REINFORCEMENT LEARNING

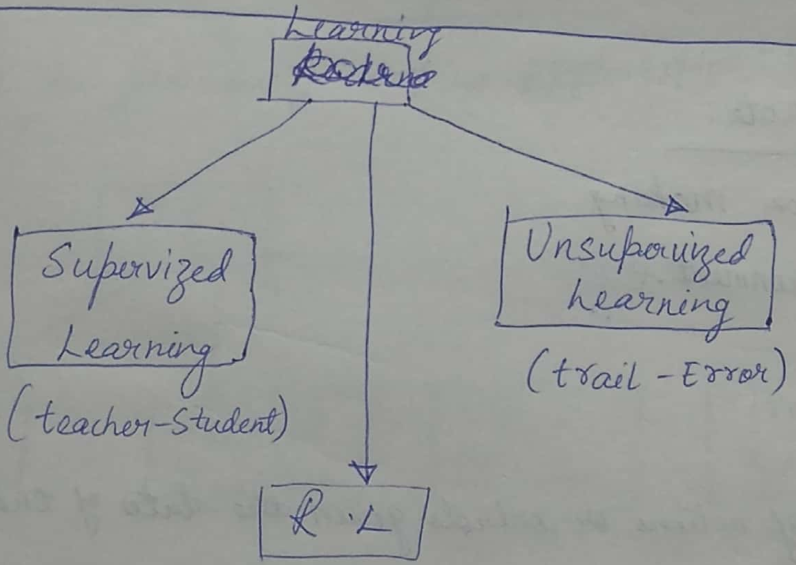
Date 10-Aug-2018

→ Richard Sutton & Barto :- http://incompleteideas.net/book/the-book-2nd.html
~~ed~~: Online draft -- II edition

→ Reference for prob : ece 313 - fa 2016 - illinois - UIUC
[Bruce Hajek]

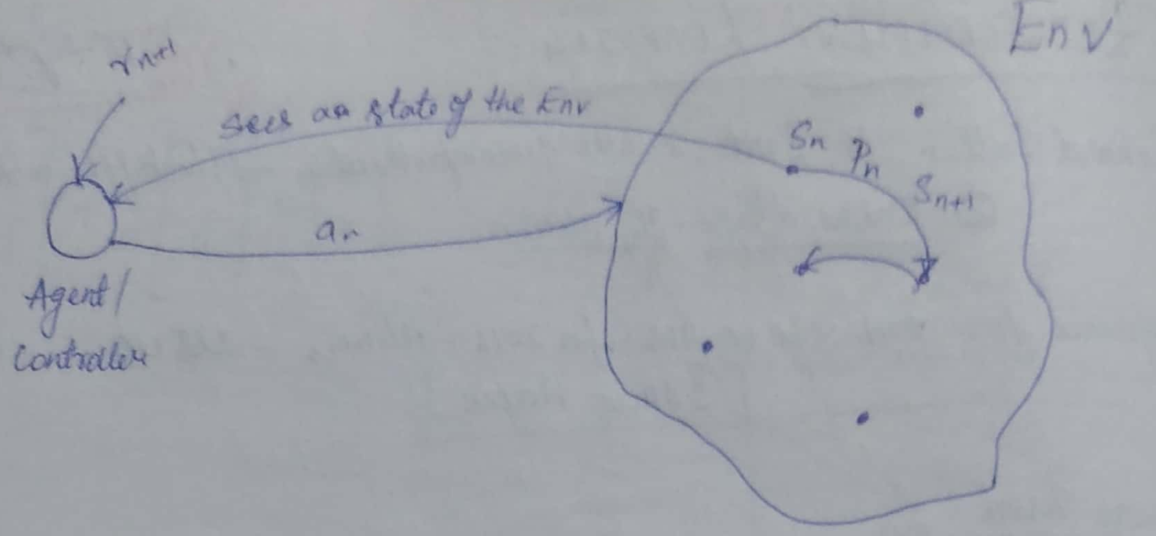
Course Eval :

→ Project evaluation
→ Mid term
→ Final Exam
→ Assignments.



A4 College Book
Kavitha

Scanned by CamScanner



S_n - State at time 'n'

a_n - action / control at time 'n'

r_n - reward at time 'n'

P_n - Prob that env moves to state S_{n+1}
(given that current state $= S_n$ & agent picks control / action a_n)

Important things to note:

1. Dynamic decision making
2. Uncertain environment.

Goal of the agent:

Learn a sequence of actions or controls given the state of the environment in order to max long term reward.

※ Agent is oblivious to the model of the system

$$\max E\left[\sum_{n=0}^{N} r_n\right] \quad \text{— ①} \quad \leftarrow \text{if the time horizon is finite.}$$

When $N = \infty$,
eqn ① doesn't fit since $\sum_{0}^{\infty}$ is always towards maximum

Hence, <u>discounted cost problem</u>

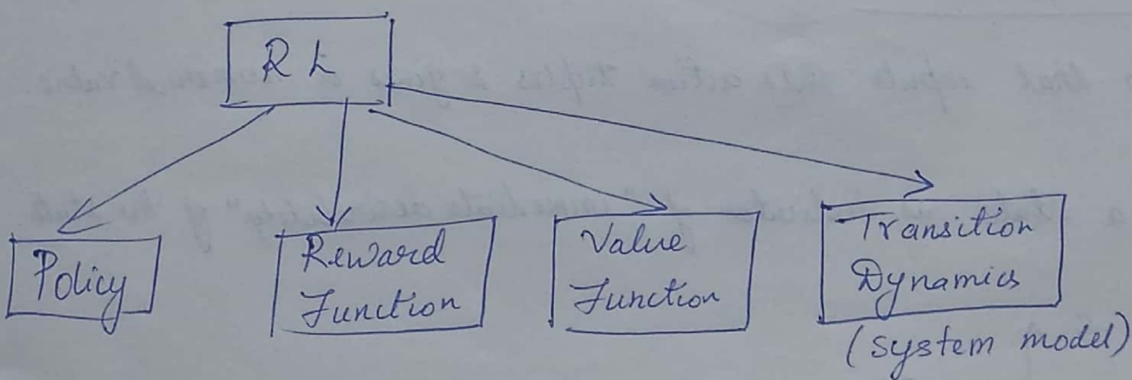
$$\max E\left[\sum_{n=0}^{\infty} \gamma^n r_n\right]$$

E – Expected Value.

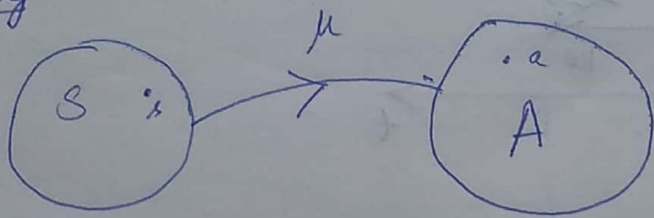
where discount factor $\gamma \in (0,1)$

<u>Exploration vs Exploitation</u>

Need to judiciously combine ~~Expl~~ explor^n & exploit^n



1. <u>Policy</u>: A function from states to action

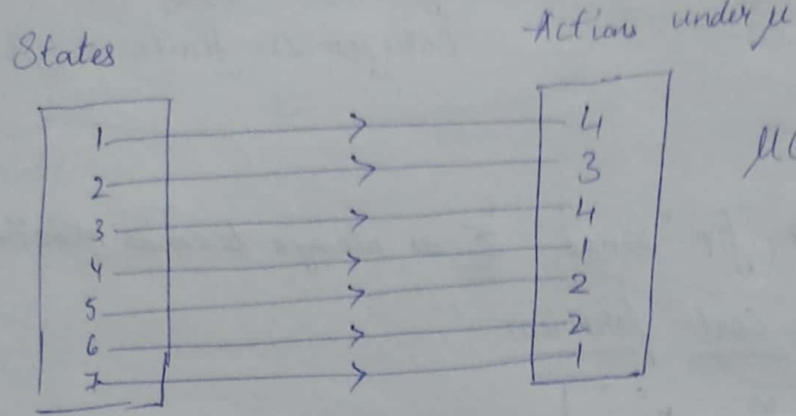


S : Set of all states

A : Set of all actions.

$$\mu(s) = a$$

Eg: States = 7, action = 4, considering a deterministic policy μ



$\mu(1) = 4$

probabilistic actions:

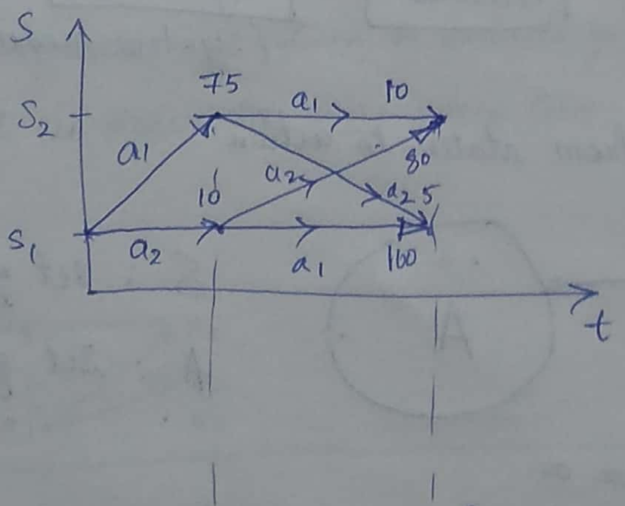
$\phi(s,1) = 0.0$
$\phi(s,2) =$

$$\phi = \phi\,(\,\phi(s,a)\,,\ s \in S,\ a \in A)$$

2. Reward function

A map that inputs state action tuples & gives a numerical value.

→ Reward in a state is indicator of "immediate desirability" of the state.



a_2 @ 0 is prefered

a_1 @ 0 is preferred

∴ long term vs short term desirability

3. Value function

A map that inputs states or state-action tuples and output a numerical quantity.

- Value function tells us the "long-term" desirability
- Decision making will involve max value f^n
- A state may get a low reward yet have high value

4. Model of the environment

→ model emulates the environment

→ Random - state transitions (transition prob $s_i \to s_j$)

$$p(s_1, a_1, s_2)$$

(the randomness is because of $s_i \to s_j$ is not fixed)

Temporal difference learning methods

For a given policy (fixed for the entire time duration), estimate the value function.

$$V(s) \longleftarrow V(s) + \alpha \left[V(s') - V(s) \right]$$

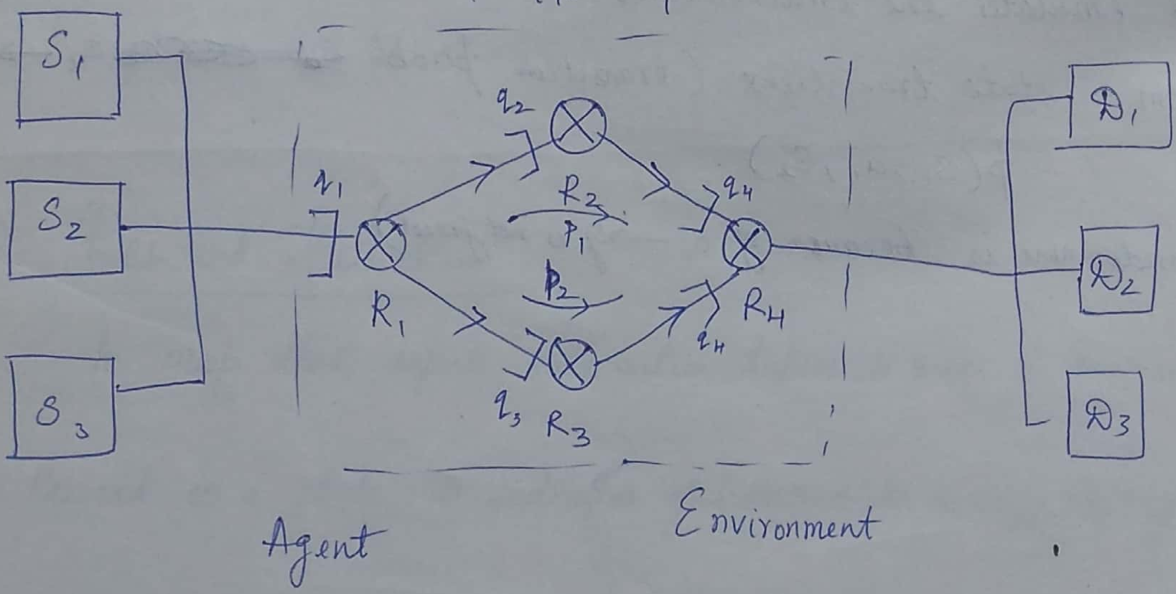
(new estimate) of value of s

$s \longrightarrow$ current state

$s' \longrightarrow$ next step.

$\alpha \longrightarrow$ a small number (increment)

p_1 & p_2 - paths



$$S = (q_1, q_2, q_3, q_4)$$

q_i = no of packets at router i, $i = 1, 2, 3, 4$.

$$a = (p_1, p_2) \in \{(1,0) \text{ or } (0,1)\}$$

either p_1 or p_2 can be take not both at same time.

Objective : minimize delay

$p_1 \longrightarrow -(q_1+q_2+q_4)$

$p_2 \longrightarrow -(q_1+q_3+q_4)$

negative state
(have least of no of packets in the queue waiting)

$$\max_{\{p_1, p_2\}} E\left[\sum_{n=0}^{\infty} r^n g_n\right]$$

Tic - Tac - Toe

Goal: learn optimal policy for player given that opponent plays same policy.

State description :-

$S = \{S_1, S_2, S_3 \ ... \ S_9, S_{10}\}$

who played first

$S_1, S_2 .. S_9 \in \{e, x, o\}$

$S_{10} \in \{x, o\}$

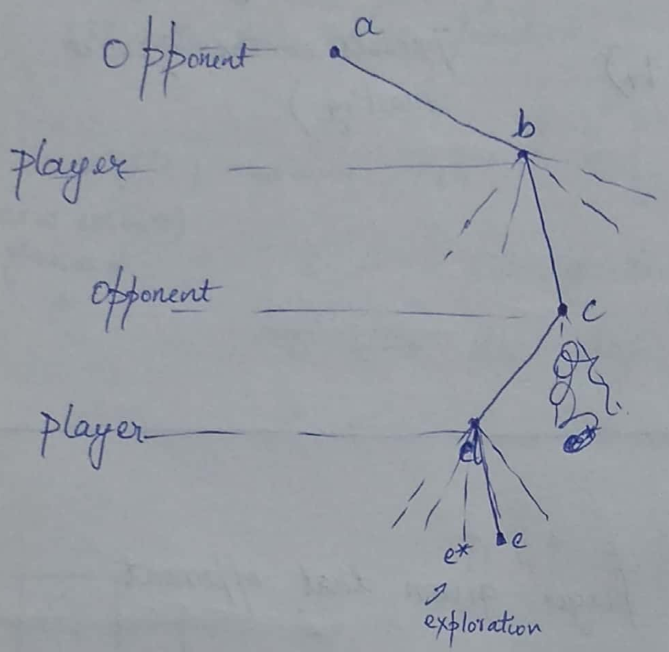
S_1	S_2	S_3
S_4	S_5	S_6
S_7	S_8	S_9

e – empty
x – player1
o – player2

Reward = $\begin{cases} 1 & \text{if player has won} \\ 0 & \text{otherwise} \end{cases}$

The game can be represented as tree structure,

Tree structure (suppose opponent is first player) & $S_{10}=0$



Q1: Suppose both player & opponent use same R.L algo to learn their moves.

Case 1: Reward $\begin{cases} 1 & \text{if Player/opp wins} \\ 0 & \text{if drawn} \\ -1 & \text{if player/opp looses} \end{cases}$ Say S_{10} - random

Zero sum game,

In case 1, since the rewards are symmetric & thus they learn same policy. & learning will converge.

Case 2: Reward $\begin{cases} 1 & \text{if player wins} \\ 0 & \text{other wise} \end{cases}$ Say S_{10} - random

in this case, learning might not converge, as the rewards is not zero sum. Since both the players try to outsmart other unlike typical RL setup one player against Env so for two player RL games, symmetric rewards ensure convergence.

Q2: Should symmetrically eqt positions have same value

A: yes

X	O	X
O		

S_1

$\equiv$

X	O	X
		O

S_2

$$V(s_1) = V(s_2)$$

Q3: Greedy play: Suppose RL player is greedy, i.e., always in exploit mode, What are the problems?

Ans:
→ smarter opponents ~~will affect~~ in future might affect player

→ policy learnt might be the best as the different rewards are not explored.