Artificial Intelligence: Search, CSPs and Logic Bayesian Networks, Probabilistic Inference

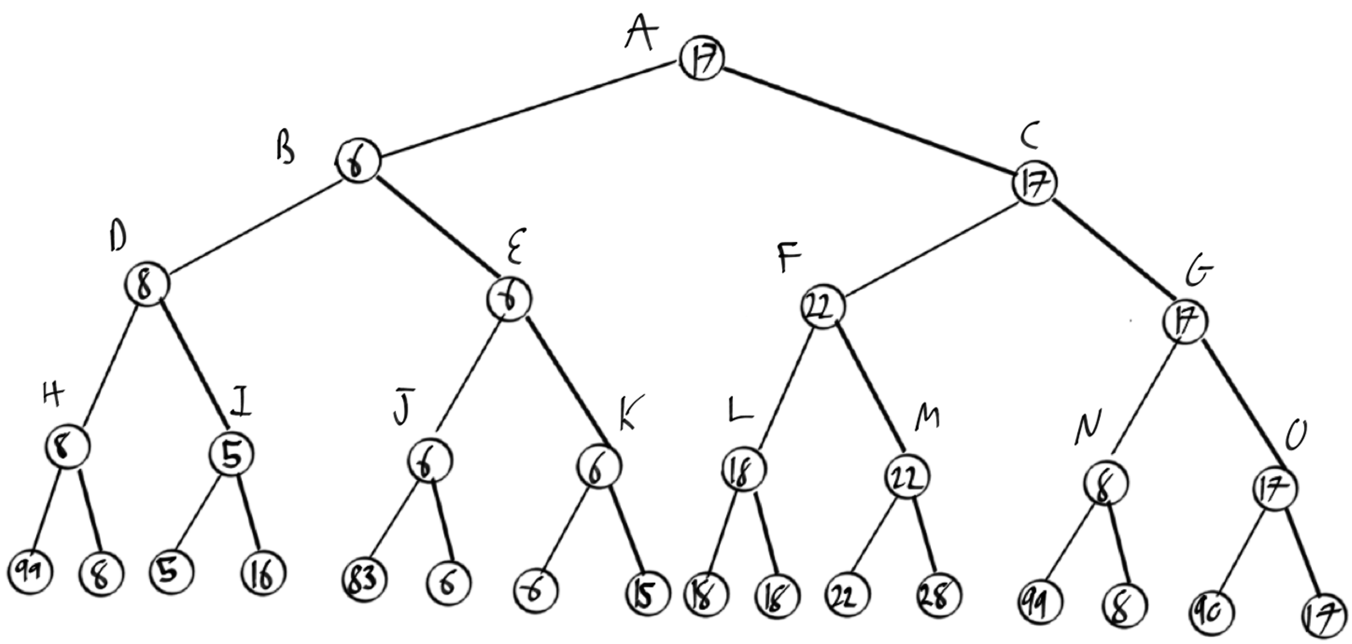
Nathan Morgenstern, Seo Bo Shim

December 8, 2017

Contents

Question 1

part a.



H: 8, I: 5, J: 6, K: 6, L: 18, M: 22, N: 8, O: 17

D: 8, E: 6, F: 22, G: 17

B: 6, C: 17

A: 17

Figure 1: Question 1: Part a

part b.

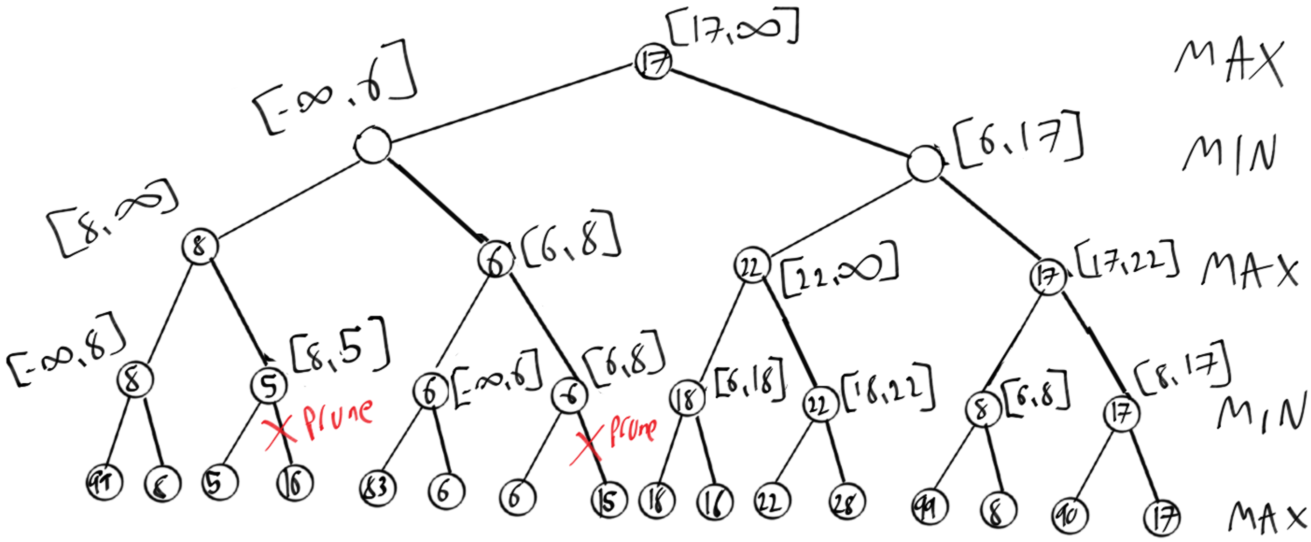


Figure 2: Question 1: Part b

part c.

NOT DONE

part d.

NOT DONE

part e.

NOT DONE

Question 2

part a.

NOT DONE

part b.

NOT DONE

part c.

NOT DONE

part d.

NOT DONE

Question 3

part a.

NOT DON3

part b.

NOT DONE

part c.

NOT DONE

Question 4

part a.

$$(\neg P_1 \vee ... \vee \neg P_m \vee Q)$$

Using DeMorgan's theorem, this term is equivalent to:

$$\neg(P_1 \wedge ... \wedge P_m \wedge \neg Q)$$

Using the Associative property

$$\neg((P_1 \wedge ... \wedge P_m) \wedge \neg Q)$$

Using the conditional equivalence: $\neg(P \rightarrow Q) = (P \wedge \neg Q)$

$$\neg(\neg(P_1 \wedge ... \wedge P_m \rightarrow Q))$$

Finally,

$$(P_1 \wedge ... \wedge P_m \rightarrow Q)$$

part b.

We can repeat the above proof by replacing Q with an expression $(Q_1 \vee ... \vee Q_n)$. A literal can be replaced with an expression.

$$(\neg P_1 \vee ... \vee \neg P_m \vee \neg(Q_1 \vee ... \vee Q_n))$$

Using DeMorgan's theorem, this term is equivalent to:

$$\neg(P_1 \wedge ... \wedge P_m \wedge \neg(Q_1 \vee ... \vee Q_n))$$

Using the Associative property

$$\neg((P_1 \wedge ... \wedge P_m) \wedge \neg(Q_1 \vee ... \vee Q_n))$$

Using the conditional equivalence: $\neg(P \rightarrow Q) = (P \wedge \neg Q)$

$$\neg(\neg((P_1 \wedge ... \wedge P_m) \rightarrow (Q_1 \vee ... \vee Q_n)))$$

Finally,

$$(P_1 \wedge ... \wedge P_m) \rightarrow (Q_1 \vee ... \vee Q_n)$$

part c.

To complete the full resolution and find the resolvent, start with the two clauses:

$$(l_1 \vee ... \vee l_k \vee m_1 \vee ... \vee m_n)$$

We can split this expression up using the associative property:

$$(l_i \vee m_j) \vee (l_1 \vee ... \vee l_{i-1} \vee l_{i+1} \vee ... \vee l_k) \vee (m_1 \vee ... \vee m_{j-1} \vee m_{j+1} \vee ... \vee m_n)$$

Using the Conditional equivalence property, similar to what we derived in part A, this term is equivalent to:

$$\neg(l_i \vee m_j) \rightarrow (l_1 \vee ... \vee l_{i-1} \vee l_{i+1} \vee ... \vee l_k) \vee (m_1 \vee ... \vee m_{j-1} \vee m_{j+1} \vee ... \vee m_n)$$

Because l_i and m_j are complimentary: $(l_i \vee m_j) = True$

$$True \rightarrow (l_1 \vee ... \vee l_{i-1} \vee l_{i+1} \vee ... \vee l_k) \vee (m_1 \vee ... \vee m_{j-1} \vee m_{j+1} \vee ... \vee m_n)$$

Therefore it is inferred that

$$(l_1 \vee ... \vee l_{i-1} \vee l_{i+1} \vee ... \vee l_k \vee m_1 \vee ... \vee m_{j-1} \vee m_{j+1} \vee ... \vee m_n)$$

Question 5

part a.

NOT DON3

part b.

NOT DONE

part c.

NOT DONE

1

Question 6

part a.

B = burglary

E = earthquake

A = alarm

J = John calls

M = Mary calls

$$P(B) = 0.001$$

$$P(E) = 0.002$$

$$P(B|J, M) = \alpha * P(B) * \sum_e P(e) * \sum_a P(a|B, e) * P(J|a) * P(M|a)$$

Compute the sum over A, which results in summing two terms as alarm is either true or false.

$$\sum_a P(a|B, e) * P(J|a) * P(M|a) = P(a|B, E) * P(j|a) * P(m|a) + P(\neg a|B, E) * P(j|\neg a) * P(m|\neg a)$$

$$P(b, e) = .95 * .9 * .7 + .05 * .05 * .01 = .598525$$

For the next summation we will also need the case where the earthquake event does not occur:

$$P(b, \neg e) = .94 * .9 * .7 + .06 * .05 * .01 = .592230$$

Then the rest of the calculations:

$$P(\neg b, e) = .29 * .9 * .7 + .71 * .05 * .01 = .183055$$

$$P(\neg b, \neg e) = .001 * .9 * .7 + .999 * .05 * .01 = .001130$$

Including these calculations into the matrix form:

$$f_6(B, E) = \begin{Bmatrix} P(b, e) & P(b, \neg e) \\ P(\neg b, e) & P(\neg b, \neg e) \end{Bmatrix}$$

$$f_6(B, E) = \begin{Bmatrix} .598525 & .592230 \\ .183055 & .001130 \end{Bmatrix}$$

Then sum over E, whether the event of an earthquake is true or false.

$$P(B|J, M) = \alpha * P(B) * \sum P(e) * f_6(B, e)$$

$$\sum_e P(e) * P(B, e) = P(e) * f_6(B, e) + P(\neg e) * f_6(B, \neg e)$$

$$\begin{Bmatrix} P(b, e) \\ P(\neg b, e) \end{Bmatrix} = .002 * \begin{Bmatrix} .598525 \\ .183055 \end{Bmatrix} + 0.998 * \begin{Bmatrix} .592230 \\ .001130 \end{Bmatrix} = \begin{Bmatrix} .590466 \\ .001494 \end{Bmatrix}$$

$$f_7(B) = \begin{Bmatrix} .590466 \\ .001494 \end{Bmatrix}$$

$$f_7(b) = .590466$$

$$f_7(\neg b) = .001494$$

Finally

$$P(B|J, M) = \alpha * P(B) * f_7(B))$$

$$P(b|J, M) = \alpha * .001 * .590466$$

$$P(\neg b|J, M) = \alpha * .999 * .001494$$

We sum up the two probabilities to find alpha

$$P(b|J,M) + P(\neg b|J,M) = 0.0020830$$

$$\alpha = 1/0.0020830 = 480.$$

$$\boldsymbol{P(b|J,M) = 480 * .001 * .590466 = 0.28347}$$

$$\boldsymbol{P(\neg b|J,M) = 480 * .999 * .001494 = 0.71653}$$

part b.

Operations in Variable Elimination

4*(4 mult 1 add) + 2*(2 mult 1 add) + 2*(2 mult):

+ 1 division and 1 addition for finding the α value:

7 additions

24 multiplications

1 division

32 total operations

Operations in tree enumeration algorithm

For each probability in the final term $P(B|j,m)$ we have three additions, 14 multiplications, and 1 division. The division is from the normalization factor.

There are 8 different cases in $P(B|j,m)$, so in total:

24 additions

112 multiplications

8 divisions

144 total operations

Variable elimination uses much less operations than the tree enumeration algorithm. It uses 22 % of the operations used by the tree enumeration algorithm.

part c.

where n is the number of boolean variables

Using variable elimination: The time and space complexity are dominated by the highest factor constructed. The largest factor is comprised of the number of a children a node in a Bayesian Network may have. If there are n boolean variables, then it is possible for n-1 boolean variables to all have the same parent node. So the largest factor is O(n-1), but this is in fact the worst case scenario. On average it will be less than n as the Bayesian Network will not always be in this configuration.

The space complexity is $0(2^{(n-1)})$ The time complexity is $O(2^{(n-1)})$

the worst case complexity is worse than the tree enumeration, however on average the time complexity is better. There is higher potential space complexity to save states so they do not need to be calculated again (reduce time complexity).

Using enumeration: The space complexity is $O(n)$ as the only requirement is to store each boolean variable The time complexity is $O(2^(n))$, as each combination of the boolean variable must be calculated.

Question 7

part a.

NOT DON3

part b.

NOT DONE

part c.

NOT DONE

Question 8

part a.

Cost of good quality car:

$$C(q^+(c_1)) = \$4000 - (3000) = \$1000. \tag{1}$$

Cost of bad quality car:

$$C(q^-(c_1)) = \$4000 - (3000 + 1400) = -\$400. \tag{2}$$

Assuming that repairs can be made without taking it to the mechanic - perhaps the \$1400 is a separate cost than the \$100 needed to check the quality of the car.

Probability of good quality car:

$$P(q^+(c_1)) = 0.7 \tag{3}$$

Probability of bad quality car:

$$P(q^-(c_1)) = 0.3 \tag{4}$$

Expected net gain:

$$E(c_1) = C(q^+(c_1)) * P(q^+(c_1)) + C(q^-(c_1)) * P(q^-(c_1))$$

$$\mathbf{E(c_1) = \$580} \tag{5}$$

part b.

$$P(Pass|q^+) = 0.8 \tag{6}$$

$$P(Pass|q^-) = 0.35 \tag{7}$$

Using 6 and 7:

$$P(\neg Pass|q^+) = 1 - P(Pass|q^+) = 0.2 \tag{8}$$

$$P(\neg Pass|q^-) = 1 - P(Pass|q^-) = 0.65 \tag{9}$$

Using 3, 4, 6, 7:

$$P(Pass) = P(Pass|q^+) * P(q^+) + P(Pass|q^-) * P(q^-) = 0.665$$

$$\mathbf{P(Pass) = 0.665} \tag{10}$$

Using 10:

$$P(\neg Pass) = 1 - P(Pass) = 0.335$$

$$\mathbf{P(\neg Pass) = 0.335} \tag{11}$$

Using 6, 3, and 10:

$$\mathbf{P(q^+|Pass) = \frac{P(Pass|q^+) * P(q^+)}{P(Pass)} = 0.842} \tag{12}$$

Using 7, 4, and 10:

$$\mathbf{P(q^-|Pass) = \frac{P(Pass|q^-) * P(q^-)}{P(Pass)} = 0.158} \tag{13}$$

Using 8, 3, and 11:

$$\mathbf{P(q^+|\neg Pass) = \frac{P(\neg Pass|q^+) * P(q^+)}{P(\neg Pass)} = 0.418} \tag{14}$$

Using 9, 4, and 11:

$$\mathbf{P(q^-|\neg Pass) = \frac{P(\neg Pass|q^-) * P(q^-)}{P(\neg Pass)} = 0.582} \tag{15}$$

part c.

Paying for the test with the mechanic, the new costs are:

$$C'(q^+(c_1)) = C(q^+(c_1)) - \$100 = \$900$$

$$C'(q^-(c_1)) = C(q^-(c_1)) - \$100 = -\$500$$

Given a pass:

$$E(c_1|Pass) = C'(q^+(c_1)) * P(q^+(c_1)|Pass) + C'(q^-(c_1)) * P(q^-(c_1)|Pass)$$

$$\mathbf{E(c_1|Pass) = \$678.8}$$

Given a failure:

$$E(c_1|\neg Pass) = C'(q^+(c_1)) * P(q^+(c_1)|\neg Pass) + C'(q^-(c_1)) * P(q^-(c_1)|\neg Pass)$$

$$\mathbf{E(c_1|\neg Pass) = \$85.2}$$

Regardless of a pass or a failure, the best decision is the sell the car as there will be a net gain.

part d.

Without the mechanic's test, the expected gain from selling the car will be $\mathbf{E(c_1) = \$580}$

With the test, the expected gain is $\mathbf{\$678.80}$. The value of the optimal information is the difference between the expected gain with the information and the expected gain without the information. The optimal information value is $\mathbf{\$98.80}$. I should take C1 to the mechanic.