Probabilistic Reasoning

School of EECS
Washington State University
Probabilistic Reasoning

- Full joint probability distribution
  - Can answer any query
  - But typically too large
- Conditional independence
  - Can reduce the number of probabilities needed
  - \( P(X \mid Y,Z) = P(X \mid Z) \), if \( X \) independent of \( Y \) given \( Z \)
- Bayesian network
  - Concise representation of above
Bayesian Network

Example

![Bayesian Network Diagram]

- $P(B) = 0.001$
- $P(E) = 0.002$

<table>
<thead>
<tr>
<th>B</th>
<th>E</th>
<th>$P(A)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>t</td>
<td>t</td>
<td>0.95</td>
</tr>
<tr>
<td>t</td>
<td>f</td>
<td>0.94</td>
</tr>
<tr>
<td>f</td>
<td>t</td>
<td>0.29</td>
</tr>
<tr>
<td>f</td>
<td>f</td>
<td>0.001</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A</th>
<th>$P(J)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>t</td>
<td>0.90</td>
</tr>
<tr>
<td>f</td>
<td>0.05</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A</th>
<th>$P(M)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>t</td>
<td>0.70</td>
</tr>
<tr>
<td>f</td>
<td>0.01</td>
</tr>
</tbody>
</table>
Bayesian Network

- Bayesian network is a directed, acyclic graph
- Each node corresponds to a random variable
- A directed link from node X to node Y implies that X “influences” Y
  - X is the parent of Y
- Each node X has a conditional probability distribution $P(X \mid \text{Parents}(X))$
  - Quantifies the influence on X from its parent nodes
  - Conditional probability table (CPT)
Bayesian Networks

- Represents full joint distribution

\[ P(X_1 = x_1 \land \ldots \land X_n = x_n) = \prod_{i=1}^{n} P(X_i = x_i \mid \text{parents}(X_i)) \]

\[ P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i \mid \text{parents}(X_i)) \]

- Represents conditional independence
  - E.g., JohnCalls is independent of Burglary and Earthquake given Alarm
Bayesian Networks

- \[ P(b, \neg e, a, j, m) = (0.001)(0.998)(0.94)(0.90)(0.70) = 0.000591 \]
Constructing Bayesian Networks

- Determine set of random variables \( \{X_1, \ldots, X_n\} \)
- Order them so that causes precede effects
- For \( i = 1 \) to \( n \) do
  - Choose minimal set of parents for \( X_i \) such that \( P(X_i \mid X_{i-1}, \ldots, X_1) = P(X_i \mid \text{Parents}(X_i)) \)
  - For each parent \( X_k \) insert link from \( X_k \) to \( X_i \)
  - Write down the CPT, \( P(X_i \mid \text{Parents}(X_i)) \)

- E.g., Burglary, Earthquake, Alarm, JohnCalls, MaryCalls
Constructing Bayesian Networks

- Bad orderings lead to more complex networks with more CPT entries
  a) MaryCalls, JohnCalls, Alarm, Burglary, Earthquake
  b) MaryCalls, JohnCalls, Earthquake, Burglary, Alarm
Example: Tooth World

- Variables: Cavity, Toothache, Catch

- Probabilities:
  - P(Toothache) = 0.6
  - P(Cavity) = 0.2
  - P(Catch) = 0.9

- Truth values:
  - Toothache: true, false
  - Catch: true, false
  - Cavity: true, false

- Joint probability table:

<table>
<thead>
<tr>
<th>Cavity</th>
<th>P(Toothache)</th>
</tr>
</thead>
<tbody>
<tr>
<td>t</td>
<td>0.6</td>
</tr>
<tr>
<td>f</td>
<td>0.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cavity</th>
<th>P(Catch)</th>
</tr>
</thead>
<tbody>
<tr>
<td>t</td>
<td>0.9</td>
</tr>
<tr>
<td>f</td>
<td>0.2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>toothache</th>
<th>¬toothache</th>
<th>catch</th>
<th>¬catch</th>
<th>catch</th>
<th>¬catch</th>
</tr>
</thead>
<tbody>
<tr>
<td>cavity</td>
<td>.108</td>
<td>.012</td>
<td>.072</td>
<td>.008</td>
<td></td>
</tr>
<tr>
<td>¬cavity</td>
<td>.016</td>
<td>.064</td>
<td>.144</td>
<td>.576</td>
<td></td>
</tr>
</tbody>
</table>
Node X is conditionally independent of its non-descendants ($Z_{ij}$’s) given its parents ($U_i$’s)

**Markov blanket** of node X is X’s parents ($U_i$’s), children ($Y_i$’s) and children’s parents ($Z_{ij}$’s)

Node X is conditionally independent of all other nodes in the network given its Markov blanket
Inference in Bayesian Networks

- Want $P(X \mid e)$
- $X$ is the **query variable** (can be more than one)
- $e$ is an observed event, i.e., values for the evidence variables $E = \{E_1, \ldots, E_m\}$
- Any other variables $Y$ are hidden variables

**Example**

- $P(\text{Burglary} \mid \text{JohnCalls}=\text{true}, \text{MaryCalls}=\text{true}) = ?$
- $X = \text{Burglary}$
- $e = \{\text{JohnCalls}=\text{true}, \text{MaryCalls}=\text{true}\}$
- $Y = \{\text{Earthquake}, \text{Alarm}\}$
Inference by Enumeration

- Enumerate over all possible values for \( Y \)
  - \( P(X \mid e) = \alpha P(X,e) = \alpha \sum_Y P(X,e,y) \)

- Example
  - \( P(\text{Burglary} \mid \text{JohnCalls}=true, \text{MaryCalls}=true) \)
  - \( P(B \mid j,m) = \alpha P(B,j,m) = \alpha \sum_e \sum_a P(B,j,m,e,a) \)
  - Let \( b \) represent \( (B=true) \)
  - \( P(b \mid j,m) = \alpha \sum_e \sum_a P(b)P(e)P(a\mid b,e)P(j\mid a)P(m\mid a) \)
  - \( P(b \mid j,m) = \alpha P(b) \sum_e P(e)\sum_a P(a\mid b,e)P(j\mid a)P(m\mid a) \)
Inference by Enumeration

- \( P(b \mid j,m) = \alpha P(b) \sum_e P(e) \sum_a P(a \mid b,e)P(j \mid a)P(m \mid a) \)
- \( P(B \mid j,m) = \alpha \langle 0.0005922, 0.0014919 \rangle = \langle 0.284, 0.716 \rangle \)
Inference by Enumeration

**function** `ENUMERATION-ASK (X, e, bn)` **returns** a distribution over X

**inputs:** X, the query variable
e, observed values of variables E
bn, a Bayes net with variables \{X\} \cup E \cup Y \quad // \text{Y = hidden variables}

Q(X) ← a distribution over X, initially empty

**for each** value \(x_i\) of \(X\) **do**

\(Q(x_i) \leftarrow \text{ENUMERATE-ALL}(bn.\text{VARS}, e_{x_i})\)

where \(e_{x_i}\) is \(e\) extended with \(X = x_i\)

**return** \(\text{NORMALIZE}(Q(X))\)

\begin{itemize}
  \item `bn.\text{VARS}` has variables in cause→effect order
\end{itemize}

**function** `ENUMERATE-ALL (vars, e)` **returns** a real number

**if** `EMPTY? (vars)` **then return** 1.0

\(Y \leftarrow \text{FIRST}(vars)\)

**if** \(Y\) has value \(y\) in \(e\)

**then return** \(P(y | parents(Y)) \times \text{ENUMERATE-ALL}((\text{REST}(vars)), e)\)

**else return** \(\sum_y P(y | parents(Y)) \times \text{ENUMERATE-ALL}((\text{REST}(vars)), e_y)\)

where \(e_y\) is \(e\) extended with \(Y = y\)
Inference by Enumeration

- **ENUMERATION-ASK** evaluates trees using depth-first recursion
- Space complexity $O(n)$
- Time complexity $O(v^n)$, where each of $n$ variables has $v$ possible values
Inference by Enumeration

Note redundant computation
Efficient Inference

- Avoid redundant computation
  - Dynamic programming
  - Store intermediate computations and reuse
- Eliminate irrelevant variables
  - Variables that are not an ancestor of a query or evidence variable
Complexity of Inference

- General case (any type of network)
  - Worst case space and time complexity is exponential
- **Polytree** is a network with at most one undirected path between any two nodes
  - Space and time complexity is linear in size of network
Approximate Inference

- “Improbability Drive”

Approximate Inference

- Exact inference can be too expensive
- Approximate inference
  - Estimate probabilities from sample, rather than computing exactly
- Monte Carlo methods
  - Choose values for hidden variables, compute query variables, repeat and average
- Direct sampling
- Markov chain sampling
Direct Sampling

- Choose value for variables according to their CPT
  - Consider variables in topological order

- E.g.,
  - \( P(B) = \langle 0.001, 0.999 \rangle, \ B=\text{false} \)
  - \( P(E) = \langle 0.002, 0.998 \rangle, \ E=\text{false} \)
  - \( P(A|B=\text{false}, E=\text{false}) = \langle 0.001, 0.999 \rangle, \ A=\text{false} \)
  - \( P(J|A=\text{false}) = \langle 0.05, 0.95 \rangle, \ J=\text{false} \)
  - \( P(M|A=\text{false}) = \langle 0.01, 0.99 \rangle, \ M=\text{false} \)
  - Sample is \([\text{false}, \text{false}, \text{false}, \text{false}, \text{false}, \text{false}, \text{false}]\)

\[
P(X = x_i) \approx \frac{|\text{samples where } X = x_i|}{|\text{samples}|}
\]
Markov Chain Sampling

- New sample generated by random changes to preceding sample
  - Not generated from scratch
- Markov Chain Monte Carlo (MCMC)
- Gibbs sampling
  - Fix evidence variables to observed values
  - Randomly set non-evidence variables
  - Repeat for desired number of samples
    - Randomly choose a non-evidence variable X
    - Randomly choose new value for X based on CPT
    - Generate sample
Gibbs sampling example

E.g., \( P(B|J=\text{true}, M=\text{true}) \)

- Evidence: \( J=\text{true}, M=\text{true} \)
- Random initial state: \( B=\text{true}, E=\text{false}, A=\text{true} \)
- \( P(B) = \langle 0.001, 0.999 \rangle, B=\text{false} \)
  - Sample 1: [false, false, true, true, true]
- \( P(E) = \langle 0.002, 0.998 \rangle, E=\text{false} \)
  - Sample 2: [false, false, true, true, true]
- \( P(A|B=\text{false}, E=\text{false}) = \langle 0.001, 0.999 \rangle, A=\text{false} \)
  - Sample 3: [false, false, false, true, true]
- Repeat for more \( B \)'s, \( E \)'s and \( A \)'s (any order)
Approximate Inference

- Sampling techniques converge to correct probabilities given enough samples
Bayes Net Software

- Commercial
  - Bayes Server (www.bayesserver.com)
  - BayesiaLab (www.bayesia.com)
  - Netica APIs (www.norsys.com)
  - HUGIN (www.hugin.com)

- Free
  - BayesPy (www.bayespy.org)
  - JavaBayes (www.cs.cmu.edu/~javabayes)
  - SMILE (www.bayesfusion.com)

- Sample networks
  - www.bnlearn.com/bnrepository
Summary: Probabilistic Reasoning

- Bayesian networks
  - Captures full joint probability distribution and conditional independence
- Exact inference
  - Intractable in worst case
- Approximate inference
  - Sampling
  - Converges to exact inference