CS440/ECE448 Spring 2019 Final Review Solutions

<u>Problem 1 Solution:</u>

Let S = event friend drives small car
L = event friend drives large car
T = event friend is at work on time

P(S|T) = P(T,S)/P(T) = P(T,S) / (P(T,S)+P(T,L))

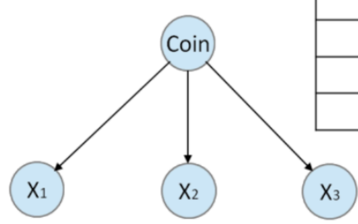
P(T,S) = P(T|S)*P(S) = 0.9*3/4 = 0.675

P(T,L) = P(T|L)*P(L) = 0.6*1/4 = 0.15

P(S|T) = 0.675 / (0.675+0.15) = 0.818

<u>Problem 2 Solution:</u>

a. Bayesian Network:



y	P(Coin=y)
a	1/3
b	1/3
c	1/3

y	$P(X_i=H\|Coin=y)$
a	0.2
b	0.6
c	0.8

b. Using the abbreviation C:=Coin to simplify the notation:

$$P(C=y|X_1,X_2,X_3) = \frac{P(C=y,X_1,X_2,X_3)}{P(X_1,X_2,X_3)} = \frac{P(X_1|C=y)P(X_2|C=y)P(X_3|C=y)P(C=y)}{P(X_1,X_2,X_3)}$$

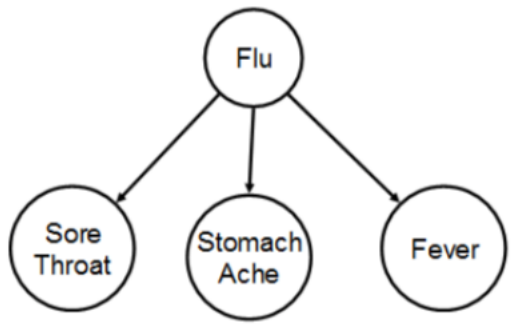
then, $argmax_y\ P(C=y|X_1,X_2,X_3) = argmax_y\ P(X_1|C=y)P(X_2|C=y)P(X_3|C=y)$

$$P(X_1=H|C=a)P(X_2=H|C=a)P(X_3=T|C=a) = 0.2^2 0.8 = 0.032$$
$$P(X_1=H|C=b)P(X_2=H|C=b)P(X_3=T|C=b) = 0.6^2 0.4 = 0.144$$
$$P(X_1=H|C=c)P(X_2=H|C=c)P(X_3=T|C=c) = 0.8^2 0.2 = 0.128$$

Hence, the coin that is most likely to have been drawn from the bag, given the observations HHT, is b.

<u>Problem 3 Solution</u>

a.



Flu	Value
T	3/7
F	4/7

	Flu	¬Flu
Sore Throat	2/3	1/2
¬Sore Throat	1/3	1/2

	Flu	¬Flu
Stomach Ache	1/3	1/2
¬Stomach Ache	2/3	1/2

	Flu	¬Flu
Fever	2/3	1/4
¬Fever	1/3	3/4

b. Let α = P(Fever, Stomach Ache, ¬Sore Throat).
P(Flu | Fever, Stomach ache, ¬Sore Throat)
= (1/α) * P(Flu) * P(Stomach ache | Flu) * P(Fever | Flu) * P(¬Sore Throat | Flu)
= (1/α) * 3/7 * 1/3 * 2/3 * 1/3 = (1/α) * 2/63 = (1/α) * 8 / 252.
P(¬Flu | Fever, Stomach ache, ¬Sore Throat)
= (1/α) * P(¬Flu) * P(Stomach ache | ¬Flu) * P(Fever | ¬Flu) * P(¬Sore Throat | ¬Flu)
= (1/α) * 4/7 * 1/2 * 1/4 * 1/2 = (1/α) * 1/28 = (1/α) * (9/252).
Adding these together, we have: α = (8+9)/252 = 17/ 252.
Thus, the probability of having flu is 8/17.

Problem 4 Solution

$$w \leftarrow w + \eta y x$$

W1	W2	W3	Bias
0	0	0	0
-1	-1	1	1
-1	-1	1	1

-1	-1	1	1
-1	-1	1	1
0	0	0	2
-1	-1	-1	1

Problem 5 Solution:

P(X | Y=0) = $\prod_{i=1}^{100} P$(x$_i$|Y=0) = $a^{50}(1\text{-}a)^{50}$

P(X | Y=1) = $\prod_{i=1}^{100} P$(x$_i$|Y=1) = $b^{50}(1\text{-}b)^{50}$
P(Y=1|X) = P(X|Y=1)P(Y=1) / [P(X|Y=0)P(Y=0) + P(X|Y=1)P(Y=1)]
= $b^{50}(1\text{-}b)^{50}$/[$a^{50}(1\text{-}a)^{50}$+ $b^{50}(1\text{-}b)^{50}$]

Problem 6 Solution:

a. Yes, there are no (undirected) cycles.

b. D and E are not independent because P(D,E) != P(D)P€. However, they are independent given B, because P(D,E|B)=P(D|B)P(E|B).

c. There are 6 random variables, so $2^6 - 1 = 63$ parameters

d. There are still 6 random variables, but each variable has 3 separate values it can take. Thus, $3^6 - 1 = 728$.
As for conditional probability tables, we have tables for P(A), P(B|A), P(C|A), P(D|B), P(E|B), and P(F|C). The number of values for these tables are 3-1=2, 3^2-3=6, 3^2-3=6, 3^2-3=6, 3^2-3=6, 3^2-3=6. Adding these up, we have 2+6+6+6+6+6=32.

e. Write down the expression for the joint probability distribution of all the variables in the network. P(A,B,C,D,E,F)=P(A)·P(B|A)·P(C|A)·P(D|B)·P(E|B)·P(F |C)

f. P (A = 0, B = 1, C = 1, D = 0) = P (A = 0) · P (B = 1 | A = 0) · P (C = 1 | A = 0) · P (D = 0 | B = 1) = (1−0.8)·0.2·0.6·(1−0.5)
= (1/5)(1/5)(3/5)(1/2) = 3/250

g. P(B|A,¬D) = P(B,A,¬D)/(P(B,A,¬D)+ P(¬B,A,¬D))
P(B,A,¬D) = P(A) P(B|A)P(¬D|B) = 0.8·0.5·0.5 = 0.2
P(¬B,A,¬D)=P(A)P(¬B|A)P(¬D|¬B) = 0.8·0.5·0.4 = 0.16

So
P(B|A,¬D)=0.2/0.36 = 20/36 = 5/9

h. P(A,B,C)=P(C|A)P(B|A)P(A)

P(B,C) = sum_A P(A,B,C)
P(E,B,C) = P(E|B) P(B,C)
P(E,C) = sum_B P(E,B,C)

Problem 7 Solution:

a. Bayesian Network:

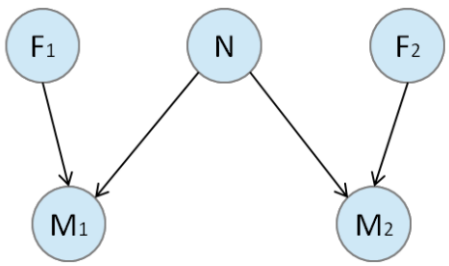


Table for N: depends on the prior probabilities of observing different numbers of stars.

Tables for F_1 and F_2:

F	Values
True	f
False	1-f

Table for M_1 (resp. M_2) given F_1 (resp. F_2) is false (we assume that the probabilities of undercounting and overcounting are the same):

	M=0	M=1	M=2	M=3	M=4
N=1	e/2	1-e	e/2	0	0
N=2	0	e/2	1-e	e/2	0
N=3	0	0	e/2	1-e	e/2

Table for M_1 (resp. M_2) given F_1 (resp. F_2) is true:

	M=0	M=1	M=2	M=3	M=4
N=1	1	0	0	0	0
N=2	1	0	0	0	0
N=3	1	0	0	0	0

b.

	M=0	M=1	M=2	M=3	M=4
N=1	(e/2)(1-f)+f	(1-e)(1-f)	(e/2)(1-f)	0	0
N=2	f	(e/2)(1-f)	(1-e)(1-f)	(e/2)(1-f)	0
N=3	f	0	(e/2)(1-f)	(1-e)(1-f)	(e/2)(1-f)

Problem 8 Solution:

a. Since reward depends only on the current state, we need to have two separate state variables for each current count: one representing a player who has decided to stop, one representing a player who has not yet decided to stop. The states are therefore: 0, 0s, 2, 2s, 3, 3s, 4, 4s, 5, 5s, 6, where the state of "6" applies to any score of 6 or more regardless of whether the player wishes to stop or draw. The actions are $\{$Draw, Stop$\}$.

b. The transition function is:

$$P(Ns|N, Stop) = 1$$

$$P(N'|N, Draw) = \begin{cases} \frac{1}{3} if\ N' - N \in \{2,3,4\} \\ \frac{1}{3} if\ N = 2\ and\ N' = 6 \\ \frac{2}{3} if\ N = 3\ and\ N' = 6 \\ 1\ if\ N \in \{4,5\}\ and\ N' = 6 \\ 0\ otherwise \end{cases}$$

The reward function is:

$$R(Ns) = N$$

$$R(N) = 0$$

c. In general, for finding the optimal policy for an MDP, we would use some method like value iteration followed by policy extraction. However, in this particular case, it is simple to work out that the optimal policy would be

$$Draw\ if\ s \leq 2, Stop\ otherwise$$

For completeness, we give below the value iteration steps based on the states and transition functions described above. The table shows only the states where the player has not yet decided to stop; from any such state, the value iteration considers two possible actions: stop, or draw. The optimal policy is given by taking the $argmax$ instead of max, in the final iteration of value iteration.

V	0	2	3	4	5	6

V1	0	0	0	0	0	0
V2	0	2	3	4	5	0
V3	3	3	3	4	5	0
V4	10/3	3	3	4	5	0

d. The smallest number of rounds: 4. After the first round of value iteration, the "stop" states have learned their value, but any non-stop state still has a value of zero. After the fourth round, the value has converged to its correct value.

Problem 9 Solution:

Unsupervised learning. When the objective is to learn the structure or the property of a given dataset.

Problem 10 Solution:

In this case the input space of all possible examples with their target outputs is:

	0	1	2
2	1	1	1
1	0	0	1
0	0	0	1

Since there is clearly no line that can separate the two classes, this function can not be learned by a linear classifier.

Problem 11 Solution:

a. The first row of the table is correctly classified, therefore the weights are not changed. The second row is incorrectly classified, therefore the weights are updated as W=W+(Y-Y')F=[1,6,4]. Using these weights results in misclassification of the third row, therefore the weights are updated again to [-1,-2,-6]. Using these weights results in misclassification of the fourth row, therefore the weights are updated again to [1,4,2]. These weights correctly classify the fifth row.

b. Yes, a perceptron can learn this function. Any weights such that $w_1=w_2$ and $w_0 = -8\ w_1$ are correct; for example, the weights [-8,1,1].

c. This problem is the arithmetic complement of the XOR problem, therefore it is not linearly separable, and cannot be learned by a perceptron.

Problem 12 Solution: convolutional layers have a lot fewer parameters than fully connected layers. Convolutional neurons have limited receptive fields (i.e., they respect image locality) and their responses are shift-invariant (i.e., the same pattern will produce the same response regardless of image location). Convolution is a very traditional image feature extraction operator and is easy to implement efficiently.

Problem 13 Solution: might include two of the following possibilities

1. Discretize the state space.
2. Design a lower-dimensional set of discrete features to represent the states.
3. Use a parametric approximator (e.g., a neural network) to estimate the Q function values and learn the parameters instead of directly learning the state-action value functions.

Problem 14: Solution

TD learning computes, in each iteration, Q(s,a)=Q(s,a)+alpha(R(s)+max_{a'} Q(s',a')-Q(s,a)). This is an O{MN} table, whose update requires one addition per time step, so the space complexity is O{MN}.
SARSA learning first chooses the action a', then computes Q(s,a)=Q(s,a)+alpha(R(s)+Q(s',a')-Q(s,a)). This has the same space complexity of O{MN}.

Problem 15: Solution

The actor computes P(a|s), the probability that a particular action, a, is the best action to take from state s. The critic computes Q(s,a), the expected long-term discounted reward achieved by taking action a from state s. If we multiply P(a|s)Q(s,a), and add over all possible actions, we get an estimate of the value of state s.

Problem 16: Solution

During the time that she lived in the White House, Malia Obama owned a dog named Bo. The front door of the White House had a dog door, so that Bo could come and go as he pleased. Every Sunday, a Secret Service agent made sure that the dog door was locked. Each weekday, on her way to school, Malia checked the dog door. If it was locked, she unlocked it with probability 1/3. If it was unlocked, she locked it with probability 1/4. On days when the dog door was unlocked, Bo escaped the house with probability 3/4, and went wandering about unleashed on the White House lawn. On days when the dog door was locked, the only way for Bo to escape was by begging to go out for a walk, and then breaking his leash; this happened with probability 1/10.

a. What was the probability that Bo was found unleashed on the White House lawn on any given Sunday?

Solution:
P(escape|Sunday)=1/10.

b. Given that Bo escaped on a Monday, what was the probability that the dog door was open on that day?
Solution:
P(escape, door open)=(1/3)(3/4)=1/4
P(escape, door closed)=(2/3)(1/10)=1/15
P(door open|escape) = (1/4)/(1/4+1/15)=15/19.

c. A new janitor was hired at the White House. He sometimes locked the dog door, and sometimes unlocked it. He also sometimes let Bo escape by accident; at other times, he captured Bo and brought him back inside. As a result of these changes, Malia discovered that her old model of Bo's behavior was out of date, and had to be re-estimated. She observed the following data:

Day	Dog Door	Bo
Sunday	Locked	Outside
Monday	Unlocked	Inside
Tuesday	Unlocked	Outside
Wednesday	Locked	Inside
Thursday	Locked	Inside
Friday	Unlocked	Outside
Saturday	Locked	Inside

Define L_t=1 to be the event "dog door locked on day t," and define O_t=1 to be the event "Bo outside on day t." Estimate the conditional probability tables $P(L_{t+1}|L_t)$ and $P(O_t|L_t)$ from the table above.

Solution:

$P(L_{t+1}=1|L_t=0)=2/3$

$P(L_{t+1}=1|L_t=1)=1/3$

$P(O_t=1|L_t=0)=2/3$

$P(O_t=1|L_t=1)=1/4$

Problem 17 Solution

The j'th softmax output, A_j, is computed from the j'th softmax input, Z_j, in two steps: (1) first, Z_j is exponentiated, (2) second, $\exp(Z_j)$ is divided by the summation of $\exp(Z_k)$ over all possible values of k. Each of these two steps is necessary so that the output, A_j, will satisfy all of the axioms of probability.

a. Which of the three axioms of probability does A_j satisfy only because it is proportional to $\exp(Z_j)$? State the axiom, in words, or give an equation.

Solution: The first axiom: A probability must be non-negative, i.e., $A_j \geq 0$.

b. Which of the three axioms of probability does A_j satisfy only because it is proportional to $1/\sum_k \exp(Z_k)$? State the axiom, in words, or give an equation.

Solution: The second axiom : P(True)=1, and the third axiom : $P(A \vee B) = P(A) + P(B) - P(A \wedge B)$. Partial credit for saying that $1 = \sum_k A_k$, since that is true, but it is not an axiom of probability.

Problem 18

The softmax function is given by

$$A_j = \frac{\exp Z_j}{\sum_k \exp Z_k}$$

Find $d \ln A_5 / dZ_2$.

Solution :

$$\frac{d \ln A_5}{dZ_2} = \frac{1}{A_5} \frac{dA_5}{dZ_2} = -\frac{1}{A_5} \frac{\exp Z_5}{(\sum_k \exp Z_k)^2} \frac{d \exp Z_2}{dZ_2} = -\frac{\exp Z_2}{\sum_k \exp Z_k} = -A_2$$

Problem 19

In an HMM-DNN hybrid, the DNN softmax outputs compute $P(Q_t|E_t)$, where Q_t is the HMM state variable, and E_t is the observed evidence at time t. Before using this probability in the HMM, we first need to divide it by $P(Q_t)$. Why?

Solution :

An HMM needs the likelihood of each state $P(E_t|Q_t)$, not the state posterior $P(Q_t|E_t)$. These two quantities are related as:

$$P(E_t|Q_t) = \frac{P(Q_t|E_t)P(E_t)}{P(Q_t)}$$

So if we compute $P(Q_t|E_t)/P(Q_t)$, that is proportional to $P(E_t|Q_t)$. The constant of proportionality is $P(E_t)$, which doesn't depend on the state variable, so it is irrelevant if our goal is to find the most probable state variable sequence (e.g., for a filtering operation, or a prediction operation).