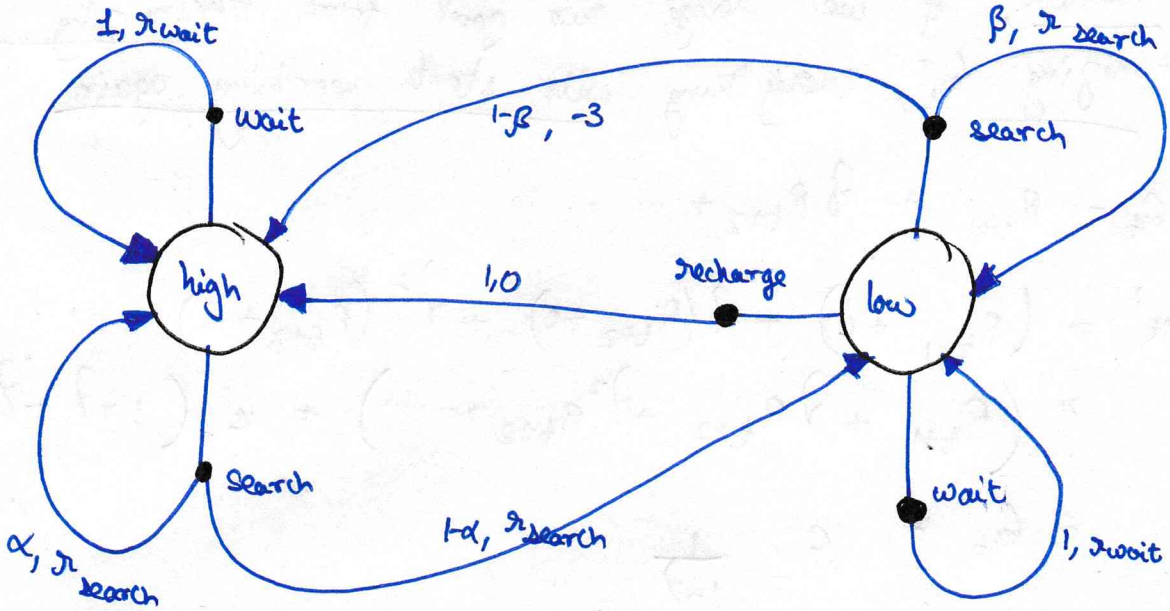
Setu Gupta
2018190

Reinforcement Learning
Dynamic Programming
ECE/CSE 564
Monsoon 20

8-1)



s	a	s'	r	p(s', r | s, a)
high	wait	high	r_{wait}	1
high	search	high	r_{search}	α
high	search	low	r_{search}	1-α
low	search	low	r_{search}	β
low	search	high	-3	1-β
low	wait	low	r_{wait}	1
low	recharge	high	0	1

This table formulates every thing that can happen in the environment.

Choose a (s, a) pair and then write down all possible (s', r) for it along with probabilities.

3-3) 3.15:

Signs are important. Suppose the signs of every reward is flipped and nothing else is changed, then the agent will be motivated to hit the walls.

However, if we change our goal from maximizing G_t to minimizing G_t, everything will start working again.

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

$$\begin{aligned} G_t' &= (R_{t+1} + c) + \gamma(R_{t+2} + c) + \gamma^2(R_{t+3} + c) + \cdots \\ &= (R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots) + c(1 + \gamma + \gamma^2 + \cdots) \\ &= G_t + c \cdot \frac{1}{1-\gamma} \\ &= G_t + \underbrace{\frac{c}{1-\gamma}}_{\text{a constant}} \end{aligned}$$

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

$$\begin{aligned} v_\pi'(s) &= E_\pi[G_t' \mid S_t = s] \\ &= E_\pi\left[G_t + \frac{c}{1-\gamma} \mid S_t = s\right] \\ &= E_\pi[G_t \mid S_t = s] + \frac{c}{1-\gamma} \end{aligned}$$

(as $\frac{c}{1-\gamma}$ is a constant and hence we can take it out of expectation)

$$\therefore \boxed{v_c = \frac{c}{1-\gamma}}$$

3.16

Case 1: Undiscounted:

Original Task: Agent gets −1 reward for every timestep that it stays in the maze.
It gets +1 reward when it gets out of the maze.
The −1 reward motivates agent to get out as quickly as possible

Modified task: c is added to all rewards. The agent gets c−1 for every step it stays in the maze & c+1 for exiting the maze.

If $c-1>0$ i.e. $c>1$, then the agent will be motivated to stay in the maze indefinitely and accumulate $(c-1)$ rewards at each time step. The rewards for exiting doesn't change with time. so agent has not motivation to exit early.
Hence the agent never exits.

Case 2: Discounted.

Original Task: Agents gets 0 reward for each time step it stays in the maze & gets reward +1 discounted by γ at each time step upon exiting the maze

Modified Task: Agent gets c rewards for each time step it stays in the maze & $(c+1)$ when it exites the maze discounted by γ.

In this case the agent has some motivation to exit the maze early because of γ.
However if c is high enough (example 1000), then it might be more benificial to stay.

Example Suppose the agent starts at position k which is at d distance away from exit. The agent can only move 1 unit distance every step.

If the agent exits as fast as possible, then

$$G_t = 1000 + \gamma 1000 + \gamma^2 1000 + \cdots + \gamma^{d-1} 1000 + \gamma^d 1001$$

$$= 1000(1 + \gamma + \gamma^2 + \cdots + \gamma^d) + \gamma^d$$

$$= \frac{1000\,(1-\gamma^{d+1})}{1-\gamma} + \gamma^d$$

If agent stays in the maze for infinite time and never exits, then

$$G_t = 1000 + \gamma 1000 + \gamma^2 1000 + \cdots$$

$$= \frac{1000}{1-\gamma}$$

Now if

$$\frac{1000\,(1-\gamma^{d+1})}{1-\gamma} + \gamma^d < \frac{1000}{1-\gamma}$$

$$\Rightarrow \frac{-\gamma^{d+1}}{1-\gamma} + \gamma^d < 0$$

$$\Rightarrow \gamma^d < \frac{\gamma^{d+1}}{1-\gamma}$$

$$\Rightarrow \cancel{\gamma^d} < \cancel{\gamma^d}\left(\frac{\gamma}{1-\gamma}\right)$$

$$\Rightarrow 1 < \frac{\gamma}{1-\gamma}$$

$$\Rightarrow 1-\gamma < \gamma$$

$$\Rightarrow 2\gamma > 1$$

$$\Rightarrow \gamma > 1/2$$

Then agent will be motivated to stay in the maze forever.

8-5.) $v_*(s) = \max_{a \in A(s)} q_*(s,a)$

8-8.) $p(s', r \mid s, a) = Pr[S_{t+1} = s', R_{t+1} = r \mid S_t = s, A_t = a]$

$r(r \mid s, a) = Pr[R_{t+1} = r \mid S_t = s, A_t = a]$

$$= \sum_{s' \in S} Pr[S_{t+1} = s', R_{t+1} = r \mid S_t = s, A_t = a]$$

$$= \sum_{s' \in S} p(s', r \mid s, a) \quad \text{---} \quad (1)$$

$v_\pi(r' \mid s, a)$ is the probability of getting reward $R_{t+2} = r'$ starting from $S_t = s$ and taking action $A_t = a$ under policy π.

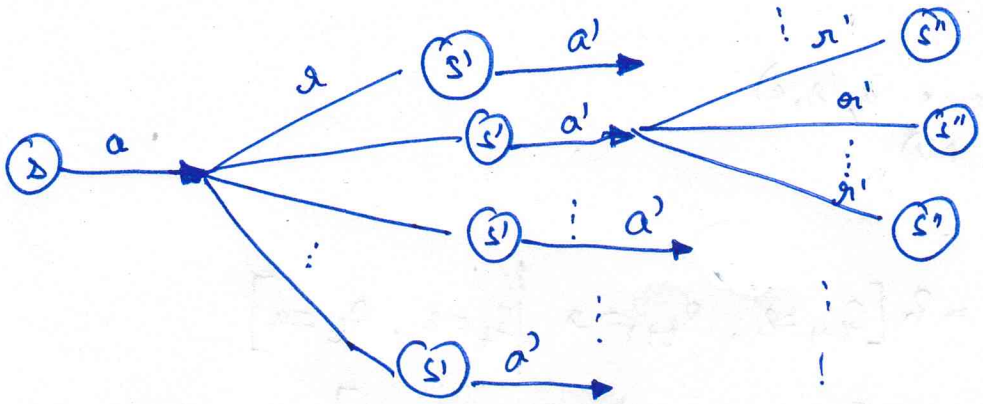
$$v_\pi(r' \mid s, a) = Pr[R_{t+2} = r' \mid S_t = s, A_t = a]$$

$$= \sum_{s' \in S} Pr[S_{t+1} = s' \mid S_t = s, A_t = a] \times \sum_{a' \in A(s')} Pr[A_{t+1} = a' \mid S_{t+1} = s', S_t = s, A_t = a]$$

$$\times Pr[R_{t+2} = r' \mid S_{t+1} = s', A_{t+1} = a', S_t = s, A_t = a] \quad \text{---} \quad (2)$$

Probability of getting reward at $t+2$, $R_{t+2} = r'$ is

probability of landing at s' x taking an action a' at s' x getting a reward r' due to a'.



$$Pr\left[S_{t+1} = s' \,\middle|\, S_t = s,\ A_t = a\right]$$

$$= \sum_{r \in R} Pr\left[S_{t+1} = s',\ R_{t+1} = r \,\middle|\, S_t = s,\ A_t = a\right]$$

$$= \sum_{r \in R} p(s', r \mid s, a) \qquad \text{——— (3)}$$

$$Pr\left[A_{t+1} = a' \,\middle|\, S_{t+1} = s',\ S_t = s,\ A_t = a\right] = Pr\left[A_{t+1} = a' \,\middle|\, S_{t+1} = s'\right]$$

$$= \pi(a' \mid s') \qquad \text{——— (4)}$$

(Because π only depends on current state)

$$Pr\left[R_{t+2} = r' \,\middle|\, S_{t+1} = s',\ A_{t+1} = a',\ S_t = s,\ A_t = a\right]$$

$$= \sum_{s'' \in S} Pr\left[S_{t+2} = s'',\ R_{t+2} = r' \,\middle|\, S_{t+1} = s',\ A_{t+1} = a',\ S_t = s,\ A_t = a\right]$$

$$= \sum_{s'' \in S} Pr\left[S_{t+2} = s'',\ R_{t+2} = r' \,\middle|\, S_{t+1} = s',\ A_{t+1} = a'\right] \quad \text{(Markov property)}$$

$$= \sum_{s'' \in S} p(s'', r' \mid s', a') \qquad \text{——— (5)}$$

$\therefore$ By ② ③, ④ and ⑤

$$\tilde{r}_\pi(r'|s,a) = \sum_{\substack{s' \in S \\ r \in R}} p(s',r|s,a) \sum_{\substack{a' \in A(s') \\ s'' \in S}} \pi(a'|s') \cdot p(s'',r'|s',a')$$

$\therefore$ $\tilde{r}_\pi$ depends on s, a and π.

Q-9) $E_\pi[R_{t+2} | S_t = s, A_t = a]$ (Expectation is dependent on policy π)

$$= \sum_{r' \in R} r' \cdot \tilde{r}_\pi(r|s,a)$$

$$= \sum_{r' \in R} r' \sum_{\substack{s' \in S \\ r \in R}} p(s',r|s,a) \sum_{\substack{a' \in A(s') \\ s'' \in S}} \pi(a'|s') \cdot p(s'',r'|s',a)$$

Q-10) $v_\pi(s) = E_\pi[G_t | S_t = s]$

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

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

$$= E_\pi[R_{t+1} | S_t = s] + \gamma E_\pi[E_\pi[G_{t+1} | S_{t+1} = s', S_t = s] | S_t = s]$$

G_{t+1} only depends on S_{t+1} & π not S_t

$\therefore$ $$= E_\pi[R_{t+1} | S_t = s] + \gamma E_\pi[E_\pi[G_{t+1} | S_{t+1} = s'] | S_t = s]$$

$$= E_\pi[R_{t+1} | S_t = s] + \gamma E_\pi[v_\pi(S_{t+1}) | S_t = s]$$

$$= E_\pi[R_{t+1} + \gamma v_\pi(S_{t+1}) | S_t = s]$$

$$= \sum_{\substack{r \in R \\ s' \in S \\ a \in A(s)}} p(r, s' | s, a) \cdot \pi(a|s) \cdot [r + \gamma v_\pi(s')]$$

8-11) $G_4 = 0$

Part1

$G_3 = R_4 + \gamma G_4$

$= -3 + (0.5)\,0$

$= -3$

$G_2 = R_3 + \gamma G_3$

$= 10 + (0.5)(-3)$

$= 10 - 1.5$

$= 8.5$

$G_1 = R_2 + \gamma G_2$

$= -1 + 0.5\,(8.5)$

$= -1 + 4.25$

$= 3.25$

$G_0 = R_1 + \gamma G_1$

$= 2 + (0.5)(3.25)$

$= 2 + 1.625$

$= 3.625$

Part2:

$R_t = c \quad \forall t$

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

$= c + \gamma c + \gamma^2 c + \gamma^3 c + \cdots$

$= c\,(1 + \gamma + \gamma^2 + \gamma^3 + \cdots)$

$= c \cdot \frac{1}{1-\gamma}$

$= \frac{c}{1-\gamma}$

Q-12)

$$\pi_*(s) = \underset{a \in A(s)}{\operatorname{argmax}} \left\{ p(s' \mid s,a) \left[r(s',a,s) + \gamma v_*(s') \right] \right\}$$

where:

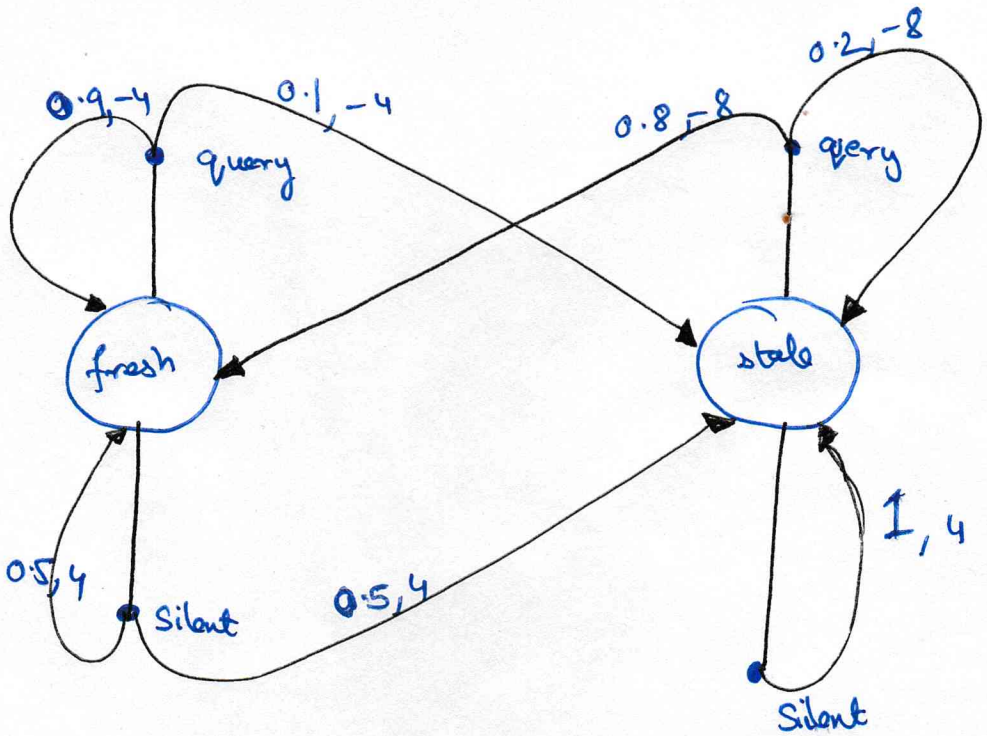
$p(s' \mid s,a)$ = probability of landing at s' if I take action a at s'

$r(s',a,s)$ = expected reward while going to s' from s via a.

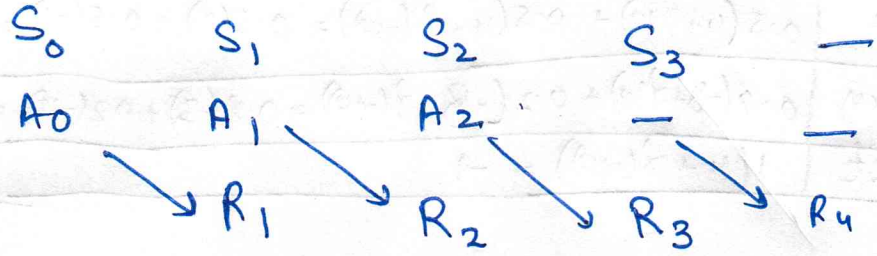
$$p(s' \mid s,a) = \sum_{r \in R} p(s', r \mid s, a)$$

$$r(s,a,s) = \sum_{r \in R} r \cdot \left[\frac{p(s', r \mid s,a)}{p(s' \mid s,a)} \right]$$

Q-13.) a)



b) Lets try to back Calculate. The sequence is

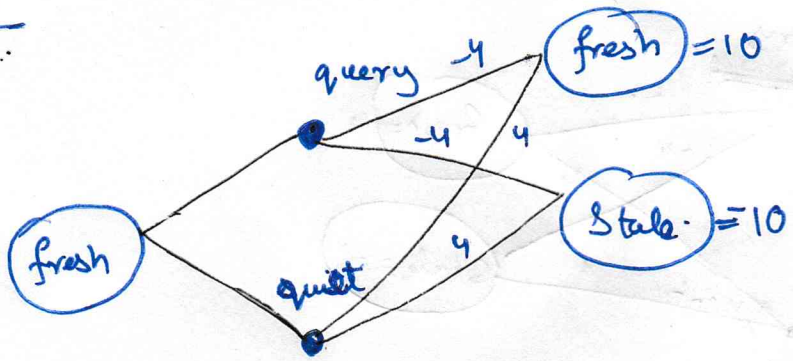


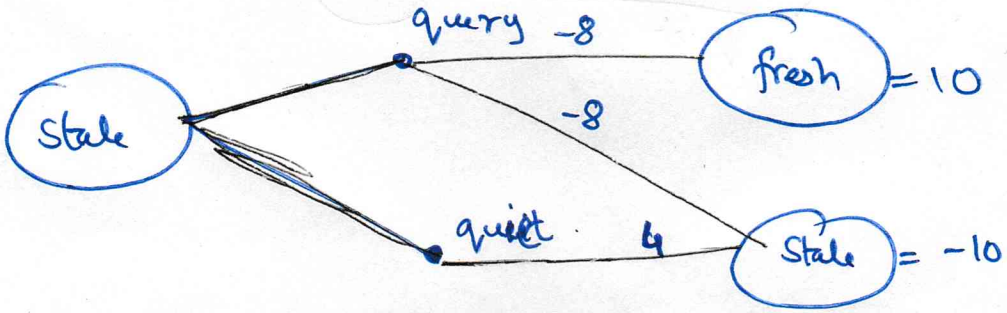
S_3:

If S_3 = fresh, $v_*(S_3 = \text{fresh}) = 10$

S_3 = Stale, $v_*(S_3 = \text{Stale}) = -10$

S_2:





∴

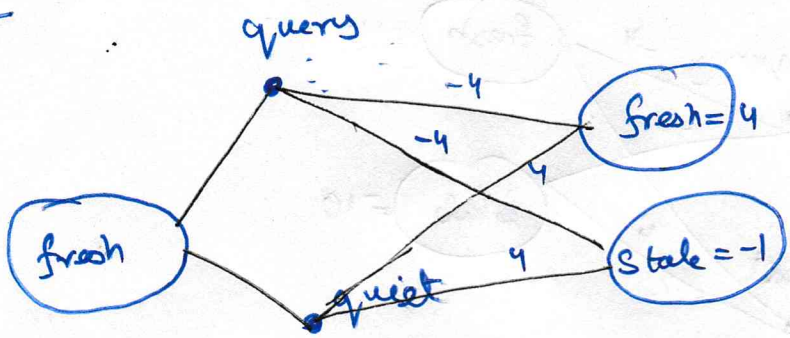
S_2	A_2	G_2
fresh	query	$0.9(-4+\gamma 10)+0.1(-4+\gamma(-10)) = 0.9(1)+0.1(-9) = 0$
fresh	quiet	$0.5(4+\gamma 10)+0.5(4+\gamma(-10)) = 0.5(9)+0.5(-1) = 4$
stale	query	$0.8(-8+\gamma 10)+0.2(-8+\gamma(-10)) = 0.8(-3)+0.2(-13) = -5$
stale	quiet	$1(4+\gamma(-10)) = -1$

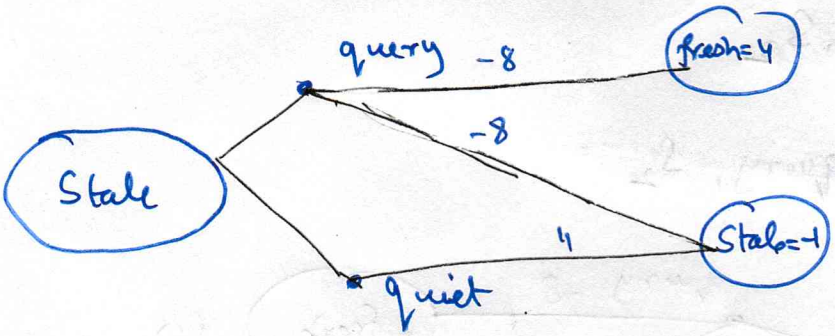
∴ At S_2 = fresh choose A_2 = quiet

S_2 = Stale choose A_2 = quiet

$V_*(S_2 = \text{fresh}) = 4$ $\quad V_*(S_2 = \text{stale}) = -1$

S_1 :





∴

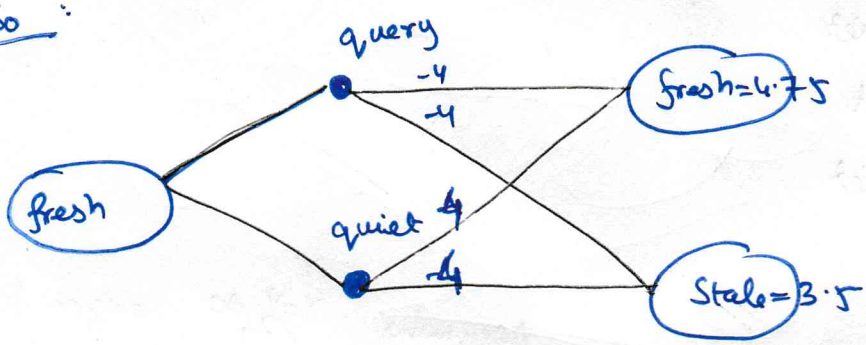
S_1	A_1	G_1
fresh	query	$0.9(-4+\gamma(4)) + 0.1(-4+\gamma(-1)) = 0.9(-2) + 0.1(-4.5) = -2.25$
fresh	quiet	$0.5(4+\gamma 4) + 0.5(4+\gamma(-1)) = 0.5(6) + 0.5(3.5) = 4.75$
stale	query	$0.8(-8+\gamma(4)) + 0.2(-8+\gamma(-1)) = 0.8(-6) + 0.2(-8.5) = -6.5$
stale	quiet	$1(4+\gamma(-1)) = 3.5$

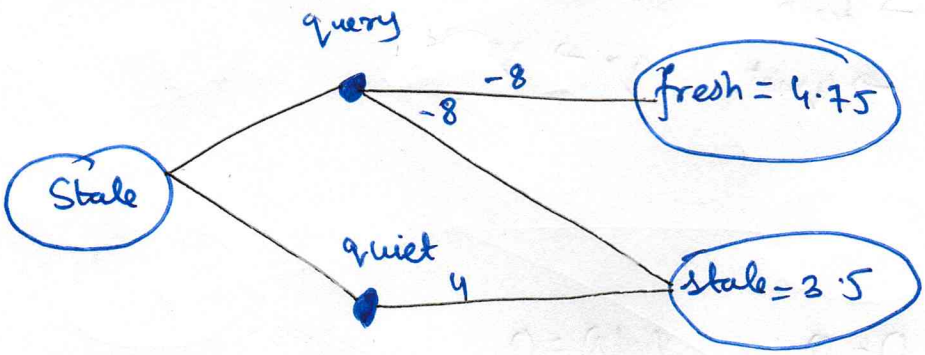
∴ At S_1 = fresh choose A_1 = quiet

S_1 = stale choose A_1 = quiet

$V_*(S_1 = \text{fresh}) = 4.75$ $\quad V_*(S_2 = \text{stale}) = 3.5$

So :





S_0	A_0	G_0
fresh	query	$0.9(-4+\gamma(4.75)) + 0.1(4+\gamma(3.5)) = -1.6875$
fresh	quiet	$0.5(4+\gamma(4.75)) + 0.5(4+\gamma(3.5)) = 6.0625$
stale	query	$0.8(-8+\gamma(4.75)) + 0.2(-8+\gamma(3.5)) = -5.75$
stale	quiet	$1(4+\gamma(3.5)) = 5.75$

$\therefore$ Take quiet action for A_0

$\therefore V_*(S_0 = \text{fresh}) = 6.0625$

$V_*(S_0 = \text{stale}) = 5.75$

Optimal policy:

A_0 : quiet

A_1 : quiet

A_2 : quiet

c.) Value iterations: Assuming $\gamma = \frac{1}{2}$, ∞ horizon problem.

Let $f \rightarrow$ fresh $\quad s \rightarrow$ state

$qt \rightarrow$ quiet $\quad qr \rightarrow$ query.

Initialization:

$V_1(f) = 0, \quad V_1(s) = 0$

$$V_{k+1}(s) = \max_a \sum_{s',r} p(s', r \mid s, a)\left(r + \gamma V_k(s')\right)$$

Iter ①:

for f:

$$V(f) = \max\{0.9(-4 + \tfrac{1}{2}(0)) + 0.1(-4 + \tfrac{1}{2}(0)),\ 0.5(4 + \tfrac{1}{2}(0)) + 0.5(4 + \tfrac{1}{2}(0))\}$$

$$= \max\{-4, 4\} = 4$$

$$V(s) = \max\{-8 + 0.8(0) + 0.2(0),\ 4 + 1(0)\}$$

$$= \max\{-8, 4\} = 4$$

Iter ②

$$V(f) = \max\{-4 + (0.9)(\tfrac{1}{2})(4) + (0.1)(\tfrac{1}{2})(4),\ 4 + (0.5)(\tfrac{1}{2})(4) + (0.5)(\tfrac{1}{2})(4)\}$$

$$= \max\{-4+2, 4+2\} = 6$$

$$V(s) = \max\{-8 + (\tfrac{1}{2})((0.8)(4) + (0.2)(4)),\ 4 + (\tfrac{1}{2})(1)(4)\}$$

$$= \max\{-8+2, 4+2\} = 6$$

Iter ③

$$V(f) = \max\{-4 + (\tfrac{1}{2})(0.9(6) + 0.1(6)),\ 4 + (\tfrac{1}{2})(0.5)6 + (0.5)6)\}$$

$$= \max\{-4+3, 4+3\} = 7$$

$$V(s) = \max\{-8 + (\tfrac{1}{2})(0.8(6) + 0.2(6)),\ 4 + (\tfrac{1}{2})(1)(6)\} = \{-8+3, 4+3\} = 7$$

Iter ④

$$v(f) = \max \{ -4 + 1/2 \{(0.9)(7) + (0.1)(7)\},$$
$$4 + 1/2 \{(0.5)(7) + (0.5)(7)\} \}$$
$$= \max\{-4 + 3.5, \; 4 + 3.5\} = 7.5$$

$$v(s) = \max \{ -8 + 1/2 (0.8(7) + 0.2(7)),$$
$$4 + 1/2 (1)(7) \}$$
$$= \max \{ -8 + 3.5, \; 4 + 3.5 \} = 7.5$$

Policy Iteration:

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

$$v_\pi(f) = \pi(q,r|f) [(0.9)(-4 + \gamma v_\pi(f)) + (0.1)(-4 + \gamma v_\pi(s))] +$$
$$\pi(q,t|f) [(0.5)(4 + \gamma v_\pi(f)) + (0.5)(4 + \gamma v_\pi(s))]$$

$$= \pi(q,r|f) (-4 + 0.45 v_\pi(f) + 0.05 v_\pi(s))$$

$$+ \pi(q,t|f) (4 + 0.25 v_\pi(f) + 0.25 v_\pi(s)) \quad \text{— (A)}$$

$$v_\pi(s) = \pi(q,r|s) [-8 + 0.4 v_\pi(f) + 0.1 v_\pi(s)]$$

$$+ \pi(q,t|s) [4 + 0.5 v_\pi(s)] \quad \text{— (B)}$$

Initialization

$P_{sl}(q,t \mid s) = 1$; $P_{sl}(q,t \mid f) = 1$; $V_{\pi_0}(s) = 0$

$P_{sl}(q,r \mid s) = 0$; $P_{sl}(q,r \mid f) = 0$; $V_{\pi_0}(f) = 0$

Iter ① of policy iteration

$$V_{\pi_0}(f) = 4 + 0.25 V_{\pi_0}(f) + 0.25 V_{\pi_0}(s) \quad \text{—— (i)}$$

$$V_{\pi_0}(s) = 4 + 0.5 V_{\pi_0}(s) \quad \text{—— (ii)}$$

Solving we get

$V_{\pi_0}(f) = 8$ $\quad$ $V_{\pi_0}(s) = 8$

Policy improvement :

$$\pi_1(f) = \text{argmax}\left\{-4 + 0.5\left(0.9 \times 8 + 0.1 \times 8\right),\ 4 + 0.5\left(0.5 \times 8 + 0.5 \times 8\right)\right\}$$

$$= \text{argmax}\{0, 8\}$$

$$\therefore \quad \pi_1(f) = q,t$$

$$\pi_1(s) = \text{argmax}\left\{-8 + 0.5\left(0.8 \times 8 + 0.2 \times 8\right),\ 4 + 0.5\left(1 \times 8\right)\right\}$$

$$= \text{argmax}\{4, 8\} = q,t$$

Policy stable Done!

Q-14) At policy improvement:

$$T V_{\pi_k} = T_{\pi_{k+1}} V_{\pi_k} \quad \text{—— (i)}$$

Also

$$T_{V_{\pi_k}} \geq V_{\pi_k} \quad \text{(As we take argmax)} \quad \text{—(ii)}$$

$\therefore$ By (i) & (ii)

$$T_{\pi_{k+1}} V_{\pi_k} \geq V_{\pi_k} \quad \text{—— (iii)}$$

Note if

$$f(s) \geq g(s) \ \forall s$$

$$\Rightarrow T_{\pi_{k+1}} f(s) \geq T_{\pi_{k+1}} g(s) \quad \text{—(iv)}$$

As

$$T_{\pi_{k+1}} f(s) = \sum_{a \in A(s)} \pi_{k+1}(a|s)\, p(s', r|s, a)(r + \gamma f(s'))$$

$$\geq \sum_{a \in A(s)} \pi_{k+1}(a|s)\, p(s', r|s, a)(r + \gamma g(s'))$$

for $\gamma \geq 0$

$$\geq T_{\pi_{k+1}} g(s)$$

By (iii) and (iv)

$$T_{\pi_{k+1}} T_{\pi_{k+1}} V_{\pi_k} \geq T_{\pi_{k+1}} V_{\pi_k} \quad \text{—— (v)}$$

By (v) and (iii)

$$T^2_{\pi_{k+1}} V_{\pi_k} \geq V_{\pi_k}$$

Repeating the same procedure:

$$\lim_{M \to \infty} T^M_{\pi_{k+1}} V_{\pi_k} \geq V_{\pi_k} \quad \text{——— (vi)}$$

But $T_{\pi_{k+1}}$ has a fix pt. $= V_{\pi_{k+1}}$

$$\therefore \quad V_{\pi_{k+1}} \geq V_{\pi_k} \quad \text{——— (vii)}$$

$\therefore$ $V_{\pi_{k+1}}$ either improves V_{π_k} i.e. π_{k+1} is an improvement over π_k or it doesn't change V_{π_k}.

In the case where

$$V_{\pi_{k+1}} = V_{\pi_k} \quad \text{——— (viii)}$$

By (i) & (viii)

$$T_{V_{\pi_{k+1}}} = T_{\pi_{k+1}} V_{\pi_{k+1}} = V_{\pi_{k+1}} \quad \text{——— (ix)}$$

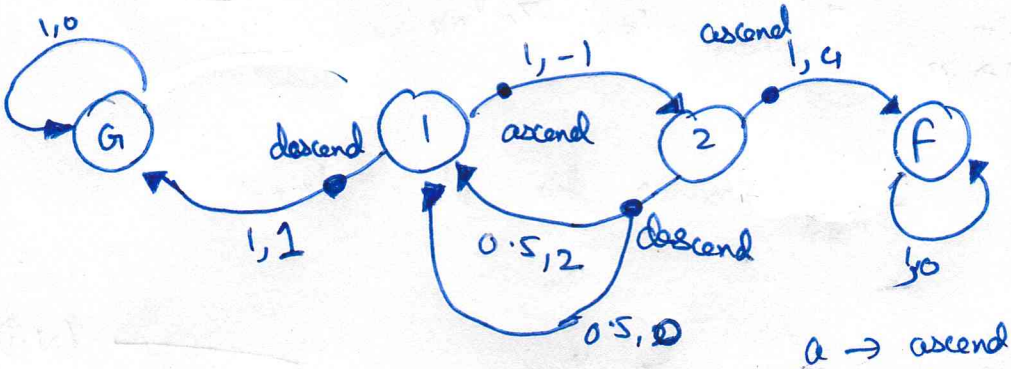
(As $V_{\pi_{k+1}}$ is fix pt. of $T_{\pi_{k+1}}$)

$\Rightarrow v_{\pi_{k+1}} = T v_{\pi_{k+1}} \Rightarrow v_{\pi_{k+1}}$ is the fix pt. of T

But T has fix pt. of v_*

$\therefore v_* = v_{\pi_{k+1}}$ $\therefore$ Optimal ■

Q-15.)



a → ascend
d → descend

Initialization:

$\pi_0(a|s)\ \forall\ s \in \{1,2\} = 0.5$

$\pi_0(d|s)\ \forall\ s \in \{1,2\} = 0.5$

$v_{\pi_0}(s)\ \forall\ s \in \{G,1,2,f\} = 0$

Note: $v_\pi(G) = 0$, $v_\pi(f) = 0$ } As terminal

$v_\pi(1) = \pi(d|1) \cdot 1\,[1 + 1(0)] + \pi(a|1) \cdot 1\,[-1 + 1\, v_\pi(2)]$

$= \pi(d|1) + \pi(a|1)\,(-1 + v_\pi(2))$ — (A)

$v_\pi(2) = \pi(d|1)\,[0.5\,(2 + 1 \cdot v_\pi(1)) + 0.5\,[0 + v_\pi(1)]] + \pi(a|1)\,[1]\,[4 + 1 \cdot (0)]$

$= \pi(d|1)\,(1 + v_\pi(1)) + 4\pi(a|1)$

— (B)

Iter ①

$$v_{\pi_0}(1) = 0.5 + 0.5(-1 + v_{\pi_0}(2)) = v_{\pi_0}(2)/2$$

$$v_{\pi_0}(2) = 0.5(1 + v_{\pi_0}(1)) + 4(0.5) = 2.5 + v_{\pi_0}(1)/2$$

$$\therefore \quad v_{\pi_0}(1) = 5/3 \quad , \quad v_{\pi_0}(2) = 10/3$$

Policy improvement

$$\pi_1(1) = \text{argmax}\{1[1+0], 1[-1 + 10/3]\}$$

$$= \text{argmax}\{1, 7/3\} = \text{ascend.}$$

$$\pi_1(2) = \text{argmax}\{0.5[0 + \tfrac{5}{3}] + 0.5[2 + \tfrac{5}{3}] + 4\}$$

$$= \text{argmax}\{\tfrac{8}{3}, 4\} = \text{ascend.}$$

Iter ②

$$v_{\pi_1}(1) = -1 + v_{\pi_1}(2) = 3$$

$$v_{\pi_1}(2) = 4$$

Policy improvement

$$\pi_2(1) = \text{argmax}\{1, -1 + 4\} = \text{ascend.}$$

$$\pi_2(2) = \text{argmax}\{1 + 3, 4\} = \text{ascend} \;\&\; \text{descend.}$$

Policy is stable !

$$\therefore \quad \pi_*(1) = \text{ascend}$$

$\pi_*(2) =$ ascend or descend. We can choose $Pr(a|2)$ & $P(d|2)$ arbitrarily

The optimal policy in this environment doesn't give any preference to ascending or descending from any state $i > 1$.

The reason is that $\gamma = 1$. I have no motivation to reach F faster.

If at state i, I traverse to i-1 and then come back to i and then go to F, I get the same reward if I had directly went from i to F.

∴ As long as I don't descend from 1, every policy is optimal. ⇒ $V_\pi(1)$ = descend is the only criteria

To promote π_1 to be optimal, I must set up V_{π_0} such that

$$\text{argmax}\{1, -1 + V_{\pi_1}(2)\} = \text{descend}$$

In other words,

$$V_{\pi_0}(2) - 1 > 1$$

$$\Rightarrow V_{\pi_0}(2) > 2$$

Lets Calculate $V_{\pi_0}(i)$

By observation we find that $\sum_{i=1}^{n} v_{\pi_0}(i) = \frac{5n}{2}$ ——— (1)

we also observe

$$v_{\pi_0}(i) = \Delta i \quad \text{where} \quad \Delta = \frac{5}{n+1} \qquad \text{——— (2)}$$

Lets verify this claim: (Under π_0 being equiprobable)

$$v_{\pi_0}(1) = 0.5 + 0.5\left[-1 + v_{\pi_0}(2)\right] = \frac{v_{\pi_0}(2)}{2} \qquad \text{——— (3)}$$

$$v_{\pi_0}(i) = 0.5\left[-1 + v_{\pi_0}(i+1)\right] + 0.5\left[0 + v_{\pi_0}(i-1)\right]$$

$$= \frac{v_{\pi_0}(i+1) + v_{\pi_0}(i-1)}{2} \qquad \text{——— (4)}$$

$$v_{\pi_0}(n) = 0.5\left[1 + v_{\pi_0}(n-1)\right] + 4(0.5)$$

$$= \frac{5 + v_{\pi_0}(n-1)}{2} \qquad \text{——— (5)}$$

Lets put (2) in (3), (4), (5)

(3): $\frac{v_{\pi_0}(2)}{2} = \frac{2\Delta}{2} = \Delta = v_{\pi_0}(1)$ Holds!

(4): $\frac{v_{\pi_0}(i+1) + v_{\pi_0}(i-1)}{2} = \frac{\Delta(i+1) + \Delta(i-1)}{2} = i\Delta$ holds!

(5): $\frac{5 + v_{\pi_0}(n-1)}{2} = \frac{5 + (n-1)\Delta}{2} = 2.5 + \frac{5}{(n+1)}\frac{(n-1)}{2}$

$$= \frac{5n+5+5n-5}{2(n+1)}$$

$$= \frac{10n}{2(n+1)} = \frac{5n}{n+1} = n\left(\frac{5}{n+1}\right) = n\Delta$$

Holds!

Now for

$$V_{\pi_0}(2) > 2$$

$$2\Delta > 2$$

$$\Delta > 1$$

$$\Rightarrow \frac{5}{n+1} > 1$$

$$\Rightarrow 5 > n+1$$

$$\Rightarrow n < 4$$

$\therefore$ for $n < 4$ π_0 will be optimal & we will only need 1 iteration!

For $n \geq 4$: If $n > 4$, $\Delta < 1$

$$\pi_1(1) = \text{argmax}\{1, -1 + V_{\pi_0}(2)\}$$

$$= \text{argmax}\{1, -1 + 2\Delta\}$$

$$= \text{descend.}$$

$$\pi_1(i) = \text{argmax}\{-1 + (i+1)\Delta, 1 + (i-1)\Delta\} \qquad 1 < i < n$$

$$= \text{argmax}\{i\Delta + (\Delta - 1), i\Delta - (\Delta - 1)\}$$

$$= \text{ascend.}$$

$$\pi_1(n) = \text{argmax}\{2n, 1 + \Delta(n-1)\}$$

$$= \text{argmax}\{2n, 1 + \frac{5n-5}{n+1}\}$$

$$= \text{argmax}\{2n, \frac{6n-4}{n+1}\}$$

$$2n > \frac{6n-4}{n+1}$$

$$\Rightarrow \quad 2n^2 + 2n > 6n - 4$$

$$\Rightarrow \quad 2n^2 > 4n - 4$$

$$\Rightarrow \quad n^2 > 2n - 1$$

$$\Rightarrow \quad n^2 - 2n + 1 > 0$$

$$n^2 - 2n + 1 = 0$$

$$\Rightarrow n = \frac{2 \pm \sqrt{4 - 4(1)(1)}}{2} = 1$$

$\therefore$ $n^2 - 2n + 1 >$ for $n > 1$, holds.

$\therefore$ $\bar{\pi}_1(n)$ = ascends.

$\therefore$ Only $\bar{\pi}_1(1)$ is correctly chosen

Lets Calculate $V_{\bar{\pi}_1}(i)$

$V_{\bar{\pi}_1}(1) = 1 \times 1 = 1$

$V_{\bar{\pi}_1}(i) = 1\,[-1 + V_{\bar{\pi}_1}(i+1)] = V_{\bar{\pi}_1}(i+1) - 1$ $\quad 1 < i < n$

$V_{\bar{\pi}_1}(n) = 1[2n-1] = 2n-1$

$\therefore$ $V_{\bar{\pi}_1}(i) = 2n - 1 - (n - i) = 2n - n - 1 + i$

$= n + i - 1$

Let Calculate π_2

$$\pi_2(i) = \text{argmax}\{1, -1 + v_{\pi_1}(2)\}$$

$$= \text{argmax}\{1, -1 + (n+2-1)\}$$

$$= \text{argmax}\{1, n\}$$

Since $n > 4$, we choose ascend $\therefore$ π_2 is optimal.

$\therefore$ 2 iterations.

For $n = 4$ $\Rightarrow \Delta = 1$

$$\pi_1(1) = \text{argmax}\{1, -1 + v_{\pi_0}(2)\}$$

$$= \text{argmax}\{1, 1\} = \text{ascend or descend.}$$

$$\pi_1(i) = \text{argmax}\{-1 + (i+1)\Delta, 1 + (i-1)\Delta\}$$

$$= \text{argmax}\{i, i\} = \text{ascend or descend.}$$

$$\pi_2(n) = \text{argmax}\{2n, 1 + \Delta(n-1)\}$$

$$= \text{argmax}\{2n, n+n-1\} = \text{ascend.}$$

In this case we may need more iterations!