Announcements

- Midterm 1 done!
- Mini-Contest 1
  - Ends 7/16 at 11:59pm
- Project 3: RL
  - Due Friday 7/17 at 11:59pm
- Homework 4
  - Due Friday 7/17 at 11:59pm
CS 188: Artificial Intelligence

Bayes’ Nets: Inference

Instructor: Nikita Kitaev --- 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.]
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

**Burglary**

<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>

**Earthquake**

<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>

**John calls**

**Mary calls**

**Alarm**

| 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

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

\[
P(+b, -e, +a, -j, +m) = P(+b)P(-e)P(+a| b, -e)P(-j| +a)P(+m| +a) = 0.001 \times 0.998 \times 0.94 \times 0.1 \times 0.7
\]
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
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

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

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

Step 1: Select the entries consistent with the evidence

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

Step 3: Normalize

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

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

* Works fine with multiple query variables, too
Inference by Enumeration in Bayes’ Net

- Given unlimited time, inference in BNs is easy
- Reminder of inference by enumeration by example:

$$P(B | +j, +m) \propto_B P(B, +j, +m)$$

$$= \sum_{e,a} P(B, e, a, +j, +m)$$

$$= \sum_{e,a} P(B)P(e)P(a|B,e)P(+j|a)P(+m|a)$$

$$= P(B)P(+e)P(+a|B,+e)P(+j| + a)P(+m| + a) + P(B)P(+e)P(-a|B,+e)P(+j| - a)P(+m| - a)$$

$$P(B)P(-e)P(+a|B,-e)P(+j| + a)P(+m| + a) + P(B)P(-e)P(-a|B,-e)P(+j| - a)P(+m| - a)$$
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
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

\[
\begin{array}{c|cc|c}
T & W & P \\
\hline
\text{hot} & \text{sun} & 0.4 \\
\text{hot} & \text{rain} & 0.1 \\
\text{cold} & \text{sun} & 0.2 \\
\text{cold} & \text{rain} & 0.3 \\
\end{array}
\]
Factor Zoo II

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

\[
P(rain\mid hot) - P(rain\mid cold)\]
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!

\[
P(L) = ?
\]

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

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

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

\[
\begin{array}{|c|c|} \hline
P(R) & \hline
+r & 0.1 \\
-r & 0.9 \\
\hline
\end{array}
\quad
\begin{array}{|c|c|} \hline
P(T|R) & \hline
+r & +t & 0.8 \\
+r & -t & 0.2 \\
-r & +t & 0.1 \\
-r & -t & 0.9 \\
\hline
\end{array}
\quad
\begin{array}{|c|c|} \hline
P(L|T) & \hline
+t & +l & 0.3 \\
+t & -l & 0.7 \\
-t & +l & 0.1 \\
-t & -l & 0.9 \\
\hline
\end{array}
\]

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

\[
\begin{array}{|c|c|} \hline
P(R) & \hline
+r & 0.1 \\
-r & 0.9 \\
\hline
\end{array}
\quad
\begin{array}{|c|c|} \hline
P(T|R) & \hline
+r & +t & 0.8 \\
+r & -t & 0.2 \\
-r & +t & 0.1 \\
-r & -t & 0.9 \\
\hline
\end{array}
\quad
\begin{array}{|c|c|} \hline
P(+l|T) & \hline
+t & +l & 0.3 \\
+t & -l & 0.7 \\
-t & +l & 0.1 \\
-t & -l & 0.9 \\
\hline
\end{array}
\]

- 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

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

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

<table>
<thead>
<tr>
<th></th>
<th>+r</th>
<th>+t</th>
<th>0.8</th>
</tr>
</thead>
<tbody>
<tr>
<td>+r</td>
<td>+t</td>
<td>-t</td>
<td>0.2</td>
</tr>
<tr>
<td>-r</td>
<td>+t</td>
<td>+t</td>
<td>0.1</td>
</tr>
<tr>
<td>-r</td>
<td>-t</td>
<td>+t</td>
<td>0.02</td>
</tr>
<tr>
<td>-r</td>
<td>-t</td>
<td>+t</td>
<td>0.09</td>
</tr>
<tr>
<td>-r</td>
<td>-t</td>
<td>-t</td>
<td>0.81</td>
</tr>
</tbody>
</table>

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

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

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

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

Join R

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

Join T

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

$$P(R, T)$$

<table>
<thead>
<tr>
<th></th>
<th>+r</th>
<th>+t</th>
<th>0.08</th>
</tr>
</thead>
<tbody>
<tr>
<td>+r</td>
<td></td>
<td>+t</td>
<td>0.02</td>
</tr>
<tr>
<td>-r</td>
<td>-t</td>
<td></td>
<td>0.09</td>
</tr>
<tr>
<td>-r</td>
<td>-t</td>
<td></td>
<td>0.81</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></td>
<td>0.83</td>
</tr>
</tbody>
</table>
Multiple Elimination

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

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

\[ P(T, L) \]

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

\[ P(L) \]

\[
\begin{array}{c}
+l & 0.134 \\
-l & 0.886 \\
\end{array}
\]
Thus Far: Multiple Join, Multiple Eliminate (= Inference by Enumeration)
Marginalizing Early (= Variable Elimination)
Traffic Domain

\[ P(L) = ? \]

- **Inference by Enumeration**

\[
= \sum_t \sum_r P(L|t) P(r) P(t|r)
\]

- **Variable Elimination**

\[
= \sum_t P(L|t) \sum_r P(r) P(t|r)
\]
Marginalizing Early! (aka VE)

\[ P(R) \]

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

\[ P(T|R) \]

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

\[ P(R, T) \]

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

\[ P(T) \]

\[
\begin{array}{c|c}
+t & 0.17 \\
-t & 0.83 \\
\end{array}
\]

\[ P(L|T) \]

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

Join R

Join T

Sum out R

Sum out T
Evidence

- If evidence, start with factors that select that evidence
  - No evidence uses these initial factors:

\[
\begin{array}{c|c}
P(R) & P(T \mid R) & P(L \mid T) \\
+ r & 0.1 & + t & 0.8 & + t & + l & 0.3 \\
- r & 0.9 & - t & 0.2 & + t & - l & 0.7 \\
+ r & + t & 0.1 & - t & + l & 0.1 & - t & - l & 0.9 \\
- r & - t & 0.9 & - t & - l & 0.9 \\
\end{array}
\]

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

\[
\begin{array}{c|c}
P(+ r) & P(T \mid + r) & P(L \mid T) \\
+ r & 0.1 & + t & 0.8 & + t & + l & 0.3 \\
+ r & + t & 0.8 & + t & - l & 0.7 \\
- r & + t & 0.1 & - t & + l & 0.1 \\
- r & - t & 0.9 & - t & - l & 0.9 \\
\end{array}
\]

- 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!
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

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

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

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

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

Finish with B

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

Normalize

\[ P(B|j, m) \]
Same Example in Equations

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

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

All we are doing is exploiting \( uwy + uwz + uxy + uxz + vwy + vwz + vxy + vxz = (u+v)(w+x)(y+z) \) 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?

- **Satisfiability (3-SAT):**
  
  $$(x_1 \lor x_2 \lor \neg x_3) \land (\neg x_1 \lor x_3 \lor \neg x_4) \land (x_2 \lor \neg x_2 \lor x_4) \land (\neg x_3 \lor \neg x_4 \lor \neg x_5) \land (x_2 \lor x_5 \lor x_7) \land (x_4 \lor x_5 \lor x_6) \land (\neg x_5 \lor \neg x_6 \lor \neg x_7) \land (\neg x_5 \lor \neg x_6 \lor x_7)$$

  $$P(X_i = 0) = P(X_i = 1) = 0.5$$

  $$Y_1 = X_1 \lor X_2 \lor \neg X_3$$

  $$\ldots$$

  $$Y_8 = \neg X_5 \lor X_6 \lor X_7$$

  $$Y_{1,2} = Y_1 \land Y_2$$

  $$\ldots$$

  $$Y_{7,8} = Y_7 \land Y_8$$

  $$Y_{1,2,3,4} = Y_{1,2} \land Y_{3,4}$$

  $$Y_{5,6,7,8} = Y_{5,6} \land Y_{7,8}$$

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

- 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