# Lecture 9: State machines & Markov decision processes

## State machines: (Slide - 34)

- A different thing about state machines is that the new state depends on the input action (which here may be to plant or to let fallow) & on the previous state.

- Previously, the loss we dealt with depended only on inputs, but here it also depends on the state.

## ★ ~~State~~-Markov Decision Process (Slide-71)

- We use a reward function to basically reward good decisions. It may be total sale in $ or kg of harvest obtained, etc. Or it can be just a representative value.
- The reward function not just depends on the input but also on the state of the ~~machine~~ (soil).

- Now, instead of saying that some transitions are possible & some aren't; we use a stochastic model to tell the transition's likelihood.
- As you can see in the state diagram & the transition matrix, you express the transition's probability

## ★ Slide - 854

- We describe the transitional model such that it takes the previous state, action / input & the state we want to go to & spits out the probability of that happening.
- It can be expressed as: (just an example)

$$P(S_t = \text{poor} \mid S_{t-1} = \text{rich}, X_{t-1} = \text{plant})$$

where $S_t$ = state at '$t^{th}$' step or iteration

$X_t$ = action taken after $S_t$

- It can also be expressed as:

$$T(\text{rich}, \text{plant}, \text{poor})$$

where the positional arguments are $S_{t-1}, X_{t-1}, S_t$

## ★ Slide- 99

- You can simply say that the types of decisions you're gonna make is called a policy.
- It's barically an action plan

Slide- 143 What's the value of a policy?

- h: horizon is basically how many steps are remaining or maximum steps you can take.
- The value of the policy is basically the reward total accumulated after 'h' depletes (or in this case 'h' growing seasons come to an end.)

- Now, the total reward can't be calculated directly because, the model is stochastic. So for further decisions, we take the weighted average of the reward based on the probability. (i.e. the probability is the weight)

- The value function is a recursive function based on the new state of soil & the next season. (Eqn highlighted in Yellow)

## ☆ Slide - 155

- We describe a criterion to decide which policy is better for a certain 'h'
- It says that at least both values of $V^h(s)$ i.e. $V^h(\text{rich})$ & $V^h(\text{poor})$ must be greater than or equal to the other policy, AND at least 1 or strictly 1 of them must be greater for the 1st policy to win.
- We see the winners, losers & ties for h = 1, 2, 3 given in yellow
- We could describe another extra criterion for comparison when there is a tie (or no policy wins)

## ★ Slide - 163

- You can see that for very low values of 'h', you want to be as greedy as possible and just plant.
- But, as 'h' increases, it becomes wiser to plant when soil is rich & let fallow if soil becomes poor.
- This is the value of delayed gratification. i.e. you

seek a higher reward ~~for a~~ after some time for an action taken now (analogous to long term investing?)

- So, maybe we can use this idea in our policy itself. i.e. make our policy dependent on 'h' left & not just on the state of the soil. $\pi = f(h, s)$

★ Slide - 171

- To find the optimum policy, we ~~use~~ introduce a new method & function $Q^h(s,a)$ which is similar to $V_\pi^h(s,a)$.
- ~~But if you see in the recursion tree for $V_\pi^h(s,a)$, we consider both possible outcomes for any random given state. i.e. we consider reward for both~~
- We used $V_\pi^h(s,a)$ to compare two different policies. But, we use $Q^h(s,a)$ to find or get an optimum ~~policy~~ action plan for 'h' reasons.
- In $V_\pi^h(s,a)$ recursion tree, a policy was already given to us which told us which action to take for a state. But in $Q^h(s,a)$ we consider both actions for a state & we choose the action for which we get max reward.
- That is to say, we form an entire binary tree of possible actions & for each ~~th~~ path that goes from root to the leaf nodes, we check the cummulative or total reward, & we choose the path with maximum reward.
- BUT, instead of going from root to leaf, we go from leaf to root. i.e. we get the best $Q^1(s,a)$ first, then get the best $Q^2(s,a)$ based on $Q^1(s,a)$, then get the best $Q^3(s,a)$ based on $Q^2(s,a)$ and so on till we get the best $Q^h(s,a)$. ~~($Q^1(s,a)$ will be always the leaf node)~~

- Now, there can be multiple policies or basically multiple paths in the recursion tree that have same reward. Thus, there can be multiple optimum policies.
- Also, the optimum policy may be non-stationary i.e. it may also depend on 'h' left. For example, for big 'h' we may ~~choose~~ get delayed gratification, but as 'h' gets smaller, we may get greedy & choose to always plant.

## ✱ Slide - 213

- The recurrence relation for $Q^h(s,a)$ is highlighted in green. You can see that, to calculate $Q^h(s,a)$ we first calculate $Q^{h-1}(s,$ & to calculate $Q^{h-1}(s,a)$ we first calculate $Q^{h-2}(s,a)$ and so on.

- This is also called 'finite-horizon value iteration' because we have a ~~horizon~~ finite horizon. This wouldn't work for infinite 'h' or you could say in the case where there isn't any deadline or so. You just keep on farming forever.

## ✱ Slide - 235

- For infinite horizon, there exists an issue while calculating value of things. For example, if your reward is in terms of money, you might wanna account for inflation. If it is in terms of kg of harvest obtained, you might wanna take into account the changing rates per kg harvest ~~as times~~.
- As another example, if you harvest 10 tonnes of crops each year & store them with you & you keep doing this for 10 years, you will have 100 tonnes. If you sell this all at once,

you might get 'x' money. But instead, if you would have sold the 10 tonnes each year for 10 years, you might have accumulated 'y' money. ~~Most probably, xxxy bcaus xxxy because~~

- If you get 1 kg harvest this year & sell it right now, you would get 'x' money. Maybe you could invest it right now or you buy yourself the goods that you need. But instead if you store that 1 kg harvest for 10 years & then sell it, you would get 'y' money. Now, the value of 'x' after 10 years will be most probably > value of 'y' gained this year.

- So, as a solution, we introduce something called the discount factor '$\gamma$': $0 < \gamma < 1$ where $\gamma^t$ will be the value of the item after t time units.

- Now, if we take an example, where farmer 'A' wants to trade 1 bushel/item each year (1 item given to farmer B each year) with V no. of items taken right now ~~from~~ from farmer 'B'. Basically, we wanna ask whats the value of 1 item a year forever. It can be given as an infinite sum of a geometric series in $\gamma$.

- The updated version of the ~~3rd~~ $V_\pi(s,a)$ for infinite horizon is given in green. For |S| no. of states, we get those many equations with same |S| no. of unknowns. Thus, you can solve them to find best policy or action plan. ($V_\pi(s)$)

- You can also use the discount factor for finite horizon.