Question 1:

180050097 SHASHANK ROY

Question 1:

1a) We create MDP (S, A, T, R, γ) for the problem:

- Since game always ends in H steps, MDP is episodic $\therefore$ let $\gamma = 1$
- We have two sets of actions attack (a) & defend (d). $\therefore A = \{a, d\}$
- We represent a state s as (w, l, t, h) where,
  - $w \rightarrow$ no. of times A scored
  - $l \rightarrow$ no. of times A conceded a goal
  - $t \rightarrow$ no. of times when nobody scored
  - $h \rightarrow$ no. of times played

Hence $h \in I$ where $h \in [0; H]$

Also for $s = (w, l, t, h)$

$$\boxed{w + l + t = h} \qquad w, l, t \geq 0$$

Meaning sum of all wins, ties and losses equals number of times played.

Hence in a nutshell ... (Set of whole numbers)

$$S = \{(w, l, t, h) \mid w, l, t, h \in W;\ h \in [0, H];\ w + l + t = h\}$$

States with $h = H$ represent terminal states.

- For T we have 2 actions to slect from state $s = (w, l, t, h)$ where $h < H$

i) Case 1 action = a. Hence there are 3 next states possible

a) $s' = (w+1, l, t, h+1)$ (A scored)
$T(s, a, s') = p_+^a$

b) $s' = (w, l+1, t, h+1)$ (A conceded)
$T(s, a, s') = p_-^a$

c) $s' = (w, l, t+1, h+1)$ (no one scored)
$T(s, a, s') = p_=^a$

ii) Case 2 action = d. As same as above

a) $s' = (w+1, l, t, h+1)$
$T(s, d, s') = p_+^d$

b) $s' = (w, l+1, t, h+1)$
$T(s, d, s') = p_-^d$

c) $s' = (w, l, t+1, h+1)$
$T(s, d, s') = p_=^d$

Note: none of the states are redundant and all states are reachable if all probabilities are non zero.

- For rewards R let $s = (w, l, t, h)$

If $h < H-1$, that is we have 2 or more games left.

$R(s, \text{action}, s') = 0$ $\forall$ action takeable $\forall$ s' we can transtion to from s.

If $h = H-1$, we have the final game left.
Let next state $s' = (w', l', t', H)$

$\therefore$ i) If $w' > l'$, $R(s, action, s') = 2$
ii) If $w' = l'$ $R(s, action, s') = 1$
iii) If $w' < l'$ $R(s, action, s') = 0$

where s' is set of states we can transition to with non-zero probability from s.

Hence we have the MDP (S, A, T, R, γ) defined from above steps.

1b) Procedure to compute optimal policy π for state $s = (w, l, t, h)$

- $h = H$. s' is terminal state hence no policy evaluation required. We set $V^{\pi}(s) = 0 \quad \forall s$ where $h = H$.

- $h = H-1$. We compute $Q(s, a)$ and $Q(s, d)$. Since next state can only be from states where $h = H$. $Q(s, \text{action})$ is computable

hence $Q(s, a) = p^a_{-}(r_1 + \gamma 0) + p^a_{+}(r_2 + \gamma 0) + p^a_{=}(r_3 + \gamma 0)$

$= \pi p^a_{-} r_1 + p^a_{+} r_2 + p^a_{=} r_3$

where $r_1, r_2, r_3 \in \{0, 1, 2\}$ depending upon the transitions to next states.

$\therefore \pi(s) = \begin{cases} a & \text{if } Q(s,a) \geq Q(s,d) \\ d & \text{if } Q(s,a) < Q(s,d) \end{cases}$

And correspondingly set $V^{\pi}(s) = Q(s, \pi(s))$

- Hence in a backpropagation manner we compute Q values for $s = (\omega, l, t, h)$ using $V^{\pi}(s')$ where $s' = (\omega', l', t', h+1)$.

This is possible since we can have transitions only from $h \rightarrow h+1$

~~Hence we update π as~~

Hence we update Q as

$$Q(s, \text{action}) = p_{+}^{a}(0 + \gamma V^{\pi}(s_1)) + p_{-}^{a} \gamma V^{\pi}(s_2) + p_{=}^{a} \gamma V^{\pi}(s_3)$$

where s_1, s_2, s_3 have $h' = h+1$

$\therefore$ update $\pi(s) = \begin{cases} a & Q(s,a) \geq Q(s,d) \\ d & Q(s,a) < Q(s,d) \end{cases}$

And we update $V^{\pi}(s) = Q(s, \pi(s))$

In this way we go from layer H to H-1 to H-2 upto layer 0.

In this way, policy π is evaluated for all states.

Number of operations :

In the process described we visit each and every state exactly once and do the following operations.

i) compute Q ii) Update π iii) Update V^{π}

Let Total number of bitwise operations from above be K.
Since Q requires 6 multiplications and 4 additions for all states, K is constant for every state.

∴ Total bitwise operations = no. of states × K

Looking at a layer h where states are of form (w, l, t, h).

No. of such states at time-step h is equal to no. of solution of equation

$w + l + t = h$ (where h is constant and $w, l, t \in [0, h]$.

No. of solutions = ${}^{h+3-1}C_{3-1} = {}^{h+2}C_2$

$$= \frac{(h+2)(h+1)}{2}$$

Hence no of states in layer h = $\frac{(h+1)(h+2)}{2}$

Total no. of states =

$$\sum_{h=0}^{H} \frac{(h+1)(h+2)}{2}$$

$$= \frac{1}{2} \sum_{h=0}^{H} h^2 + 3h + 2$$

Using results $\sum_{0}^{H} h^2 = \frac{H(H+1)(2H+1)}{6}$

and $\sum_{0}^{H} h = \frac{H(H+1)}{2}$

$\Rightarrow$ no. of states $= \frac{1}{2}\left(\frac{H(H+1)(2H+1)}{6} + \frac{3H(H+1)}{2} + 2(H+1)\right)$

$$= \frac{H^3}{6} + O(H^2)$$

$$= O(H^3)$$

$\Rightarrow$ no. of operations $= k\,O(H^3)$

$$= O(H^3)$$

$\therefore$ no. of bitwise operations is upper-bounded by $O(H^3)$ (polynomial in H)

1c) Since one step is left in game we have
$h = H - 1$
∴ current state s is of form $(w, l, t, H-1)$
∴ cases where 'a' is optimal action

• Case 1: $w = l - 1$
This means if A scores game can be tied to get score of 1. Other outcomes will give 0. Since $p_+^a > p_+^d$ A scores with higher probability by action 'a'.
We can also do this by comparing $Q(s,a)$ and $Q(s,d)$

$$Q(s,a) = p_+^a(1+\gamma 0) + p_=^a(0+\gamma 0) + p_-^a(0+\gamma 0)$$
$$= p_+^a$$

Similarly $Q(s,d) = p_+^d$

Since $p_+^a > p_+^d \Rightarrow Q(s,a) > Q(s,d)$

Hence 'a' is optimal.

• Case 2: $w = l$ AND $p_+^a - p_-^a > p_+^d - p_-^d$
This state represents when A and O have scored equal goals. Let's compare $Q(s,a)$ vs $Q(s,d)$.
As we know $p_+^a + p_-^a + p_=^a = 1$ & $p_+^d + p_-^d + p_=^d = 1$

$$Q(s,a) = p_+^a(2+0) + p_=^a(1+0) + p_-^a(0)$$
$$= 2p_+^a + p_=^a$$
$$= 2p_+^a + 1 - p_+^a - p_-^a = 1 + p_+^a - p_-^a$$

1c) Same for $Q(s,d) = 1 + p_+^d - p_-^d$

Hence if 'a' is optimal:

$Q(s,a) > Q(s,d) \Rightarrow 1 + p_+^a - p_-^a > 1 + p_+^d - p_-^d$

$\Rightarrow p_+^a - p_-^a > p_+^d - p_-^d$

With the constraints $p_+^a > p_+^d$ & $p_-^a > p_-^d$ it is not possible to determine if $p_+^a - p_-^a > p_+^d - p_-^d$ ∵ it is included in the condition.

This Case 1 and Case 2 described above are the only cases when 'a' is optimal.

Cases where 'a' is not optimal:

• Scenario 1: $w < l-1$ OR $w > l+1$

If $w < l-1$, A gets 0 points no matter the outcome. Hence A can choose one of 'a' or 'd' uniformly.

If $w > l+1$ A gets 2 points no matter the outcome. Hence A can choose 'a' or 'd' randomly.

Example:

For $w > l+1$ $Q(s,a) = 2(p_+^a + p_-^a + p_-^a)$; $Q(s,d) = 2(\Sigma p^d)$

$= 2$; $= 2$

• Scenario 2: $w = l + 1$

Over here A chooses 'd' since it does not want to give up lead.

$$Q(s,a) = p_+^a(2+0) + p_=^a(2+0) + p_-^a(1+0)$$

$$= 2p_+^a + 2p_=^a + p_-^a$$

$$= 2(1 - p_-^a) + p_-^a$$

$$= \underline{\underline{2 - p_-^a}}$$

Same for $Q(s,d) = 2 - p_-^d$

As we know $p_-^a > p_-^d$

$$\Rightarrow -p_-^a < p_-^d \Rightarrow 2 - p_-^a < 2 - p_-^d$$

$$\Rightarrow Q(s,a) < Q(s,d)$$

Hence 'd' is optimal

• Scenario 3: $w = l$ AND $p_+^a - p_-^a < p_+^d - p_-^d$

Since we have already proved for reverse case that 'a' is optimal. It is safe to say in this case 'd' is optimal

Hence we have proved Case 1 and Case 2 are when 'a' is optimal.

□

For Scenarios 1, 2 and 3 'a' is not optimal.

Question 2:

1800050097 SHASHANK ROY

Question 2:

2a.) O plays memoryless strategy o_1 with prob. q and O_2 with prob $1-q$.

Let A play action a_1. Then expected reward for this turn

$$E[r|a_1] = \frac{1}{3} \times P(O_1) + \frac{1}{2} P(O_2)$$

$$= \frac{1}{3} q + \frac{1}{2}(1-q)$$

$$= \frac{1}{2} - \frac{q}{6}$$

If A plays action a_2. Expected reward for this turn.

$$E[r|a_2] = \frac{2}{3} q + \frac{1}{4}(1-q)$$

$$= \frac{1}{4} + \frac{5q}{12}$$

Strategy:

Hence A plays the action which gives max expected reward this turn.

A plays a_1 if $\frac{1}{2} - \frac{q}{6} > \frac{1}{4} + \frac{5q}{12}$

$\Rightarrow q < 3/7$

Else A plays a_2 if $q \geq 3/7$

∴ max expected reward for T rounds $= R_T^*$

where $R_T^* = \begin{cases} \left(\frac{1}{2} - \frac{q}{6}\right)^T & q < 3/7 \\ \text{OR} & \\ \left(\frac{1}{4} + \frac{5q}{12}\right)^T & q \geq 3/7 \end{cases}$

2b) The optimization problem can be modelled as a multi-bandit problem.

A has to pick an arm a_1 or a_2.

The bandit instance for arm a_1 gives reward 1 with probability $\frac{1}{2} - \frac{q}{6}$

Similarly for arm a_2 reward is given with probability $\frac{1}{4} + \frac{5q}{12}$

Hence A can follow π_A strategy which are standard algorithms such as ϵ-greedy, KL-UCB or thompson sampling.

If we want $\lim_{T \to \infty} \frac{R_T'}{R_T^*} = 1$

we need regret which is sublinear.

2b) ∴ A follows strategy $\pi_A = \epsilon G1$

where $\epsilon_t = \frac{1}{\sqrt{T}}$ ⇒ $\epsilon_T T = \sqrt{T}$ pulls. Then exploit.

$\epsilon G1$ strategy with $\epsilon_T = \frac{1}{\sqrt{T}}$

gives sub-linear regret according to "GLIE" conditions:

- Infinite exploration satisfied since each arm pulled at least $\frac{1}{2}\sqrt{T}$ times
- Greed in limit satisfied since

$$\frac{E[\text{exploit}(T)]}{T} \geq \frac{T - \sqrt{T}}{T}$$

exploit(T) denotes no. of pulls that are greedy & for sublinear regret we need

$$\lim_{T\to\infty} \frac{E[\text{exploit}(T)]}{T} = 1$$ which is true for above.

Note:

$\epsilon G1$ is strategy where

if $t \leq \epsilon T$ sample arm uniformly

$t = \lfloor \epsilon T \rfloor$ sample a best according to highest empirical mean

$t > \epsilon T$ sample a best

2b) Hence A can follow described strategy to get average expected reward R_T.

such that $\lim_{T \to \infty} \frac{R_T}{R_T^*} = 1$

2c) If A plays memoryless strategy

$P(a_1) = p \qquad P(a_2) = 1-p.$

∴ Expected reward of 1 turn

$$E[r] = p(a_1)p(o_1)\frac{1}{3} + p(a_1)p(o_2)\frac{1}{2} + p(a_2)p(o_1)\frac{2}{3} + p(a_2)p(o_2)\frac{1}{4}$$

$$= \frac{pq}{3} + \frac{p(1-q)}{2} + \frac{(1-p)q2}{3} + \frac{(1-p)(1-q)}{4}$$

$$= \frac{-7}{12}pq + \frac{5q}{12} + \frac{p}{4} + \frac{1}{4}$$

∴ Expected reward of T rounds $= T \times E[r]$

$$= \left(\frac{-7}{12}pq + \frac{5q}{12} + \frac{p}{4} + \frac{1}{4}\right)T$$

2d) $\max_{p} \min_{q} f(p, q, t)$ represents the maximum aggregate (expected) reward that A can get against O, given O knows the strategy (the param. 'p') used by A.

It is the largest reward which A can guarantee itself to be awarded against O for 1 turn.

It also means the largest probability with which A can guarantee itself to win given O knows A's strategy for a turn.

To calculate $\max_{p} \min_{q} f(p, q, T)$

we set $T = 1$, since $f(p, q, T) = q(p, q) \times T$

We denote $f(p, q, T)$ as f.

∴ We solve this using Linear Programming.

Variables :

- f - Which we have to maximize for A.
- π_{a_1} - Represents probability A takes action a_1 ('p' in theory)
- π_{a_2} - Represents probability A takes action a_2 ('1 - p')

Constraints :

- $\pi_{a_1}, \pi_{a_2} \geq 0 \quad \rightarrow (1)$
- $\pi_{a_1} + \pi_{a_2} = 1 \quad \rightarrow (2)$

Now O wants minimize f against A's strategy in the best possible way.

$\therefore f = \min_{Q} (\pi_{a_1} \rho(a_1, O) + \pi_{a_2} \rho(a_2, O))$

where $O \in \{O_1, O_2\}$ and $\rho(a, O)$ refers to expected reward when A play 'a' and O plays 'o'

Since f has to be less than the values of O

$\therefore$ The following constraints are added :

- for $O = O1$, $\rho(a_1, O_1) = 1/3$ & $\rho(a_2, O_1) = 2/3$

$\therefore$ constraint added is :

$$\pi_{a_1} \times \frac{1}{3} + \pi_{a_2} \times \frac{2}{3} \geq f \quad \rightarrow (3)$$

- for $O = O2$; $\rho(a_1, O_2) = 1/2$ & $\rho(a_2, O_2) = 1/4$

$\therefore$ constraint is :

$$\pi_{a_1} \times \frac{1}{2} + \pi_{a_2} \times \frac{1}{4} \geq f \rightarrow (4)$$

On solving linear problem with constraints ①, ②, ③ and ④ and maximizing 'f'

we get $f = 3/7$ and $\pi_{a_1} = p = \frac{5}{7}$

Hence we get $\max_{p} \min_{q} f(p, q, T) = \frac{3}{7} \times T$

This constitutes Nash Equilibrium since A and O play best strategy against each other.

For further intuition let A play a_1 with p & O play O_1 with q.

Then $f = p(O_1)\left[p(a_1)\times\frac{1}{3} + p(a_2)\frac{2}{3}\right]$

$+ \; p(O_2)\left[p(a_1)\times\frac{1}{2} + p(a_2)\times\frac{1}{4}\right]$

$= q\left[\frac{p}{3} + (1-p)\frac{2}{3}\right] + (1-q)\left[\frac{p}{2} + \frac{(1-p)}{4}\right]$

$= q\left[\frac{2-p}{3}\right] + (1-q)\left[\frac{p+1}{4}\right]$

Since O minimizes f, O play O_1, that is $q = 1$ when:

$$\frac{2-p}{3} < \frac{p+1}{4} \Rightarrow p > \frac{5}{7}$$

And hence O plays O_2 when $p < 5/7$

However when $p = 5/7$

$$f = q\left[\frac{2-5/7}{3}\right] + (1-q)\left[\frac{5/7+1}{4}\right]$$

$$= q \times \frac{9}{21} + (1-q) \times \frac{12}{28} = \frac{3q}{7} + (1-q)\frac{3}{7}$$

$$= \frac{3}{7}(1+q-q) = \frac{3}{7}$$

Hence f becomes independent of q.

i.e. no matter what strategy O follows (any value of q). A can guarantee average reward of $3/7$ for a turn.

Hence $\max_p \min_q f = \frac{3}{7}$. Which means

A can guarantee a win with probability $3/7$ by playing a $p = \frac{5}{7}$ memoryless strategy.

Question 3:

180050097 SHASHANK ROY

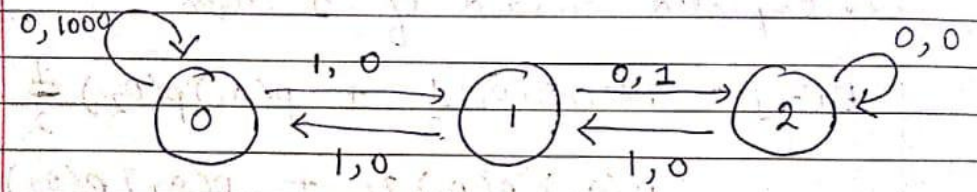
Question 3:

3a) G_1 is false.

Proof :

a, r : represents action, reward.
No need to represent transition probs. since M' has only deterministic MDPs.

Hence figure of $m \in M'$:



Let $\pi = [1, 0, 0]$ $(\pi(1) = \pi[1])$

$\therefore V^\pi(2) = 1(0 + \gamma V^\pi(2))$

$\Rightarrow (1-\gamma) V^\pi(2) = 0 \Rightarrow V^\pi(2) = 0 \quad (\gamma < 1)$

$V^\pi(1) = 1(1 + \gamma V^\pi(2)) = 1$

$V^\pi(0) = 1(0 + \gamma V^\pi(1)) = \gamma$

$\therefore V^\pi = [\gamma, 1, 0]$

Let's compute Q values

$Q(0, 0) = 1(100 + \gamma V^\pi(0)) = 100 + \gamma^2$
(state, action)

Since $Q(0,0) > V^\pi(0)$
this is improvable

$Q(1,1) = 1(0 + \gamma V^{\pi}(0))$

$= \gamma^2 \quad (\gamma < 1)$

Since $Q(1,1) < 1 = V^{\pi}(1)$ it is not improvable

Finally $Q(2,1) = 1(0 + \gamma V^{\pi}(1))$

$= \gamma$

Since $Q(2,1) > V^{\pi}(2)$ it is improvable

Hence improvable policies are

$\pi_1' = [0, 0, 0]$ or $\pi_2' = [1, 0, 1]$

or $\pi_3' = [0, 0, 1]$

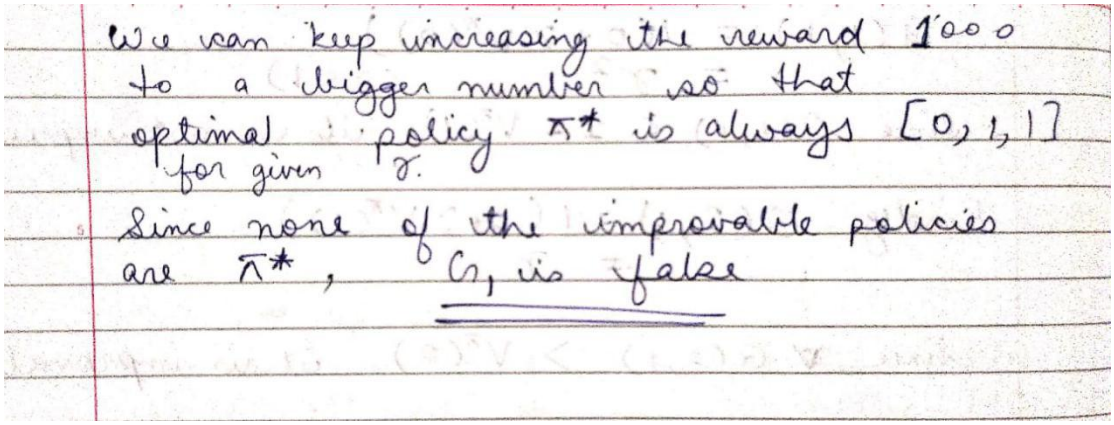
However as we can see we need $\pi'[1] = 1$ for optimal policy.

Optimal policy $\pi^* = [0, 1, 1]$

$V^{\pi^*}(0) = \frac{1000}{1-\gamma} \quad V^{\pi^*}(1) = \gamma \times \frac{1000}{1-\gamma}$

$V^{\pi^*}(2) = \frac{\gamma^2 1000}{1-\gamma}$

If we choose π_1' or π_3' (way we improve 0 always)
then also $V^{\pi'}(1)$ will not be optimal since factor of 1000 will not come in it. It will remain improvable.


We can keep increasing the reward 1000
to a bigger number so that
optimal policy π* is always [0, 1, 1]
for given γ.
Since none of the improvable policies
are π*, (c), is false

Insert q 3b)

Question 4:

Question 4:

In the reinforce algorithm updates are as follows:

$$w_{new} = w_{new} + \alpha (G_t \nabla_w \ln \pi_w(s_t, a_t)$$

↳ gradient

Over here, $G_t = Q$

$$\therefore g_i = \frac{\partial (\ln \pi_w(s^t, a^t))}{\partial w_i}$$

For derivations s^t, a^t is represented as s, a

4a) Given $\pi_w(s,a) = \frac{e^{w \cdot \phi(T(s,a))}}{\sum_{b \in A} e^{w \cdot \phi(T(s,b))}}$

Representing $T(s,x)$ as T_x

$$\pi_w(s,a) = \frac{e^{w \cdot \phi T_a}}{\sum_{b \in A} e^{w \cdot \phi(T_b)}}$$

$$\ln(\pi_w(s,a)) = \ln(e^{w \cdot \phi(T_a)}) - \ln\left(\sum_{b \in A} e^{w \cdot \phi(T_b)}\right)$$

$$= w \cdot \phi(T_a) - \ln\left(\sum_{b \in A} e^{w \cdot \phi(T_b)}\right)$$

$$\omega \cdot \phi(x) = \sum_{i=1}^{d} \omega_i \phi_i(x)$$

$$\Rightarrow \frac{\partial(\omega \cdot \phi(x))}{\partial \omega_i} = \phi_i(x)$$

$$\therefore \frac{\partial \ln(\pi_\omega(s,a))}{\partial \omega_i} = \phi_i(T_a) - \frac{\partial \ln\left(\sum_b e^{\omega \cdot \phi(T_b)}\right)}{\partial \omega_i}$$

$$\frac{\partial \ln\left(\sum_b e^{\omega \cdot \phi(T_b)}\right)}{\partial \omega_i} = \frac{1}{\sum_{b \in A} e^{\omega \cdot \phi(T_b)}} \frac{\partial\left(\sum_b e^{\omega \cdot \phi(T_b)}\right)}{\partial \omega_i}$$

$$= \frac{1}{\sum e^{\omega \cdot \phi(T_b)}} \sum_{b \in A} \left[e^{\omega \cdot \phi(T_b)} \times \phi_i(T_b) \right]$$

$$\Rightarrow \text{LHS} = \sum_{b \in A} \frac{e^{\omega \cdot \phi(T_b)}}{\sum e^{\omega \cdot \phi(T_b)}} \times \phi_i(T_b)$$

$$= \sum_{b \in A} \pi(s,b) \times \phi_i(T_b)$$

$$\Rightarrow g_i = \frac{\partial \ln \pi_\omega(s,a)}{\partial \omega_i} = \phi_i(T(s,a)) - \sum_{b \in A} \pi(s,b)\, \phi_i(T(s,b))$$

Note:

$\phi_i(x)$ denotes $[\phi(x)]_i$, the i^{th} element of vector $\phi(x)$

Pseudocode:

```
temp_π_s = [] ,  temp1 = 0

for a in A {
    dot = 0
    for i = 1, 2 . . . d {
        dot += w_i × φ_i(T(s^t, a)) }
    temp_π_s.append(e^dot)
    temp1 += e^dot
}

temp_2 = 0

for a in A {
    dot1 = temp_π_s[a] × φ_i(T(s^t, a))

    temp2 += dot1 / temp1
}

g_i = φ_i(T(s^t, a^t)) − temp2
```

For help:

$temp\pi_s[a] = e^{w \cdot \phi(T(s,a))}$ $dot = w \cdot \phi(T(s,a))$

$temp1 = \sum_{a \in A} e^{w \cdot \phi(T(s^t, a))}$

$temp2 = \sum_{b \in A} \pi(s,b)\, \phi_i(T(s,b))$

4 b) We first compute $\pi_w(s,a)$.

For derivation purposes $\sigma(w \cdot \phi(T(s,a)))$ is denoted as σ_a.

Say action 'a' is returned in one step:

$$Pr(n=1) = \frac{1}{|A|}\sigma_a$$

Action 'a' is returned in 2 steps. For this any action sampled in 1st step should not be returned.

$$\therefore Pr(n=2) = \left[\frac{1}{|A|}\left(\sum_{b \in A}(1-\sigma_b)\right)\right]\frac{1}{|A|}\sigma_a$$

$$= \frac{1}{|A|^2}(|A| - \sum_{b \in A}\sigma_b)\sigma_a$$

For n steps first $n-1$ actions should not be returned and final action 'a' is returned.

If we look at the multinomial:

$$\frac{1}{|A|^{n-1}}\left[(1-\sigma_{b_1})x_1 + (1-\sigma_{b_2})x_2 + (1-\sigma_{b_3})x_3 \ldots\right]^{n-1}$$

Let us look at coefficient of $x_1^{k_1} x_2^{k_2} \ldots x_{|A|}^{k_{|A|}}$

it is $\frac{1}{|A|^{n-1}} \prod_{b \in A}(1-\sigma_b)^{k_b} \times \frac{(n-1)!}{k_1!\, k_2! \ldots k_{|A|}!}$

This represents probability of

rejecting b_1 k_1 times, b_2 k_2 times ... $b_{|A|}$ $k_{|A|}$ times. where $\sum k_i = n-1$

Hence all of the terms represent the various cases where for $n-1$ steps actions were not returned.

Hence putting $x_1 = x_2 = x_3 \cdots = x_{|A|} = 1$

we have $\frac{1}{|A|^{n-1}} \left[\sum_{b \in A} (1 - \sigma_b) \right]^{n-1}$

$$= \frac{1}{|A|^{n-1}} \left[|A| - \sum_{b \in A} \sigma_b \right]^{n-1}$$

$$\therefore P_r(n' = n) = \frac{1}{|A|^{n-1}} (|A| - \Sigma \sigma_b)^{n-1} \frac{1}{|A|} \sigma_a$$

$$\therefore \pi_w(s,a) = \sum_{n=1}^{\infty} \frac{1}{|A|^n} (|A| - \sum_{b \in A} \sigma_b)^{n-1} \sigma_a$$

This is ∞ G.P. with ratio $= \frac{|A| - \Sigma \sigma_b}{|A|}$

$$\therefore \pi_w(s,a) = \frac{\frac{1}{|A|} \sigma_a}{1 - \left(\frac{|A| - \Sigma \sigma_b}{|A|} \right)} = \frac{\sigma_a}{\sum_{b \in A} \sigma_b}$$

Hence we have $\pi_w(s,a) = \frac{\sigma(w \cdot \phi(T(s,a)))}{\sum_{b \in A} \sigma(w \cdot \phi(T(s,b)))}$

Also as we can see $\Sigma \pi_w(s,a) = \frac{\Sigma \sigma_b}{\Sigma \sigma_b} = 1$

Hence this is a correct probability distribution

For calculating gradient we have:

$$\ln(\pi_\omega(s,a)) = \ln\left(\frac{\sigma_a}{\Sigma \sigma_b}\right)$$

$$= \ln(\sigma_a) - \ln(\Sigma \sigma_b)$$

Representing $T(s,x)$ as T_x

$$\frac{\partial(\omega \cdot \phi(T_a))}{\partial \omega_i} = \frac{\partial(\Sigma \omega_i \phi_i(T_a))}{\partial \omega_i} = \phi_i(T_a) \quad -(1)$$

$$\sigma_a = \frac{1}{1 + e^{-\omega \cdot \phi(T_a)}}$$

$$\Rightarrow \frac{\partial \sigma_a}{\partial \omega_i} = \frac{-1 \times e^{-\omega \cdot \phi(T_a)}}{(1 + e^{-\omega \cdot \phi(T_a)})^2} \times -\phi_i(T_a) \quad (\text{Using } (1))$$

$$= +\sigma_a^2 e^{-\omega \cdot \phi(T_a)} \phi_i(T_a) \quad -(2)$$

$$\therefore \frac{\partial \ln(\sigma_a)}{\partial \omega_i} = \frac{1}{\sigma_a} \frac{\partial \sigma_a}{\partial \omega_i}$$

$$= \frac{1}{\sigma_a} \sigma_a^2 e^{-\omega \cdot \phi} \phi_i(T_a) \quad \text{Using } (2)$$

$$= \sigma_a e^{-\omega \cdot \phi(T_a)} \phi_i(T_a) \quad -(3)$$

Now $$\frac{\partial \Sigma \sigma_b}{\partial \omega_i} = \sum_{b \in A} \frac{\partial \sigma_b}{\partial \omega_i}$$

$$= \sum_{b \in A} \sigma_b^2 e^{-\omega \cdot \phi(T_b)} \phi_i(T_b) \quad (\text{From } (2))$$

$$- (4)$$

$$\therefore \frac{\partial \ln(\Sigma \sigma_b)}{\partial \omega_i} = \frac{1}{(\Sigma \sigma_b)} \frac{\partial \Sigma \sigma_b}{\partial \omega_i}$$

$$\Rightarrow \frac{\partial \ln(\Sigma \sigma_b)}{\partial w_i} = \frac{1}{(\Sigma_{b_i \in A} \sigma_b)} \left[\sum_{b \in A} \sigma_b^2 e^{-w \cdot \phi(T_b)} \phi_i(T_b) \right] \quad -(5)$$

Hence

$$g_i = \frac{\partial \ln(\bar{\pi}_w(s,a))}{\partial w_i} = \frac{\partial \ln(\sigma_a)}{\partial w_i} - \frac{\partial \ln(\Sigma \sigma_b)}{\partial w_i}$$

From (3) and (5)

$$g_i = \sigma_a e^{-w \cdot \phi(T_a)} \phi_i(T_a) - \frac{1}{(\Sigma_{b \in A} \sigma_b)} \left[\sum_{b \in A} \sigma_b^2 e^{-w \cdot \phi(T_b)} \phi_i(T_b) \right]$$

For pseudocode we first define functions (which can be used can confirmed by mail from Sir):

```
def dot_prod(x, y):
    sum = 0
    for i = 1, 2, ... d:
        sum = sum + x_i * y_i
    return sum

def sigmoid(z):
    res = 1 / (1 + exp(-z))
    return res
```

Coming to the code where g_i is set:

```
dot1 = dot_prod(w, ϕ(T(s^t, a^t)))
temp1 = exp(-dot1)
s1 = sigmoid(dot1)
t1 = s1 · temp1 · ϕ_i(T(s^t, a^t))

denom = 0
numer = 0

for b in A:
    dot2 = dot_prod(w, ϕ(T(s^t, b)))
    temp2 = exp(-dot2)
    s2 = sigmoid(dot2)
    numer = numer + s2 · s2 · temp2 · ϕ_i(T(s^t, b))
    denom = denom + s2

t2 = numer / denom

g_i = t1 - t2
```