CSE 473: Artificial Intelligence

Bayes’ Nets: Inference

Steve Tanimoto

[These slides were created by Dan Klein and Pieter Abbeel for CS188 Intro to AI at UC Berkeley. All CS188 materials are available at http://ai.berkeley.edu.]
Inference

- Inference: calculating some useful quantity from a joint probability distribution

- Examples:
  - Posterior probability
    \[ P(Q|E_1 = e_1, \ldots E_k = e_k) \]
  - Most likely explanation:
    \[ \text{argmax}_q P(Q = q|E_1 = e_1 \ldots) \]
Test for Infant Metabolic Defects

Blue ovals represent chromatographic peaks, grey ovals represent 20 metabolic diseases
Inference by Enumeration

- General case:
  - Evidence variables: $E_1 \ldots E_k = e_1 \ldots e_k$
  - Query* variable: $Q$
  - Hidden variables: $H_1 \ldots H_r$

- We want:

- Step 1: Select the entries consistent with the evidence

- Step 2: Sum out H to get joint of Query and evidence

- Step 3: Normalize

We want:

\[ P(Q|e_1 \ldots e_k) \]

Step 1: Select the entries consistent with the evidence

\[ P(Q, e_1 \ldots e_k) = \sum_{h_1 \ldots h_r} P(Q, h_1 \ldots h_r, e_1 \ldots e_k) \]

Step 2: Sum out H to get joint of Query and evidence

Step 3: Normalize

\[ Z = \sum_q P(Q, e_1 \ldots e_k) \]

\[ P(Q|e_1 \ldots e_k) = \frac{1}{Z} P(Q, e_1 \ldots e_k) \]
Inference by Enumeration in Bayes’ Net

- Given unlimited time, inference in BNs is easy
- Reminder of inference by enumeration by example:

\[
P(B \mid +j, +m) \propto_B P(B, +j, +m) = \sum_{e, a} P(B, e, a, +j, +m) = \sum_{e, a} P(B)P(e)P(a\mid B, e)P(+j\mid a)P(+m\mid a)
\]

\[
= P(B)P(+e)P(+a\mid B, +e)P(+j\mid +a)P(+m\mid +a) + P(B)P(+e)P(-a\mid B, +e)P(+j\mid -a)P(+m\mid -a) + P(B)P(-e)P(+a\mid B, -e)P(+j\mid +a)P(+m\mid +a) + P(B)P(-e)P(-a\mid B, -e)P(+j\mid -a)P(+m\mid -a)
\]
Inference by Enumeration vs. Variable Elimination

- Why is inference by enumeration so slow?
  - You join up the whole joint distribution before you sum out the hidden variables

- Idea: interleave joining and marginalizing!
  - Called “Variable Elimination”
  - Still NP-hard, but usually much faster than inference by enumeration

- First we’ll need some new notation: factors
Traffic Domain

\[ P(L) = ? \]

- **Inference by Enumeration**
  \[
  = \sum_t \sum_r P(L|t)P(r)P(t|r) 
  \]
  - Join on \( r \)
  - Join on \( t \)
  - Eliminate \( r \)
  - Eliminate \( t \)

- **Variable Elimination**
  \[
  = \sum_t P(L|t) \sum_r P(r)P(t|r) 
  \]
  - Join on \( r \)
  - Eliminate \( r \)
  - Join on \( t \)
  - Eliminate \( t \)
Factor Zoo
**Factor Zoo I**

- **Joint distribution: \( P(X,Y) \)**
  - Entries \( P(x,y) \) for all \( x, y \)
  - Sums to 1

- **Selected joint: \( P(x,Y) \)**
  - A slice of the joint distribution
  - Entries \( P(x,y) \) for fixed \( x \), all \( y \)
  - Sums to \( P(x) \)

- **Number of capitals = dimensionality of the table**

**\( P(T,W) \)**

<table>
<thead>
<tr>
<th>( T )</th>
<th>( W )</th>
<th>( P )</th>
</tr>
</thead>
<tbody>
<tr>
<td>hot</td>
<td>sun</td>
<td>0.4</td>
</tr>
<tr>
<td>hot</td>
<td>rain</td>
<td>0.1</td>
</tr>
<tr>
<td>cold</td>
<td>sun</td>
<td>0.2</td>
</tr>
<tr>
<td>cold</td>
<td>rain</td>
<td>0.3</td>
</tr>
</tbody>
</table>

**\( P(\text{cold},W) \)**

<table>
<thead>
<tr>
<th>( T )</th>
<th>( W )</th>
<th>( P )</th>
</tr>
</thead>
<tbody>
<tr>
<td>cold</td>
<td>sun</td>
<td>0.2</td>
</tr>
<tr>
<td>cold</td>
<td>rain</td>
<td>0.3</td>
</tr>
</tbody>
</table>
## Factor Zoo I

### Two dimensions

<table>
<thead>
<tr>
<th></th>
<th>T</th>
<th>W</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>hot</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sun</td>
<td>0.4</td>
<td>0.1</td>
<td></td>
</tr>
<tr>
<td>rain</td>
<td>0.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cold</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sun</td>
<td>0.2</td>
<td>0.3</td>
<td></td>
</tr>
<tr>
<td>rain</td>
<td>0.3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### One dimension

<table>
<thead>
<tr>
<th></th>
<th>T</th>
<th>W</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>cold</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sun</td>
<td>0.2</td>
<td>0.3</td>
<td></td>
</tr>
<tr>
<td>rain</td>
<td>0.3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Number of capitals = dimensionality of the table
Factor Zoo II

- Single conditional: \( P(Y \mid x) \)
  - Entries \( P(y \mid x) \) for fixed \( x \), all \( y \)
  - Sums to 1

- Family of conditionals: \( P(X \mid Y) \)
  - Multiple conditionals
  - Entries \( P(x \mid y) \) for all \( x, y \)
  - Sums to \( |Y| \)

\[
\begin{array}{|c|c|c|}
\hline
T & W & P \\
\hline
\text{cold} & \text{sun} & 0.4 \\
\text{cold} & \text{rain} & 0.6 \\
\hline
\end{array}
\]

\[
\begin{array}{|c|c|c|}
\hline
T & W & P \\
\hline
\text{hot} & \text{sun} & 0.8 \\
\text{hot} & \text{rain} & 0.2 \\
\text{cold} & \text{sun} & 0.4 \\
\text{cold} & \text{rain} & 0.6 \\
\hline
\end{array}
\]
Factor Zoo III

- Specified family: $P(y | X)$
  - Entries $P(y | x)$ for fixed $y$, but for all $x$
  - Sums to ... who knows!

$$P(rain | T')$$

<table>
<thead>
<tr>
<th></th>
<th>T</th>
<th>W</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>hot</td>
<td>rain</td>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td>cold</td>
<td>rain</td>
<td>0.6</td>
<td></td>
</tr>
</tbody>
</table>

- $P(rain | hot)$
- $P(rain | cold)$
In general, when we write $P(Y_1 \ldots Y_N \mid X_1 \ldots X_M)$

- It is a “factor,” a multi-dimensional array
- Its values are $P(y_1 \ldots y_N \mid x_1 \ldots x_M)$
- Any assigned (=lower-case) $X$ or $Y$ is a dimension missing (selected) from the array
Example: Traffic Domain

- Random Variables
  - R: Raining
  - T: Traffic
  - L: Late for class!

\[ P(L) = ? \]

\[
= \sum_{r,t} P(r, t, L)
= \sum_{r,t} P(r)P(t|r)P(L|t)
\]

\[
P(R)\
\begin{array}{c|c}
+r \quad 0.1 \\
-r \quad 0.9 \\
\end{array}
\]

\[
P(T|R)\
\begin{array}{c|cc}
+r & +t \quad 0.8 & \\
+r & -t \quad 0.2 & \\
-r & +t \quad 0.1 & \\
-r & -t \quad 0.9 & \\
\end{array}
\]

\[
P(L|T)\
\begin{array}{c|cc}
+t & +l \quad 0.3 & \\
+t & -l \quad 0.7 & \\
-t & +l \quad 0.1 & \\
-t & -l \quad 0.9 & \\
\end{array}
\]
Inference by Enumeration: Procedural Outline

- Track objects called **factors**
- Initial factors are local CPTs (one per node)

\[
\begin{array}{c|c}
P(R) & P(T|R) & P(L|T) \\
\hline
+r & +t & 0.8 \quad +t & +l & 0.3 \\
+\text{r} & -t & 0.2 \quad +t & -l & 0.7 \\
-\text{r} & +t & 0.1 \quad -t & +l & 0.1 \\
-\text{r} & -t & 0.9 \quad -t & -l & 0.9 \\
\end{array}
\]

- Any known values are selected
  - E.g. if we know \(L = +l\), the initial factors are

\[
\begin{array}{c|c}
P(R) & P(T|R) & P(+l|T) \\
\hline
+\text{r} & +t & 0.8 \quad +t & +l & 0.3 \\
+\text{r} & -t & 0.2 \quad +t & -l & 0.7 \\
-\text{r} & +t & 0.1 \quad -t & +l & 0.1 \\
-\text{r} & -t & 0.9 \quad -t & -l & 0.9 \\
\end{array}
\]

- Procedure: Join all factors, then eliminate all hidden variables
Operation 1: Join Factors

- First basic operation: joining factors
- Combining factors:
  - Just like a database join
  - Get all factors over the joining variable
  - Build a new factor over the union of the variables involved

- Example: Join on R

\[
P(R) \times P(T|R) \rightarrow P(R, T)
\]

<table>
<thead>
<tr>
<th></th>
<th>+r</th>
<th>0.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>-r</td>
<td>0.9</td>
<td></td>
</tr>
</tbody>
</table>

\[
\begin{array}{cc|c}
  & +r & +t & 0.8 \\
  +r & +t & 0.8 \\
  +r & -t & 0.2 \\
  -r & +t & 0.1 \\
  -r & -t & 0.9 \\
\end{array}
\]

<table>
<thead>
<tr>
<th></th>
<th>+r</th>
<th>+t</th>
<th>0.08</th>
</tr>
</thead>
<tbody>
<tr>
<td>+r</td>
<td>+t</td>
<td>0.08</td>
<td></td>
</tr>
<tr>
<td>+r</td>
<td>-t</td>
<td>0.02</td>
<td></td>
</tr>
<tr>
<td>-r</td>
<td>+t</td>
<td>0.09</td>
<td></td>
</tr>
<tr>
<td>-r</td>
<td>-t</td>
<td>0.81</td>
<td></td>
</tr>
</tbody>
</table>

- Computation for each entry: pointwise products \( \forall r, t : P(r, t) = P(r) \cdot P(t|r) \)
Example: Multiple Joins
Example: Multiple Joins

\[
P(R)
\]
\[
\begin{array}{c|c}
+r & 0.1 \\
-r & 0.9 \\
\end{array}
\]

\[
P(T|R)
\]
\[
\begin{array}{c|c}
+r & +t 0.8 \\
+r & -t 0.2 \\
-r & +t 0.1 \\
-r & -t 0.9 \\
\end{array}
\]

\[
P(L|T)
\]
\[
\begin{array}{c|c}
+t & +l 0.3 \\
+t & -l 0.7 \\
-t & +l 0.1 \\
-t & -l 0.9 \\
\end{array}
\]

Join R

\[
P(R, T)
\]
\[
\begin{array}{c|c|c}
+r & +t & 0.08 \\
+r & -t & 0.02 \\
-r & +t & 0.09 \\
-r & -t & 0.81 \\
\end{array}
\]

Join T

\[
P(L|T)
\]
\[
\begin{array}{c|c|c}
+t & +l & 0.3 \\
+t & -l & 0.7 \\
-t & +l & 0.1 \\
-t & -l & 0.9 \\
\end{array}
\]

\[
P(R, T, L)
\]
\[
\begin{array}{c|c|c|c}
+r & +t & +l & 0.024 \\
+r & +t & -l & 0.056 \\
+r & -t & +l & 0.002 \\
+r & -t & -l & 0.018 \\
-r & +t & +l & 0.027 \\
-r & +t & -l & 0.063 \\
-r & -t & +l & 0.081 \\
-r & -t & -l & 0.729 \\
\end{array}
\]
Operation 2: Eliminate

- Second basic operation: marginalization
- Take a factor and sum out a variable
  - Shrinks a factor to a smaller one
  - An aggregate/project operation
- Example:

\[
P(R, T)
\]

<table>
<thead>
<tr>
<th></th>
<th>+t</th>
<th>-t</th>
</tr>
</thead>
<tbody>
<tr>
<td>+r</td>
<td>0.08</td>
<td></td>
</tr>
<tr>
<td>-r</td>
<td>0.02</td>
<td></td>
</tr>
<tr>
<td>sum R</td>
<td></td>
<td>0.09</td>
</tr>
</tbody>
</table>

\[
P(T)
\]

<table>
<thead>
<tr>
<th></th>
<th>+t</th>
<th>-t</th>
</tr>
</thead>
<tbody>
<tr>
<td>+t</td>
<td>0.17</td>
<td></td>
</tr>
<tr>
<td>-t</td>
<td>0.83</td>
<td></td>
</tr>
</tbody>
</table>
Multiple Elimination

\[ P(R, T, L) \]

\[
\begin{array}{ccc}
+r & +t & +l & 0.024 \\
+r & +t & -l & 0.056 \\
+r & -t & +l & 0.002 \\
+r & -t & -l & 0.018 \\
-r & +t & +l & 0.027 \\
-r & +t & -l & 0.063 \\
-r & -t & +l & 0.081 \\
-r & -t & -l & 0.729 \\
\end{array}
\]

Sum out R

\[ P(T, L) \]

\[
\begin{array}{cc}
+t & +l & 0.051 \\
+t & -l & 0.119 \\
-t & +l & 0.083 \\
-t & -l & 0.747 \\
\end{array}
\]

Sum out T

\[ P(L) \]

\[
\begin{array}{c}
+l & 0.134 \\
-l & 0.886 \\
\end{array}
\]
Thus Far: Multiple Join, Multiple Eliminate (= Inference by Enumeration)
Marginalizing Early (= Variable Elimination)
Traffic Domain

\[ P(L) = ? \]

- **Inference by Enumeration**
  \[
  = \sum_t \sum_r P(L|t)P(r)P(t|r)
  \]
  - Join on \( t \)
  - Join on \( r \)
  - Eliminate \( r \)
  - Eliminate \( t \)

- **Variable Elimination**
  \[
  = \sum_t P(L|t) \sum_r P(r)P(t|r)
  \]
  - Join on \( r \)
  - Eliminate \( r \)
  - Join on \( t \)
  - Eliminate \( t \)
Marginalizing Early! (aka VE)

\[
\begin{array}{c|cc}
R & +r & -r \\
\hline
+ & 0.1 & 0.9 \\
\end{array}
\]

\[
\begin{array}{c|ccc}
P(T|R) & +r & -r & +t \\
\hline
+ & 0.8 & 0.2 & 0.1 \\
- & 0.9 & 0.1 & 0.2 \\
\end{array}
\]

\[
\begin{array}{c|ccc}
P(L|R) & +t & -t & +l \\
\hline
+ & 0.3 & 0.7 & 0.3 \\
- & 0.1 & 0.9 & 0.1 \\
\end{array}
\]

Join R

\[
\begin{array}{c|ccc}
P(R, T) & +r & -r & +t \\
\hline
+ & 0.08 & 0.02 & 0.09 \\
- & 0.81 & 0.2 & 0.9 \\
\end{array}
\]

Sum out R

\[
\begin{array}{c|c}
P(T) & +t \\
\hline
+ & 0.17 \\
- & 0.83 \\
\end{array}
\]

Join T

\[
\begin{array}{c|ccc}
P(T, L) & +t & -t & +l \\
\hline
+ & 0.051 & 0.119 & 0.083 \\
- & 0.747 & 0.083 & 0.747 \\
\end{array}
\]

Sum out T

\[
\begin{array}{c|c}
P(L) & +l \\
\hline
+ & 0.134 \\
- & 0.866 \\
\end{array}
\]
Evidence

- If evidence, start with factors that select that evidence
  - If there is no evidence, then use these initial factors:

|   | $P(R)$ | $P(T|R)$ | $P(L|T')$ |
|---|--------|----------|----------|
| +r | 0.1    | +t 0.8   | +l 0.3   |
| -r | 0.9    | -t 0.2   | -l 0.7   |
| -r | +t 0.1 |          |          |
| -r | -t 0.9 |          |          |

- But if given some evidence, eg $+r$, then select for it...
- Computing $P(L|+r)$ the initial factors become:

|   | $P(+r)$ | $P(T|+r)$ | $P(L|T')$ |
|---|---------|-----------|----------|
| +r | 0.1     | +t 0.8    | +l 0.3   |
| +r | +t 0.8  |           | +l 0.3   |
| +r | -t 0.2  |           | -l 0.7   |

- Next do joins & eliminate, removing all vars other than query + evidence
Evidence II

- Result will be a selected joint of query and evidence
  - E.g. for $P(L | +r)$, we would end up with:

$$
\begin{array}{|c|c|c|}
\hline
+r & +l & 0.026 \\
+r & -l & 0.074 \\
\hline
\end{array}
$$

Normalize

$$
\begin{array}{|c|c|}
\hline
+l & 0.26 \\
-l & 0.74 \\
\hline
\end{array}
$$

- To get our answer, just normalize this!

- That’s it!
General Variable Elimination

- **Query:** \( P(Q|E_1 = e_1, \ldots E_k = e_k) \)

- **Start with initial factors:**
  - Local CPTs (but instantiated by evidence)

- **While there are still hidden variables (not Q or evidence):**
  - Choose a hidden variable H
  - Join all factors mentioning H
  - Eliminate (sum out) H

- Join all remaining factors and normalize
Example: Alarm Network

\[
P(B \mid j, m) = ?
\]

<table>
<thead>
<tr>
<th>B</th>
<th>P(B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>+b</td>
<td>0.001</td>
</tr>
<tr>
<td>-b</td>
<td>0.999</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>E</th>
<th>P(E)</th>
</tr>
</thead>
<tbody>
<tr>
<td>+e</td>
<td>0.002</td>
</tr>
<tr>
<td>-e</td>
<td>0.998</td>
</tr>
</tbody>
</table>

| A   | J   | P(J|A) |
|-----|-----|------|
| +a  | +j  | 0.9  |
| +a  | -j  | 0.1  |
| -a  | +j  | 0.05 |
| -a  | -j  | 0.95 |

| A   | M   | P(M|A) |
|-----|-----|------|
| +a  | +m  | 0.7  |
| +a  | -m  | 0.3  |
| -a  | +m  | 0.01 |
| -a  | -m  | 0.99 |

| B   | E   | A   | P(A|B,E) |
|-----|-----|-----|---------|
| +b  | +e  | +a  | 0.95    |
| +b  | +e  | -a  | 0.05    |
| +b  | -e  | +a  | 0.94    |
| +b  | -e  | -a  | 0.06    |
| -b  | +e  | +a  | 0.29    |
| -b  | +e  | -a  | 0.71    |
| -b  | -e  | +a  | 0.001   |
| -b  | -e  | -a  | 0.999   |
Example

\[ P(B|j, m) \propto P(B, j, m) \]

\[
\begin{array}{ccccc}
  P(B) & P(E) & P(A|B, E) & P(j|A) & P(m|A)
\end{array}
\]

**Choose A**

\[
\begin{align*}
P(A|B, E) \\
P(j|A) \\
P(m|A)
\end{align*}
\]

\[
\begin{array}{ccccc}
P(B) & P(E) & P(j, m|B, E)
\end{array}
\]
Example

Choose E

\[
P(E) \quad P(j, m \mid B, E)
\]

\[
\times \quad P(j, m, E \mid B) \quad \sum \quad P(j, m \mid B)
\]

Finish with B

\[
P(B) \quad P(j, m \mid B)
\]

\[
\times \quad P(j, m, B) \quad \text{Normalize} \quad P(B \mid j, m)
\]
Same Example in Equations

\[ P(B|j, m) \propto P(B, j, m) \]

| \( P(B) \) | \( P(E) \) | \( P(A|B, E) \) | \( P(j|A) \) | \( P(m|A) \) |
|----------------|----------------|----------------|----------------|----------------|

\[
P(B|j, m) \propto P(B, j, m) \\
= \sum_{e,a} P(B, j, m, e, a) \\
= \sum_{e,a} P(B)P(e)P(a|B, e)P(j|a)P(m|a) \\
= \sum_{e} P(B)P(e)\sum_{a} P(a|B, e)P(j|a)P(m|a) \\
= \sum_{e} P(B)P(e)f_1(B, e, j, m) \\
= P(B)\sum_{e} P(e)f_1(B, e, j, m) \\
= P(B)f_2(B, j, m)
\]

marginal can be obtained from joint by summing out

use Bayes’ net joint distribution expression

use \( xy + xz = x^*(y+z) \) \textbf{do sum first}

joining on \( a \), and then summing out gives \( f_1 \)

use \( xy + xz = x^*(y+z) \) \textbf{do sum first}

joining on \( e \), and then summing out gives \( f_2 \)

Simple! Exploiting \( uw+y+uwz+uxy+uxz+vwy+vzw+vxy+vzx = (u+v)(w+x)(y+z) \) to reduce computation
Variable Elimination

\[ P(b|j,m) = \alpha P(b) \sum_{e} P(e) \sum_{a} P(a|b,e)P(j|a)P(m,a) \]

Repeated computations \(\rightarrow\) Dynamic Programming
Choices during Variable Elimination

- **Query:** \( P(Q|E_1 = e_1, \ldots E_k = e_k) \)

- **Start with initial factors:**
  - Local CPTs (but instantiated by evidence)

- **While there are still hidden variables (not Q or evidence):**
  - **Choose** a hidden variable H
  - Join all factors mentioning H
  - Eliminate (sum out) H

- **Join all remaining factors and normalize**

\[
\begin{array}{c|c}
 x & P(x) \\
 \hline
 -5 & 0.06 \\
 -2 & 0.28 \\
 0 & 0.07 \\
 1 & 0.2 \\
 5 & 0.01 \\
\end{array}
\]
Another Variable Elimination Example

Query: \( P(X_3|Y_1 = y_1, Y_2 = y_2, Y_3 = y_3) \)

Start by inserting evidence, which gives the following initial factors:

\[
p(Z)p(X_1|Z)p(X_2|Z)p(X_3|Z)p(y_1|X_1)p(y_2|X_2)p(y_3|X_3)
\]

What variables could we eliminate?
Another Variable Elimination Example

Query: \( P(X_3|Y_1 = y_1, Y_2 = y_2, Y_3 = y_3) \)

Start by inserting evidence, which gives the following initial factors:

\[
p(Z)p(X_1|Z)p(X_2|Z)p(X_3|Z)p(y_1|X_1)p(y_2|X_2)p(y_3|X_3)
\]

Eliminate \( X_1 \), this introduces the factor \( f_1(Z, y_1) = \sum_{x_1} p(x_1|Z)p(y_1|x_1) \), and we are left with:

\[
p(Z)f_1(Z, y_1)p(X_2|Z)p(X_3|Z)p(y_2|X_2)p(y_3|X_3)
\]

Eliminate \( X_2 \), this introduces the factor \( f_2(Z, y_2) = \sum_{x_2} p(x_2|Z)p(y_2|x_2) \), and we are left with:

\[
p(Z)f_1(Z, y_1)f_2(Z, y_2)p(X_3|Z)p(y_3|X_3)
\]

Eliminate \( Z \), this introduces the factor \( f_3(y_1, y_2, X_3) = \sum_z p(z)f_1(z, y_1)f_2(z, y_2)p(X_3|z) \), and we are left:

\[
p(y_3|X_3), f_3(y_1, y_2, X_3)
\]

No hidden variables left. Join the remaining factors to get:

\[
f_4(y_1, y_2, y_3, X_3) = P(y_3|X_3)f_3(y_1, y_2, X_3).
\]

Normalizing over \( X_3 \) gives \( P(X_3|y_1, y_2, y_3) \).
Another Variable Elimination Example

Query: \( P(X_3 | Y_1 = y_1, Y_2 = y_2, Y_3 = y_3) \)

Start by inserting evidence, which gives the following initial factors:

\[
p(Z)p(X_1|Z)p(X_2|Z)p(X_3|Z)p(y_1|X_1)p(y_2|X_2)p(y_3|X_3)
\]

Alternatively, suppose we start by eliminating \( Z \):

\[
P(X_1 | Z) \times P(X_2 | Z) \times P(X_3 | Z) \rightarrow f_Z(X_1, X_2, X_3)
\]

What is the resulting factor?  \( f_Z(X_1, X_2, X_3) \)

What dimension is it?  \( 3 \)

How many entries?  \( k^3 \)
Another Variable Elimination Example

Query: \( P(X_3|Y_1 = y_1, Y_2 = y_2, Y_3 = y_3) \)

Start by inserting evidence, which gives the following initial factors:

\[
p(Z)p(X_1|Z)p(X_2|Z)p(X_3|Z)p(y_1|X_1)p(y_2|X_2)p(y_3|X_3)
\]

Eliminate \( X_1 \), this introduces the factor \( f_1(Z, y_1) = \sum_{x_1} p(x_1|Z)p(y_1|x_1) \), and we are left with:

\[
p(Z)f_1(Z, y_1)p(X_2|Z)p(X_3|Z)p(y_2|X_2)p(y_3|X_3)
\]

Eliminate \( X_2 \), this introduces the factor \( f_2(Z, y_2) = \sum_{x_2} p(x_2|Z)p(y_2|x_2) \), and we are left with:

\[
p(Z)f_1(Z, y_1)f_2(Z, y_2)p(X_3|Z)p(y_3|X_3)
\]

Eliminate \( Z \), this introduces the factor \( f_3(y_1, y_2, X_3) = \sum_z p(z)f_1(z, y_1)f_2(z, y_2)p(X_3|z) \), and we are left:

\[
p(y_3|X_3), f_3(y_1, y_2, X_3)
\]

No hidden variables left. Join the remaining factors to get:

\[
f_4(y_1, y_2, y_3, X_3) = P(y_3|X_3)f_3(y_1, y_2, X_3).
\]

Normalizing over \( X_3 \) gives \( P(X_3|y_1, y_2, y_3) \).

Computational complexity depends on the largest factor generated by the process.
Size of factor = number of entries in table.
For the query $P(X_n | y_1, ..., y_n)$ work through the following two different orderings as done in previous slide: $Z, X_1, ..., X_{n-1}$ and $X_1, ..., X_{n-1}, Z$. What is the size of the maximum factor generated for each of the orderings?

- Answer: $2^{n+1}$ versus $2^2$ (assuming binary)

- In general: the ordering can greatly affect efficiency.
VE: Computational and Space Complexity

- The computational and space complexity of variable elimination is determined by the largest factor.

- The elimination ordering can greatly affect the size of the largest factor.
  - E.g., previous slide’s example $2^n$ vs. 2

- Does there always exist an ordering that only results in small factors?
  - No!
Worst Case Complexity?

- **CSP:**

\[
(x_1 \lor x_2 \lor \neg x_3) \land (\neg x_1 \lor x_3 \lor x_4) \land (x_2 \lor \neg x_2 \lor x_4) \land (\neg x_3 \lor \neg x_4 \lor \neg x_5) \land (x_2 \lor x_5 \lor x_7) \land (x_4 \lor x_5 \lor x_6) \land (\neg x_5 \lor x_6 \lor \neg x_7) \land (\neg x_5 \lor \neg x_6 \lor x_7)
\]

\[P(X_i = 0) = P(X_i = 1) = 0.5\]

\[Y_1 = X_1 \lor X_2 \lor \neg X_3\]

\[Y_8 = \neg X_5 \lor X_6 \lor X_7\]

\[Y_{1,2} = Y_1 \land Y_2\]

\[Y_{7,8} = Y_7 \land Y_8\]

\[Y_{1,2,3,4} = Y_{1,2} \land Y_{3,4}\]

\[Y_{5,6,7,8} = Y_{5,6} \land Y_{7,8}\]

\[Z = Y_{1,2,3,4} \land Y_{5,6,7,8}\]

- If we can answer \(P(z)\) equal to zero or not, we answered whether the 3-SAT problem has a solution.

- Hence inference in Bayes’ nets is NP-hard. No known efficient probabilistic inference in general.
Bayes’ Nets

- Representation
- Conditional Independences
  - Probabilistic Inference
    - Enumeration (exact, exponential complexity)
    - Variable elimination (exact, worst-case exponential complexity, often better)
    - Inference is NP-complete
      - Sampling (approximate)
  - Learning Bayes’ Nets from Data