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

\[
P(+b, -e, +a, -j, +m) = P(+b)P(-e)P(+a|-e, b)P(-j|a)P(m|a)
\]
Example: Alarm Network

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

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

\[
0.001 \times 0.998 \times 0.94 \times 0.1 \times 0.7
\]
Announcements

- Remaining lectures:
  - Today: Inference in BNs
  - Fri: Machine Learning and Neural Net Overview
  - Next Wed: More applied
    - Sequential Neural vs. HMMs
    - Application: Language models and Machine Translation
  - Next Fri: Poster session
    - Stay tuned: might do it virtual
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:
    \[ \text{argmax}_q P(Q = q|E_1 = e_1 \ldots) \]
Inference by Enumeration in Bayes’ Net

- Given unlimited time, inference in BNs is easy

\[ P(B \mid +j, +m) \propto_B P(B, +j, +m) \]
\[ = \sum_{e,a} P(B, e, a, +j, +m) \]
\[ = \sum_{e,a} P(B)P(e)P(a \mid B, e)P(+j \mid a)P(+m \mid a) \]
\[ = P(B)P(+e)P(+a \mid B, +e)P(+j \mid +a)P(+m \mid +a) + P(B)P(+e)P(-a \mid B, +e)P(+j \mid -a)P(+m \mid -a) \]
\[ + P(B)P(-e)P(+a \mid B, -e)P(+j \mid +a)P(+m \mid +a) + P(B)P(-e)P(-a \mid B, -e)P(+j \mid -a)P(+m \mid -a) \]
### Example: Traffic Domain

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

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

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

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

- Track objects called factors
- Initial factors are local CPTs (one per node)
  
  \[
  P(R) = \begin{cases} 
  +r & 0.1 \\
  -r & 0.9 
  \end{cases} \\
  P(T|R) = \begin{cases} 
  +r & +t & 0.8 \\
  +r & -t & 0.2 \\
  -r & +t & 0.1 \\
  -r & -t & 0.9 
  \end{cases} \\
  P(L|T) = \begin{cases} 
  +t & +l & 0.3 \\
  +t & -l & 0.7 \\
  -t & +l & 0.1 \\
  -t & -l & 0.9 
  \end{cases}
  
  \]

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

\[
  P(R) = \begin{cases} 
  +r & 0.1 \\
  -r & 0.9 
  \end{cases} \\
  P(T|R) = \begin{cases} 
  +r & +t & 0.8 \\
  +r & -t & 0.2 \\
  -r & +t & 0.1 \\
  -r & -t & 0.9 
  \end{cases} \\
  P(+l|T) = \begin{cases} 
  +t & +l & 0.3 \\
  -t & +l & 0.1 
  \end{cases}
  
  \]

- 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

\[
\begin{align*}
P(R) & \quad \times \quad P(T | R) \\
\begin{array}{c|c}
+r & 0.1 \\
-r & 0.9 \\
\end{array} & \quad \begin{array}{c|cc}
+r & +t & 0.8 \\
+r & -t & 0.2 \\
-r & +t & 0.1 \\
-r & -t & 0.9 \\
\end{array} \\
\Rightarrow P(R, T) \\
\begin{array}{c|cc}
+r & +t & 0.08 \\
+r & -t & 0.02 \\
-r & +t & 0.09 \\
-r & -t & 0.81 \\
\end{array}
\end{align*}
\]

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

\[ P(R) \quad P(T \bowtie R) \quad P(T, R) \]

\[ P(1) \quad P(1) \]

\[ \text{Diagram of multiple joins} \]
Example: Multiple Joins

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

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

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

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

\[
P(R, T, L) = \begin{array}{c|c|c|c}
+r & +t & +l & 0.024 \\
+r & +t & -l & 0.056 \\
+r & -t & +l & 0.002 \\
+r & -t & -l & 0.018 \\
-r & +t & +l & 0.027 \\
-r & +t & -l & 0.063 \\
-r & -t & +l & 0.081 \\
-r & -t & -l & 0.729 \\
\end{array}
\]
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:

\[
P(R, T)\]

\[
\begin{array}{ccc}
+r & +t & 0.08 \\
+r & -t & 0.02 \\
-r & +t & 0.09 \\
-r & -t & 0.81 \\
\end{array}
\]

\[\text{sum } R \]

\[
P(T)\]

\[
\begin{array}{ccc}
+t & 0.17 \\
-t & 0.83 \\
\end{array}
\]
Multiple Elimination

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

\[
\begin{array}{|c|c|c|c|}
\hline
& +t & +l & 0.024 \\
+R & +t & -l & 0.056 \\
+R & -t & +l & 0.002 \\
+R & -t & -l & 0.018 \\
-R & +t & +l & 0.027 \\
-R & +t & -l & 0.063 \\
-R & -t & +l & 0.081 \\
-R & -t & -l & 0.729 \\
\hline
\end{array}
\]

Sum out R

\[ P(T, L) \]

\[
\begin{array}{|c|c|c|}
\hline
& +l & 0.051 \\
+t & +l & 0.119 \\
+t & -l & 0.083 \\
-t & +l & 0.747 \\
-t & -l & 0.866 \\
\hline
\end{array}
\]

Sum out T

\[ P(L) \]

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

\[ P(R) \]

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

\[ P(L|T) \]
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
Marginalizing Early! (aka VE)

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

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

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

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

\[
P(L|T')
\]
\begin{array}{c|c|c}
+ 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}
+ t & + l & 0.051 \\
+ 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}
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 \( t \)
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**
If evidence, start with factors that select that evidence

- No evidence uses these initial factors:

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

P(T|R) & \\
+ r & : + t : 0.8 \\
+ r & : - t : 0.2 \\
- r & : + t : 0.1 \\
- r & : - t : 0.9 \\

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

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

\[
\begin{align*}
P(+ r) & \\
+ r & : 0.1 \\

P(T | + r) & \\
+ r & : + t : 0.8 \\
+ r & : - t : 0.2 \\

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

- 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:

$$P(+r, L)$$

<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!
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

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

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

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

- **Computes joint** $X_1, X_2, \ldots X_n$

- **Sum out hidden variables**

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

- **General case:**
  - Evidence variables: $E_1 \ldots E_k = e_1 \ldots e_k$
  - Query* variable: $Q$
  - Hidden variables: $H_1 \ldots H_r$
  - $X_1, X_2, \ldots X_n$

- **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

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

- Interleave joining and summing out

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

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

* Works fine with multiple query variables, too
Example

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

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

Choose A

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

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

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

Choose E

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

Finish with B

\[
\begin{align*}
P(B) & \quad \times \quad P(j, m | B) \\
P(j, m | B) & \quad \implies \quad P(j, m, B) \quad \text{Normalize} \quad P(B | j, m)
\end{align*}
\]

Normalize
Example

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

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

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

marginal can be obtained from joint by summing out use Bayes’ net joint distribution expression

joining on \(a\), and then summing out gives \(f_1\)

joining on \(e\), and then summing out gives \(f_2\)
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:** $2^n$ versus 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!
Variable Elimination

- Interleave joining and marginalizing
- $d^k$ entries computed for a factor over $k$ variables with domain sizes $d$
- Ordering of elimination of hidden variables can affect size of factors generated
- Worst case: running time exponential in the size of the Bayes’ net
Approximate Inference: Sampling
Sampling

- Sampling is a lot like repeated simulation
  - Predicting the weather, basketball games, ...

- Basic idea
  - Draw N samples from a sampling distribution S
  - Compute an approximate probability

- Why sample?
  - Learning: get samples from a distribution you don’t know
  - Inference: getting a sample is faster than computing the right answer
Sampling in Bayes’ Nets

- Prior Sampling
- Rejection Sampling
- Likelihood Weighting
Prior Sampling
Prior Sampling

$P(S|C)$

$P(C)$

$P(R|C)$

$P(W|S, R)$

Samples:

$+c$, $-s$, $+r$, $+w$

$-c$, $+s$, $-r$, $+w$

...
Prior Sampling

- For $i = 1, 2, \ldots, n$
  - Sample $x_i$ from $P(X_i | \text{Parents}(X_i))$
  - Return $(x_1, x_2, \ldots, x_n)$
- We’ll get a bunch of samples from the BN:
  - $+c, -s, +r, +w$
  - $+c, +s, +r, +w$
  - $-c, +s, +r, -w$
  - $+c, -s, +r, +w$
  - $-c, -s, -r, +w$

- If we want to know $P(W)$:
  - We have counts $<+w:4, -w:1>$
  - Normalize to get $P(W) = <+w:0.8, -w:0.2>$
  - This will get closer to the true distribution with more samples
  - Can estimate anything else, too
  - What about $P(C \mid +w)$? $P(C \mid +r, +w)$? $P(C \mid -r, -w)$?
  - Fast: can use fewer samples if less time (what’s the drawback?)
Rejection Sampling
Rejection Sampling

- Let’s say we want $P(C)$
  - No point keeping all samples around
  - Just tally counts of $C$ as we go

- Let’s say we want $P(C \mid +s)$
  - Same thing: tally $C$ outcomes, but ignore (reject) samples which don’t have $S=+s$
  - This is called rejection sampling
  - It is also consistent for conditional probabilities (i.e., correct in the limit)
Rejection Sampling

- **Input:** evidence instantiation
- **For** \( i = 1, 2, \ldots, n \)
  - Sample \( x_i \) from \( P(X_i \mid \text{Parents}(X_i)) \)
  - If \( x_i \) not consistent with evidence
    - Reject: return – no sample is generated in this cycle
- **Return** \( (x_1, x_2, \ldots, x_n) \)
Likelihood Weighting
Problem with rejection sampling:
- If evidence is unlikely, rejects lots of samples
- Evidence not exploited as you sample
- Consider $P(\text{Shape} \mid \text{blue} )$

Idea: fix evidence variables and sample the rest
- Problem: sample distribution not consistent!
- Solution: weight by probability of evidence given parents

- pyramid, green
- pyramid, red
- sphere, blue
- cube, red
- sphere, green

- pyramid, blue
- pyramid, blue
- sphere, blue
- cube, blue
- sphere, blue
Likelihood Weighting

\[
P(C) = \begin{array}{c|c}
+c & 0.5 \\
-c & 0.5 \\
\end{array}
\]

\[
P(S|C) = \begin{array}{c|c|c}
+c & +s & 0.1 \\
+c & -s & 0.9 \\
-c & +s & 0.5 \\
-c & -s & 0.5 \\
\end{array}
\]

\[
P(R|C) = \begin{array}{c|c|c}
+c & +r & 0.8 \\
+c & -r & 0.2 \\
-c & +r & 0.2 \\
-c & -r & 0.8 \\
\end{array}
\]

\[
P(W|S, R) = \begin{array}{c|c|c|c|c|c|c}
+s & +r & +w & 0.99 & -w & 0.01 & \\
+s & -r & +w & 0.90 & -w & 0.10 & \\
-s & +r & +w & 0.90 & -w & 0.10 & \\
-s & -r & +w & 0.01 & -w & 0.99 & \\
\end{array}
\]

Samples:
\[ +c, +s, +r, +w \]
\[ \ldots \]

\[
w = 1.0 \times 0.1 \times 0.99
\]
Likelihood Weighting

- **Input:** evidence instantiation
- **w = 1.0**
- for $i = 1, 2, \ldots, n$
  - if $X_i$ is an evidence variable
    - $X_i =$ observation $x_i$ for $X_i$
    - Set $w = w \times P(x_i | \text{Parents}(X_i))$
  - else
    - Sample $x_i$ from $P(X_i | \text{Parents}(X_i))$
- **return** $(x_1, x_2, \ldots, x_n), w$
Likelihood Weighting

- Sampling distribution if z sampled and e fixed evidence

\[ S_{WS}(z, e) = \prod_{i=1}^{l} P(z_i | \text{Parents}(Z_i)) \]

- Now, samples have weights

\[ w(z, e) = \prod_{i=1}^{m} P(e_i | \text{Parents}(E_i)) \]

- Together, weighted sampling distribution is consistent

\[ S_{WS}(z, e) \cdot w(z, e) = \prod_{i=1}^{l} P(z_i | \text{Parents}(z_i)) \prod_{i=1}^{m} P(e_i | \text{Parents}(e_i)) \]

\[ = P(z, e) \]
Likelihood Weighting

- Likelihood weighting is good
  - We have taken evidence into account as we generate the sample
  - E.g. here, W’s value will get picked based on the evidence values of S, R
  - More of our samples will reflect the state of the world suggested by the evidence