REINFORCEMENT LEARNING :

- RL is a trial and error learning but is based on long-term rewards.

- MCTS + UCT was used in Alpha zero v/s Stock fish chess engine

MOTE-HALL PROBLEM :

- Course Evaluation :
  1) Mid sem : 25
  2) End sem : 25
  3) 2 Quizzes : 30%
  4) Mini-project : 20%

$$P\{e_1, e_2\} = P\{e_1\}\, P\{e_2/e_1\} \quad (1)$$
$$= (Y_3)$$

$$P\{e_2 e_1\} = P\{e_2\}\, P\{e_1/e_2\}$$
$$= Y_3$$

$$P\{T e_1\} = P\{T\}\, P\{e_1/T\}$$
$$= (Y_3) \quad (Y_2)$$
$$= Y_6$$

$$P\{T e_2\} = P\{T\}\, P\{e_2/T\}$$
$$= Y_6$$

R.V:

- $\mathbb{E}[X]$ is like centre of mass.
- $\text{var}[X]$ is like moment of inertia.

. RV is a fun. from outcome space to $\mathbb{R}$. (has a dist^n associated with it)

- If $U \sim \text{uniform}[0,1]$ then,

$$W = -\ln(1-U) \sim \exp$$

.

23 (7/1):

LAW OF LARGE NUMERS:

- let $X_1, X_2, \ldots, X_n$ be IID R.V's
- Then $S_n = \dfrac{\sum_{i=1}^{n} X_i}{n} \longrightarrow \mathbb{E}[X]$

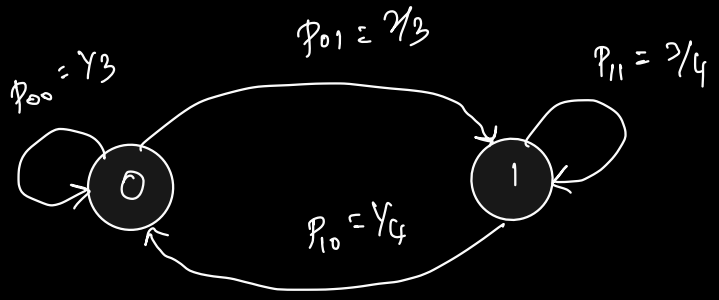
MARKOV CHAIN:

- Consider; $\{X_n\}_{n \geq 0}$ } seq^n of RV's.

def^n:

- $\mathbb{P}\{X_n = x \mid X_{n-1} \ldots X_1\} = \mathbb{P}\{X_n = x \mid X_{n-1}\}$

- IID sequence is a trivial Markov chain


$P_{00} = 1/3$
$P_{01} = 2/3$
$P_{11} = 3/4$
$P_{10} = 1/4$
0
1

24:

Markov Chain:

Characteristics of MC:

- Initial distribution : $\pi_0(x) = P\{X_0 = x\}$
- Transition probabilities:

$$P\{X_{n+1} = j \mid X_n = i\} = P_{ij}$$

- Computing the probability of being in state 'i' at time 'n':

WKT, $[P^n]_{ij} = P\{X_n = j \mid X_1 = i\}$

(P^n → PTM)

$$\Rightarrow \pi_n^T = \pi_0^T P^n$$

- Let N denote # of tosses before two consecutive heads.
- $N \in \{2, 3, \ldots, \infty\}$

$$E[N] = \sum_{n=1}^{\infty} n \, P\{N = n\}$$

$$\mathbb{P}\{N=2\} = 1/4 \qquad \{HH\}$$

$$\mathbb{P}\{N=3\} = 1/8 \qquad \{THH\}$$

$$\mathbb{P}\{N=4\} = 1/8 \qquad \{HTHH,\ TTHH\}$$

Tedious Job

Alternative way:

- $\mathbb{E}[N] = \mathbb{E}[N \mid \text{first toss} = T]\ \mathbb{P}\{\text{First toss} = T\}$
  $+ \mathbb{E}[N \mid \text{First toss} = H]\ \mathbb{P}\{\text{First toss} = H\}$

- $\mathbb{E}[N \mid \text{First toss} = T] = \sum_{n=2}^{\infty} n\, \mathbb{P}\{n \mid 1^{st} \text{ toss} = T\}$

$$= \sum_{n=3}^{\infty} n\, \mathbb{P}\{n \mid 1^{st} \text{ toss} = T\}$$

$k = n-1$

$$= \sum_{k=2}^{\infty} (k+1)\, \mathbb{P}\{k+1 \mid 1^{st} \text{ toss} = T\}$$

$$= \sum_{k=2}^{\infty} k\, \mathbb{P}\{k+1 \mid 1^{st} \text{ toss} = T\} + \sum_{k=2}^{\infty} \mathbb{P}\{k+1 \mid 1^{st} \text{ toss} = T\}$$

$$\mathbb{P}\{k+1 \mid 1^{st} \text{ toss} = T\} = \mathbb{P}\{k\}$$

$$= \sum_{k=2}^{\infty} k\, \mathbb{P}\{k\} + 1$$

$$= \mathbb{E}[N] + 1$$

$$\mathbb{E}[N \mid 1^{st} \text{ toss} = H] = \mathbb{E}[N \mid 1^{st} \text{ toss} = H, 2^{nd} \text{ toss} = H] \cdot \mathbb{P}\{2^{nd} \text{ toss} = H \mid 1^{st} \text{ toss} = H\}$$

$$+ \mathbb{E}[N \mid 1^{st} \text{ toss} = H, 2^{nd} \text{ toss} = T] \cdot \mathbb{P}\{2^{nd} \text{ toss} = T \mid 1^{st} \text{ toss} = H\}$$

$$= 2(1/2) + (\mathbb{E}[N] + 2)(1/2)$$

$$= 1 + (\mathbb{E}[N] + 2)\,1/2$$

$$= \frac{\mathbb{E}[N]}{2} + 2$$

$$\Rightarrow \mathbb{E}[N] = (\mathbb{E}[N] + 1)\,1/2 + \left(\frac{\mathbb{E}[N]}{2} + 2\right)1/2$$

$$\mathbb{E}[N] = \mathbb{E}(N)\left(\frac{3}{4}\right) + 3/2$$

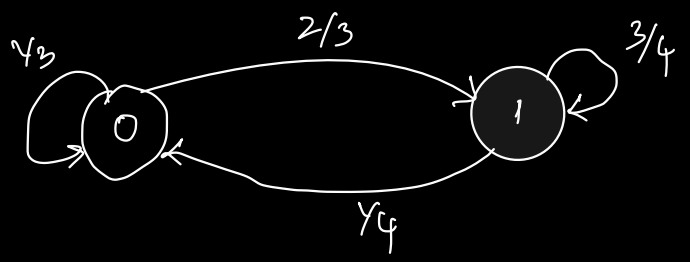
$$\mathbb{E}[N]\left(\frac{1}{4}\right) = 3 \times 2$$

$$\mathbb{E}[N] = 6$$

STATIONARY DISTRIBUTION :

$$\pi_n^T P = \pi_{n+1}^T$$

$$\pi^T P = \pi^T$$ } Eigen vector with eigen value = 1



$$\begin{bmatrix} x & y \end{bmatrix} \begin{pmatrix} 1/3 & 2/3 \\ 1/4 & 3/4 \end{pmatrix} = \begin{bmatrix} x & y \end{bmatrix}$$

$$\frac{x}{3} + \frac{y}{4} = x$$

$$\frac{2}{3}x + \frac{3y}{4} = y$$

$$4x + 3y = 12x$$

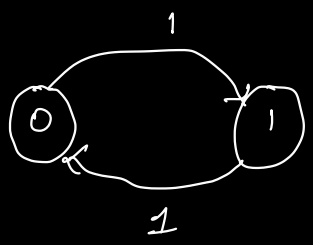
$$8x + 9y = 12x$$

$$-8x + 3y = 0$$

$$x + y = 1$$

$$-7x + 4y = 1$$

° Ex :



BANDITS : (Stateless RL)

12/1 :

. We look at Bernoulli Bandits with 2 arms and then we generalize it.

. For two arms, action space $A = \{1, 2\}$

. Let μ_i denote the avg. reward of arm i.

. Let reward at step 't' is R_t and A_t be the action at time 't'.

. Our goal is to max $\sum_{t=1}^{T} R_i$.

But here we don't have the expectation. So it's like a realization.

. We choose A_t's such that avg. reward is maximized i.e max $\mathbb{E}\left[\sum_{t=1}^{T} R_i\right]$

. Notion of regret :

Regret = Max. avg. reward that can be obtained till T
- Total expected reward obtained by an algorithm.

- Let optimal arm be i^* such that $\mu^* = \max\{\mu_1, \mu_2\}$

$$\Rightarrow \text{Regret} = \mu^* T - \underbrace{\mathbb{E}}_{\downarrow}\left[\sum_{i=1}^{T} R_i\right]$$

Randomness w.r.t reward dist^n. and also the arm selection

- Let N_1 : # of times arm 1 is pulled
  N_2 : # of times arm 2 is pulled

} Expected # of times

$$\Rightarrow \mathbb{E}\left[\sum_{i=1}^{T} R_i\right] = \mu_1 N_1 + \mu_2 N_2$$

$$\Rightarrow R_T = \mu^* \underbrace{T}_{N_1+N_2} - (\mu_1 N_1 + \mu_2 N_2)$$

$$R_T = (\mu^* - \mu_1) N_1 + (\mu^* - \mu_2) N_2$$

$$R_T = \Delta_1 N_1 + \Delta_2 N_2$$

Intuitive understanding

$\Rightarrow$ Regret $\propto$ Avg. # of pulls of a sub-optimal arm

Greedy Policy:

- For $t=1$ pull arm 1
- For $t=2$ pull arm 2
- For $t \geq 2$ pull the arm with highest estimated avg-reward.
- The estimates, $\hat{\mu}_1 = \frac{R_1^{(1)} + R_2^{(2)} + \dots + R^{(n_1)}}{n_1}$

$$\hat{\mu}_2 = \frac{R_1^{(2)} + R_2^{(2)} + \dots + R^{(n_2)}}{n_2}$$

- Regret $\geq O(T)$

- To see why regret is lower-bounded by $O(T)$ assume arm-1 is optimal. Now to bound the regret we need to find a bound to the # of times sub-optimal arm is pulled.

$$\mathbb{E}[n_2] = \mathbb{E}\left[n_2 \mid \underset{\downarrow t=1}{0}, \underset{\downarrow t=2}{1}\right] \mathbb{P}[0,1] + \mathbb{E}[n_2 \mid 0,0]\, \mathbb{P}[0,0]$$

$$+ \mathbb{E}[n_2 \mid 1,0]\, \mathbb{P}\{1,0\} + \mathbb{E}[n_2 \mid 1,1]\, \mathbb{P}\{1,1\}$$

By assuming we only exploit $\forall\ t \geq 3$

$$\geq \mathbb{E}[n_2 \mid 0,1]\, \mathbb{P}\{0,1\}$$

$$\geq (T-2)(1-\mu_1)\mu_2$$

$$\mathbb{E}[n_2] \geq O(T)$$

- Even if we explore for T_e steps regret for greedy policy will be of order (T).

ϵ-GREEDY POLICY:

- Algorithm: 1) with $(1-\epsilon)$ follow greedy policy
  2) with (ϵ) prob. choose the arm randomly.

- Regret in this case, $R_T \geq \frac{\epsilon}{2} T$

  $\because$ with atleast 'ϵ' prob. we take random action. Hence pulling suboptimal arm is of probability $(\epsilon/2)$.

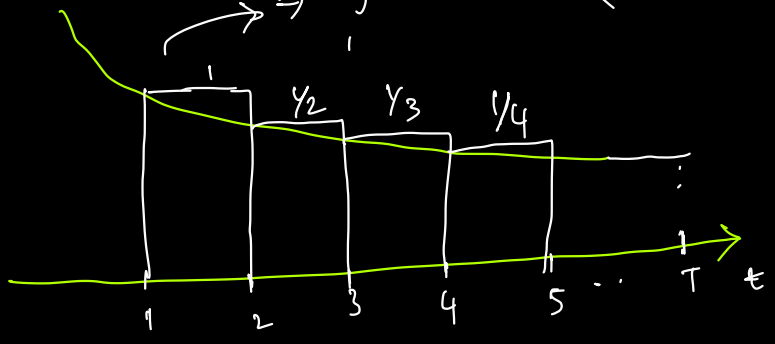
- Regret $\geq O(T)$

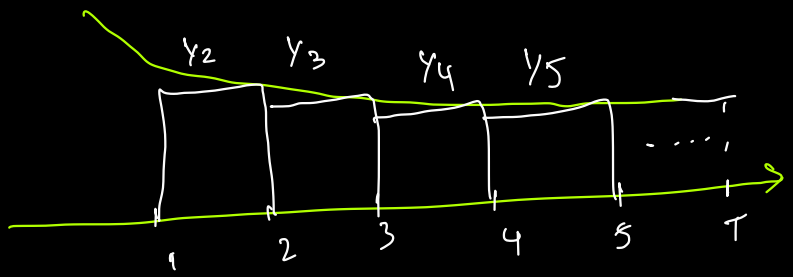
VARIABLE EPSILON GREEDY:

. Instead of fixing 'ε' what if we decrease at each time step? i.e., $\varepsilon_t = \frac{\varepsilon}{t}$

$$\Rightarrow \mathbb{E}[n_2] \geq \varepsilon\left(1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{T}\right)$$

$$\ln(T) = \int_1^T \frac{1}{t}\,dt \leq \left(1 + Y_2 + Y_3 + \cdots + Y_T\right)$$





$$\ln(T) = \int_1^T \frac{1}{t}\,dt \geq \left(1/2 + Y_3 + \cdots + Y_T\right)$$

. We can show that,

$$\text{Regret} \leq O(\ln T)$$

Not straight forward to proove. Need to use concentration inequalities.

We are lower-bounding $\mathbb{E}[n_2]$ with $(\cdot)$. Then we upperbounded $(\cdot)$ by $\ln(T)$. Then how is Regret upper bounded by $O(\ln T)$?

- Refer Auer et. al.

. Markov Chain:

. Intro. to prob. models
- Sheldon Ross

. Bandits:

. Bandit Algorithms
- Tor and Czaba

. $f(x)$ goes asymptotically as $g(x)$ means the rate at which $f(x)$ changes is same as the rate of change in $g(x)$ as $x \to \infty$.

- Prove that Regret of variable ε-greedy has regret of $O(\ln T)$

SOFTMAX STRATEGY :

. We pick each arm with certain probability (i.e action isn't deterministic at a time step)

. The prob. of picking arm 1 at time step 't' is,

$$\mathbb{P}\{a_t = 1\} = \frac{\exp\{\hat{\mu}_1(t)\}}{\sum_{i=1}^{2} \exp\{\mu_i(t)\}}$$

$$\mathbb{P}\{a_t = 2\} = \frac{\exp\{\hat{\mu}_2(t)\}}{\sum_{i=1}^{2} \exp\{\mu_i(t)\}}$$

. This has Regret of $O(T)$.

. A modification of this is softmax policy with hyperparameter called as temperature.

i.e $$\mathbb{P}\{a_t = 1\} = \frac{\exp\{\hat{\mu}_1(t)/T\}}{\sum_{i=1}^{2} \exp\{\hat{\mu}_i(t)/T\}}$$

. For high 'T' the action would be uniformly random and for small 'T' the policy would be greedy.

14/1 :

RECAP :

. The expected # of times an arm is pulled till time 'T' is ;

$$\mathbb{E}[n_i(T)] = \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{A_t = i\}\right]$$

$$= \sum_{t=1}^{T} \mathbb{E}\left[\mathbb{1}\{A_t = i\}\right]$$

$$= \sum_{t=1}^{T} \mathbb{P}\{A_t = i\}$$

. For ε-greedy : $\mathbb{P}\{A_t = i\} \geq \varepsilon/2$

. For variable ε-greedy : $\mathbb{P}\{A_t = i\} \geq \frac{\varepsilon}{t}$ ← Initial value

$$\Rightarrow \mathbb{E}[n_2(T)] \geq \varepsilon \ln T$$

. For K-arms, regret can be written as ;

$$\text{Regret} = \mu^* T - \mathbb{E}\left(\sum_{k=1}^{K} \mu_k \left(\sum_{t=1}^{T} \mathbb{1}\{A_t = k\}\right)\right)$$

$$= \mu^* T - \sum_{k=1}^{K} \mu_k \underbrace{\mathbb{E}[n_k]}_{\text{Expected \# of times arm 'k' is played.}}$$

• Without loss of generality,

$$\text{Regret} \leq \max_{k \in [K]} (\mu^* - \mu_k) \sum_{k=2}^{K} \mathbb{E}[n_k]$$

Assuming 1st arm is the optimal arm

• Generalizing the algorithms discussed for [K] arms :

GREEDY ALGORITHM :

• The estimate of mean reward for arm 'k' at round 't' is,

$$\hat{\mu}_k(t) = \frac{\sum_{\tau=1}^{t} R_k(\tau) \, \mathbb{1}\{A_\tau = k\}}{\sum_{\tau=1}^{t} \mathbb{1}\{A_\tau = k\}}$$

• Algorithm : 1) Play each arm once } Explore

2) Then play the arm with highest empirical estimate. } Exploit

$$A_{t+1} = \underset{i \in \{1,K\}}{\operatorname{argmax}} \hat{\mu}_i(t)$$

ε - GREEDY :

- For $t = 1, 2, \ldots, T$,

$$A_t = \begin{cases} \text{greedy Rule} & \text{w.p } (1-\varepsilon) \\ \text{Arm 'k'} & \text{w.p } \frac{\varepsilon}{K} \end{cases}$$

VARIABLE ε - GREEDY :

- For $t = 1, 2, \ldots, T$

$$A_t = \begin{cases} \text{greedy Rule} & \text{w.p } (1-\varepsilon_t) \\ \text{Arm 'k'} & \text{w.p } \frac{\varepsilon_t}{K} \end{cases}$$

SOFTMAX :

- For $t = 1, 2, \ldots, T$
- Pick arm 'k' w.p ,

$$\mathbb{P}\{A_t = k\} = \frac{e^{\hat{\mu}_k(t)/\text{Temp}}}{\sum_{i=1}^{K} e^{\hat{\mu}_i(t)/\text{Temp}}}$$

REGRET BOUNDS FOR THE ALGORITHMS :

1) Greedy : $R \geq O(T)$

$\Rightarrow R = O(T)$

2) ε - Greedy : $R \geq O(T)$

$\Rightarrow R = O(T)$

3) ε_t - greedy : $R \geq O(\ln T)$

. We can show that ε_t has regret, $R \leq O(\ln T)$

Refer to Auer. et. al

UPPER CONFIDENCE BOUND (UCB) :

$$P\{|\hat{\mu}_k - \mu| > \varepsilon\} \leq \frac{2e^{-2\varepsilon^2}}{\delta}$$

. $A_t = \arg\max_k \{\hat{\mu}_k + b_k\}_{k=1}^{K}$

↓

Bonus term

Here, $b_k(t) = \sqrt{\frac{2\ln t}{n_k(t)}}$

$\ln(t)$

$n_k(t)$

$\ln(t)$

. Algorithm :

1) Give 1 chance to every arm

2) For $t = K+1, \ldots, T$ pick the arm that has highest UCB i.e.,

$$A_t = \arg\max_k \{\hat{\mu}_k(t) + b_k(t)\}_{k=1}^{K}$$

. If $\sqrt{\frac{\ln(t)}{n_k(t)}}$ $\longrightarrow$ very small value as $t \to$ very large value and $n_k(t) \to$ very large value

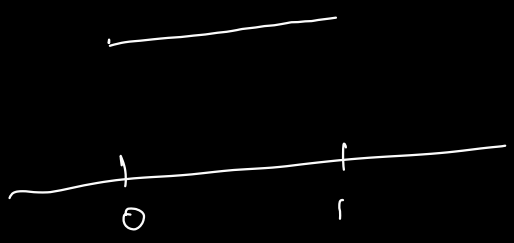
NOTE :

BAYESIAN ESTIMATION :

- Let μ_K be the unknown R.V.
- We assume prior dist^n for μ_K.
- When the Reward follows the Bernoulli distribution we assume the prior dist^n. for μ_K to be β-distribution. i.e $\mu_K \sim \beta(1,1)$

β - DISTRIBUTION :

- $\beta(1,1)$:



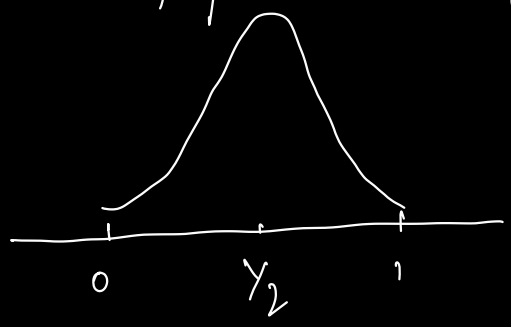
- The PDF of β-dis^n. is ;

$$f_x(x) = \frac{1}{\left(\frac{\Gamma(\alpha)\,\Gamma(\beta)}{\Gamma(\alpha+\beta)}\right)} x^{\alpha-1}(1-x)^{\beta-1} = B(\alpha,\beta) = \int_0^1 x^{\alpha-1}(1-x)^{\beta-1}\,dx$$

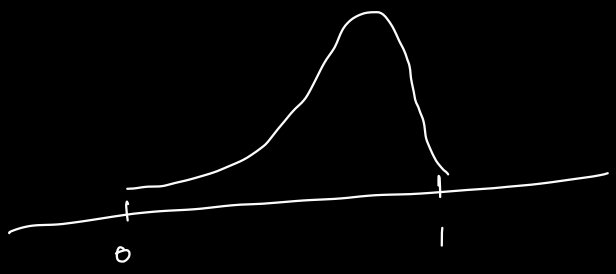
where, $\Gamma(z) = \int_0^\infty x^{z-1} e^{-x}\,dx$

$$E[x] = \frac{B(\alpha+1,\beta)}{B(\alpha,\beta)} = \alpha/\alpha+\beta$$

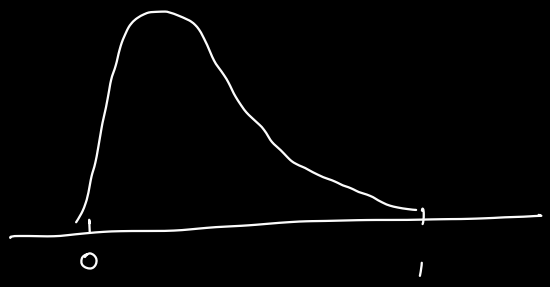
. $\alpha = 1000$, $\beta = 1000$: (Symmetric)



. $\alpha = 10,000$, $\beta = 100$



. $\alpha = 100$, $\beta = 10,000$



17/1 :

THOMPSON SAMPLING :

. Consider 2-arm case.

. Assumption : The mean-reward of each arm is a random variable.

. When rewards follow Bernoulli distribution we assume the true rewards come from β-distribution.

- I/p : $[K] = \{1, 2\}$, μ_1, μ_2
- Initialisation : $\mu_1 \sim \beta(1,1)$, $\mu_2 \sim \beta(1,1)$
- for $t = 1, 2, \ldots T$
  - sample x_1, x_2 from their respective dist^n.
  - $A_t = \underset{i}{\arg\max} \{x_i\}_{i=1}^{K}$
  - Update the dist^n of the arm played

    i.e $\mu_{A_t} \sim \beta\left(\alpha_{t-1} + R_t^{A_t}, \beta_{t-1} + 1 - R_t^{A_t}\right)$

NOTE :

. β- Distribution is a conjugate prior for Bernoulli distribution.

. Consider a coin with unknown bias 'p'.

. Assumption : Prior dist^n of $p \sim \beta(1,1)$

. Given random tosses outcomes $\{x_1, x_2, \ldots, x_n\}$ estimate p.

. PDF → not PMF

$$P\{D \mid p=\theta\} = \prod_{i=1}^{n} \theta^{x_i} (1-\theta)^{1-x_i}$$

Likelihood

$$\Rightarrow \underbrace{P\{p=\theta \mid D\}}_{\text{Posterior dist}^n} \propto P\{D \mid p=\theta\} P\{p=\theta\}$$

$$\propto \theta^{\sum x_i} (1-\theta)^{n-\sum x_i} \; \theta^{\alpha-1} (1-\theta)^{\beta-1}$$

$$\propto \theta^{\alpha+\sum x_i-1} (1-\theta)^{\beta+n-\sum x_i-1}$$

$$\Rightarrow \text{PDF}\{p=\theta \mid D\} \sim \beta\left(\alpha+\sum x_i, \beta+n-\sum x_i\right)$$

INVERSE TRANSFORM METHOD :

- Let X be the R.V which follows a dist^n and we need to find a way to sample it from that distribution given that we only use samples from uniform distribution.

- Let $F_X(x)$ be the CDF of X.

- Let $U \sim Unif[0,1]$

$$F_X(x) = P\{X \le x\}$$

$$= P\{T(U) \le x\}$$

$$= P\{U \le T^{-1}(x)\}$$

$$F_X(x) = T^{-1}(x)$$

$$\Rightarrow T^{-1} = F_X$$

$$\Rightarrow \boxed{X = F_X^{-1}(U)}$$

$$F_X(x) = 1 - e^{-x} = u$$

$$T(u) = x$$

$$\Rightarrow T^{-1}(x) = u$$

$$x = F_X^{-1}(u)$$

- Let $y \sim \exp(\lambda)$:

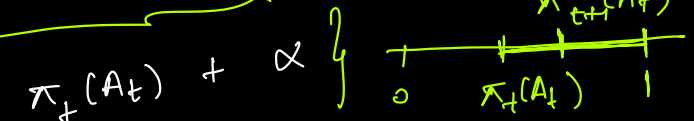
$$y = \frac{-1}{\lambda} \ln(1-U) \qquad \because F_X(x) = 1 - e^{-\lambda x}$$

POLICY BASED BANDIT ALGORITHM :

- At each round 't' we sample arm according to π_t (Prob. dist^n) and update π as,
  - If $R_t = 1$:

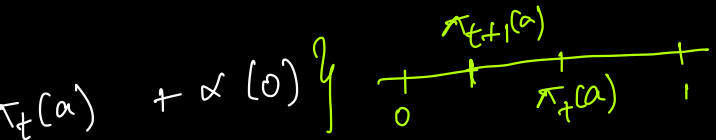
$$\pi_{t+1}(A_t) = (1-\alpha)\,\pi_t(A_t) + \alpha \{1\}$$

($\{1\}$: convex comb^n. of $\pi_t(A_t)$ and 1)



$$\pi_{t+1}(a) = (1-\alpha)\,\pi_t(a) + \alpha(0) \{0\}$$

(a : other arm)



  - If $R_t = 0$:

$$\pi_{t+1}(A_t) = (1-\beta)\,\pi_t(A_t) + \beta \cdot 0 = \pi_t(A_t) + \beta(0 - \pi_t(A_t))$$

$$\pi_{t+1}(a) = (1-\beta)\,\pi_t(a) + \beta \cdot 0 = \pi_t(a) + \beta(1 - \pi_t(a))$$

(β : step-size)

. When $\alpha = \beta$, the algorithm is called Linear reward and penalty (L_{RP}).

. When $\alpha >>> \beta$, the algorithm is L_{R-EP}.

. When to use L_{R-EP}? L_{R-I}, L_{RP}.

. When $\beta = 0$, L_{R-I} : Linear reward inaction.

The update rule for prob. of arm pulled is proportional to reward in linear fashion. (Only when reward is 1 (in-action))

REINFORCE ALGORITHM :

. Here, the prob. of picking an arm at any time 't' is parameterized by parameter 'θ'.

. simple parameterization for two-arm case is;

$$\pi_t(1, \theta) = \theta$$

$$\pi_t(2, \theta) = 1 - \theta$$

. Let $\rho_t(\theta)$ be the average reward obtained by a policy π_θ at time t,

$$\Rightarrow \quad \rho(\theta) = \mu_1 \pi(1, \theta) + \mu_2 \pi(2, \theta)$$

$$\Rightarrow \nabla_\theta J(\theta) = \mu_1 \nabla_\theta \pi(1,\theta) + \mu_2 \nabla_\theta \pi(2,\theta)$$

$$= \mu_1 \frac{1}{\pi(1,\theta)} \nabla_\theta \pi(1,\theta) \pi(1,\theta) + \mu_2 \frac{1}{\pi(2,\theta)} \nabla_\theta \pi(2,\theta) \pi(2,\theta)$$

$$= \mu_1 \nabla_\theta \ln(\pi(1,\theta)) \pi(1,\theta) + \mu_2 \nabla_\theta \ln(\pi(2,\theta)) \pi(2,\theta)$$

$$= \mu_1 \mathbb{E}\left[\nabla_\theta \ln(\pi(1,\theta))\right] + \mu_2 \mathbb{E}\left[\nabla_\theta \ln[\pi(2,\theta)]\right]$$

$$\nabla_\theta J(\theta) = \mathop{\mathbb{E}}_{a \sim \pi_\theta}\left[R^{(a)} \nabla_\theta \ln(\pi(a,\theta))\right] = \mathop{\mathbb{E}}_{a \sim \pi_\theta}\left[\mu_a \nabla_\theta \ln[\pi(a,\theta)]\right]$$

- Update Rule :

$$\theta_{n+1} = \theta_n + \alpha \nabla_\theta J(\theta_n)$$

- The estimate of $\nabla_\theta J(\theta)$ using the samples is given by,

$$\nabla_\theta \hat{J}(\theta) = \frac{1}{n} \sum_{t=1}^{n} R_t^{(a_t)} \nabla_\theta \ln(\pi(a_t,\theta))$$

$$= \frac{1}{n}\left[\sum_{i=1}^{n_1} R_i^{(1)} \nabla_\theta \ln(\pi(1,\theta)) + \sum_{i=1}^{n_2} R_i^{(2)} \nabla_\theta \ln[\pi(2,\theta)]\right]$$

$(n_1 + n_2 = n)$

$$= \sum_{i=1}^{n_1} \frac{R_i^{(1)}}{n_1} \cdot \frac{n_1}{n} \nabla_\theta \ln(\pi(1,\theta))$$

$$+ \sum_{i=1}^{n_2} \frac{R_i^{(2)}}{n_2} \cdot \frac{n_2}{n} \nabla_\theta \ln(\pi(2,\theta))$$

$\frac{n_2}{n_2} \xrightarrow{n\to\infty} \mu_2$; $\frac{n_2}{n} \xrightarrow{n\to\infty} \pi(2,\theta)$

$$\nabla_\theta \hat{J}(\theta) = \mu_1 \pi(1,\theta) \nabla \ln \pi(1,\theta)$$

$$+ \mu_2 \pi(2,\theta) \nabla \ln \pi(2,\theta)$$

$$=$$

- The stochastic gradient Ascent :

$$\theta_{n+1} = \theta_n + \alpha R_i^{(a)} \nabla \ln \pi(a,\theta)$$

REINFORCE ALGORITHM WITH BASELINE :

- The update rule is,

$$\theta_{n+1} = \theta_n + \alpha (R^{(a)} - b_n) \nabla \ln \pi(a,\theta)$$

where, $b_n = \frac{1}{n} \sum_{i=1}^{n} R_n$

- $\theta_K - \ln[\Sigma \exp\{\theta_j\}]$

- $1 - \underbrace{\frac{\exp\{\theta_K\}}{(\quad)}}_{\pi_K}$

.

2/1 :

REINFORCE ALGORITHM FOR K-ARMS :

- For arm K,

$$\pi(K, \theta) = \frac{\exp\{\theta_K\}}{\sum_{j=1}^{K} \exp\{\theta_j\}}$$

- Update Rule :

$$\theta_{n+1} = \theta_n + \nabla \rho(\theta)$$

CONCENTRATION INEQUALITY :

MARKOV INEQUALITY :

- For a non-negative R.V, X

$$\mathbb{P}\{X > a\} \leq \frac{\mathbb{E}X}{a}$$

CHEBYSHEV'S INEQUALITY :

- For a R.V with finite mean and variance,

$$\mathbb{P}\{|X - \mu| > \varepsilon\} \leq \frac{Var(X)}{\varepsilon^2}$$

WLLN :

. A dist. is heavy tail if M.G.F $= \infty$

$\mathbb{E}[e^{\lambda X}]$ Exponential dist. is heavy tail

. Let $X_1, X_2, \ldots, X_n$ be i.i.d R.V's

then $\quad \mathbb{P}\left\{\left|\frac{1}{n}\sum_{i=1}^{n} X_i - \mu\right| > \varepsilon\right\} \xrightarrow{n\to\infty} 0 \qquad (\forall\ \lambda)$

Proof: From Chebyshev's inequality,

$$\mathbb{P}\left\{\left|\frac{1}{n}\sum_{i=1}^{n} X_i - \mu\right| > \varepsilon\right\} \leq \frac{\sigma^2}{n\varepsilon^2}$$

CHERNOFF HOEFFDING INEQUALITY :

. Let $X_1, X_2, \ldots, X_n$ be i.i.d R.V with mean μ.

Then, $\quad \mathbb{P}\{|S_n - \mu| > \varepsilon\} \leq 2\exp\{-n\varepsilon^2\}$

UCB ALGORITHM :

. For $t = 1, \ldots K$ play a_K at round 'k'

. For $t = K, \ldots, T$

$$A_t = \arg\max_k \hat{\mu}_{k,t-1} + \sqrt{\frac{2\log t}{n_{k,t-1}}}$$

$\hat{\mu}_{k,t-1} + \sqrt{\frac{2\log t}{n_{k,t-1}}}$: $UCB_{k,t-1}$

. Claim: $R_T(UCB) \leq O(\log T)$

Proof:

$$R_T(UCB) = \mathbb{E}\left[\sum_{t=1}^{T} \mu^* - \mu_{I_t}\right]$$

(I_t: Arm played at round t)

$$= \sum_{k=1}^{K} \mathbb{E}[T_k(n)]\, \Delta_k$$

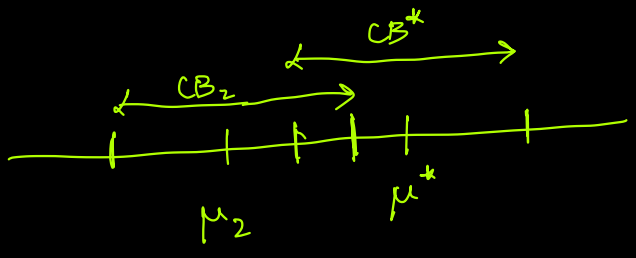
($\mathbb{E}[T_k(n)]$: Expected # of times we play arm 'k')

. Let $n_i(T)$ be the # of times arm 'i' is picked.

$$\Rightarrow n_i(T) = \sum_{t=1}^{T} \mathbb{1}\{A_t = i\}$$

. At time 't+1' if we play arm 'k'

$$\hat{\mu}_t^* + \sqrt{\frac{2\ln t}{n_{*,t}}} < \hat{\mu}_k + \sqrt{\frac{2\ln t}{n_{k,t}}}$$



. For the above inequality to hold true one of the following cases should be true :

case 1 : $\hat{\mu}_* < \mu_* - \sqrt{\frac{2 \ln t}{n_{*,t}}}$ (Underestimate optimal arm)

case 2 : $\hat{\mu}_k > \mu_k + \sqrt{\frac{2 \ln t}{n_{k,t}}}$ (Overestimate of sub-optimal arm)

case 3 : $n_i < \frac{8 \ln T}{\Delta_i^2}$ (Less exploration of arm i)

Claim :

. Suppose none of the above cases are true, then i^{th} arm will n't get picked at time $t+1$.

Proof :

$$\hat{\mu}_* \geq \mu_* - \sqrt{\frac{2 \ln t}{n_{*,t}}} \quad \text{(case 1 not true)}$$

$$\hat{\mu}_* + \sqrt{\frac{2 \ln t}{n_{*,t}}} \geq \mu_*$$

$$\geq \mu_* - \mu_i + \mu_i$$

$$\geq \Delta_i + \mu_i$$

$$\geq \sqrt{\frac{8 \ln t}{n_{i,t}}} + \mu_i \qquad \text{(Case 3 not tone)}$$

$$\geq \mu_i + \sqrt{\frac{2 \ln t}{n_{i,t}}} + \sqrt{\frac{2 \ln t}{n_{i,t}}}$$

$$\underbrace{\hat{\mu}_* + \sqrt{\frac{2 \ln t}{n_{*,t}}}}_{UCB_{*,t}} \geq \hat{\mu}_i + \sqrt{\frac{2 \ln t}{n_{i,t}}} \qquad \text{(Case 2 not tone)}$$

$$\geq UCB_{i,t}$$

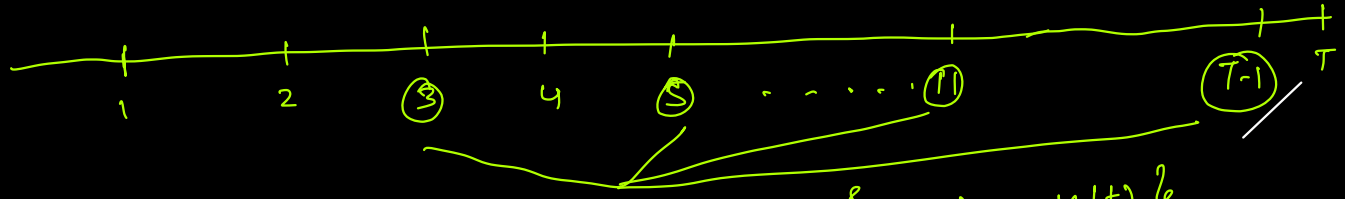
- Now consider,

$$\mathbb{E}[n_i(T)] = \mathbb{E}\left[\sum_{i=1}^{n} \mathbb{1}\{A_t = i\}\right]$$

splitting the action taken at each time on a event, $\{n_i(t) < u(t)\}$

$$= \underbrace{\mathbb{E}\left[\sum_{i=1}^{T} \mathbb{I}\{A_t = i,\ n_i(t) < u(T)\}\right]}_{①} + \mathbb{E}\left[\sum_{i=1}^{T} \mathbb{I}\{A_t = i,\ n_i(t) \geq u(T)\}\right]$$

$$u(t) = \frac{8 \ln t}{\Delta_i^2}$$



At these times $\{n_i(t) \geq u(t)\}$ and remaining times $\{n_i(t) < u(t)\}$

- WKT if $n_i(t) < \frac{8 \ln T}{\Delta_i^2} = u(T)$

then ① $\leq \frac{8 \ln T}{\Delta_i^2}$

Also $\forall$ 't' such that $n_i(t) > \frac{8 \ln T}{\Delta_i^2}$

$$n_i(t) > \frac{8 \ln T}{\Delta_i^2} > \frac{8 \ln t}{\Delta_i^2}$$

$$\Rightarrow \mathbb{E}[n_i(T)] \leq \frac{8 \ln T}{\Delta_i^2} + \sum_{t=1}^{T} \mathbb{P}\{A_t = i, n_i(t) \geq u(t)\}$$

only possible either due to case 1 or 2.

$$\leq \frac{8 \ln T}{\Delta_i^2} + \sum_{t=1}^{T} \mathbb{P}\{\text{case 1 or case 2 occurs}\}$$

$$\leq \frac{8 \ln T}{\Delta_i^2} + \sum_{t=1}^{T} \mathbb{P}\{\text{case 1 occurs}\}$$

$$+ \sum_{t=1}^{T} \mathbb{P}\{\text{case 2 occurs}\}$$

- From Hoeffding's inequality,

$$P\{\text{case 1 is done}\} = P\left\{\hat{\mu}_{*} \le \mu_{*} - \sqrt{\frac{2\ln t}{n_{*,t}}}\right\}$$

$$\le \exp\left\{-n_{*,t} \cdot \frac{2\ln t}{n_{*,t}}\right\}$$

$$= 1/t^2$$

- Similarly, $P\{\text{case 2 is done}\} = P\left\{\hat{\mu}_{i} \ge \mu_i + \sqrt{\frac{2\ln t}{n_{i,t}}}\right\}$

$$\le \exp\left\{-n_{i,t} \cdot \frac{2\ln t}{n_{i,t}}\right\}$$

$$= 1/t^2$$

$$\Rightarrow \quad \mathbb{E}[n_i(T)] \le \frac{8\ln T}{\Delta_i^2} + \sum_{t=1}^{T} 2/t^2$$

$$\le \frac{8\ln T}{\Delta_i^2} + \underbrace{\left(1 - 1/T\right)}_{C}$$

$$\Rightarrow \quad \mathbb{E}[R_T(UCB)] \leq \quad O(\ln T)$$

NOTE :

. There is a lower bound on the regret.

. For MAB (Stochastic) no algorithm can incur a regret less than logarithmic regret. . Lai - Robinson

$$\text{i.e} \quad \mathbb{E}[n_i(T)] \geq O(\ln T)$$

PAC OPTIMALITY :

. For a PAC (ε, δ) ,

Input : ε, δ

Output : One arm j such that ,

$$\mathbb{P}\{\mu_j > \mu^* - \varepsilon\} \geq 1 - \delta$$

. In other words , probability of not picking an ε - close arm is $\leq \delta$.

NAIVE ROUND ROBIN ALGORITHM :

Algorithm :

1) Play all arms in a round robin fashion for a certain no. of rounds.

2) Declare the arm with highest empirical estimate as the optimal arm.

NOTE :

. Sample complexity :

How many rounds should we play each arm such that the arm we output is ε-close to the optimal arm with a high-probability.

Claim :

. Naive-Round Robin algorithm is PAC optimal.

. Let $N = \frac{K}{\varepsilon^2} \log\left(\frac{K}{\delta}\right)$. Then the arm that we output will be ε-close to the optimal arm with a very high probability.

. Also, each arm is played $\frac{1}{2\varepsilon^2} \log\left(\frac{2K}{\delta}\right)$ times.

. Let $\{\hat{\mu}_k\}_{k=1}^{K}$ be the estimate of rewards after 'N' rounds. Then choose arm $j = \arg\max_{k \in [K]} \hat{\mu}_k$

. WKT, Picking arm 'j' will be bad if, (In relation with PAC bound.)

$$\mu_j < \mu^* - \varepsilon$$

. We want to bound,

$$\mathbb{P}\{\mu_j < \mu^* - \varepsilon\}$$

. Let 'l' be the arm such that, $\mu_l < \mu^* - \varepsilon$.

. ∴ Probability of picking the l^{th} arm at the end of 'N' rounds is,

$$\mathbb{P}\{\hat{\mu}_l \geq \{\hat{\mu}_1, \hat{\mu}_2, \ldots \hat{\mu}_K\}\}$$

28/1 :

- Quiz 1 : Feb 4
- Assignment 2 : Feb 2 (11:30 AM)

Recap :

- In PAC optimality we are interested in finding the no. of trials required to discover an ε-optimal arm with high probability of $(1-\delta)$.

- i.e determine N such that ;

$$\mathbb{P}\{ \mu_j < \mu_* - \varepsilon \} \leq \delta$$

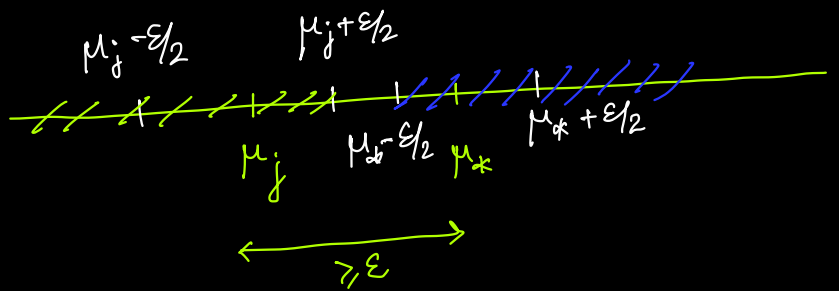
μ_j ↓ True reward of returned 'j'.

- We pick arm 'j' if ;

$$\mathbb{P}\{ \hat{\mu}_j > \hat{\mu}_* \} \leq \mathbb{P}\{ \hat{\mu}_j \geq \mu_j + \varepsilon/2 \text{ or } \hat{\mu}_* \leq \mu_* - \varepsilon/2 \}$$

Explaination for this :

- Suppose not : i.e $\hat{\mu}_j < \mu_j + \varepsilon/2$ and $\hat{\mu}_b > \mu_* - \varepsilon/2$



$$\Rightarrow \quad \hat{\mu}_{a^*} > \hat{\mu}_j$$

$$\cdot \Rightarrow \{\hat{\mu}_j > \hat{\mu}_{a^*}\} = \{\hat{\mu}_j \geq \mu_j + \varepsilon/2\} \cup \{\hat{\mu}_{a^*} \leq \mu_{a^*} - \varepsilon/2\}$$

$$\Rightarrow P\{\hat{\mu}_j > \hat{\mu}_{a^*}\} \leq \underbrace{P\{\hat{\mu}_j \geq \mu_j + \varepsilon/2\}}_{\text{Apply Hoeffding's inequality}} + P\{\hat{\mu}_{a^*} \leq \mu_{a^*} - \varepsilon/2\}$$

$$\leq \exp\{-2N_j\varepsilon^2\} + \exp\{-2N_{a^*}\varepsilon^2\}$$

$$\cdot \text{WKT,} \quad N_j = N_{a^*} = \frac{1}{2\varepsilon^2}\ln\left(2K/\delta\right)$$

$$\Rightarrow P\{\hat{\mu}_j > \hat{\mu}_{a^*}\} \leq 2\exp\{-2\ln(K/\delta)\}$$

$$\leq \delta/K$$

$\Rightarrow$ Total probability of picking the arm 'i' with $\mu_i < \mu_{a^*} - \varepsilon$

i.e., $$P\left\{\bigcup_{i \neq a^*}\{\hat{\mu}_i > \hat{\mu}_{a^*}\}\right\} \leq \sum_{i \neq a^*} \underbrace{P\{\hat{\mu}_i > \hat{\mu}_{a^*}\}}_{\delta/K}$$

$$\leq K \cdot \delta/K$$

$$= \underline{\delta}$$

∴ Naive algorithm is PAC optimal with,

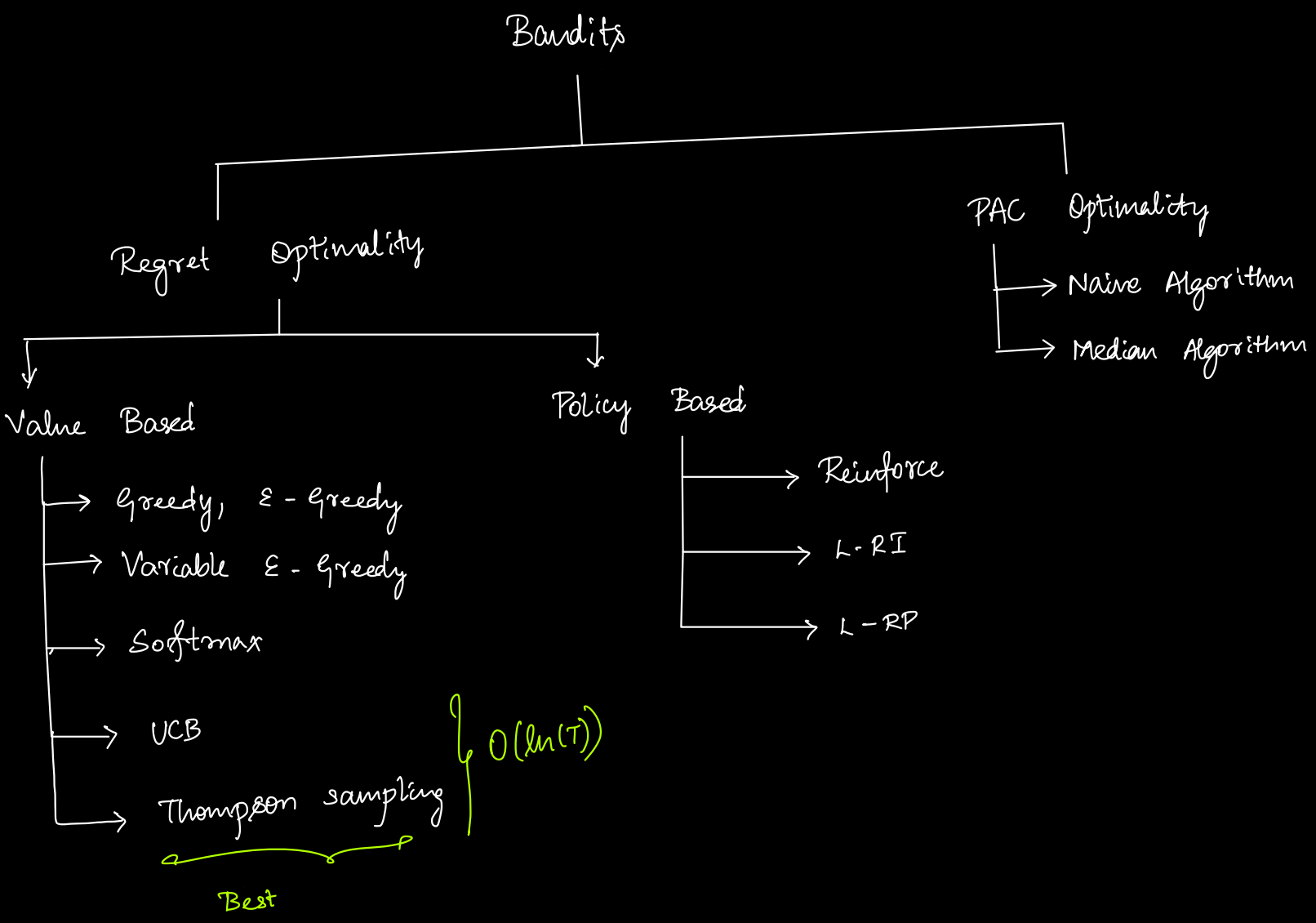
$$N = \frac{K}{2\varepsilon^2} \ln\left(\frac{2K}{\delta}\right)$$

NOTE:

, Median elimination algorithm has better PAC optimality

∴ $N_{median} \leq N_{Naive}$

BANDITS SUMMARY :



31/1 :

MARKOV DECISION PROCESS :

(think this as controlled Markov chain)

- 4 - Tuple (S, A, P, R)

- $P : S \times S \times A \rightarrow [0, 1]$

where $P_{ij}(a)$: Probability of starting from 'i' and ending up in 'j' when action 'a' is taken.

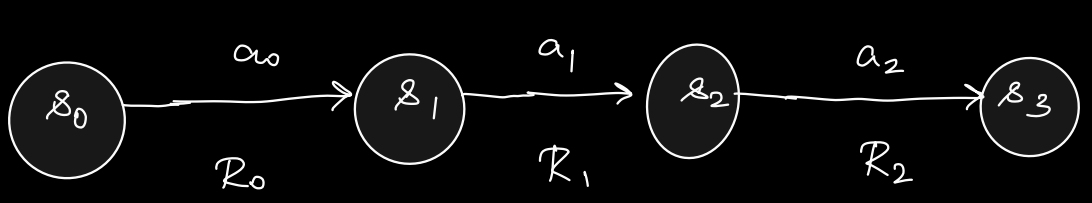
- $R : S \times A \times S \rightarrow \mathbb{R}$

where, $R_{ij}(a)$: Reward you will get starting from state 'i' and transition to 'j' when choosing action 'a'.

- If the states and actions are finite then we say MDP is finite MDP.

- For each action a, $P(a)$ is a $|S| \times |S|$ matrix

. Sequential decision and reward process :



Can be called as control Markov chain

$R_0 = R(s_0, a_0, s_1)$

. We want MDP to achieve maximum long-term reward.

TOTAL REWARD CRITERION :

. We want to maximize, $\mathbb{E}\left[\sum_{i=1}^{T} R_i\right]$ } Episodic

- Here we need a special state 's_T' in the MDP (terminal state) which is a absorbing state with $R(s_T, \cdot, s_T) = 0$

TOTAL DISCOUNTED REWARD CRITERION :

. Let $\gamma \in [0, 1)$ be the discount factor.

. Goal : $\max \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k R_k\right]$

. Using discounted reward results in :

1) Finite return
2) No need to have a terminal state
3) Weightage for a farthest reward to a action in a state decreases.

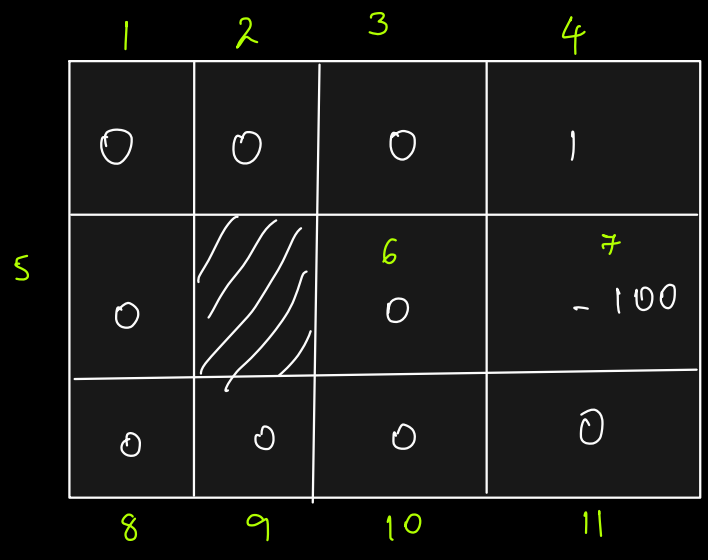
(i.e immediate rewards are given more emphasis)

AVERAGE REWARD CRITERION :

. Goal : $\max \lim_{T \to 0} \mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T} R_t\right]$

Example :

. Consider,



. Let $\gamma = 0.9$

- State space : Cells and $|S| = 11$

- Action space : $\{$ Left , Right , Up , Down $\}$

- Reward :

  - $R(i, \cdot, \cdot) = 0 \quad \forall \; i \neq 4, 7$

  - $R(4, \cdot, \cdot) = 1$

  - $R(7, \cdot, \cdot) = -100$

- Probability Transition Matrix :

  - If there is a wall then you remain in the same state.

  - 80 % of time you are going to the intended state for a given action.

  - 10 % of time you end up in a cell in the perpendicular direction.

- If we define a terminal state for the above example then we can consider total reward instead of discounted reward.

INFINITE HORIZON PROBLEM :

- 

- Here usually rewards are independant of time 't'.

FINITE HORIZON PROBLEM :

- 

- Here its possible to formulate the reward as a function of time.

- The Bandit setting has no 'Markov nature'
(No notion of states, MDP)

FINITE DISCOUNTED REWARD MDP :

- Representation : $< S, A, P, R, \gamma >$
- Goal : $\max \mathbb{E}\left[R_0 + \gamma R_1 + \gamma^2 R_2 + \dots \right]$
- Find $a_0, a_1, a_2, \dots$ such that expected discounted reward is maximized.

NOTION OF A POLICY :

joint strategy for all the actions

- Stationary deterministic policy $\pi : S \rightarrow A$,

- How many stationary deterministic policies in a finite MDP ?

Soln : $|A|^{|S|}$

2/2 :

Recap :

. Goal in the MDP setting :

- Find an action sequence $\{a_0, a_1, a_2, \ldots\}$ such that it maximizes some long run reward criterion.

. $R_t = R(s_t, a_t, s_{t+1})$

. Notion of policy :

1) Stationary

2) Deterministic

. given a policy 'μ', we associate a performance metric of 'μ' as,

$$V_\mu(s_0) = \mathbb{E}[R_0 + \gamma R_1 + \gamma^2 R_2 + \ldots \mid s_0]$$

(s_0 ↓ starting state)

where, $R_t = R(s_t, \mu(s_t), s_{t+1})$

. We define, $V_\mu = [V_\mu(1), V_\mu(2), \ldots, V_\mu(N)]$

. Here, V_μ has a POSET relation.

NOTE :

. POSET Relation :

. Any two elements in the set are comparable. (with $\leq$)

defi : 1) $a \leq a$: Reflexivity

2) If $a \leq b$, $b \leq a \Rightarrow a = b$

3) If $a \leq b$, $b \leq c \Rightarrow a \leq c$

. If we can define either $a \leq b$ or $b \leq a$ $\forall$ $a, b \in$ set then it is totally ordered.

. For a POSET relation, there exists a μ^* such that,

$$V_{\mu^*}(i) \geq V_{\mu}(i) \quad \forall \mu, \forall i \in \{1, \dots N\}$$

. Let Π be the set of all deterministic policies.

Then,

$$\mu^*(i) = \arg\max_{\mu \in \Pi} V_{\mu}(i)$$

. Why should a optimal stationary deterministic policy exist?

- TPM is stationary / Homogeneous

Is this usage correct?

- Reward function is stationary

$\therefore$ of these assumption we will have stationary deterministic policy

NOTE:

- Homogeneous Markov Chain:

$$\Rightarrow \quad P\{X_n = i \mid X_{n-1} = j\} = P\{X_{n+k} = i \mid X_{n-1+k} = j\}$$

- For a policy μ:

$$V_\mu(s_0) = \underbrace{E}_{\text{w.r.t randomness in next state and action if randomized}}\Big[R(s_0, \mu(s_0), s_1)\Big] + \gamma\, E\Big[R(s_1, \mu(s_1), s_2)\Big] + \gamma^2 E\Big[R(s_2, \mu(s_2), s_3)\Big] + \dots$$

- WKT, $E\Big[R(s_0, \mu(s_0), s_1)\Big] = \sum_{s_1=1}^{N} P_{s_0, s_1}(\mu(s_0))\, R(s_0, \mu(s_0), s_1)$

Expected immediate reward for policy 'μ'.

- We can observe that,

$$V_\mu(s_0) = \underbrace{E\Big[R(s_0, \mu(s_0), s_1)\Big]}_{\text{Expected Immediate reward following policy '}\mu\text{'}} + \gamma \underbrace{\sum P_{s_0, s_1}(\mu(s_0))\, V_\mu(s_1)}_{\text{Expected future reward}}$$

$$E\Big[R(s_1, \mu(s_1), s_2)\Big] + \gamma\, E\Big[R(s_2, \mu(s_2), s_3)\Big] + \gamma^2 E\Big[R(s_3, \mu(s_3), s_4)\Big] + \dots$$

. The above is true since,

$$V_\mu(s_0) = \sum_{t=0}^{\infty} \gamma^t \, \mathbb{E}\left[R(s_t, \mu(s_t), s_{t+1}) \right]$$

$$= \mathbb{E}\left[R(s_0, \mu(s_0), s_1) + \sum_{t=1}^{\infty} \gamma^t \, \mathbb{E}\left[R(s_t, \mu(s_t), s_{t+1}) \right] \right]$$

$$= \mathbb{E}\left[R(s_0, \mu(s_0), s_1) + \gamma \sum_{t=1}^{\infty} \gamma^{t-1} \, \mathbb{E}\left[R(s_t, \mu(s_t), s_{t+1}) \right] \right]$$

$$= \mathbb{E}\left[R(s_0, \mu(s_0), s_1) + \gamma \sum_{t=0}^{\infty} \gamma^t \, \mathbb{E}\left[R(s_{t+1}, \mu(s_{t+1}), s_{t+2}) \right] \right]$$

$$\left(\sum_{s' \in S} P_{s_0, s'}(\mu(s_0)) \sum_{t=0}^{\infty} \gamma^t \, \mathbb{E}_{\text{on}} \left[R(s', \mu(s_{t+1}), s_{t+2}) \mid s_{t+1} = s' \right] \right)$$

↓ conditional expectation on next state

$V_\mu(s')$

$$\Rightarrow V_\mu(s_0) = \mathbb{E}\left[R(s_0, \mu(s_0), s_1) \right] + \gamma \sum_{s' \in S} P_{s_0, s'}(\mu(s_0)) \, V_\mu(s')$$

$$= \sum_{s' \in S} P_{s_0, s'}(\mu(s_0)) \, R(s_0, \mu(s_0), s') + \gamma \sum_{s' \in S} P_{s_0, s'}(\mu(s_0)) \, V_\mu(s')$$

Possible to write this ∵ MDP is homogeneous and also reward function is stationary.

- WKT, $V_\mu(s_0) = \mathbb{E}[R_0] + \gamma\,\mathbb{E}[R_1] + \gamma^2\,\mathbb{E}[R_2] + \cdots\cdots$

- $\mathbb{E}[R_1] = \mathbb{E}\left[R(s_1, \mu(s_1), s_2)\right]$

$$= \mathbb{E}_{s_1}\left[\underbrace{\mathbb{E}_{s/s_1=j}\left[R(s_1, \mu(s_1), s_2) \mid s_1 = j\right]}_{(\cdot)}\right] = \sum_{s_1} P_{s_0,s_1}(\mu(s_0)) \cdot (\cdot)$$

- Similarly $\mathbb{E}[R_2] = \sum_{s_2} P^2_{s_0,s_2}(\mu(s_0))\; \mathbb{E}_{s_3/s_2}\left[R(s_2, \mu(s_2), s_3) \mid s_2\right]$

- Let, $R(s, \mu(s)) = \sum_{s' \in S} P_{s,s'}(\mu(s))\; R(s, \mu(s), s') = R_\mu(s)$

- $$\begin{bmatrix} V_\mu(1) \\ V_\mu(2) \\ \vdots \\ V_\mu(N) \end{bmatrix} = \begin{bmatrix} R_\mu(1) \\ R_\mu(2) \\ \vdots \\ R_\mu(N) \end{bmatrix} + \gamma \cdot \underbrace{P_\mu}_{S \times S} \begin{bmatrix} V_\mu(1) \\ V_\mu(2) \\ \vdots \\ V_\mu(N) \end{bmatrix}$$

- $P_\mu = P_{i,j}(\mu(i)) = \begin{bmatrix} P_{1,1}(\mu(1)) & P_{1,2}(\mu(1)) & \cdots \\ P_{2,1}(\mu(2)) & \vdots & \\ \vdots & & \end{bmatrix}$

$\Rightarrow \quad V_\mu = R_\mu + \gamma P_\mu V_\mu$

$\Rightarrow \quad V_\mu = (I - \gamma P_\mu)^{-1} R_\mu$ { Policy evaluation for policy μ.

$(I - \gamma P_\mu)^{-1}$: inverse to $\left(\frac{1}{1-x}\right)$

$\downarrow$

$(I + \gamma P_\mu + \gamma^2 P_\mu^2 + \gamma^3 P_\mu^3 + \ldots) R_\mu$

$= R_\mu + \gamma P_\mu R_\mu + \gamma^2 P_\mu^2 R_\mu + \ldots$

- Why should $(I - \gamma P_\mu)$ be invertible ?
  - If we show the eigen values of P_μ is ≤ 1 then it is obvious that eigen values of $(I - \gamma P_\mu) > 0$
  - Let λ be some eigen value of P_μ.

$\Rightarrow \quad P_\mu \cdot x = \lambda x$

  - Let $x_u = \frac{x}{\|x\|}$

$\Rightarrow \quad P_\mu x_u = \lambda x_u$

  - Consider $|x_j| = \max\{|x_{u1}|, \ldots |x_{un}|\}$

$\Rightarrow \quad \underbrace{P_\mu(j)}_{1 \times n \text{ vector}} x_u = \lambda x_j$

$$\Rightarrow \quad |\lambda x_j| = \left|\sum_i P_\mu(j,i)\, x_i\right|$$

$$\leq \sum_i P_\mu(j,i)\, |x_i|$$

$$\leq \underbrace{\sum_i P_\mu(j,i)}_{1}\, x_j$$

$$\leq |x_j|$$

$$\Rightarrow \quad |\lambda| \leq 1$$

$\Rightarrow$ Max. eigen value is 1.

- How to find the optimal policy ?

  - Naive algorithm is find V_μ $\forall$ $\mu \in \Pi$ (Policy space)

  - Choose V_{μ^*} by comparision

NOTE :

. For a matrix $A \in \mathbb{C}^{n \times n}$ with entries a_{ij}. Let R_i be the sum of absolute values of non-diagonal elements in the i^{th} row. Then, every eigenvalue of A lies within at least one of the Gershgorin discs $D(a_{ii}, R_i)$

(a_{ii} : Centre, R_i : Radius)

9/2 :

Recap :

. MDP : 4-Tuple $\langle S, A, P, R \rangle$

. Discounted reward formulation :

Goal $\max_{a_0, a_1, a_2, \ldots} \mathbb{E}\left[R_0 + \gamma R_1 + \gamma^2 R_2 + \ldots\right]$

. Stationary deterministic policy :

. $\mu : S \to A$

$|A|^{|S|}$ different policies

- We associate value function, V_μ to each policy 'μ'.

- WKT, $V_\mu(i) = \mathbb{E}_\mu[R_0 + \gamma R_1 + \gamma^2 R_2 + \cdots]$

$R_0 \downarrow R(i, \mu(i), s_1)$

$V_\mu(i)$ = Expected Immediate Reward + Expected Future reward

- Here, Expected Immediate Reward :

$$R_{\mu(i)} = \sum_{j=1}^{n} P_{ij}(\mu(i))\ R(i, \mu(i), j)$$

Expected Future Reward

$$= \gamma \sum_{j=1}^{n} P_{ij}(\mu(i))\ V_\mu(j)$$

- $\therefore V_\mu = R_\mu + \gamma P_\mu \cdot V_\mu$

- Conditions for stationary deterministic policy to be optimal.

- Interpretation of V^* :

  - $V^*(i)$: Value function for the optimal policy μ^* in state 'i'

  - $V^*(i)$: Maximum total discounted reward we collect starting from state 'i'.

- WKT, (Not a linear eq^n) $V^*(i) = \max_a \left[\underbrace{E[R(i,a,\cdot)]}_{= R(i,a)} + \gamma \sum_{j=1}^{n} P_{ij}(a) V^*(j) \right]$

$$\mu^*(i) = \underset{a}{\text{argmax}} \left[R(i,a) + \gamma \sum_{j=1}^{n} P_{ij}(a) V^*(j) \right]$$

- μ^* : Optimal stationary deterministic policy.

BELLMAN OPERATOR :

- Bellman Operator, $T : \mathbb{R}^n \to \mathbb{R}^n$

- Also, $TV^* = V^*$

- Here, $T(J(i)) = \max_a \left[R(i,a) + \gamma \sum_{j=1}^{n} P_{ij}(a) J(j) \right]$; $\forall\, i \in S$

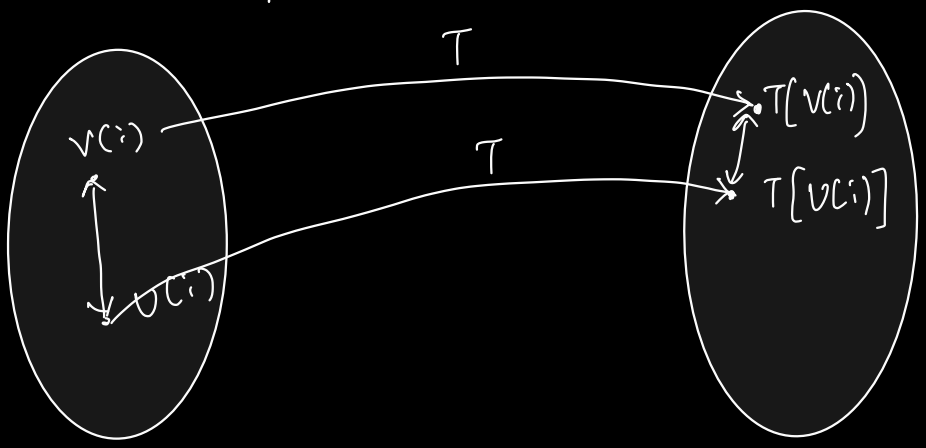
- We can show that V^* is unique.

  i.e $\quad T(V) \neq V \quad \forall \; V \neq V^*$

- V^* is called the fixed point of operator 'T'.

- The Bellman operator, T has a special property called contraction. i.e.,

$$|T(V(i)) - T(U(i))| \leq \alpha |V(i) - U(i)|$$



- The iterative procedure of applying 'T' can be done as a fixed point iteration.

- WKT,

$$V_1 = T(V_0)$$
$$V_2 = T(V_1)$$
$$\vdots$$
$$V_n = \underbrace{T^n(V_0)}_{T(T(\ldots))}$$

$$\Rightarrow \lim_{n \to \infty} T^n(V_0) = V^*$$

. We need to show that 'T' has contraction property.

VALUE ITERATION :

. Initialization : Initialize V_0

. For $t = 1, 2, \ldots$

$$V_t = T(V_{t-1})$$

$$= \max_a \left[R(i,a) + \gamma \sum_{j \in S} P_{ij}(a) V_{t-1}(j) \right] \quad \forall i$$

. Perform the above step until, $T(V) = V = V^*$

In practice check $|V_t - V_{t-1}| < \varepsilon$ then

output, $V^* = V_t$

11/1 :

1) ~~9/2~~
2) ~~11/2~~
3) ~~14/2~~
4) 16/2
5) 18/2

RECAP :

- Bellman Equation :

$$V^*(i) = \max_a \left[R(i,a) + \sum_{j=1}^{n} P_{ij}(a) V^*(j) \right]$$

- Optimal Value function : V^*

- $R(i,a) = \sum_{j \in S} P_{ij}(a) \, R(i,a,j)$

- Bellman Operator, $T : \mathbb{R}^n \to \mathbb{R}^n$ — Contraction operator

$$T[V(i)] = \max_a \left[R(i,a) + \sum_{j=1}^{n} P_{ij}(a) V(j) \right]$$

- $T(V^*) = V^*$ and V^* is unique.

STATE ACTION VALUE FUNCTION :

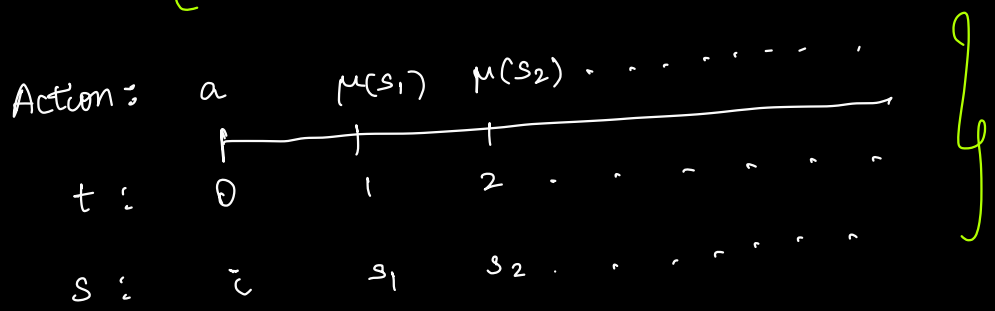
- $Q_\mu : S \times A \to \mathbb{R}$

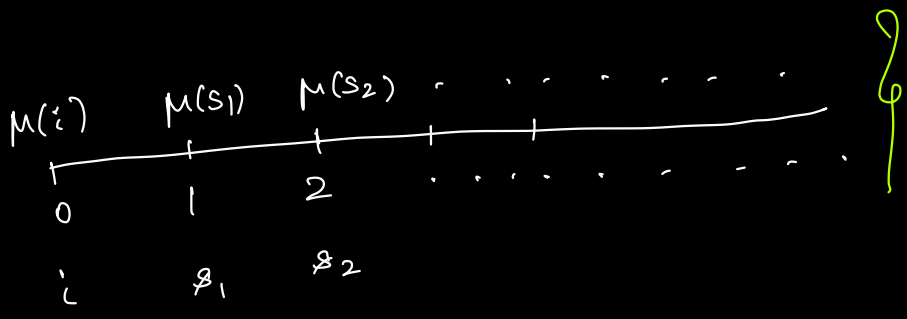
$\Rightarrow Q_\mu(i,a) \in \mathbb{R}$; $\forall i \in S$, $a \in A$

- $Q_\mu(i,a)$ is the expected reward we obtain by taking action 'a' in state 'i' and following policy 'μ' from the next state.

- $\therefore Q_\mu(i,a) = \mathbb{E}\left[R(i,a,\cdot)\right] + \gamma \sum_{j \in S} P_{i,j}(a)\, V_\mu(j)$

$$\mathbb{E}\left[R(s_0,a_0,s_1) + \gamma R(s_1,\mu(s_1),s_2) + \gamma^2 R(s_2,\mu(s_2),s_3) \mid s_0=i, a_0=a\right]$$





- Let μ' be a policy such that,

$$\mu'(s) = \begin{cases} \underset{a}{\operatorname{argmax}}\, Q_\mu(s,a) & ; \; s = i \\ \mu(s) & ; \; s \neq i \end{cases}$$

. We can show that,

$$V_{\mu'}(i) \geq V_{\mu}(i) \quad ; \quad \forall\ 'i'$$

and some $'i' \in S$, $V_{\mu'}(i) > V_{\mu}(i)$

This leads to optimal policy μ^*

POLICY ITERATION :

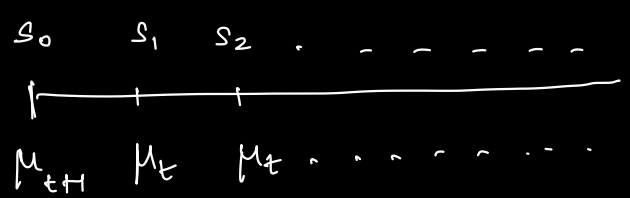
. Initialization : Start with a policy $'\mu_0'$.

. For $t = 0, 1, 2, \ldots$

- Policy evaluation : Compute V_{μ_t}
- Policy Improvement :
  - Compute $Q_{\mu_t}(i,a) \quad \forall\ i \in S, a \in A$
  - $\mu_{t+1}(i) = \underset{a}{\operatorname{argmax}}\ Q_{\mu_t}(i,a) \quad \forall\ i \in S$

. Repeat the above procedure till, $\mu_t(i) = \mu_{t+1}(i) \ ; \ \forall i \in S$

. At this point Bellman equation gets satisfied.



Claim :

. Following μ' for state s_0 at time $t=0$ and then following μ for next time-steps is inferior to following μ' at $t=0,1$ and then following μ $\forall$ $t \geq 2$.

$$\text{i.e } Q_\mu(i, \mu'(i)) = R(i, \mu'(i)) + \gamma \sum_{j=1}^{n} P_{ij}(\mu'(j))\, V_\mu(j) \quad -①$$

$$= R(i, \mu'(i)) + \gamma \sum_{j=1}^{n} P_{ij}(\mu'(j)) Q_\mu(j, \mu(j))$$

$$Q_\mu(i, \mu'(i)) \leq R(i, \mu'(i)) + \gamma \sum_{j=1}^{n} P_{ij}(\mu'(j))\, Q_\mu(j, \mu'(j))$$

Need to show this !

(.) Following μ' twice and then following μ $\forall$ $t \geq 2$

. Expanding (.) :

$$(.) = R(i, \mu'(i)) + \gamma \sum_{j=1}^{n} P_{ij}(\mu'(j)) \left[R(j, \mu'(j)) + \gamma \sum_{k=1}^{n} P_{jk}(\mu'(j))\, V_\mu(k) \right] \quad -②$$

$$\Rightarrow ① - ②$$

$$= V_\mu(j) - \left[R(j, \mu'(j)) + \gamma \sum_{k=1}^{n} P_{jk}(\mu'(j))\, V_\mu(k) \right]$$

$$= R(j, \mu(j)) + \gamma \sum_{k=1}^{n} P_{jk}(\mu(j)) V_{\mu}(k) - R(j, \mu'(j)) - \gamma \sum_{k=1}^{n} P_{jk}(\mu'(j)) V_{\mu}(k)$$

$$= \underbrace{R(j, \mu(j)) + \gamma \sum_{k=1}^{n} P_{jk}(\mu(j)) V_{\mu}(k)}_{x(\mu(j))} - \underbrace{\left[\max_{a \in A} \left(R(j,a) + \gamma \sum_{k=1}^{n} P_{jk}^{(a)} V_{\mu}(k) \right) \right]}_{x(a)}$$

- Since $\mu(j) \in A \Rightarrow \quad x(\mu(j)) \leq \max_{a \in A} x(a)$

$\Rightarrow \quad (1) - (2) \leq 0$

$\Rightarrow \quad (1) \leq (2)$

$\Rightarrow$ Taking μ' at $t=0,1$ and then μ $\forall$ $t \geq 2$ is better than taking μ' at $t=0$ and then μ $\forall$ $t \geq 1$.

14/1:

RECAP:

- Bellman Equation:

$$V^*(i) = \max_a \left[R(i,a) + \sum_{j=1}^{n} P_{ij}(a)\, V^*(j) \right]$$

$$\mu^*(i) = \arg\max_a \left[R(i,a) + \sum_{j=1}^{n} P_{ij}(a)\, V^*(j) \right]$$

- $T(V^*) = V^*$

$T : \mathbb{R}^n \longrightarrow \mathbb{R}^n$

contraction operator

- Value Iteration:

$$V_1 = TV_0$$

$$\vdots$$

$$\lim_{n\to\infty} V_n = V^*$$

- Policy Iteration:

1) Policy evaluation:

- $V_\mu = (I - \gamma P_\mu)^{-1} R_\mu$

2) Policy Improvement:

- $Q_\mu(i,a) = R(i,a) + \gamma \sum_{j=1}^{n} P_{ij}(a)\, V_\mu(j)$

- $\mu'(i) = \underset{a}{\operatorname{argmax}}\ Q\mu(i,a)$

- Let T_μ be a Bellman operator for a policy μ.

$T_\mu : \mathbb{R}^n \to \mathbb{R}^n$

- $\therefore\ T_\mu(V(i)) = R(i,\mu(i)) + \gamma \sum_{j=1}^{n} P_{ij}(\mu(i))\, V(j)$

- Also, $T_\mu(V_\mu) = V_\mu$ and V_μ is unique.

Value iteration for a single policy.

$\Rightarrow \lim_{k\to\infty} T_\mu^k(V_0) = V_\mu$.

NOTE:

- T is for the entire MDP and its fixed point is V^*.
- T_μ is for a fixed policy and its fixed point is V_μ.

- Remember, $T_\mu(V) = R_\mu + \gamma P_\mu \cdot V$

Fixed Point Of a function :

- Consider, $f : \underbrace{X}_{\text{vector space}} \rightarrow X$

- x^* is a fixed point if ,

$$f(x^*) = x^*$$

- Suppose 'f' is a contraction :

$$\Rightarrow |f(x) - f(y)| \leq \alpha |x - y| \; ; \; \alpha \in [0, 1)$$

- If 'f' is a contraction then we will have a unique fixed point.

Proof : Suppose not $\Rightarrow$ $\exists$ multiple fixed points.

- WKT, since 'f' is a contraction ,

$$|f(x^*) - f(y^*)| \leq \alpha |x^* - y^*|$$

Fixed points

$$\Rightarrow |x^* - y^*| \leq \alpha |x^* - y^*|$$

- As $\alpha \in [0, 1)$ ⇒ Contradiction.

∴ If 'f' is a contraction mapping then there exists a unique point.

Ex:

- Let $f(x) = \frac{1}{2}\left(x + \frac{\alpha}{x}\right)$ } convex comb. of x, α/x

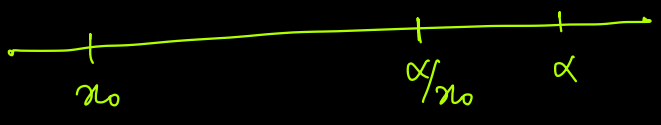
- consider, $x = \sqrt{\alpha}$ } unique

$$\Rightarrow f(\sqrt{\alpha}) = \frac{1}{2}\left(\sqrt{\alpha} + \alpha/\sqrt{\alpha}\right)$$

$$\underline{f(\sqrt{\alpha}) = \sqrt{\alpha}}$$

- $\Rightarrow f^n(x) \longrightarrow \sqrt{\alpha}$

Newton's method to find the square root.



- Let $\alpha = 16$, $x_0 = 2$

$\Rightarrow \alpha/x_0 = 8 \Rightarrow x_1 = \frac{1}{2}(2 + 8) = 5$

$\Rightarrow x_2 = \frac{1}{2}(5 + 16/5) = 4.1$

Claim :

. The operator, $T_\mu : \mathbb{R}^n \to \mathbb{R}^n$ is a contraction operator w.r.t l_∞ - norm.

$$\Rightarrow \quad \| T_\mu(V) - T_\mu(U) \|_\infty \leq \alpha \| V - U \|_\infty$$

Proof : WKT,

$$| T_\mu(V_1(i)) - T_\mu(V_2(i)) | = | \cancel{R(i,\mu)} + \gamma \sum_{j \in S} P_{ij}(\mu(i)) V_1(j) - \cancel{R(i,\mu)} - \gamma \sum_{j \in S} P_{ij}(\mu(i)) V_2(j) |$$

$$\leq \gamma \sum_{j \in S} P_{ij}(\mu(i)) \, | V_1(j) - V_2(j) |$$

(Triangle's inequality

. WKT, $$|V_1(j) - V_2(j)| \leq \max_{j \in S} |V_1(j) - V_2(j)|$$

$$= \| V_1 - V_2 \|_\infty$$

$$\Rightarrow \quad | T_\mu(V_1(i)) - T_\mu(V_2(i)) | \leq \gamma \sum_{j \in S} P_{ij}(\mu(i)) \, \| V_1 - V_2 \|_\infty$$

$$\Rightarrow \quad | T_\mu(V_1(i)) - T_\mu(V_2(i)) | \leq \gamma \| V_1 - V_2 \|_\infty \quad ; \quad \forall \, i$$

. WKT, $\max_i \left| T_\mu(V_1(i)) - T_\mu(V_2(i)) \right| \leq \gamma \| V_1 - V_2 \|_\infty$

$\Rightarrow \| T_\mu(V_1) - T_\mu(V_2) \|_\infty \leq \gamma \| V_1 - V_2 \|_\infty$

. $\therefore$ T_μ is a contraction w.r.t l_∞ - norm and with contraction operator $\gamma < 1$.

Claim :

. The Bellman operator 'T' is a contraction operator with contraction operator $\gamma < 1$ and w.r.t l_∞ - norm.

i.e $\| T(V_1) - T(V_2) \|_\infty \leq \gamma \| V_1 - V_2 \|_\infty$; $\gamma \in [0, 1)$

16/2 :

Recap :

. Policy evaluation :

Complexity is $O(n^3)$ { - Either using direct metod i.e $V_\mu = (I - \gamma P_\mu)^{-1} R_\mu$

(OR)

Complexity is $O(n^2 K)$ { - Recursively applying 'T_μ' operator with $V_0 = 0$

$\lim_{k \to \infty} V_k = V_\mu$ where, $V_k = T_\mu^k(V_0)$

Claim :

. The Bellman operator 'T' is a contraction operator with contraction operator $\gamma < 1$ and w.r.t l_∞-norm.

i.e $\| T(V_1) - T(V_2) \|_\infty \leq \gamma \| V_1 - V_2 \|_\infty \; ; \; \gamma \in [0, 1)$

. Consider state $s \in S$,

WKT, $$T(V_1(s)) = \max_{a \in A} \left[R(s,a) + \gamma \sum_{s' \in S} P_{s,s'}(a) \, V_1(s') \right]$$

$$T(V_2(s)) = \max_{a \in A} \left[R(s,a) + \gamma \sum_{s' \in S} P_{s,s'}(a) \, V_2(s') \right]$$

. Let $$a_1^* = \underset{a \in A}{\operatorname{argmax}} \left[R(s,a) + \gamma \sum_{s' \in S} P_{s,s'}(a) \, V_1(s') \right]$$

$$a_2^* = \underset{a \in A}{\operatorname{argmax}} \left[R(s,a) + \gamma \sum_{s' \in S} P_{s,s'}(a) \, V_2(s') \right]$$

$$\Rightarrow T(V_1(s)) - T(V_2(s)) = R(s, a_1^*) + \gamma \sum_{s' \in S} P_{s,s'}(a_1^*) \, V_1(s') - \left[R(s, a_2^*) + \gamma \sum_{s' \in S} P_{s,s'}(a_2^*) \, V_2(s') \right]$$

WKT, $R(s, a_2^*) + \gamma \sum_{s' \in S} P_{s,s'}(a_2^*) \, V_2(s') \geq R(s,a) + \gamma \sum_{s' \in S} P_{s,s'}(a) \, V_2(s') \; ; \; \forall a \in A$ ← Action space

$$\Rightarrow T(V_1(s)) - T(V_2(s)) \leq \cancel{R(s,a_1^*)} + \gamma \sum_{s' \in S} P_{s,s'}(a_1^*)\, V_1(s')$$

$$- \cancel{R(s,a_1^*)} + \gamma \sum_{s' \in S} P_{s,s'}(a_1^*)\, V_2(s') \quad (\because a_1^* \in A)$$

$$= \gamma \sum_{s' \in S} P_{s,s'}(a_1^*) \left(V_1(s') - V_2(s') \right)$$

- WKT $\forall s' \in S$, $\quad V_1(s') - V_2(s') \leq \max_{s' \in S} |V_1(s') - V_2(s')| = \|V_1 - V_2\|_\infty$

$$\Rightarrow T(V_1(s)) - T(V_2(s)) \leq \gamma \underbrace{\sum_{s' \in S} P_{s,s'}(a_1^*)}_{=1} \|V_1 - V_2\|_\infty$$

$$T(V_1(s)) - T(V_2(s)) \leq \gamma \|V_1 - V_2\|_\infty \quad ; \quad \forall\ s \in S \qquad -(1)$$

- similarly, $T(V_2(s)) - T(V_1(s)) \leq \cancel{R(s,a_2^*)} + \gamma \sum_{s' \in S} P_{s,s'}(a_2^*)\, V_2(s')$

$$- \cancel{R(s,a_2^*)} + \gamma \sum_{s' \in S} P_{s,s'}(a_2^*)\, V_1(s')$$

$$= \gamma \sum_{s' \in S} P_{s,s'}(a_2^*) \left(V_2(s') - V_1(s') \right)$$

- WKT $\forall s' \in S$, $\quad V_2(s') - V_1(s') \leq \max_{s' \in S} |V_1(s') - V_2(s')| = \|V_1 - V_2\|_\infty$

$$\Rightarrow T(V_2(s)) - T(V_1(s)) \leq \gamma \underbrace{\sum_{s' \in S} P_{s,s'}(a_2^*)}_{=1} \|V_1 - V_2\|_\infty$$

$$T(V_2(s)) - T(V_1(s)) \leq \gamma \|V_1 - V_2\|_\infty \quad ; \quad \forall\ s \in S \qquad -(2)$$

- $x \leq y$
- $-x \leq y$
- $\Rightarrow |x| \leq y$

$\Rightarrow \quad |T(V_2(s)) - T(V_1(s))| \leq \gamma \|V_1 - V_2\|_\infty \; ; \quad \forall \; s \in S$ (From ① and ②)

$\Rightarrow$ $\boxed{\|T(V_1) - T(V_2)\|_\infty \leq \gamma \|V_1 - V_2\|_\infty}$

• Comparision of value iteration and policy iteration.

- WKT for value iteration,

$$T[V_n(i)] = \max_a \left[R(i,a) + \sum_{j=1}^{n} P_{ij}(a)\, V_{n-1}(j) \right] ; \forall i$$

complexity, $O(|S|^2 \cdot |A|)$

- ⇒ For 'k' steps : $O(|S|^2 \cdot |A| \cdot K)$

• For policy iteration :

- Each step has complexity of $O(n^3)$
- For 'k' steps it will be $O(n^3 \cdot K)$

NOTE :

• Optimal value function is unique where as optimal policy need not be unique.

•

0	0	0	1
0	0	0	0
0	0	0	0

↑	→	→	Absorbing state
→↑	→	→	↑↑
↑↑ / →	→	→	→

All these policies are optimal

γ^3	γ^2	γ	1
γ^4	γ^3	γ^2	γ
γ^5	γ^4	γ^3	γ^2

} Unique value function.

MACHINE REPLACEMENT PROBLEM :

- MDP : $S := \{1, 2, \ldots, n\}$ } set of operations of machine

- Cost function, 'g' is a non-decreasing function of state.

- PTM : i) $P_{ij} = 0 \quad ; \quad i > j$

ii) $P_{ij} \leq P_{i+1,j} \quad ; \quad i < j$ } Stochastic dominance (check!) Yes!

NOTE :

- A MC with PTM 'P' is stochastically monotone if,

$$P_i \underset{s}{\geq} P_j \quad ; \quad \forall \; i > j$$

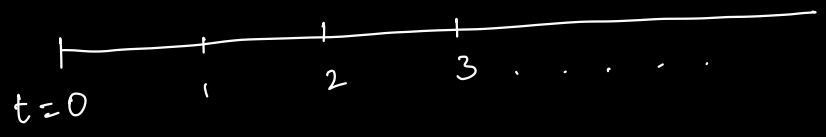
$\Rightarrow$ PMF of i^{th} row lies below j^{th} row.

• For a weakly increasing function, $V : S \to \mathbb{R}$ with $s_1 > s_2$ $(s_1, s_2 \in S)$ with PTM, P. Then,

$\mathbb{E}[V(s_{t+1}) \mid s_t = s_1] \geq \mathbb{E}[V(s_{t+1}) \mid s_t = s_2]$ iff 'P' is monotone.

. Action: 1) Replacement (OR) 2) Continue

. Machine Operation:



. If you replace the machine then it will run in state 1 for 1 time period.

COST FUNCTION:

. $c(i, c, j) = g(i)$

. $c(i, R, 1) = R + g(1)$

PROBABILITY TRANSITION MATRIX :

- $P(i, C, j) = P_{ij}$
- $P(1, R, 1) = 1$

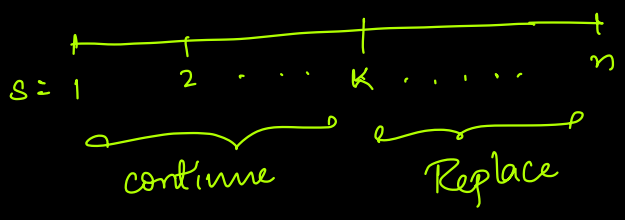
  $P(1, R, (\cdot)) = 0 \quad ; \quad \forall (\cdot) \neq 1$
- $P(i, R, 1) = 1 \quad ;$

BELLMAN EQUATION :

- $$V^*(i) = \min \left\{ g(i) + \gamma \sum_{j=1}^{n} P_{ij} V^*(j),\ R + g(1) + \gamma V^*(1) \right\}$$

- Intuitively optimal policy is a threshold type policy.

i-e



- $V^*(1) < V^*(2) < V^*(3) < V^*(4) \cdots$ (we can show this is true!)

Claim :

1) Show that V^* is monotonic (non-decreasing)
2) Show that the policy is threshold.

. We can prove claim ② assuming claim ① as follows:

$$V^*(i) = \min\left\{ \underbrace{R + g(1) + \gamma V^*(1)}_{(R)}, \underbrace{g(i) + \sum_{j=1}^{n} P_{ij} V^*(j)}_{(C)} \right\}$$

. since (R) remains same for all $i > 1$, if we show (C) is increasing w.r.t 'i' then we are done.

. WKT $g(i)$ is non-decreasing and also,

$$P_{ij} \leq P_{i+1,j}$$

. Now to prove claim 1, consider $V_0 = 0$.

WKT, $V_1 = T(V_0)$

$$V_1(i) = \underbrace{\min\{R + g(1), g(i)\}}_{\text{non-decreasing } \because g(i) \text{ is non-decreasing}}$$

$$V_2(i) = \min\left\{ R + g(1) + \gamma V_1(1), g(i) + \gamma \sum_{j=1}^{n} P_{ij} V_1(j) \right\}$$

- For some $K > i$

$$V_2(K) = \min\left\{ R + g(1) + \gamma V_1(1),\ g(K) + \gamma \sum_{j=1}^{n} P_{Kj} V_1(j) \right\}$$

- Since $g(K) > g(i)$

- Also, $\sum_{j=1}^{n} P_{Kj} V_1(j) - \sum_{j=1}^{n} P_{ij} V_1(j)$

$P_{ij} = 0 \ ; \ i > j$

$P_{ij} \le P_{i+1,j} \ ; \ i < j$

$$= \sum_{j=1}^{K} P_{Kj} V_1(j) - \sum_{j=1}^{i} P_{ij} V_1(j)$$

$$= \sum_{j=1}^{i} (P_{Kj} - P_{ij}) V_1(j) + \sum_{j=i+1}^{K} P_{Kj} V_1(j)$$

$$= \sum_{j=1}^{i} (P_{Kj} - P_{ij}) V_1(j) + \sum_{j=i+1}^{K} P_{Kj} V_1(j)$$

$(P_{Kj} - P_{ij}) \ge 0$; $V_1(j) \ge 0$; $P_{Kj} \ge 0$; $V_1(j) \ge 0$

$\Rightarrow V(i)$ is non-decreasing function in i.

18/2:

- Now when discount factor, $\gamma = 1$

  - For this case we need to have a terminal state, s^* with $V(s^*) = 0$ else, $(I - P_\mu)$ may not be invertible.

- The MDP problems with $\gamma = 1$ are called stochastic shortest path problem (OR) Total reward maximization problem.

NOTE:

- For episodic tasks we consider $\gamma = 1$ and special terminal state
- For continuing task, $\gamma < 1$.

STATIONARY RANDOMIZED POLICY:

- $\pi : S \to \mathbb{P}(A)$
- Here, $V_\pi(i) = \sum_{a,j} \pi(a|i)\, P_{ij}(a)\, R(i,a,j)$

$$+ \gamma \sum_{a=1}^{n} \pi(a|i) \sum_{j=1}^{n} P_{ij}(a)\, V_\pi(j)$$

- Vector Equation :

$$V_\pi = R_\pi + \gamma P_\pi V_\pi$$

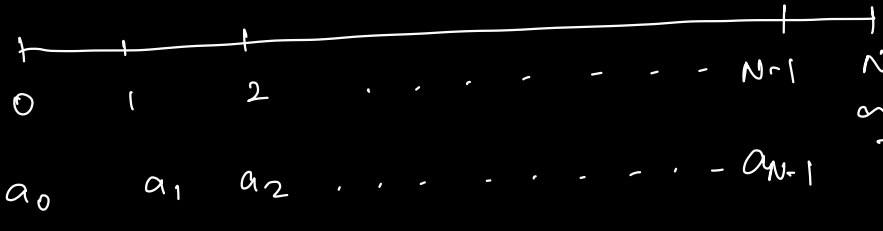
where, $R_\pi(i) = \sum_{a=1}^{m} \pi(a|i) \sum_{j=1}^{n} P_{i,j}(a) R(i,a,j)$

$$P_\pi(i,j) = \sum_{a=1}^{n} \pi(a|i) P_{i,j}(a)$$

- For a finite MDP, we will always have an optimal stationary deterministic policy.

FINITE HORIZON PROBLEM :

-



- Ex : Chess game :

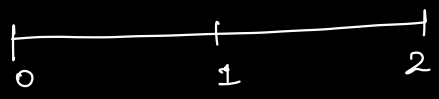
Policies : 1) Play Aggressively

Result = { win w.p P_w ; Loose w.p $1-P_w$

2) Play defensively:

$$\text{Result} = \begin{cases} \text{Draw} & \text{w.p } P_d \\ \text{Loose} & \text{w.p } 1-P_d \end{cases}$$

- Consider a 3-match series chess game:



- If we are tied after playing game '0' and '1' then for game '2' we play aggressively.

(OR)

The player who plays aggressively wins.

- State space:
  - Let $S = (w_1, w_2)$
    - w_1: # of times we win
    - w_2: # of times opponent player wins.
  - Initial state: $s_0 = (0,0)$

State at time '1'

- $s_1 \in \{(1,0), (0,1), (0.5, 0.5)\}$

State at time '2'

- $s_2 \in \{(2,0), (1,1), (1.5, 0.5), (0,2), (0.5, 1.5)\}$

. In finite horizon problem, we need not consider PTM to be stationary. Also reward function need not be stationary.

i.e PTM can vary w.r.t time.

. How to solve the finite horizon problem?

– We use the Backward induction technique.

BACKWARD INDUCTION TECHNIQUE :

. We define,

Is this our assumption?

$$V_2(s) = \begin{cases} 1 & ; \quad w_1 > w_2 \\ p_w & ; \quad w_1 = w_2 \\ 0 & ; \quad w_1 < w_2 \end{cases}$$

Is this after playing game '2'?

. How we find V_1?

WKT, $$V_1(s_1) = \max_a \left(R(s,a) + \mathbb{E}\left[J_2(s_2) \right] \right)$$

$$\mu(s_1) = \operatorname*{argmax}_a \left(R(s,a) + \sum_{s_2} P_{s_1,s_2}^{(a)} V_2(s_2) \right)$$

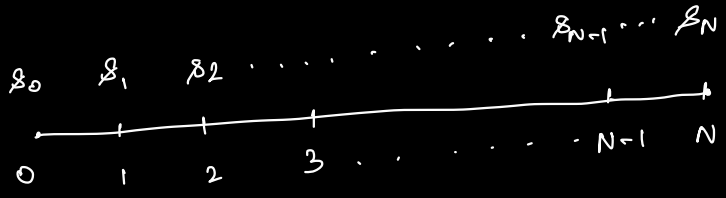
. Illy, $$J_0(s_0) = \max_{a_0} \left[R(s_0,a_0) + \sum_{s_1} P_{s_0,s_1}^{(a_0)} V_1(s_1) \right]$$

- Here, $\mu_0 : S \to A$, $\mu_1 : S \to A$, $\mu_2 : S \to A$

These policies need not be same.

28/2 :

FINITE HORIZON PROBLEM:



- $J_0(s_0) = \mathbb{E}\left[R_0 + R_1 + R_2 + \dots + R_{N-1} + R_N \right]$

  ↓ Expectation is w.r.t actions $a_0, a_1, \dots, a_{N-1}$

- $J_N(s_N) = R_N$

- $R_t = R_t(s_t, a_t, s_{t+1})$

- $P_t(s_{t+1} \mid s_t, a_t)$: Can be time dependant

- $J_{N-1}(s_{N-1}) = \max_{a_{N-1}} \mathbb{E}\left[R(s_{N-1}, a_{N-1}, s_N) + J_N(s_N) \right]$

$$= \max_{a_{N-1}} \left(\sum_{s_N \in S} \left(R(s_{N-1}, a_{N-1}, s_N) + J_N(s_N) \right) P(s_N \mid s_{N-1}, a_{N-1}) \right)$$

• In general,

$$J_K(s_K) = \max_{a_K} \left(R(s_K, a_K) + \sum_{s_{K+1}} P\{s_{K+1} | s_K, a_K\} J_{K+1}(s_{K+1}) \right)$$

• For finite horizon problem the policy is time dependant.

i.e

NOTE :

• Even when the reward and probability transitions are time independant this doesn't gaurentee that stationary policy is optimal. (∵ the horizon itself is finite)

CHESS PROBLEM :

• Let $x = w_1 - w_2$ (Net score)

$$\Rightarrow J_2(x) = \begin{cases} 1 & ; \text{ if } x > 0 \\ 0 & ; \text{ if } x < 0 \\ p_w & ; \text{ if } x = 0 \end{cases}$$

When the final score is (1,1) we enter sudden-death i.e players keep playing until someone wins.

p_w — why does this make sense when both the players have won equal games?

→ playing boldly will end the sudden-death match at once.

$\Rightarrow J_2((2,0)) = 1$

(w_1, w_2)

$\therefore J_1((1,0)) = \max\{$ Return due to playing aggressively: $p_w J_w(2,0) + (1-p_w) J_w(1,1)$, playing defensively: $p_d J_2(1.5, 0.5) + (1-p_d) J_2(1,1)$ $\}$

$= \max\{ \underbrace{p_w + (1-p_w) p_w}_{a} , \underbrace{p_d + (1-p_d) p_w}_{b} \}$

WKT, $p_d > p_w$

$\Rightarrow \cancel{2p_w} - p_w^2 \; (\;) \; p_d + \cancel{p_w} - p_d p_w$

$\Rightarrow p_w - p_w^2 - p_d + p_d p_w \; (\;) \; 0$

$\Rightarrow p_w - p_d + p_w(p_d - p_w) \; (\;) \; 0$

$\Rightarrow \underbrace{(p_w - p_d)}_{\leq 0} \underbrace{(1 - p_w)}_{\geq 0} \; (\;) \; 0$

$\Rightarrow a \leq b$

$\Rightarrow$ For $s_1 = (1,0)$ play defensively.

In general,

$$J_t(x_t) = \max \begin{cases} p_d\, J_{t+1}(x_t) + (1-p_d)\, J_{t+1}(x_t - 1) \\ p_w\, J_{t+1}(x_t + 1) + (1-p_w)\, J_{t+1}(x_t - 1) \end{cases}$$

$\downarrow$

$w_1 \cdot w_2$

$$\mu_t(x_t) = \begin{cases} \text{play bold} ; & \text{if } p_d\, J_{t+1}(x_t) + (1-p_d)\, J_{t+1}(x_t - 1) \\ & \quad < p_w\, J_{t+1}(x_t + 1) + (1-p_w)\, J_{t+1}(x_t - 1) \\ \text{play defensively} ; & \text{Otherwise} \end{cases}$$

$$\Rightarrow J_1(0-1) = \max \begin{cases} p_d\, J_2(0.5 - 1.5) + (1-p_d)\, J_2(0-2) \\ p_w\, J_2(1-1) + (1-p_w)\, J_2(0-2) \end{cases}$$

$$= \max \begin{cases} p_d\,(0) + (1-p_d)\,(0) \,,\; p_w \cdot p_w + (1-p_w)(0) \end{cases}$$

$$= \max \{ 0,\, p_w^2 \}$$

$\Rightarrow J_1(0-1) = p_w^2$; $\mu_1(0-1)$: Play aggressively

· Why $J_1(0.5-0.5) = \max\{ p_d J_2(1-1) + (1-p_d) J_2(0.5-1.5),$

$p_w J_2(1.5-0.5) + (1-p_w) J_2(0.5-1.5) \}$

$= \max\{ p_d \cdot p_w,\ p_w \}$

$= p_w \quad (\because p_d < 1)$

$\Rightarrow \mu_1(0.5-0.5)$: Play aggressively

$\therefore J_0(0-0) = \max\{ p_d \underbrace{J_1(0.5-0.5) + (1-p_d) J_1(0-1)}_{a},$

$\underbrace{p_w J_1(1-0) + (1-p_w) J_1(0-1)}_{b} \}$

$= \max\{ p_d \cdot p_w + (1-p_d) p_w^2,$

$p_w(p_d + (1-p_d) p_w) + (1-p_w) p_w^2 \}$

$\Rightarrow \cancel{p_d p_w} + (1-p_d) p_w^2 \ (\ ) \ \cancel{p_w p_d} + p_w^2 - p_w^2 p_d + p_w^2 - p_w^3$

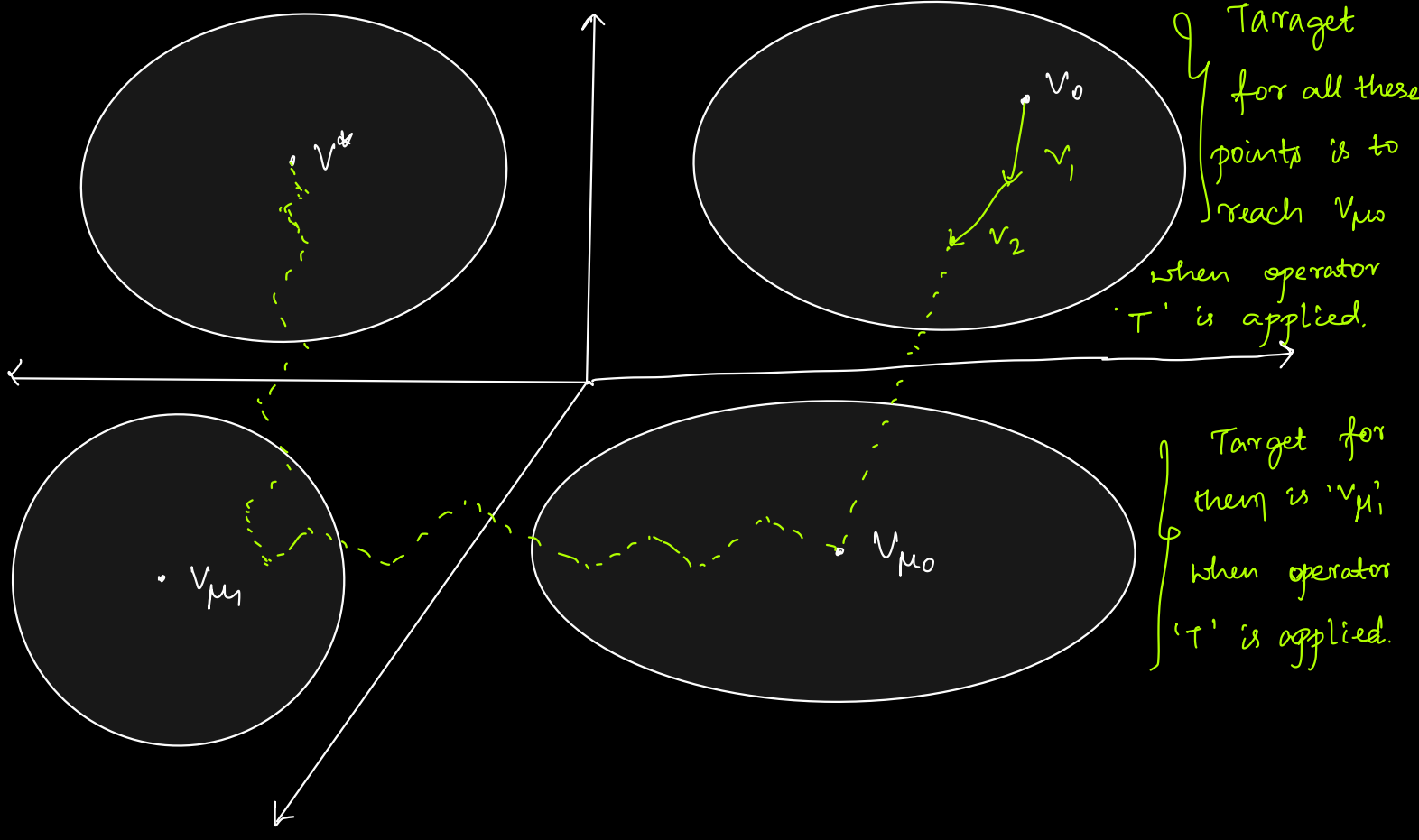
$0 \ (<) \ \underbrace{p_w^2}_{>0} \underbrace{(1-p_w)}_{>0}$

$\Rightarrow$ $\mu_0(0\text{-}0)$: Play aggressively.

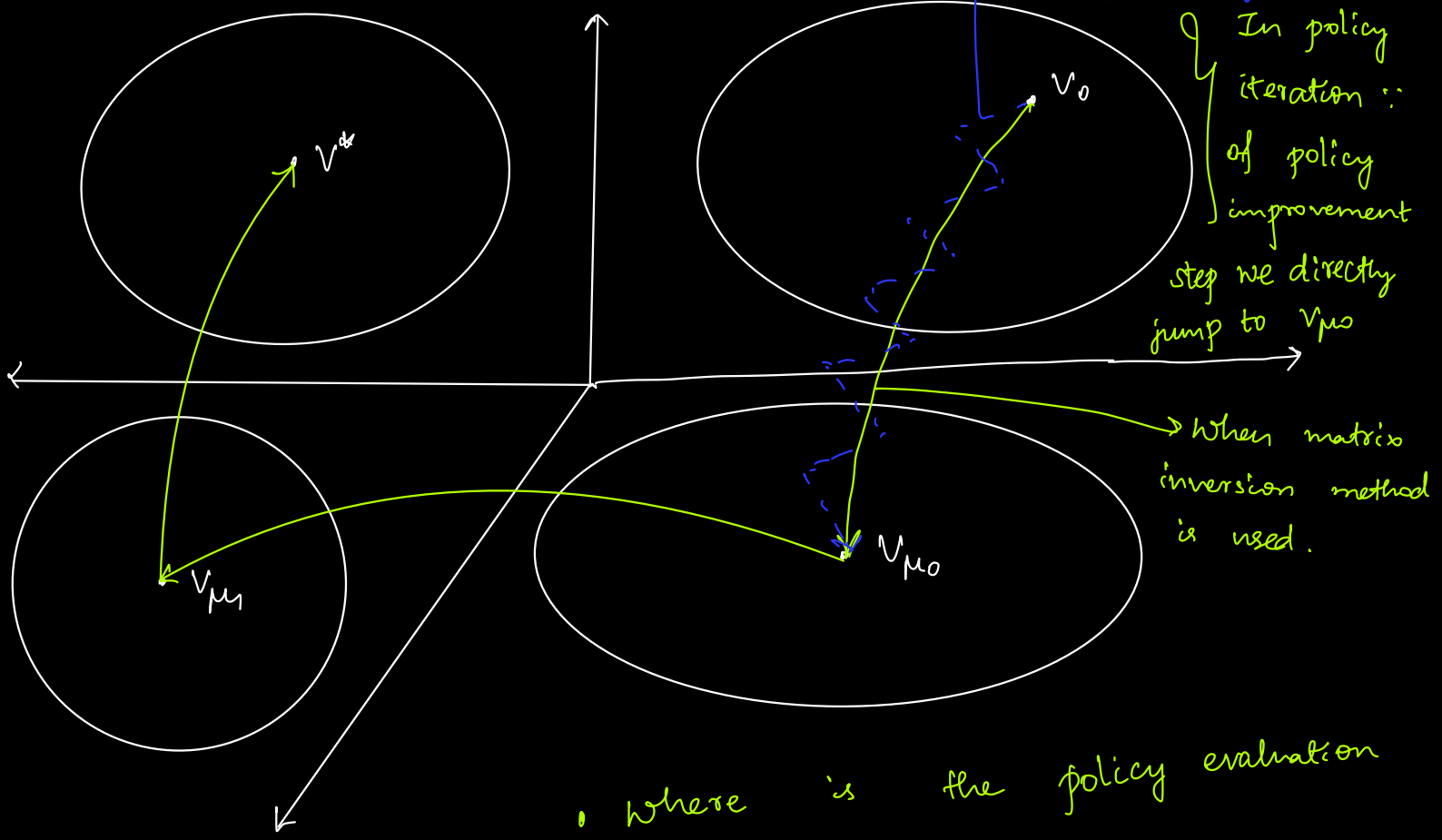
- Conclusion :
  - Play aggressively in the starting match
  - Play defensively if we are ahead in the next matches.

PICTORIAL REPRESENTATION OF VALUE AND POLICY ITERATION :

- Value Iteration : $\mathbb{R}^n$



• Policy Iteration : $\mathbb{R}^n$



• where is the policy evaluation step here ?

• Why the regions are contiguous ? (why not

both these regions have same μ as greedy policy.

- WKT, policy is greedy in a region if ;

$$R(s,\mu(s)) + \sum_{s'} P_{s,s'}(\mu(s))\, J(s) \geq R(s,a) + \sum_{s'} P_{s,s'}(a)\, J(s) \quad ; \quad \forall\ a \neq \mu(s)$$

Also true $\forall\ s \in S$

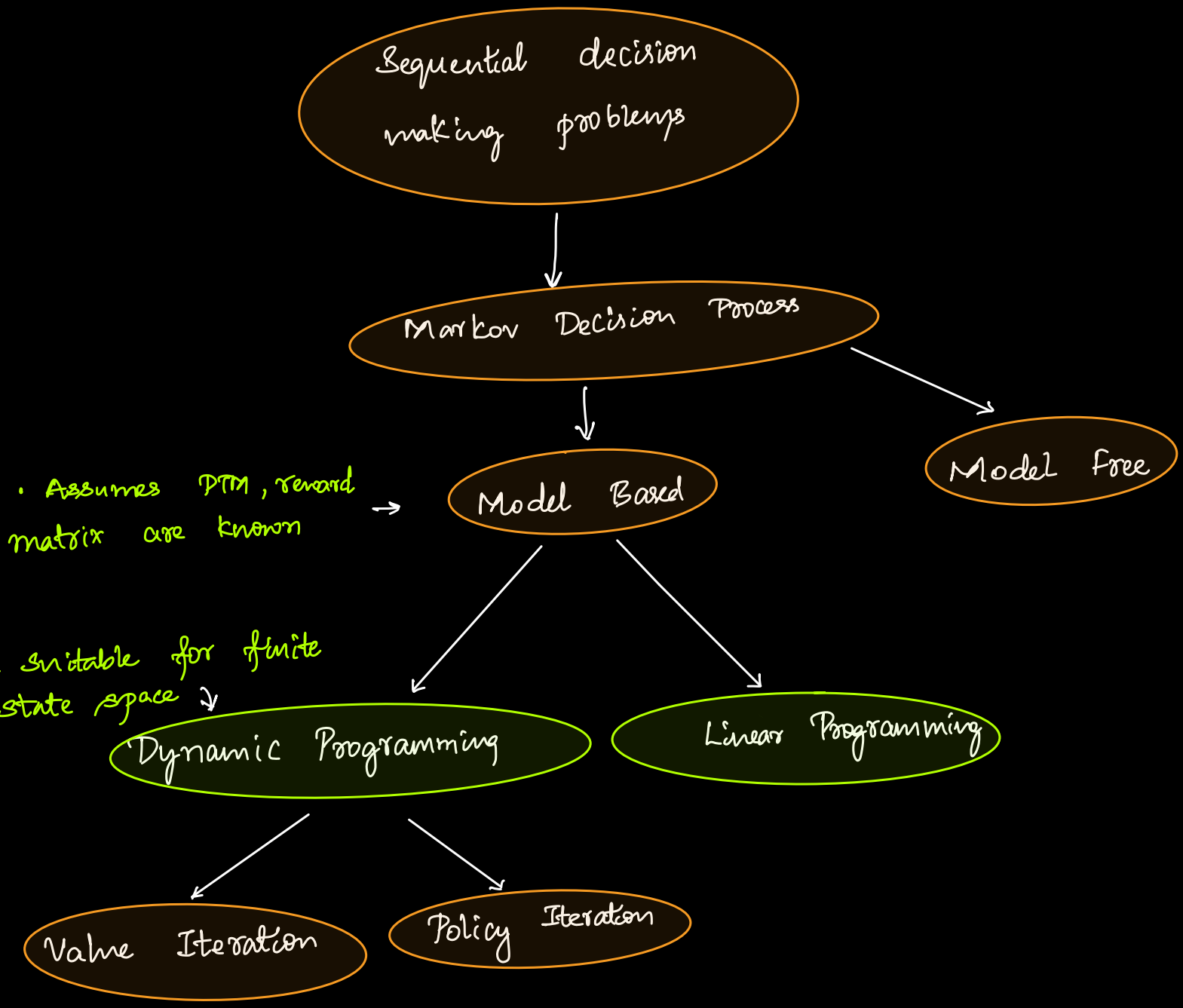
For all these ⇒ There are $|A|-1$ constraints.

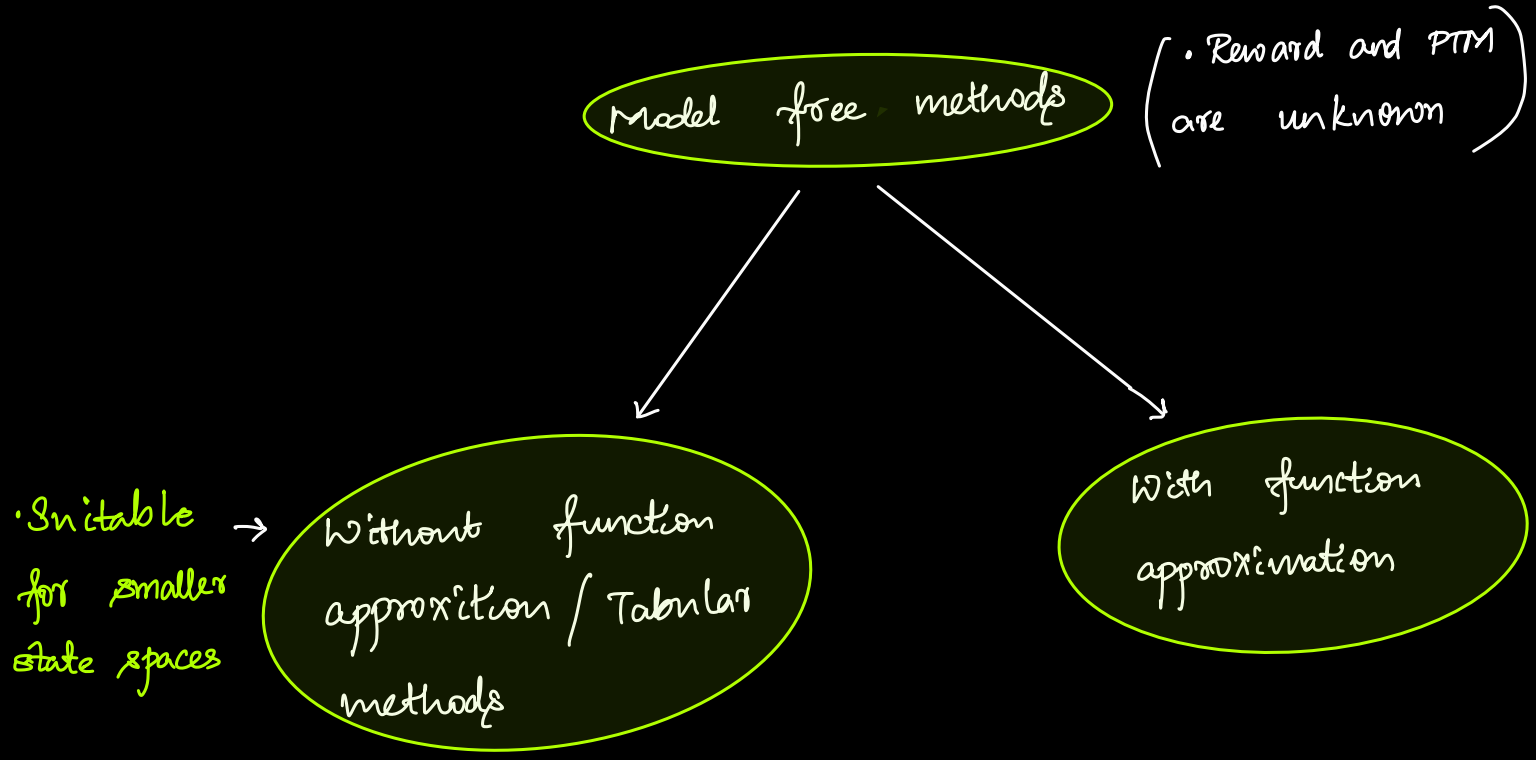
$J(s)$, $\mu(s)$ will be greedy (which are linear constraints)

. So the region will be a polytope with the region of interest being the intersection of the region formed due to linear constraints.

2/3:

BIG PICTURE OF RL :





- Model Free Methods can be classified into :

  I) Monte Carlo Methods

  II) Temporal difference methods

  depends on a particular trajectory (path) to reach the end state

- Why model free methods instead of estimating PTM and reward matrix.

  - We can do this but this might be a tedious task.

- The main steps in RL based methods is

  1) Prediction (similar to policy evaluation)

  2) Control (similar to policy improvement)

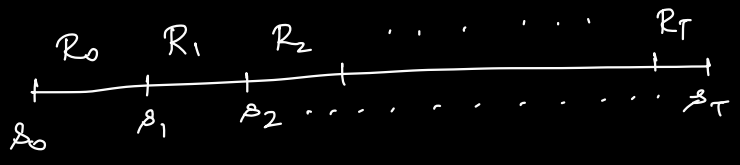
- For prediction we use either MC or TD method.

NOTE :

- In all the settings that we are going to deal from now on, we consider shortest stochastic path objective criterion (i.e total reward criterion)

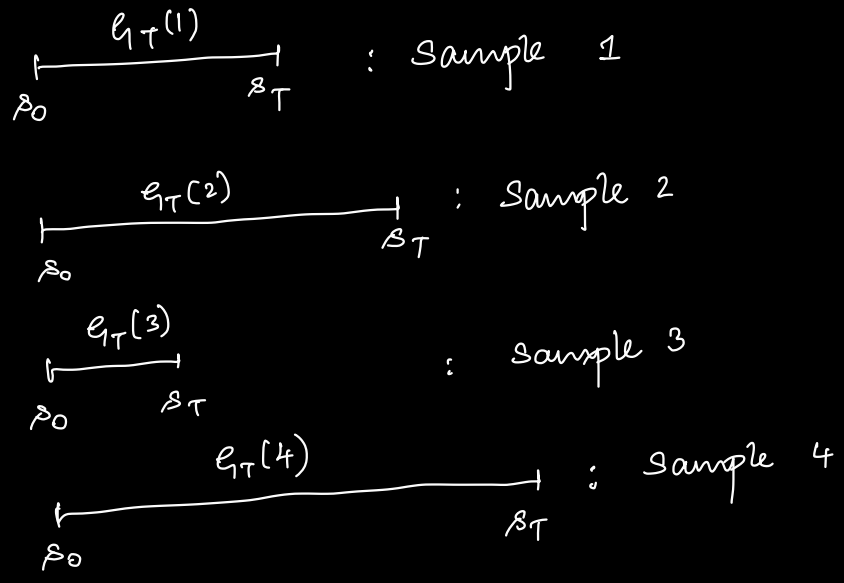
  ⇒ Assumption of terminal state.

- WKT, for a particular trajectory :



$$\Rightarrow \quad v_\mu(s_0) = \mathbb{E}\left[\sum_{t=0}^{T} R_t\right]$$

Can't be computed since the dynamics of the model is unknown.



$$\Rightarrow V_\mu(s_0) = \sum_{i=1}^{4} \frac{g_T^{(i)}}{4}$$

. In general for any state, s,

$$V_\mu(s) = \sum_{i=1}^{N} \frac{g_{[s_t = s \,:\, s_T]}^{(i)}}{N}$$

Return obtained from state 's' till the terminal state 's_T'.

. It is possible that a state 's' can be visited more than once in single trajectory itself. Based on the usage of return from this intermediate visit to the terminal we have 2 methods :

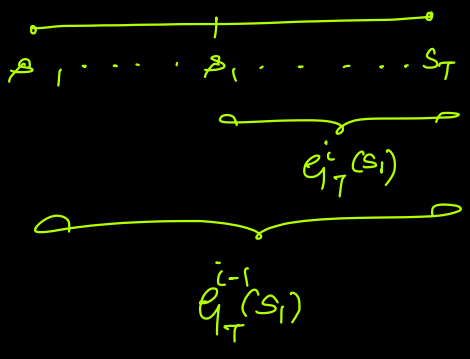
1) First visit Monte-Carlo
2) Every visit Monte-Carlo

Samples are correlated

- In First visit MC each sample is i.i.d.

  a trajectory starting with our state of interest till the end of the episode.

- In Every visit MC some samples are correlated.



This return is dependent on the trajectory of the return $G_T^{(i-1)}(s_1)$

- We can see that, first visit MC is unbiased but has high variance. ($\because$ Total variance $\propto \frac{1}{n}$ → # of samples

- Also, every visit MC is biased and the variance is low when compared with first visit MC.

  By how much the variance is low depends on how many times we visit the same state in a trajectory

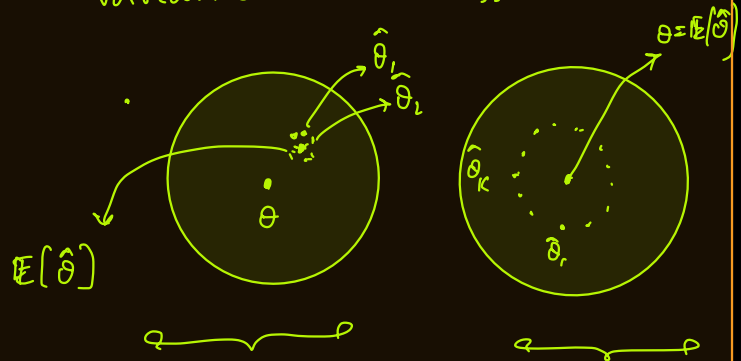
  ($\because$ Total variance $\propto \frac{1}{m}$) where $m \geq n$

• How do we improve the policy ?

- We need $Q_\mu(i,a)$ for policy improvement.

- Since we only have policy 'μ' known to us, we may not be able to estimate $Q_\mu(i,a)$ $\forall\ a \neq \mu(i)$.

• Dart Analogy for Bias-Variance Trade-Off :



Bias is non-zero but variance is less

Bias is zero but high variance

- Hence we use some exploration strategy.

i.e $\mu = \begin{cases} \mu & \text{w.p } 1-\varepsilon \\ \text{other actions} & \text{w.p } \varepsilon \end{cases}$ { on-policy MC control (Randomized Policy)

- $\mu'(s) = \underset{a}{\text{argmax}}\ Q_\mu(s,a)$

4/3 :

Recap :

- MC control is a full RL method. It consists of :

I) Policy evaluation - Stationary randomized policy (ε - greedy)

On-policy control { Evaluate Q_π using 1) First visit MC
2) Every visit MC

II) Policy improvement

$$\mu'(s) = \underset{a}{\operatorname{argmax}}\ Q_\mu(s, a)$$

Ex :

Game of Black Jack :

- Dealer 

Agent :

Goal to have score close to 21

Action : Hit / Stick

- State space : (3 dimensional)
  - Sum (11 - 20) of the player
  - Face up card of dealer : {1, ..., 10}
  - Usable ace or not : {0, 1}

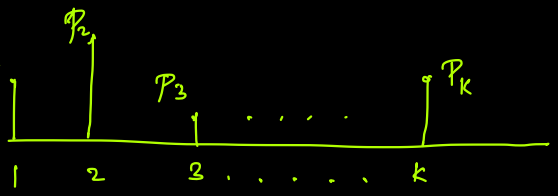
$\Rightarrow$ $|S| = 200$

- Action = {Hit, Stick}

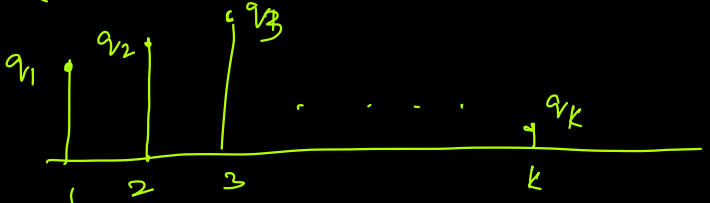
Off Policy Monte Carlo :

- In off-policy MC our goal is to learn the value function corresponding to target policy π'. However the actions corresponding to only behavior policy is allowed.
- How to estimate $V_{\pi'}$ by following policy π?
  - Use importance sampling.

NOTE :

. let $X \sim$ 

$$\mathbb{E}X = \sum_{i=1}^{k} i\, p_i$$ ← our interest

. let $Y \sim$ 

$$\mathbb{E}Y = \sum_{i=1}^{k} i\, q_i$$

. We want to estimate $\mathbb{E}X$ whose p_i's are known. However we are restricted to using only samples of Y that follows some other distribution.

. For each realization of Y we weight it by (p_i/q_i).

$$\Rightarrow = \sum_{i=1}^{n} i \cdot \frac{p_i}{q_i} \cdot q_i$$

$$\mathbb{E}X = \sum_{i=1}^{n} i\, p_i$$

- In general, let $X \sim f(x)$, $Y \sim g(x)$

WKT, $$\mathbb{E}X = \int_{-\infty}^{\infty} x \, f_X(x) \, dx$$

$$= \int_{-\infty}^{\infty} x \cdot f_X(x) \cdot \frac{g_X(x)}{g_X(x)} \cdot dx$$

$$= \int_{-\infty}^{\infty} x \cdot \frac{f_X(x)}{g_X(x)} \cdot g_X(x) \, dx$$

$$\mathbb{E}X = \mathbb{E}_{X \sim g_X}\left[x \cdot \frac{f_X}{g_X} \right]$$

ORDINARY IMPORTANCE SAMPLING :

$$\Rightarrow \frac{1}{n} \sum_{i=1}^{n} x_i \frac{f(x_i)}{g(x_i)} \longrightarrow \mathbb{E}_{x \sim f_X}[x]$$

(x_i sampled using dist^n 'g_X')

Unbiased but high variance

Importance factor : Specifies how important 'x' is while finding the expectation w.r.t 'f_X'.

WEIGHTED (Normalized) IMPORTANCE SAMPLING :

$$\frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i} \longrightarrow \mathbb{E}x$$

Biased but low variance

where, $w_i = \frac{p_i}{q_i}$

- Conditions on q_i are :

1) $q_i > 0 \quad \forall \quad p_i > 0$

- When q_i's are close to '0' then the weights w_i's will be very large resulting in instability in the (oscillations in the estimation) case of Ordinary IS.

- In weighted IS the denominator $\sum_{i=1}^{n} \underbrace{\frac{p_i}{q_i}}_{w_i}$ will take care of this. This results in ratio being very small.

$$\frac{1}{\sum_{i=1}^{n} \left(\frac{p_i}{q_i}\right)} \sum_{i=1}^{n} \frac{p_i}{q_i} x_i$$

7/3 :

9/3:

Recap:

- Monte - Carlo Prediction:

Incremental update of Monte-Carlo.

$$V_{t+1}(i) = V_t(i) + \underbrace{\alpha_n}_{\downarrow \frac{1}{n+1}} \left(G_t - V_t(i) \right)$$

- WKT, $V_\mu(i)$: Target value

  $V_t(i)$: Estimate of value function at time 't'

- New estimate = Current Estimate + α (sample - current estimate)

TEMPORAL DIFFERENCE PREDICTION:

-

TD update rule

$$V_{t+1}(i) = V_t(i) + \alpha_n \left(R_{t+1} + \gamma V_t(j) - V_t(i) \right)$$

- As $t \to \infty$, $V_t(i) \longrightarrow V_\mu(i) \quad \forall\ i$

- Convergence :

  - $V_{t+1}(i) = V_t(i) + \alpha_t \delta_t$

    ↓

    Temporal difference

  - For convergence : $\sum_{t=0}^{\infty} \alpha_t = \infty$ and $\sum_{t=0}^{\infty} \alpha_t^2 < \infty$

    $\Rightarrow V_t \xrightarrow{t \to \infty} V_\mu$

- Let T_μ be a contraction operator for a fixed policy 'μ' with contraction factor 'δ'.

  $\Rightarrow V_{t+1} = T_\mu V_t$ and $V_t \xrightarrow{t \to \infty} V_\mu$

- Now consider,

  $(1-\alpha) I + \alpha T_\mu$ — Is this a contraction

- WKT, $\| \left((1-\alpha) I + \alpha T_\mu \right) x - \left((1-\alpha) I + \alpha T_\mu \right) y \|$

  $= \| (1-\alpha) I (x-y) + \alpha T_\mu (x-y) \|$

  $\leq \| (1-\alpha) I (x-y) \| + \alpha \| T_\mu (x-y) \|$

  $\leq \underbrace{\left((1-\alpha) + \alpha \delta \right)}_{= 1 - \alpha(1-\delta) < 1} \| x-y \|$

$\Rightarrow$ $(1-\alpha)I + \alpha T_\mu$ is a contraction operator with contraction factor $(1-\alpha)I + \alpha S$.

- Comparing $1-\alpha(1-\delta) \gtrless \delta$

$$1 - \alpha + \alpha\delta - \delta \gtrless 0$$

$$1 - \alpha - \delta(1-\alpha) \gtrless 0$$

$$(1-\alpha)(1-\delta) \gtrless 0$$

($1-\alpha > 0$, $1-\delta > 0$)

$\Rightarrow$ T_μ is a better contraction operator compared to $(1-\alpha)I + \alpha T_\mu$.

(slow contraction)

- We can see that for TD prediction:

$$V_{t+1}(i) = (1-\alpha)\, V_t(i) + \alpha\,[R_t(i) + \gamma V_t(j)]$$

- Also,

$$V_{t+1}(i) = \left((1-\alpha)I + \alpha T_\mu\right) V_t(i)$$

$$x_{t+1} = T(x_t)$$

$$= (1-\alpha)\, V_t(i) + \alpha\, T_\mu(V_t(i))$$

$$V_{t+1}(i) = (1-\alpha)\, V_t(i) + \alpha \underset{a \sim \mu(i)}{\mathbb{E}} \left[R_t(i,a) + \gamma \sum_{j \in S} p_{ij}(a)\, V_t(j) \right]$$

- Using just a sample in the place of expectation,

we have :

$$V_{t+1}(i) = (1-\alpha) V_t(i) + \alpha \left[R_t(i, \mu(i), j) + \gamma V_t(j) \right]$$

III^{ly} to stochastic version of value iteration of fixed policy.

TEMPORAL DIFFERENCE CONTROL :

- Estimate Q-value function and update based on estimated Q-values.

- Update rule for state-action value is,

$$Q_{t+1}(i,a) = Q_t(i,a) + \alpha \left(R_{t+1} + \gamma Q_t(j,b) - Q_t(i,a) \right)$$

- WKT, a deterministic policy can't be used (∵ all state-action values wouldn't get updated)

. Let 'μ' be a randomized policy (ε-greedy) corresponding to deterministic policy π.

. Update, $Q_\pi(i,a) \quad \forall\ i,a$

. Update, $\pi'(i) = \underset{a}{\operatorname{argmax}}\ Q_t(i,a)$

} SARSA algorithm

. Comparision Of TD learning and MC learning :

. TD learning :

Advantages: 1) Updating is online

2) Can be used directly for continuing task

3) Very helpful if underlying problem has Markov property.

(∵ we use recursion)

. Example :

. Prediction :

Episode 1 : A 0 B 0

2 : B 1

3 : B 1

4 : B 0

5 : B 1

6 : B 1

7 : B 1

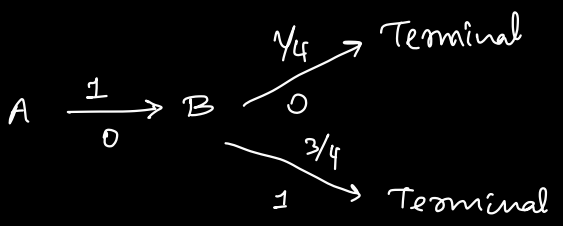
8 : B 1

MC Method :

- $V(A) = 0$
- $V(B) = 6/8$

TD Method :

This is the model the TD method has assumed



$\Rightarrow \hat{V}(B) = 3/4$

$\hat{V}(A) = 3/4$

11/3:

Recap:

- WKT in value iteration with $v_0 = 0$,
  - $v_k = T_\mu v_{k-1} \Rightarrow v_k \longrightarrow v_\mu$
  - $v_k = T v_{k-1} \Rightarrow v_k \longrightarrow v^*$
- Also, $((1-\alpha) I + \alpha T_\mu) v_k \longrightarrow v_\mu$
- TD Learning :

$$v_{t+1}(i) = v_t(i) + \alpha_t \left[R_{t+1} + \gamma v_t(j) - v_t(i) \right]$$

$$= (1-\alpha_t) v_t(i) + \alpha_t \left(R_{t+1} + \gamma v_t(j) \right)$$

- If multiple samples are allowed then we have,

$$V_{t+1}(i) = (1-\alpha_t) V_t(i) + \alpha_t \mathbb{E}\left[R_{t+1} + \gamma V_t(j) \right]$$

$\Downarrow$ in vector form

$$\left((1-\alpha_t) I + \alpha_t T_\mu \right) V_t$$

- Here, $\sum \alpha_t = \infty$, $\sum \alpha_t^2 < \infty$

  (Ex: $1/t$)

  $\sum \alpha_t = \infty$: To make sure that we can reach anywhere starting from anywhere

  $\sum \alpha_t^2 < \infty$: To make sure that there is no large deviation b/w consecutive time-steps.

- For $\alpha_t = 1/t$

$$\sum \alpha_t = 1 + 1/2 + 1/3 + \dots = O(\log t)$$

$$\sum \alpha_t^2 = 1 + 1/4 + 1/9 + \dots \to \pi^2/6$$

- $\sum \alpha_t^2 < \infty$ is to cancel the noise.

$\Rightarrow$ For later points we go slowly i.e $\alpha_t \downarrow$.

. SARSA is a on-policy algorithm ∵ the update is based on the current policy itself.

. Will $(1-\alpha)I + \alpha T \longrightarrow V^*$?

WKT, $V_{t+1} = \left((1-\alpha)I + \alpha T\right) V_t \longrightarrow V^*$ (Yes!)

. Consider the stochastic version of the above update :

$$\Rightarrow V_{t+1}(i) = (1-\alpha)\, V_t(i) + \alpha \left[\max_a \left(R(i,a) + \sum_{j=1}^{n} P_{ij}(a)\, V_t(j) \right) \right]$$

$\downarrow$ V^*

Here we cannot take a particular sample ∵ of the max operator.

. Now consider,

$$Q^*(i,a) = R(i,a) + \gamma \sum_{j \in S} P_{ij}(a) \max_b Q^*(j,b)$$

. ∴ The corresponding stochastic version is,

Q-Bellman equation

$$Q_{t+1}(i,a) = (1-\alpha)\, Q_t(i,a) + \alpha \left[R(i,a) + \gamma \max_b Q_t(j,b) \right]$$

$$Q_{t+1}(i,a) = Q_t(i,a) + \alpha \left[R(i,a) + \max_b Q_t(j,b) - Q_t(i,a) \right]$$

- $Q_t(i,a) \longrightarrow Q^*(i,a)$

- For the Q-learning the behavior policy is any randomized policy (Ex: ε-greedy policy)

- The target policy here is the stationary deterministic policy.

- ∴ Q-learning is a off-policy algorithm.

SUMMARY :

1) MC control — 2) SARSA — 3) Q-learning

For small state-action space when model info. is unknown

- Uses PI as the dynamic programming principle
- No bootstrapping
- Need to wait until the end of episode
- Mainly for episodic task
- Reliable even if Markov property is not present.

- Uses PI to improve policy but also VI to estimate the policy.
- Uses bootstrapping
- Online
- Can be used for continuous and episodic tasks
- Reliable if Markov property is present

- Uses VI ∵ of max. term in the update step.

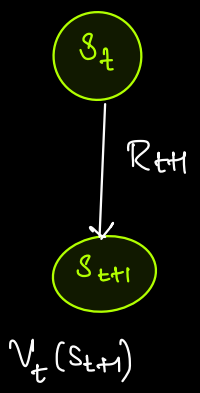
14/3 :

Recap :

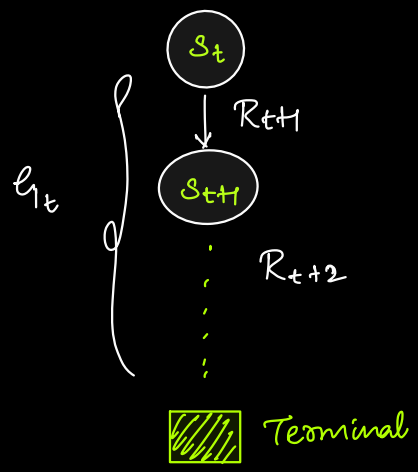
. TD prediction :

$$V_{t+1}(i) = V_t(i) + \alpha \underbrace{\left[R_{t+1} + \gamma V_t(j) - V_t(i) \right]}_{\delta_t \text{ (TD error)}}$$

. TD Backup :



. MC Backup :

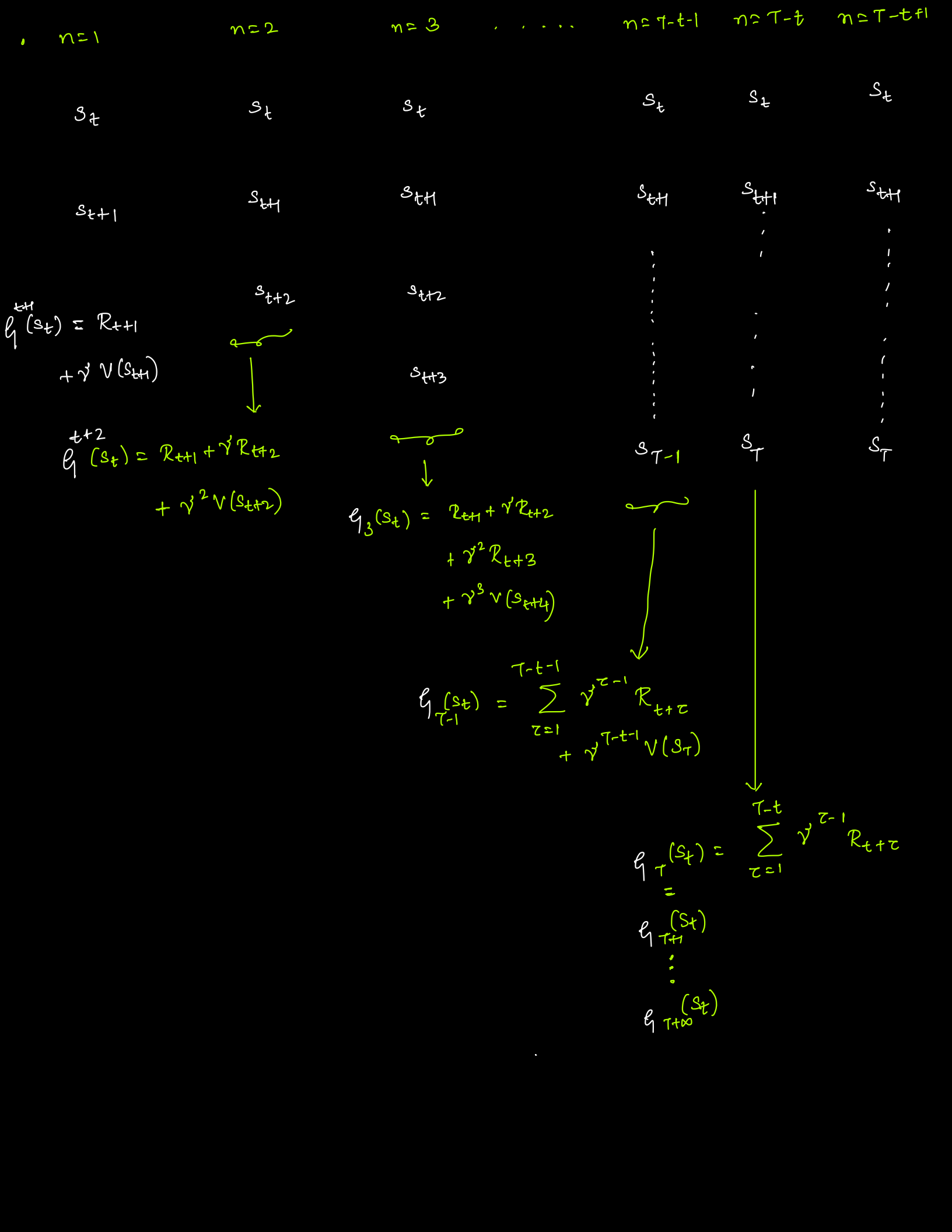


. n-step return :

$$V(S_t) = V(S_t) + \alpha \underbrace{\left(\sum_{\tau=t}^{t-n+1} \gamma^{t-\tau} \cdot R_{\tau+1} + \gamma^n V(S_{t+n}) - V(S_{t+1}) \right)}_{\delta_n(S_t)}$$

. Now consider a way to combine different n-step returns .

$$\Rightarrow V(S_t) = V(S_t) + \alpha \left[\sum_{n=1}^{\infty} w_n \delta_n(S_t) \right] \quad (\text{n}) \rightarrow$$


n=1
n=2
n=3
n=T-t-1
n=T-t
n=T-t+1
s_t
s_{t+1}
s_{t+2}
s_{t+3}
s_{T-1}
s_T
$q^{t+1}(s_t) = R_{t+1} + \gamma V(s_{t+1})$
$q^{t+2}(s_t) = R_{t+1} + \gamma R_{t+2} + \gamma^2 V(s_{t+2})$
$q_3(s_t) = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 V(s_{t+4})$
$q_{T-1}(s_t) = \sum_{\tau=1}^{T-t-1} \gamma^{\tau-1} R_{t+\tau} + \gamma^{T-t-1} V(s_T)$
$q_T(s_t) = \sum_{\tau=1}^{T-t} \gamma^{\tau-1} R_{t+\tau}$
$q_T(s_t) = q_{T+1}(s_t) = \vdots = q_{T+\infty}(s_t)$

- WKT, $\sum_{n=1}^{\infty} w_n = 1$

- Let $w_n = \lambda^n \Rightarrow \sum_{n=1}^{\infty} w_n = \frac{1}{1-\lambda}$

$$\Rightarrow (1-\lambda) \sum_{n=1}^{\infty} \lambda^n = 1$$

$$\therefore w_n = (1-\lambda)\lambda^n$$

$$\Rightarrow \underbrace{L_t}_{\text{Weighted return}} = (1-\lambda) \sum_{t=1}^{\infty} \lambda^{t-1} \, q_t^{t+n}$$

$$\Rightarrow L_t = (1-\lambda) \sum_{n=1}^{T-t-1} \lambda^{n-1} \, q_t^{t+n} + (1-\lambda) \sum_{n=T-t}^{\infty} \lambda^{n-1} \, q_t^{t+n} \quad \left.\right\} \text{For } \lambda < 1$$

$$= (1-\lambda) \sum_{t=1}^{T-t-1} \lambda^{n-1} \, q_t^{t+n} + \lambda^{T-t-1} \, q_t^T$$

- For $\lambda = 0$:

$$L_T = TD(0)$$

- For $\lambda \to 1$:

$$L_T \to MC = q_t^T \quad \left.\right\} \text{Every visit MC}$$

$\because V(s_t) = V(s_t) + \alpha \, q_t^T$

$\Rightarrow$ if $s_t = s$ and $s_{t+k} = s$

$\Rightarrow$ V(s) gets updated twice

NOTE :

. Offline Updating :

- Updates the value function estimates at the end of the trajectory. (store the correction / TD errors and update at the end)

. Online Updating :

- At every time step we update the value function estimates.

. To update the value estimate of a state we need to wait till the end of the trajectory.

. i.e we are looking at the forward states to update the value function of a state. Hence we call this as forward view or λ - return algorithm.

i.e $V_{t+1}(S_t) = V_t(S_t) + \alpha \left[L_t - V_t(S_t) \right]$

where, $L_t = (1-\lambda) \sum_{n=1}^{\infty} \lambda^{n-1} G_t^{t+n}$

practically tedious to implement !

BACKWARD VIEW (OR) TD(λ) ALGORITHM :

. The update rule is,

Online TD(λ) algorithm

$$V_{t+1}(s_t) = V_t(s_t) + \alpha_t \delta_t e_t(s_t)$$

where,

$$e_t(s) = \begin{cases} \gamma \lambda e_{t-1}(s) + 1 & ; \quad s = s_t \\ \gamma \lambda e_{t-1}(s) & ; \quad s \neq s_t \end{cases}$$

$$\delta_t = R_{t+1} + \gamma V_t(s_{t+1}) - V_t(s_t)$$

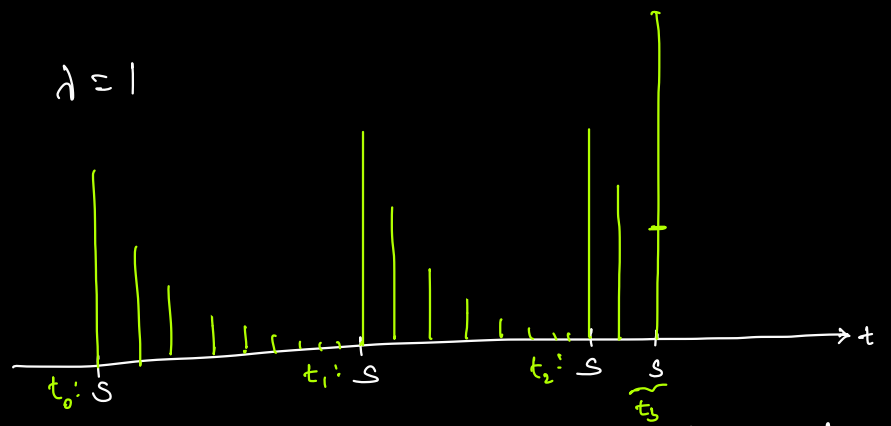
. For ~~Off~~line TD(λ),

. Let $\Delta(s) = \sum_{t=0}^{T} \delta_t e_t(s)$; $\forall$'s'

. How eligibility trace evolves ?

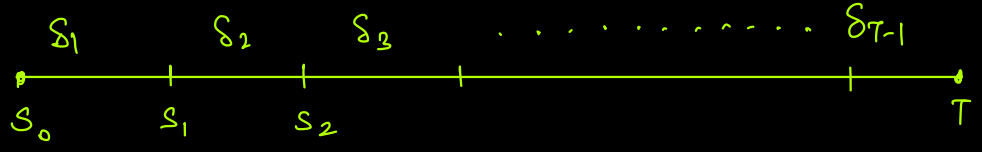
. Let $\lambda = 1$

⇒



Visit of state 's' at different time steps

- The offline update for $\lambda = 1$: (TD(1) algo.^{m})



$$\Delta(S_0) = \delta_1 e_1(S_0) + \delta_2 e_2(S_0) + \delta_3 e(S_0) + \cdots$$

$\delta_1 \rightarrow R_1 + \gamma V(S_1) - V(S_0)$

$\delta_2 \rightarrow R_2 + \gamma V(S_2) - V(S_1)$

$$\Rightarrow \Delta(S_0) = \underbrace{\left(R_1 + \gamma V(S_1) - V(S_0)\right)(1)}_{e_1(S_0)} \quad \text{: Assumption}$$

$$+ \left(R_2 + \gamma V(S_2) - V(S_1)\right)\gamma$$

$$+ \cdots$$

$$= R_1 + \gamma R_2 + \gamma^2 R_3 + \cdots$$

$$+ \gamma V(S_1) - \gamma V(S_1) + \gamma^2 V(S_2) - \gamma^2 V(S_2) + \cdots$$

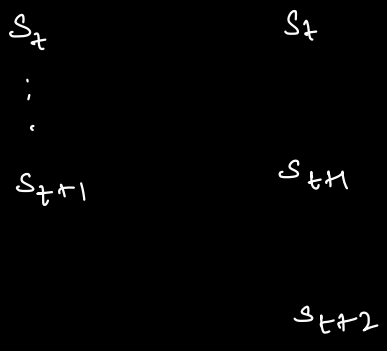
$$- V(S_0) + \gamma^{T-1} V(S_T)$$

$$\boxed{\Delta(S_0) = R_1 + \gamma R_2 + \gamma^2 R_3 + \cdots + \gamma^T V(S_T) - V(S_0)}$$

16/3:

- Recap :

1) Forward view : (λ - return algorithm)



EQUIVALENCE OF FORWARD VIEW AND BACKWARD VIEW :

. The TD error from eligibility trace update is,

$$\Delta V_t^{TD}(s) = \alpha S_t e_t(s)$$

where, $\delta_t = r_{t+1} + \gamma V(s_{t+1}) - V(s_t)$

$$e_t(s) = \gamma \lambda e_{t-1}(s) + \underbrace{I_{ss_t}}$$

$$= \begin{cases} 1 ; \text{if } s = s_t \\ 0 ; \text{o.w} \end{cases}$$

$$\Rightarrow e_t(s) = I_{ss_t} + \gamma\lambda \left(\gamma\lambda e_{t-2}(s) + I_{ss_{t-1}} \right)$$

$$= I_{ss_t} + \gamma\lambda I_{ss_{t-1}} + \gamma^2\lambda^2 \left(e_{t-3}(s) + I_{ss_{t-2}} \right)$$

$$= I_{ss_t} + \gamma\lambda I_{ss_{t-1}} + \gamma^2\lambda^2 I_{ss_{t-2}} + \gamma^3\lambda^3 I_{ss_{t-3}} + \dots\dots$$

$$\boxed{e_t(s) = \sum_{k=0}^{t} (\gamma\lambda)^{t-k} I_{ss_k}}$$

$\Rightarrow$ The sum of TD errors over the trajectory in backward view for a state 's' is,

$$\sum_{t=0}^{T-1} \Delta V_t^{TD}(s)$$

$$\Rightarrow \sum_{t=0}^{T-1} \Delta V_t^{TD}(s) = \sum_{t=0}^{T-1} \alpha \delta_t e_t(s)$$

$$= \sum_{t=0}^{T-1} \alpha \delta_t \left(\sum_{k=0}^{t} (\gamma\lambda)^{t-k} I_{ss_k} \right)$$

- Changing 't' with 'k' and 'k' with 't'

$$= \sum_{k=0}^{T-1} \alpha \delta_k \left(\sum_{t=0}^{k} (\gamma\lambda)^{k-t} I_{ss_t} \right)$$

Expanding terms;

$$= \alpha \Big[\delta_0 \left((\gamma\lambda)^0 I_{ss_0} \right)$$

$$+ \delta_1 \left((\gamma\lambda) I_{ss_0} + (\gamma\lambda)^0 I_{ss_1} \right)$$

$$+ \delta_2 \left(\gamma^2\lambda^2 I_{ss_0} + \gamma\lambda I_{ss_1} + (\gamma\lambda)^0 I_{ss_2} \right)$$

$$+ \delta_3 \left(\gamma^3\lambda^3 I_{ss_0} + \gamma^2\lambda^2 I_{ss_1} + \gamma\lambda I_{ss_2} + (\gamma\lambda)^0 I_{ss_3} \right)$$

$$+ \ldots \Big]$$

$$= \alpha \left[I_{ss_0} \sum_{k=0}^{T-1} (\gamma\lambda)^k \delta_k + I_{ss_1} \sum_{k=1}^{T-1} (\gamma\lambda)^{k-1} \delta_k + \cdots \right]$$

$$= \alpha \sum_{t=0}^{T-1} I_{ss_t} \sum_{k=t}^{T-1} (\gamma\lambda)^{k-t} \delta_k$$

$$\Rightarrow \sum_{t=0}^{T-1} V_t^{TD}(s) = \sum_{t=0}^{T-1} \alpha I_{ss_t} \sum_{k=t}^{T-1} (\gamma\lambda)^{k-t} \delta_k$$

— (1)

- From the forward view the update rule is,

$$\Delta V_t^{\lambda}(s_t) = \alpha \left[\underbrace{L_t^{\lambda}}_{\text{weighted sum of n-steps returns}} - V_t(s_t) \right]$$

$$\Rightarrow \frac{1}{\alpha} \Delta V_t^{\lambda}(s_t) = L_t^{\lambda} - V_t(s_t)$$

$$= -V_t(s_t)$$

$$+ (1-\lambda) \left[R_{t+1} + \gamma V_t(s_{t+1}) \right]$$

$$+ (1-\lambda)\lambda \left[R_{t+1} + \gamma R_{t+2} + \gamma^2 V_t(s_{t+1}) \right]$$

$$\vdots$$

$$= -V_t(s_t) \quad (1-\lambda) \underbrace{\sum_{k=t}^{\infty} \lambda^{k-t}}_{\frac{1}{(1-\lambda)}} R_{t+1} + (1-\lambda)\lambda\gamma \sum_{k=t}^{\infty} \lambda^{k-t} \cdot R_{t+2} + \dots$$

$$+ \gamma V_t(s_{t+1}) + \gamma^2 \lambda V_t(s_{t+2}) + \dots$$

$$- \lambda\gamma V_t(s_{t+1}) - \gamma^2\lambda^2 V_t(s_{t+2}) + \dots$$

$$= R_{t+1} + \gamma\lambda R_{t+2} + \dots$$

$$+ \gamma V_t(s_{t+1}) - V_t(s_t)$$

$$+ \gamma\lambda \left[\gamma V_t(s_{t+2}) - V_t(s_{t+1}) \right]$$

$$+ \gamma^2\lambda^2 \left[\gamma V_t(s_{t+3}) - V_t(s_{t+2}) \right]$$

$$+ \ldots\ldots$$

$$= (\gamma\lambda)^0 \underbrace{\left[R_{t+1} + \gamma V_t(s_{t+1}) - V_t(s_t) \right]}_{\delta_t}$$

$$+ (\gamma\lambda) \underbrace{\left[R_{t+2} + \gamma V_t(s_{t+2}) - V_t(s_{t+1}) \right]}_{\delta_{t+1}}$$

$$+ \ldots\ldots$$

$$= \sum_{k=t}^{\infty} (\gamma\lambda)^{k-t} \delta_k$$

(For offline update equality holds)

$$\Rightarrow \Delta V_t^{\lambda}(s_t) = \alpha \sum_{k=t}^{\infty} (\gamma\lambda)^{k-t} \delta_k$$

. For a state 's' the update over a trajectory under forward view is,

$$\sum_{t=0}^{T-1} \Delta V_t^{\lambda}(s) \, I_{ss_t} = \sum_{t=0}^{T-1} \alpha \sum_{k=t}^{\infty} (\gamma\lambda)^{k-t} \delta_k \cdot I_{ss_t}$$

- Since the rewards are zero $\forall\ k > T$

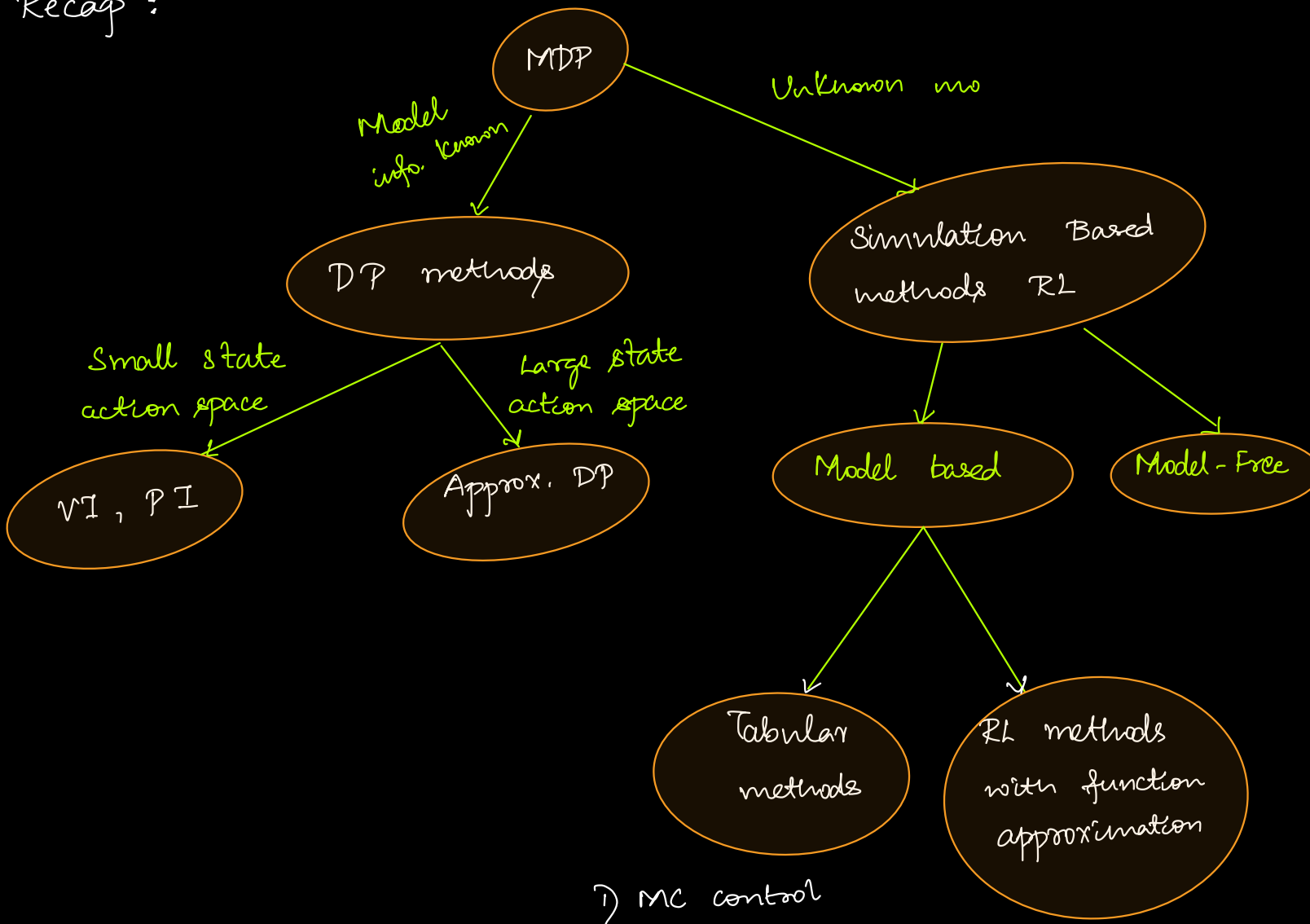
$$\Rightarrow \sum_{t=0}^{T-1} \Delta V_t^{\lambda}(s)\, I_{ss_t} = \sum_{t=0}^{T-1} \alpha \sum_{k=t}^{T-1} (\gamma\lambda)^{k-t}\, \delta_k\, I_{ss_t} \quad - (2)$$

- From (1) and (2)

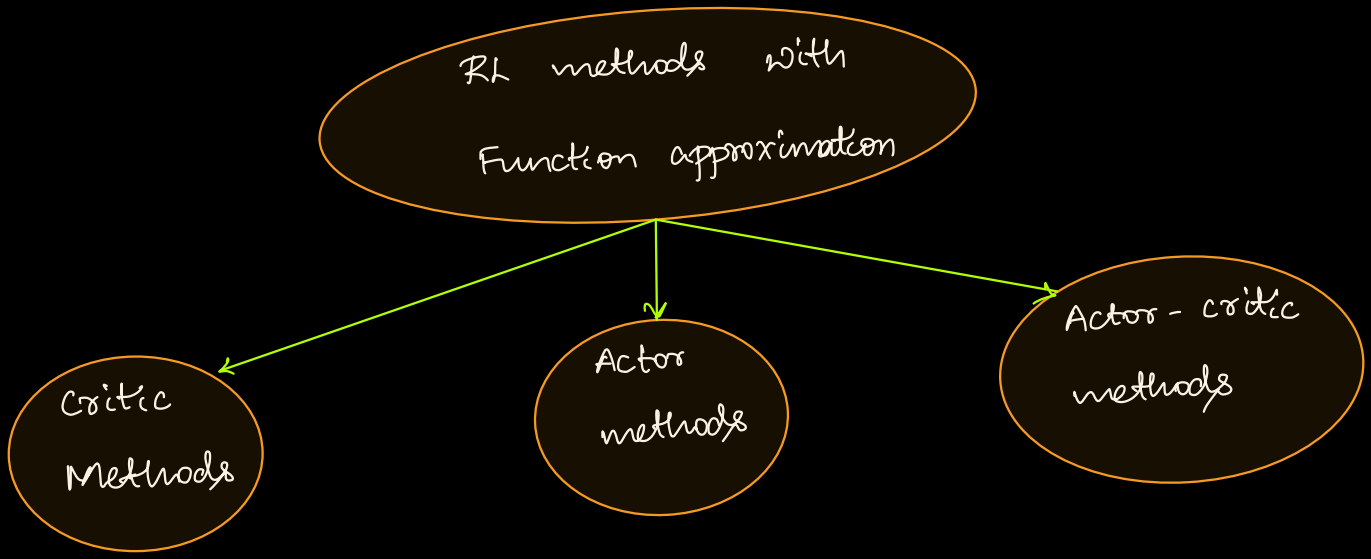
$$\sum_{t=0}^{T-1} \Delta V_t^{TD}(s) = \sum_{t=0}^{T-1} \Delta V_t^{\lambda}(s)\, I_{ss_t} \;;\ \forall\ s \in S$$

18/3 :

Recap :



1) MC control
2) SARSA
3) Q-learning



. What functions we can approximate ?

1) Value functions

2) Policy

VALUE - BASED FUNCTION APPROXIMATION :

. Why approximation ?

. Due to large - state space we can't store $V(s)$ $\forall s$ and computing $V(s)$ $\forall s$ is computationally expensive.

. How are we storing for continuous state space.

Ex: T —(-1)— A —(0)— B —(0)— C —(0)— D —(0)— E —(1)— T

- One trajectory is C D E D E T

- $V(E) = \frac{1}{2} \cdot 1 + \frac{1}{2} V(D)$

$V(D) = \frac{1}{2} V(E) + \frac{1}{2} V(C)$

$V(C) = \frac{1}{2} V(B) + \frac{1}{2} V(D)$

$V(B) = \frac{1}{2} V(C) + \frac{1}{2} V(A)$

$V(A) = \frac{1}{2} \cdot (0) + \frac{1}{2} V(B)$

$\Rightarrow \quad V_B = 2 V_A$

- A linear approximation for a fixed policy

$$\hat{V}_\mu(s, w) = \sum_{i=1}^{k} w_i \phi_i(s, w)$$

k → # of features we considered

ϕ_i → Basis function (can be polynomial | RBF)

- Our goal is $\min \sum_{s \in S} \left(V_\mu(s) - \hat{V}_\mu(s) \right)^2$

gives equal importance $\forall s \in S$

. We give more weightage to the states that are visited often.

$$\Rightarrow \text{Goal}: \quad \min \sum_{s \in S} d_\mu(s) \left(V_\mu(s) - \hat{V}_\mu(s) \right)^2$$

$d_\mu(s)$ ↓ Stationary dist^n. of state 's'

. For linear approximator,

$$J(w) = \sum_{s \in S} d_\mu(s) \left(V_\mu(s) - \phi(s)^T w \right)^2$$

where, $\phi(s) = \left(\phi_1(s), \phi_2(s), \ldots, \phi_k(s) \right)$

. Goal : $\min_w J(w)$

. Using gradient descent :

$$-2 \sum_{s \in S} d_\mu(s) \left(V_\mu(s) - \phi(s)^T w \right) \phi(s)$$

$$\Rightarrow \quad w = w + \alpha \cdot 2 \cdot \sum_{s \in S} d_\mu(s) \left(V_\mu(s) - \underbrace{\phi(s)^T w}_{\hat{V}_\mu(s)} \right) \phi(s)$$

- The stochastic update rule is,

$$w_{n+1} = w_n + \alpha \left[\underbrace{(V_\mu(s)}_{\text{Unknown}} - \hat{V}(s,w)) \, \phi(s) \right]$$

- MC function Approximation :

$$w_{n+1} = w_n + \alpha \left[G_t(s) - \hat{V}(s,w) \right] \phi(s)$$

- TD(0) Function Approximation :

$$w_{n+1} = w_n + \alpha \left[R_{t+1} + \gamma \hat{V}(s',w) - \hat{V}(s,w) \right] \phi(s)$$

- TD(λ) Function Approximation :

- $w_{n+1} = w_n + \alpha \left[R_{t+1} + \gamma \hat{V}(s',w) - \hat{V}(s,w) \right] e_t$

where, $e_t = \gamma \lambda e_t + \phi(s_t)$