31st July '25

Reinforcement Learning

valuation for a particular state for future to get the greatest amount - Value function.

pure method - trial and error method (most of them).
expecting value of each and every state of (probability) } methods.
model based methods are also used

→ evaluationary methods - hill climbing method
(searching in space of probability)
↓
can also be used to solve reinforcement learning

if you can solve the problem it would be more optimal solution for RL.

→ ~~regularly used~~
in evaluationary methods - small space is given
- large branching factor (10^{22})
- stochastic environment

in the above case it is more effective

eg: TicTacToe game - 9 initial states, moving forward reduces.
↳ learn agent through reinforcement learning.

classical methods of game theory
↳ minmax algorithm - player is imperfect (considering)
↳ search space discovers the narrow area to learn the model but it cannot learn efficiently

agent - could be in draw.
cannot giver you upto date solution.

classical approaches : dynamic approach (required prior information) not optimal

for any RL problem (no prior information is given)
it needed to gather information from experience

→ branching factor in this game is comparatively less so evaluationary methods can be used.

losing the game → reward 0
winning the game → reward 1.
probability for all states → 0.1
subsequently change the policy. (reward is changing at the time of interaction with the environment)

→ one state to another state movement with greatest reward. but sometimes we also need to explore (may be not having the greatest reward. (exploratory action)

dilemma of exploration & exploitation.

A (0.6) → killing the probability. (losing the game) → B (0.3)

so, move to the state that are close to that probability

* A's prob > B's prob. → decrease the probability of state A & viceversa.
(updating parameter in terms of learning) → for refining the environment

unsupervised learning - choosing the gradient
* but in RL → yourself findout the next state given the environment

the above is the.
Simples approach of the reinforcement learning

4th Aug/25

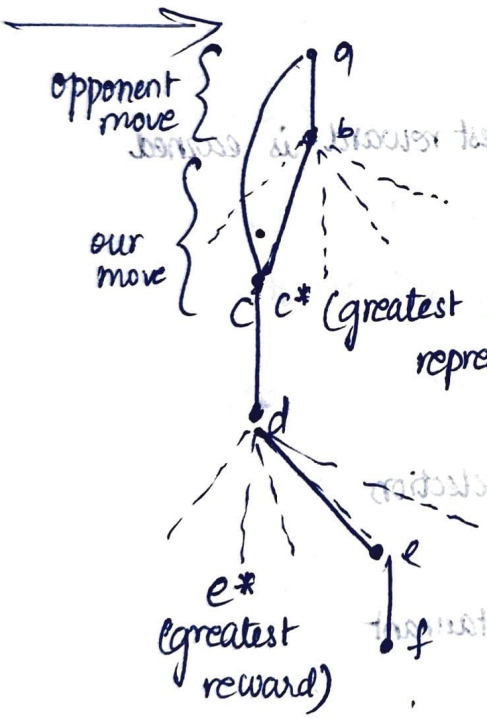
not unsupervised learning
↳ finding pattern

→ here it is not the primary goal.

evolutionary methods → cant be optimal for large searching space

Tictactoe :

minimax } classical techniques
dp ——→ creating model of opponents moves.
evolutionary methods.
RL method



↗ temporal difference learning method

$$V(S) \leftarrow V(S) + \alpha [V(S') - V(S)]$$

- $V(S)$: state before greedy move
- α: small +ve fraction (step-size parameter) ↳ influences rate of learning
- $V(S')$: state after the move

Temporal difference:

t+1 prediction is better than prediction at t ; Intuition

Explore - Exploit dilemma:

(no supervision before so, we encapsulate this)

Simplest ~~version~~ of RL → Bandit problems

problems. encapsulate explore v/s exploit

Slot machine → collecting money.

↓ few may hit the jackpot (or) nothing.

can be a type of Bandit

(Sutton & Barto). — book name Chptr-2

Multi-Arm Bandits. analyzing how a best reward is earned

RL problem : given diff states and accordingly choose action & gain the reward

Exploration & exploitation eg: restaurant selection

↓ trying a new restaurant

↳ going to the favourite restaurant

q_* → true expected value after performing an action

Action: (2) 88% chance of getting 0
12% chance of getting 100 } for this action

$$q_*(2) = 0.88 \times 0 + 0.12 \times 100 = 12.$$

(true value)

randomly -10 & 35 equiprobable. — Action (3).

deterministic → best action.
stochasticity → reduced (the chances are reduced to lose)

k-armed Bandit problem.
↳ distribution is unknown - so, knowing them & rewards are to be done.

exploration / exploitation dilemma:

$$Q_t(a) \approx q_*(a) \quad \forall a$$

$$A_t^* = \arg\max_a Q_t(a)$$

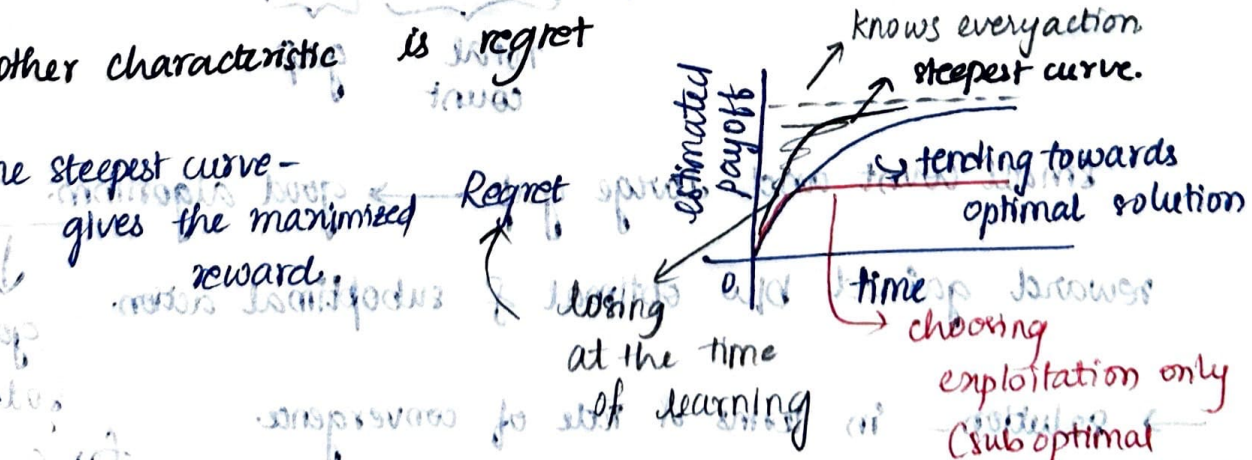
(greedy action)

$A_t = A^*_t$ → exploiting. } mathematically.
$A_t \neq A^*_t$ → exploring. }

→ **Asymptotic correction:** no bounded condition.
for algorithms in every moment making an optimal action to solve RL.
any is asymptotic correctness of the problem
(guaranteed in future)

→ Another characteristic is **regret**

the steepest curve - gives the maximized reward.



less regret area but not optimal soln.

Regret:

6th Aug '25:

$$V^* = Q(a^*) \max_{a \in A} q^*(a)$$

estimated optimal value.

Regret → opportunity loss for one step

$$l_t = E[V^* - Q(a_t)]$$

total regret - total opportunity loss.

$$L_t = \mathbb{E}\left[\sum_{\tau=1}^{t} V^* - Q(a_\tau)\right]$$

↳ timestamp=1

→ how many times we are considering an action - count

→ $\Delta a = V^* - Q(a)$

gap. estimated value of action a.

$$L_t = \mathbb{E}\left[\sum_{\tau=1}^{t} V^* - Q(a_\tau)\right]$$

$$= \sum_{a \in A} \mathbb{E}\left[N_t(a)\right](V^* - Q(a))$$

$$= \sum_{a \in A} \mathbb{E}\left[N_t(a)\right] \Delta a$$

total count, gap

small count and large gap → good algorithm.

reward gained b/w optimal & suboptimal action.

→ good solution for a bandit problem

→ solution - in terms of rate of convergence.

→ forever exploring & never exploring gives Linear regret

ϵ-greedy (epsilon greedy algorithm) - balances exploiting & exploring

decaying ϵ-greed →

Solution:

1) Asymptotic correctness

2) regret optimality.

3) pac optimality - probably approximately correct

↳ ϵΔpac optimality (Epsilon delta)

↓

probability of correctness

hypothesis : close to optimal model.

* $Q^*(a)$ - expected/best true value $\geq Q^*(a^*) - \epsilon$ ↳ 0.01

↑

$(\forall \epsilon \Delta)$

probability of this occurrence should be greater than $(1-\Delta)$

if this is guaranteed solution is having the pac optimality

⇒ no solution - log, theory.

Complexity of regret

Total regret should always be greater than $\log t$ where t is the time stamp.

$$\lim_{t \to \infty} \alpha_t \geq \log t \sum_{a | \Delta_a > 0} \frac{\Delta a}{KL(R^a || R^{a^*})}$$

Hard problems have similar looking with different means.

(difference in distributions)

$\Delta a \rightarrow$ gap b/w optimal & other arm.

$$\sum_{a | \Delta_a > 0} \frac{\Delta a}{KL(R^a || R^{a^*})}$$

→ if this is large

→ if this is small

then the total regret would be large

Action - value methods:

→ Epsilon - greedy method (ϵ is considered in such way those many times the actions are explored)

90% exploit 10% explore

disadvantage - so much computation & memory usage coz keeping track of all records & values would take large time & storage.

→ Incremental implementation.

→ Stationary v/s non stationary environment

distribution of action is not changed (fixed)

distribution of action is changed

→ Optimistic initial values

1st requirement to solve bandit problem – exploration & exploitations.

timestamp 0 → initialised every action = 0.

updating according to every action.

expected value > optimal value (Explore more to reach optimality)

UCB action selection method:

considering uncertainity → not done enough sampling for a particular action

Gradient bandit algorithm:

finding preferences of the action. → function

Chptr2 : Multi-Arm Bandit

#HAngles

Associative search: Considering on the basis of recomendations & taking the actions.

Episode: multiple time stamps.

ACTION - VALUE METHODS:

↳ stationary environment

↳ average of past rewards in terms of estimation of the rewards.

$$Q_t(a) = \frac{\text{Sum of rewards when a taken prior to } t}{\text{number of times a taken prior to } t}$$

$Q_t(a)$ — estimation at a timestamp

$$\lim_{N_t(a) \to \infty} Q_t(a) = q_*(a)$$

(converging to true values)

ε- GREEDY ACTION SELECTION:

ε - value of epsilon is considered such a way to balance exploitation & exploration.

eg:

A - one particular action.

In this, simple bandit problem we consider only one particular action at a time.

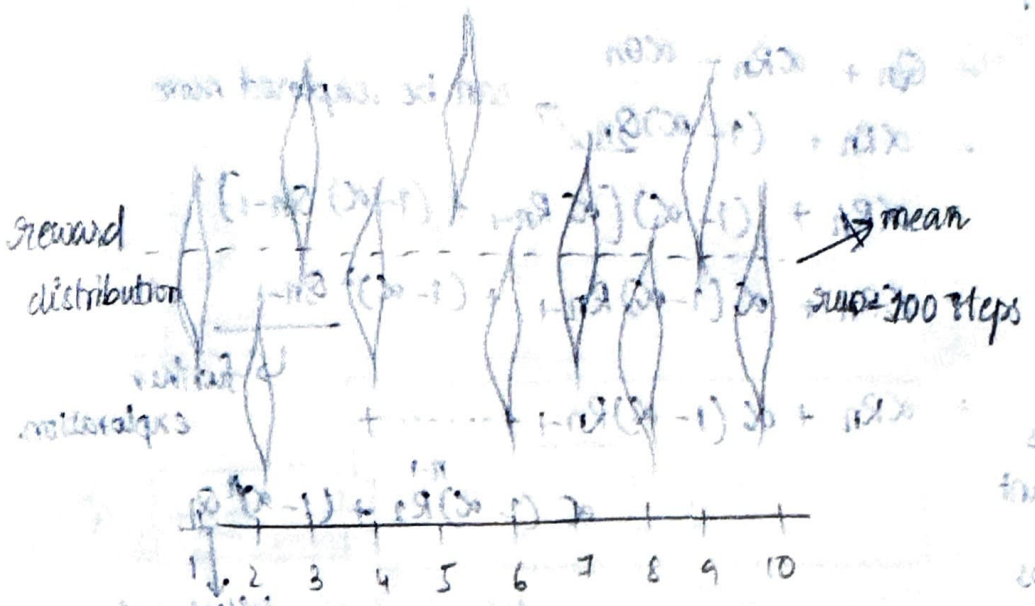
$N(A) \leftarrow N(A) + 1$ (increasing & checking other actions)

$Q(A) \leftarrow Q(A) + \frac{1}{N(A)}[R - Q(A)]$ updating values using incremental update formula

$N(A)$ → no. of times

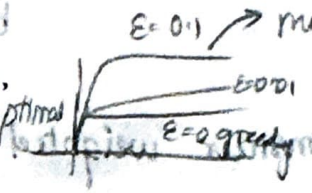
$R - Q(A)$ → reward − old estimate value

10- Armed Testband



greedy methods:
↳ average reward

ε-greedy method - optimal action.



↳ the greater the ε value
the greater is the reward obtained

Tracking a Non-stationary problem:
changes made:

→ Over time true value of every action is changed considering weighted averages.

$$Q_{n+1} = (1-\alpha)^n Q_1 + \sum_{i=1}^{n} \alpha(1-\alpha)^{n-i} R_i$$

R_i ↳ rewards

$(1-\alpha)^n Q_1$ → Initial action

$\sum_{i=1}^{n} \alpha(1-\alpha)^{n-i} R_i$ → weighted averages

→ more is the timestamp the more is its weight
i.e Q_1's weight < Q_{10}'s weight

$$Q_{n+1} = Q_n + \alpha[R_n - Q_n]$$

$$= Q_n + \alpha R_n - \alpha Q_n$$

$$= \alpha R_n + (1-\alpha) Q_n$$ ↗ can be explored more

$$= \alpha R_n + (1-\alpha)[\alpha R_{n-1} + (1-\alpha) Q_{n-1}]$$

$$= \alpha R_n + \alpha(1-\alpha) R_{n-1} + (1-\alpha)^2 Q_{n-1}$$

↳ further exploration.

$$= \boxed{\alpha R_n + \alpha(1-\alpha) R_{n-1} + \cdots + \text{till } (n-1)\ \alpha(1-\alpha)^{n-1} R_1} + \boxed{(1-\alpha)^n Q_1}$$

↓ initial value.

α - value is constant

value of steps can be changed at each timestamp

consider Q_1 value ↗ realistic / optimalistic

↓ biased.

Q) why do we consider weighted averages for non stationary environment

steps → finite (sub-optimal rewards)

$$\sum_{n=1}^{\infty} \alpha_n(a) = \infty \Rightarrow \text{converging to optimal reward}$$

$$\sum_{n=1}^{\infty} \alpha_n^2(a) < \infty \Rightarrow \text{estimation + noice is incorporated}$$

overtime noice is reduced.

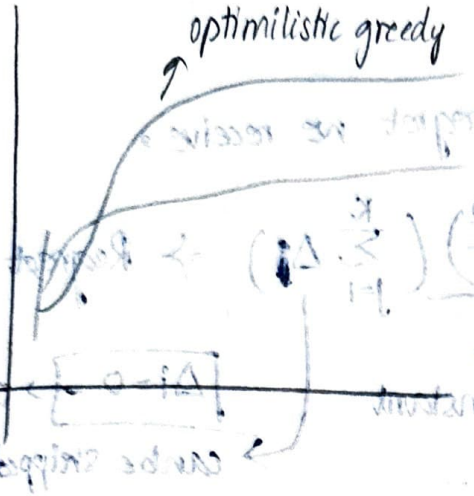
shrink the spicy part

to do this ↘ condition should be satisfied

$\alpha_n = \frac{1}{n}$ and not $\frac{1}{n^2}$ → here it is ok only positive values, so not considered to be normal.

↓

graph can be both negative & positive values.



18th August '25

one particular arm to give the optimal solution, exploration of all arms is required & considering the uncertainity level of every arm

$Q_j \neq \sqrt{\frac{2 \ln n}{n_j}}$ → how many number of times we have sampled arm

Q_j → estimation of each arm

$\sqrt{\frac{2 \ln n}{n_j}}$ → uncertainity level

$Q(1)$ $Q(2)$ $Q(3)$ $Q(4)$ → highest reward ??

prob of Q_3 is very less compared to Q_4

total expectation of each arm

as sampling increasing the confidence. with more exploration of an action.

Q_1 Q_2 Q_3 Q_4 → ! increased the

high reward

UCB algorithm,

Theorem: Almost the total regret we receive.

$$8\sum\left(\frac{\alpha(n)}{\Delta i}\right) + \underbrace{\left(1+\frac{\pi^2}{3}\right)}_{\text{constant}}\left(\sum_{j=1}^{K}\Delta i\right) \Rightarrow \text{Regret}$$

$i = a_*(i) < q_*(a^*)$

$\Delta i = 0 \rightarrow$ optimal

can be stripped if ↑

i (arms)
sub optimal arm.
incurring some loss while performing.

$$\Delta i = q_*(a^*) - q_*(i)$$

Terms used in theorem:

$T_i(n) \rightarrow$ no. of times arms has been upto n timestamps.

regret $n \rightarrow \sum_i E[T_i(n)]\,\Delta i$

(difference b/w expected value & suboptimal action)

$X_{i,n} \rightarrow$ Random variable for reward that has been obtained for playing action i at nth timestamps.

$E(X_{i,n}) = q_*(i)$

expectation of that random variable.

→ we would like to show.

$$E[T_j(n)] \leq \frac{8}{\Delta i^2}\ln(n) + c$$

Concentration bound:

creating some bound around a variable which should not be exceeded.

→ Shernoff concentration bound:

the probability for expected value we achieved is not too far from true value.

↳ dependent of timestamps & less value we are considering.

⇒ Let $X_1, \ldots X_n$ be a RV with common $[0,1]$ such that

↳ stationary problem.

$$E[X_t \mid X_1, \ldots X_{n-1}] = \mu$$

↳ expected reward.

let $S_n = \frac{X_1 + \cdots X_n}{n}$ $\forall \epsilon \geq 0$

bandit problem considers → stationary environment

According to this bound, one particular action doesn't cross a particular value.

$$P(S_n \geq \mu + \epsilon) \leq e^{-2\epsilon^2 n}$$

$$P(S_n \leq \mu - \epsilon) \leq e^{-2\epsilon^2 n}$$

value of $\epsilon = 0.1$ ← there are 2 cases

value of $\epsilon = 0.01$

in which case the probability will increase.

As, $\uparrow \epsilon \rightarrow P \downarrow$ (probability decreases)

(increase in epsilon)

Proof of theorem :-

20th Aug '25

Let $C_{n,s} = \sqrt{\frac{2 \ln(n)}{s}}$

$\{I_n : i\}$ let it is arbitrary variable with scalar value

$$T_i(n) = 1 + \sum_{m=k+L}^{n} \{I_n : i\}$$

$$\leq m + \sum_{m=k+L}^{n} \{I_m : i, T_i(m-1) \geq l\}$$

$$\Rightarrow \leq l + \sum_{m=k+1}^{n} \{ Q_{m-1}(a^*) + C_{m-1}, T_{a^*}(m-1) \leq$$

$$Q_{m-1} T_i(m-1), T_i(m-1) \geq l$$

$$P(S_n \geq \mu + E) \leq e^{-2\epsilon^2 n}$$

$$P(S_n \leq \mu - E) \leq e^{-2\epsilon^2 n}.$$

at each & every timestamp,

whatever the minimum estimated value of a^* in a particular cycle is less than the estimated value

$$\Rightarrow \leq l + \sum_{m=k+1}^{n} \{ \min_{0<s<m} (Q_s(a^*) + C_{m-1}, (a^*(s)) \}$$

$$\leq \max_{l \leq i \leq n} (Q_{s_i}(i) + (m-1, T_i(s_i)\}$$

lower bound

making upper bound for a particular count

in each & every step there are a^* values

$$\Rightarrow \leq l + \sum_{m=1}^{\infty} \sum_{s=1}^{m-1} \sum_{s_i=l}^{m-1} \{ Q_s(a^*) + C_m, T_a^*(s) \leq Q_{s_i}(i) + C_m, T_i(s_i)\}$$

whatever the probabilities we are getting

$$E[T_i(*)] \leq \left[\frac{8 \ln(m)}{\Delta i^2}\right] + \Sigma\Sigma\Sigma 2m^{-4}$$

3 conditions:

→ under counting/estimated optimal arm value - we can make a mistake.

1) underestimated optimal R then we can make a mistake.

2) overestimated - can make a mistake

3) true value of optimal & suboptimal arm's mean is almost nearer even in that case we can make a mistake.

→ underestimating a^*

① $\hat{Q}(a^*) \leq q_*(a^*) - c_{m,T_{a^*}(s_i)}$

② $\hat{Q}(s_i) \geq q_*(i) + c_{m,T_i(s_i)}$

③ $q_*(a^*) < q_*(i) + q c_{m,T_i(s_i)}$

concentration bounds.

↳ uncertainity is more b/w optimal & suboptimal so, the reward for it is not known clearly & tend to take a wrong choice.

① $P(Q_s, a^*) \leq q_*(a^*) - c_{m,T_{a^*}(s)}$ $\leq m^{-4}$ → eqn

a^* value at Q_s timestamp.

confidence interval.

Solving this eqn with $P(\bar{s}_n \geq \mu + \epsilon) \leq e^{-2\epsilon^2 n}$,

we get $e^{-4\frac{\ln(m) \times 2}{T_{a^*}(s)}} = e^{-4\ln m}$.

② $P(Q_{s,i}(i) \geq q_*(i) + C_{m,T_i(s,i)}) \leq m^{-4}$

③ $l = \left[\frac{8 \ln(m)}{\Delta i^2} \right]$

$q_*(a^*) - q_*(i) - 2 \ln T_i(s,i)$ → solve this eqn with

we get this value as $\frac{1}{2}\Delta i$

$C_{n,s} = \sqrt{\frac{2 \ln(m)}{s}}$

$q_*(a^*) - q_*(i) - \Delta i = 0.$

↓ suboptimal ↓ regret

$\text{Regret}_n = E[T_i(a)] \Delta i$

previous methods — Action value methods.

21st Aug'25

Gradient - Bandit Algorithm:

↳ concerns with every preference action.

$H_t(a)$ → preference function at particular action at time t

Initial preference ↳ '0' for all actions.
(zero)

$\pi_t(a)$ → softmax function

↓

$P_t(A_t = a)$

$H_1(a) = 0$ → initial preference of all actions for all actions.

gradient ascent / descent preference can be changed via these.

→ for one action - inc ~~de~~ preference

→ for all other actions → dec preference based on reward

baseline - $\bar{R}_t$ (mean) $\neq 0$ in this case

X_t → to introduce baseline.

Bandit Problem:

MDPs → useful to represent full RL problem in a more formal way.

25th Aug'25

States: State / environment in Bandit problem

Action on Same environment.

here → different states

which can be merged on the basis of prev time steps and action that we took.

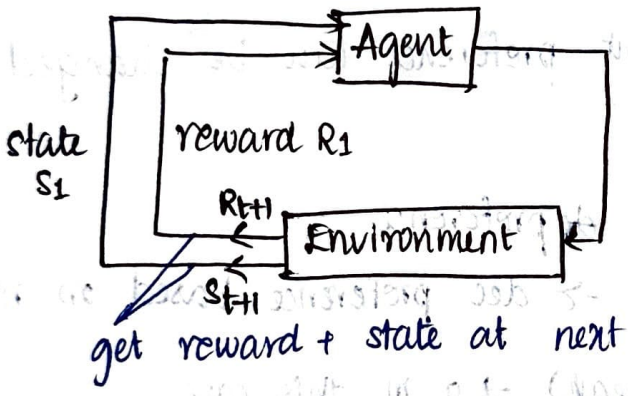
→ Delayed reward

→ Return

→ Value function (wrt each state)

→ Evaluation feedback (on the basis of each & every state)

} full RL problem



Finite MDP:

$$P(S_t = S', R_t = r \mid S_{t-1} = S, A_{t-1} = a) = P(S', r \mid S, a)$$

what does this function describe?

→ info about 'complete' RL environment!

Agent - environment interface:

$$\sum_{S' \in S} \sum_{r \in R} P(S', r \mid S, a) = 1.$$

→ we can sum probabilities for every possible value of stall and reward

★ A state having markov property → this state has all information about all past agent environment information that matter

for every state - we define some set of actions.

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

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

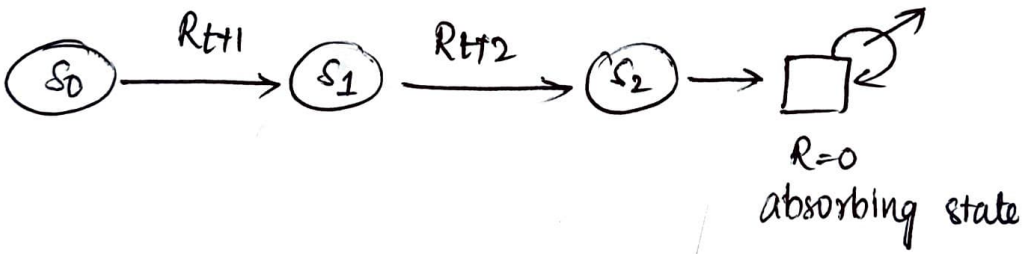
$$0 \leq \gamma \leq 1$$

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

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

$$= R_{t+1} + \gamma (R_{t+2} + \gamma R_{t+3} + \cdots)$$

$$G_t = R_{t+1} + \gamma G_{t+1}$$



POLICY: mapping of states to actions.

$\pi(a|s)$

$$v_\pi(s) = E[G_t | S_t = s]$$

↓

state value function.

$$q_\pi(a,s) = E[G_t | A_t = a, S_t = s]$$

↙ return