Eg:
Recycling Robot

16-Sept-2018

Mobile robot, — job to collect empty cans

High level decisions – how to search for cans – depends on the current charge level of battery.

State space $S = \{high, low\}$ (states correspond to charge level.)

In each state, agent needs to decide ~~whether to~~ wheter to
(i) actively search for a can for a certain period of time
(ii) remain stationary and wait for someone to bring a can
(iii) Head back home to recharge its battery.

Let $A(high)$ and $A(low)$ denote the action sets in states high and low respectively.

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

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

rewards are zero most of the time.

rewards = +1 when it gets a can.

state s	action a	next state s'	prob next state given current state action $p(s' \mid s, a)$	reward $r(s, a, s')$

PTO

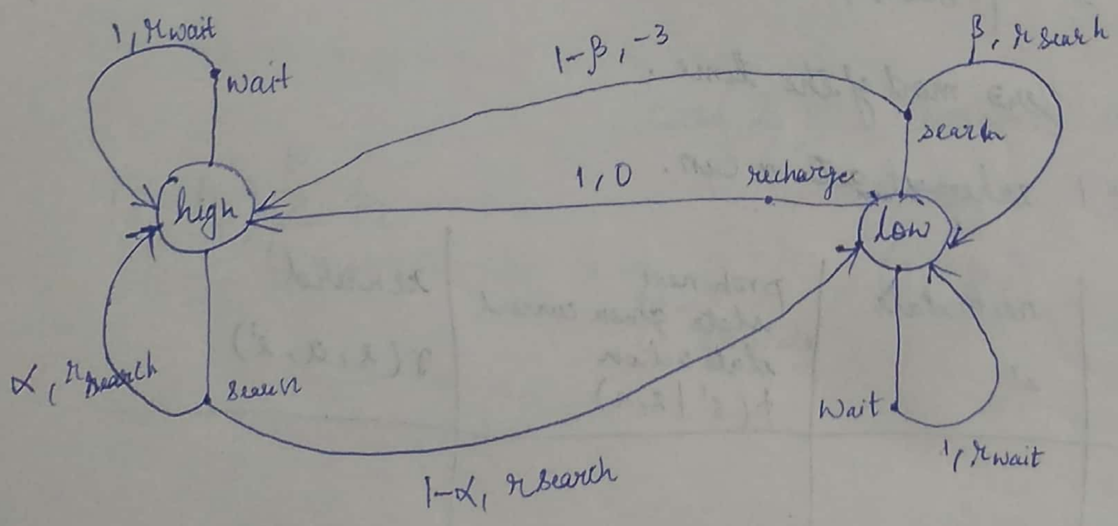
s	a	s'	$p(s' \mid s, a)$	$r(s, a, s')$
high	search	high	α	r_{search}
high	search	low	$1-\alpha$	r_{search}
low	search	high	$1-\beta$	-3
low	search	low	β	r_{search}
high	wait	high	1	r_{wait}
high	wait	low	0	r_{wait}
low	wait	low	1	r_{wait}
low	recharge	high	1	0
low	recharge	low	0	0

($1-\beta$) is charge becoming zero (someone picks the bot & recharges)

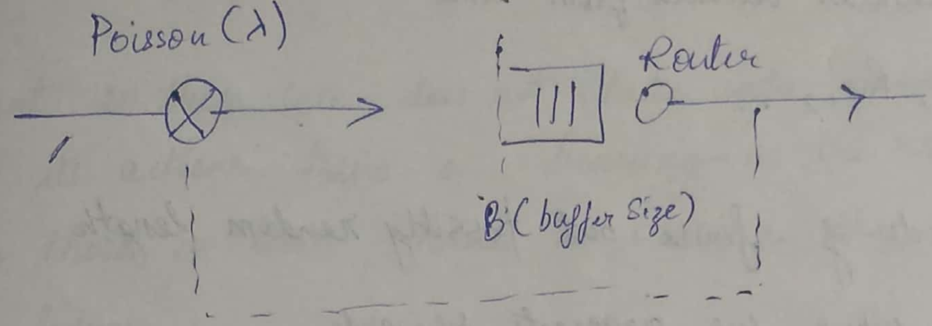
r_{search} — expected reward while searching

r_{wait} — expected reward while ~~the~~ waiting.

$$\boxed{r_{search} > r_{wait}}$$



Goals & rewards:



Goal: keep the queue length most of the time around N_0 such that $0 < N_0 < B$.

To balance conflicting objectives such as maximise throughput, minimize delay etc.

states $\in \{0, 1, 2 \ldots B\}$ [No of packets in buffer]

Action $\longrightarrow$ rate at which packets are sent to buffer

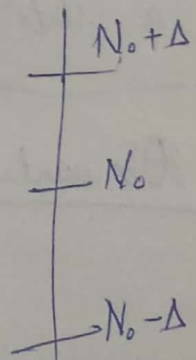
$\lambda \in \{\lambda_1, \lambda_2, \lambda_3\}$ where $\lambda_1 < \lambda_2 < \lambda_3$ (rate of packets)

rewards $= \dfrac{1}{|q_n - N_0| + 1}$ no of packets at time $n = q_n$

or ($-|q_n - N_0|$, or exponential)

A "good" policy

$$\lambda_n(q_n) = \begin{cases} \lambda_1 & \text{if } q_n > N_0 + \Delta \\ \lambda_2 & \text{if } N_0 - \Delta \le q_n \le N_0 + \Delta \\ \lambda_3 & \text{if } q_n < (N_0 - \Delta) \end{cases}$$



Returns & episodes

Sequence of rewards obtained from time t onwards,

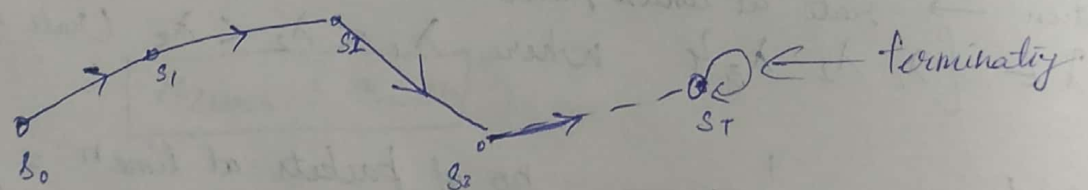
$R_{t+1}, R_{t+2} \ldots$

Episodes: Periods of finite but possibly random length on which we aggregate rewards.

Example: each episode starts with a certain state S and ends in a "terminating state" S_T (starting state need not be fixed) & lengths can be different lengths.

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

Note: each episode can have different rewards G_t.



Intial state couldbe either same state as in the previous episode or a state picked according to some distribution.

Discounted Case:

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

$$= \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \quad \text{where } 0 \leq \gamma \leq 1$$

is a parameter

In case of $\gamma = 0$, the agent is "myopic" as it considers only immediate rewards.

Agent in this case does not take into account the fact that its actions have a bearing as the next state & as those in turn have a bearing that will come in future.

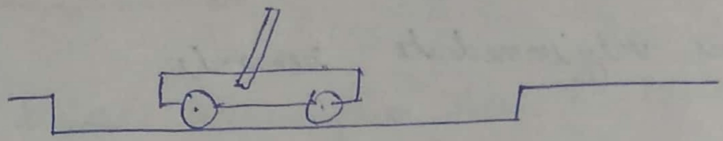
suppose $|R_{t+1}| \leq \beta \leq \infty \quad \forall t, k$

Then
$$|G_t| = \left| \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \right|$$
$$\leq \sum_{k=0}^{\infty} \gamma^k |R_{t+k+1}|$$
$$\leq \beta \sum_{k=0}^{\infty} \gamma^k = \frac{\beta}{(1-\gamma)}$$

Note that,
$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots$$
$$= R_{t+1} + \gamma (R_{t+2} + \gamma R_{t+3} + \dots)$$
$$G_t = R_{t+1} + \gamma G_{t+1}$$

Suppose we are in episodic case, where $t < T$ and we let $G_T = 0$.

Example: Pole Balancing:



Goal: Balance the pole by applying force to the cart

- A failure occurs if the pole falls beyond a certain angle from vertile position.

- Episodic task where the natural episodes are the repeated attempts to balance the pole.

- Each episode starts ~~that~~ pole reset to the vertical position after each failure and ~~eds~~ ends when the pole falls past a given angle from the vertical or cart runs off the track.

Reward = +1 for every time step that failure did not occur.

succesful balancing forever would result in reward of $+\infty$

Alternatively, one can treat the pole balancing as continuity task using discontinuing. Here, one can let rewards = -1 whenever there is a failure and reward = 0. otherwise.

Return would be related to $-\gamma^k$ where k is the number of steps before failure.

Exercise

Suppose we are designing a robot arm to run a maze

Let reward = +1 for escaping from the maze
= 0 otherwise

episodic task episodes could be ~~successful~~ successive runs thro' the maze.

After running the learning agent for a while, you find that it is showing no improving in escaping from the maze.

What is going wrong? Have you effectively communicated to the agent what you wanted to achieve?

In the case $G_t = 1$, regardless of the episode we are in,

⇒ It is not learning to escape from the maze "fast enough"

(Change rewards to 0 & 1
or −1 & 1 or discounted rewards)

Exercise:

Suppose $\gamma = 0.5$, and the following sequence of rewards is received. $R_1 = -1$, $R_2 = 2$, $R_3 = 6$, $R_4 = 3$, $R_5 = 2$ with $T = 5$

What are $G_0, G_1, G_2, G_3, G_4, G_5$?

Sol^n go backwards.

$$G_{T-1} = R_T + \gamma G_T \quad \text{where } G_T = 0$$

Since when you reach termination state, there is no reward & agent stays there.

$$G_{T-2} = R_{T-1} + \gamma G_{T-1}$$
$$= R_{T-1} + \gamma R_T$$

Episodic & Continuing task

$S_t \rightarrow$ state at time t

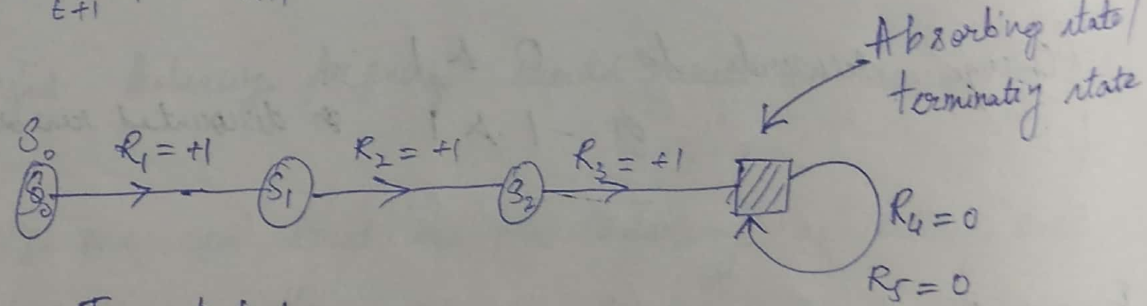
$S_{t,i} \rightarrow$ state at time t in episode i

$\pi_{t,i} \rightarrow$ policy at time t in episode i

$A_{t,i} \rightarrow$ action - - -

$R_{t,i} \rightarrow$ reward - - -

$$G_T = R_{t+1} + \gamma G_{T+1}$$



$$G_t = \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k$$

includes possiblities of $T = \infty$ - continuous

or

$\gamma = 1$ (but not both) - Episodic

Policies & Value functions.

Most RL algos estimate value functions (Value f^n gives the value of any state)

Policy: A mapping from state to the prob of selecting each possible action.

If policy chosen by agent = π

Then $P(A_t = a \mid S_t = s)$
$= \pi(a \mid s)$

$$\sum_{a \in A(s)} \pi(a \mid s) = \sum_{a \in A(s)} P(A_t = a \mid S_t = s) = 1$$

Example: Suppose $A(s) = \{a_1, a_2, a_3, a_4, a_5\}$

$\pi(a_1 \mid s) = 0.1$ $\pi(a_2 \mid s) = 0.4$ $\pi(a_3 \mid s) = 0.2$

$\pi(a_4 \mid s) = 0.1$ $\pi(a_5 \mid s) = 0.2$

Note that $\pi(\cdot \mid s)$ is a distribution on $A(s)$ conditioned on state being s.

Given state = s, set of feasible action = $A(s)$

policy π with pick action $a \in A(s)$ w.p $\pi(a \mid s)$

(So, MAB can be seen as a special case with state, confined to single state. Also note the policy is ~~same~~ constant for each each state)

Exercise: If the current state is S_t and actions are selected according to policy π, then ~~expectation~~ What is the expectation of R_{t+1} in terms of π & the four arguments f^n p? $(p(\underbrace{s', r | s, a}_{4 \text{ arguments}}))$

$$\boxed{\mathbb{E}[R_{t+1} | S_t, \pi] = \sum_a \pi(a|s) \sum_{s', r} p(s', r | s, a) \cdot r}$$

Value function of a state s under policy π $(v_\pi(s))$:

Expected return when starting in state s and following policy π thereafter.

$$V_\pi(s) = E_\pi[G_t | S_t = s]$$

$$= E_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \Big| S_t = s\right] \quad \forall s \in S$$

Similarly, one can also define the value of taking action a in state s under a policy π denoted $q_\pi(s, a)$

$q_\pi(s,a)$: Expected return when starting in state s, picking action a & subsequently following policy π

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

q_π is also called the action-value function corresponding to policy π

Exercise: Given an eqn for V_π in terms of q_π & π?

$$\boxed{V_\pi(s) = \sum_{a \in A(s)} \pi(a|s)\, q_\pi(s,a)}$$

& q_π in terms of V_π

$$q_\pi(s,a) = r_\pi(s|a) + \gamma \sum_{(s',r)} p(s',r|s,a) V_\pi(s) \cdot \pi(a|s)$$
$$= \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)\, r$$
$$+ \gamma \sum_{(s',r)} \pi(a|s)\, p(s',r|s,a) \cdot v_\pi(s)$$

We don't know system model, we can estimate v_π & q_π using G_t & taking averages over multiple episodes.

$$G_t = \sum_{k=t+1}^{T} \gamma^{k-(t+1)} R_k$$

$$v_\pi(s) = E_\pi\left[\sum_k \gamma^k R_{k+1} \mid S_0 = s\right]$$

~~nont~~ Monte Carlo Methods.

Value functions satisfy a recursive relationship

$$V_\pi(s) = E_\pi[G_t \mid S_t = s]$$

$$= E_\pi[R_{t+1} + \gamma G_{t+1} \mid S_t = s]$$

$$= E_\pi[R_{t+1} \mid S_t = s] + \gamma E_\pi[G_{t+1} \mid S_t = s] \quad \text{--- (1)}$$

$$= \sum_{a \in A(s)} \pi(a|s) \sum_{s'} \sum_{r} p(s', r|s, a)\left[r + \gamma E_\pi(G_{t+1} \mid S_{t+1} = s')\right]$$

in (1) $E_\pi \, G_{t+1}$ is future, but $S_t = s$ is state at t, to get reward at future $p(s', r|s, a)$ is needed since we have info of state at $t+1$, t is redundant hence $G_{t+1} \mid S_t = s \Longrightarrow (G_{t+1} \mid S_{t+1} = s')$

$$= \sum_{a \in A(s)} \pi(a|s) \sum_{s', r} p(s', r|s, a)\left[r + \gamma V_\pi(s')\right], \forall \in S$$

Bellman Eqn:

If $V_\pi = [V_\pi(s), s \in E]$

$$R_\pi = \left[\sum_{a \in A(s)} \pi(a|s) \sum_{s', r} p(s', r| s, a) \cdot r \quad s \in S\right]$$

$$P_\pi = \left(\left[\sum_{r, a} p(s', r|s, a)\, \pi(a|s) \quad s, s' \in S\right]\right)$$

In Vector - Matrix , we obtain

$$V_\pi = R_\pi + \gamma P_\pi V_\pi$$

Or $\cancel{(I - \gamma R_\pi)}\,(I - \gamma P_\pi) V_\pi = R_\pi$

PTO