Cs 747 foundations of intelligent and learning agents

G. Akash Reddy

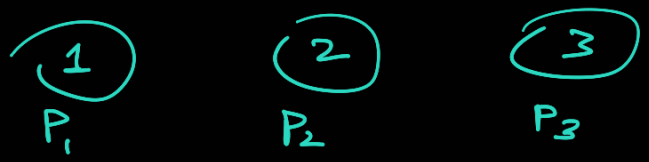
3 coding As - 3×15 = 45

midsem- 20

endsem- 35

→ Multi-armed bandits:-

coins example...



Q: get more heads in 20 tosses.

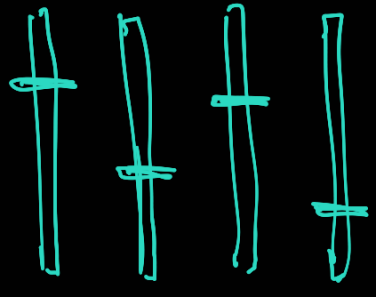
explore – exploit balance....

- explore coins to estimate p_1, p_2, p_3
- exploit the highest bias one.

- stochastic multi-armed bandit:-

monte-carlo tree search

whats this?



→ bandit instance

- n-arms each a bernoullie RV.
- A be set of arms. $a \in A$ has mean p_a.
- let p^* be highest mean.

→ Algorithm:-

in case of bandits:

for $t = 0, 1, 2, 3, \ldots T-1$:-

- say history $h^t = (a^0, r^0, a^1, r^1, a^2, r^2, \ldots a^{t-1}, r^{t-1})$
- pick an arm a^t to "pull."
- obtain reward r^t.

a^i → arm no. for $t=i$

$r^i \in \{0, 1\}$

↳ reward for $t=i$

* $T \rightarrow$ budget or "horizon" of experiment

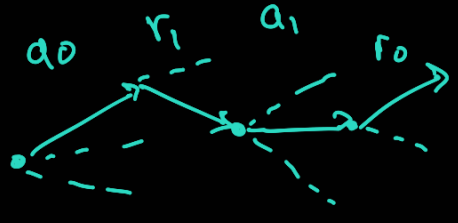
* Formally, a deterministic algorithm is a mapping
  - from set of all histories
  - to set of all arms.

  (i.e. based on history, decides which arm to pull deterministically)

* Formally, a randomised algorithm is a mapping
  - from set of all histories
  - to set of "all probability distributions over arms".

  (i.e. based on history; allots probability to pull, for each arm).

* deterministic algorithm $\longrightarrow$ 2^T histories possible.



$$h^T = (a^0, r^0, a_1, r_1 \cdots a_{t-1}, r_{t-1})$$

$$P(h^T) = \prod_{t=0}^{T-1} P(a^{t-1} | h^{t-1}) \cdot P(r^{t-1} | a^{t-1})$$

algorithm — bandit instance

→ **ϵ-greedy algorithms :-**

$\epsilon \in [0,1]$ controls the amount of exploration.

ϵG_1 :

$t < \epsilon T$; sample arms randomly

at $t = \lfloor \epsilon T \rfloor$ identify a^{best} with highest mean.

$t > \lfloor \epsilon T \rfloor$ pull a^{best}

ϵG_2 :

if $t < \epsilon T$; sample randomly

if $t > \epsilon T$; pull highest empirical mean.

ϵG_3 :

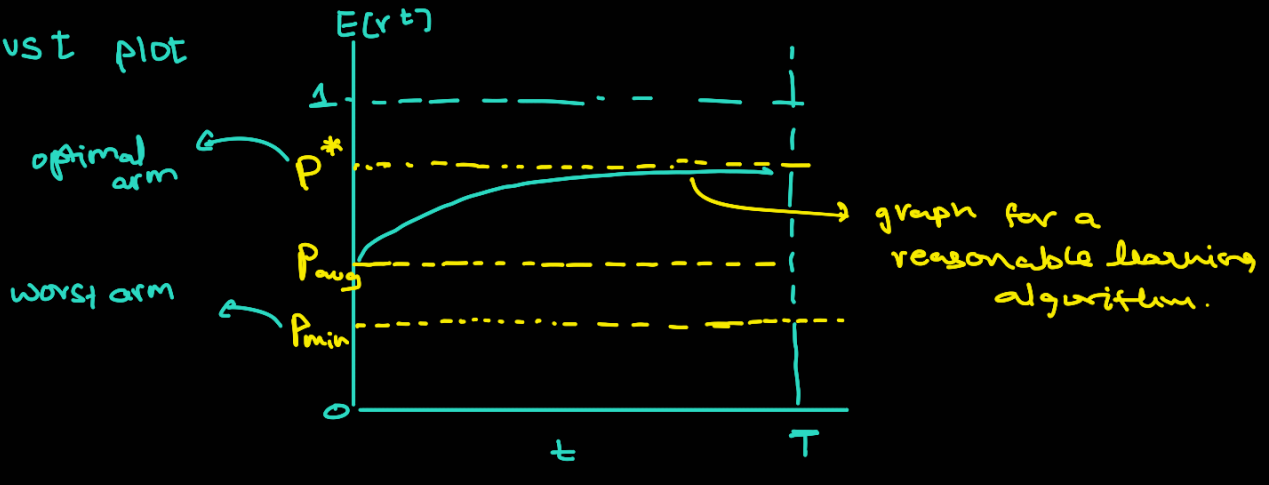
probability ϵ → pull randomly any arm

probability $(1-\epsilon)$ → pull highest mean one.

* Does ϵG_1 perform better than ϵG_2? what is a "better" algorithm?

→ **Visualizing performance :-**

$E[r^t]$ vs t plot



* consider a arm; which pulls each arm at random.

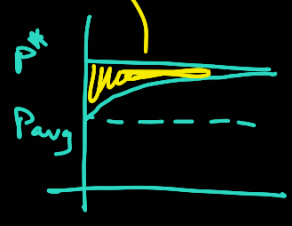
Hence $E[r^t] = \text{avg}\{p_1, p_2, \cdots p_n\}$ for any $t \leq T$

* Maximum reward for period T is Tp^*.

* Actual expected reward $= \sum_{t=0}^{T-1} E[r^t]$

* (Expected Cumulative) Regret of algorithm for horizon T is

$$R_T = Tp^* - \sum_{t=0}^{T-1} E(r^t)$$



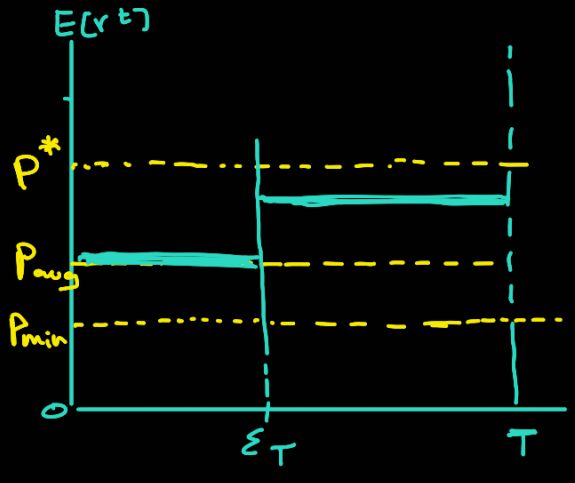
* We want R_T to be smaller. Infact $\lim_{T\to\infty} \frac{R_T}{T} = 0$.

"sub-linear" regret

we want algorithms which give sub-linear regret.

"don't all of them do that?"

→ EG1 algo:



$\therefore R_T = (p^* - P_{avg})\varepsilon T + \text{some value}$

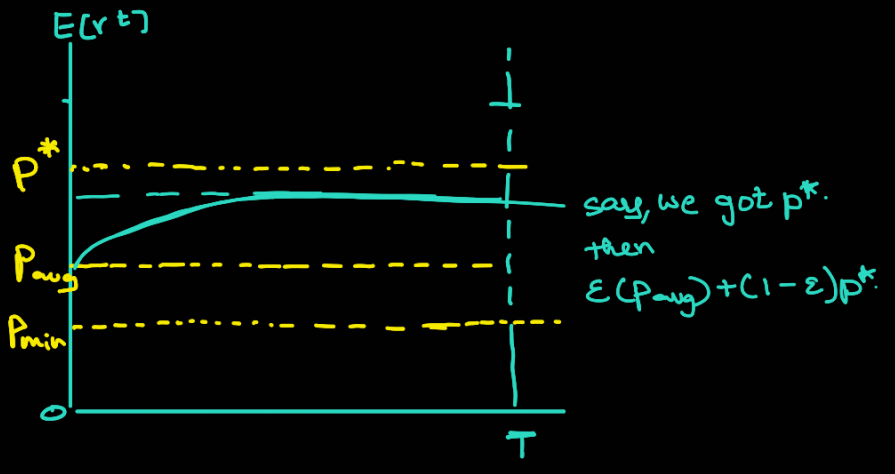
$\therefore \lim_{T\to\infty} R_T = (p^* - P_{avg})\varepsilon \neq 0$

Not sub-linear. | Same analysis for EG12

→ EG3:

uniformly with ε

exploit with $(1-\varepsilon)$



$$\therefore \quad \frac{R_T}{T} \geq p^* - \left[\varepsilon(p_{avg}) + (1-\varepsilon)(p^*)\right]$$

$$\geq \varepsilon(p^* - p_{avg})$$

$$\therefore \quad \lim_{T\to\infty} \frac{R_T}{T} \geq \underline{\varepsilon(p^* - p_{avg})}$$

linear regret.

Not sub-linear regret.

→ How to achieve sub-linear regret?

* Two conditions - C_1 and C_2

* C_1 : Infinite Exploration

* In the limit $T \to \infty$, each arm must be pulled an ∞ no. of times.

Explanation: ➢ Say an arm is at max pulled 'U' times,

➢ with a (slight) probability of $(1-p^*)^U$; the most optimal arm will have empirical mean 0.

$$\therefore \text{ The Regret } \frac{R^T}{T} \geq \underbrace{(1-p^*)^U (p^* - p_2)}_{\text{not going to 0 with } T\to\infty}$$

* to avoid this slightly probable misfortune; we need to explore endlessly.

* C_2 : Greed in the limit

* Also have to exploit enough. let exploit(T) denote no. of pulls that are greedy wrt. empirical mean...

$$\lim_{T \to \infty} \frac{E[\text{exploit}(T)]}{T} = 1$$

* let $\bar{I}$ be the set of all bandit instances with reward-means strictly less than 1,

> **RESULT:**
> An algorithm L achieves sub-linear regret on all instances $I \in \bar{I}$ if and only if it has C_1 and C_2 properties on all $I \in \bar{I}$.

GL IE conditions, greedy in limit & infinite exploration.

EG_1, EG_3 → both have infinite xploration.
→ don't have greedy in the limit.
(both are $(1-\varepsilon)$ in limit)

→ Making our EG_1 as sub-linear regret:-

* say $\varepsilon_T = \frac{1}{\sqrt{T}}$. nice! ($\varepsilon_T$ - first algorithm)

then $\varepsilon_T \cdot T = \sqrt{T}$ → infinite times exploration

& $\frac{\text{Explait}(T)}{T} = 1 - \frac{1}{\sqrt{T}}$ → 1 as $T \to \infty$
GL.

* For ϵ_T-greedy

here, say $\epsilon_t = \frac{1}{1+t}$; $\frac{1}{(1+t)^2}$ won't work! IE fails I think.

∴ at time t; ϵ_t exploration & $1-\epsilon_t$ exploitation.

* Checkin C_1: each arm gets $\sum_{t=0}^{T-1} \frac{1}{n(1+t)} = \Theta\left(\frac{\log T}{n}\right)$ pulls ... infinite!

* Checking C_2: since exploration is $\sim \log(T)$

$$\frac{\text{exploit}(T)}{T} = 1 - \frac{\log(T)}{T} \to 1 \text{ as } T \to \infty$$

∴ GL holds!

→ Lower bound on regret:-

Lai and Robbins lower bound on regret

Let L be an algorithm such that for every $I \in \bar{I}$ and for every $\alpha > 0$, as $T \to \infty$

$$R_T(L, I) = O(T^\alpha) \quad \text{i.e. sub-polynomial } \because \alpha > 0$$

Then for every instance $I \in \bar{I}$, as $T \to \infty$:

$$\frac{R_T(L,I)}{\ln(T)} \geq \sum_{a: P_a \neq P^*} \frac{P^*(I) - P_a(I)}{KL(P_a, P^*)}$$

where for $x, y \in [0,1]$

$$KL(x,y) \stackrel{\text{def}}{=} x \ln \frac{x}{y} + (1-x) \ln \frac{1-x}{1-y}$$

→ UCB - upper confidence bounds sampling :-

* at time t; define $\boxed{UCB_a^t = \hat{p}_a^t + \sqrt{\frac{2\ln t}{u_a^t}}}$

$\hat{p}_a^t$: empirical mean

u_a^t : no. of times this arm a was sampled.

- pull each arm once first.

* pull the arm with the highest UCB value.

* Achieves regret $O(\log(T))$. optimal dependency.

but factor is not exactly optimal.

→ KL UCB:

* Identical to UCB algorithm; but definition is different!

$$UCB\text{-}kl_a^t = \max q \in [\hat{p}_a^t, 1) \text{ s.t. } u_a^t \cdot KL(\hat{p}_a^t, q) \leq \ln t + c \cdot \ln(\ln t)$$

$c \geq 3$

- pull arm with max KL-UCB value.

$$\boxed{KL(x,y) \overset{def}{=} x \ln \frac{x}{y} + (1-x) \ln \frac{1-x}{1-y}}$$

* KL-UCB is tighter bound than UCB.

KL-UCB asymptotically matches Lai-Robbins bound.

nice!

so, bandits is completely solved...

atleast this particular model.

→ Thompson Sampling :-

* Beta distributions :-

$$\beta(x;\alpha,\beta) = \frac{(\alpha+\beta-1)!}{(\alpha-1)!\,(\beta-1)!}\, x^{\alpha-1}(1-x)^{\beta-1} \qquad x \in [0,1]$$

$$\text{variance} = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)} \qquad \text{mean} = \frac{\alpha}{\alpha+\beta}$$

* $\beta(1,1)$ is uniform distribution.

* At time t, let arm a have s_a^t successes & f_a^t failures.

$\beta(s_a^t+1, f_a^t+1)$ represents the belief of mean of arm a at time t.

* Sample a value for arm-a from this distribution.

* pull the arm with the highest sampling value.

* Achieves optimal regret ; excellent in practice.
Lai & Robbins lower bound.

→ Hoeffding's inequality

X → random variable $[0,1]$; $E(X) = \mu$

$x_1, x_2 \cdots x_n$ are iid of X.

* $\bar{x} = \frac{1}{n} \sum x_i$

then

$$P(\bar{x} > \mu + \varepsilon) \leq e^{-2n\varepsilon^2}$$
$$P(\bar{x} < \mu - \varepsilon) \leq e^{-2n\varepsilon^2}$$

Applications in slides

→ KL-inequality :-

* Same setting as before.

$$P\{\bar{x} > \mu + \varepsilon\} \leq e^{-nKL(\mu+\varepsilon, \mu)}$$
$$P\{\bar{x} < \mu - \varepsilon\} \leq e^{-nKL(\mu-\varepsilon, \mu)}$$

tighter than hoeffdings inequality.

* Using above results; proof that UCB is $O(\log T)$.
discussed in slides

asked in midsem... { "Analysis of UCB"

* Analysis of Thompson Sampling has basics in Bayesian inference.

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

* the logic to select which arm to pull is justified as

"The Beta distributions reflect our beliefs about the arms. Collectively, they represent our beliefs about the bandit instance.

We are sampling a bandit instance & pulling optimal arm wrt. that instance".

(Nice one!)

→ Markov Decision Problems MDPs :-

* Elements of an MDP $M = (S, A, T, R, \gamma)$

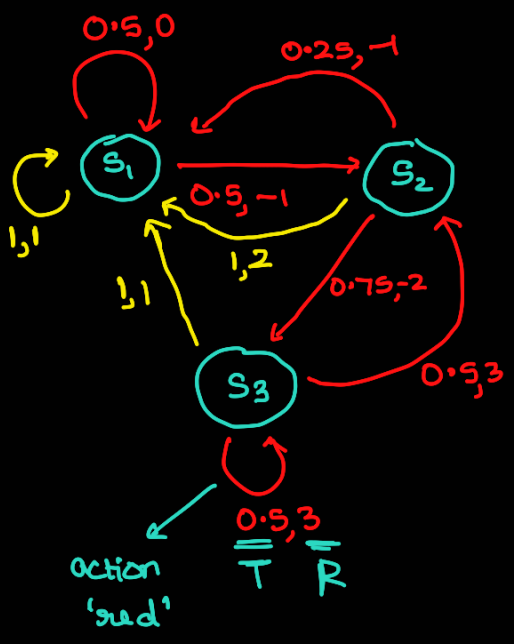
S : states

A : Actions

T : transition probabilities

R : rewards

γ : discount factor..? $\in [0, 1]$



$T(s, a, s') \in [0, 1]$

$R(s, a, s') \in [-R_{max}, R_{max}]$

Here $S = \{s_1, s_2, s_3\}$

$A = \{$'red', 'yellow'$\}$

T, S are on every edge.

* how does agent pick a^t?

In principle, it can decide by looking at the history:

$s^0, a^0, r^0, s^1, a^1, r^1, s^2, \ldots$

For now, assume a^t depends only on s^t. (markovian! stationary → const. with time. deterministic → not stochastic)

→ State values for policy π :

for $s \in S$, $V^{\pi}(s) \stackrel{def}{=} E_{\pi}[r^0 + r^1 + r^2 + \ldots], s^0 = s$

single action leads to single state; not probability over states.

But, this sum goes ∞ might

Hence, use discounting factor

$$V^{\pi}(s) \stackrel{def}{=} \underset{\pi}{E}\left[r^0 + \gamma r^1 + \gamma^2 r^2 + \gamma^3 r^3 + \cdots\right] \quad s^0 = s$$

larger the γ, farther the 'lookahead' while considering reward.

→ MDP planning :-

Given $M = (S, A, T, R, \gamma)$; find a policy π^* from the set of all policies Π such that $\forall s \in S$; $\forall \pi \in \Pi$

$$V^{\pi^*}(s) \geq V^{\pi}(s)$$

* Every MDP is guaranteed to have atleast one such markovin, deterministic, stationary optimal policy.

* Can have more than one optimal policy but only one unique optimal value function.

→ Policy evaluation: Finding $V^{\pi}(s)$ from π :-

$$V^{\pi}(s) = E_{\pi}\left(r^0 + \gamma r^1 + \gamma^2 r^2 + \cdots\right)$$

$$\boxed{V^{\pi}(s) = \sum_{s' \in S} T(s, \pi(s), s') \left(R(s, \pi(s), s') + \gamma V^{\pi}(s')\right)}$$

Bellman equations

now we have 'n' such equations
& 'n' unknowns

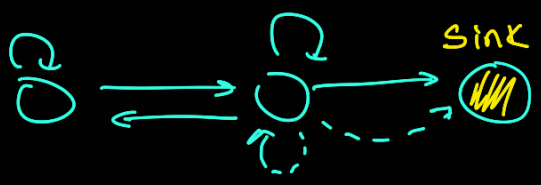
* Guaranteed to have unique solution if $\gamma < 1$.

* Using this, we iterate over all policies to find the optimal policy.

Exponential !

lets see better algorithms.

→ Episodic tasks :-



we considered infinitely long tasks.

* Now; episodic tasks have special sink states from which there are no outgoing tasks or rewards.

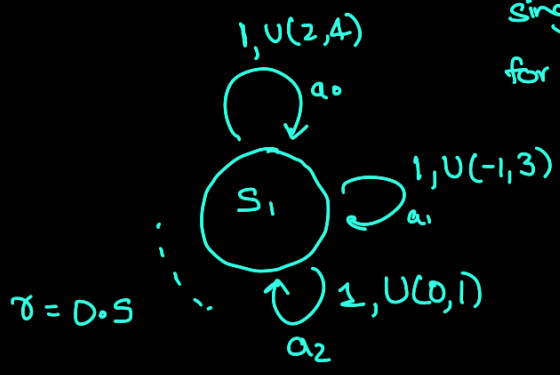
* Additionally; from every non-terminal state, for every policy; there is non-zero probability of reaching the sink in finite time.

* here; we can talk about net rewards; without discounting future rewards

$$V^{\pi}(s) \stackrel{def}{=} \mathop{\mathbb{E}}_{\pi}[r_0 + r_1 + \cdots] \mid s^0 = s$$

will be finite for episodic tasks.

* A familiar MDP?



single state; K-actions

for $a \in A$, treat $R(s,a,s')$ as a random variable

such an mdp → multi armed bandit.

→ Complete, normed vector spaces :-

* A norm $\|\cdot\|$ associates a length with each vector.

* A complete, normed vector space - $(X, \|\cdot\|)$ is one in which every cauchy sequence has a limit in X.
no need to contemplate

* A complete, normed vector space is called a Banach space.

→ **Definitions:** Say $(X, \|\cdot\|)$ is a normed vector space & $0 \leq \ell < 1$.

Contraction mapping:- A mapping $Z: X \to X$ is called a contraction mapping with contraction factor ℓ; if $\forall U, V \in X$

$$\|ZU - ZV\| \leq \ell \|V - U\|$$

Fixed point: $x^* \in X$ is called fixed point of Z if

$$Zx^* = x^*$$

→ **Banach's fixed point theorem:-**

* $(X, \|\cdot\|)$ be a banach space
* $Z: X \to X$ be contraction mapping with $\ell \in [0, 1)$

1. Z has unique fixed point $x^* \in X$
2. for $x \in X$; $m \geq 0$ $\|Z^m x - x^*\| \leq \ell^m \|x - x^*\|$

"finally; all points collapse onto fixed point".

→ **Bellmann Optimality operator:-**

* $S = \{s_1, s_2, s_3 \cdots\}$ states of an MDP.
* $F: S \to \mathbb{R}$ } could be MDP value function is a point in $\mathbb{R}^n$. (fine!)

$B^*: \mathbb{R}^n \to \mathbb{R}^n$ for MDP (S, A, T, R, γ) is

Bellmann optimality operator

point in $\mathbb{R}^n$; $S \to \mathbb{R}$.

$$(B^*(F))(s) \underset{\text{def}}{=} \max_{a \in A} \sum_{s' \in S} T(s, a, s') \{R(s, a, s') + \gamma F(s')\}$$

* $\|F\|_\infty = \max\{|f_1|, |f_2| \cdots |f_n|\}$ this is called max norm $\|\cdot\|_\infty$

* $(\mathbb{R}^n, \|\cdot\|_\infty)$ is a banach space. Established !

Fact:

> B^* is contraction mapping in $(\mathbb{R}^n, \|\cdot\|_\infty)$ Banach space with contraction factor γ.

* Proving B^* is a contraction mapping:-

we use $\left|\max_a f(a) - \max_a g(a)\right| \leq \max_a |f(a) - g(a)|$

* we can show $\|B^*(F)(s) - B^*(G)(s)\|_\infty \leq \gamma \|F(s) - G(s)\|_\infty$

→ The fixed point of B^* :-

* there is a unique fixed point of B^*.

Denote the fixed point as $V^*: S \to \mathbb{R}$ & so $B^*(V^*) = V^*$

⇒ for $s \in S$

$$V^*(s) = \max_{a \in A} \sum_{s' \in S} T(s,a,s') \left(R(s,a,s') + \gamma V^*(s') \right)$$

these are bellmann optimality equations.

n variables, n equations but non-linear

1. Value iteration
2. Linear Programming
3. Policy Iteration

} 3 family of algorithms to compute V^*

Fact:

$V^*(s)$ is the value function of every policy $\pi^*: S \to A$ such that for $s \in S$

$$\pi^*(s) = \underset{a}{\operatorname{argmax}} \sum_{s' \in S} T(s,a,s') \left(R(s,a,s') + \gamma V^*(s')\right)$$

1) Value iteration:-

* Iterative approach to compute V^*.

$V_0 \xrightarrow{B^*} V_1 \xrightarrow{B^*} \cdots V^*$ (Awesome lol!)

* Popular, easy to implement, quick to convergence

2. Linear Programming:-

* To solve real valued $x_1, x_2, \cdots x_n$;
  - given linear function is maximised
  - given linear constraints are satisfied.

* Modern linear programming solvers can solve LPs with thousands/millions of variables/constraints in reasonable time (hours/days)
  - common approaches: Simplex, interior-point methods

* Engineer's focus is on formulating, rather than solving LPs.

MDP planning using linear programming:-

1) Bellmann optimality equations : $s \in S$

$$V^*(s) = \max_{a \in A} \sum_{s' \in S} T(s,a,s') \{R(s,a,s') + \gamma V^*(s')\}$$

non linear.

We make it linear as: constraints

$$\boxed{V(s) \geq \sum_{s' \in S} T(s,a,s') \{R(s,a,s') + \gamma V^*(s')\}} \quad \forall s \in S, \ \forall a \in A$$

n.k linear constraints !

* Note that V^* is in the feasible set.
* Now; construct an objective function; so that V^* is the sole optimizer.

Digression:-

Vector Comparision:-

* for $x: S \to \mathbb{R}$ and $Y: S \to \mathbb{R}$; we define

$$X \geq Y \iff \forall s \in S \quad X(s) \geq Y(s)$$
$$X > Y \iff X \geq Y \text{ and } \exists s \in S : X(s) > Y(s)$$

* And naturally, for policies π_1, π_2 we define

$$\pi_1 \geq \pi_2 \iff V^{\pi_1} \geq V^{\pi_2}$$

$$\pi_1 > \pi_2 \iff V^{\pi_1} > V^{\pi_2}$$

* Note: we can have two incomparable policies.

$\pi_1 \geq \pi_2$ and $\pi_2 \geq \pi_1 \iff V^{\pi_1} = V^{\pi_2}$

* $B^*()$ preserves $\geq$:

Fact: for $X: S \to \mathbb{R}$
$Y: S \to \mathbb{R}$

$$X \geq Y \implies B^*(X) \geq B^*(Y)$$

proof can be done-

$$\max_a f(a) - \max_a g(a) \geq \min_a (f(a) - g(a))$$

✓✓

* Lets see the feasible set of our LP:-

Each $V: S \to \mathbb{R}$ in our set satisfies $V \geq B^*(V)$
$\Rightarrow V \geq B^{*2}(V) \geq B^{*3}(V) \cdots \geq V^*$

$\therefore$ each V in the feasible set $V \geq V^*$

we linearise this as:

$$\sum_{s \in S} V(s) \geq \sum_{s \in S} V^*(s)$$

Hence; we need to minimize $\sum_{s \in S} V(s)$.

$\Rightarrow$

maximize: $-\sum_{s \in S} V(s)$

Subject to

$$V(s) \geq \sum_{s' \in S} T(s,a,s') \{R(s,a,s') + \gamma V(s')\} \quad \forall s \in S \;\; \forall a \in A$$

our LP formulation is complete!

→ Policy Iteration:-

→ Action value function:-

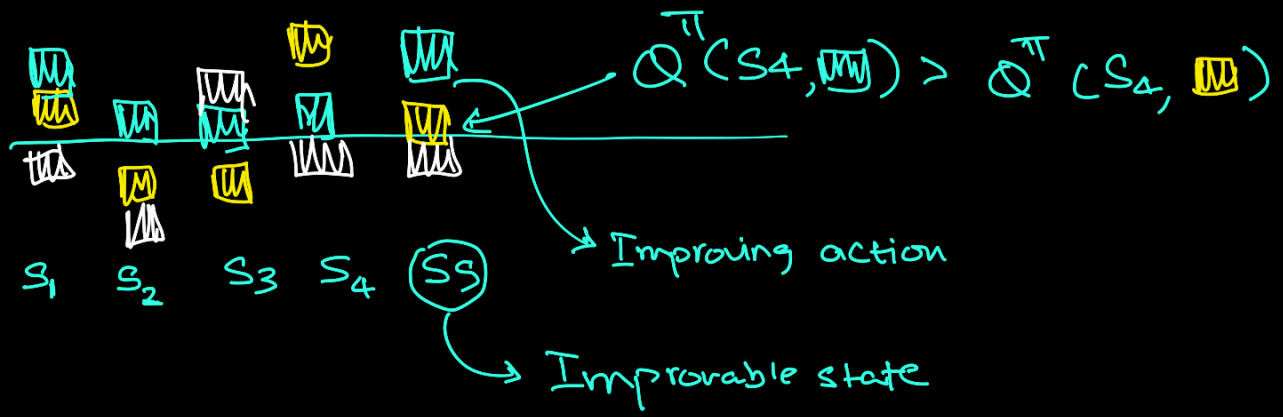
$$Q^{\pi}(s,a) \overset{def}{=} E\left[r^0 + \gamma r^1 + \gamma^2 r^2 + \cdots\right] \mid s^0 = s, a^0 = a, a^t = \pi(s^t) \text{ for } t \geq 1$$

$Q^{\pi}: S \times A \to \mathbb{R}$ is called action value function of π.

$$\boxed{Q^{\pi}(s,a) = \sum_{s' \in S} T(s,a,s')\left\{R(s,a,s') + \gamma V^{\pi}(s')\right\}}$$

* All optimal policies have same optimal action value functions
  - expected reward, starting from state s & playing action a, & following policy π from next turn.

→ Policy Improvement:-



Policy Improvement:

Given π;

- pick one or more improvable states & in these
- switch to an arbitrary improving action

$\pi_{new} > \pi_{old}$

→ Policy Improvement Theorem:-

$$V^{\pi}(s) \equiv Q^{\pi}(s, \pi(s))$$

* for $\pi \in \Pi$, $s \in S$;

$$IA(\pi, s) \stackrel{def}{=} \{ a \in A : Q^{\pi}(s,a) > V^{\pi}(s) \}$$

* for $\pi \in \Pi$,

$$IS(\pi) \stackrel{def}{=} \{ s \in S : |IA(\pi, s)| \geq 1 \}$$

* Suppose $IS(\pi) \neq \phi$ & $\pi' \in \Pi$ is obtained by policy improvement on π;

Policy Improvement Theorem;

1) if $IS(\pi) = \phi$; then π is optimal. Else
2) if π' is obtained by policy improvement on π, $\pi' \succ \pi$

→ Bellmann operator (NO optimality):-

* for policy π; we define $B^{\pi} : \mathbb{R}^n \to \mathbb{R}^n$ as:

$$(B^{\pi}(x))(s) \stackrel{def}{=} \sum_{s' \in S} T(s, \pi(s), s') \{ R(s, \pi(s), s') + \gamma x(s') \}$$

no max over a

Facts:

1. B^{π} is a contraction mapping with γ. like B^{*}.

2. $\lim_{l \to \infty} (B^{\pi})^{l}(x) = V^{\pi}$

3. $X \geq Y \Rightarrow B^{\pi}(X) \geq B^{\pi}(Y)$

Note: for $\pi, \pi' \in \Pi$ & $s \in S$

$$\boxed{B^{\pi'}(V^{\pi})(s) = Q^{\pi}(s, \pi'(s))}$$

⟶ Policy iteration Algorithm:-

$\pi \leftarrow$ arbitrary policy
while π has I.S.
  $\pi' \leftarrow$ policy improvement(π)
  $\pi \leftarrow \pi'$
return π

* no. of iterations depend on the switching strategy.
howards iteration
→ choose all I.S.

→ A more general class of policies:-

* till now, we only saw

markovian, deterministic, stationary polices.

why not: history dependent? Stochastic? non-stationary?

> FACT: NO such non-markovian, stochastic, non-stationary policy can out-perform our V^*

→ Different Policy iterations:-

1) Howard's policy iteration:-

Greedy → improve all improvable states

2) Random policy iteration:-

include / exclude a state → uniformly decided

3) Simple iteration policy:-

- Assume an indexing of states
- Improve highest index state

Upper & lower bounds -

> Analysis
> Basic tools
> open problems

} refer lecture slides

* Now; what if T, R were not given, but had to be learned from interaction?
Can we still learn to act optimally?

yes! the reinforcement learning problem.

→ Monte-Carlo methods:- ??

monte-carlo integration?

Reinforcement learning:-

1st class - intro : Tue before midsem...

2nd class - missed - Friday after midsem