Bayes’ Net Representation

- A directed, acyclic graph, one node per random variable
- A conditional probability table (CPT) for each node
  - A collection of distributions over $X$, one for each combination of parents’ values
    \[ P(X|a_1\ldots a_n) \]
- Bayes’ nets implicitly encode joint distributions
  - As a product of local conditional distributions
  - To see what probability a BN gives to a full assignment, multiply all the relevant conditionals together:
    \[ P(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} P(x_i|\text{parents}(X_i)) \]
Example: Alarm Network

\[ P(+b, -e, +a, -j, +m) = \ ? \]

\[ P(+b, -e, +a, -j, +m) = P(+b)P(-e)P(+a | b)P(-j | a)P(+m | a) = 0.001 \times 0.998 \times 0.94 \times 0.1 \times 0.7 \]
Independence in a BN

- **Important question about a BN:**
  - Are two nodes independent given certain evidence?
  - If yes, can prove using algebra (tedious in general)
  - If no, can prove with a counter example
- **Example:**

  ![Diagram of nodes X, Y, Z]

  \[
  P(X) = 0.9 \quad P(Y|X) = 1 \quad P(Z|Y) = 0.5 \\
  P(Y|\#X) = 1 \quad P(Z|\#Y) = 0.5
  \]

- **Question:** are X and Z *necessarily* independent?
  - **Answer:** no. Example: low pressure causes rain, which causes traffic.
  - X can influence Z, Z can influence X (via Y)
  - Addendum: they *could* be independent: how?

---

D-separation
D-separation: Outline

- Study independence properties for triples
- Analyze complex cases in terms of member triples
- D-separation: a condition / algorithm for answering such queries

Causal Chains

- This configuration is a “causal chain”

\[ P(x, y, z) = P(x)P(y|x)P(z|y) \]

- Guaranteed X independent of Z given Y?

\[
P(z|x, y) = \frac{P(x, y, z)}{P(x, y)}
= \frac{P(x)P(y|x)P(z|y)}{P(x)P(y|x)}
= P(z|y)
\]

Yes!

- Evidence along the chain “blocks” the influence
Common Cause

- This configuration is a “common cause”

\[ P(x, y, z) = P(y)P(x|y)P(z|y) \]

- Guaranteed X and Z independent given Y?

\[
P(z|x, y) = \frac{P(x, y, z)}{P(x, y)}
\]

\[
= \frac{P(y)P(x|y)P(z|y)}{P(y)P(x|y)}
\]

\[ = P(z|y) \]

Yes!

- Observing the cause blocks influence between effects.

Common Effect

- Last configuration: two causes of one effect (v-structures)

- Are X and Y independent?
  - Yes: the ballgame and the rain cause traffic, but they are not correlated
  - Still need to prove they must be (try it!)

- Are X and Y independent given Z?
  - No: seeing traffic puts the rain and the ballgame in competition as explanation.

- This is backwards from the other cases
  - Observing an effect activates influence between possible causes.
General question: in a given BN, are two variables independent (given evidence)?

Solution: analyze the graph

Any complex example can be broken into repetitions of the three canonical cases
Active / Inactive Paths

- **Question:** Are $X$ and $Y$ conditionally independent given evidence variables $\{Z\}$?
  - Yes, if $X$ and $Y$ “d-separated” by $Z$
  - Consider all (undirected) paths from $X$ to $Y$
  - No active paths = independence!

- A path is active if each triple is active:
  - Causal chain $A \rightarrow B \rightarrow C$ where $B$ is unobserved (either direction)
  - Common cause $A \leftarrow B \rightarrow C$ where $B$ is unobserved
  - Common effect (aka v-structure)
    - $A \rightarrow B \leftarrow C$ where $B$ or one of its descendents is observed

- All it takes to block a path is a single inactive segment

---

D-Separation

- **Query:** $X_i \perp\!\!\!\perp X_j | \{X_{k_1}, \ldots, X_{k_n}\}$ ?

- **Check all (undirected!) paths between $X_i$ and $X_j$**
  - If one or more active, then independence not guaranteed
    $$X_i \not\perp\!\!\!\perp X_j | \{X_{k_1}, \ldots, X_{k_n}\}$$
  - Otherwise (i.e. if all paths are inactive), then independence is guaranteed
    $$X_i \perp\!\!\!\perp X_j | \{X_{k_1}, \ldots, X_{k_n}\}$$
Example

\[ R \perp B \quad \text{Yes} \]
\[ R \perp B \mid T \]
\[ R \perp B \mid T' \]

Example

\[ L \perp T' \mid T \quad \text{Yes} \]
\[ L \perp B \quad \text{Yes} \]
\[ L \perp B \mid T \]
\[ L \perp B \mid T' \]
\[ L \perp B \mid T', R \quad \text{Yes} \]
Example

- Variables:
  - R: Raining
  - T: Traffic
  - D: Roof drips
  - S: I’m sad

- Questions:

  \[ T \perp D \]
  \[ T \perp D | R \quad \text{Yes} \]
  \[ T \perp D | R, S \]

Structure Implications

- Given a Bayes net structure, can run d-separation algorithm to build a complete list of conditional independences that are necessarily true of the form

  \[ X_i \perp X_j | \{X_{k_1}, \ldots, X_{k_n}\} \]

- This list determines the set of probability distributions that can be represented
Computing All Independences

Topology Limits Distributions

- Given some graph topology $G$, only certain joint distributions can be encoded
- The graph structure guarantees certain (conditional) independences
- (There might be more independence)
- Adding arcs increases the set of distributions, but has several costs
- Full conditioning can encode any distribution
Bayes Nets Representation Summary

- Bayes nets compactly encode joint distributions
- Guaranteed independencies of distributions can be deduced from BN graph structure
- D-separation gives precise conditional independence guarantees from graph alone
- A Bayes’ net’s joint distribution may have further (conditional) independence that is not detectable until you inspect its specific distribution

Bayes’ Nets

- ✔ Representation
- ✔ Conditional Independences
- ✔ Probabilistic Inference
  - Enumeration (exact, exponential complexity)
  - Variable elimination (exact, worst-case exponential complexity, often better)
  - Probabilistic inference is NP-complete
  - Sampling (approximate)
- ✔ Learning Bayes’ Nets from Data
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:
  \[ \arg\max_q P(Q = q | E_1 = e_1, \ldots) \]

Liver Disorder

Inference by Enumeration

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

$$P(Q, e_1 \ldots e_k) = \sum_{h_1, \ldots, h_r} P(Q, h_1 \ldots h_r, e_1 \ldots e_k) \prod_{i=1}^{n} \frac{X_i, X_2, \ldots, X_n}{X_1, X_2, \ldots, X_n}$$

- We want:
  - * Works fine with multiple query variables, too

$$P(Q|e_1 \ldots e_k) = \frac{1}{Z} \cdot \frac{P(Q, e_1 \ldots e_k)}{P(Q|e_1 \ldots e_k)}$$

Inference by Enumeration in Bayes’ Net

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

$$P(B | + j, +m) \propto_B P(B, +j, +m)$$

$$= \sum_{e, a} P(B, e, a, +j, +m)$$

$$= \sum_{e, a} P(B)P(e)P(a|B, e)P(+j|a)P(+m|a)$$

$$= P(B)P(+e)P(+a|B, +e)P(+j|+a)P(+m|+a) + P(B)P(+e)P(-a|B, +e)P(+j|-a)P(+m|-a)$$

$$+ P(B)P(-e)P(+a|B, -e)P(+j|+a)P(+m|+a) + P(B)P(-e)P(-a|B, -e)P(+j|-a)P(+m|-a)$$
Inference by Enumeration?

\[ P(\text{Antilock}|\text{observed variables}) = ? \]

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
Factor Zoo

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

<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>

<table>
<thead>
<tr>
<th>$P(cold, W)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T$</td>
</tr>
<tr>
<td>cold</td>
</tr>
<tr>
<td>cold</td>
</tr>
</tbody>
</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 \( |Y| \)

\[
\begin{array}{ccc}
  P(W \mid \text{cold}) \\
  \hline
  T & W & P \\
  \text{cold} & \text{sun} & 0.4 \\
  \text{cold} & \text{rain} & 0.6 \\
\end{array}
\]

Factor Zoo III

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

\[
\begin{array}{ccc}
  P(rain \mid T) \\
  \hline
  T & W & P \\
  \text{hot} & \text{rain} & 0.2 \\
  \text{cold} & \text{rain} & 0.6 \\
\end{array}
\]

\[
\begin{array}{ccc}
  P(rain \mid \text{hot}) \\
  \hline
  T & W & P \\
  \text{hot} & \text{sun} & 0.8 \\
  \text{hot} & \text{rain} & 0.2 \\
  \text{cold} & \text{sun} & 0.4 \\
  \text{cold} & \text{rain} & 0.6 \\
\end{array}
\]

\[
\begin{array}{ccc}
  P(rain \mid \text{cold}) \\
  \hline
  T & W & P \\
  \text{hot} & \text{sun} & 0.2 \\
  \text{cold} & \text{rain} & 0.6 \\
\end{array}
\]
Factor Zoo Summary

- 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!

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

\[
P(L) = \sum_{r,t} P(r)P(t|r)P(L|t)
\]
Inference by Enumeration: Procedural Outline

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

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

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

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

- 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 \( R \)

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

\[
\begin{array}{c|c|c|c|}
\text{Input Factors} & \text{Output Factors} & \text{Likelihoods} \\
\hline
P(R) & P(T|R) & P(R, T) \\
\hline
+r & 0.1 & +r & +t & 0.8 & +r & +t & 0.08 \\
+r & -t & 0.2 & +r & -t & 0.02 \\
-r & +t & 0.1 & -r & +t & 0.09 \\
-r & -t & 0.9 & -r & -t & 0.81 \\
\end{array}
\]

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

\[ P(R) \]

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

\[ P(T|R) \]

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

\[ P(L|T) \]

\[ \begin{array}{c|cc}
+\text{t} & +\text{l} & 0.3 \\
+\text{t} & -\text{l} & 0.7 \\
-\text{t} & +\text{l} & 0.1 \\
-\text{t} & -\text{l} & 0.9 \\
\end{array} \]

Join R

\[ P(R, T) \]

\[ \begin{array}{c|cc}
+\text{r} & +\text{t} & 0.08 \\
+\text{r} & -\text{t} & 0.02 \\
-\text{r} & +\text{t} & 0.09 \\
-\text{r} & -\text{t} & 0.81 \\
\end{array} \]

Join T

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

\[ \begin{array}{c|cccc}
+\text{r} & +\text{t} & +\text{l} & 0.024 \\
+\text{r} & +\text{t} & -\text{l} & 0.056 \\
+\text{r} & -\text{t} & +\text{l} & 0.002 \\
+\text{r} & -\text{t} & -\text{l} & 0.018 \\
-\text{r} & +\text{t} & +\text{l} & 0.027 \\
-\text{r} & +\text{t} & -\text{l} & 0.063 \\
-\text{r} & -\text{t} & +\text{l} & 0.081 \\
-\text{r} & -\text{t} & -\text{l} & 0.729 \\
\end{array} \]
Operation 2: Eliminate

- Second basic operation: **marginalization**
- Take a factor and sum out a variable
  - Shrinks a factor to a smaller one
  - A *projection* operation

**Example:**

\[
P(R, T)
\]

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th>(P(R, T))</th>
</tr>
</thead>
<tbody>
<tr>
<td>+r</td>
<td>+t</td>
<td>+l</td>
<td>0.024</td>
</tr>
<tr>
<td>+r</td>
<td>+t</td>
<td>-l</td>
<td>0.056</td>
</tr>
<tr>
<td>+r</td>
<td>-t</td>
<td>+l</td>
<td>0.018</td>
</tr>
<tr>
<td>+r</td>
<td>-t</td>
<td>-l</td>
<td>0.027</td>
</tr>
<tr>
<td>-r</td>
<td>+t</td>
<td>+l</td>
<td>0.063</td>
</tr>
<tr>
<td>-r</td>
<td>+t</td>
<td>-l</td>
<td>0.081</td>
</tr>
<tr>
<td>-r</td>
<td>-t</td>
<td>+l</td>
<td>0.729</td>
</tr>
</tbody>
</table>

Sum \(R\):

\[
P(T')
\]

<table>
<thead>
<tr>
<th></th>
<th>(P(T'))</th>
</tr>
</thead>
<tbody>
<tr>
<td>+t</td>
<td>0.17</td>
</tr>
<tr>
<td>-t</td>
<td>0.83</td>
</tr>
</tbody>
</table>

Multiple Elimination

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

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th>(P(R, T, L))</th>
</tr>
</thead>
<tbody>
<tr>
<td>+r</td>
<td>+t</td>
<td>+l</td>
<td>+l</td>
<td>0.024</td>
</tr>
<tr>
<td>+r</td>
<td>+t</td>
<td>+l</td>
<td>-l</td>
<td>0.056</td>
</tr>
<tr>
<td>+r</td>
<td>-t</td>
<td>+l</td>
<td>+l</td>
<td>0.018</td>
</tr>
<tr>
<td>+r</td>
<td>-t</td>
<td>+l</td>
<td>-l</td>
<td>0.027</td>
</tr>
<tr>
<td>-r</td>
<td>+t</td>
<td>+l</td>
<td>+l</td>
<td>0.063</td>
</tr>
<tr>
<td>-r</td>
<td>+t</td>
<td>+l</td>
<td>-l</td>
<td>0.081</td>
</tr>
<tr>
<td>-r</td>
<td>-t</td>
<td>+l</td>
<td>+l</td>
<td>0.729</td>
</tr>
</tbody>
</table>

Sum out \(R\):

\[
P(T, L)
\]

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>(P(T, L))</th>
</tr>
</thead>
<tbody>
<tr>
<td>+t</td>
<td>+l</td>
<td>0.051</td>
</tr>
<tr>
<td>+t</td>
<td>-l</td>
<td>0.119</td>
</tr>
<tr>
<td>-t</td>
<td>+l</td>
<td>0.083</td>
</tr>
<tr>
<td>-t</td>
<td>-l</td>
<td>0.747</td>
</tr>
</tbody>
</table>

Sum out \(T\):

\[
P(L)
\]

<table>
<thead>
<tr>
<th></th>
<th>(P(L))</th>
</tr>
</thead>
<tbody>
<tr>
<td>+l</td>
<td>0.134</td>
</tr>
<tr>
<td>-l</td>
<td>0.886</td>
</tr>
</tbody>
</table>
Thus Far: Multiple Join, Multiple Eliminate (= Inference by Enumeration)

Marginalizing Early (= Variable Elimination)
Traffic Domain

\[ P(L) = ? \]

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

- **Variable Elimination**
  \[ = \sum_t P(L|t) \sum_r P(r)P(t|r) \]

Marginalizing Early! (aka VE)

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

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

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

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

- **P(T, L)**
  \[ \begin{array}{c|c|c}
  & +l & 0.51 \\
  & +t & -l & 0.119 \\
  & -t & +l & 0.083 \\
  & -t & -l & 0.747 \\
  \end{array} \]

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

- If evidence, start with factors that select that evidence
  - No evidence uses these initial factors:

|     | $P(R)$ | $P(T|R)$ | $P(L|T)$ |
|-----|--------|----------|----------|
| $+$  | 0.1    | 0.8      | 0.3      |
| $-$  | 0.9    | 0.2      | 0.7      |
| $+$  | 0.1    | 0.9      | 0.1      |
| $-$  | 0.9    | 0.7      | 0.9      |

- Computing $P(L | +r)$ the initial factors become:

|     | $P(\bar{r})$ | $P(T | +r)$ | $P(L|T)$ |
|-----|--------------|-------------|----------|
| $+$  | 0.1          | 0.8         | 0.3      |
| $+$  | 0.2          | 0.2         | 0.7      |
| $+$  | 0.9          | 0.9         | 0.9      |

- We eliminate 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:

|     | $P(\bar{r}, L)$ | Normalize | $P(L | +r)$ |
|-----|-----------------|-----------|------------|
| $+$  | $+l$            | 0.026     | $+l$       |
| $+$  | $-l$            | 0.074     | $-l$       |

- 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):**
  - Pick a hidden variable \( H \)
  - Join all factors mentioning \( H \)
  - Eliminate (sum out) \( H \)

- **Join all remaining factors and normalize**

Example

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

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

Choose \( A \)

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

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

Choose E
\[ P(E) \times P(j, m|B, E) \]
\[ \sum \rightarrow P(j, m|B) \]

Finish with B
\[ P(B) \times P(j, m|B) \]
\[ \text{Normalize} \quad P(B|j, m) \]

Same Example in Equations

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

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

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

marginal can be obtained from joint by summing out
use Bayes’ net joint distribution expression
use \( x^*(y+z) = xy + xz \)
joining on \( a \), and then summing out gives \( f_1 \)
use \( x^*(y+z) = xy + xz \)
joining on \( e \), and then summing out gives \( f_2 \)

All we are doing is exploiting \( uwy + uwz + uxy + uxz + vwy + vwz + vxy + vxz = (u+v)(w+x)(y+z) \) to improve computational efficiency!
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 $f_3(y_1, y_2, y_3) = \sum_{z} p(z)f_1(z, y_1)f_2(z, y_2)p(X_3 | z)$, and we are left with:

$p(y_3 | X_3). f_3(y_1, y_2, X_3)\)

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

$f_3(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 critically depends on the largest factor being generated in this process. Size of factor = number of entries in table. In example above (assuming binary) all factors generated are of size $2^2$ --- as they all only have one variable ($Z, Z,$ and $X_3$ respectively).

Variable Elimination Ordering

- 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:

$$\begin{align*}
(x_1 \lor x_2 \lor \neg x_3) \land (\neg x_1 \lor x_3 \lor \neg x_4) \land (x_2 \lor \neg x_2 \lor x_4) \land (\neg x_3 \lor \neg x_4 \lor \neg x_5) \land (x_2 \lor x_5 \lor x_7) \land (x_4 \lor x_5 \lor x_6) \land (\neg x_5 \lor x_6 \lor \neg x_7) \land (\neg x_5 \lor \neg x_6 \lor x_7)
\end{align*}$$

$$P(X_1 = 0) = P(X_1 = 1) = 0.5$$

$$Y_1 = X_1 \lor X_2 \lor \neg X_3$$

$$Y_8 = \neg X_5 \lor X_6 \lor X_7$$

$$Y_{1,2} = Y_1 \land Y_2$$

$$Y_{7,8} = Y_7 \land Y_8$$

$$Y_{1,2,3,4} = Y_{1,2} \land Y_{3,4}$$

$$Y_{5,6,7,8} = Y_{5,6} \land Y_{7,8}$$

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

- 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.
Polyprees

- A polytree is a directed graph with no undirected cycles
- For poly-trees you can always find an ordering that is efficient
  - Try it!!
- Cut-set conditioning for Bayes’ net inference
  - Choose set of variables such that if removed only a polytree remains
  - Exercise: Think about how the specifics would work out!

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