CSC242: Intro to AI

Lecture 15 Bayesian Networks
Othello Tournament
Phase I Results

- DS-DH
- Deep-Blue
- EvanMariaPlayer
- Hyphaene-Thebaica
- KautzPlayer
- OR-KC
- a-a
- alaska-boat
- bent-paperclips
- blake-phelps
- crazy-pingpong
- delicious-fudge
- digital-teapot
- invalid-munmap
- is-rever
- jar-vis
- jesus-fish
- no-name
- othello-game
- othello-player
- problem-solved
- random-words
- robotics-anonymous
- saint-inferno
- samurai-sharks
- screaming-banjos
- spherical-cow
- spline-reticulators
- team-victory
Best Team Name

Third Place: Hyphaene-Thebaica
Second Place: screaming-banjos
First Place: spherical-cow

0 Extra Credit Points!
Defeated KautzPlayer

DS-DH
Hyphaene-Thebaica
OR-KC
a-a
alaska-boat
jar-vis
jesus-fish
no-name

othello-game
othello-player
problem-solved
random-words
saint-inferno
spherical-cow
spline-reticulators
team-victory

10 Extra Credit Points!
Best Performance

2nd Place:

97 alaska-boat
94 spherical-cow
94 a-a

20 Extra Credit Points!

94 a-a
Best Performance

1st Place:

106 spline-reticulators
105 Hyphaene-Thebaica

30 Extra Credit Points!

103 hyphaene-thebatica
Calendar

- April 1: ULW First Draft Due
- April 8: Project 2: Planning Due
- April 8: Exam 3: Probability
- April 29: Project 3: Neural Networks Due
- April 29: ULW Final Draft Due
- May 9: Final Exam
Bayesian Diagnosis

\[ P(\text{disease} \mid \text{symptom}) = \frac{P(\text{symptom} \mid \text{disease})P(\text{disease})}{P(\text{symptom})} \]
Cavity

Toothache

Catch
Combining Evidence

\[ P(\text{Cavity} \mid \text{toothache} \land \text{catch}) = \alpha \langle 0.180, 0.016 \rangle \approx \langle 0.871, 0.129 \rangle \]

<table>
<thead>
<tr>
<th></th>
<th>toothache</th>
<th>¬toothache</th>
</tr>
</thead>
<tbody>
<tr>
<td>catch</td>
<td>0.108</td>
<td>0.012</td>
</tr>
<tr>
<td>¬catch</td>
<td>0.008</td>
<td>0.144</td>
</tr>
<tr>
<td>cavity</td>
<td>0.016</td>
<td>0.064</td>
</tr>
<tr>
<td>¬cavity</td>
<td>0.576</td>
<td>0.072</td>
</tr>
</tbody>
</table>
### Exponential Growth of Combinations of Evidence

\[ P(\text{Cavity} \mid \text{toothache} \land \text{catch} \land \neg \text{bleeding}) \]

<table>
<thead>
<tr>
<th></th>
<th>toothache</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>toothache</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- \( \text{toothache} \) and \( \neg \text{toothache} \) are the possible states for the condition of toothache.
- \( \text{catch} \) and \( \neg \text{catch} \) are the possible states for the condition of catching something.
- \( \text{bleeding} \) and \( \neg \text{bleeding} \) are the possible states for the condition of bleeding.

- The table represents the exponential growth of combinations of evidence with four possible states for each condition.
- The probability of \( \text{Cavity} \) given the conjunction of \( \text{toothache} \), \( \text{catch} \), and \( \neg \text{bleeding} \) is shown.

The table shows all possible combinations of evidence and their probabilities.
Conditional Independence

- Both *toothache* and *catch* are caused by a cavity, but neither has a direct effect on the other.
- The variables are independent given the presence or absence of a cavity.
- Notation: $\text{Toothache} \parallel \text{Catch} \mid \text{Cavity}$
Benefit of Conditional Independence Assumptions

\[ P(Cavity \mid toothache \land catch) = \alpha P(toothache \mid Cavity)P(catch \mid Cavity)P(Cavity) \]

Only need these probabilities - linear in the number of evidence variables!
Bayesian Network

• Data structure for compactly representing a joint probability distributions
• Leverages (conditional) independencies between variables
• Can be exponentially smaller than explicit tabular representation of the joint distribution
• Supports many algorithms for inference and learning
Bayesian Networks

Random Variables

- Cavity
- Toothache
- Catch
Bayesian Networks

- Cavity
- Toothache
- Catch

"has direct influence on"
Bayesian Networks

Cavity

Toothache

Catch

conditionally independent given parents
Bayesian Networks

\[ P(\text{Toothache} \mid \text{Cavity}) \quad \text{and} \quad P(\text{Catch} \mid \text{Cavity}) \]

Conditional Probability Distributions
Bayesian Networks

Cavity

P(Toothache | Cavity)
P(Cavity)
P(Catch | Cavity)

Toothache

Catch

Prior Probability Distribution
Bayesian Networks

- Each node corresponds to a random variable
- There is a link from \( X \) to \( Y \) if \( X \) has a direct influence on \( Y \) (no cycles; DAG)
- The node for \( X_i \) stores the conditional distribution \( P(X_i \mid \text{Parents}(X_i)) \)
- Root nodes store the priors \( P(X_i) \)
Bayesian Networks
How-To

- Select random variables required to model the domain
- Add links from causes to effects
  - “Directly influences”
- No cycles
- Write down (conditional) probability distributions for each node
Semantics of Bayesian Networks

- Full joint distribution can be computed as the product of the separate conditional probabilities stored in the network

\[ P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i \mid \text{parents}(X_i)) \]
P(Toothache, cavity, catch) =
  P(Toothache | cavity)P(catch | cavity)P(cavity)
\[ P(\neg \text{toothache}, \text{cavity}, \text{catch}) = P(\neg \text{toothache} | \text{cavity}) P(\text{catch} | \text{cavity}) P(\text{cavity}) \]
P(¬toothache, ¬cavity, catch) = 
\[ P(¬\text{toothache} | ¬\text{cavity}) P(\text{catch} | ¬\text{cavity}) P(¬\text{cavity}) \]
Inference in Bayesian Networks

• A query can be answered from a Bayesian Network by computing sums of products of conditional probabilities from the network
Inference in Bayesian Networks

- A query can be answered from a Bayesian Network by computing sums of products of conditional probabilities from the network

\[ P(X \mid e) = \]
Inference in Bayesian Networks

- A query can be answered from a Bayesian Network by computing sums of products of conditional probabilities from the network

\[ P(X \mid e) = \alpha P(X, e) = \]
Inference in Bayesian Networks

- A query can be answered from a Bayesian Network by computing sums of products of conditional probabilities from the network

\[
P(X \mid e) = \alpha \: P(X, e) = \alpha \sum_y P(X, e, y)
\]
A query can be answered from a Bayesian Network by computing sums of products of conditional probabilities from the network.

\[
P(X \mid e) = \alpha \ P(X, e) = \alpha \ \sum_y P(X, e, y)
\]

\[
= \alpha \ \sum_y \prod_{i=1}^{n} P(X_i \mid \text{parents}(X_i))
\]
\begin{center}
\begin{tikzpicture}
  \node[draw, ellipse] (burglary) at (0,1) {Burglary};
  \node[draw, ellipse] (alarm) at (0,0) {Alarm};
  \node[draw, ellipse] (earthquake) at (2,1) {Earthquake};
  \draw[->] (burglary) -- node[above] {$P(B)$} (alarm); 
  \draw[->] (earthquake) -- node[above] {$P(E)$} (alarm);
  \node[draw, fill=white] at (0.5,1.5) {$P(B)$};
  \node[draw, fill=white] at (0.5,0.5) {0.001};
  \node[draw, fill=white] at (2.5,1.5) {$P(E)$};
  \node[draw, fill=white] at (2.5,0.5) {0.002};
\end{tikzpicture}
\end{center}
<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>
$$P(B | j, m)$$

$$P(B) = 0.001$$

$$P(E) = 0.002$$

<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>

| $A$ | $P(J | A)$ |
|-----|-----------|
| $t$ | 0.9       |
| $f$ | 0.05      |

| $A$ | $P(M | A)$ |
|-----|-----------|
| $t$ | 0.7       |
| $f$ | 0.01      |

$$P(Burglary \mid JohnCalls = True, MaryCalls = True)$$
\[ P(B \mid j, m) = \alpha P(B, j, m) \]
\[ P(B | j, m) = \alpha P(B, j, m) \]
\[ P(B \mid j, m) = \alpha P(B, j, m) \]
\[ P(B \mid j, m) = \alpha P(B, j, m) = \alpha \sum_e \sum_a P(B, j, m, e, a) \]
\[ P(B \mid j, m) = \alpha \quad P(B, j, m) = \alpha \sum_e \sum_a P(B, j, m, e, a) \]
$P(B \mid j, m) = \alpha P(B, j, m) = \alpha \sum_e \sum_a P(B, j, m, e, a)$

Marginalizing Joint Distribution
\[
P(B \mid j, m) = \alpha P(B, j, m) = \alpha \sum_e \sum_a P(B, j, m, e, a)
\]

\[
P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i \mid \text{parents}(X_i))
\]
\[ P(B \mid j, m) = \alpha \sum_e \sum_a P(B, j, m, e, a) \]

\[ P(b \mid j, m) = \alpha \sum_e \sum_a P(b) P(e) P(a \mid b, e) P(j \mid a) P(m \mid a) \]
\[
P(b | j, m) = \alpha P(b) P(e) P(a | b, e) P(j | a) P(m | a) + P(b) P(e) P(\neg a | b, e) P(j | \neg a) P(m | \neg a) +
\]
\[
P(b) P(\neg e) P(a | b, \neg e) P(j | a) P(m | a) +
\]
\[
P(b) P(\neg e) P(\neg a | b, \neg e) P(j | \neg a) P(m | \neg a)
\]
\[ P(B \mid j, m) = \alpha \langle 0.00059224, 0.0014919 \rangle \approx \langle 0.284, 0.716 \rangle \]
Optimizing Bayesian Network Inference

- It is often possible to **optimize** a query to a Bayesian Network
- Idea: **rearrange terms**, so that each is evaluated as few times as possible
Example: Optimizing Inference

\[ P(b | j, m) = \alpha \sum_e \sum_a P(b) P(e) P(a | b, e) P(j | a) P(m | a) \]

\[ = \alpha P(b) \sum_a \sum_e P(a | b, e) P(j | a) P(m | a) \]

\[ = \alpha P(b) \sum_a \sum_e P(a | b, e) P(j | a) P(m | a) \]

\[ = \alpha P(b) \sum_a P(j | a) P(m | a) \sum_e P(a | b, e) \]
Example: Optimizing Inference

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

Before optimization:  \(2 \times 2 \times 5 = 20\) multiplies

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

After optimization:  \(1 + 2 \times 3 = 7\) multiplies
Bayes Net Toolkits

- Many Bayesian Network tools are available
- Variety of built-in optimization routines
- Just input the network and let the system do the work!
Worst-Case Complexity

- Exact inference in Bayesian Networks can be shown to be as hard as computing the number of satisfying assignments of a propositional logic formula
- \#P-complete (harder than NP-complete)
Next Questions

• How do we learn the (conditional) probabilities for a Bayesian Network from a set of data?

• How can we do even faster approximate probabilistic inference?