Problems with high-dim. distributions

- Suppose we have \( n \) propositional symbols
- How many parameters do we need to specify \( P(X_1=x_1, \ldots, X_n=x_n) \)?

\[
\begin{array}{cccccc}
X_1 & X_2 & \ldots & X_{n-1} & X_n & P(X) \\
0 & 0 & \ldots & 0 & 0 & .01 \\
0 & 0 & \ldots & 1 & 0 & .001 \\
0 & 0 & \ldots & 1 & 1 & .213 \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
1 & 1 & \ldots & 1 & 1 & .0003 \\
\end{array}
\]

\( 2^{n-1} \) parameters! 😞
Suppose we have joint distribution $P(X_1,\ldots,X_n)$

Then

$$P(X_i = x_i) = \sum_{x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n} P(x_1,\ldots,x_n)$$

If all $X_i$ binary: How many terms?

$$2^{n-1}$$

Need because want to compute

$$P(X_1 = T | X_3 = F, X_5 = F)$$

$$P(X_1 = T, X_3 = F, X_5 = F)$$

$$P(X_3 = F, X_5 = F)$$

Marginal I. V. I.
Independent RVs

- What if RVs are independent?
  \[ P(X_1=x_1,...,X_n=x_n) = P(x_1) \times P(x_2) \times \cdots \times P(x_n) \]

- How many parameters are needed in this case?

- How about computing \( P(x_i) \)?

  \[ \text{Indep. } P(X \mid Y, Z) = P(X) \]

- Independence too strong assumption... Is there something weaker?
Key concept: Conditional independence

- Random variables $X$ and $Y$ cond. indep. given $Z$ if for all $x, y, z$:

$$P(X = x, Y = y \mid Z = z) = P(X = x \mid Z = z) \cdot P(Y = y \mid Z = z)$$

- If $P(Y = y \mid Z = z) > 0$, that’s equivalent to

$$P(X = x \mid Z = z, Y = y) = P(X = x \mid Z = z)$$

Similarly for sets of random variables $X, Y, Z$

We write: $P \models X \perp Y \mid Z$
Bayesian networks

- Compact representation of distributions over large number of variables
- (Often) allows efficient exact inference (computing marginals, etc.)

HailFinder
- 56 vars
- ~3 states each
- \( \approx 10^{26} \) terms
- > 10,000 years on Top supercomputers

JavaBayes applet
Bayesian networks

A **Bayesian network structure** is a directed, acyclic graph $G$, where each vertex $s$ of $G$ is interpreted as a random variable $X_s$ (with unspecified distribution)

A **Bayesian network** $(G,P)$ consists of
- A BN structure $G$ and ..
- ..a set of conditional probability distributions (CPTs) $P(X_s \mid \text{Pa}_{X_s})$, where $\text{Pa}_{X_s}$ are the parents of node $X_s$ such that $(G,P)$ defines joint distribution

$$P(X_1, \ldots, X_n) = \prod_{i} P(X_i \mid \text{Pa}_{X_i})$$
Representing the world using BNs

True distribution $P'$ with cond. ind. $I(P')$

- Want to make sure that $I(P)$ is a subset of $I(P')$
- Need to understand conditional independence properties of BN $(G,P)$

Bayes net $(G,P)$ with $I(P)$
BNs with 3 nodes

$x \perp \not\| y \mid (x, z)$

$x \perp \not\| y \mid l(x, z)$
When are A and I independent?

\[
\begin{align*}
A & \perp I \\
A & \perp I \mid D \\
A & \perp I \mid DH \\
A & \perp I \mid H, F
\end{align*}
\]
Active trails

- An undirected path in BN structure $G$ is called **active trail** for observed variables $O \in \{X_1,...,X_n\}$, if for every consecutive triple of vars $X,Y,Z$ on the path:
  - $X \rightarrow Y \rightarrow Z$ and $Y$ is unobserved ($Y \notin O$)
  - $X \leftarrow Y \leftarrow Z$ and $Y$ is unobserved ($Y \notin O$)
  - $X \leftarrow Y \rightarrow Z$ and $Y$ is unobserved ($Y \notin O$)
  - $X \rightarrow Y \leftarrow Z$ and $Y$ or any of $Y$’s descendants is observed

- Any variables $X_i$ and $X_j$ for which there is no active trail for observations $O$ are called **d-separated** by $O$
  - We write $d$-sep($X_i ; X_j \mid O$)

- Sets $A$ and $B$ are d-separated given $O$ if $d$-sep($X,Y \mid O$) for all $X$ in $A$, $Y$ in $B$. Write $d$-sep($A; B \mid O$)
Theorem:
\[ \text{d-sep}(X; Y \mid Z) \Rightarrow X \perp Y \mid Z \]

i.e., X cond. indep. Y given Z if there does not exist any active trail between X and Y for observations Z

- Converse does not hold in general!
- But for “almost” all distributions (except set of measure 0)
Examples

A \perp G
A \perp G \mid F, x
A \perp G \mid F, H, \checkmark
A \perp G \mid F, H, \checkmark, E, \checkmark
More examples
Algorithm for d-separation

How can we check if d-sep($X; Y \mid Z$)?

Idea: Check every possible path connecting $X$ and $Y$ and verify conditions

- Exponentially many paths!!! 😞

Linear time algorithm:
Find all nodes reachable from $X$

1. Mark $Z$ and its ancestors
2. Do breadth-first search starting from $X$; stop if path is blocked

Have to be careful with implementation details (see reading)
Typical queries: Conditional distribution

Compute distribution of some variables given values for others

\[ P(E | J = T) \ ? \]
\[ P(E, B | J = T, M = F) \ ? \]
\[ = \frac{1}{2} P(E, B, J = T, M = F) \]
Typical queries: Maximization

- **MPE (Most probable explanation):**
  Given values for some vars, compute most likely assignment to all remaining vars
  \[
  (a^*, e^*, b^*) = \arg \max_{a, b, e} P(E=e, B=b, A=a | J=T, M=F)
  \]

- **MAP (Maximum a posteriori):**
  Compute most likely assignment to some variables
  \[
  (e^*, B^*) = \arg \max_{e, b} P(e, b | J=T, M=F)
  \]

More general than MPE
Hardness of inference for general BNs

Computing conditional distributions:
- Exact solution: #P-complete
- NP-hard to obtain any nontrivial approximation
  \[ \text{E.g., NP-hard to obtain } \hat{P} \text{ s.t. } |P - \hat{P}| < \frac{1}{2} \]

Maximization:
- MPE: NP-complete
- MAP: \( \text{NP}^{\text{PP}} \)-complete

Inference in general BNs is really hard 😞

Is all hope lost?
Inference

- Can exploit structure (conditional independence) to efficiently perform **exact inference** in many practical situations

- For BNs where exact inference is not possible, can use algorithms for **approximate inference** (later)
Potential for savings: Variable elimination!

\[ P(x_1, x_2, \ldots, x_5) = P(x_1) P(x_2 | x_1) P(x_3 | x_2) \ldots P(x_5 | x_4) \]

So if we want \( P(x_5 | x_1) = \frac{1}{2} P(x_1, x_5) \)

\[ P(x_1, x_5) = \sum_{x_2} \sum_{x_3} P(x_1) P(x_2 | x_1) P(x_3 | x_2) \sum_{x_4} P(x_4 | x_3) P(x_5 | x_4) \]

For \( n \) vars:

- Naïve: \( 2^{n-2} \)
- VE: \( 2 \cdot (n-2) \)

Intermediate solutions are distributions on fewer variables!
Variable elimination in general graphs

- Push sums through product as far as possible
- Create new factor by summing out variables

\[
P(E|m) = \frac{1}{\mathcal{Z}} P(E,m) \\
= \sum_{b,a,i} P(E|m,b,a,i) \\
= \sum_{b,a} P(E) P(b) P(a|E,b) P(y|a) P(m|a) \\
= \sum_{b,a} P(E) P(b) P(a|E,b) P(m|a) \sum_{y} \mathcal{g}_6(a,E) \\
= P(E) \sum_{a} P(m|a) \mathcal{g}_6(a,E)
\]
Variable elimination algorithm

- Given BN and Query $P(X \mid E=e)$
- Choose an ordering of $X_1,\ldots,X_n$
- Set up initial factors: $f_i = P(X_i \mid Pa_i)$
- For $i = 1:n$, $X_i \not\in \{X, E\}$
  - Collect all factors $f$ that include $X_i$
  - Generate new factor by marginalizing out $X_i$
    
    $$g = \sum_{x_i} \prod_j f_j$$
  
  - Add $g$ to set of factors
- Renormalize $P(x, e)$ to get $P(x \mid e)$
Multiplying factors

\[ g = \sum \prod_{x_i} f_j \]

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>( f_1(A,B) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>.1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>.3</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>.7</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>.01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>B</th>
<th>C</th>
<th>( f_2(B,C) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>.4</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>.2</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>.5</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

\[ f' = f_1 \cdot f_2 \]
Marginalizing factors

\[ g = \sum_{i} \prod_{j} f_{j} \]

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>( f'(A,B) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>.1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>.3</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>.7</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>.01</td>
</tr>
</tbody>
</table>

\[ g = \sum_{u} f' \]

\[ \begin{array}{c|cc}
B & g(B) \\
\hline
0 & .1 + .7 \\
1 & .3 + .01 \\
\end{array} \]
The order matters!

\[ P(A,B,E,J,M) = P(E) \cdot P(B) \cdot P(A|E,B) \cdot P(J|A) \cdot P(M|A) \]

What if we eliminate A first?

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

\[ g_{a}(E,B,J,M) \]
Variable elimination for polytrees

A DAG is a polytree iff dropping edge directions results in a tree

Eliminate in increasing order of degree in undirected graph

Pick root, orient edges away from root use inverse topol. ordering
What about loops?

- Can do efficient inference on trees.
- What if the graph has loops?
Suppose we would like to compute $P(X_i \mid E=e)$.

Pick subset of variables $A$ (called “cutset”) such that remaining variables form a tree.

Calculate $P(X_i, A=a \mid E=e)$ for each assignment $A=a$.

Then $P(X_i \mid E=e) = \sum_a P(X_i, A=a \mid E=e)$.

Analog to Constraint Satisfaction Problems
Answering multiple queries

Suppose, I would like \( P(X_i \mid X_n = T) \) for all \( i \)

Naïve approach?

- Run VI for each \( i \)
  - Cost \( \Theta(n) \) for each \( i \)
  - \( \Rightarrow \Theta(n^2) \)
Reusing computation

\[ P(x_1, x_n) = \sum_{x_2 \ldots x_{n-2}} P(x_i) P(x_{n-1} | x_{n-2}) \ldots P(x_n | x_{n-2}) \sum_{x_{n-3}} P(x_{n-2} | x_{n-3}) \]

\[ P(x_2, x_m) = \sum_{x_1, x_3, \ldots, x_{n-2}} P(x_i) \ldots \sum_{x_{n-1}} P(x_{n-1} | x_{n-2}) P(x_n | x_{n-1}) \]
Reusing computation

- Often, want to compute conditional distributions of many variables, for fixed observations
- E.g., probability of *Pits* at different locations given observed *Breezes*
- Repeatedly performing variable elimination is wasteful (many factors are recomputed)
- Need right data-structure to avoid recomputation
  - Message passing on factor graphs
Factor graphs

\[ P(C, D, G, I, S, L) = P(C) \ P(I) \ P(D \mid C) \ P(G \mid D, I) \ P(S \mid I, G) \ P(L \mid S) \]

\[ f_1(c_1, d) = P(c) \ P(d \mid c) \]
\[ f_2(d, i, g) = P(i) \ P(g \mid d, i) \]
\[ f_3(i, g, s) = P(s \mid i, g) \]
\[ f_4(s, l) = P(l \mid s) \]
A factor graph for a Bayesian network is a bipartite graph consisting of

- Variables and
- Factors

Each factor is associated with a subset of variables, and all CPDs of the Bayesian network have to be assigned to one of the factor nodes.