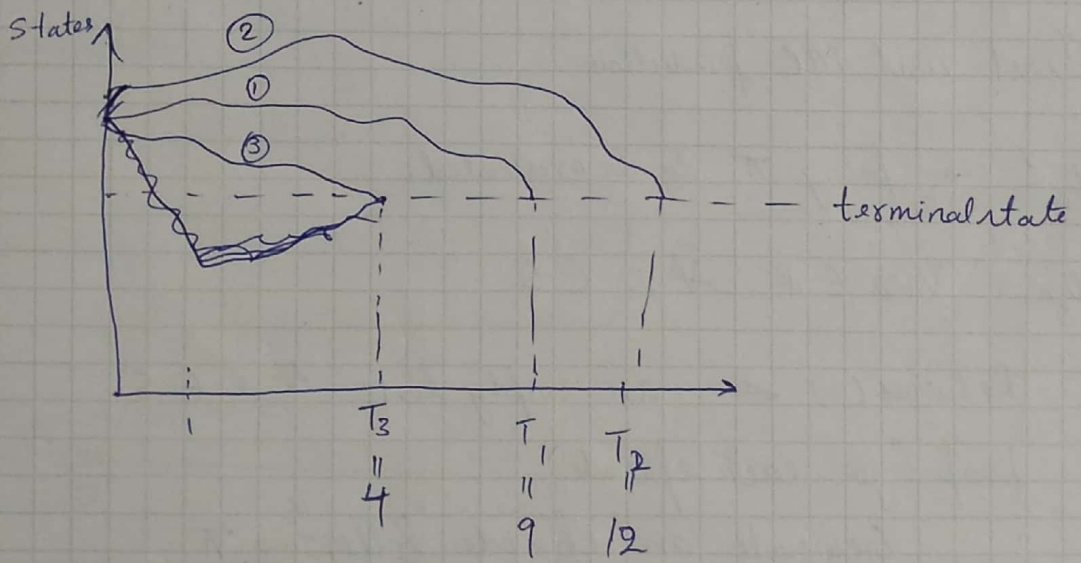
Next: $\gamma(\gamma R_T + R_{T-1}) + R_{T-2}$

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

Eventually,

$$G := R_1 + \gamma R_2 + \gamma^2 R_3 \ldots \gamma^{T-1} R_T$$

$$V_\pi(s) = E_\pi[G_T \mid S_0 = s]$$



path

$G_1 = R_1' + \gamma R_2' + \ldots \gamma^{T_1-1} R_{T_1}'$

$G_2 = R_1^2 + \gamma R_2^2 + \ldots \gamma^{T_2-1} R_{T_2}^2$

$\vdots$

$G_m = R_1^m + \gamma R_2^m \ldots \gamma^{T_m-1} R_{T_m}^M$

$$\frac{1}{M}\sum_{i=1}^{M} G_i \simeq V_\pi(s)$$

~~For~~ A primary goal: Estimate q_* (optimal action-value f)

Consider policy evaluation for given policy π.

Estimate $q_\pi(s,a)$ [expected return starting in state s & picking action a & subsequently picking actions according to π]

M.C methods for

action-value functions are same as for state-value functions except that we now look at state-action tuples (s,a) that are visited.

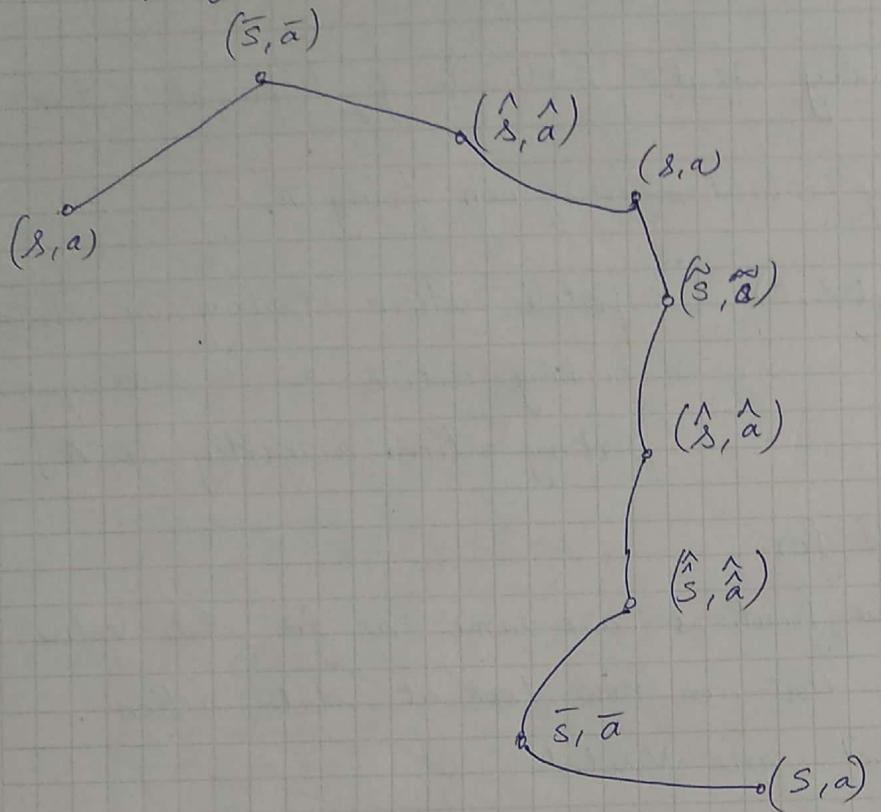
Note: If Π is deterministic, then in each state 's', we pick only the action $\pi(s) \triangleq a$

In such case, we will ~~do~~ not learn values for actions

$b \in A(s) \setminus \{a\}$

$A(s)$ difference a $\quad A(s) \cap \{a\}^c$

Every visit MC Method for action-values. under a policy



Sequence of state-action tuples $\{(X_n, A_n)\}$ where $A_n \simeq \pi(\cdot | S_n)$ forms a markov chain.

Transition prob for (X_n, A_n), $n \geq 0$

$$P(X_{n+1}=j, A_{n+1}=b \mid X_n=i, A_n=a)$$

$$= P(A_{n+1}=b \mid X_{n+1}=j, X_n=i, A_n=a) * P(X_{n+1}=j \mid X_n=i, A_n=a)$$

$$= \pi(b|j)\, p(i,a,j)$$ — i, a doesn't matter for b

$$= \sum_j \underbrace{\sum_b \pi(b|j)}_{1}\, p(i,a,j) = \sum_j 1 \cdot p(i,a,j)$$

$$= 1$$