CS 188: Artificial Intelligence

Bayes’ Nets: Inference

Instructors: Dan Klein and Pieter Abbeel --- University of California, Berkeley

[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.]
Review: Independence

Conditional Independence

In 3 easy steps!
The General Case

- 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
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: \( X_i \perp\!\!\!\!\!\!\!\!\perp X_j \mid \{X_{k_1}, \ldots, X_{k_n}\} \) ?

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

\[ R \perp B \]
\[ R \perp B \mid T \]
\[ R \perp B \mid T' \]

Yes
Example

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

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

- **Questions:**
  
  \[
  T \perp D \\
  T \perp D|R \\
  T \perp D|R, S
  \]

  Yes
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

$$X_i \independent X_j | \{X_{k_1}, \ldots, X_{k_n}\}$$

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

Compute ALL THE 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:
    \[ \text{argmax}_q P(Q = q|E_1 = e_1) \]
Inference by Enumeration

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

  \[ \{ \begin{align*}
  X_1, X_2, \ldots X_n \\
  \text{All variables}
  \end{align*} \]

- **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

<table>
<thead>
<tr>
<th>( x )</th>
<th>( P(x) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.05</td>
</tr>
<tr>
<td>1</td>
<td>0.25</td>
</tr>
<tr>
<td>2</td>
<td>0.07</td>
</tr>
<tr>
<td>3</td>
<td>0.2</td>
</tr>
<tr>
<td>5</td>
<td>0.01</td>
</tr>
</tbody>
</table>

- **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?
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 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>

\[
P(cold, W)
\]

<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 II

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

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

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

$$P(rain \mid T')$$

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

$$P(rain \mid hot) = 0.2$$

$$P(rain \mid cold) = 0.6$$
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!

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

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

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

\[
\begin{array}{c|c}
+r & 0.1 \\
-r & 0.9 \\
\end{array} \quad \begin{array}{c|c|c}
+r & +t & 0.8 \\
+r & -t & 0.2 \\
-r & +t & 0.1 \\
-r & -t & 0.9 \\
\end{array} \quad \begin{array}{c|c|c}
+t & +l & 0.3 \\
+t & -l & 0.7 \\
-t & +l & 0.1 \\
-t & -l & 0.9 \\
\end{array}
\]

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

\[
P(R) \quad P(T|R) \quad P(+l|T)
\]

\[
\begin{array}{c|c}
+r & 0.1 \\
-r & 0.9 \\
\end{array} \quad \begin{array}{c|c|c}
+r & +t & 0.8 \\
+r & -t & 0.2 \\
-r & +t & 0.1 \\
-r & -t & 0.9 \\
\end{array} \quad \begin{array}{c|c|c}
+t & +l & 0.3 \\
-t & +l & 0.1 \\
\end{array}
\]

- Procedure: Join all factors, eliminate all hidden variables, normalize
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 $R$

$$P(R) \times P(T|R) \rightarrow P(R, T)$$

| $R$   | $P(R)$ | $T$   | $P(T|R)$ | $P(R, T)$ |
|-------|--------|-------|----------|-----------|
| $+r$  | 0.1    | $+r$  | 0.8      | 0.08      |
| $-r$  | 0.9    | $+t$  | 0.2      | 0.02      |
|       |        | $-r$  | 0.1      | 0.09      |
|       |        | $-r$  | 0.9      | 0.81      |

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

\[
P(R) = \begin{pmatrix} +r & 0.1 \\ -r & 0.9 \end{pmatrix}
\]

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

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

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

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

\[
P(R, T, L) = \begin{pmatrix} +r & +t & +l & 0.024 \\ +r & +t & -l & 0.056 \\ +r & -t & +l & 0.002 \\ +r & -t & -l & 0.018 \\ -r & +t & +l & 0.027 \\ -r & +t & -l & 0.063 \\ -r & -t & +l & 0.081 \\ -r & -t & -l & 0.729 \end{pmatrix}
\]
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>+r</th>
<th>+t</th>
<th>0.08</th>
</tr>
</thead>
<tbody>
<tr>
<td>+r</td>
<td>-r</td>
<td>-t</td>
<td>0.02</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>

sum \( R \)

\[
P(T)
\]

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

\[
 P(T)
\]
Multiple Elimination

- **Sum out R**

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

<table>
<thead>
<tr>
<th></th>
<th>+t</th>
<th>+l</th>
<th>Sum out R</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>0.002</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 out T**

  \[ P(T, L) \]

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

- **P(L)**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>+l</td>
<td>0.134</td>
</tr>
<tr>
<td>-l</td>
<td>0.886</td>
</tr>
</tbody>
</table>

**Diagram:**

- **R, T, L**
- **T, L**
- **L**
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 $r$
  - 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! (aka VE)

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

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

\[
\begin{align*}
P(L) & \\
+ l & 0.134 \\
- l & 0.866
\end{align*}
\]

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

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

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

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

Join R

Join T

Sum out T
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     |
| +r    |           | +t 0.1     | +l 0.1     |
| -r    |           | -t 0.9     | -l 0.9     |

- Computing \(P(L \mid + r)\) the initial factors become:

|       | \(P(+r)\) | \(P(T|+r)\) | \(P(L|T)\) |
|-------|-----------|-------------|------------|
| +r    | 0.1       | +t 0.8      | +l 0.3     |
| +r    |           | -t 0.2      | -l 0.7     |
| -r    |           | +t 0.1      | +l 0.1     |
| -r    |           | -t 0.9      | -l 0.9     |

- We eliminate all vars other than query + evidence
Result will be a selected joint of query and evidence

E.g. for $P(L \mid +r)$, we would end up with:

$$P(\neg r, L)$$

<table>
<thead>
<tr>
<th>$+r$</th>
<th>$+l$</th>
<th>0.026</th>
</tr>
</thead>
<tbody>
<tr>
<td>$+r$</td>
<td>$-l$</td>
<td>0.074</td>
</tr>
</tbody>
</table>

Normalize

$$P(L \mid +r)$$

<table>
<thead>
<tr>
<th>$+l$</th>
<th>0.26</th>
</tr>
</thead>
<tbody>
<tr>
<td>$-l$</td>
<td>0.74</td>
</tr>
</tbody>
</table>

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

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

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

Choose A

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

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

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

Finish with B

\[
\begin{align*}
P(B) & \\
P(j, m | B) & \rightarrow \ P(j, m, B) & \rightarrow & \text{Normalize} & \ P(B | j, m)
\end{align*}
\]
Same Example in Equations

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

\[
\begin{array}{c|c|c|c|c|c}
P(B) & P(E) & P(A|B,E) & P(j|A) & P(m|A) \\
\hline
P(B|j,m) & \propto & P(B,j,m) & \sum_{e,a} P(B,j,m,e,a) & \sum_{e,a} P(B)P(e)P(a|B,e)P(j|a)P(m|a) & \sum_{e} P(B)P(e)\sum_{a} P(a|B,e)P(j|a)P(m|a) \\
& & & \text{marginal obtained from joint by summing out} & \text{use Bayes’ net joint distribution expression} & \text{joining on } a, \text{ and then summing out gives } f_1 \\
& & & \text{use } x^*(y+z) = xy + xz & \text{use } x^*(y+z) = xy + xz & \text{joining on } e, \text{ and then summing out gives } f_2 \\
& & & \sum_{e} P(B)P(e)f_1(B,e,j,m) & \sum_{e} P(B)P(e)f_1(B,e,j,m) & \sum_{e} P(B)P(e)f_1(B,e,j,m) \\
& & & P(B)f_2(B,j,m) & P(B)f_2(B,j,m) & P(B)f_2(B,j,m) \\
\end{array}
\]

All we are doing is exploiting \( uwy + u wz + u xy + uxz + v w y + v w z + v x y + v x z = (u+v)(w+x)(y+z) \) 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 \( f_3(y_1,y_2,X_3) = \sum_z p(z)f_1(z,y_1)f_2(z,y_2)p(X_3|z) \), 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 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).
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.
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!!

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