CS5010 8 HOUR TAKE HOME EXAM

Matriculation number: 210016568

December 4, 2021

Special Notes:

For this exam, there may be too many formulas to write, and scanner of my accommodation is quite hard to use, so I choose to use latex to write down my solution.

1 Learning

1.1 Question a

In this case, because this person has no information about the question, the probability of each option is equal.

i.e.,

$$P(A) = \frac{1}{4}, P(B) = \frac{1}{4}, P(C) = \frac{1}{4}, P(D) = \frac{1}{4}$$

In this situation, information entropy should be:

$$-P(A)log_2P(A) - P(B)log_2P(B) - P(C)log_2P(C) - P(D)log_2P(D) = 2\ bits$$

So to solve this question, this person need to get 2 $bits$ information.

1.2 Question b

After getting the information from "Phone a friend", the probability of each option became as following:

$$P(A) = 0.7, P(B) = 0.2, P(C) = 0.05, P(D) = 0.05$$

In this situation, information entropy should be:

$$-P(A)log_2P(A) - P(B)log_2P(B) - P(C)log_2P(C) - P(D)log_2P(D) = 1.257\ bits$$

So this phone call give this person $2\ bits - 1.257\ bits = 0.743\ bits$ information.
The result also mean that to solve this question, this person need to get 1.257 $bits$ information.

1.3 Question c

After getting the information from "lifeline, 50-50", option B and D are dropped, so the left option is A and C. We assume that $P'(A) = 0.7, P'(C) = 0.05$ the probability of each option became as following:

$$P(A) = \frac{P'(A)}{P'(A) + P'(C)} = 0.93$$

$$P(C) = \frac{P'(C)}{P'(A) + P'(C)} = 0.07$$

In this situation, information entropy should be:

$$-P(A)log_2P(A) - P(C)log_2P(C) = 0.366\ bits$$

So "lifeline, 50-50" provided $1.257\ bits - 0.366\ bits = 0.891\ bits$ information based on Question b.
The result also mean that to solve this question, this person need to get 0.366 $bits$ information.

1.4 Question d

For this question, I referred the textbook (AIAMA) from page 727 to 744 (but the exam policy showed I do not need to create a page about reference).

Given:

$$E = -\sum_j t_j \ln o_j, \qquad o_j = \frac{e^{z_j}}{\sum_i e^{z_i}}$$

$$\frac{\partial E}{\partial z_j} = \sum_k (\frac{\partial E_k}{\partial o_k} \cdot \frac{\partial o_k}{\partial z_j})$$

$$Since \qquad \frac{\partial E}{\partial z_j} = \frac{\partial(-t_k \ln o_k)}{\partial o_k} = -\frac{t_k}{o_k}$$

In backpropagation, we need to figure out the derivative of z_j,
if $j = k$:

$$\frac{\partial o_j}{\partial z_j} = \frac{\partial(\frac{e^{z_j}}{\sum_i e^{z_i}})}{\partial z_j} = \frac{\sum_i e^{z_j} e^{z_i} - (e^{z_j})^2}{(\sum_i e^{z_i})^2} = o_j(1 - o_j)$$

if $j \neq k$:

$$\frac{\partial o_j}{\partial z_j} = \frac{\partial(\frac{e^{z_k}}{\sum_i e^{z_i}})}{\partial z_j} = -e^{z_k}(\sum_i e^{z_i})^{-2} e^{z_j} = -a_j a_k$$

$$\begin{aligned}
\frac{\partial E}{\partial z_j} &= \sum_k (\frac{\partial E_k}{\partial o_k} \cdot \frac{\partial o_k}{\partial z_j}) \\
&= \sum_{j \neq k} (\frac{\partial E_k}{\partial o_k} \cdot \frac{\partial o_k}{\partial z_j}) + \sum_{j=k} (\frac{\partial E_k}{\partial o_k} \cdot \frac{\partial o_k}{\partial z_j}) \\
&= \sum_{j \neq k} \frac{t_k}{o_k} o_j o_k + t_j(o_j - 1) \\
&= o_j \sum_k t_k - t_j
\end{aligned}$$

Based on the above expression, we can calculate the gradient of each parameter, and then can use gradient descent algorithm to update the weights.

2 Uncertainty

2.1 Question a

(i) the number of parents of each node is showed in table:

Node	number of parents
Z_0	0
Z_1	1
Z_2	1
Z_3	1
Z_4	1
S_1	0
S_2	0
S_3	0
S_4	0
Y_1	2
Y_2	2
Y_3	2
Y_4	2

Because all nodes are binary, so the we need $2^0 + (2^1) * 4 + (2^0) * 4 + (2^2) * 4 = 29$ parameters.

(ii)

$$\begin{aligned}
& P(Z_0, Z_1, Z_2, Z_3, Z_4, S_1, S_2, S_3, S_4, Y_1, Y_2, Y_3, Y_4) \\
= \; & P(S_1)P(S_2)P(S_3)P(S_4)P(Z_0)P(Z_1|Z_0)P(Z_2|Z_1)P(Z_3|Z_2)P(Z_4|Z_3)P(Y_1|S_1, Z_1)P(Y_2|S_2, Z_2) \\
& P(Y_3|S_3, Z_3)P(Y_4|S_4, Z_4)
\end{aligned}$$

2.2 Question b

(i)

$$markov\_blanket(D) \;\; includes \begin{cases} parents\ of\ D: & A \\ children\ of\ D: & C \\ co-parents\ of\ the\ Children\ of\ D: & B \end{cases}$$

So D's Markov blanket is A, B, C.

(ii)

joint distribution of this net:

$$P(A, B, C, D) = P(A)P(B|A)P(D|B)P(C|A, D)$$

So,

$$\begin{aligned}
P(C|B = false) &= \alpha P(C, B = false) \\
&= \alpha \sum_a \sum_d P(A = a, B = false, C, D = d) \\
&= \alpha \sum_a \sum_d P(A = a)P(B = false|A = a)P(D = d|B = false)P(C|A = a, D = d)
\end{aligned}$$

After calculating, the result is

$$P(C|B = false) = \begin{cases} 0.523 & \text{C=true} \\ 0.477 & \text{C=false} \end{cases}$$

SO $P(C = true|B = false) = 0.523$

2.3 Question c

Scenario Analysis: the r.v.s can be identified as following:

r.v.	discription	domain
COVID	if the person get COVID	true, false
Virus Load	the virus load of the testee	zero, low, medium, high
Collector	the collector of the test	staff, testee
Result of Method A	the result of A method test	positive, negative

The Bayesian Network for this problem is showed in Figure 1.

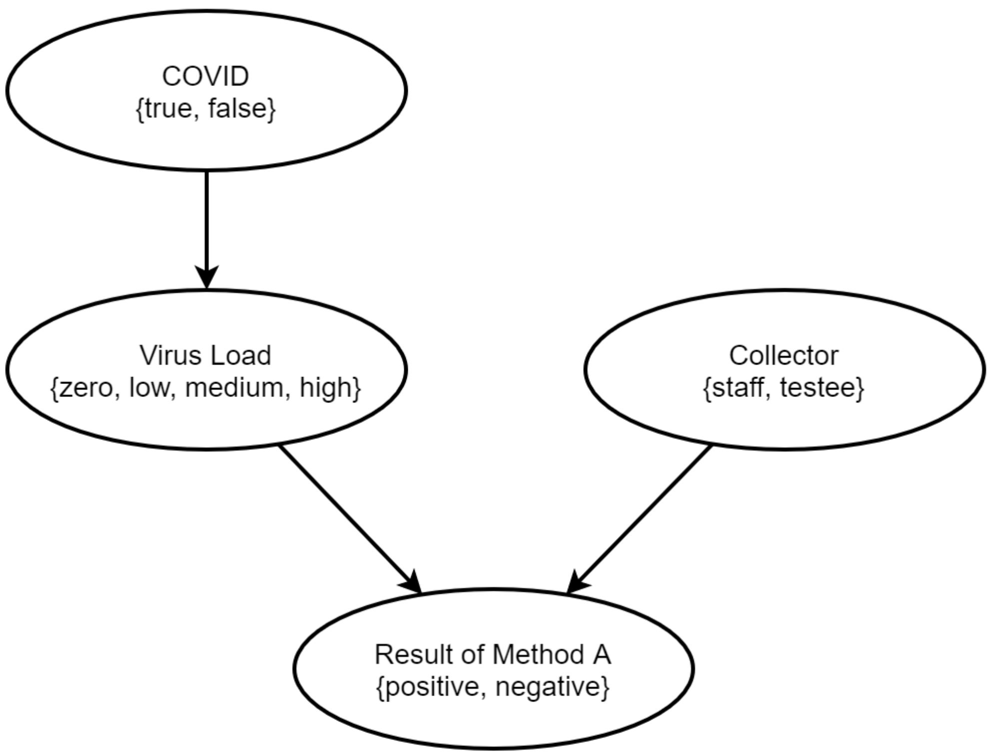

Figure 1: Bayesian Network

the length of domain and the number of parents are listed following:

	length of domain	number of parents
COVID	2	0
Virus Load	4	1
Collector	2	0
Result of Method A	2	2

So the number of parameters for this Bayesian Network is $2^0 + 4^1 + 2^0 + 2^2 = 10$.

2.4 Question d

(i)
To solve this question, r.v.s are needed to be listed:
T_0: switching point; $T_0 \in \{1, 2, \cdots, N-1\}$
P_2: the unknown probabilities of the bent coin; $P_2 \in [0, 1]$. Following the uniform distribution.
D: coin tossing data; $D = Y_1, Y_2, \cdots, Y_N$; $Y_i \in \{0, 1\}$

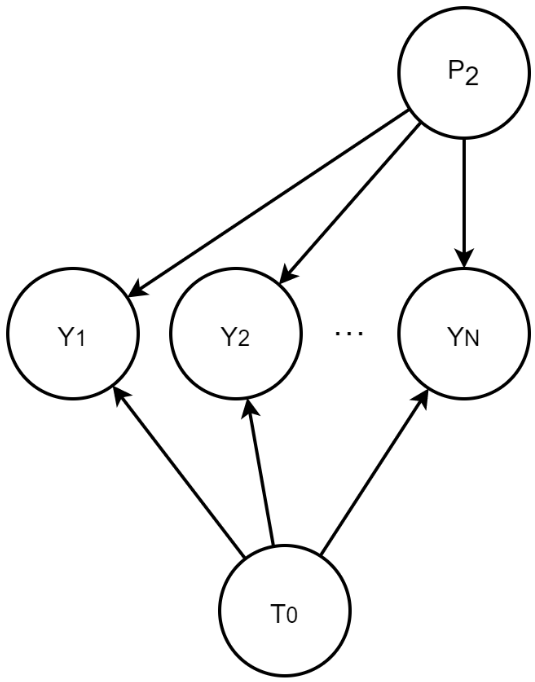

Figure 2: Bayesian Network

For CPTs, it is hard to draw, so I choose it illustrate it instead of drawing it.
For P_2, it follow the uniform distribution, we assume it is discrete, i.e., its domain is {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0}, the probability of taking each value is equal.
For T_0, its distribution is similar as P_2.
For Y_i ($i \in \{1, 2, \cdots, N\}$), when $i < T_0$, the probability of head is 0.5, when $i < T_0$, the probability of head is p_2.

(ii)Gibbs Sampling
The Evidence r.v. is D, Non-Evidence r.v.s are P_2, T_0.

- Step 0
  Initialise some starting values $t_0^{(0)}, p_2^{(0)}$ (generally by random guess)
- Step 1
  Repeat
  for $i = 1, \cdots, m$
  sample $t_0^{(i)} \sim P(T_0 | p_2^{(i-1)}, D)$
  sample $p_2^{(i)} \sim P(P_2 | t_0^{(i-1)}, D)$
  Keep $[t_0^{(i)}, p_2^{(i)}]$ as a sample
- Step 2
  discard the first sample.

3 Searching

3.1 Question a

(i)

Because all the 4 entity must be a side of river, so at most $2^4 = 16$.

But wolf and goat cannot get together, goat and cabbage cannot get together, so we use $||$ represent river, w, g, f, c represent wolf, goat, farmer, cabbage,the following situations are illegal: $wg||fc, fc||wg, gc||fw, fw||gc, f||wgc, wgc||f$. The graphic is showed in Figure 3:

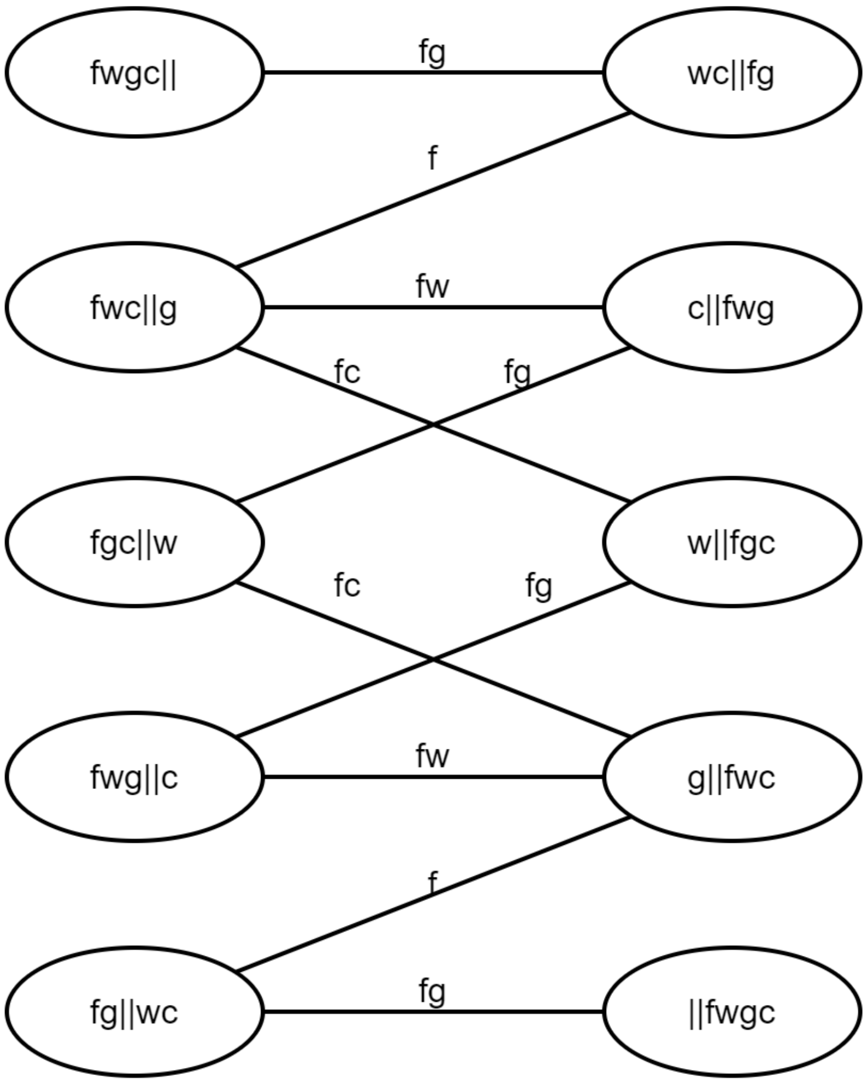

Figure 3: Graphic for this problem

So the size of the state space of this problem is $2^4 - 6 = 10$; the branching factor is 3 (the maximum number of possible edges for a node).

(ii)

The legal situation for this new problem is same as last problem, the only difference is the size of costs in edges can be 3 (e.g., fwc, fgc, $\cdots$). Because the edges of some nodes are too dense to plot cost well, I omitted cost in the figure. The graphic I drawn is showed in Figure 4:

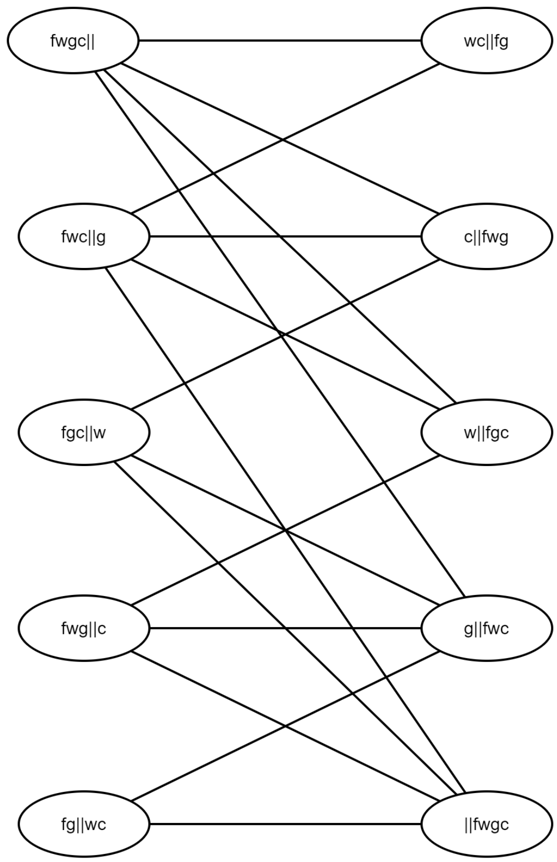

Figure 4: Graphic for this problem

So the the size of the state space of this problem is 10; the branching factor is 4; the minimum number of crossings needed is 3.

3.2 Question b

(i)

search tree is showed in Figure 5.

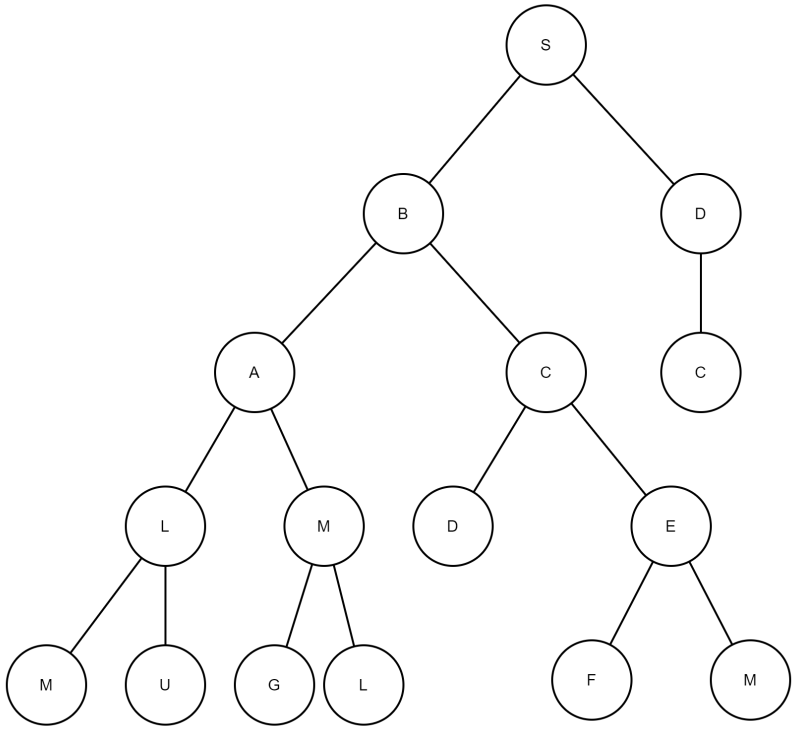

Figure 5: Search tree

The details of extended list are showed following:

step	agenda	extended list
0	[S]	{}
1	[SB, SD]	{S}
2	[SD, SBA, SBC]	{S, B}
3	[SBA, SBC, SDC]	{S, B, D}
4	[SBC, SDC, SBAL, SBAM]	{S, B, D, A}
5	[SDC, SBAL, SBAM, SBCD, SBCE]	{S, B, D, A, C}
6	[SBAL, SBAM, SBCD, SBCE]	{S, B, D, A, C}
7	[SBAM, SBCD, SBCE, SBALM, SBALU]	{S, B, D, A, C, L}
8	[SBCD, SBCE, SBALM, SBALU, SBAMG, SBAML]	{S, B, D, A, C, L, M}
9	[SBCE, SBALM, SBALU, SBAMG, SBAML]	{S, B, D, A, C, L, M}
10	[SBALM, SBALU, SBAMG, SBAML, SBCEF, SBCEM]	{S, B, D, A, C, L, M, E}
11	[SBALU, SBAMG, SBAML, SBCEF, SBCEM]	{S, B, D, A, C, L, M, E}
12	[SBAMG, SBAML, SBCEF, SBCEM]	{S, B, D, A, C, L, M, E}

Finally, the path returned is **SBALU**.

(ii)

The search tree for A* algorithm is showed in Figure 6:

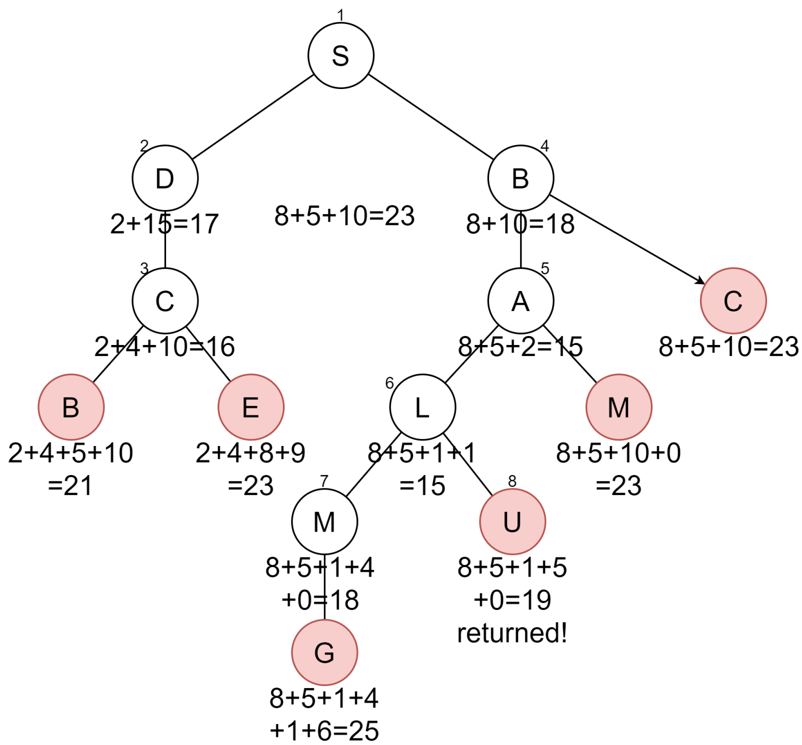

Figure 6: Search tree

The path I found is **SBALU** with 19 estimated cost.

Because is A* search optimal or not depends on the the quality of heuristic, so to find a better path, it should use UCS algorithm to find.

step	agenda	extended list
0	[S]	{}
1	[SB(8), SD(2)]	{S}
2	[SD(2), SBA(13), SBC(13)]	{S, B}
3	[SDC(6), SBA(13), SBC(13)]	{S, B, D}
4	[SDCB(11), SBA(13), SBC(13), SDCE(14)]	{S, B, D, C}
5	[SBA(13), SBC(13), SDCE(14)]	{S, B, D, C}
6	[SBA(13), SBC(13), SDCE(14)]	{S, B, D, C, A}
7	[SBC(13), SBAL(14), SDCE(14), SBAM(23)]	{S, B, D, C, A}
8	[SBAL(14), SDCE(14), SBAM(23)]	{S, B, D, C, A, L}
9	[SDCE(14), SBALM(18), SBALU(19), SBAM(23)]	{S, B, D, C, A, L, E}
10	[SBALM(18), SBALU(19), SDCEM(19), SBAM(23)]	{S, B, D, C, A, L, E, M}
11	[SBALMG(19), SBALU(19), SDCEM(19), SBALME(23), SBAM(23)]	{S, B, D, C, A, L, E, M, G}
12	[SBALU(19), SDCEM(19), SBALME(23), SBAM(23), SBALMGU(25)]	{S, B, D, C, A, L, E, M, G}
13	[SDCEM(19), SBALME(23), SBAM(23), SBALMGU(25)]	{S, B, D, C, A, L, E, M, G}

As the table showed, UCS will return the path **SBALU**, so the path A* search algorithm found is the best.