Informatics 2D – Reasoning and Agents
Semester 2, 2018–2019

Alex Lascarides
alex@inf.ed.ac.uk

Lecture 23 – Probabilistic Reasoning with Bayesian Networks
14th March 2019
Where are we?

Last time . . .

- Using JPD tables for probabilistic inference
- Concepts of absolute and conditional independence
- Bayes’ rule

Today . . .

- **Probabilistic Reasoning with Bayesian Networks**
Representing knowledge in an uncertain domain

- Full joint probability distributions can become intractably large very quickly
- Conditional independence helps to reduce the number of probabilities required to specify the JPD
- Now we will introduce **Bayesian networks** (BNs) to systematically describe dependencies between random variables
- Roughly speaking, BNs are graphs that connect nodes representing variables with each other whenever they depend on each other
Bayesian networks

- A BN is a directed acyclic graph (DAG) with nodes annotated with probability information.
- The nodes represent random variables (discrete/continuous).
- Links connect nodes. If there is an arrow from $X$ to $Y$, we call $X$ a parent of $Y$.
- Each node $X_i$ has a conditional probability distribution (CPD) attached to it.
- The CPD describes how $X_i$ depends on its parents, i.e. its entries describe $P(X_i | \text{Parents}(X_i))$. 


Bayesian networks

- Topology of graphs describes conditional independence relationships.
- Intuitively, links describe **direct effects** of variables on each other in the domain.
- Assumption: anything that is not directly connected does not directly depend on each other.
- In previous dentist/weather example:
Example

- New burglar alarm has been fitted, fairly reliable but sometimes reacts to earthquakes
- Neighbours John and Mary promise to call when they hear alarm
- John sometimes mistakes phone for alarm, and Mary listens to loud music and sometimes doesn’t hear alarm

![Bayesian Network Diagram]

<table>
<thead>
<tr>
<th>Burglary</th>
<th>Earthquake</th>
</tr>
</thead>
<tbody>
<tr>
<td>P(B)</td>
<td>.001</td>
</tr>
<tr>
<td>P(E)</td>
<td>.002</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Alarm</th>
</tr>
</thead>
<tbody>
<tr>
<td>P(A)</td>
</tr>
<tr>
<td>B</td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>T</td>
</tr>
<tr>
<td>T</td>
</tr>
<tr>
<td>F</td>
</tr>
<tr>
<td>F</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>JohnCalls</th>
<th>MaryCalls</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>P(J)</td>
</tr>
<tr>
<td>---</td>
<td>-------</td>
</tr>
<tr>
<td>T</td>
<td>.90</td>
</tr>
<tr>
<td>F</td>
<td>.05</td>
</tr>
<tr>
<td>A</td>
<td>P(M)</td>
</tr>
<tr>
<td>---</td>
<td>-------</td>
</tr>
<tr>
<td>T</td>
<td>.70</td>
</tr>
<tr>
<td>F</td>
<td>.01</td>
</tr>
</tbody>
</table>
Example – things to note

- No perception of earthquake by John or Mary
- No explicit modelling of phone ring confusing John, or of Mary’s loud music (summarised in uncertainty regarding their reaction)
- Actually this uncertainty summarises any kind of failure
  - almost impossible to enumerate all possible causes,
  - and we don’t have estimates for their probabilities anyway
- Each row in CPTs contains a **conditioning case** (configuration of parent values)
- For \( k \) parents, \( 2^k \) possible cases
- We often omit \( P(\neg x_i | \text{Parents}(X_i)) \) from CPT for node \( X_i \) (computes as \( 1 - P(x_i | \text{Parents}(X_i)) \))
The semantics of Bayesian Networks

- Two views:
  - BN as representation of JPD (useful for constructing BNs)
  - BN as collection of conditional independence statements (useful for designing inference procedures)

- Every entry $P(X_1 = x_1 \land \ldots \land X_n = x_n)$ in the JPD can be calculated from a BN (abbreviate by $P(x_1, \ldots, x_n)$)

- $P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i|\text{parents}(X_i))$

- Example:

  \[
P(j \land m \land a \land \neg b \land \neg e)
  = P(j|a)P(m|a)P(a|\neg b \land \neg e)P(\neg b)P(\neg e)
  = 0.9 \times 0.7 \times 0.001 \times 0.999 \times 0.998 = 0.00062
  \]

- As before, this can be used to answer any query
A method for constructing BNs

Recall product rule for $n$ variables:

$$P(x_1, \ldots, x_n) = P(x_n \mid x_{n-1}, \ldots, x_1)P(x_{n-1}, \ldots, x_1)$$

Repeated application of this yields the so-called **chain rule**:

$$P(x_1, \ldots, x_n) = P(x_n \mid x_{n-1}, \ldots, x_1)P(x_{n-1} \mid x_{n-2}, \ldots, x_1) \cdots P(x_2 \mid x_1)P(x_1)$$

$$= \prod_{i=1}^{n} P(x_i \mid x_{i-1}, \ldots, x_1)$$

With this we obtain $P(X_i \mid X_{i-1}, \ldots, X_1) = P(X_i \mid \text{Parents}(X_i))$ as long as $\text{Parents}(X_i) \subseteq \{X_{i-1}, \ldots, X_1\}$ (this can be ensured by labelling nodes appropriately).

For example, it is reasonable to assume that

$$P(\text{MaryCalls} \mid \text{JohnCalls}, \text{Alarm}, \text{Earthquake}, \text{Burglary}) = P(\text{MaryCalls} \mid \text{Alarm})$$
Compactness and node ordering

- BNs examples of **locally structured** (sparse) systems: subcomponents only interact with small number of other components
- E.g. if 30 nodes and every node depends on 5 nodes, BN will have $30 \times 2^5 = 960$ probabilities stored in the CPDs, while JPD would have $2^{30} \approx 1000^3$ entries
- But remember that this is based on designer’s independence assumptions!
- Also not trivial to determine good BN structure:

  Add “root causes” first, then variables they influence, and so on, until we reach “leaves” which have no influence on other variables
Conditional independence relations in BNs

Have provided “numerical” semantics, but can also look at (equivalent) “topological” semantics, namely:

1. A node is conditionally independent of its **non-descendants**, given its parents
2. A node is conditionally independent of all other nodes, given its parents, children and children’s parents, i.e. its **Markov blanket**
Efficient representation of conditional distributions

- Even the $2^k$ ($k$ parents) conditioning cases that have to be provided require a great deal of experience and knowledge of the domain.
- Arbitrary relationships are unlikely, often describable by **canonical distributions** that fit some standard pattern.
- By specifying pattern by a few parameters we can save a lot of space!
- Simplest case: **deterministic node** that can be directly inferred from values of parents.
- For example, logical or mathematical functions.
Noisy-OR relationships

- Any cause can make effect true, but won’t necessarily (effect inhibited; $P(\text{effect}|\text{cause}) < 1$)
- Assumes all causes are listed (leak node can be used to cater for “miscellaneous” unlisted causes)
- Also assumes inhibitions are mutually conditionally independent
  - Whatever inhibits $C_1$ from making $E$ true is independent of what inhibits $C_2$ from making $E$ true.
- So $E$ is false only if each of its true parents are inhibited and we can compute this likelihood from product of probabilities for each individual cause inhibiting $E$.
- How does this help?
Example of Noisy-OR

- Fever is caused by Cold, Flu or Malaria and that’s all (!!)
- Inhibitions of Cold, Flu and Malaria are mutually conditionally independent
- Likelihood that Cold is inhibited from causing Fever is \[ P(\neg\text{fever}|\text{cold}, \neg\text{flu}, \neg\text{malaria}) \]
  (similarly for other causes)
- Individual inhibition probabilities:
  \begin{align*}
  P(\neg\text{fever}|\text{cold}, \neg\text{flu}, \neg\text{malaria}) &= 0.6 \\
  P(\neg\text{fever}|\neg\text{cold}, \text{flu}, \neg\text{malaria}) &= 0.2 \\
  P(\neg\text{fever}|\neg\text{cold}, \neg\text{flu}, \text{malaria}) &= 0.1
  \end{align*}
- Inhibitions mutually independent, so:
  \[ P(\neg\text{fever}|\text{cold}, \text{flu}, \neg\text{malaria}) = \]
  \[ P(\neg\text{fever}|\text{cold}, \neg\text{flu}, \neg\text{malaria})P(\neg\text{fever}|\neg\text{cold}, \text{flu}, \neg\text{malaria}) \]
Noisy-OR relationships

- We can construct entire CPT from this information

<table>
<thead>
<tr>
<th>Cold</th>
<th>Flu</th>
<th>Malaria</th>
<th>P(Fever)</th>
<th>P(¬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 × 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 × 0.1</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>0.88</td>
<td>0.12 = 0.6 × 0.2</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>0.988</td>
<td>0.012 = 0.6 × 0.2 × 0.1</td>
</tr>
</tbody>
</table>

- Encodes CPT with k instead of $2^k$ values!
BNs with continuous variables

- Often variables range over continuous domains
- **Discretisation** one possible solution but often leads to inaccuracy or requires a lot of discrete values
- Other solution: use of standard families of probability distributions specified in terms of a few parameters
- Example: normal/Gaussian distribution $N(\mu, \sigma^2)(x)$ defined in terms of mean $\mu$ and variance $\sigma^2$ (needs just two parameters)
- **Hybrid Bayesian Networks** use mixture of discrete and continuous variables (special methods to deal with links between different types – not discussed here)
Summary

- Introduced Bayesian Networks as a structured way of reasoning under uncertainty using probabilities and independence.
- Defined their semantics in terms of JPD representation, and conditional independence statements.
- Gave numerical and topological interpretation of semantics.
- Talked about issues of efficient representation of CPTs.
- Discussed continuous variables and hybrid networks.
- Next time: **Exact Inference in Bayesian Networks**