Lecture 13
Variable Elimination Algorithm

Alice Gao

July 4, 2021

Contents

1 Learning Goals

By the end of the lecture, you should be able to

- Explain how we can perform probabilistic inference more efficiently using the variable elimination algorithm.
- Define factors. Manipulate factors using operations restrict, sum out, multiply, and normalize.
- Describe/trace/implement the variable elimination algorithm for calculating a prior or a posterior probability given a Bayesian network.
- Explain how the elimination ordering affects the complexity of the variable elimination algorithm.
- Identify the variables that are irrelevant to a query.

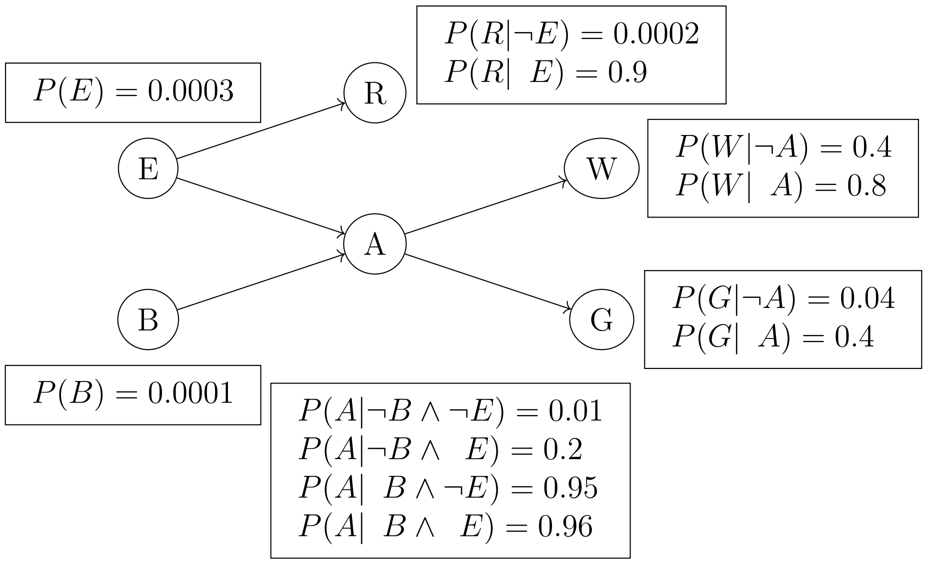
2 Why Use the Variable Elimination Algorithm

One reason for constructing a Bayesian network is to calculate probabilities. There are many probabilities that we may want to calculate. For example, for the Holmes scenario, we may want to calculate the prior probability for the Alarm to go off, without knowing anything else. Also, suppose that one day, both Watson and Gibbon call Mr. Holmes. Given both calls, Mr. Holmes may want to calculate the probability that a Burglary is happening.

In the next section, I will introduce the variable elimination algorithm, which we can use to calculate a probability more efficiently. My goal in this section is to show you why the variable elimination algorithm is more efficient than calculating a probability using the joint distribution.

2.1 A question about the Holmes scenario

Let's consider the Holmes Bayesian network again.



Suppose that we want to answer this question.

> What is the probability that a burglary is happening given that Dr. Watson and Mrs. Gibbon both call?

This seems like a question that Mr. Holmes would want to answer. To answer this question, we need to calculate the probability of Burglary given that Watson and Gibbon are both true.

To write down the expression, I need to introduce some new notation. So far, we've used upper-case letters to represent both the random variable and its values. B means that B=true and $\neg B$ means that B is false. Let's use a new notation to distinguish the random variable and its values. In the new notation, B represents the random variable only. b and $\neg b$ represent its values true and false respectively. Sometimes, we may also use b to represent an undetermined value of the random variable.

Mathematically, we need to calculate the following probability.

$$P(B|w \wedge g)$$

Based on the query, we can divide the variables into three categories.

- B is a **query variable** since we want to calculate its probability (it's on the left-hand side of the bar in the probability above.).
- W and G are **evidence variables** since we observe their values.
- Any other variables (A, E, and R) are **hidden variables** since they do not appear in the probability above.

2.2 Answering the query

Suppose that we want to calculate the conditional probability below.

$$P(B|w \wedge g)$$

To calculate this probability, we need to re-write it using the quantities from the Bayesian network. Whenever we want to calculate a conditional probability, it's always a good idea to convert it into an expression involving joint probabilities only. Let's use the product rule in reverse and write $P(B|w \wedge g)$ as $P(B \wedge w \wedge g)$ over $P(w \wedge g)$.

$$P(B|w \wedge g) = \frac{P(B \wedge w \wedge g)}{P(w \wedge g)} \tag{1}$$

Next, let's re-write the denominator to be in a similar form as the numerator. We'll use the sum rule in reverse and introduce B into the expression. $P(w \wedge g)$ is equal to $P(b \wedge w \wedge g)$ plus $P(\neg b \wedge w \wedge g)$.

$$\frac{P(B \wedge w \wedge g)}{P(w \wedge g)} = \frac{P(B \wedge w \wedge g)}{P(b \wedge w \wedge g) + P(\neg b \wedge w \wedge g)}. \tag{2}$$

Given our derivation so far, calculating $P(B|w \wedge g)$ is equivalent to calculating the joint probability $P(B \wedge w \wedge g)$. This joint probability doesn't contain all the variables, so we cannot calculate it directly using the Bayesian network yet. Let's introduce all the other variables into the expression. Again, we'll use the sum rule in reverse and introduce the hidden variables, E, A, and R. Each term in the sum is a joint probability over all six variables.

$$\begin{align} & P(B \wedge w \wedge g) \tag{3} \\ &= \sum_e \sum_a \sum_r P(B \wedge e \wedge a \wedge w \wedge g \wedge r) \tag{4} \\ &= \sum_e \sum_a \sum_r P(B)P(e)P(r|e)P(a|B \wedge e)P(w|a)P(g|a). \tag{5} \end{align}$$

Now we are ready to use the Bayesian network to calculate the joint probability. The joint probability is a product of six terms. Each term is the conditional probability for one variable in the network.

This expression illustrates how we can calculate the probability by using the joint distribution. First, we'll calculate all the joint probabilities using the Bayesian network. After that, we'll sum out the hidden variables to get the joint probability over B, w and g. Finally, we'll use the product rule in reverse to calculate the conditional probability.

Later on, I want to compare this approach to the variable elimination algorithm. To do this, I need to measure its efficiency. One way to measure its efficiency is to count the number of additions and multiplications we need to perform to calculate the expression.

Problem: How many addition and multiplication operations do we need to perform to evaluate the following expression?

$$\begin{aligned} & P(B \land w \land g) && (6) \\ &= \sum_e \sum_a \sum_r P(B)P(e)P(r|e)P(a|B \land e)P(w|a)P(g|a) && (7) \end{aligned}$$

(A) ≤ 10

(B) 11-20

(C) 21-40

(D) 41-60

(E) ≥ 61

Solution: This question asks you to count the number of operations. The purpose of this question is not to get the number exactly right. That's why the options are ranges rather than individual numbers. Rather, the purpose is to get a rough sense of how many operations are required to calculate this expression. If the number of operations seems large, then the approach is likely not very efficient.

The correct answer is D. We need to perform 47 operations. Let me talk through the process briefly. There are three nested summations over three binary variables. We are summing up eight terms. In each term, we have to perform five multiplications. That's 5 * 8 = 40 multiplications. After that, we need to add the 8 terms using 7 additions. So the total number of operations is 47.

Let me make some observations about this expression. We have to perform many duplicate computations when we calculate this expression.

For example, P(B) appears in every term, but it doesn't involve any of the three hidden variables. You can think of P(B) as a constant. For each of the 8 terms, we need to perform 1 multiplication for P(B). That's 8 multiplications. If we take P(B) outside of all the summations, then we only need to perform 1 multiplication for P(B). This simple change could have saved us from doing 7 operations.

We can make similar observations for E, A and R. Take R as an example. Most terms do not have R in it. Yet these terms are inside the summation over R. So we have to calculate the product of these terms twice, once for R is true and once for R is false. If we take these terms outside of the summation over R, we only need to calculate their product once.

2.3 Answering the query more efficiently

So far, we can calculate the probability by using the joint distribution. Let's find a way to calculate the expression more efficiently. The key idea is that, we only want to perform a summation over terms that involve the variable being summed over. If a term does not involve the variable that we are summing over, we should take that term out of the summation. For our expression, this idea is equivalent to moving the summation to the right as much as possible. These changes will help reduce the number of operations required. Let's try it out.

$$P(B \wedge w \wedge g) \tag{8}$$

$$= \sum_e \sum_a \sum_r P(B)P(e)P(r|e)P(a|B \wedge e)P(w|a)P(g|a) \tag{9}$$

The rightmost summation is over R. The only term involving R is P(r—e). Let's move the summation all the way to the right until it is over P(r—e) only.

$$P(B \wedge w \wedge g) \tag{10}$$

$$= \sum_e \sum_a P(B)P(e)P(a|B \wedge e)P(w|a)P(g|a) \sum_r P(r|e) \tag{11}$$

The next summation is over A. Let's gather all the terms involving A: $P(a|B \wedge e)$, $P(w|a)$, and $P(g|a)$. Put these terms together to the right as much as possible. Next, we'll move the summation to the right until it is only over these three terms.

$$P(B \wedge w \wedge g) \tag{12}$$

$$= \sum_e P(B)P(e) \sum_a P(a|B \wedge e)P(w|a)P(g|a) \sum_r P(r|e) \tag{13}$$

The final summation is over E. The additional term involving E is P(e). We'll move the summation to the right until it is at the left of P(e). This is equivalent to taking P(B) and moving it out of the summation.

$$P(B \wedge w \wedge g) \tag{14}$$

$$= P(B) \sum_e P(e) \sum_a P(a|B \wedge e)P(w|a)P(g|a) \sum_r P(r|e) \tag{15}$$

There is one more simplification we can do. Take a look at the rightmost term: $\sum_r P(r|e)$. What is the value of this term? Think about this for a few seconds. Then, keep reading.

This term is equal to one based on the definition of a conditional probability. So we can remove it from the product. Here is our final expression.

$$P(B \wedge w \wedge g) \tag{16}$$

$$= P(B) \sum_e P(e) \sum_a P(a|B \wedge e)P(w|a)P(g|a) \tag{17}$$

If we use the variable elimination algorithm, we would end up calculating the probability using this expression. Let me ask you the same question as before. What is the number of additions and multiplications required to calculate this expression? Again, the purpose is not to get the number exactly right. This number gives us a rough sense of the algorithm's efficiency. If the variable elimination algorithm is more efficient than the previous approach, we should get a smaller number.

Problem: How many addition and multiplication operations do we need to perform to evaluate the following expression?

$$P(B \wedge w \wedge g) \tag{18}$$

$$= P(B) \sum_e \left(P(e) \sum_a \left(P(a|B \wedge e) P(w|a) P(g|a) \right) \right) \tag{19}$$

(A) ≤ 10

(B) 11-20

(C) 21-40

(D) 41-60

(E) ≥ 61

Solution: The correct answer is B. We need to perform 14 operations. Here is a brief explanation. For the inner sum, we need 1 addition plus 4 multiplications, so 5 operations in total. For the next sum, we need 1 addition plus (1 multiplication + 5 operations) * 2 = 13 operations. Finally, we need one more multiplication. The total is 14 operations.

Again, the exact number is not the point here. Note that by moving the summations inward, we reduced the number of operations by a lot, from 47 to 14.

Let me summarize some important ideas.

We want to calculate a probability. Our first approach is to calculate it using the joint probabilities. We begin by recovering the joint probabilities from the Bayesian network. Next, we will sum out all the hidden variables one by one. This approach works but is quite inefficient since it performs a lot of duplicate computations.

To make it more efficient, the idea is to perform each summation only over terms that involve the variable being summed over. This is the high-level idea that leads to the variable elimination algorithm.

We transformed the expression using this idea. First, we chose an order of the hidden variables. For each hidden variable, we found all the terms involving the hidden variable and multiplied them together. Then, we summed out the hidden variable from the product. This is what we are doing by moving the summation over the hidden variable to the right as much as possible. Repeat this for every other hidden variable until all of them are summed out. This leads to our final expression. Calculating this expression is more efficient since it requires a much smaller number of operations.

3 The Variable Elimination Algorithm

3.1 Introduction

Performing probabilistic inference is challenging. Calculating the posterior distribution of one or more query variables given some evidence is in #NP. Even if we consider the relaxed problem of estimating the posterior probability within some error bound given a Bayesian network, the relaxed problem is already NP-hard. There are no general efficient implementations for probabilistic inference.

However, probabilistic inference is an important task. There are two main approaches to probabilistic inference with Bayesian networks: exact inference and approximate inference. In this course, we will discuss exact inference only. See the textbooks for discussion on approximate inference.

One naive approach to exact inference is to enumerate all the worlds consistent with the evidence in the network. We can do much better by using the variable elimination algorithm.

The variable elimination algorithm uses dynamic programming (storing intermediate calculations) and exploits the conditional independence present in the Bayesian network. At a high level, the variable elimination algorithm first defines factors, then performs operations on these factors (restrict, sum out, multiply, normalize) to make the inference.

3.2 Defining Factors

A factor is a function from a set of random variables to a number. Formally, let f denote a factor and let X_1 to X_j denote the variables in the factor.

A factor is an abstract concept. It can represent a joint probability or a conditional probability. For example, a factor with two variables X_1 and X_2 can represent $P(X_1 \wedge X_2)$ — this is a joint probability. It can also represent $P(X_1|X_2)$ — this is a conditional probability. A third possibility is representing $P(X_1 and X_3 = v_3|X_2)$. For the third probability, although X_3 appears, it is not a variable since we already assigned a value to it. When you are given a factor, the meanings of the values may not be obvious. For instance, if the factor does not represent a joint distribution, the values may not sum to 1.

Defining factors is the first step of the variable elimination algorithm. For each node in the Bayesian network, we need to take the conditional probability distribution and convert it into a factor. For example, for the Holmes network, we need to define 6 factors.

For the variable elimination algorithm, we will **define a factor for every variable/node in the Bayesian network**. The initial factor for each variable/node captures the conditional probability distribution for that variable/node.

Example: Let's do an example. Take the Radio node in the Holmes network. The node represents the conditional probability of Radio given Earthquake.

Let's convert the distribution to a factor involving two variables Radio and Earthquake. With two binary variables, it should have 4 values. Take a look at this factor. The factor doesn't tell us that this is a conditional probability distribution. We can infer it by looking at which pair of probabilities sums up to one.

$f(E, R)$:

E	R	val
t	t	0.9
t	f	0.1
f	t	0.0002
f	f	0.9998

3.3 Restrict

Restrict is the first operation in the algorithm. If we want to calculate a conditional probability, we need to assign the observed values to the evidence variables. This is done by the restrict operation.

Suppose that we have a factor f containing variables from X_1 to X_j. We want to restrict the factor to the case when $X_1 = v_1$. This operation produces a new factor which only contains the variables X_2 to X_j. Since X_1 took a value, it is no longer in the new factor.

The restrict operation has a special case. We can keep performing the restrict operation until all of the variables have values. At this point, the factor has no variable in it and it contains a single number.

In the variable elimination algorithm, the restrict operation is the first one after we define factors. For each evidence variable, we need to find all the factors containing it. In general, the evidence variable can appear in multiple factors. We need to apply the restrict operation to every factor containing the variable.

Example: Let's look at an example. f_1 is a factor with three variables.

$f_1(X, Y, Z)$:

X	Y	Z	val
t	t	t	0.1
t	t	f	0.9
t	f	t	0.2
t	f	f	0.8
f	t	t	0.4
f	t	f	0.6
f	f	t	0.3
f	f	f	0.7

We want to set X to be true. We can do this by keeping only the rows for which X is true and deleting all other rows.

$f_2(Y, Z)$:

Y	Z	val
t	t	0.1
t	f	0.9
f	t	0.2
f	f	0.8

One quick question for you: What is the size of the new factor and how does it compare to the old factor?

Since every variable is binary, restrict causes us to remove half of the rows. The new factor is half the size of the old factor. For this example, we are keeping the first four rows where X is true. This is the new factor f2 with two variables left.

I have two practice questions for you. Restrict f2 to produce f3 by setting Z to be false. Restrict f3 to produce f4 by setting Y to be false. The correct answers are below. Note that f4 contains a single number since it has no variables left.

If we restrict the factor f_2 to $Z = f$, we will create a new factor factor $f_3(Y)$.

$f_3(Y)$:

Y	val
t	0.9
f	0.8

Finally, restricting f_3 to $Y = f$ produces the factor $f_4()$.

$f_4()$: 0.8

3.4 Sum Out

Recall that we divide the variables into three categories: query, evidence, and hidden variables. Since the hidden variables do not appear in the query, we need to eliminate them.

To eliminate a hidden variable, we need to find all the factors containing the variable, multiply the factors together, and sum out the variable from the product. The sum out operation is the last part of the step. Also, the sum out operation is similar to the sum rule of probability.

Suppose that we have a factor f with variables X_1 up to X_j. Suppose that we want to sum out X_1. X1's domain contains k values, v_1 up to v_k.

We need to calculate a summation over all possible values of X1. For each value combination of the other variables, we will sum up all the values in the factor for every value of X1. After this, X1 disappears from the factor and the new factor has variables X_2 to X_j only.

$$(\sum_{X_1} f)(X_2, \ldots, X_j) = f(X_1 = v_1, \ldots, X_j) + \cdots + f(X_1 = v_k, \ldots, X_j).$$

Example: Let's look at an example of sum out. The factor f1 has three variables: X, Y, and Z. What happens if we sum out Y? Since we summed out Y, Y will disappear from the factor. The new factor f2 has X and Z only.

$f_1(X, Y, Z)$:

X	Y	Z	val
t	t	t	0.03
t	t	f	0.07
t	f	t	0.54
t	f	f	0.36
f	t	t	0.06
f	t	f	0.14
f	f	t	0.48
f	f	f	0.32

Which rows should be in the new factor? The sum-out operation is similar to the sum-out rule in probability. We can start by filling in the value combinations of X and Z.

Next, for each combination, go to the old factor and find all the rows containing this combination. We don't care about the value of Y. Since Y is binary, we should find two rows. Add the two values and put it into the new factor.

For example, when X and Z are both true, we have 0.03 and 0.54. The sum is 0.57. Let me do another example. Consider the case when X is false and Z is true. When Y is true, we have 0.06. When Y is false, we have 0.48. The sum is 0.54.

The new factor f_2 is given below.

$f_2(X, Z)$:

X	Z	val
t	t	0.57
t	f	0.43
f	t	0.54
f	f	0.46

After the sum out operation, how did the size of the factor change? It changed in exactly the same way as the restrict operation. Since Y is binary, sum out reduced the factor by half.

3.5 Multiply

Multiply is the most complex operation in the variable elimination algorithm. When we need to eliminate a hidden variable, we need to find all the factors containing the hidden variable, multiply the factors together, and sum out the hidden variable from the product. Multiply is the first part of this step.

Multiplying two factors is non-trivial when the factors do not have the same set of variables. Consider two factors, f1 and f2. They have different variables. Think of X, Y, and Z as groups of variables rather than individual variables. f1 and f2 have some variables in common and this group of variables is denoted by Y.

If we multiply f1 and f2 together, we will produce a new factor which contains all the variables that appear in either factor. The set of variables in the new factor should be the union of the sets of variables in the two factors.

How do we fill in the values in the new factor? Multiply is an element-wise operation. Here is how you can perform it by hand. Pick a value combination for the variables in the new factor. In each factor in the product, find the corresponding value combination for the variables, find the value in the row and put it into the product. This description is quite abstract. It will become clear when I do an example.

The multiply operation has some special cases. For example, f1 and f2 may have the same set of variables. This is an easy case. Another example is that f1 and f2 have no variables in common. We can handle this case as usual by performing element-wise multiplications.

Example:

We have two factors. f1 has variables X and Y. f2 has variables Y and Z. Let's multiply them together.

f_1:

X	Y	val
t	t	0.1
t	f	0.9
f	t	0.2
f	f	0.8

f_2:

Y	Z	val
t	t	0.3
t	f	0.7
f	t	0.6
f	f	0.4

What variables are in the new factor? The new set of variables should be the union of the two sets. It should contain all three variables X, Y, and Z. All three variables are binary, so the new factor has 8 rows. Let me fill in the values of the variables.

Next, let's fill in some values. Suppose that I want to fill in the value for X = true, Y = true, and Z = false. In f1, I need the value for X = true and Y = true. The value is 0.1. In f2, I need the value for Y = true and Z = false. The value is 0.7. Multiply 0.1 and 0.7 together and we get the value 0.07. Put this into the corresponding row in the new factor.

$$(f_1 \times f_2)(X = \text{true}, Y = \text{true}, Z = \text{false}) \tag{20}$$
$$= f_1(X = \text{true}, Y = \text{true}) * f_2(Y = \text{true}, Z = \text{false}) \tag{21}$$
$$= 0.1 * 0.7 \tag{22}$$
$$= 0.07. \tag{23}$$

Let me do another example. Consider the case when X is false, Y is false and Z is true. In f1, we need the value for X is false and Y is false. The value is 0.8. In f2, we need the value for Y is false and Z is true. The value is 0.6. Multiply the two values together and we get 0.48. Put this into the corresponding row in the new factor.

$$
\begin{aligned}
&(f_1 \times f_2)(X = \text{false}, Y = \text{false}, Z = \text{true}) && (24)\\
&= f_1(X = \text{false}, Y = \text{false}) * f_2(Y = \text{false}, Z = \text{true}) && (25)\\
&= 0.8 * 0.6 && (26)\\
&= 0.48. && (27)
\end{aligned}
$$

The new factor $f_1 \times f_2$ is given below.

$f_1 \times f_2$:

X	Y	Z	val
t	t	t	0.03
t	t	f	0.07
t	f	t	0.54
t	f	f	0.36
f	t	t	0.06
f	t	f	0.14
f	f	t	0.48
f	f	f	0.32

Let me summarize the high-level idea again. Pick a row in the new factor. Note the values of the variables. Go to each factor in the product, find the corresponding values of the variables, and find the value. Put the value into the product. Once we calculate the product, put it into the row in the new factor.

3.6 Normalize

I already mentioned the idea of normalization when I discussed the Bayes' rule. The purpose of normalization is to convert some numbers into a valid probability distribution.

Normalize is the last step of the variable elimination algorithm. Before this step, we have one factor left but the values may not sum to 1. After normalizing the values, they will sum to 1 and represent valid probabilities.

Example: Suppose that we want to normalize the factor f1. By normalizing, we will produce a new factor f2. Normalize does not change the size of the factor. So f2 has the same size as f1. Each value in f2 is the original value divided by the sum of all the values in f1. The sum is 0.8. So f2 contains 0.25 and 0.75.

f_1:

Y	val
t	0.2
f	0.6

f_2:

Y	val
t	0.25
f	0.75

As the last step of the variable elimination algorithm, we will normalize the final factor to convert it to a valid conditional probability distribution.

3.7 Putting it together

Let's look at the description of the complete variable elimination algorithm.

Suppose that we want to compute

$$P(X_q | X_{o_1} = v_1 \wedge \ldots \wedge X_{o_j} = v_j).$$

For this query, we have

- one query variable X_q, and
- j evidence variables $X_{o_1}, \ldots, X_{o_j}$.
- The rest of the variables are hidden variables (they do not appear in the query).

The complete variable elimination algorithm is described below.

- **Construct a factor** for each node in the network.
- For each evidence variable and for each factor that contains the evidence variable, **restrict** the factor by assigning the observed value to the evidence variable.
- Eliminate each hidden variable X_{h_j} (based on some order).
  - **Multiply** all the factors that contain X_{h_j}.
  - **Sum out** the variable X_{h_j} from the factor g_j.
- **Multiply** the remaining factors.
- **Normalize** the factor.

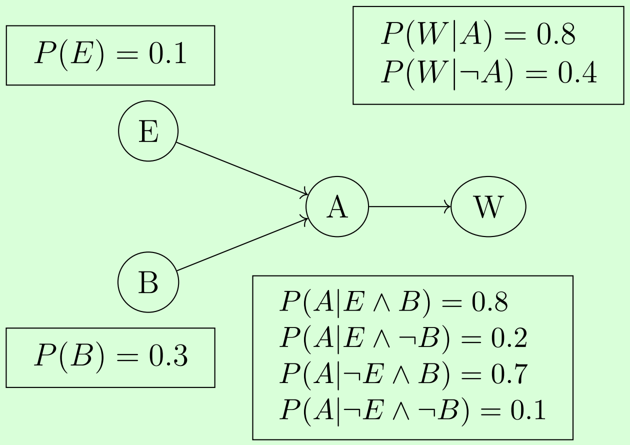
In every step except the first one, we are using one or more factors to make a new factor. We are always creating a new factor instead of modifying existing factors. Every time we use one or more factors to compute a new factor, we will remove all the used factors from the list of remaining factors and add the new factor to the list of remaining factors. The algorithm uses each factor exactly once.

The algorithm is abstract because it needs to work for any general Bayesian network.

3.8 Applying VEA to the Holmes scenario

Let's consider a portion of the Holmes network, consisting of Burglary, Earthquake, Alarm, and Watson. I also modified the probabilities slightly. Our goal is to calculate the conditional probability of Burglary given that the Alarm is not going off.

Problem: Given a portion of the Holmes scenario below, calculate $P(B|\neg A)$ using the variable elimination algorithm.



$P(E) = 0.1$
$P(W|A) = 0.8$
$P(W|\neg A) = 0.4$
E
A
W
B
$P(B) = 0.3$
$P(A|E \wedge B) = 0.8$
$P(A|E \wedge \neg B) = 0.2$
$P(A|\neg E \wedge B) = 0.7$
$P(A|\neg E \wedge \neg B) = 0.1$

Solution: First, let's write down what we need to calculate using probabilities. We will divide up the variables into three types. B is the query variable. A is the evidence variable. E and W are the hidden variables.

To calculate the conditional probability of B given not A, it is sufficient to calculate the joint probability of B and not A.

$$P(B|a) = \frac{P(B \wedge \neg a)}{P(b \wedge \neg a) + P(\neg b \wedge \neg a)}$$

To do this, we can start by calculating the joint probability using the Bayesian network, and then sum out the hidden variables E and W.

$$\begin{aligned}&P(B \wedge \neg a)\\ &= \sum_e \sum_w P(B)P(e)P(\neg a|B \wedge e)P(w|\neg a)\end{aligned}$$

To make this more efficient, we can move the summations to right until they only contain the variables being summed over. The final expression is what we are going to calculate using the variable elimination algorithm.

$$\begin{aligned}&P(B \wedge \neg a)\\ &= P(B)\left(\sum_e P(e)P(\neg a|B \wedge e)\right)\left(\sum_w P(w|\neg a)\right)\end{aligned}$$

Let's execute the variable elimination algorithm.

(1) Define factors.

We will define one factor for each node in the Bayesian network.

$$\begin{array}{cccc} P(B) & P(E) & P(A|B \land E) & P(W|A) \\ f_1(B) & f_2(E) & f_3(A,B,E) & f_4(W,A) \end{array}$$

$f_1(B)$:

B	val
t	0.3
f	0.7

$f_2(E)$:

E	val
t	0.1
f	0.9

$f_3(A,B,E)$:

A	B	E	val
t	t	t	0.8
t	t	f	0.7
t	f	t	0.2
t	f	f	0.1
f	t	t	0.2
f	t	f	0.3
f	f	t	0.8
f	f	f	0.9

$f_4(W,A)$:

W	A	val
t	t	0.8
t	f	0.4
f	t	0.2
f	f	0.6

(2) Restrict factors.

This step assigns the observed values to the evidence variables.

For each evidence variable, find all the factors that contain the variable. For each such factor, restrict the factor such that the evidence variable takes its observed value.

- Restrict $f_3(A,B,E)$ to $\neg a$ to produce $f_5(B,E)$.

  (Intuitively, this step is converting $P(A|B \land E)$ to $P(\neg a|B \land E)$.)

$f_5(B, E)$:

B	E	val
t	t	0.2
t	f	0.3
f	t	0.8
f	f	0.9

- Restrict $f_4(W, A)$ to $\neg a$ to produce $f_6(W)$.

  (Intuitively, this step is converting $P(W|A)$ to $P(W|\neg a)$.)

  $f_6(W)$:

W	val
t	0.4
f	0.6

  Whenever we perform any operation, discard all the factors that we used and create a new factor to store the result. We will use each factor exactly once. After the restrict operations, we have a new list of factors containing f1, f2, f5 and f6.

New factor list: $f_1(B)$, $f_2(E)$, $f_5(B, E)$, $f_6(W)$.

(3) Eliminate the hidden variables since they do not appear in the probability that we want to calculate.

First, we need to choose an order of the hidden variables. For this example, let's eliminate W first and then E.

For each hidden variable, we will do two things. First, find all the factors containing the variable and multiply them together. Second, sum out the hidden variable from the product.

i. In this step, we will eliminate the hidden variable W.

   (Intuitively, this step is calculating $\sum_w P(w|\neg a)$.)

   Multiply all the factors containing W: $f_6(W)$. Only one factor contains W, so we do not have to multiply any factors.

   Sum out W from $f_6(W)$ to get $f_7()$. Note that, W was the only variable in f6, so f7 has no variable and contains a number.

   $f_7()$:

val
1.0

ii. In this step, we will eliminate the hidden variable E.

(Intuitively, this step is calculating $\sum_e P(e)P(\neg a|B \wedge e)$.)

Multiply all the factors containing E: $f_2(E) \times f_5(B,E) = f_8(B,E)$.

$f_8(B,E)$:

B	E	val
t	t	0.02
t	f	0.27
f	t	0.08
f	f	0.81

Sum out E from $f_8(B,E)$ to get $f_9(B)$.

$f_9(B)$:

B	val
t	0.29
f	0.89

New factor list: $f_1(B)$, $f_7()$, $f_9(B)$.

At this point, each remaining factor should contain at most the query variables. Whether we need to perform step 4 depends on the number of factors remaining.

If there are multiple factors left, we need to multiply all the factors together. For our example, we multiply f1, f7, and f9 together to produce a new factor f10. If there is only one factor left, we can skip step 5 and continue to the normalization step.

(4) Multiply all remaining factors.

$$f_1(B) \times f_7() \times f_9(B) = f_{10}(B).$$

$f_{10}(B)$:

B	val
t	0.087
f	0.623

New factor list: $f_{10}(B)$.

(5) Normalize $f_{10}(B)$ to get $f_{11}(B)$.

We will normalize the final factor so that it represents a probability distribution. Normalizing the factor produces a new factor f11. The final factor is a distribution containing two probabilities P(B is true given A is false) and P(B is false given A is false).

$f_{11}(B)$:

B	val
t	0.123
f	0.877

The final factor is $f_{11}(B)$. Thus, $P(B = \text{t}|\neg a) = 0.123$ and $P(B = \text{f}|\neg a) = 0.877$.