Jennifer Shi
20662230 CS486 j77shi@uwaterloo.ca

A3 Write-up

1 The Variable Elimination Algorithm

1.1 Compute some probabilities using VEA

(A) Aria is howling. You walk to the kitchen and see that Aria has not eaten and her food bowl is full. Compute the probability that Aria is sick. Eliminate the hidden variables in lexicographical order (AB, AH, AS, M, NA, NH).

Answer: 0.706 or 0.706759768864287

Computing P(AS | AB and AH)

Define factors f(AB AS) f(AH AS M NH) f(AS) f(M) f(M NA NH) f(NA)

Restrict f(AB AS) to AB = 1 to produce f(AS)

AS, Prob

True,0.8

False,0.2

Restrict f(AH AS M NH) to AH = 1 to produce f(AS M NH)

AS,M,NH, Prob

True,True,True,0.95

True,True,False,0.85

True,False,True,0.7

True,False,False,0.55

False,True,True,0.65

False,True,False,0.3

False,False,True,0.15

False,False,False,0.0

Multiply f(M) f(M NA NH) f(AS M NH) to produce f(AS M NA NH)

AS,M,NA,NH, Prob

True,True,True,True,0.030535714285714285

True,True,False,True,0.010178571428571426

True,True,True,False,0.0030357142857142857

True,True,False,False,0.021249999999999998

True,False,True,True,0.40499999999999999

True,False,False,True,0.0

True,False,True,False,0.21214285714285716

True,False,False,False,0.5303571428571429

False,True,True,True,0.02089285714285714

False,True,False,True,0.006964285714285714

False,True,True,False,0.0010714285714285713

False,True,False,False,0.0074999999999999999

False,False,True,True,0.08678571428571427

False,False,False,True,0.0

False,False,True,False,0.0

False,False,False,False,0.0

Sum out M from f(AS M NA NH) to produce f(AS NA NH)

AS,NA,NH, Prob

True,True,True,0.4355357142857142

True,False,True,0.010178571428571426

True,True,False,0.21517857142857144

True,False,False,0.5516071428571429

False,True,True,0.10767857142857142

False,False,True,0.006964285714285714

False,True,False,0.0010714285714285713

False,False,False,0.0074999999999999999

Multiply f(NA) f(AS NA NH) to produce f(AS NA NH)

AS,NA,NH, Prob

True,True,True,0.17421428571428568

Jennifer Shi
20662230 j77shi@uwaterloo.ca

True,True,False,0.08607142857142858

False,True,True,0.043071428571428566

False,True,False,0.00042857142857142855

True,False,True,0.006107142857142855

True,False,False,0.3309642857142857

False,False,True,0.004178571428571428

False,False,False,0.0044999999999999999

Sum out NA from f(AS NA NH) to produce f(AS NH)

AS,NH, Prob

True,True,0.18032142857142855

True,False,0.4170357142857143

False,True,0.04724999999999999

False,False,0.004928571428571427

Sum out NH from f(AS NH) to produce f(AS)

AS, Prob

True,0.5973571428571428

False,0.05217857142857142

Multiply f(AS) f(AS) f(AS) to produce f(AS)

AS, Prob

True,0.02389428571428572

False,0.00991392857142857

Normalize f(AS) to produce f(AS)

AS, Prob

True,0.706759768864287

False,0.29324023113571296

(B) (Aria is howling and Aria's food bowl is full.) You look out the window and see that the moon is full. You decide to call your neighbour to see if they are home or not. Your neighbour is home and answered the call quickly. Given the new information, compute the probability that Aria is sick. Eliminate the hidden variables in lexicographical order (AB, AH, AS, M, NA, NH).

Answer: 0.313 or 0.3138653588943379

Computing P(AS | AB and AH and M and not NA)

Define factors f(AB AS) f(AH AS M NH) f(AS) f(M) f(M NA NH) f(NA)

Restrict f(AB AS) to AB = 1 to produce f(AS)

AS, Prob

True,0.8

False,0.2

Restrict f(AH AS M NH) to AH = 1 to produce f(AS M NH)

AS,M,NH, Prob

True,True,True,0.95

True,True,False,0.85

True,False,True,0.7

True,False,False,0.55

False,True,True,0.65

False,True,False,0.3

False,False,True,0.15

False,False,False,0.0

Restrict f(AS M NH) to M = 1 to produce f(AS NH)

AS,NH, Prob

True,True,0.95

True,False,0.85

False,True,0.65

False,False,0.3

Restrict f(M) to M = 1 to produce f()

Prob

0.03571428571428571

Restrict f(M NA NH) to M = 1 to produce f(NA NH)

NA,NH, Prob

True,True,0.9

True,False,0.1

False,True,0.3

False,False,0.7

Restrict f(NA NH) to NA = 0 to produce f(NH)

NH, Prob

True,0.3

False,0.7

Restrict f(NA) to NA = 0 to produce f()

Prob

0.6

Multiply f(AS NH) f(NH) to produce f(AS NH)

AS,NH, Prob

True,True,0.285

True,False,0.595

False,True,0.195

False,False,0.21

Sum out NH from f(AS NH) to produce f(AS)

AS, Prob

True,0.8799999999999999

False,0.405

Multiply f(AS) f(AS) f() f() f(AS) to produce f(AS)

AS, Prob

True,0.0007542857142857142

False,0.0016489285714285714

Normalize f(AS) to produce f(AS)

AS, Prob

True,0.3138653588943379

False,0.6861346411056621

A.2 The complexity of VEA

(A) Suppose that we want to compute the prior probability that Aria howls (i.e. *P*(*AH*)). Execute the variable elimination algorithm. Eliminate the hidden variables using the order: **AH, NH, M, AS, AB, NA**.
Write down the size of the largest factor created during the algorithm execution. (All the variables are binary. Therefore, if a factor involves *n* variables, its size is 2*n*.) (If there is a tie, choose the first one.) You do not need to show the steps of the algorithm execution. Justify your answer.

There are six factors initially: f1(AB AS) f2(AH AS M NH) f3(AS) f4(M) f5(M NA NH) f6(NA)

The largest factor created during the algorithm execution is when we multiply all the factors containing NH, trying to sum out NH.

The factors containing NH are f2(AH AS M NH) and f5(M NA NH)

By multiplying the above two factors we get: f7(AH AS M NA NH) = 2^5 = 32

It is the largest factor as it is the factor that contains all the variables except for variable AB. During the execution, we cannot create a factor containing all 6 variables, since there is only one factor contains AB, which is f1(AB AS). All the possible new factors created by VEA, multiplying f1(AB AS) by other factors that contain AS will have at most 5 factors in it. So, the largest factor we can get is f7(AH AS M NA NH) = 2^5 = 32. As NH is the first factor to be eliminated in the VEA execution.

Jennifer Shi
20662230 CS486 j77shi@uwaterloo.ca

(B) In practice, it is challenging to determine the optimal order of eliminating the hidden variables (to minimize the size of the largest factor created during algorithm execution). A common approach is to use a **greedy** strategy, which attempts to **minimize the size of the next factor created** when eliminating each hidden variable during the algorithm execution.
Suppose that we want to compute the prior probability that Aria howls (i.e. $P(AH)$). Execute the variable elimination algorithm. Assume that we use the greedy strategy to determine the order of eliminating the hidden variables. If there is a tie, choose to eliminate the hidden variable that comes first in lexicographical order. For example, if eliminating the variables AB and AH results in a factor of the same size, choose to eliminate AB first.
Write down the order of eliminating the hidden variables based on the greedy strategy.

We are querying about AH.

Step 1:

Factor list: f1(AB AS) f2(AH AS M NH) f3(AS) f4(M) f5(M NA NH) f6(NA)

Hidden Variables: AH, NH, M, AS, AB, NA

Eliminate AH – we cannot eliminate it since it is the query variable

Eliminate NH – the next factor created will have 5 variables

Eliminate M – the next factor created will have 5 variables

Eliminate AS - the next factor created will have 5 variables

Eliminate AB - the next factor created will have 1 variable

Eliminate NA – the next factor created will have 3 variables

We choose to eliminate AB, sum out f1(AB AS) to produce f7(AS)

Order: AB

Step 2:

Factor list: f2(AH AS M NH) f3(AS) f4(M) f5(M NA NH) f6(NA) f7(AS)

Hidden Variables: AH, NH, M, AS, NA

Eliminate AH – we cannot eliminate it since it is the query variable

Eliminate NH – the next factor created will have 5 variables

Eliminate M – the next factor created will have 5 variables

Eliminate AS - the next factor created will have 4 variables

Eliminate NA – the next factor created will have 3 variables

We choose to eliminate NA, multiply f5(M NA NH) to f6(NA) to produce f8(M NA NH), and sum it out to produce f9(M NH)

Order: AB, NA

Step 3:

Factor list: f2(AH AS M NH) f3(AS) f4(M) f7(AS) f9(M NH)

Hidden Variables: AH, NH, M, AS

> **Eliminate AH – we cannot eliminate it since it is the query variable**
>
> **Eliminate NH – the next factor created will have 4 variables**
>
> **Eliminate M – the next factor created will have 4 variables**
>
> **Eliminate AS - the next factor created will have 4 variables**

We choose to eliminate AS, multiply f2(AH AS M NH) f3(AS) f7(AS)to produce f10(AH AS M NH) and sum out AS to produce f11(AH M NH)

Order: AB, NA, AS

Step 4:

Factor list: f4(M) f9(M NH) f11(AH M NH)

Hidden Variables: AH, NH, M

> **Eliminate AH – we cannot eliminate it since it is the query variable**
>
> **Eliminate NH – the next factor created will have 3 variables**
>
> **Eliminate M – the next factor created will have 3 variables**

We choose to eliminate M, multiply f4(M) f9(M NH) f11(AH M NH) to produce f12(AH M NH) and sum out M to produce f13(AH NH)

Order: AB, NA, AS, M

Factor list: f13(AH NH)

Step 5: lastly, we eliminate NH

Order: AB, NA, AS, M, NH

The order of AH in the list doesn't matter, as we are querying for AH.

Final greedy order: AB, NA, AS, M, NH, AH

Jennifer Shi
20662230 CS486 j77shi@uwaterloo.ca

(C) To compute the prior probability that Aria howls, how would you choose an order of eliminating the hidden variables? Why?
You may choose an order from the previous two parts. You may also choose another order that has not been mentioned in the assignment.
State your answer and justify your answer

I would choose the greedy order as the largest factor created will be containing 4 variables. All the other possible orders may create factors containing 5 variables.

Jennifer Shi
20662230 CS486 j77shi@uwaterloo.ca

2 Constructing a Bayesian Network (25 marks + 10 bonus marks)

Consider the following Bayesian network. We can construct the network below by adding the variables to the network using the following order: **A, B, C, D, E**.

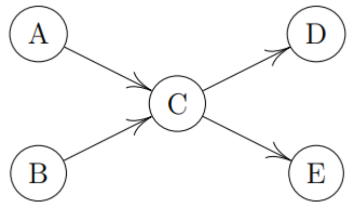
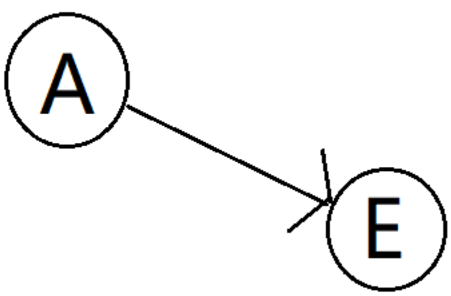


Figure 2: The original Bayesian network

Student ID: 20662230 (even)
Order: A, E, C, B, D

1. Add the first node in the order to the Bayesian network.

2. Add the second node to the network.
Determine a parent set for the second node and justify your choice. Draw the resulting partial Bayesian network.



E is dependent of A, since C is not given. By d-separation, there is a path between A and E, since C is not given.

Jennifer Shi
20662230 CS486 j77shi@uwaterloo.ca

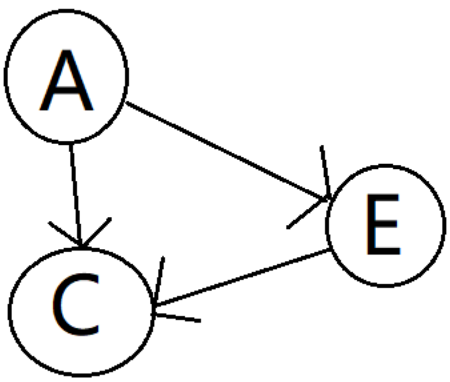
3. Add the third node to the network.
   Determine a parent set for the third node and justify your choice. Draw the resulting partial Bayesian network.

   C is not independent of A and E. Because we can see from figure2 directly C is a child of A and C is the parent of E.

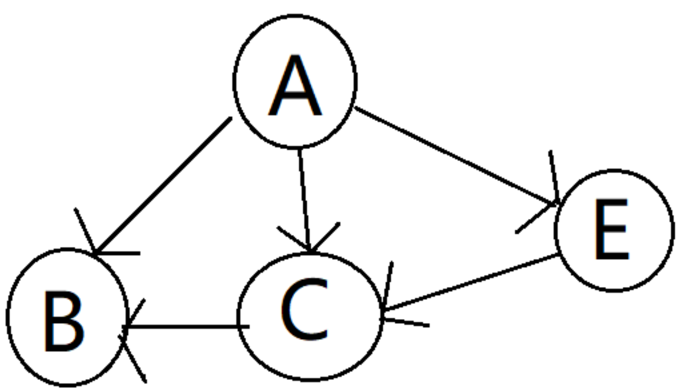
   C is not independent of E given A. By d-separation, there still exists a path between C and E given A. Since C is a parent of E.

   C is not independent of A given E. By d-separation, there still exists a path between C and A given E. Since C is child of A.

   Both nodes A and E should be the parents of C.



4. Add the fourth node to the Bayesian network.
   Determine a parent set for the fourth node and justify your choice. Draw the resulting partial Bayesian network.
   B is dependent of C since B is a parent of C in figure2, so there exists a path.
   Given C, B is independent of E but not independent of A. E is not independent of A given C because C is the descendant of both B and A, and E and A are independent if and only if C and all of C's descendants are NOT observed. So, we will also need to add A to the parent of B.
   Therefore, we choose A and C to be the parents of B



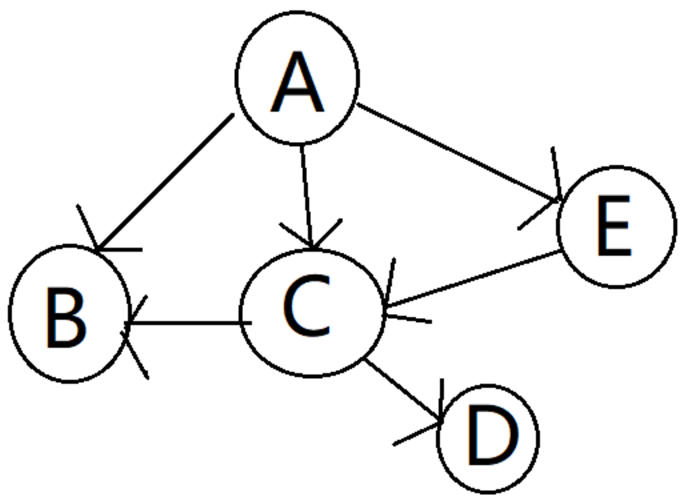
Jennifer Shi
20662230 CS486 j77shi@uwaterloo.ca

5. Add the fifth node to the Bayesian network.
   Determine a parent set for the fifth node and justify your choice. Draw the complete Bayesian network.

   D is dependent of C since C is a direct parent of D, so there exists a path.

   D is not dependent on anyone else given C since C will block the path to every other variable.

   C is the only parent of D.



6. For the following parts, let network A denote the Bayesian network shown in Figure 2, and let network B denote the Bayesian network you constructed in the previous parts of this question.
   To fully define network A, we need a minimum of 10 probabilities.
   • For A, we need 1 probability.
   • For B, we need 1 probability.
   • For C, we need 4 probabilities.
   • For D, we need 2 probabilities.
   • For E, we need 2 probabilities.

What is the minimum number of probabilities required to fully define network B? Justify your answer as shown above.

We need a minimum of 13 probabilities.

• For A, we need 1 probability.
• For B, we need 4 probability.
• For C, we need 4 probabilities.
• For D, we need 2 probabilities.
• For E, we need 2 probabilities.

7. Identify two **conditional or unconditional independence** relationships that are required by network A but not required by network B.
   You should identify an **independence** relationship, not a **dependence** relationship.
   See a valid answer below.
   X and Y are conditionally independent given Z.
   See an invalid answer below.
   X and Y are unconditionally dependent.

   A and B are unconditionally independent in network A.

   A and E are conditionally independent given C.

8. Out of the two networks, which Bayesian network would you prefer? Justify your answer.
   I would prefer network A, since it is a more concise Bayesian network, as fewer probabilities needed to define network A.

9. **Bonus question:** Can you come up with an order of adding the variables to the Bayesian network such that the resulting Bayesian network is full? Being a full network means that every node has all of the previous nodes as its parents. State the order of adding the variables. Explain why the resulting network is full given this order of adding the variables.
   Come up with a simple sufficient condition such that if the order of adding the variables satisfies this sufficient condition, then the resulting network is full. The simpler the condition, the higher you mark will be. Explain why this condition is sufficient.

   Order: D-E-A-B-C

   This order of adding the variables will result in a full Bayesian network because when D and E are added prior to C then they must be dependent. And when A and B are added prior to C and after D and E will make A and B dependent and also dependent to D and E, because we observe C's descendants while not observing C. Adding C last will make it dependent on all other variables, since all other variables are either parent or child of C.

   Condition:

   1) Add D or E before A B and C
   2) Add A or B before C and after D or E
   3) Add the remaining variables before C in arbitrary order
   4) Add C last

   This condition is sufficient for the Bayesian Network for Figure 2 because if we add D/E to the network then it guarantees to be dependent on all other nodes as long as C is not observed, since C is the only path that connects all other variables.

Jennifer Shi
20662230 CS486 j77shi@uwaterloo.ca

We need to add A or B after D or E and prior to C because A and B are unconditionally independent, and in order to make A and B dependent, we need to observe C or some descendants of C. However, we don't want to observe C because by observing C will cut off the paths between A and E, B and D, and D and E.

Adding the remaining variables in arbitrary order will not change the dependence of the variables, since their paths can only be blocked by variable C.

Add C to the network last, it depends on all other variables as there is a direct path to all the variables.