Instructors: Anca Dragan --- University of California, Berkeley

[Slides by Dan Klein, Pieter Abbeel, Anca Dragan. http://ai.berkeley.edu.]
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

<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>

| 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   |

| 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  |
Example: Alarm Network

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

<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: Alarm Network

\[ P(+b, -e, +a, -j, +m) = \]
\[ P(+b)P(-e)P(+a| +b, -e)P(-j| +a)P(+m| +a) = \]
\[ 0.001 \times 0.998 \times 0.94 \times 0.1 \times 0.7 \]
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:
    \[ \arg\max_q P(Q = q|E_1 = e_1 \ldots) \]
Inference by Enumeration

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

\[ X_1, X_2, \ldots X_n \]

*Works fine with multiple query variables, too

- **We want:**

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

- **Step 1:** Select the entries consistent with the evidence

- **Step 2:** Sum out \( H \) to get joint of Query and evidence

- **Step 3:** Normalize

\[ \frac{1}{Z} \]

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

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

- Given unlimited time, inference in BNs is easy

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

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

  \[
  \{ X_1, X_2, \ldots X_n \} \quad \text{All variables}
  \]

- **Step 1:** Select the entries consistent with the evidence

  - Table:
    | x | Prob |
    |---|------|
    | -3 | 0.05 |
    | -1 | 0.25 |
    | 0  | 0.07 |
    | 1  | 0.2  |
    | 2  | 0.15 |
    | 5  | 0.01 |

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

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

  - Sum out hidden variables $X_1, X_2, \ldots X_n$

- **Step 3:** Normalize

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

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

- **We want:**

  * Works fine with multiple query variables, too
Example: Traffic Domain

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

\[
P(L) = ?
\]

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

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

**$P(R)$**

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

**$P(T|R)$**

<table>
<thead>
<tr>
<th></th>
<th>$+$</th>
<th>$-$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r$</td>
<td>0.8</td>
<td>0.2</td>
</tr>
<tr>
<td>$-r$</td>
<td>0.1</td>
<td>0.9</td>
</tr>
</tbody>
</table>

**$P(L|T)$**

<table>
<thead>
<tr>
<th></th>
<th>$+$</th>
<th>$-$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t$</td>
<td>0.3</td>
<td>0.7</td>
</tr>
<tr>
<td>$-t$</td>
<td>0.1</td>
<td>0.9</td>
</tr>
</tbody>
</table>
Inference by Enumeration: Procedural Outline

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

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

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

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

- Procedure: Join all factors, then sum out 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>( R )</th>
<th>( T )</th>
</tr>
</thead>
<tbody>
<tr>
<td>+r</td>
<td>0.1</td>
</tr>
<tr>
<td>-r</td>
<td>0.9</td>
</tr>
</tbody>
</table>

  | \( +r \) | \( +t \) | 0.8 |
  | \( +r \) | \( -t \) | 0.2 |
  | \( -r \) | \( +t \) | 0.1 |
  | \( -r \) | \( -t \) | 0.9 |

  \[ P(r,t) = P(r) \cdot P(t|r) \]

  - Computation for each entry: pointwise products
Example: Multiple Joins
Example: Multiple Joins

- $P(R)$
  
<table>
<thead>
<tr>
<th></th>
<th>+r</th>
<th>-r</th>
</tr>
</thead>
<tbody>
<tr>
<td>r</td>
<td>0.1</td>
<td>0.9</td>
</tr>
</tbody>
</table>
  
- $P(T|R)$
  
<table>
<thead>
<tr>
<th></th>
<th>+r</th>
<th>+t</th>
<th>-r</th>
<th>-t</th>
</tr>
</thead>
<tbody>
<tr>
<td>r</td>
<td>0.8</td>
<td>0.2</td>
<td>0.1</td>
<td>0.9</td>
</tr>
</tbody>
</table>
  
- $P(L|T)$
  
<table>
<thead>
<tr>
<th></th>
<th>+t</th>
<th>+l</th>
<th>-t</th>
<th>-l</th>
</tr>
</thead>
<tbody>
<tr>
<td>r</td>
<td>0.3</td>
<td>0.7</td>
<td>0.1</td>
<td>0.9</td>
</tr>
</tbody>
</table>

Join R

- $P(R, T)$
  
<table>
<thead>
<tr>
<th></th>
<th>+r</th>
<th>+t</th>
<th>-r</th>
<th>-t</th>
</tr>
</thead>
<tbody>
<tr>
<td>r</td>
<td>0.08</td>
<td>0.02</td>
<td>0.09</td>
<td>0.81</td>
</tr>
</tbody>
</table>

Join T

- $P(R, T, L)$
  
<table>
<thead>
<tr>
<th></th>
<th>+r</th>
<th>+t</th>
<th>+l</th>
<th>+r</th>
<th>+t</th>
<th>-l</th>
<th>-r</th>
<th>-t</th>
<th>+l</th>
<th>-r</th>
<th>-t</th>
<th>-l</th>
</tr>
</thead>
<tbody>
<tr>
<td>r</td>
<td>0.024</td>
<td>0.056</td>
<td>0.052</td>
<td>0.018</td>
<td>0.027</td>
<td>0.063</td>
<td>0.081</td>
<td>0.729</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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:

\[
P(R, T) \quad \text{sum } R \quad P(T)
\]

<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>-r</td>
<td>0.09</td>
<td></td>
</tr>
<tr>
<td>-r</td>
<td>0.81</td>
<td></td>
</tr>
</tbody>
</table>

<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

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

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

Sum out R

\[ P(T, L) \]

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

Sum out T

\[ P(L) \]

\[
\begin{array}{c}
+\text{l} & 0.134 \\
-\text{l} & 0.866 \\
\end{array}
\]
Thus Far: Multiple Join, Multiple Eliminate (= Inference by Enumeration)

\[
P(R) \quad \rightarrow \quad P(T|R) \quad \rightarrow \quad P(R, T, L) \quad \rightarrow \quad P(L)
\]

\[
P(L|T)
\]
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:
  \[ P(Q|e_1 \ldots e_k) \]

* Works fine with multiple query variables, too

- Step 1: Select the entries consistent with the evidence
- Step 2: Sum out \( H \) to get joint of Query and evidence
- Step 3: Normalize

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

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

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

\[ X_1, X_2, \ldots X_n \]

All variables

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

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

\[ \frac{1}{Z} \]

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

\[ \sum_{h_1} P(Q, h_1 \ldots h_r) \]
Thus Far: Multiple Join, Multiple Eliminate (= Inference by Enumeration)

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

- Compute joint
- Sum out hidden variables
- [Step 3: Normalize]
Thus Far: Multiple Join, Multiple Eliminate (= Inference by Enumeration)
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
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 \( t \)
Marginalizing Early (= Variable Elimination)
Marginalizing Early! (aka VE)

\[ P(R) \]

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

\[ P(T|R) \]

<table>
<thead>
<tr>
<th></th>
<th>+r</th>
<th>-r</th>
</tr>
</thead>
<tbody>
<tr>
<td>+r</td>
<td>0.8</td>
<td></td>
</tr>
<tr>
<td>+r</td>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td>-r</td>
<td>0.1</td>
<td></td>
</tr>
<tr>
<td>-r</td>
<td>0.9</td>
<td></td>
</tr>
</tbody>
</table>

\[ P(L|T) \]

<table>
<thead>
<tr>
<th></th>
<th>+t</th>
<th>-t</th>
</tr>
</thead>
<tbody>
<tr>
<td>+t</td>
<td>0.3</td>
<td></td>
</tr>
<tr>
<td>+t</td>
<td>0.7</td>
<td></td>
</tr>
<tr>
<td>-t</td>
<td>0.1</td>
<td></td>
</tr>
<tr>
<td>-t</td>
<td>0.9</td>
<td></td>
</tr>
</tbody>
</table>

\[ P(R, T) \]

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

Join R

\[ P(T) \]

<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>

Sum out R

Join T

\[ P(T, L) \]

<table>
<thead>
<tr>
<th></th>
<th>+t</th>
<th>-t</th>
</tr>
</thead>
<tbody>
<tr>
<td>+t</td>
<td>0.051</td>
<td></td>
</tr>
<tr>
<td>+t</td>
<td>0.119</td>
<td></td>
</tr>
<tr>
<td>-t</td>
<td>0.083</td>
<td></td>
</tr>
<tr>
<td>-t</td>
<td>0.747</td>
<td></td>
</tr>
</tbody>
</table>

Sum out T

\[ P(L) \]

<table>
<thead>
<tr>
<th></th>
<th>+l</th>
<th>-l</th>
</tr>
</thead>
<tbody>
<tr>
<td>+l</td>
<td>0.134</td>
<td></td>
</tr>
<tr>
<td>-l</td>
<td>0.866</td>
<td></td>
</tr>
</tbody>
</table>
Evidence

- If evidence, start with factors that select that evidence
  - No evidence uses 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  |
  | +t | 0.1    | +l 0.1  | 0.9     |
  | -t | 0.9    | -l 0.1  | 0.7     |

- 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  |
  | +t | 0.1    | -l 0.7     | 0.9     |
  | -t | 0.9    | +l 0.1     | 0.7     |
  | -l | 0.9    | -r 0.2     | 0.1     |

- We eliminate all vars other than query + evidence
**Evidence II**

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

  \[
  \begin{array}{ccc}
  P(+r, L) & \text{Normalize} & P(L \mid +r) \\
  +r & +l & 0.026 \\
  +r & -l & 0.074 \\
  \end{array}
  \]

  \[
  \begin{array}{ccc}
  +l & 0.26 \\
  -l & 0.74 \\
  \end{array}
  \]

- To get our answer, just normalize this!

- That’s it!
Inference by Enumeration

- **General case:**
  - Evidence variables:
  - Query* variable:
  - Hidden variables:

\[
\begin{align*}
E_1 \ldots E_k &= e_1 \ldots e_k \\
Q &= X_1, X_2, \ldots X_n \\
H_1 \ldots H_r &= \text{All variables}
\end{align*}
\]

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

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

- **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

- **Compute joint**

- **Step 3:** Normalize

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

\[
P(Q|e_1 \ldots e_k) = \frac{1}{Z} P(Q, e_1 \ldots e_k)
\]
Variable Elimination

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

  \[
  \begin{array}{c}
  \{ X_1, X_2, \ldots X_n, \text{All variables} \}
  \end{array}
  \]

- **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**

  \[
  Z = \sum_{q} P(Q, e_1 \ldots e_k)
  \]

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

* Works fine with multiple query variables, too

- Interleave joining and summing out $X_1, X_2, \ldots X_n$
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

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

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

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

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 \( uwy + uwz + uxy + uxz + vwy + vwx + vxy + vxz = (u+v)(w+x)(y+z) \) to improve computational efficiency!
Example

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

Choose A

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

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

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

Choose E

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

Finish with B

\[ P(B) \times P(j, m, B) \text{ Normalize} P(B | j, m) \]
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 \( f_1(y_1|Z) = \sum_{x_1} P(x_1|Z)P(y_1|x_1) \), and we are left with:

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

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

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

Eliminate \( Z \), this introduces the factor \( f_3(y_1, y_2, X_3) = \sum_z P(z)P(X_3|z)f_1(y_1|Z)f_2(y_2|Z) \), 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:

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

Normalizing over \( X_3 \) gives \( P(X_3|y_1, y_2, y_3) = f_4(y_1, y_2, y_3, X_3)/\sum_{x_3} f_4(y_1, y_2, y_3, x_3) \)

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 --- as they all only have one variable (\( Z \), \( Z \), and \( X_3 \) respectively).
Variable Elimination Ordering

For the query $P(X_n | y_1, \ldots, y_n)$ work through the following two different orderings as done in previous slide: $Z, X_1, \ldots, X_{n-1}$ and $X_1, \ldots, X_{n-1}, Z$. What is the size of the maximum factor generated for each of the orderings?

- Answer: $2^n$ versus 2 (assuming binary)
- In general: the ordering can greatly affect efficiency.
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:**

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

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

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

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

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

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

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

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

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

- 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.
A polytree is a directed graph with no undirected cycles.

For poly-trees you can always find an ordering that is efficient.
  Try it!!
Bayes Nets

- Representation
- Probabilistic Inference
  - Enumeration (exact, exponential complexity)
  - Variable elimination (exact, worst-case exponential complexity, often better)
  - Probabilistic inference is NP-complete
- Conditional Independences
  - Sampling
  - Learning from data