Exact Inference: Variable Elimination

Variable Elimination

Recall:
- Let $X$ be a set of random variables
- A factor $\phi$ is a function from $\text{Val}(X) \rightarrow \mathcal{R}$
- The set of variables $X$ is called the scope of the factor and denoted $\text{Scope}[\phi]$

We will be manipulating factors

Variable Elimination

Let $X$ be a set of variables, and $Y \not\in X$ a variable. Let $\phi(X,Y)$ be a factor. We define the factor marginalization of $Y$ in $\phi$, denoted $\sum_Y \phi$ to be a factor $\psi$ over $X$ such that:

$$\psi(X) = \sum_Y \phi(X,Y)$$

This operation is also called summing out of $Y$ in $\psi$

Variable Elimination

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>P(A,B,C)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.25</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0.35</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0.08</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
<td>0.16</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0.05</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>0.07</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>0.15</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>2</td>
<td>0.21</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0.09</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>2</td>
<td>0.18</td>
</tr>
</tbody>
</table>

This table represents $P(A,B,C)$.

Summing out B

<table>
<thead>
<tr>
<th>A</th>
<th>C</th>
<th>P(A,C)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0.25+0.08=0.33</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>0.35+0.16=0.51</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0.05+0=0.05</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>0.07+0=0.07</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0.15+0.09+0.24</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>0.21+0.18=0.39</td>
</tr>
</tbody>
</table>

Note: we only sum up entries in the table where the values of $X$ match up.
Variable Elimination

In a Bayesian network:
• Summing out all variables results in a factor with value 1

In a Markov network:
• Summing out all variables in the unnormalized distribution \( P_0 \) defined by the product of factors in the Markov network results in the partition function

Example of a factor product:

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>( \phi(A,B) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0.8</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0.1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0.2</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0.3</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0.9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>B</th>
<th>C</th>
<th>( \phi(B,C) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0.5</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0.7</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0.1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0.3</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0.2</td>
</tr>
</tbody>
</table>

Operations over factors:
• Addition is commutative: \( \sum_x \sum_y \phi = \sum_y \sum_x \phi \)
• Multiplication is commutative: \( \phi_1 \cdot \phi_2 = \phi_2 \cdot \phi_1 \)
• Products are associative: \( (\phi_1 \cdot \phi_2) \cdot \phi_3 = \phi_1 \cdot (\phi_2 \cdot \phi_3) \)
• Exchanging summations and products:
  If \( X \neq \text{Scope}\{\phi_i}\), \( \sum_x (\phi_1 \cdot \phi_2) = \phi_1 \sum_x \phi_2 \)

(X is not in the terms of \( \phi_i \))
**Variable Elimination**

Example:  
\[ P(D) = \sum_C \sum_B \sum_A P(A, B, C, D) \]
\[ = \sum_C \sum_B \sum_A \varphi_A \cdot \varphi_B \cdot \varphi_C \cdot \varphi_D \]
\[ = \sum_C \sum_B \varphi_C \cdot \varphi_D \cdot \varphi_B \left( \sum_A \varphi_A \right) \]
\[ = \varphi_D \sum_C \varphi_C \cdot \left( \sum_B \varphi_B \cdot \left( \sum_A \varphi_A \right) \right) \]

**Variable Elimination**

The general problem involves a sum-product inference task:
\[ \sum_{Z} \prod_{\varphi \in \Phi} \varphi \]

Trick: Push in the summations as far as you can

---

**Variable Elimination**

Procedure **Sum-Product-VE(**

- **\( \Phi \).** // A set of factors
- **\( Z \).** // Set of variables to be eliminated
- **<** // Ordering on \( Z \)

1. Let \( Z_1, ..., Z_k \) be an ordering of \( Z \) such that
2. \( Z_i < Z_j \) if and only if \( i < j \)
3. for \( i = 1, ..., k \)
4. \( \Phi \leftarrow \text{Sum-Product-Eliminate-Var}(\Phi, Z_i) \)
5. \( \phi^* \leftarrow \prod_{\varphi \in \Phi} \phi \)
6. Return \( \phi^* \)

---

**Variable Elimination**

Procedure **Sum-Product-Eliminate-Var(**

- **\( \Phi \).** // A set of factors
- **\( Z \).** // Variable to be eliminated

1. \( \Phi' \leftarrow \{ \varphi \in \Phi : Z \in \text{Scope}(\varphi) \} \)
2. \( \Phi'' \leftarrow \Phi - \Phi' \)
3. \( \psi \leftarrow \prod_{\varphi \in \Phi} \varphi \)
4. \( \tau \leftarrow \sum_{Z} \psi \)
5. Return \( \Phi'' \cup \{ \tau \} \)
Variable Elimination

Let $X$ be some set of variables, and let $\Phi$ be a set of factors such that for each $\phi \in \Phi$, $\text{Scope}[\phi] \subseteq X$. Let $Y \subset X$ be a set of query variables, and let $Z = X - Y$. Then for any ordering $<$ over $Z$, $\text{Sum-Product-VE}(\Phi, Z, <)$ returns a factor $\phi^*(Y)$ such that

$$\phi^*(Y) = \sum_{\prod_{\phi \in \Phi} \phi} \prod_{\phi \in \Phi} \phi$$

Example: compute $P_B(Y)$ for Bayesian network $B$. Let:

- $Z = \{Z_1, \ldots, Z_m\} = X - Y$ (eliminate all non-query variables)

Note: We can do the exact same thing on a Markov network except the final factor $\phi^*(Y)$ is unnormalized.

Variable Elimination

We will compute $P(J)$ using the elimination ordering $C, D, I, H, G, S, L$.

Note that:


1. Eliminating C:

$\psi_1(C, D) = \phi_C(C) \phi_D(D, C)$

$\tau_1(D) = \sum_C \psi_1(C, D)$

2. Eliminating D:

$\psi_2(G, I, D) = \phi_G(G, I, D) \cdot \tau_1(D)$

$\tau_2(G, I) = \sum_D \psi_2(G, I, D)$
Variable Elimination

Elimination ordering: C, D, I, H, G, S, L

3. Eliminating I:
\[ \Psi_l(G, I, S) = \phi_l(I) \cdot \psi_l(S, I) \cdot \psi_l(G, I) \]
\[ \tau_l(G, S) = \sum_I \Psi_l(G, I, S) \]

4. Eliminating H:
\[ \Psi_s(G, J, H) = \phi_s(H, G, J) \]
\[ \tau_s(G, J) = \sum_H \Psi_s(G, J, H) \]

Note: \( \tau_s = 1 \) since \( \sum_H P(H|G,J) \). However, in this elimination ordering, you do need to generate this factor for the next step.

5. Eliminating G:
\[ \Psi_5(G, J, L, S) = \tau_5(G, J) \cdot \psi_5(G, S) \cdot \phi_5(L, G) \]
\[ \tau_5(J, L, S) = \sum_G \Psi_5(G, J, L, S) \]

6. Eliminating S:
\[ \Psi_6(J, L, S) = \tau_5(J, L, S) \cdot \phi_6(J, L, S) \]
\[ \tau_6(J, L) = \sum_S \Psi_6(J, L, S) \]

How do we deal with evidence?
eg. \( P(J = \text{true} | I = \text{high}, H = \text{false}) \)?

Note that:
\[ P(J | I = \text{high}, H = \text{false}) = \frac{P(J, I = \text{high}, H = \text{false})}{P(I = \text{high}, H = \text{false})} \]
Variable Elimination

Proposition 4.7:
Let $\mathcal{B}$ be a Bayesian network over $\mathcal{X}$ and $E = e$ an observation. Let $W = \mathcal{X} - E$. Then $P_{\mathcal{B}}(W | e)$ is a Gibbs distribution defined by the factors where
\[
\phi_{X_i} = P_{\mathcal{B}}(X_i | \text{Parents}(X_i))[E = e]
\]
This is a reduced factor, meaning it is the factor with entries inconsistent with $E=e$ removed.

The partition function for this Gibbs distribution is $P(e)$.

Variable Elimination

Procedure Cond-Prob-VE(
\begin{align*}
\mathcal{K}, & \quad \text{// A network over } \mathcal{X} \\
Y, & \quad \text{// Set of query variables} \\
E = e & \quad \text{// Evidence}
\end{align*}
)
1. $\phi \leftarrow$ Factors parameterizing $\mathcal{K}$
2. Replace each $\phi \in \phi$ by $\phi[E=e]$
3. Select an elimination ordering $<$
4. $\mathcal{Z} \leftarrow \mathcal{X} - Y - E$
5. $\phi^* \leftarrow \text{Sum-Product-VE}(\phi, <, \mathcal{Z})$
6. $\alpha \leftarrow \sum_{y \in \text{Val}(Y)} \phi^*(y)$
7. return $\alpha, \phi^*$

Note: $\phi^*$ represents $P(Y, e)$ so divide $\phi^*$ by $\alpha$ to get $P(Y|e)$

Variable Elimination

• Replace each factor $\phi$ with $\phi[E=e]$
• Then do variable elimination like normal
• Remember to normalize with $P(E = e)$

Variable Elimination

Computing: $P(J,l=\text{high},H=\text{false})$

<table>
<thead>
<tr>
<th>Step</th>
<th>Variable Eliminated</th>
<th>Factors Used</th>
<th>Variables Involved</th>
<th>New Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>1'</td>
<td>C</td>
<td>$\phi_c(C), \phi_D(D, C)$</td>
<td>C, D</td>
<td>$\tau_1(D)$</td>
</tr>
<tr>
<td>2'</td>
<td>D</td>
<td>$\phi_G[I=\text{high}](G, D), \phi_I<a href="I">I=\text{high}</a>, \tau_2(D)$</td>
<td>G, D</td>
<td>$\tau_2(G)$</td>
</tr>
<tr>
<td>5'</td>
<td>G</td>
<td>$\tau_4(G), \phi_L(L, G), \phi_H[H=\text{false}](G, J)$</td>
<td>G, L, J</td>
<td>$\tau_5(J, L)$</td>
</tr>
<tr>
<td>6'</td>
<td>S</td>
<td>$\phi_J<a href="S">I=\text{high}</a>, \phi_J(J, L, S)$</td>
<td>J, L, S</td>
<td>$\tau_6(J, L)$</td>
</tr>
<tr>
<td>7'</td>
<td>L</td>
<td>$\tau_7(J, L), \tau_8(J, L)$</td>
<td>J, L</td>
<td>$\tau_7(J)$</td>
</tr>
</tbody>
</table>

Variable Elimination