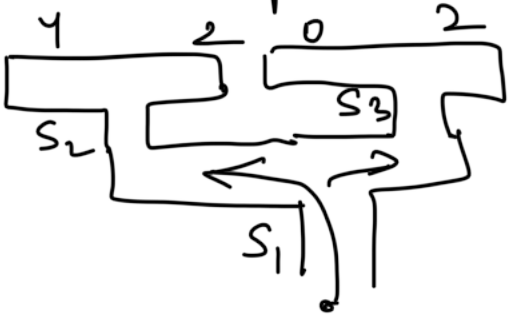
RL Week-2 Notes

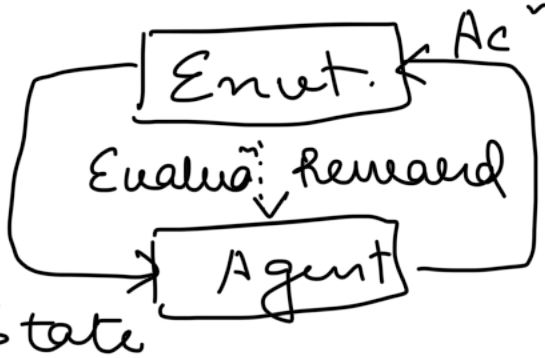
t1 Full RL Problem

- Immediate RL, Bandit Problem, contextual bandit problem.
- What abt. Tic-Tac-Toe? Seq. of actions, you get results at the end.
- your 2nd problem you see, is due to the ac^n that you took for 1st problem.
- Ac^n at a (Temporal Distance)



→ app. ac^n at S1
→ depends on ac^n S2 & S3
→ ∴, no immediate feedback.

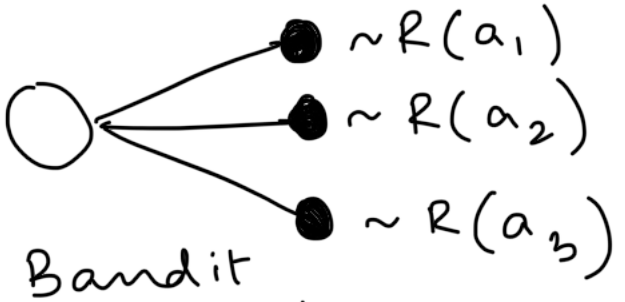
- RL Framework



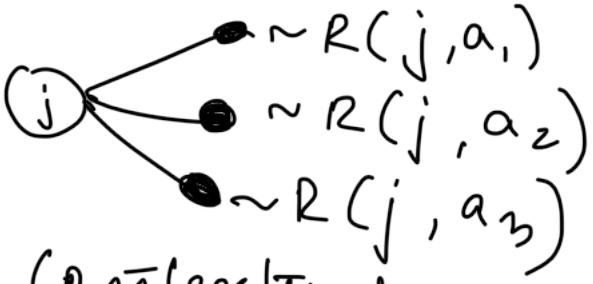
→ learning in close envt.
→ stochastic envt.
→ noisy delayed scalar evalu^n

→ goal is to maximize the long term results.
→ Agent learns the mapping from state to ac^n, to get the best evalu^n.
→ inherently though the model is deterministic but currently model does not know, ∴, it is stochastic in nature.

- ○ state ● ac^n

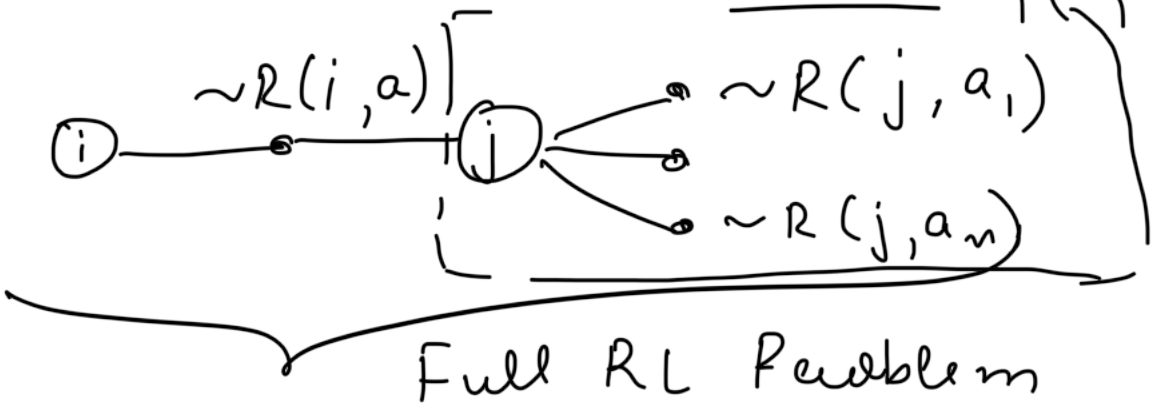


Bandit Problem



Contextual

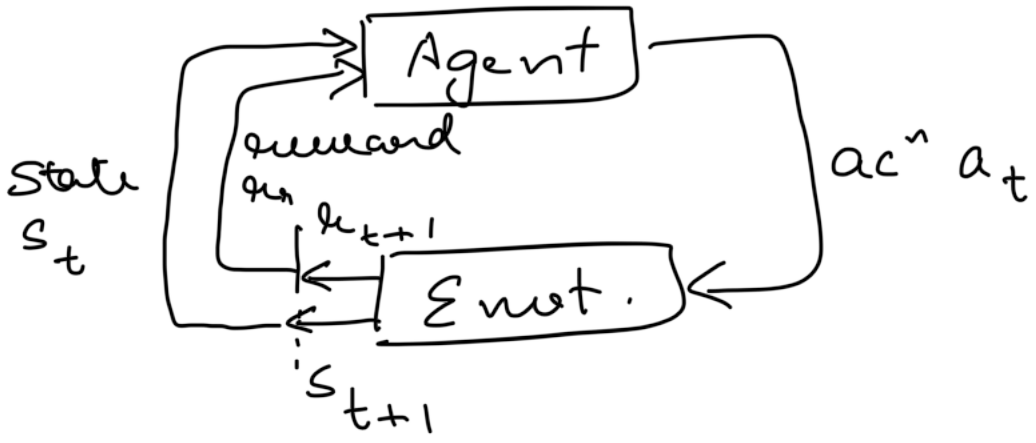
$Q(a_i^*) = \max_i \{Q(a_i)\}$ $\quad Q(j, a_i^*) = \max_i \{Q(j, a_i)\}$



Full RL Problem

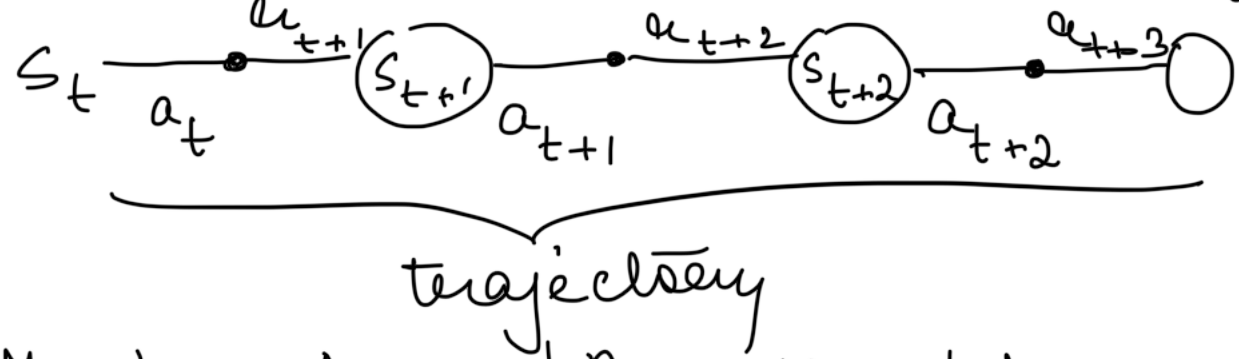
- basically we are trying to use predic^n as surrogate feedback.
- for every state, you have multiple ac^ns
  ↳ this basically leads to DP.

L2 Markov decision Process



- agent & envt. are working in discrete time steps $t \in 0, 1, 2$ (not term clock)

- state at step t : $s_t \in S$ (discrete steps)
- produces ac^n at step t : $a_t \in A(s_t)$
  you get a reward : $r_{t+1} \in R$
  then the state becomes $\rightarrow s_{t+1}$



trajectory

- Markov Assump^n $\rightarrow$ the state $\rightarrow$ all the relevant info. available to the agent about the envt. State can be abstract entity or a seq. of sensa^{ns}. But it should have "essential" info.

$$Pr\{s_{t+1} = s', r_{t+1} = r \mid s_t, a_t\}$$

$\rightarrow$ it should not req. the history.

$p(s', r \mid s, a)$

↳ it holds the Markov Property.

- MDP (Markov decision Process)

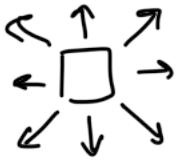
$M = \langle S, A, p, r \rangle \Rightarrow$ Tuple

- S : set of states
- A : set of ac^{ns}
- p : probabs. of transi^n $S \times A \times S \rightarrow [0,1]$
- r : exp. reward $S \times A \times S \rightarrow R$

- Policy : $\pi : S \times A \rightarrow [0,1]$ (can be state to ac^n map deterministic)

- we want to max. total expected reward, such, policy is c/d an optimal policy.

L3 MDP: Problem to Formulation

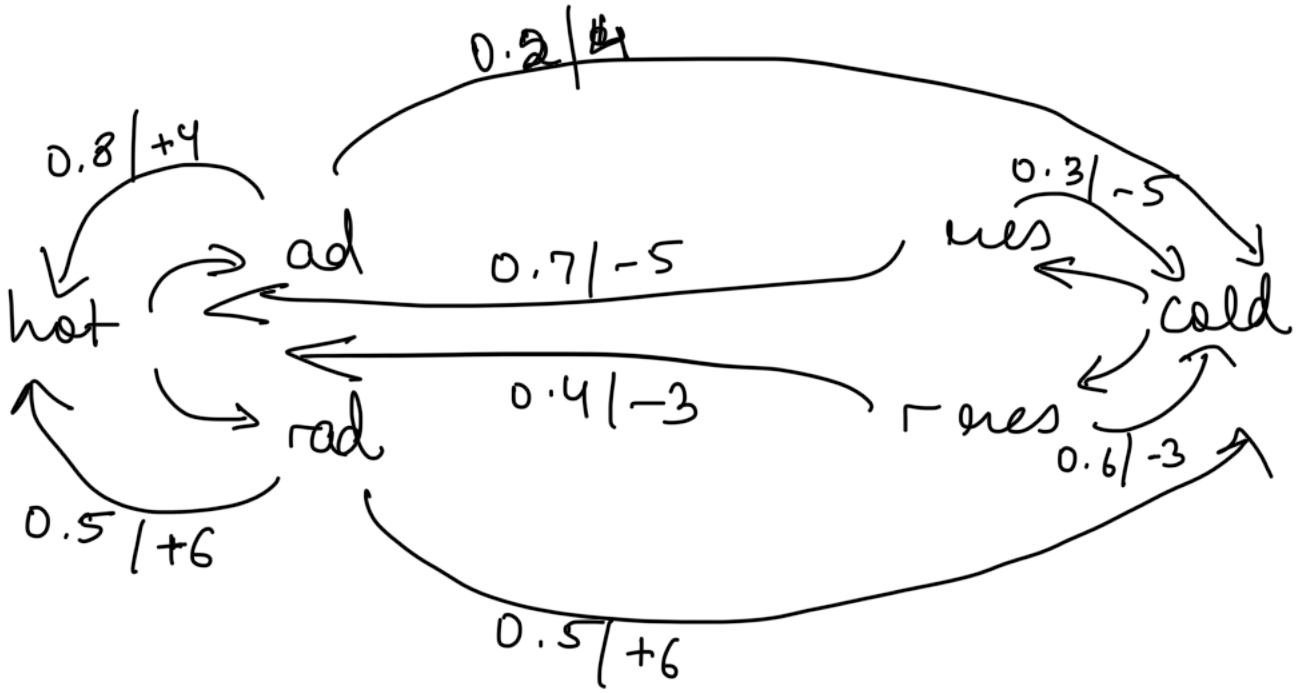
- states → enough info to take decision, raw input (not enough)
- actions → control vars., modes in the game → can be discrete / continuous
  ↳ they are not always so easy, learn how to run a car.
- rewards → defines the problem game.
- eg. of student defining the reward func^n to learn how to cycle. It ended up doing rounds only. But remember there are numerous succes stories as well.
- Eg. 2D Workspace  } 8 direc^{ns} to go.
- Eg. Computer Manufacturer Company (you should be able to convert words into MDP)

Actions → $A = \{ ad, \neg ad, res., \neg res. \}$

States → $S = \{ hot, cold \}$

$A(hot) = \{ ad, \neg ad \}$

$A(cold) = \{ res, \neg res \}$



- Eg. Robot control → input, actⁿ, +ve/-ve rewards.

L4 Returns

- Agent is learning a policy.

  Policy at t, π:

$$\pi_t(s, a) = P(a_t = a, S_t = s)$$

  mapping states to actⁿ probabilities

- RL methods define how the policy is for the agent based on experience. Max. the reward goal remains same.
- Episodic tasks → interactⁿ breaks into naturally broken tasks.

$$G_t = R_{t+1} + R_{t+2} + \ldots + R_T$$

  T is the final time step, when you reach the terminal state.

- Continuing Task - w/o any natural eps.
  ↳ Discounted return

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

$$= \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

discount rate

Short sighted $\leq \gamma \leq 1$ far sighted

- we are maximizing the expected return, $E\{G_t\}$, for each step t. If you are certain, the summ^n won't go to ∞, then you can use $\gamma = 1$

LS Value Func^n

- expected future rewards starts from state s following policy π.

State - value func^n for π.

$$v_\pi(s) = E_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \,\middle|\, S_t = s\right]$$

Ac^n value func^n

$$q_\pi(s,a) = E_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \,\middle|\, S_t = s,\ A_t = a\right]$$

$$v_\pi(s) = \sum_a \pi(a|s) q_\pi(s,a)$$

$$= \mathop{\mathbb{E}}_\pi \left[q_\pi(s,a) \right]$$

L6 MDP: Recycling Robot

- robot → collect empty cans, sensors to detect cans, arms to collect it
- room → empty cans, humans, recharge stat^n (for robot)
- Interface
  1. Naviga^n → go to cans, $\Delta\tau$ to motors, +ve if bot goes to can, −ve if bot topples.
  2. Pick & Place → pick the can, put to bin, $\Delta\tau$ to motors, +ve/−ve respectively.
  3. Search → search for cans, search, wait or recharge, +ve if bot collects the can, −ve for bot gets 0.
- charge lvl. → {high, low}

high → search, wait

low → search, wait, recharge

outcome

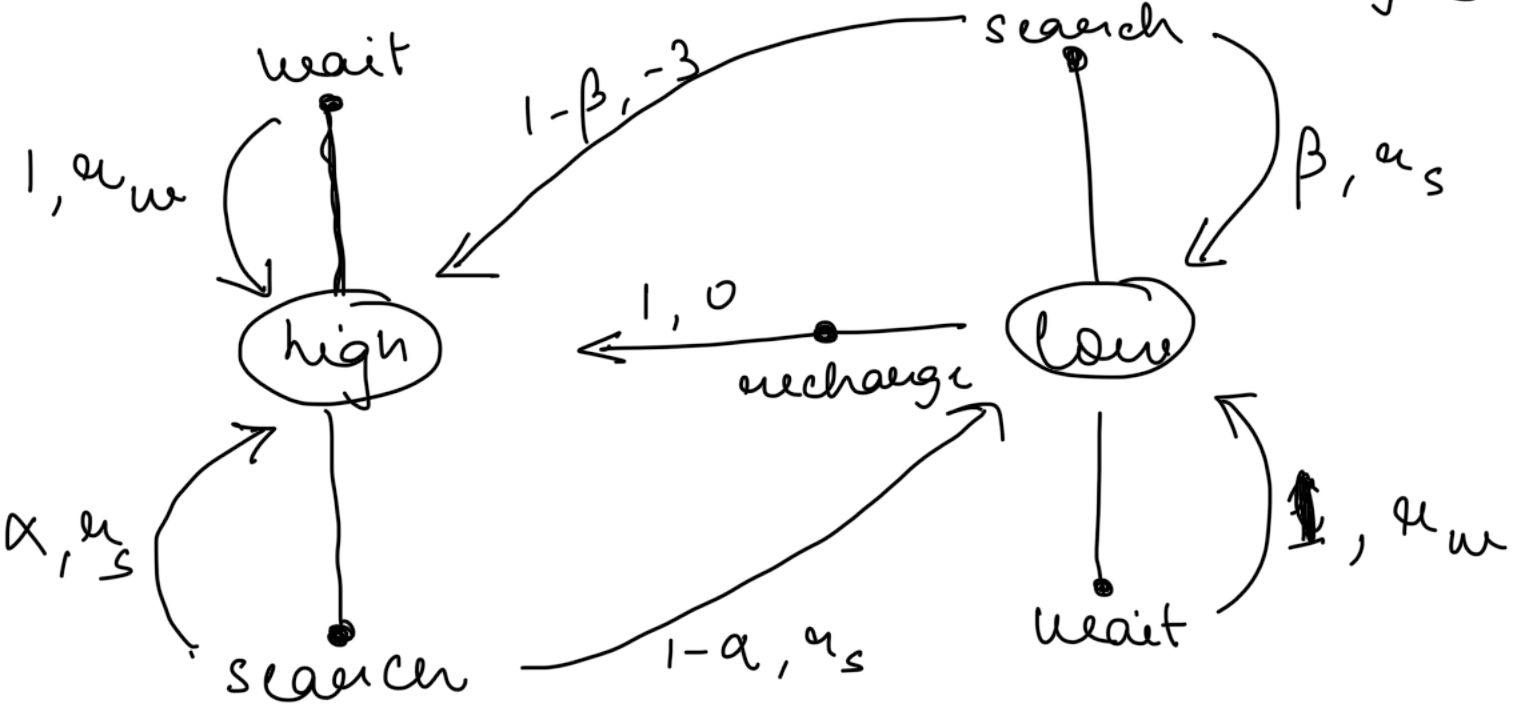
+ → collects a can

− → battery runs down.

$S = \{ high, low \}$

$A(high) = \{ search, wait \}$

$A(low) = \{ search, wait, recharge \}$



Transition Graph

○ state
● Acⁿ

$r_w \rightarrow r_{wait}$

$r_s \rightarrow r_{search}$

s	a	s'	$p(s' \mid s, a)$	$r(s, a, s')$
high	search	high	α	r_s
⋮				
—	—	—	—	—

Finite MDP ⟶

Deterministic Policy ⟶ Stochastic

s	a	$\pi(a \mid s)$	
high	search	1	0.8
⋮	⋮	⋮	⋮
low	recharge	0	0.3

trajectory ↓

$s_0 \; a_0 \; s_1 \; r_1 \; a_1 \; s_2 \; r_2 \; a_2 \; s_3 \ldots$

high search high 2 search low 1 recharge high ...

$$G_0 = 2 + \gamma 1 + \gamma^2 0 + \cdots$$

$$v_\pi(s) = \mathbb{E}_\pi \left[G_t \mid S_t = s \right]$$

$$v_\pi(\cdot) = \begin{bmatrix} v_\pi(\text{low}) \\ v_\pi(\text{high}) \end{bmatrix}$$

L7 Haunted House Example

- Read the para given in lecture.

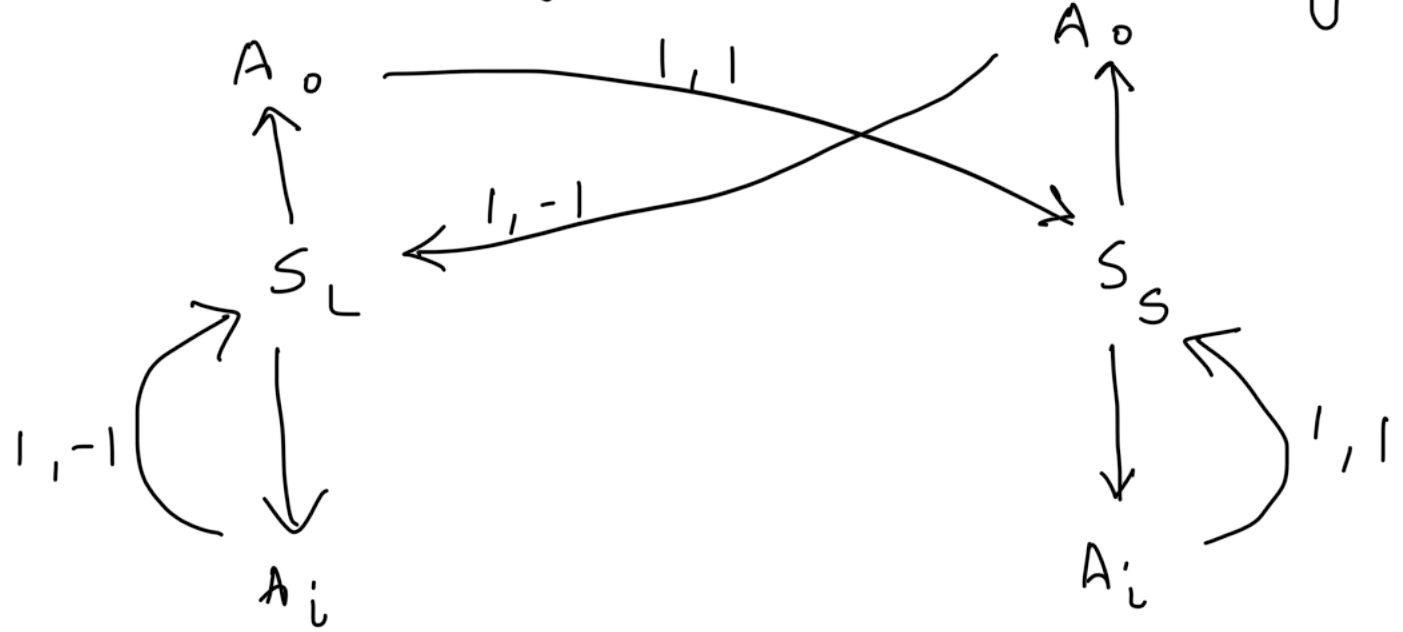
states — {laughter S_L, silence S_S}

acns — {organ A_0, incense A_i}

$\gamma = 0.9$

$R \rightarrow \{+1, -1\}$

$S_0 = S_L$, goal = silent always



s	a	s'	$p(s' \mid s, a)$	$r(s, a, s')$
S_L	A_0	S_L	0	—
S_L	A_0	S_S	1	1
S_L	A_i	S_L	1	−1
S_L	A_i	S_S	0	—
⋮				

	$\pi(a \mid s)$	
S_L	A_0	1
S_L	A_i	0
S_S	A_0	0
S_S	A_i	1

} Policy

$S_0\ a_0 S_1\ r_1\ a_1\ S_2\ r_2\ a_2\ S_3\ r_3 \ldots$

$S_L\ A_0 S_S\ 1\ A_i\ S_S\ 1\ A_i\ S_S\ 1 \ldots$

Trajectory

$$G_0 = 1 + \gamma \cdot 1 + \gamma^2 \cdot 1 + \ldots$$

$$= \frac{1}{1-\gamma} = \frac{1}{1-0.9} = \frac{1}{0.1} = 10$$

$$v_\pi(S_L) = E_\pi\left[G_T \mid S_t = S_L \right] = \frac{1}{1-\gamma} = 10$$

$$v_\pi(S_S) = E_\pi\left[G_t \mid S_t = S_S \right] = \frac{1}{1-\gamma} = 10$$