Reasoning Under Uncertainty: Variable Elimination Example

CPSC 322 – Uncertainty 7

Textbook §10.5
Lecture Overview

1. Recap
2. Variable Elimination
3. Variable Elimination Example
Factors and Assigning Variables

- A **factor** is a representation of a function from a tuple of random variables into a number.
- A factor denotes a distribution over the given tuple of variables in some (unspecified) context.

We can make new factors out of an existing factor.

Our first operation: assign some or all of the variables of a factor.

- \( f(X_1 = v_1, X_2, \ldots, X_j), \) where \( v_1 \in \text{dom}(X_1) \), is a factor on \( X_2, \ldots, X_j \).
- \( f(X_1 = v_1, X_2 = v_2, \ldots, X_j = v_j) \) is a number that is the value of \( f \) when each \( X_i \) has value \( v_i \).

The former is also written as

\[
f(X_1, X_2, \ldots, X_j)_{X_1 = v_1, \ldots, X_j = v_j}\]
Our second operation: we can sum out a variable, say $X_1$ with domain $\{v_1, \ldots, v_k\}$, from factor $f(X_1, \ldots, X_j)$, resulting in a factor on $X_2, \ldots, X_j$ defined by:

$$\left(\sum_{X_1} f\right)(X_2, \ldots, X_j)$$

$$= f(X_1 = v_1, \ldots, X_j) + \cdots + f(X_1 = v_k, \ldots, X_j)$$
Recap

Variable Elimination

Variable Elimination Example

Multiplying factors

- Our third operation: factors can be multiplied together.
- The product of factor $f_1(X, Y)$ and $f_2(Y, Z)$, where $Y$ are the variables in common, is the factor $(f_1 \times f_2)(X, Y, Z)$ defined by:

\[
(f_1 \times f_2)(X, Y, Z) = f_1(X, Y) f_2(Y, Z).
\]
Probability of a conjunction

- Suppose the variables of the belief network are $X_1, \ldots, X_n$.
- What we want to compute: the factor
  
  $$P(X_q, X_{o_1} = v_1, \ldots, X_{o_j} = v_j)$$

- We can compute $P(X_q, X_{o_1} = v_1, \ldots, X_{o_j} = v_j)$ by summing out the variables
  
  $$X_{s_1}, \ldots, X_{s_k} = \{X_1, \ldots, X_n\} \setminus \{X_q, X_{o_1}, \ldots, X_{o_j}\}.$$ 

- We sum out these variables one at a time
  
  - the order in which we do this is called our elimination ordering.

$$P(X_q, X_{o_1} = v_1, \ldots, X_{o_j} = v_j) = \sum_{X_{s_k}} \cdots \sum_{X_{s_1}} P(X_1, \ldots, X_n | X_{o_1} = v_1, \ldots, X_{o_j} = v_j).$$
Probability of a conjunction

- What we know: the factors $P(X_i | pX_i)$.
- Using the chain rule and the definition of a belief network, we can write $P(X_1, \ldots, X_n)$ as $\prod_{i=1}^{n} P(X_i | pX_i)$. Thus:

\[
P(X_q, X_{o_1} = v_1, \ldots, X_{o_j} = v_j) = \sum_{X_{s_k}} \cdots \sum_{X_{s_1}} P(X_1, \ldots, X_n)_{X_{o_1} = v_1, \ldots, X_{o_j} = v_j}.
\]

\[
= \sum_{X_{s_k}} \cdots \sum_{X_{s_1}} \prod_{i=1}^{n} P(X_i | pX_i)_{X_{o_1} = v_1, \ldots, X_{o_j} = v_j}.
\]
Lecture Overview

1 Recap

2 Variable Elimination

3 Variable Elimination Example
Computing sums of products

Computation in belief networks thus reduces to computing the sums of products.

- It takes 14 multiplications or additions to evaluate the expression $ab + ac + ad + aeh + afh + agh$. How can this expression be evaluated more efficiently?
Computing sums of products

Computation in belief networks thus reduces to computing the sums of products.

- It takes 14 multiplications or additions to evaluate the expression $ab + ac + ad + aeh + afh + agh$. How can this expression be evaluated more efficiently?
  - factor out the $a$ and then the $h$ giving $a(b + c + d + h(e + f + g))$
  - this takes only 7 multiplications or additions
Computing sums of products

Computation in belief networks thus reduces to computing the sums of products.

- It takes 14 multiplications or additions to evaluate the expression $ab + ac + ad + aeh + afh + agh$. How can this expression be evaluated more efficiently?
  - factor out the $a$ and then the $h$ giving $a(b + c + d + h(e + f + g))$
  - this takes only 7 multiplications or additions

- How can we compute $\sum_{X_{s1}} \prod_{i=1}^{n} P(X_i|pX_i)$ efficiently?
Computing sums of products

Computation in belief networks thus reduces to computing the sums of products.

- It takes 14 multiplications or additions to evaluate the expression \(ab + ac + ad + aeh + afh + agh\). How can this expression be evaluated more efficiently?
  - factor out the \(a\) and then the \(h\) giving \(a(b + c + d + h(e + f + g))\)
  - this takes only 7 multiplications or additions

- How can we compute \(\sum_{X_s} \prod_{i=1}^{n} P(X_i | pX_i)\) efficiently?

- Factor out those terms that don’t involve \(X_s\):
  \[
  \left( \prod_{i | X_s \not\in \{X_i\} \cup pX_i} P(X_i | pX_i) \right) \left( \sum_{X_s} \prod_{i | X_s \in \{X_i\} \cup pX_i} P(X_i | pX_i) \right)
  \]
  (terms that do not involve \(X_s\))
  (terms that involve \(X_s\))
Summing out a variable efficiently

To sum out a variable $X_{s_j}$ from a product $f_1, \ldots, f_k$ of factors:

- Partition the factors into
  - those that don’t contain $X_{s_j}$, say $f_1, \ldots, f_i$,
  - those that contain $X_{s_j}$, say $f_{i+1}, \ldots, f_k$

We know:

\[
\sum_{X_{s_j}} f_1 \times \cdots \times f_k = (f_1 \times \cdots \times f_i) \left( \sum_{X_{s_j}} f_{i+1} \times \cdots \times f_k \right).
\]

- $\left( \sum_{X_{s_j}} f_{i+1} \times \cdots \times f_k \right)$ is a new factor; let’s call it $f'$.
- Now we have $\sum_{X_{s_j}} f_1 \times \cdots \times f_k = f_1 \times \cdots \times f_i \times f'$.
- Store $f'$ explicitly, and discard $f_{i+1}, \ldots, f_k$.
  - Now we’ve summed out $X_{s_j}$.
Variable elimination algorithm

To compute $P(X_q | X_{o1} = v_1 \land \ldots \land X_{oj} = v_j)$:

- **Construct a factor** for each conditional probability.
- **Set the observed variables** to their observed values.
- **For each of the other variables** $X_{s_i} \in \{X_{s1}, \ldots, X_{sk}\}$, sum out $X_{s_i}$
- **Multiply** the remaining factors.
- **Normalize** by dividing the resulting factor $f(X_q)$ by $\sum_{X_q} f(X_q)$. 
Lecture Overview

1. Recap
2. Variable Elimination
3. Variable Elimination Example
Variable elimination example

Compute $P(G|H = h_1)$. Elimination order: $A, C, E, H, I, B, D, F$

- $P(G, H) = \sum_{A,B,C,D,E,F,I} P(A, B, C, D, E, F, G, H, I)$
- $P(G, H) = \sum_{A,B,C,D,E,F,I} P(A) \cdot P(B|A) \cdot P(C) \cdot P(D|B,C) \cdot P(E|C) \cdot P(F|D) \cdot P(G|F,E) \cdot P(H|G) \cdot P(I|G)$
Recap Variable Elimination

Variable elimination example

Compute $P(G|H = h_1)$. Elimination order: $A, C, E, H, I, B, D, F$

- $P(G, H) = \sum_{A,B,C,D,E,F,I} P(A) \cdot P(B|A) \cdot P(C) \cdot P(D|B,C) \cdot P(E|C) \cdot P(F|D) \cdot P(G|F,E) \cdot P(H|G) \cdot P(I|G)$

- Eliminate $A$: $P(G, H) = \sum_{B,C,D,E,F,I} f_1(B) \cdot P(C) \cdot P(D|B,C) \cdot P(E|C) \cdot P(F|D) \cdot P(G|F,E) \cdot P(H|G) \cdot P(I|G)$

- $f_1(B) := \sum_{a \in \text{dom}(A)} P(A = a) \cdot P(B|A = a)$
Variable elimination example

Compute $P(G|H = h_1)$. Elimination order: $A, C, E, H, I, B, D, F$

- $P(G, H) = \sum_{B,C,D,E,F,I} f_1(B) \cdot P(C) \cdot P(D|B,C) \cdot P(E|C) \cdot P(F|D) \cdot P(G|F,E) \cdot P(H|G) \cdot P(I|G)$

- Eliminate $C$: $P(G, H) = \sum_{B,D,E,F,I} f_1(B) \cdot f_2(B, D, E) \cdot P(F|D) \cdot P(G|F,E) \cdot P(H|G) \cdot P(I|G)$

- $f_1(B) := \sum_{a \in \text{dom}(A)} P(A = a) \cdot P(B|A = a)$

- $f_2(B, D, E) := \sum_{c \in \text{dom}(C)} P(C = c) \cdot P(D|B, C = c) \cdot P(E|C = c)$
Variable elimination example

Compute $P(G|H = h_1)$. Elimination order: $A, C, E, H, I, B, D, F$

- $P(G, H) = \sum_{B, D, E, F, I} f_1(B) \cdot f_2(B, D, E) \cdot P(F|D) \cdot P(G|F, E) \cdot P(H|G) \cdot P(I|G)$
- Eliminate $E$:
  
  $P(G, H) = \sum_{B, D, F, I} f_1(B) \cdot f_3(B, D, F, G) \cdot P(F|D) \cdot P(H|G) \cdot P(I|G)$

- $f_1(B) := \sum_{a \in \text{dom}(A)} P(A = a) \cdot P(B|A = a)$
- $f_2(B, D, E) := \sum_{c \in \text{dom}(C)} P(C = c) \cdot P(D|B, C = c) \cdot P(E|C = c)$
- $f_3(B, D, F, G) := \sum_{e \in \text{dom}(E)} f_2(B, D, E = e) \cdot P(G|F, E = e)$
Compute $P(G|H = h_1)$. Elimination order: $A, C, E, H, I, B, D, F$

- $P(G, H) = \sum_{B,D,F,I} f_1(B) \cdot f_3(B, D, F, G) \cdot P(F|D) \cdot P(H|G) \cdot P(I|G)$

- Observe $H = h_1$:
  $$P(G, H = h_1) = \sum_{B,D,F,I} f_1(B) \cdot f_3(B, D, F, G) \cdot P(F|D) \cdot f_4(G) \cdot P(I|G)$$

- $f_1(B) := \sum_{a \in \text{dom}(A)} P(A = a) \cdot P(B|A = a)$
- $f_2(B, D, E) := \sum_{c \in \text{dom}(C)} P(C = c) \cdot P(D|B, C = c) \cdot P(E|C = c)$
- $f_3(B, D, F, G) := \sum_{e \in \text{dom}(E)} f_2(B, D, E = e) \cdot P(G|F, E = e)$
- $f_4(G) := P(H = h_1|G)$
Variable elimination example

Compute $P(G|H = h_1)$. Elimination order: $A, C, E, H, I, B, D, F$

- $P(G, H = h_1) = \sum_{B,D,F,I} f_1(B) \cdot f_3(B, D, F, G) \cdot P(F|D) \cdot f_4(G) \cdot P(I|G)$
- Eliminate $I$:
  
  $P(G, H = h_1) = \sum_{B,D,F} f_1(B) \cdot f_3(B, D, F, G) \cdot P(F|D) \cdot f_4(G) \cdot f_5(G)$

\[f_1(B) := \sum_{a \in \text{dom}(A)} P(A = a) \cdot P(B|A = a)\]
\[f_2(B, D, E) := \sum_{c \in \text{dom}(C)} P(C = c) \cdot P(D|B, C = c) \cdot P(E|C = c)\]
\[f_3(B, D, F, G) := \sum_{e \in \text{dom}(E)} f_2(B, D, E = e) \cdot P(G|F, E = e)\]
\[f_4(G) := P(H = h_1|G)\]
\[f_5(G) := \sum_{i \in \text{dom}(I)} P(I = i|G)\]
Compute $P(G|H = h_1)$. Elimination order: $A, C, E, H, I, B, D, F$

- $P(G, H = h_1) = \sum_{B,D,F} f_1(B) \cdot f_3(B, D, F, G) \cdot P(F|D) \cdot f_4(G) \cdot f_5(G)$

- Eliminate $B$:
  \[
P(G, H = h_1) = \sum_{D,F} f_6(D, F, G) \cdot P(F|D) \cdot f_4(G) \cdot f_5(G)
  \]

- $f_1(B) := \sum_{a \in \text{dom}(A)} P(A = a) \cdot P(B|A = a)$
- $f_2(B, D, E) := \sum_{c \in \text{dom}(C)} P(C = c) \cdot P(D|B, C = c) \cdot P(E|C = c)$
- $f_3(B, D, F, G) := \sum_{e \in \text{dom}(E)} f_2(B, D, E = e) \cdot P(G|F, E = e)$
- $f_4(G) := P(H = h_1|G)$
- $f_5(G) := \sum_{i \in \text{dom}(I)} P(I = i|G)$
- $f_6(D, F, G) := \sum_{b \in \text{dom}(B)} f_1(B = b) \cdot f_3(B = b, D, F, G)$
Variable elimination example

Compute \( P(G|H = h_1) \). Elimination order: \( A, C, E, H, I, B, D, F \)

- \( P(G, H = h_1) = \sum_{D,F} f_6(D, F, G) \cdot P(F|D) \cdot f_4(G) \cdot f_5(G) \)
- **Eliminate** \( D \): \( P(G, H = h_1) = \sum_{F} f_7(F, G) \cdot f_4(G) \cdot f_5(G) \)

**f_1(B) := \sum_{a \in \text{dom}(A)} P(A = a) \cdot P(B|A = a)\)**

**f_2(B, D, E) := \sum_{c \in \text{dom}(C)} P(C = c) \cdot P(D|B, C = c) \cdot P(E|C = c)\)**

**f_3(B, D, F, G) := \sum_{e \in \text{dom}(E)} f_2(B, D, E = e) \cdot P(G|F, E = e)\)**

**f_4(G') := P(H = h_1|G)\)**

**f_5(G') := \sum_{i \in \text{dom}(I)} P(I = i|G)\)**

**f_6(D, F, G) := \sum_{b \in \text{dom}(B)} f_1(B = b) \cdot f_3(B = b, D, F, G)\)**

**f_7(F, G') := \sum_{d \in \text{dom}(D)} f_6(D = d, F, G) \cdot P(F|D = d)\)**
Variable elimination example

Compute $P(G|H = h_1)$. Elimination order: $A, C, E, H, I, B, D, F$

- $P(G, H = h_1) = \sum_{F} f_7(F, G) \cdot f_4(G') \cdot f_5(G')$
- Eliminate $F$: $P(G, H = h_1) = f_8(G') \cdot f_4(G') \cdot f_5(G')$

- $f_1(B) := \sum_{a \in \text{dom}(A)} P(A = a) \cdot P(B|A = a)$
- $f_2(B, D, E) := \sum_{c \in \text{dom}(C)} P(C = c) \cdot P(D|B, C = c) \cdot P(E|C = c)$
- $f_3(B, D, F, G) := \sum_{e \in \text{dom}(E)} f_2(B, D, E = e) \cdot P(G|F, E = e)$
- $f_4(G') := P(H = h_1|G)$
- $f_5(G') := \sum_{i \in \text{dom}(I)} P(I = i|G)$
- $f_6(D, F, G) := \sum_{b \in \text{dom}(B)} f_1(B = b) \cdot f_3(B = b, D, F, G')$
- $f_7(F, G') := \sum_{d \in \text{dom}(D)} f_6(D = d, F, G') \cdot P(F|D = d)$
- $f_8(G') := \sum_{f \in \text{dom}(F)} f_7(F = f, G')$
Variable elimination example

Compute \( P(G|H = h_1) \). Elimination order: \( A, C, E, H, I, B, D, F \)

- \( P(G, H = h_1) = f_8(G) \cdot f_4(G) \cdot f_5(G) \)
- Normalize: \( P(G|H = h_1) = \frac{P(G, H = h_1)}{\sum_{g \in \text{dom}(G)} P(G, H = h_1)} \)

\[
\begin{align*}
f_1(B) & := \sum_{a \in \text{dom}(A)} P(A = a) \cdot P(B|A = a) \\
f_2(B, D, E) & := \sum_{c \in \text{dom}(C)} P(C = c) \cdot P(D|B, C = c) \cdot P(E|C = c) \\
f_3(B, D, F, G) & := \sum_{e \in \text{dom}(E)} f_2(B, D, E = e) \cdot P(G|F, E = e) \\
f_4(G') & := P(H = h_1|G) \\
f_5(G') & := \sum_{i \in \text{dom}(I)} P(I = i|G) \\
f_6(D, F, G) & := \sum_{b \in \text{dom}(B)} f_1(B = b) \cdot f_3(B = b, D, F, G) \\
f_7(F, G') & := \sum_{d \in \text{dom}(D)} f_6(D = d, F, G) \cdot P(F|D = d) \\
f_8(G') & := \sum_{f \in \text{dom}(F')} f_7(F = f, G)
\end{align*}
\]
What good was Conditional Independence?

- That’s great... but it looks incredibly painful for large graphs.
- And... why did we bother learning conditional independence? Does it help us at all?
What good was Conditional Independence?

- That’s great… but it looks incredibly painful for large graphs.
- And… why did we bother learning conditional independence? Does it help us at all?
  - yes—we use the chain rule decomposition right at the beginning
What good was Conditional Independence?

- That’s great... but it looks incredibly painful for large graphs.
- And... why did we bother learning conditional independence? Does it help us at all?
  - yes—we use the chain rule decomposition right at the beginning

- Can we use our knowledge of conditional independence to make this calculation even simpler?
What good was Conditional Independence?

That’s great... but it looks incredibly painful for large graphs.

And... why did we bother learning conditional independence? Does it help us at all?

- Yes—we use the chain rule decomposition right at the beginning

Can we use our knowledge of conditional independence to make this calculation even simpler?

- Yes—there are some variables that we don’t have to sum out
- Intuitively, they’re the ones that are “pre-summed-out” in our tables
- Example: summing out $I$ on the previous slide
One Last Trick

One last trick to simplify calculations: we can repeatedly eliminate all leaf nodes that are neither observed nor queried, until we reach a fixed point.
One Last Trick

One last trick to simplify calculations: we can repeatedly eliminate all leaf nodes that are neither observed nor queried, until we reach a fixed point.

Can we justify that on a three-node graph—Fire, Alarm, and Smoke—when we ask for:

\[ P(Fire) \]
One Last Trick

One last trick to simplify calculations: we can repeatedly eliminate all leaf nodes that are neither observed nor queried, until we reach a fixed point.

Can we justify that on a three-node graph—Fire, Alarm, and Smoke—when we ask for:

- \( P(Fire) \)?
- \( P(Fire \mid Alarm) \)?