Finite horizon Markov decision processes (MDPs)

(Lecture - 1)

Example 1: Machine replacement

A problem over N stages

(e.g. think of maintaining a bus, with each stage corresponding to a month).

The machine can be in one of the "n" states, i.e.,

$\{1, \dots, n\}$

1 ↑ perfect condition

n ↑ worst condition

Operating cost: $g(i)$ in state i

$$g(1) \leq g(2) \leq \dots \leq g(n)$$

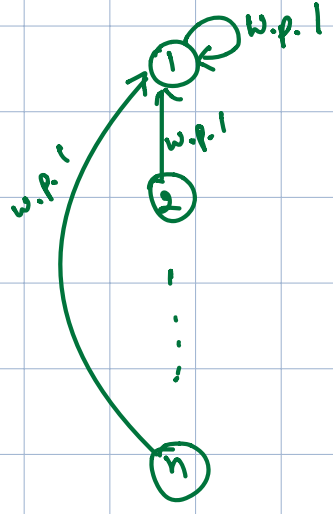
Actions: "Repair" (↓ machine becomes new) or "Do nothing" ("machine becomes progressively worse")

"Stochastic system"

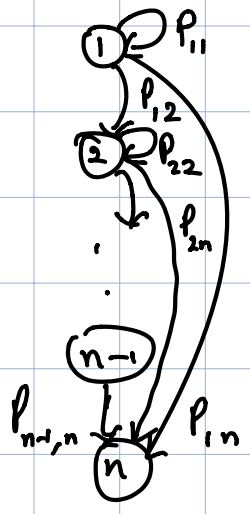
P_{ij} : probability that the machine goes from state i to state j

Transition diagram:

"Repair"



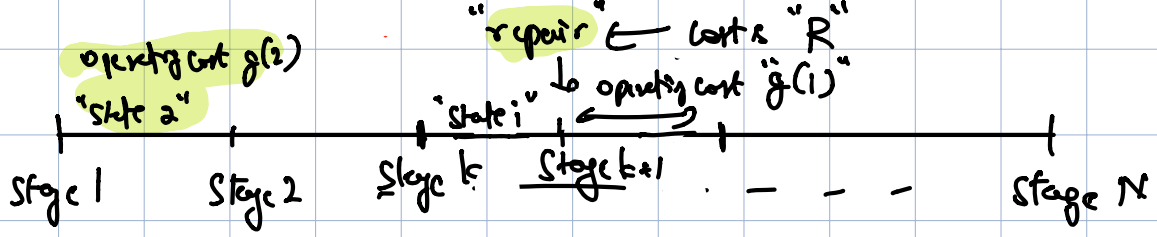
"Do nothing"



"$P_{ij} = 0$ if $j < i$"

On "repair", machine goes to state "1", & remains there for one stage. Repair cost is "R".

Goal: Choose actions so that the total cost (i.e., cumulative cost over N stages) is minimized.



Takeaway: MDP has "states", "actions", "stochastic transitions" & the goal is to minimize "total cost".

Another example:

"Chess match"

"Fix opponent"

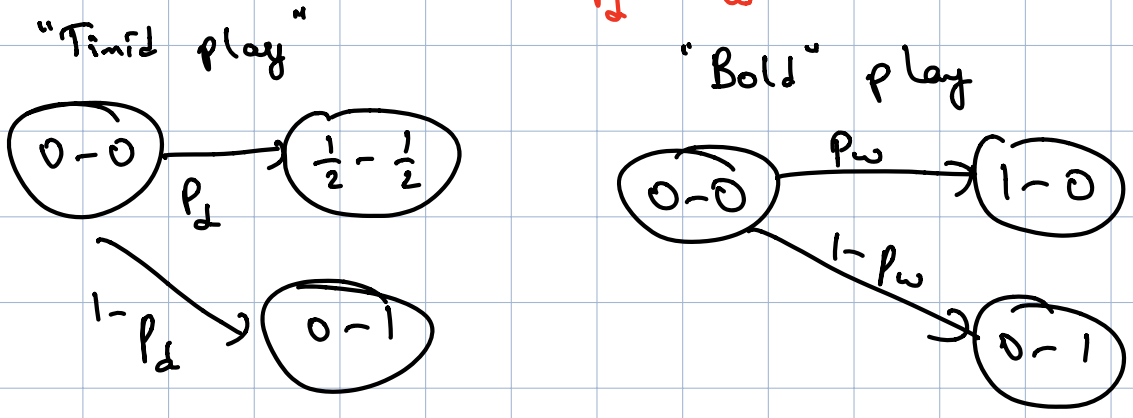
You play 2 games → 1 pt win
→ 0 loss
→ ½ draw

If there is a tie, then "sudden death" phase where you play games one after the other until a decisive result.

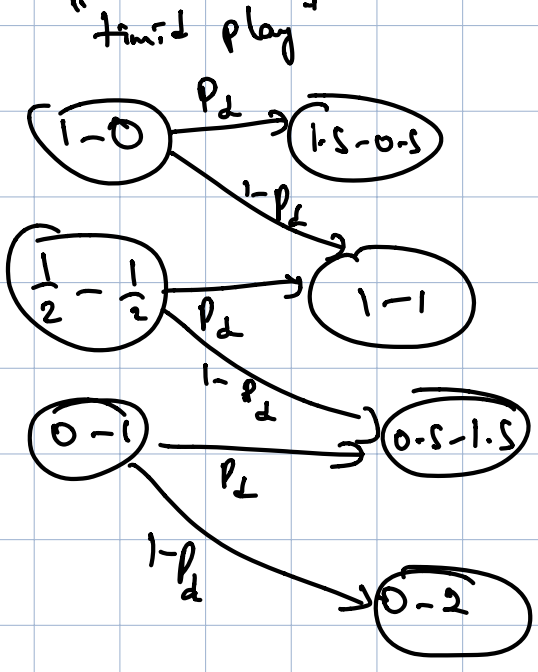
Actions: playing styles → timid: draw w.p. p_d & lose w.p. $1-p_d$
→ bold: win w.p. p_w & lose w.p. $1-p_w$

$p_d > p_w$

Game 1:



Game 2:



"Bold play"

H.W.

MDP framework:

Let $\mathcal{X}$ denote the state space,

$\mathcal{A}$ denote the action space.

$x_k \rightarrow$ state in stage k, $x_k \in \mathcal{X}$

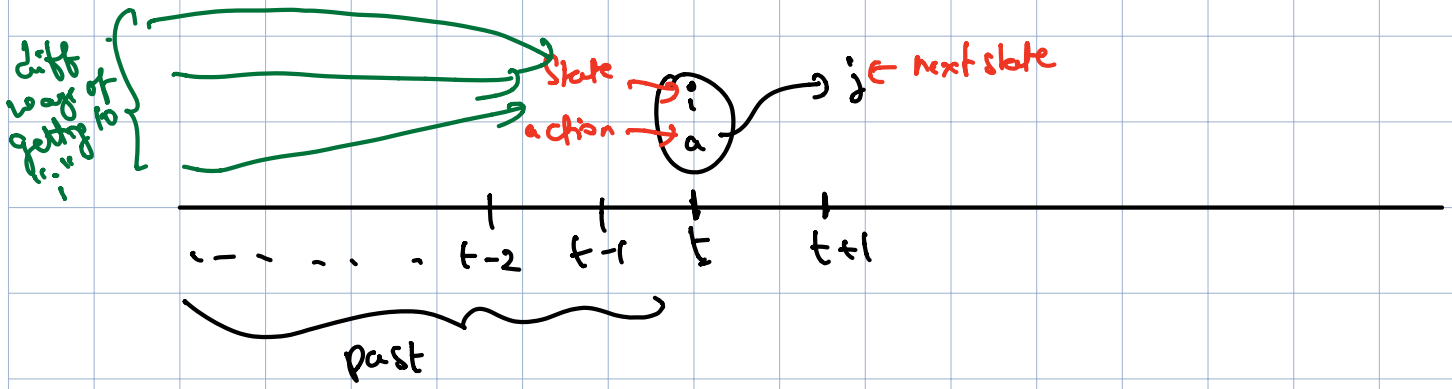
$a_k \rightarrow$ action that can be taken in state x_k, $a_k \in \mathcal{A}(x_k) \subset \mathcal{A}$.

(set of possible actions depend on the current state)

Transition probabilities:

$$p_{ij}(a) = P(x_{k+1} = j \mid x_k = i, a_k = a)$$

State evolution satisfies "Controlled Markov property".



$$P(x_{t+1} = j \mid x_t = i, a_t = a, x_{t-1}, a_{t-1}, \dots x_0)$$

$$= p_{ij}(a)$$

"Single-stage cost" $g_k(i,a,j)$ ← cost in stage k, if state is i & action taken is a, & the next state is j

$g_N(i)$ ← termination cost in the final stage in state i, $i \in \mathcal{S}$

Policy $\{\mu_0, \dots, \mu_{N-1}\}$

$\mu_k : \mathcal{S} \to \mathcal{A}$ mapping that specifies how actions are taken in stage k.

"**Admissible policy**": $\mu_k(i) \in \mathcal{A}(i)$ [set of feasible actions in state i]

$\pi = \{\mu_0, \dots, \mu_{N-1}\}$ is admissible if $\mu_k(i) \in \mathcal{A}(i) \quad \forall k = 0 \dots N-1$

Total cost: Initial state $x_0 \in \mathcal{S}$

$$J_\pi(x_0) = \underset{x_1 \dots x_N}{E}\left[g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, \mu_k(x_k), x_{k+1}) \right]$$

(J_π: $\pi = \{\mu_0, \dots \mu_{N-1}\}$; g_N: terminal cost; g_k: cost in stage k; x_k: current state; $\mu_k(x_k)$: action chosen; x_{k+1}: next state)

Side note: Fixing π, the sequence of states $x_0, x_1, \dots x_N$ is a Markov chain

"Optimization objective"

$$J_{\pi^*}(x_0) = \min_{\pi \in \Pi} J_\pi(x_0)$$

(J_{π^*}: optimal total cost; π^*: optimal policy; J_π: total cost of policy π; Π: set of "admissible" policies)

$$\pi^* = \underset{\pi \in \Pi}{\arg\min}\ J_\pi(x_0)$$

Lecture-2*

Open vs. closed loop policies:

Closed loop policy: $\pi = \{\mu_0, \mu_1, \cdots, \mu_{N-1}\}$

μ_i: decision based on the state

$\mu_i(x_i)$, $x_i \in S$.

Open loop policy: Fix the sequence of actions beforehand

Chess match (revisited):

Possible open loop policies:

	Game 1	Game 2	Prob(win)?
Policy π_1	Timid	Timid	$p_d^2 p_w$
Policy π_2	Bold	Bold	$p_w^2 + 2p_w^2(1-p_w)$
Policy π_3	Bold	Timid	$p_w p_d + p_w^2(1-p_d)$
Policy π_4	Timid	Bold	$p_w p_d + p_w^2(1-p_d)$

Lets ignore π_1 (If $3p_w > p_d$, then π_2 is better than π_1)

Which among π_2, π_3, π_4 is the best?

$$\max\left(p_w^2(3-2p_w),\ p_w p_d + p_w^2(1-p_d)\right)$$

$$= \max\left(p_w^2 + 2p_w^2(1-p_w),\ p_w^2 + p_w p_d(1-p_w)\right)$$

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

If $2p_w < p_d$, then π_3/π_4 are better

else, π_2 is better.

Set $p_w = 0.45$, $p_d = 0.9$. Then Prob(winning match) $= 0.425$

< 50% chance of win

Closed loop policy: Play timid if leading, else play bold.

π_c

Prob(match win with π_c)

$$= p_w p_d + p_w\left(p_w(1-p_d) + (1-p_w)p_w\right)$$

$$= p_w^2(2-p_w) + p_w(1-p_w)p_d$$

$$= 0.53 \quad \text{for} \quad p_w = 0.45 \;\&\; p_d = 0.9$$

> 50% chance of match win

Optimality principle

Let $\pi^* = \{\mu_0^*, \mu_1^*, \ldots, \mu_{N-1}^*\}$ denote the optimal policy.

Consider the tail sub-problem

$$\min_{\pi^i = (\mu_i, \ldots, \mu_{N-1})} E\left(g_N(x_N) + \sum_{k=i}^{N-1} g_k(x_k, \mu_k(x_k), x_{k+1})\right) \quad \text{—(*)}$$

The policy $\{\mu_i^*, \ldots, \mu_{N-1}^*\}$ is optimal for the $(N-i)$ stage problem in (*)

Proof: see Sec 1.3 of Bertsekas DPOC. Vol. I

DP algorithm:

Set $J_N(x_N) = g_N(x_N), \quad \forall x_N \in S$

For $k = N-1, \dots, 0$, do

$$J_k(x_k) = \min_{a_k \in A(x_k)} E_{x_{k+1}} \left(g_k(x_k, a_k, x_{k+1}) + J_{k+1}(x_{k+1}) \right), \quad \forall x_k \in S$$

Applying DP algorithm to "machine replacement" example:

$$J_N(i) = 0 \qquad \text{(no terminal cost)}$$

$$J_k(i) = \min \left(\underbrace{R + g(1) + J_{k+1}(1)}_{\text{repair}}, \underbrace{g(i) + \sum_{j=i}^{n} p_{ij} J_{k+1}(j)}_{\text{do nothing}} \right)$$

"DP algorithm finds the best policy"

Claim: $\forall x_0 \in S$, the function $J_0(x_0)$ obtained at the end of the DP algorithm coincides with the optimal cost $J^*(x_0) (= J_{\pi^*}(x_0))$.

Proof: For any admissible policy $\pi = \{\mu_0, \dots, \mu_{N-1}\}$, let $\pi^k = \{\mu_k, \dots, \mu_{N-1}\}$

$J_k^*(x_k)$ be the optimal cost of the tail subproblem beginning in stage k, in state x_k.

$$J_k^*(x_k) = \min_{\pi^k = \{\mu_k, \dots, \mu_{N-1}\}} E\left(g_N(x_N) + \sum_{i=k}^{N-1} g_i(x_i, \mu_i(x_i), x_{i+1}) \right)$$

Claim: $J_k^*(x_k) = J_k(x_k)$

↳ obtained by DP algorithm

<pf with inpf?> $J_N^*(x_N) = g_N(x_N) = J_N(x_N)$.

Induction hypothesis: Assume for $k+1$

$$J_{k+1}^*(x_{k+1}) = J_{k+1}(x_{k+1}) \quad \forall x_{k+1}.$$

$$J_k^*(x_k) = \min_{(\mu_k, \pi^{k+1})} E\left(g_N(x_N) + g_k(x_k, \mu_k(x_k), x_{k+1}) + \sum_{i=k+1}^{N-1} g_i(x_i, \mu_i(x_i), x_{i+1}) \right)$$

$$= \min_{\mu_k} E_{x_{k+1}}\left[g_k(x_k, \mu_k(x_k), x_{k+1}) + \min_{\pi^{k+1}} E_{x_{k+2}, \dots, x_N}\left[g_N(x_N) + \sum_{i=k+1}^{N-1} g_i(x_i, \mu_i(x_i), x_{i+1}) \,\middle|\, x_{k+1} \right] \right]$$

$= J_{k+1}^*(x_{k+1}) = J_{k+1}(x_{k+1})$ by induction hypothesis.

$$= \min_{\mu_k} E\left(g_k(x_k, \mu_k(x_k), x_{k+1}) + J_{k+1}(x_{k+1}) \right)$$

(**) $$= \min_{a_k \in A(x_k)} E\left(g_k(x_k, \mu_k(x_k), x_{k+1}) + J_{k+1}(x_{k+1}) \right) = J_k(x_k)$$

So, $J_k^*(x_k) = J_k(x_k)$ ■ end of pf (or the pf?)

To infer (**), we used

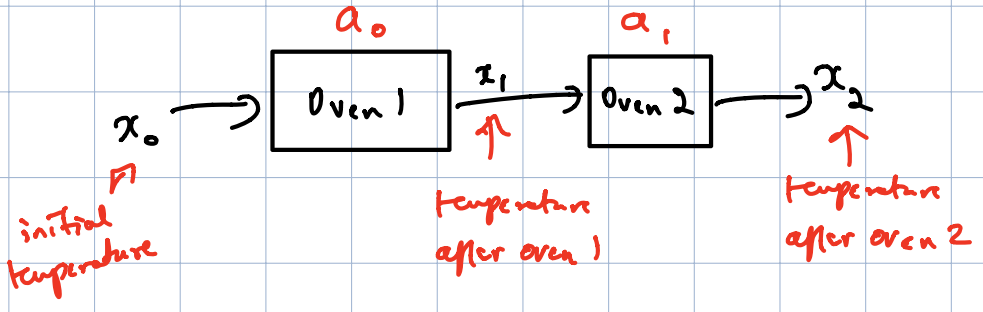
$$\min_{\mu \in B} F(x, \mu(x)) = \min_{a \in A(x)} F(x, a),$$

where $B = \{\mu \mid \mu(x) \in A(x)\}$ ■

Lecture - 3

Examples:

(1) **Linear system with quadratic cost**



Goal: to get x_2 as close as possible to a target temperature T

State Evolution:

$$x_{k+1} = (1-\alpha) x_k + \alpha a_k, \quad k = 0, 1 \qquad \text{——(**)}$$

linear state evolution

$\alpha \in (0,1)$ fixed

Total cost (to be minimized) : $a_0^2 + a_1^2 + (x_2 - T)^2$

Apply DP algorithm:

Final stage : $J_2(x_2) = (x_2 - T)^2$ — (**)

Going back : $J_1(x_1) = \min_{a_1} \left(a_1^2 + J_2(x_2) \right)$

from (*) $= \min_{a_1} \left(a_1^2 + J_2\left((1-\alpha)x_1 + \alpha a_1\right) \right)$

using (**) $= \min_{a_1} \left(a_1^2 + \left((1-\alpha)x_1 + \alpha a_1 - T\right)^2 \right)$

optimal action $\mu_1^*(x_1) = \dfrac{\alpha\left(T - (1-\alpha)x_1\right)}{1+\alpha^2}$ ← linear

$J_1(x_1) = \dfrac{\left((1-\alpha)x_1 - T\right)^2}{1+\alpha^2}$ ← quadratic

Check:

$\mu_0^*(x_0) = \dfrac{(1-\alpha)\alpha\left(T - (1-\alpha)^2 x_0\right)}{1+\alpha^2\left(1+(1-\alpha)^2\right)}$ ← linear

$J_0^*(x_0) = \dfrac{\left((1-\alpha)^2 x_0 - T\right)^2}{1+\alpha^2\left(1+(1-\alpha)^2\right)}$ ← quadratic

Adding randomness to ovens:

$$x_{k+1} = (1-\alpha)x_k + \alpha a_k + w_k, \quad k = 0, 1$$

w_k: zero-mean r.v. iid (indep of $\{x_k\}$) with bounded variance (e.g. $N(0, \sigma^2)$)

Applying DP algorithm:

$$J_1(x_1) = \min_{a_1} E\left(a_1^2 + \left((1-\alpha)x_1 + \alpha a_1 + w_1 - T\right)^2 \right)$$

$$= \min_{a_1} \Big(a_1^2 + \left((1-\alpha)x_1 + \alpha a_1 - T\right)^2 + 2 E w_1 \left((1-\alpha)x_1 + \alpha a_1 - T\right) + E\, w_1^2 \Big)$$

$E w_1 = 0$ $\rightarrow$

$$= \min_{a_1} \Big(a_1^2 + \left((1-\alpha)x_1 + \alpha a_1 - T\right)^2 + E\, w_1^2 \Big)$$

Minimizing RHS above leads to the same $\mu_1^*(x_1)$ as before.

Lecture-4*

Chess match - revisited (last time)

Consider an extension to N games

timid $\rightarrow p_d$, bold $\rightarrow p_w$

Players play N games & enter sudden death if the score is tied.

State: net score (e.g. (0-1) state = -1)

Apply DP algorithm:

$$J_k(x_k) = \max\Big(\underbrace{p_d\, J_{k+1}(x_k) + (1-p_d) J_{k+1}(x_k - 1)}_{\text{timid play}},\ \underbrace{p_w\, J_{k+1}(x_k + 1) + (1-p_w) J_{k+1}(x_k - 1)}_{\text{bold play}} \Big)$$

(*)

we are playing for rewards (choose min to not in DP algo)

$$J_N(x_N) = \begin{cases} 1 & \text{if } x_N > 0 \\ p_w & \text{if } x_N = 0 \\ 0 & \text{if } x_N < 0 \end{cases}$$

It is better to play bold when

$$\frac{p_w}{p_d} > \frac{J_{k+1}(x_k) - J_{k+1}(x_k - 1)}{J_{k+1}(x_k + 1) - J_{k+1}(x_k - 1)}$$

→ inferred from (*)

Given that we have J_N specified, we can go back & calculate J_{N-1}.

x_{N-1}	J_{N-1}	Best play
>1	1	does not matter
1	$\max(p_d + (1-p_d)p_w,\ p_w + (1-p_w)p_w)$ $= p_d + (1-p_d)p_w$	Timid
0	p_w	Bold
-1	p_w^2	Bold
<-1	0	does not matter

For the 2-game match $\Rightarrow$ we can figure the optimal strategy by knowing $J_{N-2}(0)$

$$J_{N-2}(0) = \max\Big(\underbrace{p_d p_w + (1-p_d)p_w^2}_{\text{timid}},\ \underbrace{p_w\big(p_d + (1-p_d)p_w + (1-p_w)p_w^2\big)}_{\text{bold play}}\Big)$$

$$= \max\Big\{ p_w\big(p_d + (1-p_d)p_w\big),\ p_w\big(p_d + (1-p_d)p_w + (1-p_w)p_w\big)\Big\}$$

$$= p_w\big(p_d + (1-p_d)p_w + (1-p_w)p_w\big) \Rightarrow \text{play bold}$$

As noted before, one could choose $p_w < 0.5$ & still get a better than 50-50 chance of winning the match if $p_w\big(p_d + (1-p_d)p_w + (1-p_w)p_w\big) > 0.5$

Another example: (Job scheduling)

N jobs to schedule

T_i → time taken for ith job to complete

T_i is a r.v. $\{T_i, i = 1 \ldots N\}$ independent

Each job "i" has a reward R_i associated with it.

So, if job "i" finishes at time "t", then

the reward is $\alpha^t R_i$, α = discount factor $0 < \alpha < 1$

Cumulative reward = Sum of each job's reward.

Goal: schedule jobs to maximize cumulative reward.

"Interchange argument" to figure optimal schedule

$$L = \{ i_0, i_1, \ldots, i_{k-1}, i, j, i_{k+2}, \ldots, i_{N-1} \}$$

$$L' = \{ i_0, i_1, \ldots, i_{k-1}, j, i, i_{k+2}, \ldots, i_{N-1} \}$$

$$J_L = E\left[\alpha^{t_0} R_{i_0} + \ldots + \alpha^{t_{k-1}} R_{i_{k-1}} + \alpha^{t_{k-1}+T_i} R_i + \alpha^{t_{k-1}+T_i+T_j} R_j + \ldots + \alpha^{t_{N-1}} R_{i_{N-1}} \right]$$

$$J_{L'} = E\left[\alpha^{t_0} R_{i_0} + \ldots + \alpha^{t_{k-1}} R_{i_{k-1}} + \alpha^{t_{k-1}+T_j} R_j + \alpha^{t_{k-1}+T_i+T_j} R_i + \ldots + \alpha^{t_{N-1}} R_{i_{N-1}} \right]$$

Schedule L is better than L' if

$$E\left[\alpha^{t_{k-1}+T_i} R_i + \alpha^{t_{k-1}+T_i+T_j} R_j \right] \geq E\left[\alpha^{t_{k-1}+T_j} R_j + \alpha^{t_{k-1}+T_i+T_j} R_i \right]$$

Using t_{k-1}, T_i, T_j are independent,

$$\frac{E(\alpha^{T_i})\,R_i}{1-E(\alpha^{T_i})} \geq \frac{E(\alpha^{T_j})\,R_j}{1-E(\alpha^{T_j})} \quad \text{---} \ (*)$$

From (*), the optimal schedule works out as follows:

Assign $\mu_i = \dfrac{E(\alpha^{T_i})R_i}{1-E(\alpha^{T_i})}$ as the index for job i, $i = 1 \cdots N$

Order $\{\mu_1 \cdots \mu_N\}$, say $\mu_{[1]} \geq \mu_{[2]} \cdots \geq \mu_{[N]}$

Optimal schedule $= \{[1], [2], \cdots [N]\}$

index-based optimal policy

Further reading: Check out Gittins index Sec 1.3 of PPOC Vol.II

Yet-another example: <Optimal stopping>

"Asset-selling"

A technical note before asset-selling:

Discrete-time MDPs can be formulated as

$$x_{k+1} = f(x_k, a_k, w_k)$$

w_k → disturbance

$\{w_k\}$ could be i.i.d or could depend on x_k, a_k

$x_k \in$ infinite set.

For the case when $x_k \in \{1, \dots n\}$, it is enough to know $p_{ij}^a = P(x_{k+1}=j \mid x_k=i, a_k=a)$

Now to asset-selling!

Want to sell an asset.

You get offers $w_0, w_1, \dots, w_{N-1}$

Assume: $\{w_k\}$ i.i.d, finite mean

Action → sell the asset (by accepting the offer) a^1

↘ don't sell & wait for more offers a^2

Add a special state "T" to denote that the asset is sold.

$A(s_0, x_0=0$

$x_{k+1} = f(x_k, a_k, w_k)$, where

$$f(x_k, a_k, w_k) = \begin{cases} T & \text{if } (x_k \neq T, a_k = a^1) \text{ or } (x_k = T) \\ w_k & \text{else.} \end{cases}$$

Work with rewards.

Goal: maximize $\underset{\text{expectation over } w_0, w_1, \dots w_{N-1}}{E}\left[g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, a_k, w_k) \right]$

$$g_N(x_N) = \begin{cases} x_N & \text{if } x_N \neq T \\ 0 & \text{else} \end{cases}$$

$$g_k(x_k, a_k, w_k) = \begin{cases} (1+r)^{N-k} x_k & x_k \neq T,\ a_k = a' \\ 0 & \text{else} \end{cases}$$

asset not sold — sell

$r > 0$ is the interest rate.

Apply DP algorithm:

$$J_N(x_N) = \begin{cases} x_N & \text{if } x_N \neq T \\ 0 & \text{else} \end{cases}$$

$$J_k(x_k) = \begin{cases} \max\left(\underbrace{(1+r)^{N-k} x_k}_{\text{sell}},\ \underbrace{E(J_{k+1}(w_k))}_{\text{don't sell}} \right) & \text{if } x_k \neq T \\ 0 & \text{if } x_k = T \end{cases}$$

Let $\alpha_k = \dfrac{E(J_{k+1}(w_k))}{(1+r)^{N-k}}$

Optimal policy: <threshold-based policy> "non-stationary thresholds α_k"

sell if $x_k > \alpha_k$

don't sell if $x_k < \alpha_k$

if $x_k = \alpha_k$ both actions are fine

Understanding the optimal policy:

Suppose $\alpha_k \geq \alpha_{k+1} \quad \forall k$

For notational convenience, let $V_k(x_k) = \dfrac{J_k(x_k)}{(1+r)^{N-k}}$,

for $x_k \neq T$

DP algo in "V" notation is

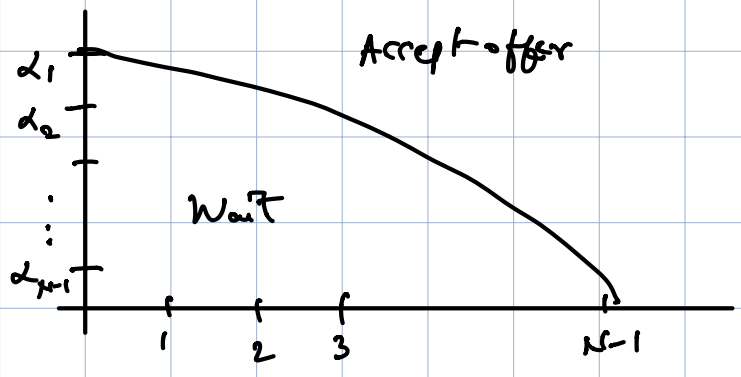
$$V_N(x_N) = x_N$$

$$V_k(x_k) = \max\left(x_k, \frac{E(V_{k+1}(\omega_k))}{(1+r)} \right)$$

optimal policy $\mu_k^*(x_k) = \begin{cases} a^1 & \text{if } x_k \geq \alpha_k \\ a^2 & \text{else} \end{cases}$

Lecture 5*

Claim: $\alpha_k \geq \alpha_{k+1}$



$$\alpha_k = \frac{E(V_{k+1}(\omega))}{1+r}$$

To show $\alpha_k \geq \alpha_{k+1}$, it is enough if we establish that $V_k(x) \geq V_{k+1}(x) \quad \forall x$

For $k = N-1$,

$$V_{N-1}(x) = \max\left(x, \frac{E(V_N(\omega))}{1+r}\right)$$

$$= \max\left(x, \frac{E(\omega)}{1+r}\right) \geq x = V_N(x)$$

For $k = N-2$,

$$V_{N-2}(x) = \max\left(x, \frac{E(V_{N-1}(\omega))}{1+r}\right)$$

$$\geq \max\left(x, \frac{E(V_N(\omega))}{1+r}\right)$$

$$= V_{N-1}(x)$$

Proceeding similarly, we get $V_k(x) \geq V_{k+1}(x), \forall x, \forall k$

Understanding the asset selling problem for large N:

Suppose "ω" is a continuous, positive-valued r.v. with distribution F_ω & density h.

$$V_{k+1}(\omega) = \begin{cases} \omega & \text{if } \alpha_{k+1} \leq \omega \\ \alpha_{k+1} & \text{else} \end{cases}$$

$$\alpha_k = \frac{E\left(V_{k+1}(\omega)\right)}{1+r}$$

$$= \frac{1}{1+r}\int_0^{\infty} V_{k+1}(\omega)\, h(\omega)\, d\omega$$

$$= \frac{1}{1+r}\int_0^{\alpha_{k+1}} \alpha_{k+1}\, h(\omega)\, d\omega + \frac{1}{1+r}\int_{\alpha_{k+1}}^{\infty} \omega\, h(\omega)\, d\omega$$

$$(*) \qquad \alpha_k = \frac{\alpha_{k+1}}{1+r} F_\omega(\alpha_{k+1}) + \frac{1}{1+r}\int_{\alpha_{k+1}}^{\infty} \omega\, h(\omega)\, d\omega$$

$$\leq \frac{\alpha_{k+1} F_\omega(\alpha_{k+1})}{1+r} + \frac{1}{1+r}\int_0^{\infty} \omega\, h(\omega)\, d\omega$$

$$\alpha_k \leq \frac{\alpha_k}{1+r} + \frac{E(\omega)}{1+r}$$

From the above, we can conclude

$$0 \leq \alpha_k \leq \frac{E\omega}{r}$$

Since $\alpha_{k+1} \leq \alpha_k$, the sequence $\{\alpha_k\}$ converges as $k \to -\infty$, say to some

$$\bar{\alpha} \in \left[0, \frac{E\omega}{r}\right]$$

Taking $k \to -\infty$ in (*), we obtain

$$\bar{\alpha} = \frac{\bar{\alpha}\, F_w(\bar{\alpha})}{1+r} + \frac{1}{1+r} \int_{\bar{\alpha}}^{\infty} w\, h(w)\, dw$$

$\bar{\alpha}$ can be calculated using distribution of w when N is large

So, an approximation to the optimal policy is to use $\bar{\alpha}$ in place of α_k at stage k, i.e.,

sell at x_k if $x_k > \bar{\alpha}$

don't sell if $x_k \le \bar{\alpha}$

} a threshold-based policy with "stationary" thresholds

<End of finite-horizon MDPs>