Bayesian Networks

CS171, Fall 2016
Introduction to Artificial Intelligence
Prof. Alexander Ihler

Reading: R&N Ch 14
Why Bayesian Networks?

- Knowledge Representation & Reasoning (Inference)
  - Propositional Logic
    - Knowledge Base: Propositional logic sentences
    - Reasoning: KB |= Theory
      - Find a model or Count models
  - Probabilistic Reasoning
    - Knowledge Base: Full joint probability over all random variables
    - Reasoning: Compute Pr ( KB |= Theory )
      - Find the most probable assignments
      - Compute marginal / conditional probability

- Why Bayesian Net?
  - Manipulating full joint probability distribution is very hard!
  - Exploit conditional independence properties of our distribution
  - Bayesian Network captures conditional independence
    - Graphical Representation (Probabilistic Graphical Models)
    - Tool for Reasoning, Computation (Probabilistic Reasoning bases on the Graph)
Conditional independence

- Recall: chain rule of probability
  - $p(x,y,z) = p(x) \ p(y|x) \ p(z|x,y)$

- Some of these models will be conditionally independent
  - e.g., $p(x,y,z) = p(x) \ p(y|x) \ p(z|x)$

- Some models may have even more independence
  - E.g., $p(x,y,z) = p(x) \ p(y) \ p(z)$
Bayesian networks

• Directed graphical model
• Nodes associated with variables
• “Draw” independence in conditional probability expansion
  — Parents in graph are the RHS of conditional

• Ex: \( p(x, y, z) = p(x) \ p(y \mid x) \ p(z \mid y) \)

\[
\begin{array}{ccc}
  \text{x} & \rightarrow & \text{y} \\
  \rightarrow & \text{z}
\end{array}
\]

• Ex: \( p(a, b, c, d) = p(a) \ p(b \mid a) \ p(c \mid a, b) \ p(d \mid b) \)

\[
\begin{array}{ccc}
  \text{a} & \rightarrow & \text{b} \\
  \rightarrow & \text{d} \\
  \rightarrow & \text{c} \\
\end{array}
\]

Graph must be acyclic

Corresponds to an order over the variables (chain rule)
Example

Consider the following 5 binary variables:
- B = a burglary occurs at your house
- E = an earthquake occurs at your house
- A = the alarm goes off
- J = John calls to report the alarm
- M = Mary calls to report the alarm

What is P(B | M, J) ? (for example)

We can use the full joint distribution to answer this question

- Requires $2^5 = 32$ probabilities

- Can we use prior domain knowledge to come up with a Bayesian network that requires fewer probabilities?
Constructing a Bayesian network

• Order the variables in terms of causality (may be a partial order)
  – e.g., \{ E, B \} \rightarrow \{ A \} \rightarrow \{ J, M \}

• Now, apply the chain rule, and simplify based on assumptions

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

  – These assumptions are reflected in the graph structure of the Bayesian network
Constructing a Bayesian network

- Given \( p(J, M, A, E, B) = p(E) \ p(B) \ p(A | E, B) \ p(J | A) \ p(M | A) \)
- Define probabilities: \(1 + 1 + 4 + 2 + 2\)
- Where do these come from?
  - Expert knowledge; estimate from data; some combination

<table>
<thead>
<tr>
<th>P(E)</th>
<th>0.002</th>
</tr>
</thead>
<tbody>
<tr>
<td>P(B)</td>
<td>0.001</td>
</tr>
<tr>
<td>E</td>
<td>B</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>A</td>
<td>P(J</td>
</tr>
<tr>
<td>0</td>
<td>0.05</td>
</tr>
<tr>
<td>1</td>
<td>0.90</td>
</tr>
<tr>
<td>A</td>
<td>P(M</td>
</tr>
<tr>
<td>0</td>
<td>0.01</td>
</tr>
<tr>
<td>1</td>
<td>0.70</td>
</tr>
</tbody>
</table>
## Constructing a Bayesian network

- **Joint distribution**

Full joint distribution: $2^5 = 32$ probabilities

Structured distribution: specify 10 parameters

<table>
<thead>
<tr>
<th>E</th>
<th>B</th>
<th>A</th>
<th>J</th>
<th>M</th>
<th>P(…)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>.93674</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>.00133</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>.00005</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>.00000</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>.00003</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>.00002</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>.00003</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>.00000</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>.00946</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>.00001</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>.00000</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>.00000</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>.00007</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>.00004</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>.00007</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>.00000</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>.00050</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>.00000</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>.00000</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>.00000</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>.00063</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>.00037</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>.00059</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>.00000</td>
</tr>
</tbody>
</table>
The “alarm” network: 37 variables, 509 parameters (rather than $2^{37} = 10^{11}$!)
Network structure and ordering

- The network structure depends on the conditioning order
  - Suppose we choose ordering M, J, A, B, E
Network structure and ordering

- The network structure depends on the conditioning order
  - Suppose we choose ordering M, J, A, B, E

- “Non-causal” ordering
  - Deciding independence is harder
  - Selecting probabilities is harder
  - Representation is less efficient

$$1 + 2 + 4 + 2 + 4 = 13$$ probabilities
Network structure and ordering

- The network structure depends on the conditioning order
  - Suppose we choose ordering M, J, A, B, E

- “Non-causal” ordering
  - Deciding independence is harder
  - Selecting probabilities is harder
  - Representation is less efficient

- Some orders may not reveal any independence!

\[ p(J, M, A, E, B) = p(M) p(J|M) p(E|M, J) p(B|M, J, E) p(A|M, J, E, B) \]
Reasoning in Bayesian networks

• Suppose we observe J
  – Observing J makes A more likely
  – A being more likely makes B more likely

• Suppose we observe A
  – Makes M more likely

• Observe A and J?
  – J doesn’t add anything to M
  – Observing A makes J, M independent

• How can we read independence directly from the graph?
Reasoning in Bayesian networks

• How are J, M related given A?
  - \( P(M) = 0.0117 \)
  - \( P(M|A) = 0.7 \)
  - \( P(M|A,J) = 0.7 \)
  - Conditionally independent

  *(we actually know this by construction!)*

• Proof:

\[
p(J, M | a) \propto \sum_{e,b} p(e) \ p(b) \ p(a|e,b) \ p(J|a) \ p(M|a)
\]

\[
= \left( \sum_{e,b} p(e, b, a) \right) \ p(J|a) \ p(M|a)
\]

\[
= p(a) \ p(J|a) \ p(M|a)
\]

\[
= c_a \ f_a(J) \ g_a(M)
\]
Reasoning in Bayesian networks

- How are J,B related given A?
  - $P(B) = 0.001$
  - $P(B|A) = 0.3735$
  - $P(B|A,J) = 0.3735$
  - Conditionally independent

- Proof:

$$p(J, B|a) \propto \sum_{e, m} p(e) \ p(B) \ p(a|e, B) \ p(J|a) \ p(m|a)$$

$$= \left( \sum_{e} p(e, B, a) \right) \ p(J|a) \ \left( \sum_{m} p(m|a) \right)$$

$$= p(B, a) \ p(J|a)$$

$$= f_{a}(B) \ g_{a}(J)$$
Reasoning in Bayesian networks

- How are E, B related?
  - $P(B) = 0.001$
  - $P(B|E) = 0.001$
  - (Marginally) independent

- What about given A?
  - $P(B|A) = 0.3735$
  - $P(B|A,E) = 0.0032$
  - Not conditionally independent!
  - The “causes” of A become coupled by observing its value
  - Sometimes called “explaining away”
D-Separation

• Prove sets X,Y independent given Z?
• Check all *undirected* paths from X to Y
• A path is “inactive” if it passes through:
  1. A “chain” with an observed variable
  2. A “split” with an observed variable
  3. A “vee” with only *unobserved* variables below it
• If all paths are inactive, conditionally independent!
A node is conditionally independent of all other nodes in the network given its Markov blanket (in gray).
Graphs and Independence

- Graph structure allows us to infer independence in $p(.)$
  - $X,Y$ d-separated given $Z$?

- Adding edges
  - Fewer independencies inferred, but still valid to represent $p(.)$
  - Complete graph: can represent any distribution $p(.)$

| E  | A  | $P(J | E,A)$ |
|----|----|-------------|
| 0  | 0  | 0.05        |
| 0  | 1  | 0.90        |
| 1  | 0  | 0.05        |
| 1  | 1  | 0.90        |

| E  | B  | $P(A | E,B)$ |
|----|----|-------------|
| 0  | 0  | 0.001       |
| 0  | 1  | 0.29        |
| 1  | 0  | 0.94        |
| 1  | 1  | 0.95        |

| A  | $P(M | A)$ |
|----|-----------|
| 0  | 0.01      |
| 1  | 0.70      |
Example: Car diagnosis

Initial evidence: car won’t start
Testable variables (green), “broken, so fix it” variables (orange)
Hidden variables (gray) ensure sparse structure, reduce parameters
Compact conditional distributions contd.

Noisy-OR distributions model multiple noninteracting causes
1) Parents \( U_1 \ldots U_k \) include all causes (can add leak node)
2) Independent failure probability \( q_i \) for each cause alone
   \[ P(X|U_1 \ldots U_j, \neg U_{j+1} \ldots \neg U_k) = 1 - \prod_{i=1}^{j} q_i \]

<table>
<thead>
<tr>
<th>Cold</th>
<th>Flu</th>
<th>Malaria</th>
<th>( P(\text{Fever}) )</th>
<th>( P(\neg \text{Fever}) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>0.0</td>
<td>1.0</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>0.9</td>
<td>0.1</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>0.8</td>
<td>0.2</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>0.98</td>
<td>0.02 = 0.2 \times 0.1</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>0.4</td>
<td>0.6</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>0.94</td>
<td>0.06 = 0.6 \times 0.1</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>0.88</td>
<td>0.12 = 0.6 \times 0.2</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>0.988</td>
<td>0.012 = 0.6 \times 0.2 \times 0.1</td>
</tr>
</tbody>
</table>

Number of parameters linear in number of parents
Naïve Bayes Model

\[ P(C | X_1,\ldots,X_n) = \alpha \prod P(X_i | C) \cdot P(C) \]

Features X are conditionally independent given the class variable C

Widely used in machine learning
  e.g., spam email classification: X’s = counts of words in emails

Probabilities P(C) and P(Xi | C) can easily be estimated from labeled data
Naïve Bayes Model (2)

\[ P(C | X_1, \ldots X_n) = \alpha \prod P(X_i | C) \cdot P(C) \]

<Learning Naïve Bayes Model>

Probabilities \( P(C) \) and \( P(X_i | C) \) can easily be estimated from labeled data

\[
P(C = c_j) \approx \frac{\#(\text{Examples with class label } c_j)}{\#(\text{Examples})}
\]

\[
P(X_i = x_{ik} | C = c_j) \approx \frac{\#(\text{Examples with } X_i \text{ value } x_{ik} \text{ and class label } c_j)}{\#(\text{Examples with class label } c_j)}
\]

Usually easiest to work with logs

\[
\log [ P(C | X_1, \ldots X_n) ] = \log \alpha + \sum [ \log P(X_i | C) + \log P(C) ]
\]

DANGER: Suppose ZERO examples with \( X_i \) value \( x_{ik} \) and class label \( c_j \)? An unseen example with \( X_i \) value \( x_{ik} \) will NEVER predict class label \( c_j \)!

**Practical solutions:** Pseudocounts, e.g., add 1 to every \#() , etc.

**Theoretical solutions:** Bayesian inference, beta distribution, etc.
Hidden Markov Models

- Two key assumptions
  - Hidden state sequence is Markov
  - Observations $o_t$ is conditionally independent given state $x_t$

- Widely used in:
  - speech recognition, protein sequence models, ...

- Bayesian network is a tree, so inference is linear in $n$
  - Exploit graph structure for efficient computation (as in CSPs)
You should know...

• Basic concepts and vocabulary of Bayesian networks.
  – Nodes represent random variables.
  – Directed arcs represent (informally) direct influences.
  – Conditional probability tables, \( P( X_i \mid \text{Parents}(X_i) ) \).

• Given a Bayesian network:
  – Write down the full joint distribution it represents.

• Given a full joint distribution in factored form:
  – Draw the Bayesian network that represents it.

• Given a variable ordering and some background assertions of conditional independence among the variables:
  – Write down the factored form of the full joint distribution, as simplified by the conditional independence assertions.
Summary

• Bayesian networks represent a joint distribution using a graph

• The graph encodes a set of conditional independence assumptions

• Answering queries (or inference or reasoning) in a Bayesian network amounts to efficient computation of appropriate conditional probabilities

• Probabilistic inference is intractable in the general case
  – But can be carried out in linear time for certain classes of Bayesian networks