Name:	**Student ID:**

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

01.112 Machine Learning, Fall 2018
Final Exam
Sample Solutions

Date: 14 Dec 2018
Time: 15:00 - 17:00

Instructions:

1. Write your name and student ID at the top of this page.
2. This paper consists of 4 main questions and 20 printed pages.
3. The problems are not necessarily in order of difficulty. We recommend that you scan through all the questions first, and then decide on the order to answer them.
4. Write your answers in the space provided.
5. You may refer to your one-sided A4-sized cheat sheet.
6. You are allowed to use non-programmable calculators.
7. You may NOT refer to any other material.
8. You may NOT access the Internet.
9. You may NOT communicate via any means with anyone (aside from the invigilators).

For staff's use:

Qs 1	/8
Qs 2	/18
Qs 3	/18
Qs 4	/6
Total	**/50**

Question 1. (8 points)
Please indicate whether the following statements are true (**T**) or false (**F**).

1. We can use the hard EM or the soft EM algorithm to perform unsupervised learning of the HMM. The E-step of the hard EM algorithm will be less expensive than the soft EM algorithm in terms of time complexity. *(1 point)*

   Answer : *F*

2. The Viterbi algorithm is used for decoding. When doing decoding, we assume we know the exact values for both the transition and emission parameters. *(1 point)*

   Answer : *T*

3. When we perform learning in Bayesian networks, if the structure of a Bayesian network is given, we can use the log-likelihood as the criterion for supervised learning based on a collection of samples. *(1 point)*

   Answer : *T*

4. In the Markov decision process, we are interested in learning a policy π, which specifies for each state an action to take. *(1 point)*

   Answer : *T*

5. In a hidden Markov model that we learned in class, the very first variable that is generated is $y_0 = \texttt{START}$, whose Markov blanket only consists of one variable, which is y_1. *(1 point)*

   Answer : *T*

6. The naive Bayes model can be regarded as a special Bayesian network. *(1 point)*

   Answer : *T*

7. For supervised learning where both inputs X and outputs Y are given, the generative model naive Bayes is interested in modeling $P(X, Y)$, while the discriminative model logistic regression is interested in modeling $P(Y|X)$. *(1 point)*

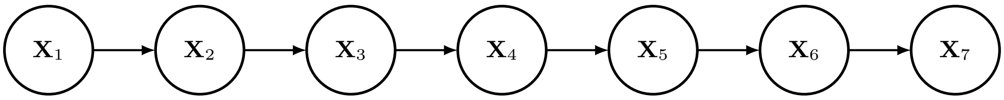
   Answer : *T*

8. Exact inference in a general Bayesian networks is NP-hard. *(1 point)*

   Answer : *T*

Question 2. (18 points)

1. Consider the following Bayesian network with 7 variables.



where the probability table for $\mathbf{X}_1$ and $\mathbf{X}_k$ $(k = 2, 3, \ldots, 7)$ are as follows:

	$\mathbf{X}_1$	
	1	2
	0.5	0.5

	$\mathbf{X}_k$	
$\mathbf{X}_{k-1}$	1	2
1	0.2	0.8
2	0.3	0.7

(a) What is the number of *free parameters* for this Bayesian network? Answer: ___13___

(2 points)

(b) Is $P(\mathbf{X}_3|\mathbf{X}_4) = P(\mathbf{X}_3|\mathbf{X}_5, \mathbf{X}_4)$ always true no matter what values $\mathbf{X}_3, \mathbf{X}_4, \mathbf{X}_5$ take?

Answer (*yes/no*): ___yes___

(2 points)

(c) What about $P(\mathbf{X}_1|\mathbf{X}_7) = P(\mathbf{X}_1|\mathbf{X}_7, \mathbf{X}_4)$? Answer (*yes/no*): ___no___

(2 points)

(d) What about $P(\mathbf{X}_3, \mathbf{X}_5) = P(\mathbf{X}_3)P(\mathbf{X}_5)$? Answer (*yes/no*): ___no___

(2 points)

(e) Calculate $P(\mathbf{X}_2 = 1)$.

(2 points)

$$
\begin{aligned}
&P(\mathbf{X}_2 = 1) \\
&= \textstyle\sum_{x_1} P(\mathbf{X}_1 = x_1, \mathbf{X}_2 = 1) \\
&= P(\mathbf{X}_1 = 1, \mathbf{X}_2 = 1) + P(\mathbf{X}_1 = 2, \mathbf{X}_2 = 1) \\
&= P(\mathbf{X}_1 = 1)P(\mathbf{X}_2 = 1|\mathbf{X}_1 = 1) + P(\mathbf{X}_1 = 2)P(\mathbf{X}_2 = 1|\mathbf{X}_1 = 2) \\
&= 0.5 \times 0.2 + 0.5 \times 0.3 \\
&= 0.25
\end{aligned}
$$

(f) Calculate $P(\mathbf{X}_7 = 1)$. *(4 points)*

Hints: What if you already know $P(\mathbf{X_6} = 1)$ and $P(\mathbf{X_6} = 2)$? Dynamic programming? How did we arrive at the forward algorithm?

$P(\mathbf{X}_k = 1)$
$= \sum_u P(\mathbf{X}_{k-1} = u)P(\mathbf{X}_k = 1|\mathbf{X}_{k-1} = u)$
$= P(\mathbf{X}_{k-1} = 1)P(\mathbf{X}_k = 1|\mathbf{X}_{k-1} = 1) + P(\mathbf{X}_{k-1} = 2)P(\mathbf{X}_k = 1|\mathbf{X}_{k-1} = 2)$
$= 0.2 \times P(\mathbf{X}_{k-1} = 1) + 0.3 \times P(\mathbf{X}_{k-1} = 2)$

$P(\mathbf{X}_k = 2)$
$= \sum_u P(\mathbf{X}_{k-1} = u)P(\mathbf{X}_k = 2|\mathbf{X}_{k-1} = u)$
$= P(\mathbf{X}_{k-1} = 1)P(\mathbf{X}_k = 2|\mathbf{X}_{k-1} = 1) + P(\mathbf{X}_{k-1} = 2)P(\mathbf{X}_k = 2|\mathbf{X}_{k-1} = 2)$
$= 0.8 \times P(\mathbf{X}_{k-1} = 1) + 0.7 \times P(\mathbf{X}_{k-1} = 2)$

Using the above recurrence relations we will be able to calculate $P(\mathbf{X}_7 = 1)$ using a dynamic programming algorithm that runs from the left to the right (which is similar to the idea behind the forward algorithm).

The final answer is 0.2727275.

2. Now you are given a **different** Bayesian network that has 6 variables. You are also told that this Bayesian network comes with the following 6 probability tables:

	$\mathbf{X}_1$ = 1	$\mathbf{X}_1$ = 2
	0.5	0.5

$\mathbf{X}_3$	$\mathbf{X}_2$ = 1	$\mathbf{X}_2$ = 2
1	0.3	0.7
2	0.2	0.8

$\mathbf{X}_1$	$\mathbf{X}_3$ = 1	$\mathbf{X}_3$ = 2
1	0.2	0.8
2	0.3	0.7

$\mathbf{X}_2$	$\mathbf{X}_5$	$\mathbf{X}_4$ = 1	$\mathbf{X}_4$ = 2
1	1	0.1	0.9
1	2	0.2	0.8
2	1	0.2	0.8
2	2	0.3	0.7

$\mathbf{X}_6$	$\mathbf{X}_5$ = 1	$\mathbf{X}_5$ = 2
1	0.1	0.9
2	0.3	0.7

	$\mathbf{X}_6$ = 1	$\mathbf{X}_6$ = 2
	0.1	0.9

Calculate $P(\mathbf{X}_3 = 1|\mathbf{X}_6 = 1)$. *(4 points)*
Hint: Try to find a short answer.

From the graph we can see $\mathbf{X}_3$ and $\mathbf{X}_6$ are independent of each other when no other variable is given.

That means $P(\mathbf{X}_3 = 1|\mathbf{X}_6 = 1) = P(\mathbf{X}_3 = 1) = \sum_{x_1} P(\mathbf{X}_3 = 1, \mathbf{X}_1 = x_1) = 0.5 \times 0.2 + 0.5 \times 0.3 = 0.25$.

Question 3. (18 points)
In this problem, we would like to look at the Hidden Markov Model (HMM).

(a) Assume that we have the following data available for us to estimate the model parameters:

State sequence	Observation sequence
(X, Y, X, X)	(a, a, a, a)
(X, Y, Y)	(b, b, b)
(Z, Z, X, Z, Y)	(c, c, c, c, c)
(Y, Y, Y)	(a, b, a)
(Y, Y, Y)	(a, c, a)

Under the maximum likelihood estimation (MLE), what would be the optimal model parameters? Fill up the following emission and transition probability tables. *(7 points)*

$b_u(o)$ $u \backslash o$	a	b	c
X	3/5	1/5	1/5
Y	1/2	3/10	1/5
Z	0	0	1

$a_{u,v}$ $u \backslash v$	X	Y	Z	STOP
START	2/5	2/5	1/5	0
X	1/5	2/5	1/5	1/5
Y	1/10	1/2	0	2/5
Z	1/3	1/3	1/3	0

(b) Consider a HMM where the transition and emission probabilities are given as follows. Use the Viterbi algorithm discussed in class to find the optimal state sequence for a given input observation sequence (a, b, b). Clearly show the steps that lead to your answer. *(4 points)*

$a_{u,v}$ $u \backslash v$	X	Y	Z	STOP
START	0.2	0.4	0.4	0.0
X	0.2	0.2	0.1	0.5
Y	0.1	0.2	0.2	0.5
Z	0.3	0.5	0.1	0.1

$b_u(o)$ $u \backslash o$	a	b	c
X	0.6	0.1	0.3
Y	0.4	0.1	0.5
Z	0.5	0.4	0.1

Base case:

$$\pi(0, \texttt{START}) = 1$$

Moving forward:

$$\begin{aligned}
\pi(1, X) &= a_{\texttt{START},X} \times b_X(a) = 0.2 \times 0.6 = 0.12 \\
\pi(1, Y) &= a_{\texttt{START},Y} \times b_Y(a) = 0.4 \times 0.4 = 0.16 \\
\pi(1, Z) &= a_{\texttt{START},Z} \times b_Z(a) = 0.4 \times 0.5 = 0.2
\end{aligned}$$

$$\begin{aligned}
\pi(2, X) &= \max_{u \in \mathcal{T}} \{\pi(1, u) \times a_{u,X} \times b_X(b)\} \\
&= \max\{0.12 \times 0.2 \times 0.1,\ 0.16 \times 0.1 \times 0.1,\ 0.2 \times 0.3 \times 0.1\} \quad = 0.006,\ \textit{Best} : Z \\
\pi(2, Y) &= \max_{u \in \mathcal{T}} \{\pi(1, u) \times a_{u,Y} \times b_Y(b)\} \\
&= \max\{0.12 \times 0.2 \times 0.1,\ 0.16 \times 0.2 \times 0.1,\ 0.2 \times 0.5 \times 0.1\} \quad = 0.01,\ \textit{Best} : Z \\
\pi(2, Z) &= \max_{u \in \mathcal{T}} \{\pi(1, u) \times a_{u,Z} \times b_Z(b)\} \\
&= \max\{0.12 \times 0.1 \times 0.4,\ 0.16 \times 0.2 \times 0.4,\ 0.2 \times 0.1 \times 0.4\} \quad = 0.0128,\ \textit{Best} : Y
\end{aligned}$$

$$\begin{aligned}
\pi(3, X) &= \max_{u \in \mathcal{T}} \{\pi(2, u) \times a_{u,X} \times b_X(b)\} \\
&= \max\{0.006 \times 0.2 \times 0.1,\ 0.01 \times 0.1 \times 0.1,\ 0.0128 \times 0.3 \times 0.1\} \quad = 3.84 \times 10^{-4},\ \textit{Best} : Z \\
\pi(3, Y) &= \max_{u \in \mathcal{T}} \{\pi(2, u) \times a_{u,Y} \times b_Y(b)\} \\
&= \max\{0.006 \times 0.2 \times 0.1,\ 0.01 \times 0.2 \times 0.1,\ 0.0128 \times 0.5 \times 0.1\} \quad = 6.4 \times 10^{-4},\ \textit{Best} : Z \\
\pi(3, Z) &= \max_{u \in \mathcal{T}} \{\pi(2, u) \times a_{u,Z} \times b_Z(b)\} \\
&= \max\{0.006 \times 0.1 \times 0.4,\ 0.01 \times 0.2 \times 0.4,\ 0.0128 \times 0.1 \times 0.4\} \quad = 8 \times 10^{-4},\ \textit{Best} : Y
\end{aligned}$$

$$
\begin{aligned}
\pi(4, \texttt{STOP}) &= \max_{u \in \mathcal{T}} \{\pi(3, u) \times a_{u,\texttt{STOP}}\} \\
&= \max\{3.84 \times 10^{-4} \times 0.5, 6.4 \times 10^{-4} \times 0.5, 8 \times 10^{-4} \times 0.1\} \\
&= 3.2 \times 10^{-4}, \qquad Best : Y
\end{aligned}
$$

Backtracking:

$$
\begin{aligned}
y_3^* &= \arg\max_{v \in \mathcal{T}} \{\pi(3, v) \times a_{v,\texttt{STOP}}\} = Y \\
y_2^* &= \arg\max_{v \in \mathcal{T}} \{\pi(2, v) \times a_{v,Y}\} = Z \\
y_1^* &= \arg\max_{v \in \mathcal{T}} \{\pi(1, v) \times a_{v,Z}\} = Y
\end{aligned}
$$

(c) The Viterbi algorithm discussed in class consists of two steps. First it runs from the left to the right (i.e., from position 0 to position $n+1$ for an instance of length n) to calculate the score (π) that should be stored inside each node of the graph that we showed in class. It is then followed by the backtracking stage for recovering the optimal output sequence. It is actually possible to design an alternative dynamic programming algorithm that first runs from the right to the left (i.e., from position $n+1$ to position 0 for an instance of length n) to calculate the scores (under a new definition), followed by the backtracking stage that runs from the left to the right. Describe such a new dynamic programming algorithm clearly using pseudocode (in a way similar to how Viterbi was described in the notes or during class). Use the new algorithm to find the optimal state sequence of the same observation sequence (a, b, b), based on the probability tables from the previous question, which are copied below for your convenience. Show your steps clearly based on the algorithm. *(7 points)*

$a_{u,v}$ $u\backslash v$	X	Y	Z	STOP
START	0.2	0.4	0.4	0.0
X	0.2	0.2	0.1	0.5
Y	0.1	0.2	0.2	0.5
Z	0.3	0.5	0.1	0.1

$b_u(o)$ $u\backslash o$	a	b	c
X	0.6	0.1	0.3
Y	0.4	0.1	0.5
Z	0.5	0.4	0.1

Let $\pi(k, u)$ be the score of a partial sequence starting from state u at position k. And $\mathcal{T}$ is the set of all possible states.

Base case:

$$\pi(n+1, \texttt{STOP}) = 1$$
$$\pi(n, u) = a_{u,\texttt{STOP}} \times b_u(x_n), \forall u \in \mathcal{T}$$

Moving Foward: $\forall u \in \mathcal{T}, j = n-1, ..., 2, 1$

$$\pi(j, u) = \max_{v\in\mathcal{T}}\{\pi(j+1, v) \times a_{u,v} \times b_u(x_j)\}$$

START Case:

$$\pi(0, \texttt{START}) = \max_{v\in\mathcal{T}}\{\pi(1, v) \times a_{\texttt{START},v}\}$$

Backtracking:

At every step, by solving $\arg\max_{v\in\mathcal{T}}$, we can obtain the most likely y_1^*.

$$y_1^* = \arg\max_{v\in\mathcal{T}}\{\pi(1, v) \times a_{\texttt{START},v}\}$$

Similarly, then we can fix the current y_k^* and work backwards to obtain the optimal sequence, by using:

$$y_{j+1}^* = \arg\max_{v\in\mathcal{T}}\{\pi(j, v) \times a_{y_j^*,v}\}.$$

To find most likely sequence for (a,b,b), by apply the proposed algorithm, we can also find the optimal sequence.

Base case:

$$\pi(4, \texttt{STOP}) = 1$$
$$\pi(3, X) = a_{X,\texttt{STOP}} \times b_X(b) = 0.5 \times 0.1 = 0.05$$
$$\pi(3, Y) = a_{Y,\texttt{STOP}} \times b_Y(b) = 0.5 \times 0.1 = 0.05$$
$$\pi(3, Z) = a_{Z,\texttt{STOP}} \times b_Z(b) = 0.1 \times 0.4 = 0.04$$

Foward:

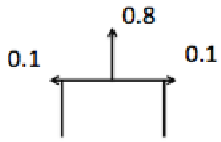
$$\begin{aligned}
\pi(2, X) &= \max_{v \in \mathcal{T}} \{\pi(3, u) \times a_{X,v} \times b_X(b)\} \\
&= \max\{0.05 \times 0.2 \times 0.1, 0.05 \times 0.2 \times 0.1, 0.04 \times 0.1 \times 0.1\} \\
&= 0.001, \quad Best : either X or Y \\
\pi(2, Y) &= \max\{0.05 \times 0.1 \times 0.1, 0.05 \times 0.2 \times 0.1, 0.04 \times 0.2 \times 0.1\} \\
&= 0.001, \quad Best : Y \\
\pi(2, Z) &= \max\{0.05 \times 0.3 \times 0.4, 0.05 \times 0.5 \times 0.4, 0.04 \times 0.1 \times 0.4\} \\
&= 0.01, \quad Best : Y \\
\pi(1, X) &= \max\{0.001 \times 0.2 \times 0.6, 0.001 \times 0.2 \times 0.6, 0.01 \times 0.2 \times 0.4\} \\
&= 6 \times 10^{-4}, \quad Best : Z \\
\pi(1, Y) &= \max\{0.001 \times 0.1 \times 0.4, 0.001 \times 0.2 \times 0.4, 0.01 \times 0.2 \times 0.4\} \\
&= 8 \times 10^{-4}, \quad Best : Z \\
\pi(1, Z) &= \max\{0.001 \times 0.3 \times 0.5, 0.001 \times 0.5 \times 0.5, 0.01 \times 0.1 \times 0.5\} \\
&= 5 \times 10^{-4}, \quad Best : Z
\end{aligned}$$

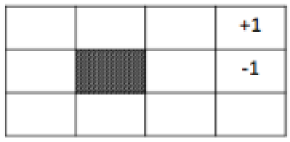
START Case:

$$\begin{aligned}
\pi(0, \texttt{START}) &= \max_{v \in \mathcal{T}} \{\pi(1, v) \times a_{\texttt{START},v}\} \\
&= \max\{6 \times 10^{-4} \times 0.2, 8 \times 10^{-4} \times 0.4, 5 \times 10^{-4} \times 0.4\} \\
&= 3.2 \times 10^{-4}, \quad Best : Y
\end{aligned}$$

By using the backtracking, we can obtain the optimal sequence, Y, Z, Y

Question 4. (6 points)
Recall the grid world example we discussed in class when learning the Markov decision process (MDP).

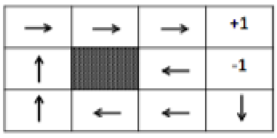


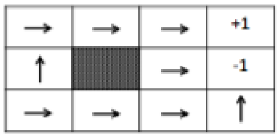


In this example as shown above, each grid is a position/state. For any direction the robot moves towards, there is a 0.8 probability that the robot will successfully arrive at the desired grid (which is the grid in front of the robot), and a 0.1 probability each that it will arrive at the left and right grid respectively. The robot will get bounced back if the destination grid is a wall and will stay at the current grid.

We assume the reward functions are defined as $R(s)$ for each state s. The rewards for the two grids on the upper right corner are already specified (as +1 and -1 respectively, as shown in the picture). When the reward functions for all other states are given, assuming there is a reasonable choice of the discount factor γ that is close to (but less than) 1, we can use algorithms such as value iteration or Q-value iteration introduced in class to figure out an optimal policy.

1. Which of the following two policies is more likely to be the optimal policy when $R(s) = -0.009$ for any other state s? Briefly explain why. *(3 points)*

(a) 

(b) 

Answer:(a/b) __a__.
Explanation:

The reward -0.009 is much better than -1. The robot is willing to stay in such states to move into the +1 state.

2. Which of the following two policies is more likely to be the optimal policy when $R(s) = -1.999$ for any other state s? Briefly explain why. *(3 points)*

(a)

→	→	→	+1
↑		←	-1
↑	←	←	↓

(b)

→	→	→	+1
↑		→	-1
→	→	→	↑

Answer:(*a*/*b*) ___b___.
Explanation:

The reward -1.999 is much worse than -1. The robot is willing to move out of such states sooner than later.