CS565 - INTRO TO ARTIFICIAL INTELLIGENCE

NAME : MADHAN THANGAVEL
B ID : B00814916
ODIN : mthanga1

HOMEWORK - 3

① Q-LEARNING

Given :

learning rate = 0.1 [α]
discount factor = 0.9 [γ]

According to Q learning, we know that,

$$Q_{new}(s,a) \leftarrow (1-\alpha)\phi(s,a)_{old} + \alpha[R(s,a,s') + \gamma \max_{a'} Q(s',a')]$$

We need to find the Q-value for each of the actions as mentioned in the question.

For Action -1 :

$(S_2, \text{north}, S_3, r = -0.1)$

Here $\alpha = 0.1$, $\gamma = 0.9$, $R = -0.1$

∴ using the above formulae

$$Q_{new}(s,a) \leftarrow (1-0.1)*0 + 0.1[-0.1 + 0.9(0.5)]$$

$$Q_{new}(s,a) \leftarrow 0.1[0.35]$$

$$Q_{new}(s,a) \leftarrow 0.035$$

For Action -2 :

$(S_3, \text{east}, S_4, r = -0.1)$

$\gamma = 0.9$, $\alpha = 0.1$, $R = -0.1$

∴ using the above formulae,

$$Q_{new}(s,a) \leftarrow (1-0.1)*0 + 0.1[-0.1 + 0.9*0.9]$$

$Q_{new}(s,a) \leftarrow 0.071$ //

For Action - 3:

$(s_4, north, s_4, r=1)$

$\alpha = 0.1, \gamma = 0.9, R = 1$.

∴ using the formulae.

$Q_{new}(s,a) \leftarrow (1-0.1) * 0.9 + (0.1)[1 + 0.9 \times 0.9]$

$Q_{new}(s,a) \leftarrow 0.81 + 0.181$

$Q_{new}(s,a) \leftarrow 0.991$ //

updated Q-values are:

* $Q(s_2, north, s_3) = 0.035$
* $Q(s_3, east, s_4) = 0.071$.
* $Q(s_4, north\ s_4) = 0.991$.

2. YES/ NO.

a) NO, but it assumes there exists an MVP to compute policy.

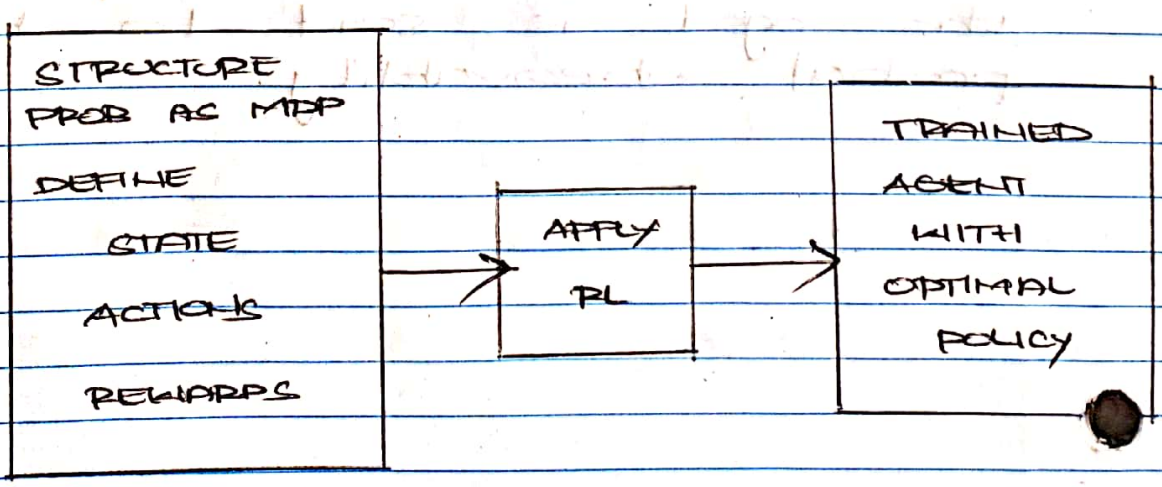
b) YES, but at sometime we dont know transition, reward on both of them. [addition to previous answer a)].

c) NO, as MDP assumes the agent directly observes the current environmental state: In this case the problem is said to have fully observable. If the agent only has access to subset of states, and the states corrupted to noise then the agent is said to have poabial observability.

d) YES, it assumes that the reward function $R(s, a, s')$ is available as an MDP. (From RL definition).

e) NO, immediate reward is not available as the transition and reward are not known during run time.

3.a) When we have MDP problem and RL algorithm, we will apply the RL algorithm to build an agent model and train it to find the optimal policy. Finding the optimal policy essentially solves the problem.



like using VI/PI on approx MDP and PE on approx MDP. Using Model-Free we use Q-learning and value learning to evaluate the policy.

b) When an RL problem and MDP algorithm are given then we know the MDP then our goal is to compute V^*, Q^* and π^* using value/policy iteration.

Similarly evaluating the fixed policy π using policy evaluation.

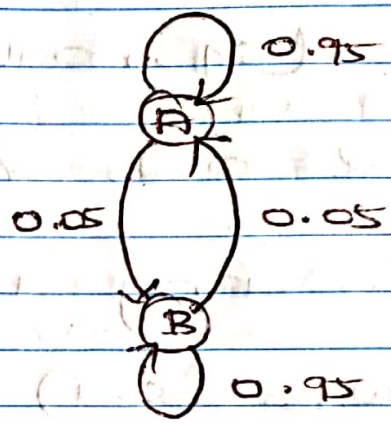
In other words, to find the optimal policy for the agent, once it has the optimal policy it simply uses that policy to pick actions from any state.

Thus we will apply MDP algorithm to build an agent model and train it to find the optimal policy. Finding the optimal policy essentially solves the RL problem.

④ HMM MODEL :

a) $P(S) = ?$

S	O	P(O/S)
A	0	0.8
A	1	0.2
B	0	0.1
B	1	0.9



$$P = \begin{array}{c} \\ A \\ B \end{array} \begin{array}{c} \begin{array}{cc} A & B \end{array} \\ \begin{bmatrix} 0.95 & 0.05 \\ 0.05 & 0.95 \end{bmatrix} \end{array}$$

stationary distribution is the distribution we end up with P_∞ of the chain.

It satisfies

$$P_\infty(x) = P_{\infty+1}(x) = \sum P(x|y) P_\infty x$$

$$P_\infty(A) = P(A|0) P_\infty(A) + P(A|1) P_\infty(B)$$

$$P_\infty(B) = P(B|0) P_\infty(A) + P(B|1) P_\infty(B)$$

$$P(S) = P_\infty(A) + P_\infty(B) = 1.$$

$P_\infty(A) = 0.8\,P_\infty(A) + 0.1\,P_\infty(B)$

$P_\infty(A) =$

$2\,P_\infty(A) = 1\,P_\infty(B)$

$P_\infty(A) = 1/2\,P_\infty(B) \rightarrow (1)$

$P_\infty(B) = 0.2\,P_\infty(A) + 0.9\,P_\infty(B)$

$P_\infty(B) = 2\,P_\infty(A) \rightarrow (2)$

Also,

$P_\infty(A) + P_\infty(B) = 1.$

$P_\infty(A) + 2P_\infty(A) = 1.$

$3P_\infty(A) = 1$

$P_\infty(A) = 1/3$

$P_\infty(B) = 2/3$

(b) $P(O_1=0, O_2=0, O_3=1)$

So → stationary distribution.

$P(O_1=0, O_2=0, O_3=1)$ can be computed using the forward algorithm. The past is independent of the future, given the present.

We can compute the value by hiding each alpha (α) values using forward algo.

Thus,

$$\alpha_1^A = P(O/S) \times P(S_A) \quad (A)$$

$$= 0.8 \times 0.9 = 0.72$$

$$\alpha_1^B = P(O|S) * P(S_1) \quad (B)$$

$$= 0.1 \times 0.1 = 0.01.$$

Now,

$$\alpha_2^A = 0.8 * [0.72 \times 0.95] + [0.01 \times 0.05]$$

$= 0.8 \times [0.6845]$

$= 0.5476$

$\alpha_2^B = 0.7[0.72 \times 0.05] + [0.01 \times 0.95]$

$= 0.1 \times 0.0455 = 0.0046$

$\alpha_3^A = 0.2[0.5476 \times 0.95] + [0.0045 \times 0.05]$

$= 0.2 \times 0.53$

$= 0.106$

$\alpha_3^B = 0.9[(0.5476 \times 0.05) + (0.0046 \times 0.95)]$

$= 0.9 \times 0.0317$

$= 0.0285$

thus

$$P(O_1=0, O_2=0, O_3=1)$$

$$= 0\left[\alpha_1^A + \alpha_1^B\right] + 0\left[\alpha_2^A + \alpha_2^B\right] + 1\left[\alpha_3^A + \alpha_3^B\right]$$

$$= 0.706 + 0.0285$$

$$= 0.1345$$

so

$$P(O_1=0, O_2=0, O_3=1) = 0.1345$$

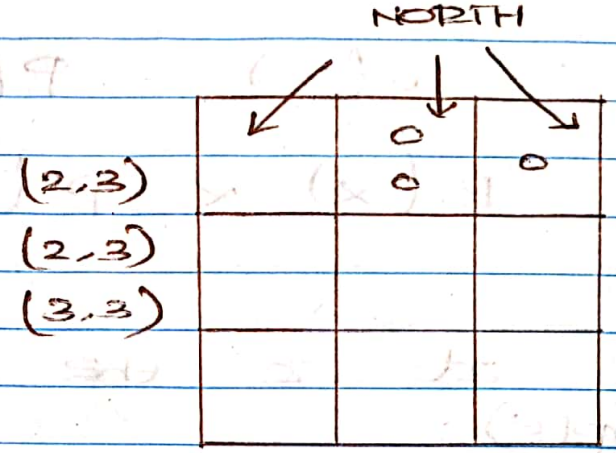
(c) Belief change after each observation of O_1, O_2, and O_3 is,

After $O_1 = 0$, $= [0.72, 0.01]$

After $O_2 = 0$, $= [0.5476, 0.0046]$

After $O_3 = 1$, $= [0.706, 0.0285]$

(5) PARTICLE FILTER



Given

a) $p(x) = 0.8$

Moves → clockwise

Here we follow the HMM structure.

1) Elapse (passage of time)
2) Weight (observation update)
3) Re-sample (resampled particles)

1) Elapse

$$x' = \text{sample}(P(x'|x))$$

this is like prior sampling - sampling frequencies reflect the transition probabilities.

these enough samples close to exact values before and after.

2) observe (weight)

$$w(x) = P(e|x)$$

$$B(x) \propto P(e|x) B'(x)$$

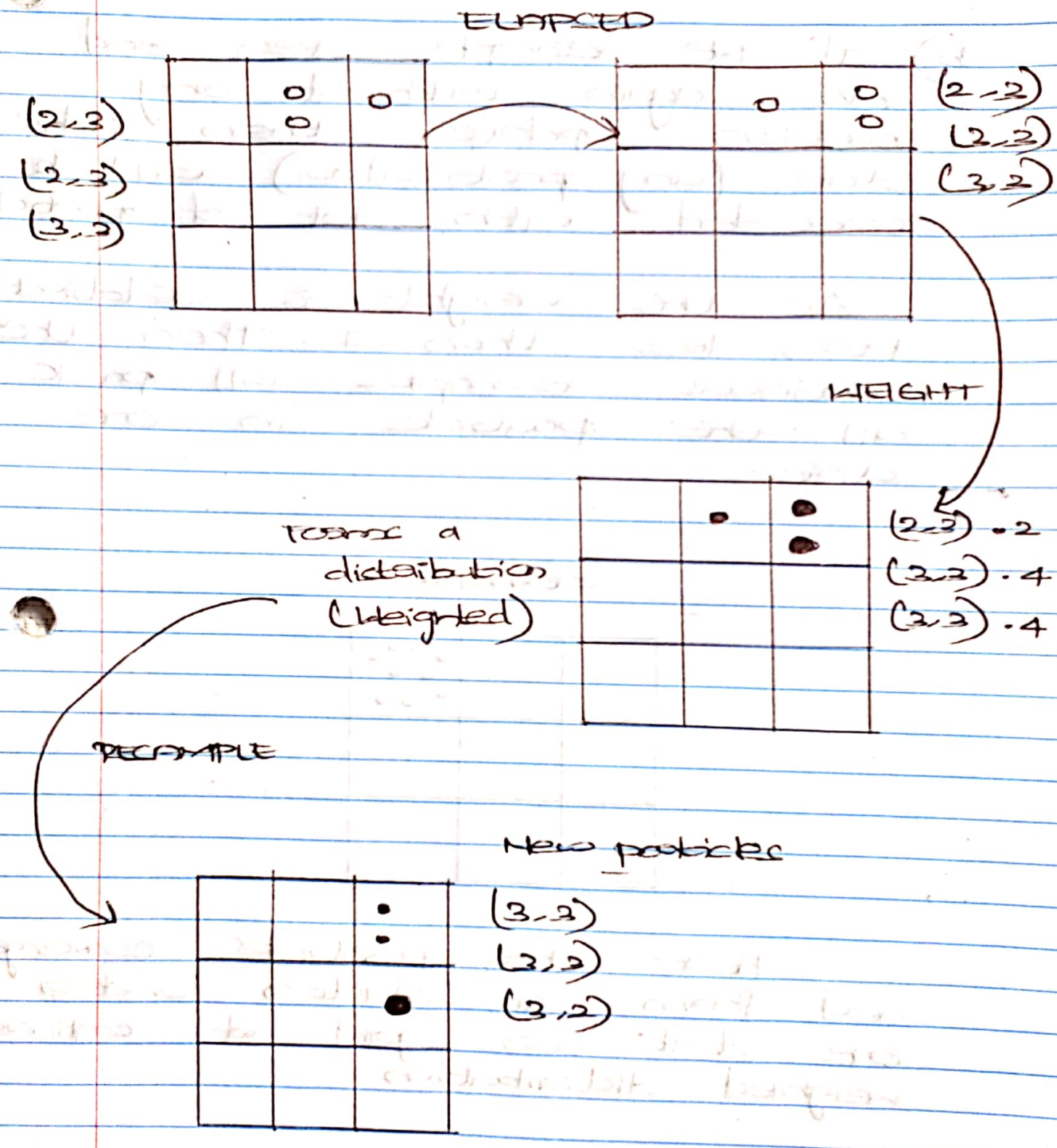
It is the approximation of $P(e)$.

2) Resample particles.

In accordance with weight the particles are resampled.

It is also called likelyhood weighting.

To draw the replacement we choose from the weighted sample distribution. This equivalent to renormalizing the distribution



these are used to improve localization.

First we do the sampling, then the particle moves to the next state as it moves in the clock wise direction.

then we observe and motion dynamics to weighted particles using probability distribution.

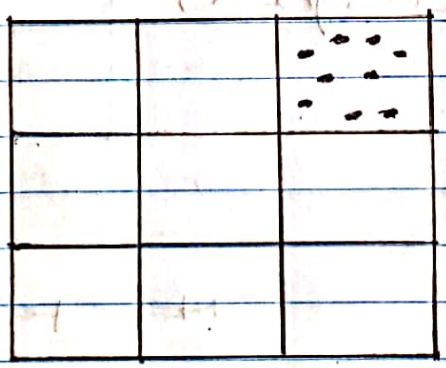
We rewrite them according to the evidence and use the weighted particles to form the distribution.

Again last we resampled the particles form weighted to unweighed particles using resampling distribution. Where the moving weighted particles has more importance and these states are stable.

b) If we resample over and over again without any motion update, then the states (any particular) will be crowded with lot of particles.

As the weight is relatively low, less than 1. Then the continous resampler will pack all the particles in one state.

<EXAMPLE>



Here the particles converge and forms a cluster under one state or grid of continous weighted distribution.