CSE 415

Assignment 6

UW NetID = vaibhavi@uw.edu

Problem 1: Should Anyone Panic?

Case 1: Lucy, in Seattle, tests Positive for HPAI

Event Definitions

E = Evidence event: Event that a person tests Positive for HPAI
H = Hypothesis event: Event that a person is infected HPAI in Seattle

Prior Probability

P(H) = Chance of being infected with HPAI = 0.001
Reason: We are given that in this season, one out of 1000 folks in Seattle are affected by HPAI

Likelihood

Chance that you get to see some evidence in conjunction with the hypothesis.

P(~E|~H) = Chance of testing negative if person doesn't have the disease = 0.95
P(E|~H) = 1- P(E|H) = 0.05
Reason: The HPAI test is 95% effective, chance of False Positive = 0.05

P(~E|H) = Chance of testing negative if the person has the disease = 0
P(E|H) = 1- P(~E|H) = 1
Reason: Probability of a false negative in the HPAI test is 0

Marginal Likelihood (Model Evidence)

Chance that you observe the evidence by itself across all hypotheses.

P(E) = P(E|H).P(H) + P(E|~H).P(~H)
= 1*0.001 + 0.05*0.999
= 0.05095

P(~E) = P(~E|H).P(H) + P(~E|~H).P(~H)
=0*0.001 + 0.95*0.999
= 0.94905

The table with conditions, outcomes, and conditional probabilities can be now constructed.

Conditions\ Outcomes	*Positive Test (E)*	*Negative Test (~E)*	*Prior Probabilities*
Person has HPAI (H)	P(E|H) = 1	P(~E|H) = 0	P(H) = 0.001
Person does not have HPAI (~H)	P(E|~H) = 0.05	P(~E|~H) = 0.95	P(~H) = 0.999
Marginal Likelihood	P(E) = 0.05095	P(~E) = 0.94905	

(a) What's the updated probability that she has HPAI?
The probability that Lucy has HPAI, given she tested positive = P(H|E)

Using Bayes Theorem,

Posterior Probability = Prior Probability * Likelihood / Marginal Likelihood

$P(H|E) = P(H) * P(E|H) / P(E)$

$= 0.001*1 / 0.05095$

$= \mathbf{0.0196 = 1.96\%}$

Conclusion:

Lucy should not be worried about further assistance, as she most likely doesn't have the disease.

Case 2: James, arriving from Belize, tests Positive for HPAI

Event Definitions

E = Evidence event: Event that a person tests Positive for HPAI

H_B = Hypothesis event: Event that a person is infected HPAI arriving from Belize

Prior Probability

$P(H_B)$ = Chance of being infected with HPAI coming back from Belize = 0.0125

$P(\sim H_B)$ = Chance of not being infected with HPAI coming back from Belize = 0.9875

Reason: We are given that 1 out of 80 people coming back from Belize come back with HPAI. Let's assume that James was away from U.S. during the flu season, and the chance of getting flu depends upon conditions in Belize alone.

Likelihood

Since it was the same test as Lucy, the Likelihoods for test accuracy are not going to change.

$P(\sim E|\sim H_B) = 0.95$

$P(E|\sim H_B) = 1- P(E|H_B) = 0.05$

Reason: The HPAI test is 95% effective, Chance of False Positive = 0.05

$P(\sim E|H_B) = 0$
$P(E|H_B) = 1 - P(E|\sim H_B) = 1$
Reason: Probability of a false negative in the HPAI test is 0

Marginal Likelihood (Model Evidence)

Chance that you observe the evidence by itself across the hypotheses.

$$\begin{aligned} P(E) &= P(E|H_B).P(H_B) + P(E|\sim H_B).P(\sim H_B) \\ &= 1*0.0125 + 0.05*0.9875 \\ &= 0.061875 \end{aligned}$$

$$\begin{aligned} P(\sim E) &= P(\sim E|H_B).P(H_B) + P(\sim E|\sim H_B).P(\sim H_B) \\ &= 0*0.0125 + 0.95*0.9875 \\ &= 0.938125 \end{aligned}$$

(b) What's the updated probably that James has HPAI?

Conditions \ Outcomes	*Positive Test (E)*	*Negative Test (~E)*	*Prior Probabilities*
Person has HPAI from Belize (H_B)	$P(E\|H_B) = 1$	$P(\sim E\|H_B) = 0$	$P(H_B) = 0.0125$
Person does not have HPAI from Belize ($\sim H_B$)	$P(E\|\sim H_B) = 0.05$	$P(\sim E\|\sim H_B) = 0.95$	$P(\sim H_B) = 0.9875$
Marginal Likelihood	$P(E) = 0.061875$	$P(\sim E) = 0.938125$	

Using Bayes Theorem,

Posterior Probability = Prior Probability * Likelihood / Marginal Likelihood

$$\begin{aligned} P(H_B|E) &= P(H_B) * P(E|H_B) / P(E) \\ &= 0.0125 * 1 / 0.061875 \\ &= \mathbf{0.2020} \\ &= \mathbf{20.2\%} \end{aligned}$$

Conclusion:

James should be slightly worried, and seek further assistance, as he could have HPAI with a 20.2% chance.

Problem 2: Mecha-Mouse in the Hostel for Travelling Droids

A Markov Decision Process (MDP) can be defined by the following parameters:

MDP (S, A, R, T)

Set of states, S:

Dormitory (D), Lavatory (L), Pantry (P), Mess Hall (M).
Ambushed, Kaput

Constraints:

There is some danger than the "Compu-Cat" will ambush the mouse at any time, putting it in the Ambushed state, from which it can only go to the dead-end Kaput state.

Set of Actions, A

X: exit current room;
Y: alternative action;
Z: remain as is

Transition Table T(s,a,s')

[Model Dynamics]

s, a	**Dormitory**	**Lavatory**	**Pantry**	**Mess Hall**	**Ambushed**	**Kaput**
Dormitory, X	0	0.4	0	0.6	0	0
Dormitory, Y	0	0.6	0	0.4	0	0
Dormitory, Z	0.75	0	0	0	0.25	0
Lavatory, X	0.4	0	0.6	0	0	0
Lavatory, Y	0.6	0	0.4	0	0	0
Lavatory, Z	0	0.75	0	0	0.25	0
Pantry, X	0	0.6	0	0.4	0	0
Pantry, Y	0	0.4	0	0.6	0	0
Pantry, Z	0	0	0.75	0	0.25	0
Mess Hall, X	0.4	0	0.6	0	0	0
Mess Hall, Y	0.6	0	0.4	0	0	0
Mess Hall, Z	0	0	0	0.75	0.25	0
Ambushed, *	0	0	0	0	0	1.0
Kaput, *	0	0	0	0	0	1.0

Rewards Table R(s,a,s')
[Depends only on the current state s]

State s	R(s)
Dormitory (D)	0
Lavatory (L)	4
Pantry (P)	10
Mess Hall (M)	2
Ambushed (A)	-50
Kaput (K)	0

Note: To get the rewards at a state, one needs to take an action to leave the current state.

1. Give the number of different policies that are possible for Mecha-mouse in the hostel.

A policy is a mapping from each state to a possible action from that state.

All 6 states (D, L, M, P, A, K) have 3 actions each (X,Y,Z).

The total number of policies equals the number of mappings possible for states.
This is a combinatorics problem of choosing 1 out of 3 option for 6 states.

Total number of policies = 3 x 3 x 3 x 3 x 3 x 3 = 3^6 = **729**

Note: Even though Kaput is a dead-end state, we cannot assume that there are fewer than 3 actions, since the question explicitly mentions that X,Y,Z are available to all states.

---

2. Manually apply the values iteration method to this problem for six iterations. Show the value at each state in each iteration. Assume that the discount factor is 0.5.

Discount Factor , γ
γ = 0.5

Terminology for upcoming tables for iterations.

We iteratively apply the Bellman Update Step to each combination of **start state= s**, **action= a**, **next state= s'**, **optimal action so far= a*** to find Value of state **V(s)**.
State s ∈ S = {D, L, P, M, A, K}

Bellman Formulae

$$V^*(s) = \max_a Q^*(s,a)$$

$$Q^*(s,a) = \sum_{s'} T(s,a,s') \left[R(s,a,s') + \gamma V^*(s')\right]$$

$$V^*(s) = \max_a \sum_{s'} T(s,a,s') \left[R(s,a,s') + \gamma V^*(s')\right]$$

Bellman Value Iteration Update Step

$$V_{k+1}(s) \leftarrow \max_a \sum_{s'} T(s,a,s') \left[R(s,a,s') + \gamma V_k(s')\right]$$

Iteration 0: Initial setup

We Start with V0(s) = 0 for all states.

State s	**V_0(s)**
Dormitory , D	0
Lavatory, L	0
Pantry, P	0
Mess Hall , M	0
Ambushed, A	0
Kaput , K	0

Iterations 1 to 6 are calculated in the following pages.

Iteration 1

R & T values fetched from Rewards and Transition Table

Q_1 (s,a) = $\sum_{s'} \mathrm{T}(\mathrm{s,a,s'}) * (\mathrm{R}(\mathrm{s'}) + \gamma\, \mathrm{V0}(\mathrm{s}))$ for each (s,a) combination

V_1(s) = max $_a$(Q_1(s,a)), i.e. Q value associated with best action for given s.

s	V_0(s)	a	s'	T(s,a,s')	R(s')	γ V_0(s)	T(s,a,s') *(R(s') + γ V_0(s))	Q_1(s,a)	V_1(s)	a*
D	0	X	L	0.4	4	0	1.6	2.8	**3.2**	Y
			M	0.6	2		1.2			
		Y	L	0.6	4		2.4	3.2		
			M	0.4	2		0.8			
		Z	D	0.75	0		0	-12.5		
			A	0.25	-50		-12.5			
L	0	X	D	0.4	0	0	0	6	**6**	X
			P	0.6	10		6			
		Y	D	0.6	0		0	4		
			P	0.4	10		4			
		Z	L	0.75	4		3	-9.5		
			A	0.25	-50		-12.5			
P	0	X	L	0.6	4	0	2.4	3.2	**3.2**	X
			M	0.4	2		0.8			
		Y	L	0.4	4		1.6	2.8		
			M	0.6	2		1.2			
		Z	P	0.75	10		7.5	-5		
			A	0.25	-50		-12.5			
M	0	X	D	0.4	0	0	0	6	**6**	X
			P	0.6	10		6			
		Y	D	0.6	0		0	4		
			P	0.4	10		4			
		Z	M	0.75	2		1.5	-11		
			A	0.25	-50		-12.5			
A	0	X	K	1	0	0	0	0	**0**	*
		Y	K	1	0		0	0		
		Z	K	1	0		0	0		
K	0	X	K	1	0	0	0	0	**0**	*
		Y	K	1	0		0	0		
		Z	K	1	0		0	0		

Iteration 2

R & T values fetched from Rewards and Transition Table

$Q_2(s,a) = \sum_{s'} T(s,a,s') * (R(s') + \gamma V1(s))$ for each (s,a) combination

$V_2(s) = \max_a(Q_2(s,a))$, i.e. Q value associated with best action for given s.

s	$V_1(s)$	a	s'	T(s,a,s')	R(s')	$\gamma V_1(s)$	$T(s,a,s') * (R(s') + \gamma V_1(s))$	$Q_2(s,a)$	$V_2(s)$	a*
D	3.2	X	L	0.4	4	1.6	2.24	4.4	**4.8**	Y
			M	0.6	2		2.16			
		Y	L	0.6	4		3.36	4.8		
			M	0.4	2		1.44			
		Z	D	0.75	0		1.2	-10.9		
			A	0.25	-50		-12.1			
L	6	X	D	0.4	0	3	1.2	9	**9**	X
			P	0.6	10		7.8			
		Y	D	0.6	0		1.8	7		
			P	0.4	10		5.2			
		Z	L	0.75	4		5.25	-6.5		
			A	0.25	-50		-11.75			
P	3.2	X	L	0.6	4	1.6	3.36	4.8	**4.8**	X
			M	0.4	2		1.44			
		Y	L	0.4	4		2.24	4.4		
			M	0.6	2		2.16			
		Z	P	0.75	10		8.7	-3.4		
			A	0.25	-50		-12.1			
M	6	X	D	0.4	0	3	1.2	9	**9**	X
			P	0.6	10		7.8			
		Y	D	0.6	0		1.8	7		
			P	0.4	10		5.2			
		Z	M	0.75	2		3.75	-8		
			A	0.25	-50		-11.75			
A	0	X	K	1	0	0	0	0	**0**	*
		Y	K	1	0		0	0		
		Z	K	1	0		0	0		
K	0	X	K	1	0	0	0	0	**0**	*
		Y	K	1	0		0	0		
		Z	K	1	0		0	0		

Iteration 3

R & T values fetched from Rewards and Transition Table

$Q_3(s,a) = \sum_{s'} \mathrm{T}(s, a, s') * (\mathrm{R}(s') + \gamma \mathrm{V2}(s))$ for each (s,a) combination

$V_3(s) = \max_a(Q_3(s,a))$, i.e. Q value associated with best action for given s.

s	$V_2(s)$	a	s'	T(s,a,s')	R(s')	γ $V_2(s)$	T(s,a,s') *(R(s') + γ $V_2(s)$)	$Q_3(s,a)$	$V_3(s)$	a*
D	4.8	X	L	0.4	4	2.4	2.56	5.2	**5.6**	Y
			M	0.6	2		2.64			
		Y	L	0.6	4		3.84	5.6		
			M	0.4	2		1.76			
		Z	D	0.75	0		1.8	-10.1		
			A	0.25	-50		-11.9			
L	9	X	D	0.4	0	4.5	1.8	10.5	**10.5**	X
			P	0.6	10		8.7			
		Y	D	0.6	0		2.7	8.5		
			P	0.4	10		5.8			
		Z	L	0.75	4		6.375	-5		
			A	0.25	-50		-11.375			
P	4.8	X	L	0.6	4	2.4	3.84	5.6	**5.6**	X
			M	0.4	2		1.76			
		Y	L	0.4	4		2.56	5.2		
			M	0.6	2		2.64			
		Z	P	0.75	10		9.3	-2.6		
			A	0.25	-50		-11.9			
M	9	X	D	0.4	0	4.5	1.8	10.5	**10.5**	X
			P	0.6	10		8.7			
		Y	D	0.6	0		2.7	8.5		
			P	0.4	10		5.8			
		Z	M	0.75	2		4.875	-6.5		
			A	0.25	-50		-11.375			
A	0	X	K	1	0	0	0	0	**0**	*
		Y	K	1	0		0	0		
		Z	K	1	0		0	0		
K	0	X	K	1	0	0	0	0	**0**	*
		Y	K	1	0		0	0		
		Z	K	1	0		0	0		

Iteration 4

R & T values fetched from Rewards and Transition Table

Q_4 (s,a) = $\sum_{s'} \mathrm{T}(\mathrm{s,a,s'}) * (\mathrm{R}(\mathrm{s'}) + \gamma \, \mathrm{V3}(\mathrm{s}))$ for each (s,a) combination

V_4 (s) = max $_a$(Q_4 (s,a)), i.e. Q value associated with best action for given s.

s	V_3(s)	a	s'	T(s,a,s')	R(s')	γ V_3(s)	T(s,a,s') *(R(s') + γ V_3(s))	Q_4(s,a)	V_4(s)	a*
D	5.6	X	L	0.4	4	2.8	2.72	5.6	**6**	Y
			M	0.6	2		2.88			
		Y	L	0.6	4		4.08	6		
			M	0.4	2		1.92			
		Z	D	0.75	0		2.1	-9.7		
			A	0.25	-50		-11.8			
L	10.5	X	D	0.4	0	5.25	2.1	11.25	**11.25**	X
			P	0.6	10		9.15			
		Y	D	0.6	0		3.15	9.25		
			P	0.4	10		6.1			
		Z	L	0.75	4		6.9375	-4.25		
			A	0.25	-50		-11.1875			
P	5.6	X	L	0.6	4	2.8	4.08	6	**6**	X
			M	0.4	2		1.92			
		Y	L	0.4	4		2.72	5.6		
			M	0.6	2		2.88			
		Z	P	0.75	10		9.6	-2.2		
			A	0.25	-50		-11.8			
M	10.5	X	D	0.4	0	5.25	2.1	11.25	**11.25**	X
			P	0.6	10		9.15			
		Y	D	0.6	0		3.15	9.25		
			P	0.4	10		6.1			
		Z	M	0.75	2		5.4375	-5.75		
			A	0.25	-50		-11.1875			
A	0	X	K	1	0	0	0	0	**0**	*
		Y	K	1	0		0	0		
		Z	K	1	0		0	0		
K	0	X	K	1	0	0	0	0	**0**	*
		Y	K	1	0		0	0		
		Z	K	1	0		0	0		

Iteration 5

R & T values fetched from Rewards and Transition Table

Q_5 (s,a) = $\sum_{s'} \mathrm{T(s, a, s')} * (\mathrm{R(s')} + \gamma \mathrm{V4(s)})$ for each (s,a) combination

V_5 (s) = max $_a$(Q_5 (s,a)), i.e. Q value associated with best action for given s.

s	V_4(s)	a	s'	T(s,a,s')	R(s')	γ V_4(s)	T(s,a,s') *(R(s') + γ V_4(s))	Q_5(s,a)	V_5(s)	a*
D	6	X	L	0.4	4	3	2.8	5.8	**6.2**	Y
			M	0.6	2		3			
		Y	L	0.6	4		4.2	6.2		
			M	0.4	2		2			
		Z	D	0.75	0		2.25	-9.5		
			A	0.25	-50		-11.75			
L	11.25	X	D	0.4	0	5.625	2.25	11.625	**11.625**	X
			P	0.6	10		9.375			
		Y	D	0.6	0		3.375	9.625		
			P	0.4	10		6.25			
		Z	L	0.75	4		7.21875	-3.875		
			A	0.25	-50		-11.09375			
P	6	X	L	0.6	4	3	4.2	6.2	**6.2**	X
			M	0.4	2		2			
		Y	L	0.4	4		2.8	5.8		
			M	0.6	2		3			
		Z	P	0.75	10		9.75	-2		
			A	0.25	-50		-11.75			
M	11.25	X	D	0.4	0	5.625	2.25	11.625	**11.625**	X
			P	0.6	10		9.375			
		Y	D	0.6	0		3.375	9.625		
			P	0.4	10		6.25			
		Z	M	0.75	2		5.71875	-5.375		
			A	0.25	-50		-11.09375			
A	0	X	K	1	0	0	0	0	**0**	*
		Y	K	1	0		0	0		
		Z	K	1	0		0	0		
K	0	X	K	1	0	0	0	0	**0**	*
		Y	K	1	0		0	0		
		Z	K	1	0		0	0		

Iteration 6

R & T values fetched from Rewards and Transition Table

Q_6 (s,a) = $\sum_{s'} T(s,a,s') * (R(s') + \gamma V5(s))$ for each (s,a) combination

V_6 (s) = max $_a$(Q_6 (s,a)), i.e. Q value associated with best action for given s.

s	V_5(s)	a	s'	T(s,a,s')	R(s')	γ V_5(s)	T(s,a,s') *(R(s') + γ V_5(s))	Q_6(s,a)	V_6(s)	a*
D	6.2	X	L	0.4	4	3.1	2.84	5.9	**6.3**	Y
			M	0.6	2		3.06			
		Y	L	0.6	4		4.26	6.3		
			M	0.4	2		2.04			
		Z	D	0.75	0		2.325	-9.4		
			A	0.25	-50		-11.725			
L	11.625	X	D	0.4	0	5.8125	2.325	11.8125	**11.813**	X
			P	0.6	10		9.4875			
		Y	D	0.6	0		3.4875	9.8125		
			P	0.4	10		6.325			
		Z	L	0.75	4		7.359375	-3.6875		
			A	0.25	-50		-11.046875			
P	6.2	X	L	0.6	4	3.1	4.26	6.3	**6.3**	X
			M	0.4	2		2.04			
		Y	L	0.4	4		2.84	5.9		
			M	0.6	2		3.06			
		Z	P	0.75	10		9.825	-1.9		
			A	0.25	-50		-11.725			
M	11.625	X	D	0.4	0	5.8125	2.325	11.8125	**11.813**	X
			P	0.6	10		9.4875			
		Y	D	0.6	0		3.4875	9.8125		
			P	0.4	10		6.325			
		Z	M	0.75	2		5.859375	-5.1875		
			A	0.25	-50		-11.046875			
A	0	X	K	1	0	0	0	0	**0**	*
		Y	K	1	0		0	0		
		Z	K	1	0		0	0		
K	0	X	K	1	0	0	0	0	**0**	*
		Y	K	1	0		0	0		
		Z	K	1	0		0	0		

3. Based on your analysis, give the optimal policy as an action for each state.

State s	$V_0(s)$	$V_1(s)$	$V_2(s)$	$V_3(s)$	$V_4(s)$	$V_5(s)$	$V_6(s)$	Optimal Policy
D	0	3.2	4.8	5.6	6	6.2	6.3	**Y**
L	0	6	9	10.5	11.25	11.625	11.8125	**X**
P	0	3.2	4.8	5.6	6	6.2	6.3	**X**
M	0	6	9	10.5	11.25	11.625	11.8125	**X**
A	0	0	0	0	0	0	0	*
K	0	0	0	0	0	0	0	*

The Optimal Policy is of the format:
Dormitory D: Y, Lavatory L : X, Pantry P: X,
Mess Hall M: X, Ambushed A: Any(X/Y/Z), Kaput K: Any (X,Y,Z)

Thus, there are 9 distinct optimal policies considering 3 actions at Ambushed, and Kaput states.

Visualization of Optimal State Values

Note: The state values for Ambushed and Kaput States, always stay remain 0.