Jivitesh Poojary
jpoojary
2000116443

AML Homework 4

1)

		-2.2	-2	-0.3	0.1	0.2	0.4	1.6	1.7	1.9	2.0
		P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
-2.2	P1	0	0.2	1.9	2.3	2.4	2.6	3.8	3.9	4.1	4.2
-2	P2		0	1.7	2.1	2.2	2.4	3.6	3.7	3.9	4
-0.3	P3			0	0.4	0.5	0.7	1.9	2	2.2	2.3
0.1	P4				0	0.1	0.3	1.5	1.6	1.8	1.9
0.2	P5					0	0.2	1.4	1.5	1.7	1.8
0.4	P6						0	1.2	1.3	1.5	1.6
1.6	P7							0	0.1	0.3	0.4
1.7	P8								0	0.2	0.3
1.9	P9									0	0.1
2	P10										0

The distance matrix for the points

→ Single Link

(1) P4 – P5
P7 – P8
P9 – P10 } 0.1

(2) P1 – P2
(P4 – P5) – P6
(P7 – P8) – (P9 – 10) } 0.2 using MIN

(3) P3 – (P4 P5 P6) } 0.4.

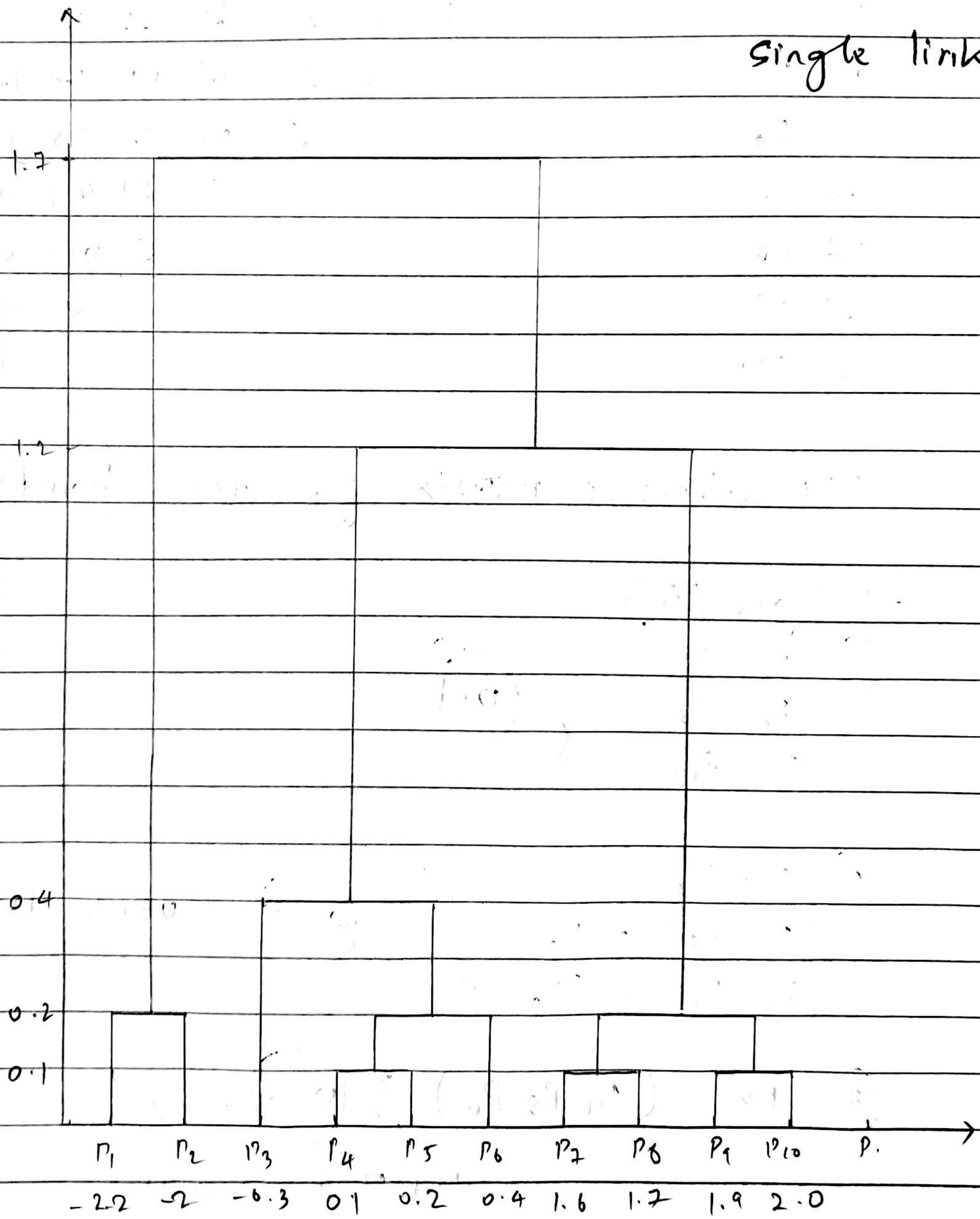
MIN(0.4, 0.6, 0.7)

MIN(1.9, 2 ... 1.2, 1.3, 1.3, 1.6)

(4) $(P_3\ P_4\ P_5\ P_6) - (P_7\ P_8\ P_9\ P_{10})$ } 1.2

(5) $(P_1\ P_2) - (P_3\ P_4\ P_5\ P_6\ P_7\ P_8\ P_9\ P_{10})$ } 1.7

MIN(1.9, 1.7 ... 4.2, 4) → single link



3) Complete Link :

(1) $P_4 - P_5$
$P_7 - P_8$
$P_9 - P_{10}$ } 0.1

(2) $P_1 - P_2$ } 0.2

(3) $P_3 - (P_4\ P_5\ P_6) = MAX(0.4, 0.5, 0.7) = 0.7$ }

(4) $(P_7 - P_8) - (P_9 - P_{10}) = MAX(0.3, 0.4, 0.2, 0.3) = 0.4$ }

(6) $(P_3\ P_4\ P_5\ P_6) - (P_7\ P_8\ P_9\ P_{10})$

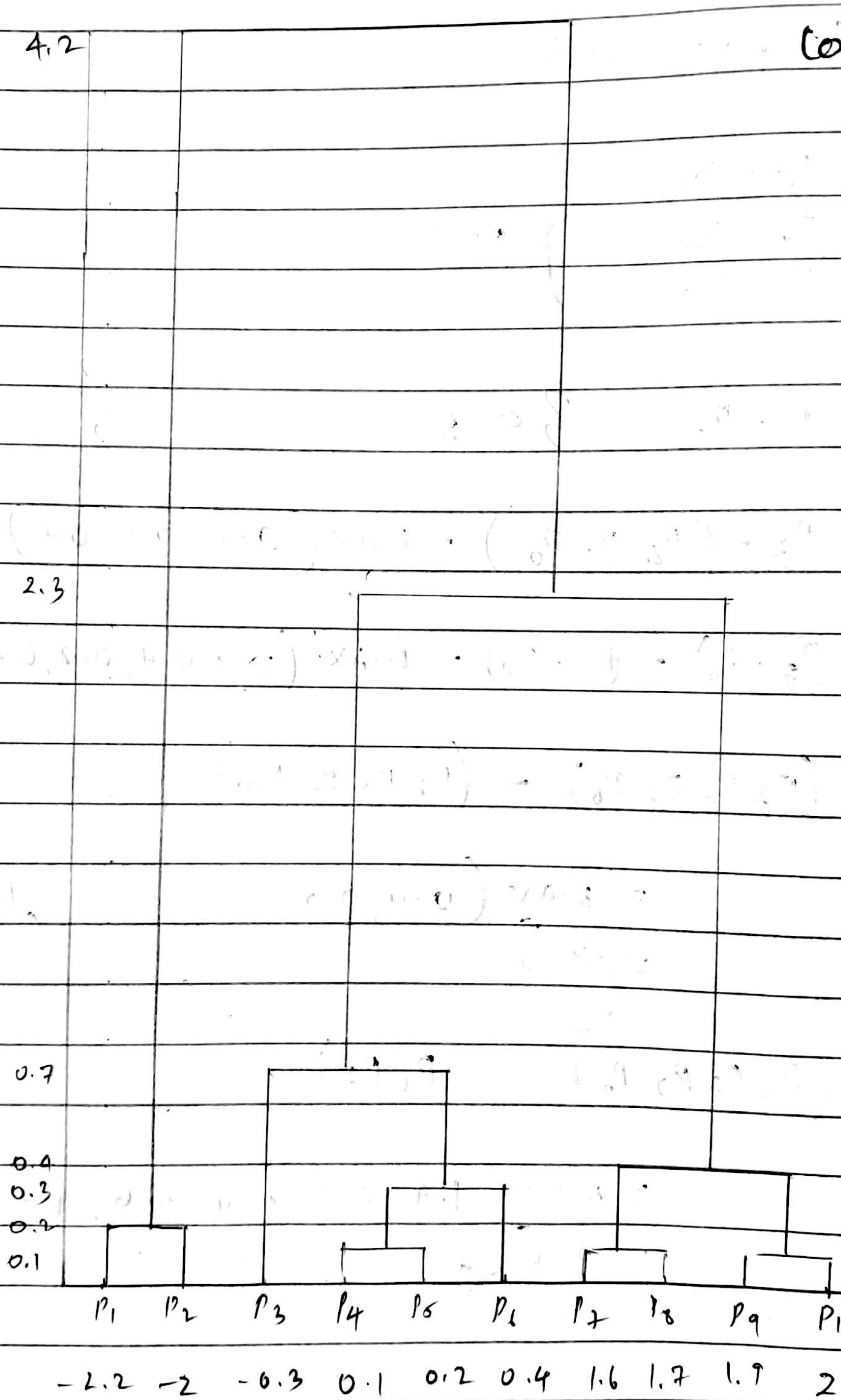
$= MAX(0.4, 0.5 \ldots\ldots, 1.5, 1.6)$
$= 2.3$

$(P_3\ P_4\ P_5\ P_6) - (P_1\ P_2)$

$= MAX(1.9, 2.3, 2.4, 2.6, 1.7, 2.1, 2.2, 2.4)$
$= 2.6$ ×

(7) $(P_1\ P_2) - (P_3\ P_4\ P_5\ P_6\ P_7\ P_8\ P_9\ P_{10})$
$= max(0.2 \ldots\ 4.2 \ldots\ 0.1)$
$= 4.2$

Complete Link



Iteration 1 :

2)

	X	Y
Centroid 1	0	0
Centroid 2	4	3

	X	Y	Dist from C_1	Dist fr. C_2
x_1	2	1	2.23606	2.8284
x_2	0	1	1	4.4721
x_3	2	2	2.8284	2.23606
x_4	4	0	4	3
P_5	6	1	6.0827	2.8284

x_1	Cen 1	Cluster 1
x_2	Cen 1	Cluster 1
x_3	Cen 2	Cluster 2
x_4	Cen 2	Cluster 2
x_5	Cen 2	Cluster 2

New Centroid :

Cluster 1 = $(2+0)/2$, $(1+1)/2$

cent 1 = $(1$, $1)$

Cluster 2 = $(2+4+6)/3$, $(2+0+1)/3$

cent 2 = $(4$, $1)$

Scanned by CamScanner

Iteration 2:

	X	Y
Centroid 1	1	1
Centroid 2	4	1

	X	Y	Dist for C_1	Dist for C_2
x_1	2	1	1	2
x_2	0	1	1	4
x_3	2	2	1.414	2.23606
x_4	4	0	3.162	1
x_5	6	1	5	2

x_1	cen1	Cluster 1	1
x_2	cen1	Cluster 1	1
x_3	cen1	Cluster 2	-1
x_4	cen2	Cluster 2	-1
x_5	cen2	Cluster 2	1

New Centroid

Cluster 1 $\quad (2+0+2)/3, \ (1+1+2)/3$

Cen 1 $\quad (1.333, \ 1.333)$

Cluster 2 $\quad (4+6)/2, \ (0+1)/2$

Cen 2 $\quad (5, \ 0.5)$

c) If we have new point $x_6 = (2,3)$

	X	Y	Dist from C_1	Dist fr C_2
x_6	2	3	1.7950	3.9051

x_6 is closer to C_1, hence it is part of cluster 1

	label
x_1	1
x_2	1
x_3	-1

$\}$ majority class = 1.

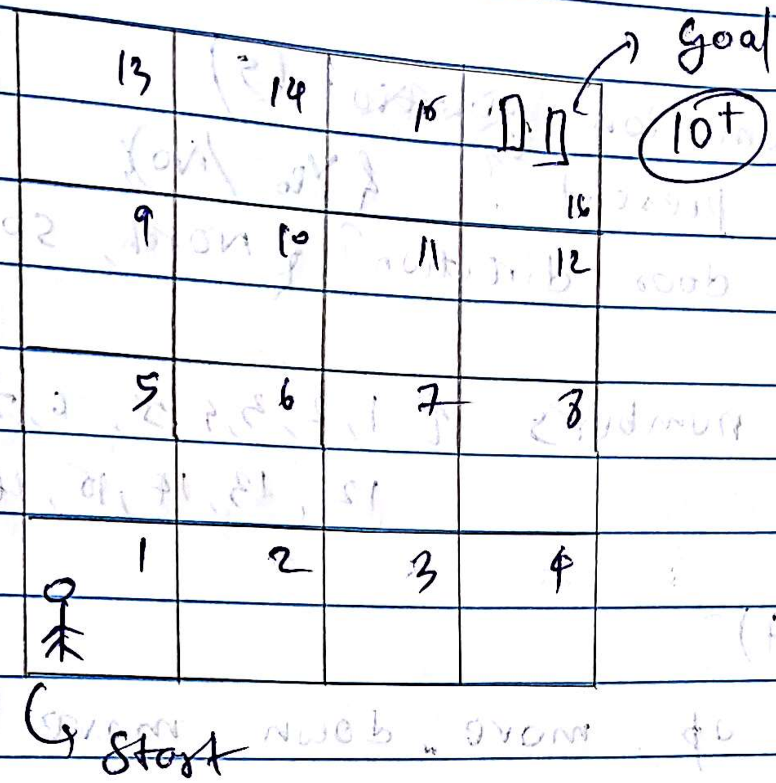
Hence label of x_6 is 1.

3)

Value iteration	Policy iteration
1) Here we start with 0 value for all states and recursively updating the value using Bellman update	1) It evalutes policies by calculating utilities for some fixed policy until convergence, the improved policy by updating it using one-step look-ahead.
2) It may take more time to converge	2) Under some conditions It may take less time to converge.

Q-Learning	Value iteration
1) In Q-Learning the agent does not have knowledge of the transition and reward functions	1) Here we do not make such an assumption.
2) It has state action pair	2) It has only state as it is for all actions
3) Most practical applicat use Q-learning	3) It is prefered when δ & R are known

A)

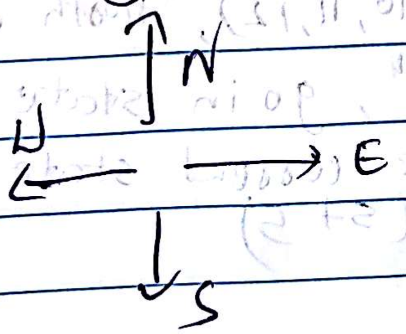


There are 16 states in all

Discount:

Gamma $(\gamma) = 0.95$

Assuming the direction:



Hence the doors associated with each state.

⇒ State information ~~(xxxxx)~~: (S)

- Gold present? {Yes / No}
- open door direction? {North, south, east, west}
- State numbers {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16}.

⇒ Actions: (A)

{move up, move down, move left, move right, open north door, open south door, open east door, open west door}

⇒ Transition function: (T):

T (s, a, s') = Probability.

$T(\underbrace{\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}}^{SE}$, north door opens + "move north", go in state above current state (s + 5)) = 1

$T(\overbrace{\{1, 2, 3, 5, 6, 7, 9, 10, 11, 13, 14, 15\}}^{SE}$, east door opens + "move left", go in state left of current state, (s + 3)) = 1

T (open north door ~~open~~, ~~move~~ north door open)
T (open south door ~~open~~, ~~move~~ south ~~door~~ open)
T (open east door ~~open~~, ~~move~~ east door open)
T (open west door, west door open).

s ∈

T ({5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16}, open south door + "more south", go in state below the current state (s-5)) = 1

s ∈

T ({2, 3, 4, 6, 7, 8, 10, 11, 12, 14, 15, 16}, west door open + "more west", go in state left of current state (s-1)) = 1

⇒ Reward function:
R (s, a, s')

1 Goal is to reach the gold with opening minimum number of doors

$$R = \begin{cases} +10 & \text{for getting gold} \\ -1 & \text{for each door opened} \\ 0 & \text{for anything else} \end{cases}$$

(e) The goal is to reach the gold with minimum number of steps

$$R = \begin{cases} +10 & \text{for getting gold} \\ -1 & \text{for taken any step} \\ 0 & \text{for everything else} \end{cases}$$

The difference between (1) & (2) is the change in the reword function in (1) you are penalized for opening a door, while in (2) you are penalized for taking a step through the door. the objective is different.

5)

Given: $P(y=1|x) = \dfrac{e^{\psi(x;y)}}{1+e^{\psi(x;y)}}$

taking log on both sides,

$$\log\left(P(y=1|x)\right) = \log\left(\frac{e^{\psi(x;y)}}{1+e^{\psi(x;y)}}\right)$$

$$\log(P(y=1|x)) = \log\left(e^{\psi(x;y)}\right) - \log\left(1+e^{\psi(x;y)}\right)$$

$$\therefore \log[P(y=1|x)] = \psi(x;y) - \log\left(1+e^{\psi(x;y)}\right)$$

Differentiating both sides w.r.t $\psi(x_i, y_i)$

$$\frac{\partial \log[P(y=1|x)]}{\partial \psi(x_i, y_i)} = \frac{\partial}{\partial \psi(x_i, y_i)} \psi(x;y) - \frac{\partial}{\partial \psi(x_i, y_i)} \left[\log\left(1+e^{\psi(x;y)}\right)\right]$$

$$= I - \frac{1}{1+e^{\psi(x;y)}} \frac{\partial}{\partial \psi(x_i, y_i)} e^{\psi(x,y)}$$

$$= I - \frac{1}{}$$

$$= I - \frac{1}{1+e^{\phi(x_i;y)}} \times e^{\phi(x_i;y)} \times (1)$$

$$= I - \frac{e^{\psi(x;y)}}{1+e^{\psi(x_i;y)}}$$

Thus,

$$\frac{\partial \log P(y_i=1/x_i)}{\partial \psi(x_i;y_i)} = I(y_i=1) - P(y_i=1/x_i)$$