Introduction to Bayesian networks
Summary •
Learning •
Inference •
Some common graphical models •
Bayesian networks •
Conditional independence •
Probabilistic models •
Reasoning with uncertainty: Rules vs probability theory •

Contents
How to reason?

- Summarizing them → uncertainty
- Ignoring them → too simple
- Enumerating them → too complex

Why reason with uncertainty?

Reasoning with uncertainty
Holmes thinks: If it rained then that explains why my grass is wet, so there is no reason to believe that the sprinkler has been on.

Also wet. Holmes is almost sure that it has been raining.

Next he notices that the grass of his neighbor, Dr. Watson, is wet. His belief in both events increases.

When Mr. Holmes leaves the house, he notices that his grass is wet. Mr. Holmes has a garden with a nice grass. One morning,
Problem: Rules are context free

If Rained then Wet
If Grass then Sprinkler
If Sprinkler then Rained, Sprinkler

How to combine rules?

If Sprinkler then Wet Grass with certainty 9
If Rained then Wet Grass with certainty 9
If Rained then Wet Grass with certainty 9

E.g.

(f(x) if condition with certainty x then fact with certainty f(x)) Rules

Rules with uncertainty
Probability theory is context sensitive.

\[
\frac{(\mathcal{C}, \mathcal{V})_d}{(\mathcal{C}, \mathcal{V})_d} = \frac{(\mathcal{C})_d}{(\mathcal{C}, \mathcal{V})_d} = (\mathcal{C}|\mathcal{V})_d
\]

Conditional probability is related to marginal probabilities.

Given that I know \( \mathcal{C} \) (context), then the probability of \( \mathcal{A} \) is

A conditional probability is

\[
\mathbb{P}(\mathcal{A} \mid \mathcal{C}) = (\mathcal{A})_d = \frac{(\mathcal{A}, \mathcal{C})_d}{(\mathcal{A})_d}
\]
<table>
<thead>
<tr>
<th></th>
<th>Rained</th>
<th>Sprinkler</th>
<th>Wet</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td></td>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td></td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td></td>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td></td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td></td>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td></td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td></td>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td></td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td></td>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td></td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td></td>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td></td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td></td>
<td>0.38</td>
<td>0.00</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>0.00</td>
<td>0.01</td>
<td>0.05</td>
</tr>
<tr>
<td></td>
<td>0.05</td>
<td>0.50</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>0.00</td>
<td>0.01</td>
<td>0.01</td>
</tr>
</tbody>
</table>

Wet grass with probabilities

\[
P(S|R,W) = P(S|W) = P(S) = P(R|W) = P(R) = 0.02 = 0.02 = 0.11 = 0.07 = 0.92 = 0.56
\]
\[
\frac{\left( \frac{\partial}{\partial x} f(x) \right)_{d}}{(f(x) f(x)'|x)'_{d}} = (f(x)' f(x)|x)_{d}
\]

- Conditional probability

\[
(u_x, \ldots, x_1)_{d} \big\{ u_{x_1+1}^{1+x_1} x_1^{1-x_1} \} \leq (x)_{d}
\]

- Marginal distribution of the probabilities of interest

- Inference: computation of the probabilities of interest

\[
(u_x, \ldots, x_n)_{d} \big\{ u_{x_1}^{1+x_1} x_1^{1-x_1} \}
\]

- Joint probability distribution (JPD)

- Random variables \( x_1, \ldots, x_n \)

- Probabilistic model

Probabilistic models
Graphical models \( \iff \) Conditional Independences

Solution: simplify model + computation by assuming

The numbers defining the table of \( P \) are not meaningful.

- the remaining \( n - 1 \) variables.
  - require summing \( P(x) \) over all \( 2^{n-1} \) combinations of
    - \( u \) terms. For instance, computing the marginal \( P(x) \) would
    - inference would require summatons over exponentially many
      - \( P \) the definition of \( P \) would require a table with \( 2^n \) entries.

Approach leads to several serious problems:

- all entries in \( P(x) \). Unless \( u \) is extremely small, this
- naive approach: probabilistic model requires the literal encoding of

Problems with naive probabilistic approach
\[(z|x)d = (z', h|x)d\]  
\text{given } z \text{, knowing } y \text{ tells me nothing about } x \text{, i.e.,}

\text{Conditionally independent given } z\]

Equivalently (but maybe more intuitively), variables \(x\) and \(y\) are conditionally independent given \(z\) if

\[(z|h)d(z|x)d = (z|h', x)d\]

\text{variables } x \text{ and } y \text{ are conditionally independent given } z \text{ if}

Similarly,

\[(x)d = (h|x)d\]  
\text{Knowing } y \text{ tells me nothing about } x \text{, i.e.,}

\text{marginally independent if}

Equivalently (but maybe more intuitively), variables \(x\) and \(y\) are

\[(h)d(x)d = (h', x)d\]

\text{variables } x \text{ and } y \text{ are marginally independent if}

Conditional independence
Next, it is given that John and Peter are both 10 years old and does \( x \) tell about \( y \) that for 10 year old boys, the mean weight = 35 kg, \( sd = 6 \) kg.

Next, it is given that John and Peter are of the same age. What does \( x \) tell about \( y \)?

Next, it is given that Peter weighs \( x \) kg. What does \( x \) tell about \( y \)?

Given: Peter weighs \( x \) kg. By age and sex, I want to know the weight of John.

Example: I have a book tabulating population weight distributions conditional on dependent \( z \) and vice versa.

Two variables, \( y \) that are marginally independent can be conditional independence (example)

NB:

Conditional independence (example)
Bayesian networks

\[ p(B, F, C, I, S) = p(B) p(I) p(F | I) p(C | F, I) p(S | F, I) \]

rather than entitles in the full JPD.

The model parameters are the conditional probability tables.

Graph (DAG), hence the term "Graphical model."

The factorization is uniquely represented by a Directed Acyclic graph.

A Bayesian network is a JPD defined by a factorization into conditional probability distributions.

Bayesian networks
\[(u x \cdots 1^{+} x | \forall x) d \equiv \forall u \bigwedge = (u x \cdots 1^{1} x) d\]

and we end up with the product,

\[\forall x \cdots \exists x d (u x \cdots \exists x | \forall x) d = (u x \cdots \exists x | \forall x) d = (u x \cdots \forall x) d\]

etc.

So,

\[(q) d (q | p) d = (q | p) d\]

In general,

**Chain rule**
Bayesian networks and chain rule

Bayesian network: $(u^x, \ldots, u^1) \rightarrow (u^1) \rightarrow (u^2) \rightarrow \cdots \rightarrow (u^n)$

Joint distribution of the Bayesian network:

$(u^x, \ldots, u^1) \rightarrow (u^1) \rightarrow (u^2) \rightarrow \cdots \rightarrow (u^n)$

Satisfy conditional probability distributions:

$\forall x, \ldots, x^{n-1} \in \mathbb{R}$

$\{x^i \mid i \in \mathbb{N}_{\leq n} \}$
(assumptions)
less conditional independence

\[ I_{\text{Ordering}} = \mathcal{G}, B, I', S' \]

\[ S_{\text{Ordering}} = \mathcal{F}, B, I', I' \]

Often, causal ordering gives simple and understandable models. All orderings are valid, but may yield awkward results.

Compactness and node ordering
Formalized in d-separation

(otherwise, explaining away)
free: \( x \) and independent

conditional connections •

claimed: \( x \) and independent

divergent connections •

claimed: \( x \) and independent

serial connections •

Conditional Independence Relations in BNs
Given \( D \), the \( F_i \)'s are independent (diverse connections).

Only \((N+1)\) parameters.

\[
(d|^{\mathcal{N}F})d \cdots (d|^{\mathcal{F}_2}d)(d|^{\mathcal{F}_1}d)(d) = (^{\mathcal{N}F}d, ^{\mathcal{F}_2}d, ^{\mathcal{F}_1}d, d)
\]

Finding (symptom) \( F \) mutually exclusive

\{\{u_1p', \cdots , u_n p' \} \} = D,

Disease state \( D \) : Disease.

Model often used for (naive) medical diagnoses.

**Naive Bayes**
... Extensions: mixtures of ...

Blind source separation (cocktail party problem).

\[ X \sim \text{Gaussian}, \quad XW = \lambda X + \text{noise}. \]

\[ \text{ICA} \]

Independent Component Analysis (ICA):

Dimensionality reduction.

Factor Analysis: \[ X \sim \text{Gaussian}, \quad XM = \lambda X + \text{noise}. \]

Clustering

Mixture of Gaussians: discrete, \[ \lambda \] Gaussian.

\[ y: \text{observations} \]

\[ x: \text{hiddeens (Latent)} \]

Hidden variable models for determining
Temporal models: Dynamic Bayesian networks

Typical applications: Speech recognition (HMM), Tracking

Examples: Hidden Markov Model (HMM), Kalman filter

\( P(\mathbf{X}^T | \mathbf{Y}) \) and \( P(\mathbf{X}^T | \mathbf{X}) \)

Time invariant transition probabilities
The message from node 1 to node 2:

\[(\forall x') \forall f (\exists x', \exists z_2 x ) f_{12} \equiv \]

\[(\forall x', \exists x, \exists z_2) f(2_1 z_2 f) f_{12} \equiv \]

\[(\forall x', \exists x, \exists z_2 f (2_1 z_2 f) f_{12} [ (2_1 z_1 f \cap x] = \]

\[(\forall x', \exists x, \exists z_2 f (2_1 z_2 f) f_{12} [ (2_1 z_1 f + (2_1 z_2 f) f_{12} [ (2_1 z_1 f + (\forall x', \exists x, \exists z_2 f (2_1 z_1 f) f_{12} [ x ]

First, sum over \( x \):

\[(\forall x', x, \exists z_2, x_1) f \cap x = (\forall x', x, z_2, x_1) f \cap x \]

Compute the marginal „marginal“ by local computation
\[(\exists x)\exists y (\exists x) \exists z y =
\]
\[
(\forall x, \exists y (\exists x) \exists z y)
\]
\[
(\forall x) (\exists y (\exists x) \exists z y)
\]
\[
(\forall x, \exists y (\exists x) \exists z y)
\]
\[
(\forall x) (\exists y (\exists x) \exists z y)
\]
\[
(\exists x) f
\]

So, \((\exists x) f\) is product over incoming messages.

\[
(\forall x, \exists y (\exists x) \exists z y)
\]
\[
(\forall x) (\exists y (\exists x) \exists z y)
\]

Sum over \(x\) can also be interpreted as messages:

\[
(\exists x) f = (\forall x, \exists y (\exists x) \exists z y)
\]

Finally, \(\forall x\) sum over \(x\),

\[
(\forall x, \exists y (\exists x) \exists z y)
\]
\[
(\forall x) (\exists y (\exists x) \exists z y)
\]

Next, \(\exists x\) sum over \(x\),
\[
\begin{align*}
\text{etc} & \quad (\exists x, \exists \zeta \exists \chi) \exists \gamma = (\exists x) \cdot f \\
(1) x & \exists \gamma = (1) x \cdot f
\end{align*}
\]

All marginals can now be computed.

First compute messages, from "outside inwards"

local computations

Message propagation (computing marginals by

\[
\begin{align*}
(\forall x, \exists \zeta, \exists \chi) \exists \gamma & = (\forall x) \cdot (\exists \zeta, \exists \chi) \cdot (\exists \gamma) \\
(\exists x, \exists \zeta) \cdot (\exists \chi) & = (\exists x) \cdot (\exists \zeta, \exists \chi) \\
(\exists x, \exists \zeta) \cdot (\exists \chi) & = (\exists x) \cdot (\exists \zeta, \exists \chi) \\
(\exists x, \exists \zeta) \cdot (\exists \chi) & = (\exists x) \cdot (\exists \zeta, \exists \chi)
\end{align*}
\]
Computational complexity is linear in the number of nodes.

Node marginal can be computed by multiplying messages.

If a node received the messages from its both neighbours, the received the messages of its other neighbour.

A node i can send its message to its neighbour as soon as

Message propagation in chains
\[(x) f \{ y \neq x \} \subseteq = (\exists x) y \forall (\exists x) y \forall (\exists x) y \forall = (\exists x) y f \]

Messages:

\[(\exists x, \forall x) y f (\exists x, \forall x) y f \subseteq = (\exists x) y f \]
\[(\exists x, \forall x) y f (\exists x, \forall x) y f \subseteq = (\exists x) y f \]
\[(\exists x, \forall x) y f (\exists x, \forall x) y f \subseteq = (\exists x) y f \]

Messages in trees:

\[f\]
Computational complexity is linear in the number of nodes.

If a node received the messages from all its neighbors, the node can send its message to its neighbor as soon as it received the messages of its other neighbors.

Message propagation in trees
Efficient implementation is the function tree algorithm.

Heuristics exists.

To find the optimal clustering is NP hard, but reasonable clusters.

Computation is exponential in the number of nodes in the clusters.

Computation is linear in number of clusters (cliques or super-nodes).

Graph is to be transformed into a tree of clusters of nodes.

Inference in General (loop graphs)
Maximum Likelihood: $\max_{\theta} P(D|\theta) = \max_{\theta} P(D|\theta) P(\theta) \propto P(D, \theta)$

MAP (Maximum a posteriori): $\max_{\theta} P(D|\theta) P(\theta) \propto P(D) P(\theta)$

Bayes' Rule: $P(D|\theta) = \frac{P(D, \theta)}{P(\theta)}$

\[
\begin{align*}
(\theta \mid D) &= \frac{P(D \mid \theta) P(\theta)}{\int P(D \mid \theta) P(\theta) \, d\theta} \\
&= \frac{P(D \mid \theta) P(\theta)}{\int P(D \mid \theta, \omega) P(\theta \mid \omega) \, d\theta} \\
&= \frac{P(D \mid \theta) P(\theta)}{\int P(D \mid \theta, \omega) P(\theta \mid \omega) \, d\theta}
\end{align*}
\]

E.g. prediction of $x_{\text{new}}$.

Full Bayesian Learning: probabilistic distribution of models.

E.g. model parameters (e.g. conditional probability tables) $\theta = \{x_1, x_2, \ldots, x_n\}$, data set $D$.
\[
\frac{(c = C, q = B, v = V) N^\alpha}{(c = C, q = B, v = V) N} = (c = C, q = B | v = V) \Pr
\]

**M-step** Maximize likelihood

\[
(c = C, q = V | v = B) \Pr (c = C, q = V) N
\]

\[
= (c = C, q = B, v = V) N
\]

*: \text{if } B \text{ is hidden}

**E-step** Estimate missing values by the current model \( \Pr \text{.}

**Missing data (Hidden variables): EM-algorithm:**

\[
\frac{(c = C, q = B, v = V) N^\alpha}{(c = C, q = B, v = V) N} = (c = C, q = B | v = V) \Pr
\]

*: Continuing

**Fully observable:** Maximum likelihood solution (conditional)

**ML in a Bayesian network**
\[ \theta \mathcal{P}(\theta) d(\theta | \{x\}) d \prod_{d} (\theta | \text{mean} x) d \int_{\infty}^{\theta} (dx' \cdots x'_{\text{new}} | x_{\text{new}}) d \]

Learning from data as a graphical model
Learning and reasoning in same framework

- From data

Modeling with expert knowledge easily combined with learning

- Model in the cluster-size

Efficient inference (linear in the number of clusters, exponential in dependencies)

- Large scale modeling feasible by exploiting conditional independences

- Large scale modeling feasible by exploiting conditional independences

Convenient graphical language to describe a large class of

Bayesian networks are a class of probabilistic models

- Reasoning with uncertainty

- Probability theory provides a consistent and correct framework for

Summary