Lecture 2

1) A State S_t is markov if and only if

$$P[S_{t+1}/S_t] = P[S_{t+1}/S_1, S_2, \dots S_t]$$

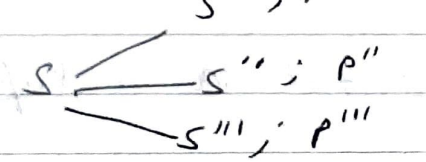
we dont need to know S_1 to S_{t-1} to understand $P[S_{t+1}]$. S_t is sufficient if it is an MDP

→ State Transition matrix

the transition probability from State S to S' in an episode is $P_{ss'} = P[S_{t+1}=S'/S_t=S]$

$P_{ss''} = P[S_{t+1}=S''/S_t=S]$

! so on



State transition matrix is just a matrix representation of these Probabilities. i.e $P_{ss'}$ or $P_{ss''}$ ....

We are in state S, what is the probability of us going to state S' from there → $P_{ss'}$

For 'n' states

$$P = \begin{matrix} \text{From} \downarrow \\ \\ \\ \end{matrix} \overset{\text{to} \rightarrow}{\begin{bmatrix} P_{11} & \cdots & P_{1n} \\ \vdots & & \\ P_{n1} & \cdots & P_{nn} \end{bmatrix}}$$

Sum of each row is 1

each element represent transition Prob.

P from this state to this state

State transition matrix gives us the complete structure of the markov problem. It tells us the complete picture, from one state (the row number), how likely I am going to the next state (the column number).

→ A markov process is therefore a random process which satisfies markov property

It can be represented as a tuple
$\langle S, P \rangle$

- S is a finite set of state (state space)
- P is a state transition probability matrix.

It is sort of a representation of the system in which our agent is operating

→ Markov Reward Process → MOVROV process with values → how good is ~~goo~~ it to be in a particular state

→ It is represented as a tupple. We add reword and discount factor γ to the previous tupple $< S, P, R, \gamma >$

R is a reward that I get from a state at that moment
We want to minimize the cumulated sum of rewards in an episode

$$R_s = E[R_{t+1} / S_t = S]$$ it is stocastic

$\gamma \in [0, 1]$ it is used to regulate the total reward. Weather we want to give more weight to future rewards or the inidiate reward.

↳ Returns → This is what we care about
It is the total discounted reward from time step t

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} \cdots = \sum \gamma^k R_{t+k+1}$$

Note: $t+1$ because we were ~~at~~ in a state at t
everything after that happens at $t+1$
G_t is the Returns generated from a random episode observed by the agent out of all the episodes possible.

→ Value Function → $V(s) = E[G_t | S_t = S]$

Value function is basically, if we are at some state S, what is the Expected value of G_t that we can get given all the possible episodes that can happen after this state and G_t for those episodes. Episodes that have higher probabilities of happening will "contribute" more to the Expected value. We have expectation because the Rewards are stocastic.

So we have two ~~vars~~ stocastic variables here which will determine the value function, 1) Reward, 2) Transition probability.

→ Bellman Equation for MRP

↳ DP

↳ $V(s) = E[G_t | S_t = S]$

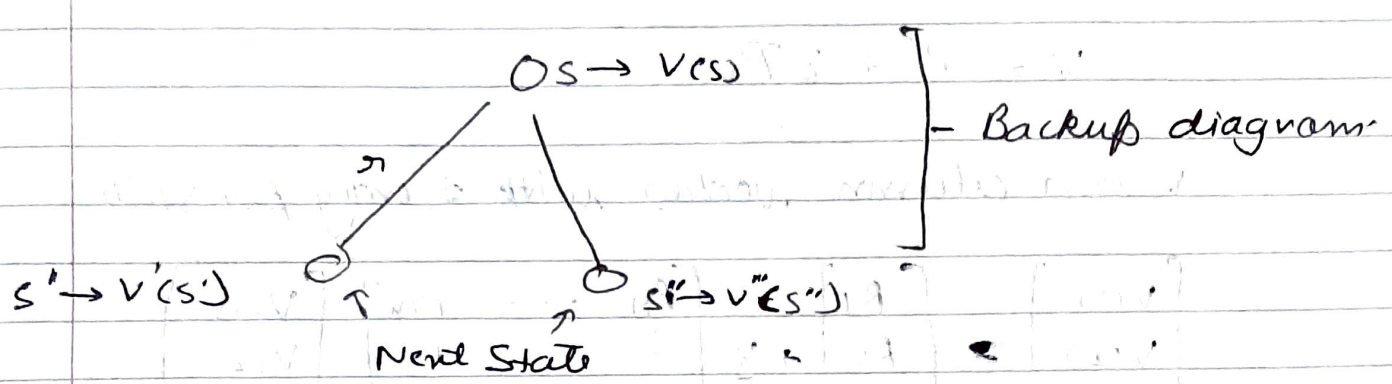
$$V(s) = E[R_{t+1} + \gamma R_{t+2} + \cdots | S_t = S]$$

$$V(s) = E[R_{t+1} + \gamma [R_{t+2} + \gamma R_{t+3} \cdots] | S_t = S]$$

$$V(s) = E[R_{t+1} + \gamma G_{t+1} | S_t = S]$$

$$V(s) = E[R_{t+1} + \gamma V(S_{t+1}) | S_t = S]$$

$$V(s) = E[R_{t+1} + \gamma V(S_{t+1}) \mid S_t = S]$$



We can calculate the valuefunction of state S given we know the value function of all the next states, their transition property and the reward for state S.

Consider we are at state S and the next possible states are S' and S''. Calculate the value function of state S

↳ through bellman equation →
Since we have multiple possible states for the next step. to know the ~~expected~~ expected value of 'at' for state S, we will sort of average the $V(S_{t+1})$ using the transition probabilities of all the possible states for the next step

$$V(s) = \underbrace{R_s}_{\text{reward for state S}} + \gamma \sum_{s' \in S} P_{ss'} V(s')$$

↓ state space.

→ Bellman Equation in Matrix form

$$V = R + \gamma P V$$

V is a column vector with 1 entry per state

$$\begin{bmatrix} V(1) \\ V(2) \\ \vdots \\ V(n) \end{bmatrix} = \begin{bmatrix} R_1 \\ R_2 \\ \vdots \\ R_n \end{bmatrix} + \gamma \begin{bmatrix} P_{11} & \cdots & P_{1n} \\ \vdots & & \vdots \\ P_{n1} & \cdots & P_{nn} \end{bmatrix} \begin{bmatrix} V(1) \\ V(2) \\ \vdots \\ V(n) \end{bmatrix}$$

$$\Rightarrow (1 - \gamma P) V = R$$

$$\Rightarrow V = (1 - \gamma P)^{-1} R$$

$O(n^3)$ for n states.

→ Markov Decision Process → Markov reward process with decisions given all the states are markov.
It is a tuple $\langle S, A, P, R, \gamma \rangle$

A (action set)

A is a finite set of action the agent can take.

P → $P^a_{ss'} = P[S_{t+1} = s' / S_t = s, A_t = a]$

state transition probability from s to s' given action A

R is a reward function for state s given action a $R^a_s = E[R_{t+1} / S_t = s, A_t = a]$

P is now dependent on the action. So we can have seperate transition probability matrix for each action in the action set. [A, n, n]

→ Policy → Stocastic in nature → Distribution of action given states → Each action has some probability of being taken given the state.

$\pi(a|s) = P[A_t = a / S_t = s]$

as it is an MDP, the policy only depends on the current states

⇒ we have ~~stational~~ stationary policy i.e the policy for a state is indipendent of the time step. It remains the same.

In practice, this may not be ~~true~~ true our environment may not support MDP. For eg. Partially observed environment.

now we have another stocastic variable, π which applies the R for a state and the transition probability

⇒ the average reward for a state given a policy would be an average over all actions

$$R_s^{\pi} = \sum_{a \in A} \pi(a|s) R_s^a$$

⇒ the average transition probability for all action given π from s to s' would be an average of $P_{ss'}$ over all actions

$$P_{s,s'}^{\pi} = \sum_{a \in A} \pi(a|s) P R_{s,s'}^a$$

→ Value function is therefore dependant on policy as some actions have more chances of being taken than other and therefore the rewards and the transition probabilities beyond that state

⇒ $V_\pi(s) = E_\pi [G_t / S_t = s]$

How good is it to be in a state s if we follow the policy π,

→ Action value function → Unlike the state value function which tells us how good is it to be in a particular state, action value function tells us how good is it to take a particular action. Its a criterion to evaluate and compare all the possible action from the given state.

It is the expected return starting from state s, taking action a, and then following policy π.

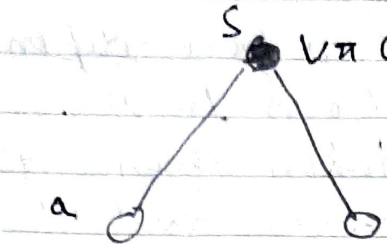
$$q_\pi(s,a) = E_\pi [G_t / S_t = s, A_t = a]$$

→ $V_\pi(s) = E_\pi [R_{t+1} + \gamma V_\pi(S_{t+1}) / S_t = s]$

~~$q_\pi(s) = E_\pi [R_{t+1} + \gamma q_\pi(S_{t+1}) /$~~

$$q_\pi(s,a) = E_\pi [R_{t+1} + \gamma q_\pi(S_{t+1}, A_{t+1}) / S_t = s, A_t = a]$$

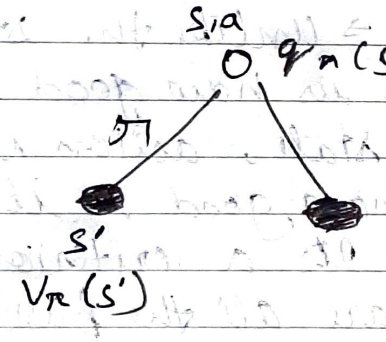
1)



$$V_\pi(s) = \sum_{a \in A} \pi(a|s)\, q_\pi(s,a)$$

↑ probability of taking that action.

2)



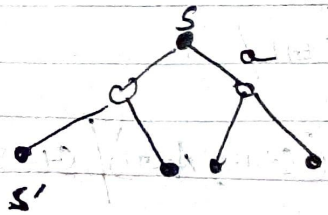
⇒ I took an action while at state s and now will be directed to s'

How good is q_π considering all the possible next states

$$q_\pi(s,a) = R_s^a + \sum_{s' \in S} P_{ss'}^a\, V_\pi(s')$$

combining ① and ②

$$V(s) = \sum_{a \in A} \pi(a|s) \left[R_s^a + \sum_{s' \in S} P_{ss'}^a\, V_\pi(s') \right]$$



→ Matrix form → $V_\pi = R^\pi + \gamma P^\pi V_\pi$

$$V_\pi = (1 - \gamma P^\pi)^{-1} R^\pi$$

Optimal Value Function –

↓

$V_*(s)$ → It is the max value function over all policies → The max possible value of the value function at a state

→ The value function that we get for the best policy i.e the maximum value possible reward

$$V_*(s) = \max_\pi V_\pi(s)$$

Optimal action value function is the maximum action value function over all policies.

$$q_*(s,a) = \max_\pi q_\pi(s,a)$$

to behave optimally at state, we pick the action which has the highest $q_*(s,a)$ and therefore for this policy the $V^*(s) = q^*(s,a) + R^a_s$

→ Optimal policy → Best possible ways to behave in an MDP.

$\pi \geq \pi'$ if $V_\pi(s) \geq V_{\pi'}(s)$, $\forall s$

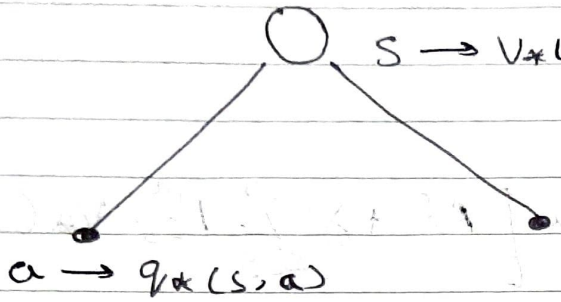
If the value function of π > value function of π' for all states then $\pi \geq \pi'$

For any MDP, there is always an MDP which is better than any other policy (or equal to)

It is possible to have more than one optimal policy. If they are 2 optimal π, they will have the same value function i.e. the optimal value function and also the optimal state value function.

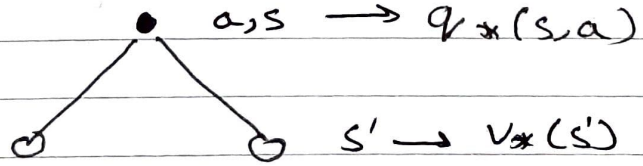
optimal policy can be found by maximi-sing over $q_*(s,a)$

→ Bellman optimality equations



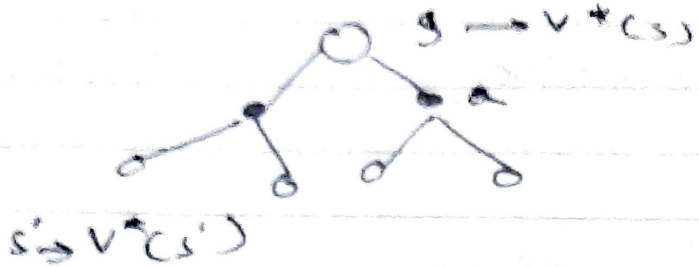
For optimality, we want to pick the action having maximum q^*. So, instead of taking the average of the q values for all the actions, the V^* would be the max q^* over all the actions

$$V^*(s) = \max_a q^*(s,a)$$

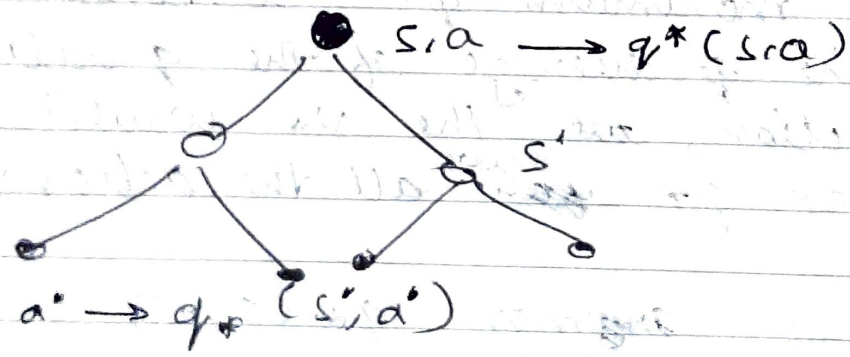


$$q^*(s,a) = R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a V^*(s')$$

we dont get to pick the next state. It happens according to the transition probability so we average the V^* values for the next states according to their transition probabilities.



$$V^*(s) = \max_a \left[R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a V_*(s') \right]$$



$$q_*(s,a) = R_s^a + \gamma \sum P_{ss'}^a \max_{a'} q_*(s',a')$$