Probabilistic Reasoning

Lecturer: dr. S. Renooij
tel. (253)9266, e-mail: s.renooij@uu.nl

Prerequisites: probability theory & graph theory

Literature: syllabus & slides & studymanual

Form: lectures & exercises (formative self assessment)
(tip: discuss exercises on Blackboard forum)

Grading: practical assignments & written exam

Additional info: see course website: http://www.cs.uu.nl/docs/vakken/prob/
Chapter 1:

Introduction
Reasoning under uncertainty

In numerous application areas of knowledge-based systems we have

- uncertainty concerning the general domain knowledge;
- problem-specific information that is often uncertain, incomplete and even contradictory.

A knowledge-based system should be capable of dealing with these types of knowledge.
Application of probability theory

Consider a discrete joint probability distribution $\Pr$ on a set of statistical variables. In general we have that:

- the representation of $\Pr$ requires exponential space (and how do you get the numbers?);
- calculating the (conditional) probability of a value of a variable by conditioning and marginalisation requires exponential time.

This cannot be improved without additional knowledge about the probability distribution.
The diagnosis problem

Let $\mathbf{H} = \{ h_1, \ldots, h_n \}, n \geq 1$, be a set of hypotheses, and let $\mathbf{E} = \{ e_1, \ldots, e_m \}, m \geq 1$, be a set of relevant findings (evidence).

These hypotheses and findings together span a search space:

Suppose that the findings $\mathbf{e} \subseteq \mathbf{E}$ are actually observed. The diagnosis is the set of hypotheses $\mathbf{h} \subseteq \mathbf{H}$ that best explains $\mathbf{e}$.
Pioneering in the ’60s

HYPOTHESES: $h_1, \ldots, h_n$  
$H = \{h_1, \ldots, h_n\}$

FINDINGS: $e_1, \ldots, e_j, \ldots, e_m$  
$E = \{e_1, \ldots, e_m\}$

Determine the diagnosis given findings $e \subseteq E$.

The approach: Compute for each $h \subseteq H$ the probability

$$Pr(h \mid e) = \frac{Pr(e \mid h) \cdot Pr(h)}{Pr(e)}$$

Drawback: An exponential number of probabilities need to be computed; storage is also exponential.
Pioneering in the ’60s

HYPOTHESES: $h_1, \ldots, h_n$

$\Omega = \{h_1, \ldots, h_n\}$

FINDINGS: $e_1, \ldots, e_m$

$E = \{e_1, \ldots, e_m\}$

Determine the diagnosis given findings $e \subseteq E$.

The approach: Assume $h_i \in H$ mutually exclusive,

$\bigcup_{i=1}^{n} \{h_i\} = \Omega$.

Then, compute for each $h_i \in H$:

$$
Pr(h_i \mid e) = \frac{Pr(e \mid h_i) Pr(h_i)}{Pr(e)} = \frac{Pr(e \mid h_i) Pr(h_i)}{\sum_{k=1}^{n} Pr(e \mid h_k) Pr(h_k)}
$$

Drawback: We compute only $n - 1$ probabilities, but computation still requires an exponential number of probabilities.
Pioneering in the ’60s

Determine the diagnosis given findings $e = \{e_p, \ldots, e_q\}$, $1 \leq p, q \leq m$.

**The approach:** Assume in addition that all findings $e_1, \ldots, e_m$ are conditionally independent given $h_i$, $i = 1, \ldots, n$. Then:

$$
\Pr(h_i \mid e) = \frac{\Pr(e_p, \ldots, e_q \mid h_i) \Pr(h_i)}{\sum_{k=1}^{n} \Pr(e_p, \ldots, e_q \mid h_k) \Pr(h_k)}
$$

$$
= \frac{\Pr(e_p \mid h_i) \cdots \Pr(e_q \mid h_i) \Pr(h_i)}{\sum_{k=1}^{n} \Pr(e_p \mid h_k) \cdots \Pr(e_q \mid h_k) \Pr(h_k)}
$$

**Benefit:** Only $m \cdot n$ conditional probabilities and $n - 1$ prior probabilities are required for the computation.
GLADYS (GLASGOW DYSPEPSIA SYSTEM) is a system for diagnosing dyspepsia.

The global structure of the system:

- **Interview**
- **Differential diagnosis**
- **Therapy selection**

Developed with data collected from ±1200 patients.

## Symptoms and diseases

**Context**: patients with an Ulcer. **Question**: which type?

<table>
<thead>
<tr>
<th></th>
<th>duodenal ulcer $(n = 248)$</th>
<th>gastric ulcer $(n = 43)$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sex:</strong></td>
<td>male</td>
<td>169</td>
</tr>
<tr>
<td></td>
<td>female</td>
<td>79</td>
</tr>
<tr>
<td><strong>Age:</strong></td>
<td>&lt; 26</td>
<td>43</td>
</tr>
<tr>
<td></td>
<td>26 - 40</td>
<td>82</td>
</tr>
<tr>
<td></td>
<td>41 - 55</td>
<td>87</td>
</tr>
<tr>
<td></td>
<td>&gt; 55</td>
<td>36</td>
</tr>
<tr>
<td><strong>Daily pain:</strong></td>
<td>yes</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>no</td>
<td>214</td>
</tr>
<tr>
<td><strong>Effect food on pain:</strong></td>
<td>worsens</td>
<td>44</td>
</tr>
<tr>
<td></td>
<td>no effect</td>
<td>82</td>
</tr>
<tr>
<td></td>
<td>relieves</td>
<td>104</td>
</tr>
<tr>
<td><strong>Probability</strong></td>
<td></td>
<td>0.85</td>
</tr>
</tbody>
</table>
The idea

Let $\Pr$ be a joint distribution on the diagnosis search space including hypothesis $h$ and observed findings $e$.

The prior odds for $h$, and posterior odds for $h$ given $e$, are defined by

$$O(h) = \frac{\Pr(h)}{1 - \Pr(h)} = \frac{\Pr(h)}{\Pr(\neg h)}, \quad \text{and} \quad O(h \mid e) = \frac{\Pr(h \mid e)}{\Pr(\neg h \mid e)}$$

Assume that all findings $e_i \in e$ are conditionally independent given $h$, then

$$O(h \mid e) = \frac{\Pr(e \mid h) \cdot \Pr(h)}{\Pr(e \mid \neg h) \cdot \Pr(\neg h)} = \prod_i \frac{\Pr(e_i \mid h)}{\Pr(e_i \mid \neg h)} \cdot O(h)$$

Now consider the following transformation: $10 \cdot \ln O(h \mid e)$. . .
The idea (cntd)

Applying the transformation $10 \cdot \ln$ to

$$O(h \mid e) = \prod_i \lambda_i \cdot O(h), \text{ where } \lambda_i = \frac{\Pr(e_i \mid h)}{\Pr(e_i \mid \neg h)}$$

results in a score $s$:

$$s = 10 \cdot \ln O(h \mid e) = 10 \cdot \ln O(h) + \sum_i 10 \cdot \ln \lambda_i = w_0 + \sum_i w_i$$

where $w_i$ is a weight for finding $e_i$.

The probability $\Pr(h \mid e)$ is now computed from

$$\Pr(h \mid e) = \frac{O(h \mid e)}{1 + O(h \mid e)} = \frac{e^{s/10}}{1 + e^{s/10}} = \frac{1}{1 + e^{-s/10}}$$
A scoring system

\[ h: \text{ duodenal ulcer (du)} \quad \neg h: \text{ gastric ulcer (gu)} \]
\[ (n = 248) \quad (n = 43) \]

\begin{tabular}{|c|c|c|}
\hline
 & \text{male (m)} & \text{female (f)} \\
\hline
\text{du} & 169 & 79 \\
\text{gu} & 17 & 26 \\
\hline
\end{tabular}

Calculation of probabilities, likelihood ratios and weights:

\[ \Pr(m \mid du) = \frac{169}{248} \sim 0.68, \quad \Pr(m \mid gu) \sim 0.40 \quad \Rightarrow \]

\[ \lambda_m = \frac{\Pr(m \mid du)}{\Pr(m \mid gu)} = \frac{0.68}{0.40} \sim 1.7 \quad \Rightarrow \quad w_m = 10 \cdot \ln \lambda_m \sim 5 \]

\[ \Pr(f \mid du) = \frac{79}{248} \sim 0.32, \quad \Pr(f \mid gu) \sim 0.60 \quad \Rightarrow \]

\[ \lambda_f = \frac{\Pr(f \mid du)}{\Pr(f \mid gu)} = \frac{0.32}{0.60} \sim 0.53 \quad \Rightarrow \quad w_f = 10 \cdot \ln \lambda_f \sim -6 \]
# Symptoms and their weights

<table>
<thead>
<tr>
<th></th>
<th>duodenal ulcer ((n = 248))</th>
<th>gastric ulcer ((n = 43))</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sex:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>male</td>
<td>169</td>
<td>17</td>
<td>5</td>
</tr>
<tr>
<td>female</td>
<td>79</td>
<td>26</td>
<td>−6</td>
</tr>
<tr>
<td><strong>Age:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt; 26</td>
<td>43</td>
<td>1</td>
<td>18</td>
</tr>
<tr>
<td>26 - 40</td>
<td>82</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>41 - 55</td>
<td>87</td>
<td>19</td>
<td>−2</td>
</tr>
<tr>
<td>&gt; 55</td>
<td>36</td>
<td>18</td>
<td>−10</td>
</tr>
<tr>
<td><strong>Daily pain:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>yes</td>
<td>21</td>
<td>11</td>
<td>−12</td>
</tr>
<tr>
<td>no</td>
<td>214</td>
<td>27</td>
<td>3</td>
</tr>
<tr>
<td><strong>Effect food on pain:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>worsens</td>
<td>44</td>
<td>11</td>
<td>−4</td>
</tr>
<tr>
<td>no effect</td>
<td>82</td>
<td>9</td>
<td>4</td>
</tr>
<tr>
<td>relieves</td>
<td>104</td>
<td>17</td>
<td>0</td>
</tr>
</tbody>
</table>

prior: 0.85 0.15 17
A 30 year old woman reports to the clinic. She has pain in the abdominal area, but not on a daily basis; the pain worsens as soon as she eats.

Calculation of the score:

- the initial score: +17
- the patient is female: −6
- her age is 30: +10
- she is in pain, but not every day: +3
- food intake worsens the pain: −4

$$\frac{-20}{10}$$

$$\frac{-20}{10}$$

Given that the patient has one of the two diseases, duodenal ulcer and gastric ulcer, she has with probability

$$\left(1 + e^{-\frac{20}{10}}\right)^{-1} \approx 1.14^{-1} \approx 0.88$$

a duodenal ulcer and a gastric ulcer with probability 0.12.
Reviewing ‘Idiot’s Bayes’

The naive Bayes approach is

• mathematically correct, and
• computationally easy.

However, for larger applications

• underlying assumptions usually unacceptable;
• # of hypotheses often large → undoable to compute each \( \Pr(h_i \mid e) \);
• at the time, often not enough information for reliable probability assessments.
The most likely hypothesis given observed findings is determined as follows:

- prune the search space using **heuristic rules**;
- **approximate** the missing probabilities required, for example with:

\[
Pr(e_i \land e_j) = \min\{Pr(e_i), Pr(e_j)\};
\]

- select the hypothesis with the highest probability.
Reviewing the quasi-probabilistic models

The quasi-probabilistic models are

- computationally easy, and
- easy to use,

even for larger applications.

However, these models are

- mathematically incorrect, and
- even as an approximation model not convincing.
The rehabilitation of probability theory in the ’80s

A Bayesian network is a very compact representation of a joint probability distribution $\Pr$. Such a network comprises

- **qualitative** knowledge of $\Pr$: a graphical representation of the independences between the variables involved;
- **quantitative** knowledge of $\Pr$: conditional probability distributions that describe $\Pr$ ‘locally’ per group of variables.

Associated with a Bayesian network are algorithms for computing probabilities and for processing evidence.
An example: Classical Swine Fever (CSF)

The classical swine fever network is a knowledge-based system for the early detection of classical swine fever (varkenspest).

- early detection of CSF is important, but hard;
- the network has been developed in cooperation with 2 veterinarians of the Central Veterinary Institute of Wageningen UR;
- part of european EPIZONE project;
- veterinarians all over the country collect data with PDAs.
The Classical swine fever network: graphical structure
The Classical swine fever network: probability tables

\[ \Pr(A) \mid B \land C \]

Pr(Appetite \mid BodyTemp \land Malaise)
Classical swine fever: prior probabilities

Faeces

Prim. Other Infection

Reproduction phase

Respiratory problems
Classical swine fever: diagnostic reasoning
Classical swine fever: prognostic reasoning
Definition:
A Bayesian network is a pair $\mathcal{B} = (G, \Gamma)$ such that

- $G$ is an **acyclic directed graph** with nodes representing a set of **statistical variables** $V$;
- $\Gamma = \{\gamma_{V_i} \mid V_i \in V\}$ is a set of **assessment functions**.

Property:

\[
\Pr(V) = \prod_{V_i \in V} \gamma_{V_i}(V_i \mid \rho(V_i))
\]

defines a **joint probability distribution** $\Pr$ on $V$ such that $G$ is a **directed I-map** for the **independence relation** $I_{\Pr}$ of $\Pr$. 
About this course . . .

The following subjects will be addressed in this course:

• the syntactics and semantics of a Bayesian network;
• algorithms for reasoning with a Bayesian network;
• methods for constructing a Bayesian network for a domain of application;
• methods for evaluating a Bayesian network’s performance and behaviour;
• algorithms for controlling reasoning;
Overview of subjects

Graph theory

Graphical models

Framework / formalism of probabilistic networks

Inference algorithms (Pearl)

Translation $V \rightarrow V(\mathcal{G})$

Qualitative notion of independence

Assessment functions

Domain Knowledge

Probabilistic network (application)

"Decision support" system

"Meta" algorithms

2.1

3.1

3.2

4.1

4.2

5

6

Group "Decision support systems"
Department of Information and Computing Sciences
Universiteit Utrecht
Chapter 2:

Preliminaries
Statistical variables

Let \( V = \{V_1, \ldots, V_n\} \), \( n \geq 1 \), be a set of statistical variables. Each variable \( V_i \in V \) can take on one of \( m \geq 2 \) values; for now we consider 2-valued variables:

- \( V_i = true \), denoted by \( v_i \);
- \( V_i = false \), denoted by \( \neg v_i \) (or by \( \overline{v_i} \)).

The set \( V \) spans a Boolean Algebra of logical propositions \( \mathcal{V} \):

- \( T(\text{rue}), F(\text{alse}) \in \mathcal{V} \);
- for all variables \( V_i \in V \) we have that \( v_i \in \mathcal{V} \);
- for all \( x \in \mathcal{V} \) we have that \( \neg x \in \mathcal{V} \);
- for all \( x, y \in \mathcal{V} \) we have that \( x \land y \in \mathcal{V} \) and \( x \lor y \in \mathcal{V} \).

The elements of \( \mathcal{V} \) obey the usual rules of propositional logic.
The joint probability distribution

Definition:

Let \( \mathcal{V} \) be the Boolean Algebra of propositions spanned by a set of statistical variables \( \mathcal{V} \). Let \( \Pr : \mathcal{V} \rightarrow [0, 1] \) be a function such that

- \( \Pr \) is positive: for each \( x \in \mathcal{V} \) we have that \( \Pr(x) \geq 0 \) and, more specifically, \( \Pr(F) = 0 \);
- \( \Pr \) is normed: \( \Pr(T) = 1 \);
- \( \Pr \) is additive: we have, for each \( x, y \in \mathcal{V} \) with \( x \land y \equiv F \), that \( \Pr(x \lor y) = \Pr(x) + \Pr(y) \).

The function \( \Pr \) is a joint probability distribution on \( \mathcal{V} \); the function value \( \Pr(x) \) is the probability of \( x \).
Independence of propositions

**Definition:** Let $\mathcal{V}$ be the Boolean Algebra of propositions spanned by a set of statistical variables $\mathcal{V}$. Let $Pr$ be a joint probability distribution on $\mathcal{V}$.

Two propositions $x, y \in \mathcal{V}$ are called independent in $Pr$ if

$$Pr(x \land y) = Pr(x) \cdot Pr(y)$$

The propositions $x, y \in \mathcal{V}$ are called conditionally independent given the proposition $z \in \mathcal{V}$ if we have that

$$Pr(x \land y \mid z) = Pr(x \mid z) \cdot Pr(y \mid z)$$
The two notions of independence (1)

- Consider two propositions \( x, y \in \mathcal{V} \) such that \( x \) and \( y \) are independent \(^1\):

  \[ \begin{array}{c}
  x \\
  \hline \\
  y \\
  \end{array} \]

Can \( z \in \mathcal{V} \) exist such that \( x \) and \( y \) are dependent given \( z \)?

- Yes:

\[ \begin{array}{c}
  x \\
  \hline \\
  y \\
  \end{array} \]

\(^1\) The square has area 1, representing the total probability mass.
The two notions of independence (2)

• Consider two propositions \( x, y \in \mathcal{V} \) such that \( x \) and \( y \) are dependent:

Can \( z \in \mathcal{V} \) exist such that \( x \) and \( y \) are conditionally independent given \( z \)?

• Yes:
Configurations

Let $V$ be a set of statistical variables and let $W \subseteq V$.

- a configuration $c_W$ of $W$ is a conjunction of value assignments to the variables from $W$;
- convention: $c_\emptyset = T$;
- $w$ is used to denote a specific configuration of $W$.
- $W$ also indicates all possible configurations to the set $W$ (notation abuse!): $W$ is then considered to be a template that can be filled in with any configuration $c_W$.

Example: Let $W = \{V_1, V_3, V_7\}$. $W = V_1 \land V_3 \land V_7$ denotes a configuration template: filling in values for $V_i$ results in proper propositions/configurations. Some configurations $c_W$ of $W$ are:

$$
V_1 = true \land V_3 = true \land V_7 = false \\

v_1 \land \neg v_3 \land v_7 \\
\neg v_1 \land v_3 \land \neg v_7
$$
### Conventions and notation

In the remainder of this course, for distributions on $\Pr(V)$:

- rather than talking about propositions $x \in \mathcal{V}$ spanned by $\mathcal{V}$
- we refer to configurations $c_\mathcal{V}$ of $\mathcal{V}$

<table>
<thead>
<tr>
<th>Variables/templates (capital)</th>
<th>Set (bold faced)</th>
<th>Singleton</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{V}$</td>
<td>$\mathcal{V}$</td>
<td>$\mathcal{V}$</td>
</tr>
<tr>
<td>$c_\mathcal{V}, v$</td>
<td>$c_\mathcal{V}, v$</td>
<td>$c_\mathcal{V}, v$</td>
</tr>
</tbody>
</table>

- conjunctions are often left implicit: e.g. $v_1 \land v_2$ denotes $v_1 \land v_2$;
- note the following differences (!)
  - probabilities: $\Pr(c_\mathcal{V}), \Pr(c_\mathcal{V}), \Pr(v), \Pr(v), \Pr(v \mid c_\mathcal{E})$
  - distributions: $\Pr(V), \Pr(V), \Pr(V \mid e)$
  - distribution sets: $\Pr(V \mid \mathcal{E}), \Pr(V \mid \mathcal{E})$
Independence of variables

**Definition:** Let $V$ be a set of statistical variables and let $X, Y, Z \subseteq V$. Let $\Pr$ be a joint distribution on $V$.

The set of variables $X$ is called conditionally independent of the set $Y$ given the set $Z$ in $\Pr$, if we have that

$$\Pr(X \mid Y \land Z) = \Pr(X \mid Z)$$

**Remarks:**

- the expression $\Pr(X \mid Y \land Z) = \Pr(X \mid Z)$ represents that $\Pr(c_X \mid c_Y \land c_Z) = \Pr(c_X \mid c_Z)$ holds for all configurations $c_X$, $c_Y$ and $c_Z$ of $X$, $Y$ and $Z$;
- $\Pr(X \mid Y \land Z) = \Pr(X \mid Z) \Rightarrow \Pr(X \land Y \mid Z) = \Pr(X \mid Z) \cdot \Pr(Y \mid Z)$ (what about $\leftarrow$?).
Chapter 3:

Independences and Graphical Representations
\[ P_{d_1}(s_1, s_2, ... s_j) = \frac{P_{d_1}s_1d_1(1-P_{s_2}d_2)...P_{s_j}d_j}{\sum_{\text{all } k} P_{d_k}s_kd_k(1-P_{s_j}d_j)...P_{s_j}d_k} \]
A qualitative notion of independence

**Observation:**

People are capable of making statements about independences among variables without having to perform numerical calculations.

**Conclusion:**

In human reasoning behaviour, the qualitative notion of independence is more fundamental than the quantitative notion of independence.
The (probabilistic) independence relation of a joint distribution

**Definition:** Let $V$ be a set of statistical variables and let $Pr$ be a joint probability distribution on $V$.

The independence relation $I_{Pr}$ of $Pr$ is a set

$I_{Pr} \subseteq \mathcal{P}(V) \times \mathcal{P}(V) \times \mathcal{P}(V)$, defined for all $X, Y, Z \subseteq V$ by

$$(X, Z, Y) \in I_{Pr} \text{ if and only if } Pr(X \mid Y \land Z) = Pr(X \mid Z)$$

**Remarks:**

- $(X, Z, Y) \in I_{Pr}$ will be written as $I_{Pr}(X, Z, Y);$
  $(X, Z, Y) \notin I_{Pr}$ will be written as $\neg I_{Pr}(X, Z, Y);$
- a statement $I_{Pr}(X, Z, Y)$ is called an independence statement for the joint distribution $Pr$. 


Properties of $I_{Pr}$: symmetry

**Lemma:** \( I_{Pr}(X, Z, Y) \) if and only if \( I_{Pr}(Y, Z, X) \)

**Proof:**

\[
I_{Pr}(X, Z, Y) \iff \Pr(X \mid Y \wedge Z) = \Pr(X \mid Z)
\]

\[
\iff \frac{\Pr(X \wedge Y \wedge Z)}{\Pr(Y \wedge Z)} = \frac{\Pr(X \wedge Z)}{\Pr(Z)}
\]

\[
\iff \frac{\Pr(X \wedge Y \wedge Z)}{\Pr(X \wedge Z)} = \frac{\Pr(Y \wedge Z)}{\Pr(Z)}
\]

\[
\iff \Pr(Y \mid X \wedge Z) = \Pr(Y \mid Z)
\]

\[
\iff I_{Pr}(Y, Z, X)
\]
Properties of $I_{Pr}$: decomposition

**Lemma:** $I_{Pr}(X, Z, Y \cup W) \Rightarrow I_{Pr}(X, Z, Y) \land I_{Pr}(X, Z, W)$

**Proof:** (sketch) (Note: $c_{Y \cup W} = c_Y \land c_W$!) Suppose that

$Pr(X \mid Y \land W \land Z) = Pr(X \mid Z)$. Then, by definition,

$$Pr(X \land Y \land W \land Z) = Pr(Y \land W \land Z) \cdot \frac{Pr(X \land Z)}{Pr(Z)}$$

For $Pr(X \mid Y \land Z)$ we find that

$$Pr(X \mid Y \land Z) = \frac{Pr(X \land Y \land Z)}{Pr(Y \land Z)}$$

$$= \sum_{c_W} \frac{Pr(X \land Y \land Z \land c_W)}{Pr(Y \land Z)}$$

$$= \frac{Pr(X \land Z)}{Pr(Z)} = Pr(X \mid Z)$$
Properties of $I_{Pr}$: weak union, contraction

**Lemma:**

- if $I_{Pr}(X, Z, Y \cup W)$ then $I_{Pr}(X, Z \cup W, Y)$ (weak union);

- if $I_{Pr}(X, Z, W)$ and $I_{Pr}(X, Z \cup W, Y)$ then $I_{Pr}(X, Z, Y \cup W)$ (contraction)

- (for strictly positive $Pr$ also the intersection property holds; see syllabus)

**Proof:** left as exercise 3.1.

What about $\iff$?
The definition of the independence relation

Joint Distribution $\text{Pr}$

Independence relation $I_{\text{Pr}}$

Properties: symmetry, decomposition, weak union, contraction

Axioms: symmetry, decomposition, weak union, contraction

Independence relation $I$

Judea Pearl: 1936 -
The (qualitative) independence relation \( I \)

**Definition:**
Let \( V \) be a set of statistical variables and let \( X, Y, Z, W \subseteq V \).

An independence relation \( I \) on \( V \) is a ternary relation \( I \subseteq \mathcal{P}(V) \times \mathcal{P}(V) \times \mathcal{P}(V) \) that satisfies the following properties:

- if \( I(X, Z, Y) \) then \( I(Y, Z, X) \);
- if \( I(X, Z, Y \cup W) \) then \( I(X, Z, Y) \) and \( I(X, Z, W) \);
- if \( I(X, Z, Y \cup W) \) then \( I(X, Z \cup W, Y) \);
- if \( I(X, Z, W) \) and \( I(X, Z \cup W, Y) \) then \( I(X, Z, Y \cup W) \).

The first property is called the symmetry *axiom*; the second is called the decomposition axiom; the third is referred to as the weak union axiom; the last one is called contraction.
Lemma:
Let $I$ be an independence relation on a set of statistical variables $V$. We have that

if $I(X, Z, Y)$ and $I(X \cup Z, Y, W)$ then $I(X, Z, W)$

for all $X, Y, Z, W \subseteq V$.

Proof:
We observe that

$I(X \cup Z, Y, W) \Rightarrow_{\text{symm}} I(W, Y, X \cup Z) \Rightarrow_{\text{weakunion}}$

$\Rightarrow I(W, Y \cup Z, X) \Rightarrow_{\text{symm}} I(X, Y \cup Z, W)$

From $I(X, Z, Y)$, $I(X, Y \cup Z, W)$ and the contraction axiom we have that $I(X, Z, W \cup Y)$; decomposition now gives $I(X, Z, W)$. ■
Representing independences

Different ways exist of representing an independence relation:

• all independence statements of the relation are explicitly stated;
• only the independence statements of a suitable subset of the relation are explicitly stated — all other statements are implicitly represented by means of the axioms;
• the independence relation is coded in a graph;
• . . .
An example

Consider $V = \{V_1, V_2, V_3, V_4\}$ and independence relation $I$ on $V$:

- $I(\{V_1\}, \emptyset, \{V_4\})$
- $I(\{V_2\}, \emptyset, \{V_4\})$
- $I(\{V_3\}, \emptyset, \{V_4\})$
- $I(\{V_4\}, \emptyset, \{V_1\})$
- $I(\{V_2\}, \emptyset, \{V_1, V_4\})$
- $I(\{V_2, V_4\}, \emptyset, \{V_1\})$
- $I(\{V_2, V_4\}, \emptyset, \{V_1, V_4\})$
- $I(\{V_4\}, \{V_1\}, \{V_2\})$
- $I(\{V_4\}, \{V_1\}, \{V_3\})$
- $I(\{V_4\}, \{V_1\}, \{V_2, V_3\})$
- $I(\{V_1\}, \{V_2\}, \{V_4\})$
- $I(\{V_1\}, \{V_2\}, \{V_3\})$
- $I(\{V_1\}, \{V_3\}, \{V_4\})$
- $I(\{V_1\}, \{V_3\}, \{V_2\})$
- $I(\{V_1, V_3\}, \{V_2\}, \{V_4\})$
- $I(\{V_1, V_3\}, \{V_2\}, \{V_1\})$
- $I(\{V_4\}, \{V_2\}, \{V_1\})$
- $I(\{V_4\}, \{V_2\}, \{V_3\})$
- $I(\{V_4\}, \{V_2\}, \{V_1, V_3\})$
- $I(\{V_4\}, \{V_2\}, \{V_1, V_3\})$
- $I(\{V_4\}, \{V_3\}, \{V_2\})$
- $I(\{V_4\}, \{V_3\}, \{V_1\})$
- $I(\{V_4\}, \{V_3\}, \{V_1\})$
- $I(\{V_4\}, \{V_3\}, \{V_1\})$
- $I(\{V_1\}, \{V_2\}, \{V_3\}, \{V_4\})$
- $I(\{V_1\}, \{V_2\}, \{V_3\}, \{V_4\})$
- $I(\{V_1\}, \{V_2\}, \{V_3\}, \{V_4\})$
The representation of an independence relation in an undirected graph

Consider an independence relation \( I \) and an undirected graph:

the global idea is:

- represent each variable \( V_i \) by a node \( V_i \) in the graph, and v.v.;
- code the independence statements of \( I \) by means of missing edges.
The separation criterion: introduction

Definition:
Let $G = (V_G, E_G)$ be an undirected graph with edges $E_G$ and nodes $V_G = \{V_1, \ldots, V_n\}, n > 1$.

Let $s$ be a path in $G$ from a node $V_i$ to a node $V_j$.

The path $s$ is blocked by a set of nodes $Z \subseteq V_G$, if at least one node from $Z$ is on the path $s$.

If $s$ is not blocked by $Z$, the path is called active given $Z$. 
The separation criterion

Definition:
Let \( G = (V_G, E_G) \) be an undirected graph. Let \( X, Y, Z \subseteq V_G \) be sets of nodes in \( G \).

The set \( Z \) separates the set \( X \) from \( Y \) in \( G \)— Notation: \( \langle X \mid Z \mid Y \rangle_G \)— if every simple path in \( G \) from a node in \( X \) to a node in \( Y \) is blocked by \( Z \).

Remarks:
- the above notion is known as the separation criterion for undirected graphs;
- if there is no path between the nodes \( X \) and \( Y \) in a graph \( G \), then \( \langle X \mid \emptyset \mid Y \rangle_G \).
An example

Which of the following separation statements are valid?

a) $\langle \{V_1\} \mid \{V_2\} \mid \{V_3, V_6\}\rangle_G$

b) $\langle \{V_4\} \mid \{V_2, V_5\} \mid \{V_6\}\rangle_G$

c) $\langle \{V_4\} \mid \{V_1, V_2, V_5\} \mid \{V_6\}\rangle_G$

d) $\langle \{V_1\} \mid \{V_4\} \mid \{V_5\}\rangle_G$

e) $\langle \{V_1, V_5, V_6\} \mid \emptyset \mid \{V_7\}\rangle_G$

f) $\langle \{V_2\} \mid \{V_5\} \mid \{V_7\}\rangle_G$

g) $\langle \{V_1\} \mid \{V_5\} \mid \{V_2\}\rangle_G$
Independence relations and undirected graphs

Definition: Let $I$ be an independence relation on a set of statistical variables $V$. Let $G = (V_G, E_G)$ be an undirected graph with $V_G = V$.

- graph $G$ is called a dependency map (D-map) for $I$ if for all $X, Y, Z \subseteq V$ we have:
  
  \[
  \text{if } I(X, Z, Y) \text{ then } \langle X \mid Z \mid Y \rangle_G; 
  \]

- graph $G$ is called an independency map (I-map) for $I$ if for all $X, Y, Z \subseteq V$ we have:
  
  \[
  \text{if } \langle X \mid Z \mid Y \rangle_G \text{ then } I(X, Z, Y); 
  \]

- graph $G$ is called a perfect map (P-map) for $I$ if $G$ is both a dependency map and an independency map for $I$. 

undirected D-maps: what do they tell?

Let $I$ be an independence relation and $G$ an undirected graph.

Consider a D-map for $I$, then

$V_1$ and $V_2$ neighbours $\implies V_1, V_2$ dependent

$\neg \langle \{V_1\} \mid Z \mid \{V_2\} \rangle_G \implies \neg I(\{V_1\}, Z, \{V_2\})$

$V_1$ and $V_2$ non-neighbours $\implies$ ??

$\langle \{V_1\} \mid Z \mid \{V_2\} \rangle_G$ dependent

$\langle \{V_1\} \mid Z \mid \{V_2\} \rangle_G$ independent

$\langle \{V_1\} \mid Z \mid \{V_2\} \rangle_G$ conditionally independent

Note: statements hold for all $Z \subseteq V_G \setminus (\{V_1\} \cup \{V_2\})$!
An example

Consider the independence relation $I$ on $V = \{V_1, \ldots, V_4\}$, defined by

$$I(\{V_1\}, \{V_2, V_3\}, \{V_4\}) \text{ and } I(\{V_2\}, \{V_1, V_4\}, \{V_3\})$$

Which of the following undirected graphs are examples of D-maps for $I$?
Undirected I-maps: what do they tell?

Let $I$ be an independence relation and $G$ an undirected graph.

Consider an I-map for $I$, then

$V_1$ and $V_2$ non-neighbours $\implies V_1, V_2$ (cond.) independent

$\langle \{V_1\} \mid Z \mid \{V_2\} \rangle_G \quad I(\{V_1\}, Z, \{V_2\})$

$V_1$ and $V_2$ neighbours $\implies ??$

$\neg \langle \{V_1\} \mid Z \mid \{V_2\} \rangle_G \quad$ dependent

$\quad$ independent

$\quad$ conditionally independent

Note: statements hold for all $Z \subseteq V_G \setminus (\{V_1\} \cup \{V_2\})$!
An example

Consider the independence relation $I$ on $V = \{V_1, \ldots, V_4\}$, defined by

$$I(\{V_1\}, \{V_2, V_3\}, \{V_4\}) \text{ and } I(\{V_2\}, \{V_1, V_4\}, \{V_3\})$$

Which of the following undirected graphs are examples of $I$-maps for $I$?
Properties of $I$

Let $I$ be an independence relation on a set of statistical variables $V$.

**Lemma:**
Every independence relation $I$ has an undirected D-map.

**Proof:**
The undirected graph $G = (V, \emptyset)$ is a D-map for $I$. ■

**Lemma:**
Every independence relation $I$ has an undirected I-map.

**Proof:**
The undirected graph $G' = (V, V \times V)$ is an I-map for $I$. ■
An example

Consider the independence relation $I$ on $V = \{V_1, \ldots, V_4\}$, defined by

$$I(\{V_1\}, \{V_2, V_3\}, \{V_4\}) \text{ and } I(\{V_2\}, \{V_1, V_4\}, \{V_3\})$$

The following undirected graph is a perfect map for $I$:

Is this P-map for $I$ unique? Does every $I$ have a P-map?
Consider an experiment with two coins and a bell: the bell sounds iff the two coins have the same outcome after a toss.

Consider: variable $C_1$: the outcome of tossing coin one; variable $C_2$: the outcome of tossing coin two; variable $B$: whether or not the bell sounds; independence relation $I$ for this experiment.

We have, among others, that

$I(\{C_1\}, \emptyset, \{C_2\}) \quad \neg I(\{C_1\}, \{B\}, \{C_2\})$
$I(\{C_1\}, \emptyset, \{B\}) \quad \neg I(\{C_1\}, \{C_2\}, \{B\})$
$I(\{C_2\}, \emptyset, \{B\}) \quad \neg I(\{C_2\}, \{C_1\}, \{B\})$

This independence relation is an example of an independence relation with an induced dependency.
Reconsider the experiment with the two coins and the bell.

- the following graph is a D-map for the independence relation $I$ of this experiment:

- the following graph is an I-map for $I$:

- Does $I$ have a perfect map?
The representation of an independence relation in a directed graph

Consider an independence relation \( I \) and a directed graph \( G \):

The global idea is:

- represent each variable \( V_i \) of \( I \) by a node \( V_i \) in \( G \), and v.v.;
- code the independence statements of \( I \) by means of missing arcs in the graph;
- use the direction of the arcs to represent induced dependencies.
The formalism of the directed graph is more expressive than the formalism of the undirected graph:
Causality?

Consider the following examples:

1. length → age → reading
2. weather → harvest → grain price
3. burglar → burglary alarm → earthquake
Introduction, continued

We aim to represent the following (in)dependences with directed graphs:

1. \( I(\{V_2\}, \emptyset, \{V_3\}) \) and \( \neg I(\{V_2\}, \{V_1\}, \{V_3\}) \):

```
V_2  V_3
  ^   ^
  |   |
V_1
```

2. \( I(\{V_2\}, \{V_1\}, \{V_3\}) \) and \( \neg I(\{V_2\}, \emptyset, \{V_3\}) \):

```
V_2  V_3
  ^   ^
  |   |
V_1
```

3. \( I(\{V_2\}, \{V_1\}, \{V_3\}) \) and \( \neg I(\{V_2\}, \emptyset, \{V_3\}) \):

```
V_2  V_3
  ^   ^
  |   |
V_1
```
The d-separation criterion: introduction

**Definition:** Let $G = (V_G, A_G)$ be an acyclic directed graph (DAG), and let $s$ be a chain in $G$ between $V_i$ and $V_j \in V_G$. Chain $s$ is blocked (or: in-active) by a set $Z \subseteq V_G$ if $s$ contains a node $W$ for which one of the following holds:

- $W \in Z$ and $W$ has at most one incoming arc on chain $s$:
  \[
  V_i/V_j = \overset{W}{\bullet} \rightarrow \overset{\cdots}{\bullet} \rightarrow \overset{W}{\bullet} = V_i/V_j
  \]
- \[
  \sigma^*(W) \cap Z = \emptyset \text{ and } W \text{ has two incoming arcs on chain } s:
  \]
  \[
  \overset{\cdots}{\bullet} \rightarrow \overset{W}{\bullet} \rightarrow \overset{\cdots}{\bullet}
  \]
An example

Consider the following DAG and some of its chains:

1) $V_4, V_2, V_5$ from $V_4$ to $V_5$
2) $V_1, V_2, V_5, V_6, V_7$ from $V_1$ to $V_7$
3) $V_3, V_4, V_6, V_5$ from $V_3$ to $V_5$
4) $V_2, V_4$ from $V_2$ to $V_4$

Which of these chains is blocked by which of the following sets?

\{V_2\}, \{V_5\}, \{V_2, V_5\}, \{V_4\}, \{V_6\}, \{V_4, V_6\}
The d-separation criterion

**Definition:**
Let $G = (V_G, A_G)$ be an acyclic directed graph. Let $X, Y, Z \subseteq V_G$ be sets of nodes in $G$.

The set $Z$ d-separates $X$ from $Y$ in $G$—notation: $\langle X \mid Z \mid Y \rangle^d_G$—if every simple chain in $G$ from a node in $X$ to a node in $Y$ is blocked by $Z$.

**Remarks:**
- The above notion is known as the d-separation criterion;
- $\langle X \mid \emptyset \mid Y \rangle^d_G$ indicates that all chains between $X$ and $Y$, if any, contain a head-to-head node;
- if $X$ and $Y$ are not d-separated by $Z$, we say that they are d-connected given $Z$. 
An example

Consider the following DAG and d-separation statements:

V₁ → V₂ → V₃
V₁ → V₄ → V₅
V₁ → V₃

a) \( \langle \{V₁\} \mid \{V₂, V₃\} \mid \{V₅\}\rangle^d_G \)
b) \( \langle \{V₁\} \mid \{V₄\} \mid \{V₅\}\rangle^d_G \)
c) \( \langle \{V₂\} \mid \{V₁\} \mid \{V₃\}\rangle^d_G \)
d) \( \langle \{V₂\} \mid \{V₁, V₅\} \mid \{V₃\}\rangle^d_G \)
e) \( \langle \{V₂\} \mid \emptyset \mid \{V₃\}\rangle^d_G \)
f) \( \langle \{V₁\} \mid \{V₃, V₄\} \mid \{V₂\}\rangle^d_G \)

Which d-separation statements are valid in the graph?
Bayes-Ball for determining d-separation

Determine if $\langle X \mid Z \mid Y \rangle^d_G$ by dropping bouncing balls at $X$ and following the 10 rules of Bayes-ball:

- $Z$ is shaded
- a chain is active until a ball travelling along it meets a stop
- any node visited by a Bayes ball cannot be in $Y$
Independence relations and directed graphs

Definition:
Let $I$ be an independence relation on a set of statistical variables $V$. Let $G = (V_G, A_G)$ be an acyclic directed graph with $V_G = V$.

- the graph $G$ is called a (directed) dependency map (D-map) for $I$ if for every $X, Y, Z \subseteq V$ we have that:
  
  $$I(X, Z, Y) \text{ then } \langle X \mid Z \mid Y \rangle^d_G;$$

- the graph $G$ is called a (directed) independency map (I-map) for $I$ if for every $X, Y, Z \subseteq V$ we have that:
  
  $$\langle X \mid Z \mid Y \rangle^d_G \text{ then } I(X, Z, Y);$$

- the graph $G$ is called a (directed) perfect map (P-map) for $I$ if $G$ is both a dependency map and an independency map for $I$. 
Directed D-maps: what do they tell?

Let $I$ be an independence relation and $G$ a DAG.

Consider a D-map for $I$, then

$V_1$ and $V_2$ neighbours $\implies V_1, V_2$ dependent

$\neg \langle \{V_1\} \mid Z \mid \{V_2\} \rangle_G \implies \neg I(\{V_1\}, Z, \{V_2\})$

$V_1$ and $V_2$ non-neighbours $\implies$ ??

$\langle \{V_1\} \mid Z \mid \{V_2\} \rangle_G \implies$ dependent

independent

conditionally dependent ($Z = \emptyset$)

conditionally independent ($Z \neq \emptyset$)

Note: statements hold for all $Z \subseteq V_G \setminus (\{V_1\} \cup \{V_2\})$!
An example

Consider the independence relation $I$ on $V = \{V_1, \ldots, V_4\}$ defined by

$I(\{V_1\}, \emptyset, \{V_2\})$ and $I(\{V_1, V_2\}, \{V_3\}, \{V_4\})$

Which of the following DAGs are D-maps for $I$?
Directed I-maps

Let $I$ be an independence relation and $G$ a DAG.

Consider an I-map for $I$, then

$V_1$ and $V_2$ non-neighbours $\implies V_1, V_2$ (cond.) independent, or
cond. dependent (= induced)

$\langle \{V_1\} \mid Z \mid \{V_2\} \rangle_G$

$I(\{V_1\}, Z, \{V_2\})$

$V_1$ and $V_2$ neighbours $\implies ??$

$\neg \langle \{V_1\} \mid Z \mid \{V_2\} \rangle_G$

dependent
independent
conditionally dependent
conditionally independent

Note: statements hold for all $Z \subseteq V_G \setminus (\{V_1\} \cup \{V_2\})$!
An example

Consider the independence relation $I$ on $V = \{V_1, \ldots, V_4\}$ defined by

$$I(\{V_1\}, \emptyset, \{V_2\}) \text{ and } I(\{V_1, V_2\}, \{V_3\}, \{V_4\})$$

Which of the following DAGs are $I$-maps for $I$?
An example

Consider the independence relation $I$ on $V = \{V_1, \ldots, V_4\}$ defined by

$$I(\{V_1\}, \emptyset, \{V_2\}) \text{ and } I(\{V_1, V_2\}, \{V_3\}, \{V_4\})$$

The following DAG is a perfect map for $I$:

Is this P-map for $I$ unique?
An example

Consider the independence relation $I$ on $\mathbf{V} = \{ V_1, \ldots, V_4 \}$ defined by

$$I(\{ V_1 \}, \{ V_2, V_3 \}, \{ V_4 \}) \quad \text{and} \quad I(\{ V_2 \}, \{ V_1, V_4 \}, \{ V_3 \})$$

The relation $I$ does not have a directed perfect map. Consider for example the following DAG $G$:

In graph $G$ we have that $\langle \{ V_1 \} \mid \{ V_2, V_3 \} \mid \{ V_4 \} \rangle^d_G$, but also that $\langle \{ V_2 \} \mid \{ V_1 \} \mid \{ V_3 \} \rangle^d_G$!
Independence relations and their graphical representation

(Graph-isomorph: independence relation with perfect map.)
An I-map or a D-map?

Reconsider the independence relation $I$ on $V = \{V_1, \ldots, V_4\}$ defined by

$$I(\{V_1\}, \{V_2, V_3\}, \{V_4\}) \text{ and } I(\{V_2\}, \{V_1, V_4\}, \{V_3\})$$

Compare the following two representations of independence relation $I$:

- A D-map
- An I-map
Recall what we were looking for . . .

- Compact representation of independence relation of $Pr$;
- Factorise joint more efficiently than with chain rule $\rightarrow$ store (conditional) distributions involving less variables:

$$Pr(V) = Pr(V_n | V_{n-1} \land \ldots \land V_1) \cdot \ldots \cdot Pr(V_2 | V_1) \cdot Pr(V_1)$$

(chain rule)

$$= \ldots$$

$$= \ldots$$

$$= Pr(V_n) \cdot \ldots \cdot Pr(V_2) \cdot Pr(V_1)$$

(assuming mutual independence among all $V_i$)

- $Pr(X \land Y) = Pr(X) \cdot Pr(Y)$ is mathematically correct only if $X$ is truly independent of $Y$.
A minimal I-map

**Definition:** Let $I$ be an independence relation on a set of statistical variables $V$. Let $G = (V_G, A_G)$ be a graph with $V_G = V$.

The graph $G$ is called a **minimal I-map** for $I$ if the following conditions hold:

- $G$ is an I-map for $I$, and
- no proper subgraph of $G$ is an I-map for $I$. 
An example

Consider the independence relation \( I \) on \( V = \{V_1, \ldots, V_4\} \) defined by

\[
I(\{V_1\}, \{V_2, V_3\}, \{V_4\}) \quad \text{and} \quad I(\{V_2\}, \{V_1, V_4\}, \{V_3\})
\]

The following DAG is a minimal \( I \)-map for \( I \):

![Diagram of a minimal I-map](attachment:image.png)

Is this minimal \( I \)-map for \( I \) unique?
Directed or undirected ? (I)

Directed and undirected I-maps are related:

**Definition:** The moral graph of a DAG \( G = (V_G, A_G) \) is the undirected graph obtained as follows:

- for each \( V_k \in V_G \) add an edge between each pair of vertices \( V_i, V_j \in \rho_G(V_k) \), with \((V_i, V_j), (V_j, V_i) \notin A_G\);
- drop the directions of all arcs.

**Proposition:** Let \( I \) be an independence relation over \( V \). Let \( G = (V_G, A_G) \) be a DAG and \( G' = (V, E_{G'}) \) its moral graph. Then,

\[
G \text{ is an I-map for } I \quad \Rightarrow \quad G' \text{ is an I-map for } I
\]

\[
& \Leftrightarrow \quad & G' \text{ is chordal (triangulated)}
\]
Directed or undirected? (II)

Consider independence relation $I_{\text{Pr}}$ over $V$ and graph $G$ with $V = V_G$. Consider the following properties (partly proven later):

- Let $G$ be a DAG. Then $G$ is a minimal directed I-map of $I_{\text{Pr}}$ if and only if $\text{Pr}$ factorises as
  \[
  \text{Pr}(V) = \prod_{V_i} \text{Pr}(V_i \mid \rho_G(V_i))
  \]

- Let $G$ be an undirected graph. Then $G$ is an undirected I-map of $I_{\text{Pr}}$ if and only if $\text{Pr}$ can be written as
  \[
  \text{Pr}(V) = K \cdot \prod_{C_i} \Phi(C_i)
  \]

for some normalisation factor $K$.

What's the meaning of these clique potentials?!?
Chapter 4:
The Bayesian Network Framework
The network formalism, informal

A Bayesian network combines two types of domain knowledge to represent a joint probability distribution:

- **qualitative knowledge**: a minimal directed I-map for the independence relation that exists on the variables of the domain;
- **quantitative knowledge**: a set of local conditional probability distributions.
A Bayesian network

**Definition:**
A Bayesian network is a pair $\mathcal{B} = (G, \Gamma)$ such that

- $G = (V_G, A_G)$ is a DAG with arcs $A_G$ and nodes $V_G = V$, representing a set of statistical variables $V = \{V_1, \ldots V_n\}$, $n \geq 1$;
- $\Gamma = \{\gamma_{V_i} \mid V_i \in V\}$ is a set of non-negative functions
  $$\gamma_{V_i} : \{c_{V_i}\} \times \{c_{\rho(V_i)}\} \to [0, 1]$$
  such that for each configuration $c_{\rho(V_i)}$ of the set $\rho(V_i)$ of parents of $V_i$ in $G$, we have that
  $$\sum_{c_{V_i}} \gamma_{V_i}(c_{V_i} \mid c_{\rho(V_i)}) = 1$$
  for $i = 1, \ldots n$; these functions are called the assessment functions for $G$. 
An Example

Consider the following piece of ‘medical knowledge’:

“A metastatic carcinoma can cause a brain tumour and is also a possible explanation for an increased concentration of calcium in the blood. Both a brain tumour and an increased calcium concentration can result in a patient falling into a coma. A brain tumour can cause severe headaches.”

The independencies between the variables are represented in the following DAG $G$: 

![DAG Diagram]

- Carcinoma
- Brain tumour
- Calcium concentr.
- Headache
- Coma
An example – continued

Reconsider the following DAG $G$, and assume each $V \in V$ to be binary-valued.

With $G$ we associate a set of assessment functions
\[ \Gamma = \{ \gamma_{Car}, \gamma_B, \gamma_{Cal}, \gamma_H, \gamma_Co \}. \]

For the function $\gamma_{Car}$ the following function values are specified:
\[ \gamma_{Car}(\text{carc}) = 0.2, \quad \gamma_{Car}(\neg \text{carc}) = 0.8 \]

For the function $\gamma_B$ the following function values are specified:
\[ \gamma_B(\text{tum} \mid \text{carc}) = 0.2, \quad \gamma_B(\text{tum} \mid \neg \text{carc}) = 0.05 \]
\[ \gamma_B(\neg \text{tum} \mid \text{carc}) = 0.8, \quad \gamma_B(\neg \text{tum} \mid \neg \text{carc}) = 0.95 \]
An example – continued

Reconsider the following DAG $G$, and assume each $V \in V$ to be binary-valued.

With $G$ we associate a set of assessment functions
$\Gamma = \{\gamma_{\text{Car}}, \gamma_{\text{B}}, \gamma_{\text{Cal}}, \gamma_{\text{H}}, \gamma_{\text{Co}}\}$.

For the function $\gamma_{\text{Co}}$ the following function values are specified:

$\gamma_{\text{Co}}(\text{co} \mid \text{tum} \land \text{cal conc}) = 0.9 \quad \gamma_{\text{Co}}(\text{co} \mid \lnot \text{tum} \land \text{cal conc}) = 0.8$

$\gamma_{\text{Co}}(\text{co} \mid \text{tum} \land \lnot \text{cal conc}) = 0.7 \quad \gamma_{\text{Co}}(\text{co} \mid \lnot \text{tum} \land \lnot \text{cal conc}) = 0.05$

$\gamma_{\text{Co}}(\lnot \text{co} \mid \text{tum} \land \text{cal conc}) = 0.1 \quad \gamma_{\text{Co}}(\lnot \text{co} \mid \lnot \text{tum} \land \text{cal conc}) = 0.2$

$\gamma_{\text{Co}}(\lnot \text{co} \mid \text{tum} \land \lnot \text{cal conc}) = 0.3 \gamma_{\text{Co}}(\lnot \text{co} \mid \lnot \text{tum} \land \lnot \text{cal conc}) = 0.95$

The pair $\mathcal{B} = (G, \Gamma)$ is a Bayesian network.
A probabilistic interpretation

Proposition:

Let \( B = (G, \Gamma) \) be a Bayesian network with \( G = (V_G, A_G) \) and nodes \( V_G = V \), representing a set of statistical variables \( V = \{V_1, \ldots V_n\}, n \geq 1 \). Then

\[
\Pr(V) = \prod_{i=1}^{n} \gamma_{V_i}(V_i | \rho(V_i))
\]

defines a joint probability distribution \( \Pr \) on \( V \) such that \( G \) is a directed I-map for the independence relation \( I_{\Pr} \) of \( \Pr \).

\( \Pr \) is called the joint distribution defined by \( B \) and is said to respect the independences portrayed in \( G \).

NB we will often omit the subscript in \( \gamma \) if no confusion is possible.
An example

Consider the Bayesian network $\mathcal{B}$:

Let $\Pr$ be the joint distribution defined by $\mathcal{B}$. Then, for example

$$\Pr(v_1 \land v_2 \land v_3 \land v_4 \land v_5) =$$

$$= \gamma(v_5 \mid v_2) \cdot \gamma(v_4 \mid v_2 \land v_3) \cdot \gamma(v_3 \mid v_1) \cdot \gamma(v_2 \mid v_1) \cdot \gamma(v_1) =$$

$$= 0.4 \cdot 0.1 \cdot 0.2 \cdot 0.9 \cdot 0.8 = 0.00576$$

Note that $\Pr$ is described by only 11 probabilities; a naive representation of $\Pr$ would require 31 probabilities.
A probabilistic interpretation

**Proof:** (sketch)
Let $\mathcal{B} = (G, \Gamma)$ be a Bayesian network with $G = (V_G, A_G), V_G = V = \{V_1, \ldots, V_n\}, n \geq 1.$

The *acyclic* digraph $G$ allows a total ordering $\iota_G : V_G \leftrightarrow \{1, \ldots, n\}$ such that $\iota_G(V_i) < \iota_G(V_j)$ whenever there is a *directed path* from $V_i$ to $V_j$, $i \neq j$, in $G$.

**Example:**

![Diagram of a Bayesian network with nodes $V_1, V_2, V_3, V_4, V_5$ and directed edges illustrating the total ordering $\iota_G$.]
A probabilistic interpretation: proof continued

Take ordering $\nu_G$ as an ordering on the statistical variables $V_1, \ldots V_n$ as well.

Let $P$ be an arbitrary joint distribution on $V$ such that $G$ is a directed I-map for the independences in $P$.

Now apply the chain rule using $\nu_G$.

Example:

$$P(V_1 \land \ldots \land V_5) =$$

$$P(V_5 \mid V_1 \land \ldots \land V_4) \cdot P(V_4 \mid V_1 \land V_2 \land V_3) \cdot$$

$$P(V_3 \mid V_1 \land V_2) \cdot P(V_2 \mid V_1) \cdot P(V_1)$$
**Example:**

\[
P(V_1 \land \ldots \land V_5) = P(V_5 \mid V_1 \land \ldots \land V_4) \cdot P(V_4 \mid V_1 \land V_2 \land V_3) \cdot P(V_3 \mid V_1 \land V_2) \cdot P(V_2 \mid V_1) \cdot P(V_1)
\]

Each \( V_j \) is conditioned on just those \( V_i \) with \( \iota_G(V_i) < \iota_G(V_j) \).

Use the fact that \( G \) is an I-map for \( P \).

**Example:**

\[
P(V_1 \land \ldots \land V_5) = P(V_5 \mid V_2) \cdot P(V_4 \mid V_2 \land V_3) \cdot P(V_3 \mid V_1) \cdot P(V_2 \mid V_1) \cdot P(V_1)
\]

We have that

\[
P(V_1 \land \ldots \land V_n) = \prod_{V_i \in V} P(V_i \mid \rho(V_i))
\]
A probabilistic interpretation: proof continued

With graph $G$ is associated a set $\Gamma$ of assessment functions $\gamma(V_i \mid \rho(V_i))$. If we choose $\Pr(V_i \mid \rho(V_i)) = \gamma(V_i \mid \rho(V_i))$, then

$$\Pr(V_1 \land \ldots \land V_n) = \prod_{V_i \in V} \gamma(V_i \mid \rho(V_i))$$

defines a unique joint distribution on $V$ that respects the independences in $G$.

**Example:** The joint distribution $\Pr$ defined by

$$\Pr(V_1 \land \ldots \land V_5) = \gamma(V_5 \mid V_2) \cdot \gamma(V_4 \mid V_2 \land V_3) \cdot \gamma(V_3 \mid V_1) \cdot \gamma(V_2 \mid V_1) \cdot \gamma(V_1)$$

respects the independences in $G$.  

■
Consequences of probabilistic interpretation

Bayesian network $\mathcal{B}$ defines a joint distribution $\Pr(V)$ which respects the independences — read from graph $G$ by means of the d-separation criterion — stated in independence relation $I_{\Pr}$.

- $\mathcal{B}$ is a very compact representation of $\Pr$;
- any prior $\Pr(c_W)$ for $W \subseteq V$ can be computed from $\Pr$;
- same for any posterior $\Pr(c_W \mid c_E)$ for $W, E \subseteq V$;
- blocking sets $Z$ for d-separation now have an intuitive meaning: if we have evidence / observations for variables $E \subseteq V$ then we typically investigate blocking set $Z = E$. 
An example

Consider Bayesian network $B$, defining joint distribution $Pr$:

How can we compute $Pr(v_1 \land v_3 \land v_4 \land v_5)$?

$Pr(v_1 \land v_2 \land v_3 \land v_4 \land v_5) = 0.00576$

$Pr(v_1 \land \neg v_2 \land v_3 \land v_4 \land v_5) = 0.0016$

$Pr(v_1 \land v_3 \land v_4 \land v_5) =$

$= Pr(v_1 \land v_2 \land v_3 \land v_4 \land v_5) + Pr(v_1 \land \neg v_2 \land v_3 \land v_4 \land v_5)$

$= 0.00576 + 0.0016 = 0.00736$
Exact inference algorithms

- efficiently compute probabilities of interest from a network;
- efficiently process evidence.

The best-known algorithms, which serve to compute *marginals* over $V_i \in V$ (i.e. $\Pr(V_i)$ or $\Pr(V_i | c_E)$), are:


The algorithms are quite different in terms of the underlying ideas and their complexity.
Variable elimination: idea and complexity

Consider the computation of \( \Pr(d \mid e) = \frac{1}{\Pr(e)} \cdot \Pr(d \land e) \)

\[
\alpha \cdot \sum_{c_{ABC}} \Pr(c_A) \cdot \Pr(c_B \mid c_C) \cdot \Pr(c_C \mid c_A \land e) \cdot \Pr(d \mid c_C) \cdot \Pr(e)
\]

- summations can be moved into the factorisation
- only multiply factors when variables are to be summed out
- efficiency depends on order of variable elimination

\[
\alpha \cdot \Pr(e) \cdot \sum_{c_A} \Pr(c_A) \cdot \sum_{c_C} \Pr(c_C \mid c_A \land e) \cdot \Pr(d \mid c_C) \cdot \sum_{c_B} \Pr(c_B \mid c_C)
\]

Complexity for individual \( \Pr(V_i \mid c_E) \):

- singly connected graphs: linear in # of local probabilities;
- multiply connected graphs: exponential in # of nodes, even for bounded number of parents.
Join-tree propagation: idea and complexity

Idea of Join-tree propagation (L&S):

- moralise $G$, *triangulate* $G$, organise cliques into a *join tree*
- translate $\Gamma$ into clique potentials
- update clique potentials by message passing between cliques

Complexity for all $\Pr(V_i \mid c_E)$ simultaneously:

- linear in # of nodes, exponential in clique size (*tree-width*)
Pearl’s computational architecture

In Pearl's algorithm the graph of a Bayesian network is used as a computational architecture:

- each node in the graph is an autonomous object;
- each object has a local memory that stores the assessment functions of the associated node;
- each object has available a local processor that can do (simple) probabilistic computations;
- each arc in the graph is a (bi-directional) communication channel, through which connected objects can send each other messages.
A computational architecture

count count count
123 123

count count count
A computational architecture
A computational architecture
A computational architecture
A computational architecture
Consider Bayesian network $\mathcal{B}$ with the following graph:

Let $\Pr$ be the joint distribution defined by $\mathcal{B}$. We consider the situation without evidence.

- Can node $V_1$ compute the probabilities $\Pr(V_1)$? If so, how?
- Can node $V_2$ compute the probabilities $\Pr(V_2)$? If so, how?
Consider Bayesian network $\mathcal{B}$ with the following graph:

Let $\Pr$ be the joint distribution defined by $\mathcal{B}$.

We consider the situation without evidence.

- node $V_1$ can determine the probabilities for its own values:
  \[
  \Pr(v_1) = \gamma(v_1), \quad \Pr(\neg v_1) = \gamma(\neg v_1)
  \]

- node $V_2$ cannot determine $\Pr(V_2)$, but does know all four conditional probabilities: $\Pr(V_2 \mid V_1) = \gamma(V_2 \mid V_1)$

$V_2$ can compute its probabilities given information from $V_1$:

\[
\begin{align*}
\Pr(v_2) &= \Pr(v_2 \mid v_1) \cdot \Pr(v_1) + \Pr(v_2 \mid \neg v_1) \cdot \Pr(\neg v_1) \\
\Pr(\neg v_2) &= \Pr(\neg v_2 \mid v_1) \cdot \Pr(v_1) + \Pr(\neg v_2 \mid \neg v_1) \cdot \Pr(\neg v_1)
\end{align*}
\]
Consider Bayesian network $B$ with the following graph:

We consider the situation without evidence.

- Can node $V_1$ compute the probabilities $\Pr(V_1)$?
- Can node $V_2$ compute the probabilities $\Pr(V_2)$?
- Can node $V_3$ compute the probabilities $\Pr(V_3)$? If so, how?
Consider Bayesian network $\mathcal{B}$ with the following graph:

We consider the situation without evidence.

Given information from $V_1$, node $V_2$ can compute $\Pr(v_2)$ and $\Pr(\neg v_2)$.

Node $V_2$ now sends node $V_3$ the required information; node $V_3$ computes:

\[
\Pr(v_3) = \Pr(v_3 | v_2) \cdot \Pr(v_2) + \Pr(v_3 | \neg v_2) \cdot \Pr(\neg v_2)
\]

\[
= \gamma_{v_3}(v_3 | v_2) \cdot \Pr(v_2) + \gamma_{v_3}(v_3 | \neg v_2) \cdot \Pr(\neg v_2)
\]

\[
\Pr(\neg v_3) = \gamma_{v_3}(\neg v_3 | v_2) \cdot \Pr(v_2) + \gamma_{v_3}(\neg v_3 | \neg v_2) \cdot \Pr(\neg v_2)
\]
Reconsider Bayesian network $B$ without observations:

Node $V_1$ sends a message enabling $V_2$ to compute the probabilities for its values.

This message is a function $\pi_{V_1 \to V_2} : \{v_1, \neg v_1\} \to [0, 1]$ that attaches a number to each value of $V_1$, such that

$$\sum_{c_{V_1}} \pi_{V_1 \to V_2}(c_{V_1}) = 1$$

The function $\pi_{V_1 \to V_2}$ is called the causal parameter from $V_1$ to $V_2$. 
Causal parameters: an example

Consider the following Bayesian network without observations:

Node $V_1$:
- receives no messages
- computes and sends to $V_2$: causal parameter $\pi_{V_1}^{V_2}$

Node $V_1$ computes $\Pr(V_1)$:

\[
\Pr(v_1) = \pi_{V_1}^{V_2}(v_1) = 0.7; \quad \Pr(\neg v_1) = 0.3
\]
Causal parameters: an example (cntd)

Node $V_2$:
- receives causal parameter $\pi_{V_1}^{V_2}$ from $V_1$
- computes and sends to $V_3$: causal parameter $\pi_{V_3}^{V_2}$

with

\[ \pi_{V_3}^{V_2}(v_2) = \Pr(v_2 | v_1) \cdot \Pr(v_1) + \Pr(v_2 | \neg v_1) \cdot \Pr(\neg v_1) \]
\[ = \gamma(v_2 | v_1) \cdot \pi_{V_2}^{V_3}(v_1) + \gamma(v_2 | \neg v_1) \cdot \pi_{V_2}^{V_3}(\neg v_1) \]
\[ = 0.2 \cdot 0.7 + 0.5 \cdot 0.3 = 0.29 \]
\[ \pi_{V_3}^{V_2}(\neg v_2) = 0.8 \cdot 0.7 + 0.5 \cdot 0.3 = 0.71 \]

Node $V_2$ computes $\Pr(V_2)$:

\[ \Pr(v_2) = \pi_{V_3}^{V_2}(v_2) = 0.29; \quad \Pr(\neg v_2) = 0.71 \]
Causal parameters: an example (cntd)

Node $V_3$:
- receives causal parameter $\pi_{V_2}^{V_3}$ from $V_2$
- sends no messages

Node $V_3$ computes $\Pr(V_3)$:

$$\Pr(v_3) = \gamma(v_3 | v_2) \cdot \pi_{V_3}^{V_2}(v_2) + \gamma(v_3 | \neg v_2) \cdot \pi_{V_3}^{V_2}(\neg v_2)$$

$$= 0.6 \cdot 0.29 + 0.1 \cdot 0.71 = 0.245$$

$$\Pr(\neg v_3) = 0.4 \cdot 0.29 + 0.9 \cdot 0.71 = 0.755$$
Consider the Bayesian networks $B$ with the following graphs:

$V_1$  $\gamma(v_1 | v_2), \gamma(\neg v_1 | v_2)$
$\gamma(v_1 | \neg v_2), \gamma(\neg v_1 | \neg v_2)$

$V_2$  $\gamma(v_2), \gamma(\neg v_2)$

$V_3$  $\gamma(v_3 | v_2), \gamma(\neg v_3 | v_2)$
$\gamma(v_3 | \neg v_2), \gamma(\neg v_3 | \neg v_2)$

$V_1$  $\gamma(v_1), \gamma(\neg v_1)$

$V_2$  $\gamma(v_2 | v_1 \wedge v_3), \gamma(v_2 | v_1 \wedge \neg v_3)$
$\gamma(v_2 | \neg v_1 \wedge v_3), \gamma(v_2 | \neg v_1 \wedge \neg v_3)$
$\ldots$

$V_3$  $\gamma(v_3), \gamma(\neg v_3)$

We consider the situation without observations. In each of the above networks, can nodes $V_1$, $V_2$, and $V_3$ compute the probabilities $\Pr(V_1)$, $\Pr(V_2)$, and $\Pr(V_3)$, respectively. And if so, how?
Consider Bayesian network $\mathcal{B}$ with evidence $V_1 = true (v_1)$ and the following graph:

Node $V_1$ updates its probabilities and causal parameter:

$$\pi^{V_1}_{V_2} (v_1) = Pr^{v_1}(v_1)$$
$$= Pr(v_1 \mid v_1) = 1$$
$$\pi^{V_1}_{V_2} (\neg v_1) = Pr^{v_1}(\neg v_1) = 0$$

Given the updated information from $V_1$, node $V_2$ updates the probabilities for its own values:

$$\Pr^{v_1}(v_2) = \gamma(v_2 \mid v_1) \cdot \pi^{V_1}_{V_2} (v_1) + \gamma(v_2 \mid \neg v_1) \cdot \pi^{V_1}_{V_2} (\neg v_1)$$
$$= \gamma(v_2 \mid v_1)$$

$$\Pr^{v_1}(\neg v_2) = \gamma(\neg v_2 \mid v_1) \cdot \pi^{V_1}_{V_2} (v_1) + \gamma(\neg v_2 \mid \neg v_1) \cdot \pi^{V_1}_{V_2} (\neg v_1)$$
$$= \gamma(\neg v_2 \mid v_1)$$

Note that the function $\gamma_{V_1}$ remains unchanged!
Understanding Pearl with evidence (2a)

Consider Bayesian network $\mathcal{B}$ with the following graph:

Suppose we have evidence $V_2 = \text{true}$ for node $V_2$.

- Can node $V_1$ compute the probabilities $\Pr^{v_2}(V_1)$? If so, how?
- Can node $V_2$ compute the probabilities $\Pr^{v_2}(V_2)$? If so, how?
Consider Bayesian network $\mathcal{B}$ with evidence $V_2 = \text{true}$ and the following graph:

Node $V_1$ cannot update its probabilities using its own knowledge; it requires information from $V_2$! What information does $V_1$ require?

Consider the following properties:

\[
Pr^{v_2}(v_1) = \frac{Pr(v_2 | v_1) \cdot Pr(v_1)}{Pr(v_2)} \propto Pr(v_2 | v_1) \cdot Pr(v_1)
\]

\[
Pr^{v_2}(-v_1) = \frac{Pr(v_2 | -v_1) \cdot Pr(-v_1)}{Pr(v_2)} \propto Pr(v_2 | -v_1) \cdot Pr(-v_1)
\]
Introduction to diagnostic parameters

Reconsider Bayesian network $\mathcal{B}$:

Node $V_2$ sends a message enabling $V_1$ to update the probabilities for its values.

This message is a function $\lambda_{V_1}^{V_2}$: $\{v_1, \neg v_1\} \rightarrow [0, 1]$ that attaches a number to each value of $V_1$.

The message basically tells $V_1$ what node $V_2$ knows about $V_1$; in general:

$$\sum_{c_{V_1}} \lambda_{V_2}^{V_1}(c_{V_1}) \neq 1$$

The function $\lambda_{V_1}^{V_2}$ is called the diagnostic parameter from $V_2$ to $V_1$. 
Diagnostic parameters: an example

Consider the following Bayesian network $\mathcal{B}$ with evidence $V_2 = true$:

\[
\begin{align*}
V_1 & \quad \gamma_{V_1}(v_1) = 0.8, \quad \gamma_{V_1}(\neg v_1) = 0.2 \\
\lambda_{V_1}^{V_2} & \\
V_2 & \quad \gamma_{V_2}(v_2 \mid v_1) = 0.4, \quad \gamma_{V_2}(\neg v_2 \mid v_1) = 0.6 \\
& \quad \gamma_{V_2}(v_2 \mid \neg v_1) = 0.9, \quad \gamma_{V_2}(\neg v_2 \mid \neg v_1) = 0.1
\end{align*}
\]

Node $V_2$:

- computes and sends to $V_1$: diagnostic parameter $\lambda_{V_1}^{V_2}$ with

\[
\begin{align*}
\lambda_{V_2}^{V_1}(v_1) &= \Pr(v_2 \mid v_1) = \gamma(v_2 \mid v_1) = 0.4 \\
\lambda_{V_2}^{V_1}(\neg v_1) &= \gamma(v_2 \mid \neg v_1) = 0.9
\end{align*}
\]

Note that $\sum_{c_{V_1}} \lambda(c_{V_1}) = 1.3 > 1!$
Diagnostic parameters: an example (cntd)

Node $V_1$ receives from $V_2$ the diagnostic parameter $\lambda_{V_1}^{V_2}$

$V_1 \quad \gamma_{V_1}(v_1) = 0.8, \gamma_{V_1}(\neg v_1) = 0.2$

$V_2 \quad \gamma_{V_2}(v_2 | v_1) = 0.4, \gamma_{V_2}(\neg v_2 | v_1) = 0.6$
$\gamma_{V_2}(v_2 | \neg v_1) = 0.9, \gamma_{V_2}(\neg v_2 | \neg v_1) = 0.1$

Node $V_1$ computes:

$\Pr^{v_2}(v_1) = \alpha \cdot \Pr(v_2 | v_1) \cdot \Pr(v_1) = \alpha \cdot \lambda_{V_1}^{V_2}(v_1) \cdot \gamma(v_1) = \alpha \cdot 0.4 \cdot 0.8 = \alpha \cdot 0.32$

$\Pr^{v_2}(\neg v_1) = \alpha \cdot \lambda_{V_1}^{V_2}(\neg v_1) \cdot \gamma(\neg v_1) = \alpha \cdot 0.9 \cdot 0.2 = \alpha \cdot 0.18$

Node $V_1$ now normalises its probabilities using

$\Pr^{v_2}(v_1) + \Pr^{v_2}(\neg v_1) = 1: \alpha \cdot 0.32 + \alpha \cdot 0.18 = 1 \implies \alpha = 2$

resulting in

$\Pr^{v_2}(v_1) = 0.64 \quad \Pr^{v_2}(\neg v_1) = 0.36$
Understanding Pearl: directed path with evidence

Consider Bayesian network $\mathcal{B}$ with the following graph:

Suppose we have evidence $V_3 = true$ for node $V_3$.

- Can node $V_1$ compute the probabilities $\Pr^{v_3}(V_1)$?
- Can node $V_2$ compute the probabilities $\Pr^{v_3}(V_2)$? If so, how?
- Can node $V_3$ compute the probabilities $\Pr^{v_3}(V_3)$?

What if node $V_1$, node $V_2$, or both have evidence instead?
Pearl on directed paths – An example (1)

Consider Bayesian network $\mathcal{B}$ with evidence $V_3 = \text{true}$ and the following graph:

Node $V_1$:
- receives diagnostic parameter $\lambda^{V_1}_{V_2}(V_1)$
- computes and sends to $V_2$: causal parameter $\pi^{V_1}_{V_2}(V_1) = \gamma(V_1)$

Node $V_1$ computes

$$
\Pr^{v_3}(v_1) = \alpha \cdot \Pr(v_3 | v_1) \cdot \Pr(v_1) = \alpha \cdot \lambda^{V_1}_{V_2}(v_1) \cdot \gamma(v_1)
$$

$$
\Pr^{v_3}(\neg v_1) = \alpha \cdot \Pr(v_3 | \neg v_1) \cdot \Pr(\neg v_1) = \alpha \cdot \lambda^{V_1}_{V_2}(\neg v_1) \cdot \gamma(\neg v_1)
$$
Pearl on directed paths – An example (2)

Node $V_2$:
- receives causal parameter $\pi_{V_2}(V_1)$
- receives diagnostic parameter $\lambda_{V_3}(V_2)$
- computes and sends to $V_3$: $\pi_{V_3}(V_2)$

Node $V_2$ computes and sends to $V_1$: diagnostic parameter $\lambda_{V_2}(V_1)$ with

$$\lambda_{V_2}(v_1) = \Pr(v_3 | v_1)$$
$$= \Pr(v_3 | v_2) \cdot \Pr(v_2 | v_1) + \Pr(v_3 | \neg v_2) \cdot \Pr(\neg v_2 | v_1)$$
$$= \lambda_{V_3}(v_2) \cdot \gamma(v_2 | v_1) + \lambda_{V_3}(\neg v_2) \cdot \gamma(\neg v_2 | v_1)$$

$$\lambda_{V_2}(\neg v_1) = \Pr(v_3 | \neg v_1) = \ldots$$

The node then computes $\Pr^{v_3}(V_2)$...
Pearl on directed paths – An example (3)

Node $V_3$:

- receives causal parameter $\pi_{V_3}^{V_2}(V_2)$
- computes and sends to $V_2$: diagnostic parameter $\lambda_{V_3}^{V_2}(V_2)$ with

\[
\lambda_{V_3}^{V_2}(v_2) = \Pr(v_3 \mid v_2) = \gamma(v_3 \mid v_2)
\]

\[
\lambda_{V_3}^{V_2}(\neg v_2) = \Pr(v_3 \mid \neg v_2) = \gamma(v_3 \mid \neg v_2)
\]

- computes $\Pr_v^{V_3}(V_3)$
Consider the Bayesian networks $B$ with the following graphs:

Suppose we have evidence $V_3 = \text{true}$ for $V_3$.

Answer the following questions for each network above:

Can nodes $V_1$, $V_2$, and $V_3$ compute the probabilities $\Pr^{v_3}(V_1)$, $\Pr^{v_3}(V_2)$, and $\Pr^{v_3}(V_3)$, respectively. And if so, how?
The parameters as messages

Consider the graph of a Bayesian network as a computational architecture.

The separate causal and diagnostic parameters can be considered messages that are passed between objects through communication channels.
Pearl’s algorithm (high-level)

Let $\mathcal{B} = (G, \Gamma)$ be a Bayesian network with $G = (V_G, A_G)$; let $\Pr$ be the joint distribution defined by $\mathcal{B}$.

For each $V_i \in V_G$ do
- await messages from parents (if any) and compute $\pi(V_i)$
- await messages from children (if any) and compute $\lambda(V_i)$

compute and send messages $\pi_{V_i}^{V_i_j}(V_i)$ to all children $V_i_j$
compute and send messages $\lambda_{V_i}^{V_j_k}(V_j_k)$ to all parents $V_j_k$
compute $\Pr(V_i \mid c_E)$ for evidence $c_E$ (if any)

In the prior network message passing starts at ‘root’ nodes; upon processing evidence, message passing is initiated at observed nodes.
Notation: partial configurations

**Definition:**

A statistical variable \( V_j \in V \) is called instantiated if evidence \( V_j = true \) or \( V_j = false \) is obtained; otherwise \( V_j \) is called uninstantiated.

Let \( E \subseteq V \) be the subset of instantiated variables. The obtained configuration \( c_E \) is called a partial configuration of \( V \), written \( \tilde{c}_V \).

**Example:** Consider \( V = \{V_1, V_2, V_3\} \).

If no evidence is obtained (\( E = \emptyset \)) then: \( \tilde{c}_V = T(\text{true}) \)

If evidence \( V_2 = false \) is obtained, then: \( \tilde{c}_V = \neg v_2 \)

Note: with \( \tilde{c}_V \) we can refer to evidence without specifying \( E \).
Singly connected graphs (SCGs)

**Definition:** A directed graph $G$ is called singly connected if the underlying graph of $G$ is acyclic.

**Example:** The following graph is singly connected:

![Diagram of a singly connected graph]

**Lemma:** Let $G$ be a singly connected graph. Each graph that is obtained from $G$ by removing an arc, is not connected.

**Definition:** A (directed) tree is a singly connected graph where each node has at most one incoming arc.
Notation: lowergraphs and uppergraphs

**Definition:** Let $G = (V_G, A_G)$ be a singly connected graph and let $G_{(V_i,V_j)}$ be the subgraph of $G$ after removing the arc $(V_i, V_j) \in A_G$:

$$G_{(V_i,V_j)} = (V_G, A_G \setminus \{(V_i, V_j)\})$$

Now consider a node $V_i \in V_G$:

For each node $V_j \in \rho(V_i)$, let $G^+_{(V_j,V_i)}$ be the component of $G_{(V_j,V_i)}$ that contains $V_j$; $G^+_{(V_j,V_i)}$ is called an uppergraph of $V_i$.

For each node $V_k \in \sigma(V_i)$, let $G^-_{(V_i,V_k)}$ be the component of $G_{(V_i,V_k)}$ that contains $V_k$; $G^-_{(V_i,V_k)}$ is called a lowergraph of $V_i$. 
Node $V_0$ has:

- two uppergraphs $G^+(V_1, V_0)$ and $G^+(V_2, V_0)$
- two lowergraphs $G^-(V_0, V_3)$ and $G^-(V_0, V_4)$

For this graph we have, for example, that

$I( V_{G^+(V_1, V_0)}, \{V_0\}, V_{G^-(V_0, V_3)} )$

$I( V_{G^-(V_0, V_3)}, \{V_0\}, V_{G^-(V_0, V_4)} )$

$I( V_{G^+(V_1, V_0)}, \emptyset, V_{G^+(V_2, V_0)} )$
Lemma:

Let $\mathcal{B} = (G, \Gamma)$ be a Bayesian network with singly connected graph $G = (V_G, A_G)$ with $V_G = V = \{V_1, \ldots, V_n\}, n \geq 1$; let $\Pr$ be the joint distribution defined by $\mathcal{B}$.

For $V_i \in V$, let $V_i^+ = \bigcup_{V_j \in \rho(V_i)} V_{G(v_j, v_i)}^+$ and $V_i^- = V \setminus V_i^+$.

Then

$$\Pr(V_i \mid \tilde{c}_V) = \alpha \cdot \Pr(\tilde{c}_{V_i^-} \mid V_i) \cdot \Pr(V_i \mid \tilde{c}_{V_i^+})$$

where $\tilde{c}_V = \tilde{c}_{V_i^-} \land \tilde{c}_{V_i^+}$ and $\alpha$ is a normalisation constant.
Computing probabilities in singly connected graphs

Proof:

\[
\Pr(V_i \mid \tilde{c}_V) = \Pr(V_i \mid \tilde{c}_{V_i}^- \land \tilde{c}_{V_i}^+)
\]

\[
= \frac{\Pr(\tilde{c}_{V_i}^- \mid V_i) \cdot \Pr(\tilde{c}_{V_i}^+ \mid V_i) \cdot \Pr(V_i)}{\Pr(\tilde{c}_{V_i}^- \land \tilde{c}_{V_i}^+)}
\]

\[
= \Pr(\tilde{c}_{V_i}^- \mid V_i) \cdot \Pr(V_i \mid \tilde{c}_{V_i}^+) \cdot \frac{\Pr(\tilde{c}_{V_i}^+)}{\Pr(\tilde{c}_{V_i}^- \land \tilde{c}_{V_i}^+)}
\]

\[
= \alpha \cdot \Pr(\tilde{c}_{V_i}^- \mid V_i) \cdot \Pr(V_i \mid \tilde{c}_{V_i}^+)
\]

where \( \alpha = \frac{1}{\Pr(\tilde{c}_{V_i}^- \mid \tilde{c}_{V_i}^+)}. \)
Compound parameters: definition

**Definition:**

Let $\mathcal{B} = (G, \Gamma)$ be a Bayesian network with singly connected graph $G = (V_G, A_G)$; let $\Pr$ be the joint distribution defined by $\mathcal{B}$.

For $V_i \in V_G$, let $V_i^+$ and $V_i^-$ be as before;

- the function $\pi : \{v_i, \neg v_i\} \rightarrow [0, 1]$ for node $V_i$ is defined by
  \[
  \pi(V_i) = \Pr(V_i | \tilde{c}_{V_i^+})
  \]
  and is called the **compound causal parameter** for $V_i$;

- the function $\lambda : \{v_i, \neg v_i\} \rightarrow [0, 1]$ for node $V_i$ is defined by
  \[
  \lambda(V_i) = \Pr(\tilde{c}_{V_i^-} | V_i)
  \]
  and is called the **compound diagnostic parameter** for $V_i$.  

Computing probabilities in singly connected graphs

**Lemma:** (‘Data Fusion’)

Let \( B = (G, \Gamma) \) be a Bayesian network with singly connected graph \( G = (V_G, A_G) \); let \( \Pr \) be the joint distribution defined by \( B \). Then

\[
\text{for each } V_i \in V_G : \quad \Pr(V_i \mid \tilde{c}_{V_G}) = \alpha \cdot \pi(V_i) \cdot \lambda(V_i)
\]

with compound causal parameter \( \pi \), compound diagnostic parameter \( \lambda \), and normalisation constant \( \alpha \).

**Proof:**
Follows directly from the previous lemma and the definitions of the compound parameters. ■
The separate parameters defined

Definition:

Let $\mathcal{B} = (G, \Gamma)$ be a Bayesian network with singly connected graph $G = (V_G, A_G)$; let $\Pr$ be the joint distribution defined by $\mathcal{B}$.

Let $V_i \in V_G$ be a node with child $V_k \in \sigma(V_i)$ and parent $V_j \in \rho(V_i)$;

- the function $\pi_{V_k}^{V_i} : \{v_i, \neg v_i\} \rightarrow [0, 1]$ is defined by
  \[
  \pi_{V_k}^{V_i}(V_i) = \Pr(V_i | \tilde{c}_{G^+(V_i, V_k)}^{G(V_i, V_k)})
  \]
  and is called the causal parameter from $V_i$ to $V_k$.

- the function $\lambda_{V_i}^{V_j} : \{v_j, \neg v_j\} \rightarrow [0, 1]$ is defined by
  \[
  \lambda_{V_i}^{V_j}(V_j) = \Pr(\tilde{c}_{G(V_j, V_i)}^{G(V_j, V_i)} | V_j)
  \]
  and is called the diagnostic parameter from $V_i$ to $V_j$. 

\[ V(G_{(V_i, V_k)}^+) \]

\[ V(G_{(V_j, V_i)}^-) \]
Separate parameters in directed trees

\[ V^+_k \]

\[ V_i \]

\[ V_j \]

\[ V_i^- \]
Computing compound causal parameters in singly connected graphs

**Lemma:**

Let $\mathcal{B} = (G, \Gamma)$ be as before. Consider a node $V_i \in \mathcal{V}_G$ and its parents $\rho(V_i) = \{V_{i_1}, \ldots, V_{i_m}\}$, $m \geq 1$.

Then

$$\pi(V_i) = \sum_{c_{\rho(V_i)}} \gamma(V_i \mid c_{\rho(V_i)}) \cdot \prod_{j=1,\ldots,m} \pi_{V_i}^{V_{i_j}}(c_{V_{i_j}})$$

where $c_{\rho(V_i)} = \bigwedge_{j=1,\ldots,m} c_{V_{i_j}}$
\( V(G_+^{(V_1)}(V_c, V_i)) \)
Computing compound causal parameters in singly connected graphs

**Proof:**

Let \( \Pr \) be the joint distribution defined by \( \mathcal{B} \). Then

\[
\pi(V_i) \overset{\text{DEF}}{=} \Pr(V_i \mid \tilde{c}_{V_i^+}) = \Pr(V_i \mid \tilde{c}_{V}^{G^+(V_{i_1}, V_i)} \land \ldots \land \tilde{c}_{V}^{G^+(V_{i_m}, V_i)} )
\]

\[
= \sum_{c_{\rho}(V_i)} \Pr(V_i \mid c_{\rho}(V_i) \land \tilde{c}_{V}^{G^+(V_{i_1}, V_i)} \land \ldots \land \tilde{c}_{V}^{G^+(V_{i_m}, V_i)} ) \cdot \Pr(c_{\rho}(V_i) \mid \tilde{c}_{V}^{G^+(V_{i_1}, V_i)} \land \ldots \land \tilde{c}_{V}^{G^+(V_{i_m}, V_i)} )
\]

\[
= \sum_{c_{\rho}(V_i)} \Pr(V_i \mid c_{\rho}(V_i)) \cdot \prod_{j=1,\ldots,m} \Pr(c_{V_{ij}} \mid \tilde{c}_{V}^{G^+(V_{ij}, V_i)} )
\]

\[
= \sum_{c_{\rho}(V_i)} \gamma(V_i \mid c_{\rho}(V_i)) \cdot \prod_{j=1,\ldots,m} \pi_{V_{ij}}^{V_i} (c_{V_{ij}} )
\]

where \( c_{\rho}(V_i) = \bigwedge_{j=1,\ldots,m} c_{V_{ij}} \)
Computing $\pi$ in directed trees

**Lemma:**

Let $\mathcal{B} = (G, \Gamma)$ be a Bayesian network with directed tree $G$.

Consider a node $V_i \in V_G$ and its parent $\rho(V_i) = \{V_j\}$.

Then

$$\pi(V_i) = \sum_{c_{V_j}} \gamma(V_i \mid c_{V_j}) \cdot \pi_{V_j}^{V_i}(c_{V_j})$$

**Proof:**

See the proof for the general case where $G$ is a singly connected graph. Take into account that $V_i$ now only has a single parent $V_j$. ■
Computing causal parameters in singly connected graphs

Lemma:

Let $\mathcal{B} = (G, \Gamma)$ be a Bayesian network with singly connected graph $G = (V_G, A_G)$.

Consider an uninstantiated node $V_i \in V_G$ with $m \geq 1$ children $\sigma(V_i) = \{V_{i1}, \ldots, V_{im}\}$.

Then

$$\pi_{V_{ij}}^V_i (V_i) = \alpha \cdot \pi(V_i) \cdot \prod_{k=1,\ldots,m, k \neq j} \lambda_{V_{ik}}^V_i (V_i)$$

where $\alpha$ is a normalisation constant.
Computing causal parameters in singly connected graphs

**Proof:**

Let \( \Pr \) be the joint distribution defined by \( \beta \). Then

\[
\pi_{v_{ij}}(v_i) \overset{\text{DEF}}{=} \Pr(v_i \mid \tilde{c}_{v_{i_j}}^{G(v_i,v_{i_j})})
\]

\[
= \alpha' \cdot \Pr(\tilde{c}_{v_{i_j}}^{G(v_i,v_{i_j})} \mid v_i) \cdot \Pr(v_i)
\]

\[
= \alpha' \cdot \Pr(\tilde{c}_{v_i^+} \land (\bigwedge_{k \neq j} \tilde{c}_{v_{i_k}}^{G(v_i,v_{i_k})}) \mid v_i) \cdot \Pr(v_i)
\]

\[
= \alpha' \cdot \Pr(\tilde{c}_{v_i^+} \mid v_i) \cdot \prod_{k \neq j} \Pr(\tilde{c}_{v_{i_k}}^{G(v_i,v_{i_k})} \mid v_i) \cdot \Pr(v_i)
\]

\[
= \alpha \cdot \Pr(v_i \mid \tilde{c}_{v_i^+}) \cdot \prod_{k \neq j} \Pr(\tilde{c}_{v_{i_k}}^{G(v_i,v_{i_k})} \mid v_i)
\]

\[
= \alpha \cdot \pi(v_i) \cdot \prod_{k \neq j} \lambda_{v_{i_k}}^{v_i}(v_i)
\]
Computing compound diagnostic parameters in singly connected graphs

Lemma:

Let $\mathcal{B} = (G, \Gamma)$ be as before.

Consider an uninstantiated node $V_i \in V_G$ with $m \geq 1$ children $\sigma(V_i) = \{V_{i1}, \ldots, V_{im}\}$.

Then

$$\lambda(V_i) = \prod_{j=1,\ldots,m} \lambda_{V_{ij}}^V(V_i)$$
Computing compound diagnostic parameters in singly connected graphs

**Proof:** Let $Pr$ be the joint distribution defined by $B$. Then

$$\lambda(V_i) \overset{\text{DEF}}{=} Pr(\tilde{c}_{V_i}^- \mid V_i)$$

$$= Pr(\tilde{c}_{V_{G_{(V_i,V_{i_1}})}}^- \land \ldots \land \tilde{c}_{V_{G_{(V_i,V_{i_m}})}}^- \mid V_i)$$

$$= Pr(\tilde{c}_{V_{G_{(V_i,V_{i_1}})}}^- \mid V_i) \cdot \ldots \cdot Pr(\tilde{c}_{V_{G_{(V_i,V_{i_m}})}}^- \mid V_i)$$

$$= \lambda_{V_{i_1}}(V_i) \cdot \ldots \cdot \lambda_{V_{i_m}}(V_i)$$

$$= \prod_{j=1,\ldots,m} \lambda_{V_{i_j}}(V_i)$$
Lemma:

Let $B = (G, \Gamma)$ be as before. Consider a node $V_i \in V_G$ with $n \geq 1$ parents $\rho(V_i) = \{V_{j_1}, \ldots, V_{j_n}\}$. Then

$$\lambda_{V_i}^{V_{j_k}}(V_{j_k}) = \alpha \cdot \sum_{c_{V_i}} \lambda(c_{V_i}) \cdot \left[ \sum_{x=c_{\rho(V_i)}\setminus\{V_{j_k}\}} (\gamma(c_{V_i} | x \wedge V_{j_k}) \cdot \prod_{l=1,\ldots,n, l \neq k} \pi_{V_i}^{V_{j_l}}(c_{V_{j_l}})) \right]$$

where $\alpha$ is a normalisation constant.

Proof:

We will only illustrate the proof for the special case where graph $G$ takes the form of a directed tree (see next slide).
Computing separate $\lambda$'s in directed trees

Lemma:

Let $\mathcal{B} = (G, \Gamma)$ be a Bayesian network with directed tree $G$.

Consider a node $V_i \in V_G$ and its parent $\rho(V_i) = \{V_j\}$.

Then

$$\lambda_{V_i}^{V_j}(V_j) = \sum_{c_{V_i}} \lambda(c_{V_i}) \cdot \gamma(c_{V_i} | V_j)$$
Computing separate $\lambda$’s in directed trees

**Proof:** Let $\Pr$ be the joint distribution defined by $\mathcal{B}$. Then

$$\lambda_{V_i}^V_j (V_j) \overset{\text{DEF}}{=} \Pr(\tilde{c}_{V_i}^{-} \mid V_j)$$

$$= \Pr(\tilde{c}_{V_i}^{-} \mid v_i \land V_j) \cdot \Pr(v_i \mid V_j)$$

$$+ \Pr(\tilde{c}_{V_i}^{-} \mid \neg v_i \land V_j) \cdot \Pr(\neg v_i \mid V_j)$$

$$= \Pr(\tilde{c}_{V_i}^{-} \mid v_i) \cdot \Pr(v_i \mid V_j)$$

$$+ \Pr(\tilde{c}_{V_i}^{-} \mid \neg v_i) \cdot \Pr(\neg v_i \mid V_j)$$

$$= \lambda(v_i) \cdot \gamma(v_i \mid V_j) + \lambda(\neg v_i) \cdot \gamma(\neg v_i \mid V_j)$$

$$= \sum_{c_{V_i}} \lambda(c_{V_i}) \cdot \gamma(c_{V_i} \mid V_j)$$
Pearl’s algorithm: detailed computation rules for inference

For $V_i \in V_G$ with $\rho(V_i) = \{V_{j_1}, \ldots, V_{j_n}\}$, $\sigma(V_i) = \{V_{i_1}, \ldots, V_{i_m}\}$:

$$\Pr(V_i | \widetilde{c}_V) = \alpha \cdot \pi(V_i) \cdot \lambda(V_i)$$

$$\pi(V_i) = \sum_{c_{\rho(V_i)}} \gamma(V_i | c_{\rho(V_i)}) \cdot \prod_{k=1}^{n} \pi_{V_{j_k}}(c_{V_{j_k}})$$

$$\lambda(V_i) = \prod_{j=1}^{m} \lambda^{V_i}_{V_{i_j}}(V_i)$$

$$\pi^{V_i}_{V_{i_j}}(V_i) = \alpha' \cdot \pi(V_i) \cdot \prod_{k=1, k \neq j}^{m} \lambda^{V_i}_{V_{i_k}}(V_i)$$

$$\lambda^{V_{j_k}}_{V_i}(V_{j_k}) = \alpha'' \cdot \sum_{c_{V_i}} \lambda(c_{V_i}) \cdot \left[ \sum_{x=c_{\rho(V_i)} \setminus \{V_{j_k}\}} \gamma(c_{V_i} | x \land V_{j_k}) \cdot \prod_{l=1, l \neq k}^{n} \pi^{V_{j_l}}_{V_i}(c_{V_{j_l}}) \right]$$

with normalisation constants $\alpha, \alpha'$, and $\alpha''$. 
Special cases: roots

Let $\mathcal{B} = (G, \Gamma)$ be a Bayesian network with singly connected graph $G$; let $\Pr$ be the joint distribution defined by $\mathcal{B}$.

- Consider a node $W \in V_G$ with $\rho(W) = \emptyset$

  The compound causal parameter

  $\pi : \{w, \neg w\} \rightarrow [0, 1]$ for $W$ is defined by

  \[
  \pi(W) = \Pr(W \mid \tilde{c}_W^+) \quad \text{(definition)}
  \]

  \[
  = \Pr(W \mid \top) \quad (W^+ = \emptyset)
  \]

  \[
  = \Pr(W)
  \]

  \[
  = \gamma(W)
  \]
Special cases: leaves

Let $\mathcal{B} = (G, \Gamma)$ and $Pr$ be as before.

- Consider a node $V$ with $\sigma(V) = \emptyset$

  The compound diagnostic parameter $\lambda : \{v, \neg v\} \rightarrow [0, 1]$ for $V$ is defined as follows:

  - if node $V$ is uninstantiated, then
    \[
    \lambda(V) = \Pr(\tilde{c}_V^- | V) \quad \text{(definition)}
    = \Pr(T | V) \quad (V^- = \{V\}, V \text{ uninst.})
    = 1
    \]

  - if node $V$ is instantiated, then
    \[
    \lambda(V) = \Pr(\tilde{c}_V^- | V) \quad \text{(definition)}
    = \Pr(\tilde{c}_V | V) \quad (\sigma(V) = \emptyset)
    = \begin{cases} 
      1 & \text{for } c_V = \tilde{c}_V \\
      0 & \text{for } c_V \neq \tilde{c}_V
    \end{cases}
    \]
Special cases: uninstantiated (sub)graphs

“a useful property”

- Consider a node $V \in V_G$ and assume that $\widetilde{c}_{V_G} = \text{T(rue)}$.
  The compound diagnostic parameter

  $\lambda : \{v, \neg v\} \rightarrow [0, 1]$ for $V$ is defined as follows:

  \[
  \lambda(V) = \Pr(\widetilde{c}_V^- \mid V) \quad \text{(definition)} \\
  = \Pr(T \mid V) \quad (\widetilde{c}_{V_G} = \text{T}) \\
  = 1
  \]

  From the above it is clear that this property also holds for any node $V$ for which $\widetilde{c}_V^- = \text{T}$. 
Pearl’s algorithm: a tree example

Consider Bayesian network $\mathcal{B} = (G, \Gamma)$:

Let $Pr$ be the joint distribution defined by $\mathcal{B}$.

**Assignment:** compute $Pr(V_i)$, $i = 1, \ldots, 5$.

**Start:** $Pr(V_i) = \alpha \cdot \pi(V_i) \cdot \lambda(V_i)$, $i = 1, \ldots, 5$.

$\lambda(V_i) = 1$ for all $V_i$. **Why?** As a result, no normalisation is required and $Pr(V_i) = \pi(V_i)$. 
An example (2)

\[ \gamma(v_2 \mid v_1) = 0.5 \]
\[ \gamma(v_2 \mid \neg v_1) = 0.4 \]
\[ \gamma(v_3 \mid v_2) = 0.2 \]
\[ \gamma(v_3 \mid \neg v_2) = 0.3 \]
\[ \gamma(v_4 \mid v_2) = 0.8 \]
\[ \gamma(v_4 \mid \neg v_2) = 0 \]
\[ \gamma(v_5 \mid v_1) = 0.1 \]
\[ \gamma(v_5 \mid \neg v_1) = 0.8 \]

\[ \pi(V_1) = \gamma(V_1). \text{ Why? Node } V_1 \text{ computes:} \]
\[ \Pr(v_1) = \pi(v_1) = \gamma(v_1) = 0.7 \]
\[ \Pr(\neg v_1) = \pi(\neg v_1) = \gamma(\neg v_1) = 0.3 \]

Node \( V_1 \) computes for node \( V_2 \):

\[ \pi_{V_2}^V_1(V_1) = \pi(V_1) \]

(why?)
An example (3)

Node $V_2$ computes:

\[
\Pr(v_2) = \pi(v_2) = \gamma(v_2 \mid v_1) \cdot \pi_{V_1}(v_1) + \gamma(v_2 \mid \neg v_1) \cdot \pi_{V_2}(\neg v_1)
\]

\[
= 0.5 \cdot 0.7 + 0.4 \cdot 0.3 = 0.47
\]

\[
\Pr(\neg v_2) = \pi(\neg v_2) = 0.5 \cdot 0.7 + 0.6 \cdot 0.3 = 0.53
\]
An example (4)

\[ \gamma(v_2 \mid v_1) = 0.5 \]
\[ \gamma(v_2 \mid \neg v_1) = 0.4 \]
\[ \gamma(v_3 \mid v_2) = 0.2 \]
\[ \gamma(v_3 \mid \neg v_2) = 0.3 \]
\[ \gamma(v_4 \mid v_2) = 0.8 \]
\[ \gamma(v_4 \mid \neg v_2) = 0 \]
\[ \gamma(v_5 \mid v_1) = 0.1 \]
\[ \gamma(v_5 \mid \neg v_1) = 0.8 \]

Node \( V_2 \) computes for node \( V_3 \):

\[ \pi_{V_3}^{V_2}(V_2) = \pi(V_2) \]

Are all causal parameters sent by a node equal to its compound causal parameter?
An example (5)

Node $V_3$ computes:

\[
\Pr(v_3) = \pi(v_3) \\
= \gamma(v_3 | v_2) \cdot \pi_{V_3}^{V_2}(v_2) + \gamma(v_3 | -v_2) \cdot \pi_{V_3}^{V_2}(-v_2) \\
= 0.2 \cdot 0.47 + 0.3 \cdot 0.53 = 0.253
\]

\[
\Pr(-v_3) = \pi(-v_3) = 0.8 \cdot 0.47 + 0.7 \cdot 0.53 \\
= 0.747
\]
An example (6)

\[ \gamma(v_2 | v_1) = 0.5 \]
\[ \gamma(v_2 | \neg v_1) = 0.4 \]
\[ \gamma(v_3 | v_2) = 0.2 \]
\[ \gamma(v_3 | \neg v_2) = 0.3 \]
\[ \gamma(v_4 | v_2) = 0.8 \]
\[ \gamma(v_4 | \neg v_2) = 0 \]
\[ \gamma(v_5 | v_1) = 0.1 \]
\[ \gamma(v_5 | \neg v_1) = 0.8 \]

In a similar way, we find that

\[ \Pr(v_4) = 0.376, \quad \Pr(\neg v_4) = 0.624 \]
\[ \Pr(v_5) = 0.310, \quad \Pr(\neg v_5) = 0.690 \]
Pearl’s algorithm: a singly connected example

Consider Bayesian network $\mathcal{B} = (G, \Gamma)$:

\begin{align*}
\gamma(v_2) &= 0.1 \\
\gamma(\neg v_2) &= 0.9 \\
\gamma(v_1 \mid v_2 \land v_3) &= 0.8 \\
\gamma(v_1 \mid \neg v_2 \land v_3) &= 0.9 \\
\gamma(v_1 \mid v_2 \land \neg v_3) &= 0.5 \\
\gamma(v_1 \mid \neg v_2 \land \neg v_3) &= 0.6 \\
\gamma(v_3) &= 0.4 \\
\gamma(\neg v_3) &= 0.6 \\
\gamma(\neg v_1 \mid v_2 \land v_3) &= 0.2 \\
\gamma(\neg v_1 \mid \neg v_2 \land v_3) &= 0.1 \\
\gamma(\neg v_1 \mid v_2 \land \neg v_3) &= 0.5 \\
\gamma(\neg v_1 \mid \neg v_2 \land \neg v_3) &= 0.4
\end{align*}

Let $\Pr$ be the joint distribution defined by $\mathcal{B}$.

**Assignment:** compute $\Pr(V_1) = \alpha \cdot \pi(V_1) \cdot \lambda(V_1)$.

$\lambda(V_1) = 1$, so no normalisation is required.
An example (2)

\[ \gamma(v_2) = 0.1 \]
\[ \gamma(\neg v_2) = 0.9 \]
\[ \gamma(v_1 \mid v_2 \land v_3) = 0.8 \]
\[ \gamma(v_1 \mid \neg v_2 \land v_3) = 0.9 \]
\[ \gamma(v_1 \mid v_2 \land \neg v_3) = 0.5 \]
\[ \gamma(v_1 \mid \neg v_2 \land \neg v_3) = 0.6 \]

\[ \gamma(v_3) = 0.4 \]
\[ \gamma(\neg v_3) = 0.6 \]
\[ \gamma(v_1 \mid v_2 \land v_3) = 0.2 \]
\[ \gamma(v_1 \mid \neg v_2 \land v_3) = 0.1 \]
\[ \gamma(v_1 \mid v_2 \land \neg v_3) = 0.5 \]
\[ \gamma(v_1 \mid \neg v_2 \land \neg v_3) = 0.4 \]

Node \( V_1 \) computes:

\[
\Pr(v_1) = \pi(v_1) = \gamma(v_1 \mid v_2 \land v_3) \cdot \pi_{V_1}^{V_2}(v_2) \cdot \pi_{V_1}^{V_3}(v_3) + \\
+ \gamma(v_1 \mid \neg v_2 \land v_3) \cdot \pi_{V_1}^{V_2}(\neg v_2) \cdot \pi_{V_1}^{V_3}(v_3) + \\
+ \gamma(v_1 \mid v_2 \land \neg v_3) \cdot \pi_{V_1}^{V_2}(v_2) \cdot \pi_{V_1}^{V_3}(\neg v_3) + \\
+ \gamma(v_1 \mid \neg v_2 \land \neg v_3) \cdot \pi_{V_1}^{V_2}(\neg v_2) \cdot \pi_{V_1}^{V_3}(\neg v_3) \\
= 0.8 \cdot 0.1 \cdot 0.4 + 0.9 \cdot 0.9 \cdot 0.4 + \\
+ 0.5 \cdot 0.1 \cdot 0.6 + 0.6 \cdot 0.9 \cdot 0.6 = 0.71
\]

\[
\Pr(\neg v_1) = 0.29
\]
Instantiated nodes

Let $B = (G, \Gamma)$ be a Bayesian network with singly connected graph $G$; let $Pr$ be as before.

- Consider an instantiated node $V \in V_G$, for which evidence $V = true$ is obtained.

For the compound diagnostic parameter $\lambda : \{v, \neg v\} \rightarrow [0, 1]$ for $V$ we have that

\[
\lambda(v) = \Pr(\overline{c_v} | v) \quad \text{(definition)}
\]
\[
= \Pr(\overline{c_v} \land \neg \{V\} \land v | v)
\]
\[
= ??
\]

(unless $\sigma(V) = \emptyset$ in which case $\lambda(v) = 1$)

\[
\lambda(\neg v) = \Pr(\overline{c_v} | \neg v) \quad \text{(definition)}
\]
\[
= \Pr(\overline{c_v} \land \neg \{V\} \land v | \neg v)
\]
\[
= 0
\]

The case with evidence $V = false$ is similar.
Entering evidence

Consider the following fragment of graph $G$ (in black) of a Bayesian network:

Suppose evidence is obtained for node $V$.

Entering evidence is modelled by extending $G$ with a ‘dummy’ child $D$ for $V$.

The dummy node sends the diagnostic parameter $\lambda^V_D$ to $V$ with

\[
\begin{align*}
\lambda^V_D(v) &= 1, & \lambda^V_D(\neg v) &= 0 \quad & \text{for evidence } V = true \\
\lambda^V_D(v) &= 0, & \lambda^V_D(\neg v) &= 1 \quad & \text{for evidence } V = false
\end{align*}
\]
**Entering evidence: a tree example**

Let \( \Pr \) and \( \mathcal{B} \) be as before:

\[
\begin{align*}
\gamma(v_2 | v_1) &= 0.5 \\
\gamma(v_2 | \neg v_1) &= 0.4 \\
\gamma(v_3 | v_2) &= 0.2 \\
\gamma(v_3 | \neg v_2) &= 0.3 \\
\gamma(v_4 | v_2) &= 0.8 \\
\gamma(v_4 | \neg v_2) &= 0 \\
\gamma(v_5 | v_1) &= 0.1 \\
\gamma(v_5 | \neg v_1) &= 0.8 \\
\gamma(v_1) &= 0.7
\end{align*}
\]

Evidence \( V_1 = false \) is entered.

**Assignment:** compute \( \Pr^{\neg v_1}(V_i) \).

**Start:**

\[
\Pr^{\neg v_1}(V_i) = \alpha \cdot \pi(V_i) \cdot \lambda(V_i), \ i = 1, \ldots, 5.
\]

For \( i = 2, \ldots, 5 \), we have that \( \lambda(V_i) = 1 \). Why? For those nodes we thus have \( \Pr(V_i) = \pi(V_i) \).
An example with evidence $V_1 = false$ (2)

Node $V_1$ now computes:

$$\Pr^{-v_1}(v_1) = \alpha \cdot \pi(v_1) \cdot \lambda(v_1) = 0$$

$$\Pr^{-v_1}(\neg v_1) = \alpha \cdot \pi(\neg v_1) \cdot \lambda(\neg v_1) = \alpha \cdot 0.3$$

Normalisation gives: $\Pr^{-v_1}(v_1) = 0$, $\Pr^{-v_1}(\neg v_1) = 1$

Node $V_1$ computes for node $V_2$:

$$\pi_{V_2}(V_1) = \alpha \cdot \pi(V_1) \cdot \lambda_{V_5}^{V_1}(V_1) \cdot \lambda_D^{V_1}(V_1) = ?$$
An example with evidence $V_1 = false (3)$

$\gamma(v_1) = 0.7$

$\gamma(v_2 | v_1) = 0.5$
$\gamma(v_2 | \neg v_1) = 0.4$

$\gamma(v_3 | v_2) = 0.2$
$\gamma(v_3 | \neg v_2) = 0.3$

$\gamma(v_4 | v_2) = 0.8$
$\gamma(v_4 | \neg v_2) = 0$

$\gamma(v_5 | v_1) = 0.1$
$\gamma(v_5 | \neg v_1) = 0.8$

Node $V_2$ computes:

$$\Pr^{\neg v_1}(v_2) = \pi(v_2)$$
$$= \gamma(v_2 | v_1) \cdot \pi_{V_2}^V(v_1) + \gamma(v_2 | \neg v_1) \cdot \pi_{V_2}^V(\neg v_1)$$
$$= 0.5 \cdot 0 + 0.4 \cdot 1 = 0.4$$

$$\Pr^{\neg v_1}(\neg v_2) = \pi(\neg v_2) = 0.5 \cdot 0 + 0.6 \cdot 1 = 0.6$$

Node $V_2$ computes for node $V_3$: $\pi_{V_3}^V(V_2) = \pi(V_2)$
An example with evidence $V_1 = \text{false}$ (4)

\[ \gamma(v_2 | v_1) = 0.5 \]
\[ \gamma(v_2 | \neg v_1) = 0.4 \]
\[ \gamma(v_3 | v_2) = 0.2 \]
\[ \gamma(v_3 | \neg v_2) = 0.3 \]
\[ \gamma(v_4 | v_2) = 0.8 \]
\[ \gamma(v_4 | \neg v_2) = 0 \]
\[ \gamma(v_5 | v_1) = 0.1 \]
\[ \gamma(v_5 | \neg v_1) = 0.8 \]

Node $V_3$ computes:

\[
\Pr^{\neg v_1}(v_3) = \pi(v_3) = \gamma(v_3 | v_2) \cdot \pi_{V_3}^v(v_2) + \gamma(v_3 | \neg v_2) \cdot \pi_{V_3}^\neg(v_2)
\]
\[
= 0.2 \cdot 0.4 + 0.3 \cdot 0.6 = 0.26
\]
\[
\Pr^{\neg v_1}(\neg v_3) = 0.8 \cdot 0.4 + 0.7 \cdot 0.6 = 0.74
\]
An example with evidence $V_1 = false$ (5)

In a similar way, we find that

$$\Pr^{-v_1}(v_4) = 0.32, \quad \Pr^{-v_1}(-v_4) = 0.68$$

$$\Pr^{-v_1}(v_5) = 0.80, \quad \Pr^{-v_1}(-v_5) = 0.20$$
Another piece of evidence: tree example

Let \( P_r \) and \( B \) be as before:

\[
\begin{align*}
\gamma(v_2 \mid v_1) &= 0.5 \\
\gamma(v_2 \mid \neg v_1) &= 0.4 \\
\gamma(v_3 \mid v_2) &= 0.2 \\
\gamma(v_3 \mid \neg v_2) &= 0.3 \\
\gamma(v_4 \mid v_2) &= 0.8 \\
\gamma(v_4 \mid \neg v_2) &= 0 \\
\gamma(v_5 \mid v_1) &= 0.1 \\
\gamma(v_5 \mid \neg v_1) &= 0.8 \\
\gamma(v_1) &= 0.7
\end{align*}
\]

The additional evidence \( V_3 = \text{true} \) is entered.

**Assignment:** compute \( P_r^{-v_1,v_3}(V_i) \).

**Start:** \( P_r^{-v_1,v_3}(V_i) = \alpha \cdot \pi(V_i) \cdot \lambda(V_i), \ i = 1, \ldots, 5. \)

Which parameters can be re-used and which should be updated?
Another example (2)

For $i = 4, 5$, we have that $\lambda(V_i) = 1$. For those two nodes we thus have $\Pr(V_i) = \pi(V_i)$.

The probabilities for $V_1$ remain unchanged:

\[
\Pr^{\neg v_1, v_3}(v_1) = 0, \quad \Pr^{\neg v_1, v_3}(\neg v_1) = 1
\]

The probabilities for node $V_5$ remain unchanged. Why?

Therefore

\[
\Pr^{\neg v_1, v_3}(v_5) = \Pr^{\neg v_1}(\neg v_5) = 0.8, \quad \Pr^{\neg v_1, v_3}(\neg v_5) = 0.2
\]
Another example (3)

\[
\begin{align*}
\gamma(v_2 | v_1) &= 0.5 \\
\gamma(v_2 | \neg v_1) &= 0.4 \\
\gamma(v_3 | v_2) &= 0.2 \\
\gamma(v_3 | \neg v_2) &= 0.3 \\
\gamma(v_1) &= 0.7 \\
\gamma(v_5 | \neg v_1) &= 0.8 \\
\gamma(v_5 | v_1) &= 0.1 \\
\gamma(v_4 | v_2) &= 0.8 \\
\gamma(v_4 | \neg v_2) &= 0
\end{align*}
\]

Node \(V_3\) computes:

\[
\Pr^{\neg v_1, v_3}(v_3) = \alpha \cdot \pi(v_3) \cdot \lambda(v_3) = \alpha \cdot \pi(v_3) = \alpha \cdot 0.26 \quad \text{Why?}
\]
\[
\Pr^{\neg v_1, v_3}(\neg v_3) = \alpha \cdot \pi(\neg v_3) \cdot \lambda(\neg v_3) = 0
\]

After normalisation: \( \Pr^{\neg v_1, v_3}(v_3) = 1 \), \( \Pr^{\neg v_1, v_3}(\neg v_3) = 0 \)

Node \(V_3\) computes for node \(V_2\):

\[
\lambda_{V_3}^{V_2}(V_2) = \sum_{c_{V_3}} \lambda(V_3) \cdot \gamma(c_{V_3} | V_2)
\]
Another example (4)

\[ \gamma(v_1 | v_2) = 0.5 \]
\[ \gamma(v_1 | \neg v_2) = 0.4 \]

\[ \gamma(v_2 | v_3) = 0.2 \]
\[ \gamma(v_2 | \neg v_3) = 0.3 \]

Node \( V_2 \) computes:

\[
\Pr^{\neg v_1, v_3}(v_2) = \alpha \cdot \pi(v_2) \cdot \lambda(v_2) = \alpha \cdot \pi(v_2) \cdot \lambda_{V_3}^V(v_2) \cdot \lambda_{V_4}^V(v_2) \\
= \alpha \cdot \pi(v_2) \cdot \gamma(v_3 | v_2) = \alpha \cdot 0.4 \cdot 0.2 = \alpha \cdot 0.08
\]

\[
\Pr^{\neg v_1, v_3}(\neg v_2) = \alpha \cdot \pi(\neg v_2) \cdot \lambda(\neg v_2) = \alpha \cdot \pi(\neg v_2) \cdot \lambda_{V_3}^V(\neg v_2) \cdot \lambda_{V_4}^V(\neg v_2) \\
= \alpha \cdot \pi(\neg v_2) \cdot \gamma(v_3 | \neg v_2) = \alpha \cdot 0.6 \cdot 0.3 = \alpha \cdot 0.18
\]

Normalisation results in:
\[ \Pr^{\neg v_1, v_3}(v_2) = 0.31, \quad \Pr^{\neg v_1, v_3}(\neg v_2) = 0.69 \]
Another example (5)

\[ \gamma(v_2 \mid v_1) = 0.5 \]
\[ \gamma(v_2 \mid \neg v_1) = 0.4 \]
\[ \gamma(v_3 \mid v_2) = 0.2 \]
\[ \gamma(v_3 \mid \neg v_2) = 0.3 \]
\[ \gamma(v_5 \mid v_1) = 0.1 \]
\[ \gamma(v_5 \mid \neg v_1) = 0.8 \]
\[ \gamma(v_4 \mid v_2) = 0.8 \]
\[ \gamma(v_4 \mid \neg v_2) = 0 \]

Node \( V_2 \) computes for node \( V_4 \):

\[ \pi_{V_4}^{V_2}(V_2) = \alpha \cdot \pi(V_2) \cdot \lambda_{V_3}^{V_2}(V_2) \Rightarrow 0.31 \text{ and } 0.69 \]

Node \( V_4 \) computes:

\[ \Pr^{v_1,v_3}(v_4) = \pi(v_4) = \gamma(v_4 \mid v_2) \cdot \pi_{V_4}^{V_2}(v_2) + \gamma(v_4 \mid \neg v_2) \cdot \pi_{V_4}^{V_2}(\neg v_2) \]
\[ = \gamma(v_4 \mid v_2) \cdot \pi_{V_4}^{V_2}(v_2) + 0 = 0.8 \cdot 0.31 = 0.248 \]
\[ \Pr^{v_1,v_3}(\neg v_4) = 0.2 \cdot 0.31 + 1.0 \cdot 0.69 = 0.752 \]
Entering evidence: a singly connected example

Let $Pr$ and $B$ be as before:

$\gamma(v_2) = 0.1$
$\gamma(\neg v_2) = 0.9$

$\gamma(v_1 \mid v_2 \land v_3) = 0.8$
$\gamma(v_1 \mid \neg v_2 \land v_3) = 0.9$
$\gamma(v_1 \mid v_2 \land \neg v_3) = 0.5$
$\gamma(v_1 \mid \neg v_2 \land \neg v_3) = 0.6$

$\gamma(v_3) = 0.4$
$\gamma(\neg v_3) = 0.6$

$\gamma(\neg v_1 \mid v_2 \land v_3) = 0.2$
$\gamma(\neg v_1 \mid \neg v_2 \land v_3) = 0.1$
$\gamma(\neg v_1 \mid v_2 \land \neg v_3) = 0.5$
$\gamma(\neg v_1 \mid \neg v_2 \land \neg v_3) = 0.4$

Evidence $V_1 = true$ is entered.

**Assignment:** compute $Pr^{v_1}(V_2) = \alpha \cdot \pi(V_2) \cdot \lambda(V_2)$. 
An example with evidence $V_1 = true$ (2)

\[
\begin{align*}
\gamma(v_2) &= 0.1 \\
\gamma(\neg v_2) &= 0.9 \\
\gamma(v_1 \mid v_2 \land v_3) &= 0.8 \\
\gamma(v_1 \mid \neg v_2 \land v_3) &= 0.9 \\
\gamma(v_1 \mid v_2 \land \neg v_3) &= 0.5 \\
\gamma(v_1 \mid \neg v_2 \land \neg v_3) &= 0.6 \\
\gamma(v_3) &= 0.4 \\
\gamma(\neg v_3) &= 0.6 \\
\gamma(\neg v_1 \mid v_2 \land v_3) &= 0.2 \\
\gamma(\neg v_1 \mid \neg v_2 \land v_3) &= 0.1 \\
\gamma(\neg v_1 \mid v_2 \land \neg v_3) &= 0.5 \\
\gamma(\neg v_1 \mid \neg v_2 \land \neg v_3) &= 0.4
\end{align*}
\]

Node $V_1$ computes for node $V_2$:

\[
\begin{align*}
\lambda_{V_1}^{V_2}(v_2) &= \lambda(v_1) \cdot [\gamma(v_1 \mid v_2 \land v_3) \cdot \pi_{V_1}^{V_3}(v_3) + \\
&\quad \gamma(v_1 \mid v_2 \land \neg v_3) \cdot \pi_{V_1}^{V_3}(\neg v_3)] + \\
&\lambda(\neg v_1) \cdot [\gamma(\neg v_1 \mid v_2 \land v_3) \cdot \pi_{V_1}^{V_3}(v_3) + \\
&\quad \gamma(\neg v_1 \mid v_2 \land \neg v_3) \cdot \pi_{V_1}^{V_3}(\neg v_3)] \\
&= 0.8 \cdot 0.4 + 0.5 \cdot 0.6 = 0.62 \\
\lambda_{V_1}^{V_2}(\neg v_2) &= 0.9 \cdot 0.4 + 0.6 \cdot 0.6 = 0.72
\end{align*}
\]
An example with evidence $V_1 = true$ (3)

Node $V_2$ computes:

$$\Pr^{v_1}(v_2) = \alpha \cdot \pi(v_2) \cdot \lambda(v_2) = \alpha \cdot \gamma(v_2) \cdot \lambda^{V_2}_{V_1}(v_2) = \alpha \cdot 0.1 \cdot 0.62 = 0.062\alpha$$

$$\Pr^{v_1}(\neg v_2) = \alpha \cdot 0.9 \cdot 0.72 = 0.648\alpha$$

Normalisation gives: $\Pr^{v_1}(v_2) \sim 0.087$, $\Pr^{v_1}(\neg v_2) \sim 0.913$
Initially, the Bayesian network is in a stable situation.

Once evidence is entered into the network, this stability is disturbed.
Evidence initiates message passing throughout the entire network:

When each node in the network has been visited by the message passing algorithm, the network returns to a new stable situation.
Pearl: some complexity issues

Consider a Bayesian network \( \mathcal{B} \) with singly connected digraph \( G \) with \( n \geq 1 \) nodes. Suppose that node \( V \) has \( O(n) \) parents and \( O(n) \) children:

\[
\begin{align*}
W_1 & \quad \cdots \quad W_i & \quad \cdots \quad W_p \\
\vdots & \quad \cdots \quad \vdots & \quad \cdots \quad \vdots \\
Z_1 & \quad \cdots \quad Z_j & \quad \cdots \quad Z_s
\end{align*}
\]

\( \rho(V) \)

\( \sigma(V) \)

- Computing the compound causal parameter requires at most \( O(2^n) \) time:

\[
\pi(V) = \sum_{c_{\rho(V)}} \gamma(V \mid c_{\rho(V)}) \cdot \prod_{k=1,\ldots,p} \pi_{V}^{W_i}(c_{W_i})
\]
Computing the compound diagnostic parameter requires at most $O(n)$ time:

$$\lambda(V) = \prod_{j=1,\ldots,s} \lambda_{Z_j}^V(V)$$

A node can therefore compute the probabilities for its values in at most $O(2^n)$ time.
Computing a causal parameter requires constant time:

$$\pi_{Z_j}^V (V) = \alpha \cdot \pi(V) \cdot \prod_{k=1,\ldots,s,k \neq j} \lambda_{Z_k}^V (V) = \frac{\Pr(V)}{\lambda_{Z_j}^V (V)}$$
• Computing a diagnostic parameter requires at most $O(2^n)$ time: $\lambda_V^{W_i}(W_i) = \alpha \cdot \sum_{c_V} \lambda(c_V) \cdot \left[ \sum_{c_{\rho(V)} \setminus \{W_i\}} (\gamma(V \mid c_{\rho(V)} \setminus \{W_i\} \land W_i) \cdot \prod_{k=1,\ldots,p,k\neq i} \pi_V^{W_k}(c_{W_k})) \right]$

A node can compute the parameters for all its neighbours in at most $O(n \cdot 2^n)$ time. Processing evidence requires at most $O(n^2 \cdot 2^n)$ time.
Inference in multiply connected digraphs

When applying Pearl’s algorithm to a Bayesian network with a multiply connected digraph, the following problems result:

- the message passing does not necessarily reach an equilibrium;
- even if an equilibrium is reached, the computed probabilities are not necessarily correct.

These problems result from the fact that Pearl’s algorithm assumes independencies that are invalid in the Bayesian network to which it is applied.

⇒ approximation algorithm ’Loopy belief propagation’
No equilibrium: an example

Consider the Bayesian network $\mathcal{B} = (G, \Gamma)$ with the following multiply connected digraph $G$:

![Diagram of a Bayesian network with nodes $V_1$ to $V_5$. $V_1$ points to $V_2$, $V_2$ points to $V_3$ and $V_4$, $V_3$ points to $V_4$, and $V_4$ points to $V_5$.]

If node $V_5$ is instantiated, then the message passing does not necessarily reach an equilibrium.

Why?
Incorrect computations: an example (1)

Consider the Bayesian network with digraph:

```
V1  
|   
V2  V3  V4  V5
```

Suppose that evidence $V_1 = \text{true}$ is obtained and that we are interested in $\Pr^{v_1}(V_5)$.

We have, by marginalisation and independence, that

$$\Pr^{v_1}(V_5) = \sum_{c\{v_2, v_3, v_4\}} \Pr(V_5 \wedge c\{v_2, v_3, v_4\} \mid v_1)$$

$$= \sum_{c\{v_3, v_4\}} \Pr(V_5 \mid c\{v_3, v_4\}) \cdot \sum_{c_{v_2}} \Pr(c_{v_3} \mid c_{v_2}) \cdot \Pr(c_{v_4} \mid c_{v_2}) \cdot \Pr(c_{v_2} \mid v_1)$$

Note the same value $c_{v_2}$ in the product of the last three terms!
Incorrect computations: an example (2)

Consider the Bayesian network with digraph:

\[ V_1 \rightarrow V_2 \leftarrow V_3 \rightarrow V_4 \leftarrow V_5 \]

Suppose that evidence \( V_1 = true \) is obtained and that we are interested in \( \Pr^{v_1}(V_5) \).

Pearl’s algorithm basically computes:

\[
\Pr^{v_1}(V_5) = \Pr(V_5 \mid v_3 \land v_4) \cdot \Pr(v_3 \mid v_1) \cdot \Pr(v_4 \mid v_1) \\
+ \Pr(V_5 \mid \neg v_3 \land v_4) \cdot \Pr(\neg v_3 \mid v_1) \cdot \Pr(v_4 \mid v_1) \\
+ \Pr(V_5 \mid v_3 \land \neg v_4) \cdot \Pr(v_3 \mid v_1) \cdot \Pr(\neg v_4 \mid v_1) \\
+ \Pr(V_5 \mid \neg v_3 \land \neg v_4) \cdot \Pr(\neg v_3 \mid v_1) \cdot \Pr(\neg v_4 \mid v_1)
\]

and

\[
\Pr(V_3 \mid v_1) = \Pr(V_3 \mid v_2) \cdot \Pr(v_2 \mid v_1) + \Pr(V_3 \mid \neg v_2) \cdot \Pr(\neg v_2 \mid v_1) \\
\Pr(V_4 \mid v_1) = \Pr(V_4 \mid v_2) \cdot \Pr(v_2 \mid v_1) + \Pr(V_4 \mid \neg v_2) \cdot \Pr(\neg v_2 \mid v_1)
\]
Incorrect computations: an example (3)

Suppose that evidence $V_1 = true$ is obtained and that we are interested in $\Pr^{v_1}(V_5)$.

Substitution of $\Pr(V_3 \mid v_1)$ and $\Pr(V_4 \mid v_1)$ thus results in incorrect terms, such as for example

$$\Pr(v_5 \mid v_3 \land v_4) \cdot \Pr(v_3 \mid v_2) \cdot \Pr(v_2 \mid v_1) \cdot \Pr(v_4 \mid \neg v_2) \cdot \Pr(\neg v_2 \mid v_1)$$

What is causing this problem? How can we solve this?
Correct computations: an example

Suppose that evidence $V_1 = true$ is obtained and that we are interested in $\Pr^{v_1}(V_5)$.
We have, by conditioning, that:

$$
\Pr^{v_1}(V_5) = \Pr(V_5 | v_2 \land v_1) \cdot \Pr(v_2 | v_1) +
+ \Pr(V_5 | \neg v_2 \land v_1) \cdot \Pr(\neg v_2 | v_1)
$$

Pearl’s algorithm can correctly compute:
$\Pr^{v_1}(V_5 | V_2)$, e.g.

$$
\Pr^{v_1}(V_5 | v_2) = \Pr(V_5 | v_3 \land v_4) \cdot \Pr(v_3 | v_2 \land v_1) \cdot \Pr(v_4 | v_2 \land v_1) +
+ \Pr(V_5 | \neg v_3 \land v_4) \cdot \Pr(\neg v_3 | v_2 \land v_1) \cdot \Pr(v_4 | v_2 \land v_1) +
+ \Pr(V_5 | v_3 \land \neg v_4) \cdot \Pr(v_3 | v_2 \land v_1) \cdot \Pr(\neg v_4 | v_2 \land v_1) +
+ \Pr(V_5 | \neg v_3 \land \neg v_4) \cdot \Pr(\neg v_3 | v_2 \land v_1) \cdot \Pr(\neg v_4 | v_2 \land v_1)
$$

Summing out $V_2$ equals:

$$
\Pr^{v_1}(V_5) = \sum_{c\{v_2,v_3,v_4\}} \Pr(V_5 \land c_{\{v_2,v_3,v_4\}} | v_1)
$$
Consider the Bayesian network $\mathcal{B} = (\mathcal{G}, \Gamma)$ with the following digraph $\mathcal{G}$:

When node $V_2$ is instantiated, the digraph $\mathcal{G}$ behaves as a singly connected digraph:

For which of the other nodes does a similar observation hold?
A solution: Cutset Conditioning

Let \( G = (V_G, A_G) \) be an acyclic digraph.

The idea behind cutset conditioning is:

1. Select a loop cutset of \( G \): nodes \( L_G \subseteq V_G \) such that instantiating \( L_G \) makes the digraph ‘behave’ as if it were singly connected.

2. Compute for all possible loop cutset configurations \( c_{L_G} \) the probabilities \( \Pr(V \mid c_{L_G}) \) for each \( V \in V_G \).

3. Marginalise out (= sum out) the loop cutset node(s) \( L_G \).
A loop cutset

**Definition:** Let $G = (V_G, A_G)$ be an acyclic digraph.

A set $L_G \subseteq V_G$ is called a loop cutset of $G$ if:

- every simple cyclic chain (loop) $s$ in $G$ contains a node $X$ such that:
  - $X \in L_G$, and
  - $X$ has at most one incoming arc on $s$. 
An example: loop cutsets

Consider the following digraph $G$:

- How many loops does $G$ contain?
- Which of the following sets are loop cutsets of $G$?:
  - $\emptyset$
  - $\{V_1\}$
  - $\{V_3\}$
  - $\{V_1, V_5\}$
  - $\{V_2, V_7\}$
  - $\{V_4, V_7\}$
  - $\{V_1, V_2, V_3\}$
  - $\{V_1, V_4, V_5, V_6, V_7\}$
Consider Bayesian network $B$ with multiply connected digraph $G$:

\[
\begin{align*}
\gamma(v_1) &= 0.8 \\
\gamma(v_2 \mid v_1) &= 0.9 \\
\gamma(v_2 \mid \neg v_1) &= 0.3 \\
\gamma(v_3 \mid v_1) &= 0.2 \\
\gamma(v_3 \mid \neg v_1) &= 0.6 \\
\gamma(v_4 \mid v_2 \land v_3) &= 0.1 \\
\gamma(v_4 \mid \neg v_2 \land v_3) &= 0.2 \\
\gamma(v_4 \mid v_2 \land \neg v_3) &= 0.6 \\
\gamma(v_5 \mid v_2) &= 0.4 \\
\gamma(v_5 \mid \neg v_2) &= 0.5
\end{align*}
\]

We are interested in the probabilities $\Pr(v_4)$ and $\Pr(\neg v_4)$. We choose $L_G = \{V_1\}$. Pearl’s algorithm is now applied twice:

(I) $V_1 = true$  (II) $V_1 = false$
Pearl with cutset conditioning: example (2: general)

Pearl applied to (I) gives $\Pr(v_4 \mid v_1)$ and $\Pr(\neg v_4 \mid v_1)$;
Pearl applied to (II) gives $\Pr(v_4 \mid \neg v_1)$ and $\Pr(\neg v_4 \mid \neg v_1)$.

The probabilities of interest are finally computed using marginalisation (probability theory):

\[
\begin{align*}
\Pr(v_4) & = \Pr(v_4 \mid v_1) \cdot \Pr(v_1) + \Pr(v_4 \mid \neg v_1) \cdot \Pr(\neg v_1) \\
\Pr(\neg v_4) & = \Pr(\neg v_4 \mid v_1) \cdot \Pr(v_1) + \Pr(\neg v_4 \mid \neg v_1) \cdot \Pr(\neg v_1)
\end{align*}
\]

where $\Pr(v_1) = 0.8$, $\Pr(\neg v_1) = 0.2$ are the prior probabilities for node $V_1$ (not conditioned on loop cutset configurations!)
Pearl with cutset conditioning: example (3: in detail)

\[ \gamma(v_1) = 0.8 \]
\[ \gamma(v_2 \mid v_1) = 0.9 \]
\[ \gamma(v_2 \mid \neg v_1) = 0.3 \]
\[ \gamma(v_3 \mid v_1) = 0.2 \]
\[ \gamma(v_3 \mid \neg v_1) = 0.6 \]
\[ \gamma(v_4 \mid v_2 \land v_3) = 0.1 \]
\[ \gamma(v_4 \mid \neg v_2 \land v_3) = 0.2 \]
\[ \gamma(v_4 \mid v_2 \land \neg v_3) = 0.6 \]
\[ \gamma(v_4 \mid \neg v_2 \land \neg v_3) = 0.1 \]

Pearl applied to situation (I) where \( V_1 = true \):

\[
\Pr(v_4 \mid v_1) = \Pr^{v_1}(v_4) = \alpha \cdot \pi(v_4) \cdot \lambda(v_4) = \pi(v_4)
\]
\[
\Pr(\neg v_4 \mid v_1) = \Pr^{v_1}(\neg v_4) = \pi(\neg v_4)
\]

The compound causal parameter is computed:

\[
\pi(v_4) = \gamma(v_4 \mid v_2 \land v_3) \cdot \pi_{V_2}^{V_4}(v_2) \cdot \pi_{V_3}^{V_4}(v_3) +
\gamma(v_4 \mid \neg v_2 \land v_3) \cdot \pi_{V_2}^{V_4}(\neg v_2) \cdot \pi_{V_3}^{V_4}(v_3) +
\gamma(v_4 \mid v_2 \land \neg v_3) \cdot \pi_{V_2}^{V_4}(v_2) \cdot \pi_{V_3}^{V_4}(\neg v_3) +
\gamma(v_4 \mid \neg v_2 \land \neg v_3) \cdot \pi_{V_2}^{V_4}(\neg v_2) \cdot \pi_{V_3}^{V_4}(\neg v_3) = \ldots
\]
Pearl with cutset conditioning: example (4)

\[ \gamma(v_1) = 0.8 \]
\[ \gamma(v_2 \mid v_1) = 0.9 \]
\[ \gamma(v_2 \mid \neg v_1) = 0.3 \]
\[ \gamma(v_3 \mid v_1) = 0.2 \]
\[ \gamma(v_3 \mid \neg v_1) = 0.6 \]
\[ \gamma(v_4 \mid v_2 \land v_3) = 0.1 \]
\[ \gamma(v_4 \mid \neg v_2 \land v_3) = 0.2 \]
\[ \gamma(v_4 \mid v_2 \land \neg v_3) = 0.6 \]
\[ \gamma(v_4 \mid \neg v_2 \land \neg v_3) = 0.1 \]

\[ \pi(v_4) = 0.1 \cdot 0.9 \cdot 0.2 + 0.2 \cdot 0.1 \cdot 0.2 + 
\quad + 0.6 \cdot 0.9 \cdot 0.8 + 0.1 \cdot 0.1 \cdot 0.8 = 0.462 \]

Similarly, we find \( \pi(\neg v_4) = 0.538 \)
Pearl with cutset conditioning: example (5)

\[
\gamma(v_1) = 0.8 \\
\gamma(v_2 | v_1) = 0.9 \\
\gamma(v_2 | \neg v_1) = 0.3 \\
\gamma(v_3 | v_1) = 0.2 \\
\gamma(v_3 | \neg v_1) = 0.6 \\
\gamma(v_5 | v_2) = 0.4 \\
\gamma(v_5 | \neg v_2) = 0.5 \\
\gamma(v_4 | v_2 \land v_3) = 0.1 \\
\gamma(v_4 | \neg v_2 \land v_3) = 0.2 \\
\gamma(v_4 | v_2 \land \neg v_3) = 0.6 \\
\gamma(v_4 | \neg v_2 \land \neg v_3) = 0.1
\]

Pearl applied to situation (II) where \( V_1 = false \):

\[
\Pr(v_4 | \neg v_1) = \alpha \cdot \pi(v_4) \cdot \lambda(v_4) = \pi(v_4) \\
\Pr(\neg v_4 | \neg v_1) = \pi(\neg v_4)
\]

where

\[
\pi(v_4) = \gamma(v_4 | v_2 \land v_3) \cdot \pi_{V_2}^V(v_2) \cdot \pi_{V_3}^V(v_3) + \\
\gamma(v_4 | \neg v_2 \land v_3) \cdot \pi_{V_4}^V(\neg v_2) \cdot \pi_{V_4}^V(v_3) + \\
\gamma(v_4 | v_2 \land \neg v_3) \cdot \pi_{V_4}^V(v_2) \cdot \pi_{V_4}^V(\neg v_3) + \\
\gamma(v_4 | \neg v_2 \land \neg v_3) \cdot \pi_{V_4}^V(\neg v_2) \cdot \pi_{V_4}^V(\neg v_3) = \ldots
\]
Pearl with cutset conditioning: example (6)

\[ \gamma(v_1) = 0.8 \]

\[ \gamma(v_2 | v_1) = 0.9 \]
\[ \gamma(v_2 | \neg v_1) = 0.3 \]

\[ \gamma(v_3 | v_1) = 0.2 \]
\[ \gamma(v_3 | \neg v_1) = 0.6 \]

\[ \gamma(v_4 | v_2 \land v_3) = 0.1 \]
\[ \gamma(v_4 | \neg v_2 \land v_3) = 0.2 \]
\[ \gamma(v_4 | v_2 \land \neg v_3) = 0.6 \]
\[ \gamma(v_4 | \neg v_2 \land \neg v_3) = 0.1 \]

\[ \pi(v_4) = 0.1 \cdot 0.3 \cdot 0.6 + 0.2 \cdot 0.7 \cdot 0.6 + \\
+ 0.6 \cdot 0.3 \cdot 0.4 + 0.1 \cdot 0.7 \cdot 0.4 = 0.202 \]

Similarly, we find \( \pi(\neg v_4) = 0.798 \)
Pearl with cutset conditioning: example (7)

Recall: we are interested in $\Pr(v_4)$ and $\Pr(\neg v_4)$. With Pearl’s algorithm we computed

$$
\begin{align*}
\Pr(v_4 \mid v_1) &= 0.462 \\
\Pr(\neg v_4 \mid v_1) &= 0.538 \\
\Pr(v_4 \mid \neg v_1) &= 0.202 \\
\Pr(\neg v_4 \mid \neg v_1) &= 0.798
\end{align*}
$$

From the assessment functions we establish that

$$
\Pr(v_1) = 0.8, \quad \Pr(\neg v_1) = 0.2
$$

Resulting in (marginalisation)

$$
\begin{align*}
\Pr(v_4) &= \Pr(v_4 \mid v_1) \cdot \Pr(v_1) + \Pr(v_4 \mid \neg v_1) \cdot \Pr(\neg v_1) \\
&= 0.462 \cdot 0.8 + 0.202 \cdot 0.2 = 0.41 \\
\Pr(\neg v_4) &= \Pr(\neg v_4 \mid v_1) \cdot \Pr(v_1) + \Pr(\neg v_4 \mid \neg v_1) \cdot \Pr(\neg v_1) \\
&= 0.538 \cdot 0.8 + 0.798 \cdot 0.2 = 0.59
\end{align*}
$$
Cutset conditioning with evidence $\tilde{c}_{VG}$

Let $L_G$ be a loop cutset for digraph $G$. Then cutset conditioning exploits that for all $V_i \in V_G$:

$$\Pr(V_i \mid \tilde{c}_V) = \sum_{c_L} \Pr(V_i \mid \tilde{c}_V \land c_L) \cdot \Pr(c_L \mid \tilde{c}_V)$$

Pearl (from $B$) recursively

Recursion: step 1 for 1-st piece of evidence $e_1$:

$$\Pr(c_L \mid e_1) = \alpha \cdot \Pr(e_1 \mid c_L) \cdot \Pr(c_L)$$

Pearl (from $B$) marginalisation (from $\Pr$!)

Recursion: step $j$

$$\Pr(c_L \mid e_1 \land \ldots \land e_j) = \alpha \cdot \Pr(e_j \mid c_L \land e_1 \land \ldots \land e_{j-1}) \cdot \Pr(c_L \mid e_1 \land \ldots \land e_{j-1})$$

Pearl (from $B$)

Step $j - 1$
An example: cutset conditioning with evidence

Reconsider the Bayesian network $\mathcal{B}$:

$\gamma(v_1) = 0.8$

$\gamma(v_2 | v_1) = 0.9$
$\gamma(v_2 | \neg v_1) = 0.3$

$\gamma(v_3 | v_1) = 0.2$
$\gamma(v_3 | \neg v_1) = 0.6$

$\gamma(v_4 | v_2 \land v_3) = 0.1$
$\gamma(v_4 | \neg v_2 \land v_3) = 0.2$
$\gamma(v_4 | v_2 \land \neg v_3) = 0.6$
$\gamma(v_4 | \neg v_2 \land \neg v_3) = 0.1$

$\gamma(v_5 | v_2) = 0.4$
$\gamma(v_5 | \neg v_2) = 0.5$

Use loop cutset $\{V_1\}$.

Initially we have loop cutset configurations:

$\Pr(v_1) = 0.8$ and $\Pr(\neg v_1) = 0.2$.

Let’s process evidence $V_3 = false$. Updated probabilities are now established for the loop cutset configurations:

Pearl old

$\Pr^{-v_3}(v_1) = \alpha \cdot \Pr(\neg v_3 | v_1) \cdot \Pr(v_1) = \alpha \cdot 0.8 \cdot 0.8 = \alpha \cdot 0.64$

$\Rightarrow 0.89$

$\Pr^{-v_3}(\neg v_1) = \alpha \cdot \Pr(\neg v_3 | \neg v_1) \cdot \Pr(\neg v_1) = \alpha \cdot 0.4 \cdot 0.2 = \alpha \cdot 0.08$

$\Rightarrow 0.11$
We are interested in $\Pr^{\neg v_3}(v_4)$ and $\Pr^{\neg v_3}(-v_4)$. Pearl’s algorithm is applied twice:

(I) $V_1 = true$

\[
\begin{align*}
\Pr(v_4 | v_1 \land \neg v_3) &= 0.55 \\
\Pr(-v_4 | v_1 \land \neg v_3) &= 0.45
\end{align*}
\]

(II) $V_1 = false$

\[
\begin{align*}
\Pr(v_4 | \neg v_1 \land \neg v_3) &= 0.25 \\
\Pr(-v_4 | \neg v_1 \land \neg v_3) &= 0.75
\end{align*}
\]

Recall that $\Pr^{\neg v_3}(v_1) = 0.89$, $\Pr^{\neg v_3}(-v_1) = 0.11$. The probabilities of interest are now computed from

\[
\begin{align*}
\Pr^{\neg v_3}(v_4) &= \Pr(v_4 | v_1 \land \neg v_3) \cdot \Pr(v_1 | \neg v_3) \\
&\quad + \Pr(v_4 | \neg v_1 \land \neg v_3) \cdot \Pr(-v_1 | \neg v_3) \\
&= 0.55 \cdot 0.89 + 0.25 \cdot 0.11 = 0.52
\end{align*}
\]

$\Pr^{\neg v_3}(-v_4) = 0.48$
**Minimal and optimal loop cutsets**

**Definition:** A loop cutset $L_G$ for acyclic digraph $G$ is called

- **minimal:** if no real subset $L \subset L_G$ is a loop cutset for $G$;
- **optimal:** if for all loop cutsets $L'_G \neq L_G$ for $G$: $|L'_G| \geq |L_G|$.

**Example:** Consider the following acyclic digraph $G$:

Which of the following loop cutsets for $G$ are *minimal*; which are *optimal*?

$\{V_3\}$, $\{V_1, V_3\}$, $\{V_1, V_5\}$
Finding an optimal loop cutset

**Lemma:** The problem of finding an optimal loop cutset for an acyclic digraph is NP-hard.

**Proof:** The property can be proven by reduction from the “Minimal Vertex Cover”-Problem. For details, see

A heuristic algorithm

The following algorithm is a heuristic for finding an optimal loop cutset for a given acyclic digraph $G$:

**PROCEDURE** **LOOP-CUTSET**($G$, $L_G$):

WHILE THERE ARE NODES IN $G$ DO

  IF THERE IS A NODE $V_i \in V_G$ WITH $\text{degree}(V_i) \leq 1$ THEN SELECT NODE $V_i$

  ELSE DETERMINE ALL NODES $K = \{ V \in V_G | \text{indegree}(V) \leq 1 \}$ (THE CANDIDATES FOR THE LOOP CUTSET);

    SELECT A CANDIDATE NODE $V_i \in K$ WITH $\text{degree}(V_i) \geq \text{degree}(V)$ FOR ALL OTHER $V \in K$;

    ADD NODE $V_i$ TO THE LOOP CUTSET $L_G$

  FI;

DELETE NODE $V_i$ AND ITS INCIDENT ARCS FROM $G$

OD;

END
An example

Consider the following acyclic digraph:

(Recursively) deleting all nodes $V_i$ with $\text{degree}(V_i) \leq 1$ results in ...

An example

(Recursively) deleting all nodes $V_i$ with $\text{degree}(V_i) \leq 1$ results in:

Which nodes are candidates for the loopcutset?

Suppose that node $V_4$ is selected and added to the loop cutset...
After deleting node $V_4$ and recursively deleting all remaining $V_i$ with $\text{degree}(V_i) \leq 1$ we get:

Which nodes are candidates for the loop cutset?

Suppose that node $V_7$ is now selected for the loop cutset. After deleting node $V_7$ and recursively deleting all remaining nodes $V_i$ with $\text{degree}(V_i) \leq 1$ the empty graph results.

The loop cutset found is $\{V_4, V_7\}$.

Are there other possibilities?
Some properties of the heuristic algorithm

- It always finds a loop cutset for a given acyclic digraph;
- It does not always find an optimal loop cutset;

**Example:** Consider the following graph $G$:

What is the optimal loop cutset for $G$? Why won’t the algorithm find this loop cutset?

- It found an optimal loop cutset for 70% of the graphs randomly generated in an experiment.
Some properties – continued

• the heuristic does not always find a minimal loop cutset.

**Example:** Reconsider graph $G$:

![Graph Diagram]

The algorithm could, for example, return the loop cutset $\{V_1, V_3\}$ for $G$; this loop cutset is not minimal.

Note that this problem can be easily resolved afterwards. How?
the heuristic can select nodes for the loop cutset that are not on a cyclic chain.

**Example:**
Consider the following graph $G$, where $G_1, \ldots, G_k$, $k \gg 1$, are non-singly connected graphs:

The algorithm can select node $V$ for addition to the loop cutset. ■

Can this be resolved easily?
Pearl: complexity issues

Consider a Bayesian network $\mathcal{B} = (G, \Gamma)$.

- Let $G$ be a singly connected digraph with $n \geq 1$ nodes $V_i \in V_G$.
  
  If $|\rho(V_i)|$ in $G$ is bounded by a constant, then $V_i$ can compute the probabilities for its values and the parameters for its neighbours in polynomial time.

- Let $G$ be a multiply connected digraph with $n \geq 1$ nodes $V_i \in V_G$ and let $L_G$ be a loop cutset for $G$.

  If Pearl’s algorithm is used in combination with loop cutset conditioning, then node $V_i$ does its calculations $2^{|L_G|}$ times.
Summary Pearl: idea and complexity

Idea of Pearl’s algorithm extended with loop cutset conditioning:

- loop cutset $\rightarrow$ multiply connected graph behaves singly connected
- update probabilities by message passing between nodes (= ‘standard’ Pearl)
- marginalise out loop cutset

Complexity for all $\Pr(V_i \mid c_E)$ simultaneously:

- singly connected graphs: polynomial in # of nodes, for bounded number of parents;
- multiply connected graphs: exponential in lcs size, even for bounded number of parents.
Probabilistic inference: complexity issues

- In general, probabilistic inference with an arbitrary Bayesian network is NP-hard;
  

  This even holds for approximation algorithms, such as e.g. *loopy propagation*!

- all existing algorithms for probabilistic inference have an exponential worst-case complexity;

- the existing algorithms for probabilistic inference have a polynomial time complexity for certain types of Bayesian network (the sparser the graph, the better).
Chapter 5:

Building a Bayesian Network
The construction of a Bayesian network

Construction of a Bayesian network for an application domain involves three different tasks:

- to identify the (statistical) variables and their values;
- to construct the digraph of the network;
- to assess the (conditional) probabilities required for the variables.

Methodologies for building networks by hand do not yet abound!

Building a Bayesian network resembles building any type of system, thereby warranting the use of an overall systems-engineering approach.

In practice, the construction of a Bayesian network is an iterative process involving testing and evaluation as well.
The trade-off in construction

The construction of a Bayesian network requires a careful trade-off between

- the desire for a rich and detailed model;
- the costs of construction and maintenance;
- the run-time complexity of probabilistic inference.
Establishing variables and their values

Establishing the variables and their values for a Bayesian network amounts to

- identifying the important domain variables and values from
  - an introductory study of the domain literature;
  - interviews with one or more domain experts;

- modelling the identified domain variables:

  domain variables are captured as statistical variables in such a way that their values are
  - mutually exclusive;
  - collectively exhaustive;

- giving an unambiguous description of the modelled variables and values.
Modelling domain variables

Single-valued domain variables are relatively easy to capture as statistical variables:

- single-valued discrete variables can be modelled directly;
- single-valued continuous cannot be modelled directly: the range of values should be discretised;

Multi-valued domain variables cannot be directly captured as statistical variables.
Single-valued variables

The value range of a single-valued variable with a large range of ordered values can be divided into intervals.

- For a continuous variable the value range must always be divided into intervals.

  **Example:** For a variable *Fever* we can distinguish the intervals $[36; 37)$, $[37; 38)$, $[38; 39)$ and $[39; 40]$.

- For a discrete variable pragmatical reasons can exist to divide its value range into intervals.

  **Example:** For a variable *Age* we can distinguish the intervals $[0; 50)$, $[50; 65)$, $[65; 70)$, $[70; 75)$, $[75; 80)$ and $[80; 120]$.

Each single interval of domain values is considered a single value of the corresponding statistical variable.
Modelling Multi-valued variables

If a variable is multi-valued then this often indicates that it is composed of various other variables.

- a multi-valued domain variable can sometimes be modelled as a single single-valued statistical variable;
- a multi-valued variable is usually modelled as a collection of single-valued statistical variables.
Multi-valued variables, an example

Consider the domain variable \textit{BloodCount} that adopts one or more of the values \textit{normal}, \textit{lymphocytosis}, \textit{lymphocytopenia}, \textit{leucocytosis}, and \textit{leucocytopenia}; possible combinations are:

\begin{align*}
\{\text{normal}\} & \quad \{\text{lymphocytosis, leucocytosis}\} \\
\{\text{leucocytosis}\} & \quad \{\text{lymphocytosis, leucocytopenia}\} \\
\{\text{lymphocytosis}\} & \quad \{\text{lymphocytopenia, leucocytosis}\} \\
\{\text{leucocytopenia}\} & \quad \{\text{lymphocytopenia, leucocytopenia}\} \\
\{\text{lymphocytopenia}\} &
\end{align*}

- the variable can be modelled as a single statistical variable with the nine possible combinations of its values;
- the variable can be modelled by two statistical variables:
  - the variable \textit{LymphocyteCount} with the three values \textit{normal}, \textit{lymphocytosis}, \textit{lymphocytopenia};
  - the variable \textit{LeucocyteCount} with the three values \textit{normal}, \textit{leucocytosis}, \textit{leucocytopenia}. 
A trade-off in modelling domain variables

The difference between variables and values is not always clear; the choice of representation can have a large impact.

**Example:** Consider modelling the depth of invasion of an oesophageal tumour

- as the single variable *Invasion*, with seven values: *T1, T2, T3, diaphragm, mediastinum, trachea, and heart*
A trade-off in modelling domain variables

The difference between variables and values is not always clear; the choice of representation can have a large impact.

Example: Consider modelling the depth of invasion of an oesophageal tumour as a single variable:
A trade-off in modelling domain variables

The difference between variables and values is not always clear; the choice of representation can have a large impact.

Example: Consider modelling the depth of invasion of an oesophageal tumour

• as the single variable Invasion
• as a combination of the two variables Invasion Wall (with four values: T1, T2, T3 and T4) and Invasion Organs (with five values: none, diaphragm, mediastinum, trachea and heart, where T1 ∨ T2 ∨ T3 is equivalent to none)
A trade-off in modelling domain variables

The difference between variables and values is not always clear; the choice of representation can have a large impact.

Example: Consider modelling the depth of invasion of an oesophageal tumour with two variables:

\[
\begin{align*}
\text{Location} & \rightarrow \text{Shape} & \rightarrow \text{Length} & \rightarrow \text{Circumf.} \\
\text{Invasion organs} & \rightarrow \text{Necrosis} & \rightarrow \text{Lymph. metas.} & \rightarrow \text{Haema. metas.} \\
\text{CT-organs} & \rightarrow \text{Bronchoscopy} & \rightarrow \text{Fistula} & \rightarrow \text{Invasion wall}
\end{align*}
\]
A trade-off in modelling domain variables

The difference between variables and values is not always clear; the choice of representation can have a large impact.

Example: Consider modelling the depth of invasion of an oesophageal tumour

• as the single variable *Invasion*
• as a combination of the two variables *Invasion Wall* and *Invasion Organs*

The number of non-redundant probability assessments required in the second representation is less than 40% of that required in the first representation!
The level of detail of modelling heavily depends on the purpose of the constructed system.

**Example:**

Compare the variables *Cardiovascular condition* and *Pulmonary condition* to the level of representation detail of invasion and the process of metastasis of the tumour.
Definition: The variable \textit{Location} models the longitudinal position in the oesophagus of the center of the primary tumour, relative to the location of the stomach.

Causes: The location of the primary tumour has no direct causes, but is strongly correlated to its histological type.

Values: The variable \textit{Location} can adopt one of the values \textit{proximal}, \textit{mid} and \textit{distal}:

- \textit{proximal}: the tumour’s center is in the upper $\frac{1}{3}$ of the oesophagus;
- \textit{mid}: the tumour’s center is in the middle $\frac{1}{3}$ of the oesophagus;
- \textit{distal}: the tumour’s center is in the lower $\frac{1}{3}$ of the oesophagus.

Probabilistic information: For the variable \textit{Location} are specified 3 probabilities: $\Pr(\text{Location})$
The construction of the digraph

- the digraph of a Bayesian network can be constructed by hand, with the help of domain expert(s);

- the digraph of a Bayesian network can sometimes be constructed automatically from an up-to-date dataset.
Constructing the digraph by hand

For the construction of the digraph of a Bayesian network by hand, the notion of causality is used as a heuristic guiding principle:

“What could cause this effect?”
“What manifestations could this cause have?”

The elicited causal relationships are directed from cause to effect.

Since causality is merely a guiding principle, the resulting independences need to be verified explicitly!
Fine-tuning the digraph: correlations

By using causality as a guiding principle, correlations are hard to capture.

Domain experts often have trouble indicating a direction for such a non-causal relation.

**Possible solutions:**

- introduce an intermediate variable to capture a common cause;
- assign a direction to the correlation based on independence.
Fine-tuning the digraph: indirect arcs

By using causality as a guiding principle, superfluous arcs may arise.

Domain experts sometimes have trouble indicating the difference between indirect and direct causes and effects.

The independences can be reviewed by means of case descriptions.

Example:

“Suppose that, for a patient with a circular tumour, you have made an assessment of his ability to swallow food. Can additional knowledge of the tumour’s length change your assessment?”
Fine-tuning the digraph: cycles

By using causality as a guiding principle, cycles may arise.

- the cycle can be the consequence of an erroneous arc;
- the cycle can model a feedback process in the domain of application.

Any cycle needs to be broken, for example by

- deleting an appropriate arc, based upon domain knowledge;
- reversing an appropriate arc (not violating independences !);
- explicitly modelling the evolution of time of the underlying process.
An example cycle from a feedback process

Cirrhosis

Liver architecture

Portal hypertension

Portal blood flow

Splenomegaly

Functional splenomegaly

Liver cell mass

Liver synthesis capacity

Liver clearance capacity

Systemic antigens

Portasystemic collaterals

Portasystemic shunting

Congestive splenomegaly
An example cycle from a feedback process

A possible solution for breaking the cycle:
Experiences with handcrafting the digraph

Although handcrafting the digraph of a Bayesian network can take considerable time, it is doable:

- domain experts are allowed to express their knowledge and experience in either causal or diagnostic direction;
- domain experts tend to feel comfortable with digraphs as representations of their knowledge and experience;
- in various domains reusable components are available.
Consider a set of variables $V$. A Bayesian network can be automatically constructed from a dataset $D$:

- use some procedure to create a DAG $G$ with nodes $V$;
- use some procedure to establish the joint distribution over $V$ in $G$ from the information in the dataset;

These algorithms are often called learning algorithms and are typically iterative.

In general, we can distinguish two approaches to learning:

- conditional independence: learns either structure or probabilities;
- metric: does both, either supervised or unsupervised
A dataset

**Definition:**

Let $V$ be a set of domain variables. A dataset $D$ over $V$ is a multi-set of cases, which are configurations $c_V$ of $V$.

$D$ can be used for learning a Bayesian network $B = (G, \Gamma)$ if:

- the variables and values in $D$ are (easily) translated to the variables and values of the network under construction;
- every case in $D$ specifies a value for each variable;
- the cases in $D$ are generated independently;
- $D$ reflects a time-independent process;
- $D$ contains sufficient and reliable information.

The information in a dataset describes a joint probability distribution $\Pr_D(V)$ over its variables; this is an approximation of the true distribution $\Pr(V)$. 
Assessing probabilities from data

Let $V = \{V_1, \ldots, V_n\}, n \geq 1$, be a set of variables and let $D$ be a dataset over $V$ with $N$ cases. Any probability from $\Pr_D$ can now be obtained from $D$ by frequency counting.

For example, consider a variable $V_i \in V$ and a subset of variables $W \subseteq V \setminus \{V_i\}$. Then, e.g.

$$\Pr_D(c_{V_i}) = \frac{N(c_{V_i})}{N}, \quad \text{and}$$

$$\Pr_D(c_{V_i} \mid c_W) = \frac{\Pr_D(c_{V_i} \wedge c_W)}{\Pr_D(c_W)} = \frac{N(c_{V_i} \wedge c_W)/N}{N(c_W)/N} = \frac{N(c_{V_i} \wedge c_W)}{N(c_W)}$$

where $N(c)$ is the number of cases consistent with $c$. 


A conditional independence structure learning algorithm (brief)

Order the variables under consideration: \( V_1, \ldots, V_n \);
For \( i = 2 \) to \( n \) do

find a minimal set \( \delta(V_i) \subseteq \{V_1, \ldots, V_{i-1}\} \) such that
\[
I_D(\{V_i\}, \delta(V_i), \{V_1, \ldots, V_{i-1}\} \setminus \delta(V_i));
\]
\[
\rho(V_i) \leftarrow \delta(V_i);
\]

Benefit: guaranteed acyclic
Drawback: structure, and hence compactness, depends heavily on chosen ordering
A metric algorithm

An (unsupervised metric) algorithm for automated construction of a Bayesian network $\mathcal{B}$ from a dataset $\mathcal{D}$ consists of two components:

- a quality measure: indicates how good the learned model $\mathcal{B}$ “explains” the data, i.e. does $\Pr_{\mathcal{B}}$ match $\Pr_{\mathcal{D}}$?

  We consider the MDL quality measure. The measure requires a complete network with probabilities; these are again obtained by counting.

- a search procedure: a heuristic for finding a network with the highest quality given the dataset

  We consider the $B$ search heuristic (a hill-climber).
Assessing the probabilities for $B$

Let $V = \{V_1, \ldots, V_n\}, n \geq 1$, be a set of variables and let $D$ be a dataset over $V$ with $N$ cases. Let $G = (V_G, A_G)$ be a DAG with $V_G = V$.

For $G$, a corresponding set $\Gamma = \{\gamma_{V_i} \mid V_i \in V_G\}$ of assessment functions is obtained from $D$, by frequency counting. That is,

$$\gamma(c_{V_i} \mid c_{\rho(V_i)}) = Pr_D(c_{V_i} \mid c_{\rho(V_i)})$$

for each variable $V_i \in V$, every configuration $c_{V_i}$ of $V_i$ and all configurations $c_{\rho(V_i)}$ of the parent set $\rho(V_i)$ of $V_i$ in $G$.

Recall: if $\rho(V_i) = \emptyset$ then $c_{\rho(V_i)} = T \rightarrow N(T) = N$ for counting.
An example

Consider the following dataset $D$ and graph $G$:

$$
\neg v_1 \land \neg v_2 \land v_3 \land \neg v_4 \quad \checkmark \\
v_1 \land v_2 \land \neg v_3 \land \neg v_4 \\
v_1 \land v_2 \land v_3 \land \neg v_4 \\
\neg v_1 \land \neg v_2 \land v_3 \land v_4 \quad \checkmark \\
v_1 \land v_2 \land \neg v_3 \land \neg v_4 \\
v_1 \land v_2 \land \neg v_3 \land \neg v_4 \\
v_1 \land v_2 \land \neg v_3 \land v_4 \\
\neg v_1 \land \neg v_2 \land v_3 \land \neg v_4 \quad \checkmark \\
v_1 \land v_2 \land \neg v_3 \land v_4
$$

The values of $\gamma_{V_1}$ are assessed as follows:

$$
\gamma(\neg v_1) = \frac{N(\neg v_1)}{N} = \frac{6}{15} = 0.4 \quad \text{and} \quad \gamma(v_1) = \frac{N(v_1)}{N} = \frac{9}{15} = 0.6
$$
An example

Consider the following dataset \( D \) and graph \( G \):

\[
\begin{align*}
\neg v_1 & \land \neg v_2 \land v_3 \land \neg v_4 & & \checkmark \\
v_1 & \land v_2 \land \neg v_3 \land \neg v_4 & & \checkmark \\
v_1 & \land v_2 \land v_3 \land \neg v_4 & & \\
\neg v_1 & \land \neg v_2 \land v_3 \land v_4 & & \checkmark \\
v_1 & \land v_2 \land \neg v_3 \land \neg v_4 & & \checkmark \\
v_1 & \land v_2 \land \neg v_3 \land \neg v_4 & & \\
v_1 & \land v_2 \land \neg v_3 \land v_4 & & \checkmark \\
\neg v_1 & \land \neg v_2 \land v_3 \land \neg v_4 & & \checkmark \\
\end{align*}
\]

The values of \( \gamma_{V_2} \) are assessed as follows:

\[
\gamma(v_2 \mid \neg v_1) = \frac{N(\neg v_1 \land v_2)}{N(\neg v_1)} = \frac{3}{6} = 0.5, \text{ etc.}
\]
The quality of a graph

**Definition:** (‘MDL quality measure’) Let \( V = \{V_1, \ldots, V_n\}, n \geq 1, \) be a set of variables and let \( D \) be a dataset over \( V \) with \( N \) cases.

Let \( P \) be a joint distribution over the set of all DAGs \( G = (V_G, A_G) \) with node set \( V_G = V \).

The quality of \( G \) given \( D \), notation: \( Q(G, D) \), is defined as

\[
Q(G, D) = \log P(G) - N \cdot H(G, D) - \frac{1}{2} K \cdot \log N
\]

where

\[
H(G, D) = - \sum_{V_i \in V} \sum_{c_{V_i}} \sum_{c_{\rho(V_i)}} \left( \frac{N(c_{V_i} \land c_{\rho(V_i)})}{N} \right) \cdot \log \left( \frac{N(c_{V_i} \land c_{\rho(V_i)})}{N(c_{\rho(V_i)})} \right)
\]

and \( K = \sum_{V_i \in V} 2^{\left| \rho(V_i) \right|} \) for binary-valued variables.
The entropy term $H(G, D)$

Let $V$ and $D$ be as before. Let $\Pr$ be the joint distribution defined by $B = (G, \Gamma)$, where $G = (V_G, A_G)$ is a DAG with $V_G = V$, and $\Gamma$ is obtained from $D$. Then,

$$\log P'(D | B) = \log \prod_{c_V \in D} \Pr(c_V) = \log \prod_{c_V \in D} \prod_{V_i \in V} \gamma(c_{V_i} | c_{\rho(V_i)}) =$$

$$= \log \prod_{V_i \in V} \prod_{c_{V_i}, c_{\rho(V_i)}} \gamma_{V_i}(c_{V_i} | c_{\rho(V_i)})^{N(c_{V_i} \land c_{\rho(V_i)})} =$$

$$= \log \prod_{V_i \in V} \prod_{c_{V_i}, c_{\rho(V_i)}} \left( \frac{N(c_{V_i} \land c_{\rho(V_i)})}{N(c_{\rho(V_i)})} \right)^{N(c_{V_i} \land c_{\rho(V_i)})} =$$

$$= N \cdot \sum_{V_i \in V} \sum_{c_{V_i}} \sum_{c_{\rho(V_i)}} \left( \frac{N(c_{V_i} \land c_{\rho(V_i)})}{N} \right) \cdot \log \left( \frac{N(c_{V_i} \land c_{\rho(V_i)})}{N(c_{\rho(V_i)})} \right) =$$

$$= -N \cdot H(G, D)$$
Computing the quality $Q(G, D)$ of $G$ given $D$: an example

Consider the same dataset $D$ as before and the following graph $G$.

We first compute $-N \cdot H(G, D)$:

For $V_1$:

$$N(v_1) \log \frac{N(v_1)}{N} + N(\neg v_1) \log \frac{N(\neg v_1)}{N} = 9 \cdot \log \frac{9}{15} + 6 \cdot \log \frac{6}{15} = -4.384$$

(if we use the $^{10}\log$ for easy computation)
Computing the quality $Q(G, D)$ of $G$ given $D$: an example

Consider the same dataset $D$ as before and the following graph $G$.

We first compute $-N \cdot H(G, D)$:

For $V_2$:

$$N(v_2 \land v_1) \log \frac{N(v_2 \land v_1)}{N(v_1)} + N(\neg v_2 \land v_1) \log \frac{N(\neg v_2 \land v_1)}{N(v_1)} +$$

$$+ N(v_2 \land \neg v_1) \log \frac{N(v_2 \land \neg v_1)}{N(\neg v_1)} + N(\neg v_2 \land \neg v_1) \log \frac{N(\neg v_2 \land \neg v_1)}{N(\neg v_1)} =$$

$$= 9 \log \frac{9}{9} + 0 + 3 \log \frac{3}{6} + 3 \log \frac{3}{6} = -1.806$$

(if we use the $10 \log$ for easy computation)
Computing the quality $Q(G, D)$ of $G$ given $D$: an example

Consider the same dataset $D$ as before and the following graph $G$.

We first compute $-N \cdot H(G, D)$:

For $V_3$:

$$-N \cdot H(G, D) = -3 \log \frac{3}{9} + 6 \log \frac{6}{9} + 0 + 0 = -2.49$$
Computing the quality $Q(G, D)$ of $G$ given $D$: an example

Consider the same dataset $D$ as before and the following graph $G$.

We first compute $-N \cdot H(G, D)$:

For $V_4$:

\[
N(v_4 \land v_2 \land v_3) \log \frac{N(v_4 \land v_2 \land v_3)}{N(v_2 \land v_3)} + N(\neg v_4 \land v_2 \land v_3) \log \frac{N(\neg v_4 \land v_2 \land v_3)}{N(v_2 \land v_3)}
\]

\[+ N(v_4 \land \neg v_2 \land v_3) \log \frac{N(v_4 \land \neg v_2 \land v_3)}{N(\neg v_2 \land v_3)} + N(\neg v_4 \land \neg v_2 \land v_3) \log \frac{N(\neg v_4 \land \neg v_2 \land v_3)}{N(\neg v_2 \land v_3)}
\]

\[+ N(v_4 \land v_2 \land \neg v_3) \log \frac{N(v_4 \land v_2 \land \neg v_3)}{N(v_2 \land \neg v_3)} + N(\neg v_4 \land v_2 \land \neg v_3) \log \frac{N(\neg v_4 \land v_2 \land \neg v_3)}{N(v_2 \land \neg v_3)}
\]

\[+ N(v_4 \land \neg v_2 \land \neg v_3) \log \frac{N(v_4 \land \neg v_2 \land \neg v_3)}{N(\neg v_2 \land \neg v_3)} + N(\neg v_4 \land \neg v_2 \land \neg v_3) \log \frac{N(\neg v_4 \land \neg v_2 \land \neg v_3)}{N(\neg v_2 \land \neg v_3)}
\]

\[= 0 + 0 + 1 \log \frac{1}{3} + 2 \log \frac{2}{3} + 2 \log \frac{2}{6} + 4 \log \frac{4}{6} + 0 + 0 = -2.488
\]
Computing the quality $Q(G, D)$ of $G$ given $D$: an example

Consider the same dataset $D$ as before and the following graph $G$.

We first compute $-N \cdot H(G, D)$:

$$-N \cdot H(G, D) = -4.384 - 1.806 - 2.488 - 2.488 = -11.167$$

(if we use the $10\log$ for easy computation)
Computing the quality $Q(G, D)$ of $G$ given $D$: an example

Consider the same dataset $D$ as before and the following graph $G$.

We have that

- $-N \cdot H(G, D) = -11.167$
- $-1/2 K \cdot \log N = -1/2 \cdot (1 + 2 + 2 + 4) \cdot \log 15 = -5.292$

Suppose that $P$ is a uniform distribution with $\log P(G) = C$. Then

$$Q(G, D) = C - 16.459$$

What does this mean?
Comparing graphs: an example

Consider the same dataset $D$ as before. Consider the following graphs and their quality with respect to $D$:

1. $V_1 \rightarrow V_2 \rightarrow V_3 \rightarrow V_4 \quad C = 16.459$
2. $V_1 \rightarrow V_2 \rightarrow V_3 \quad C = 17.324$
3. $V_1 \rightarrow V_2 \rightarrow V_3 \quad C = 16.941$
4. $V_1 \rightarrow V_2 \rightarrow V_3 \rightarrow V_4 \quad C = 17.636$

Which of these graphs best captures the joint distribution reflected in the data?
Which graph is best? The interaction among the terms

Reconsider the quality of acyclic digraph $G$ given dataset $D$:

$$Q(G, D) = \log P(G) - N \cdot H(G, D) - \frac{1}{2} K \cdot \log N$$

Assuming uniform $P$, the following interactions exist among the different terms of $Q(G, D)$: NB: $x$-axis captures density of $G$
Finding the best graph: a search procedure

The search procedure of the learning algorithm is a heuristic for finding a DAG with the highest quality given the data.

<table>
<thead>
<tr>
<th>number of nodes</th>
<th>number of acyclic digraphs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>25</td>
</tr>
<tr>
<td>4</td>
<td>543</td>
</tr>
<tr>
<td>5</td>
<td>29,281</td>
</tr>
<tr>
<td>6</td>
<td>3,781,503</td>
</tr>
<tr>
<td>7</td>
<td>1,138,779,265</td>
</tr>
<tr>
<td>8</td>
<td>783,702,329,343</td>
</tr>
<tr>
<td>9</td>
<td>1,213,442,454,842,881</td>
</tr>
<tr>
<td>10</td>
<td>4,175,098,976,430,598,143</td>
</tr>
</tbody>
</table>
B search: the basic idea

The search procedure starts with a graph without arcs to which it adds appropriate arcs:

- compute for every possible arc that can be added, the increase in quality of the graph;
- choose the arc that results in the largest increase in quality and add this arc to the graph.

Repeated until an increase in quality can no longer be achieved.
The B search heuristic

**PROCEDURE** CONSTRUCT-DIGRAPH \((V, D, G)\):

FOR EACH \(V_i \in V\) DO

\[
\rho(V_i) := \emptyset
\]

OD;

REPEAT

FOR EACH PAIR \(V_i, V_j \in V\) SUCH THAT ADDITION OF THE ARC \((V_i, V_j)\) TO \(G\) DOES NOT INTRODUCE A CYCLE DO

\[
diff(V_i, V_j) := q(V_j, \rho(V_j) \cup \{V_i\}, D) - q(V_j, \rho(V_j), D)
\]

OD;

SELECT THE PAIR \(V_i, V_j \in V\) FOR WHICH \(diff(V_i, V_j)\) IS MAXIMAL;

IF \(diff(V_i, V_j) > 0\)

THEN \(\rho(V_j) := \rho(V_j) \cup \{V_i\}\)

FI

UNTIL \(diff(V_i, V_j) \leq 0\).
An example

Consider the same dataset $D$ as before and suppose (!) that the search procedure has constructed the following graph:

For which of the following arcs does the search procedure compute the increase in quality?

- $(V_1, V_2)$
- $(V_2, V_1)$
- $(V_4, V_2)$
- $(V_1, V_4)$
- $(V_4, V_1)$
- $(V_3, V_1)$
- $(V_2, V_3)$
- $(V_3, V_2)$
- $(V_4, V_3)$
The quality of a node

**Definition:** Let $V$, $D$, $N$ and $G$ be as before.

The quality of a node $V_i \in V_G$ given $D$, notation: $q(V_i, \rho(V_i), D)$, is defined as

$$q(V_i, \rho(V_i), D) = \sum_{c_{V_i}} \sum_{c_{\rho(V_i)}} N(c_{V_i} \wedge c_{\rho(V_i)}) \cdot \log \left( \frac{N(c_{V_i} \wedge c_{\rho(V_i)})}{N(c_{\rho(V_i)})} \right)$$

$$- \frac{1}{2} \cdot 2^{|\rho(V_i)|} \cdot \log N$$

**Lemma:** (without proof)

$$Q(G, D) = \log P(G) + \sum_{V_i \in V_G} q(V_i, \rho(V_i), D)$$
An example

Consider the same dataset $D$ as before and suppose (!) that the search procedure has constructed the following graph:

![Graph with vertices V1, V2, V3, V4]

We consider the increase in quality for arc $(V_2, V_3)$:

$$\text{diff}(V_2, V_3) = q(V_3, \{V_1, V_2\}, D) - q(V_3, \{V_1\}, D)$$
An example

Consider the same dataset $D$ as before and suppose (!) that the search procedure has constructed the following graph:

$$q(V_3, \{V_1, V_2\}, D) =$$

$$= N(v_3 \land v_1 \land v_2) \log \frac{N(v_3 \land v_1 \land v_2)}{N(v_1 \land v_2)} + N(\bar{v}_3 \land v_1 \land v_2) \log \frac{N(\bar{v}_3 \land v_1 \land v_2)}{N(v_1 \land v_2)}$$

$$+ N(v_3 \land \bar{v}_1 \land v_2) \log \frac{N(v_3 \land \bar{v}_1 \land v_2)}{N(\bar{v}_1 \land v_2)} + N(\bar{v}_3 \land \bar{v}_1 \land v_2) \log \frac{N(\bar{v}_3 \land \bar{v}_1 \land v_2)}{N(\bar{v}_1 \land v_2)}$$

$$+ N(v_3 \land v_1 \land \bar{v}_2) \log \frac{N(v_3 \land v_1 \land \bar{v}_2)}{N(v_1 \land \bar{v}_2)} + N(\bar{v}_3 \land v_1 \land \bar{v}_2) \log \frac{N(\bar{v}_3 \land v_1 \land \bar{v}_2)}{N(v_1 \land \bar{v}_2)}$$

$$+ N(v_3 \land \bar{v}_1 \land \bar{v}_2) \log \frac{N(v_3 \land \bar{v}_1 \land \bar{v}_2)}{N(\bar{v}_1 \land \bar{v}_2)} + N(\bar{v}_3 \land \bar{v}_1 \land \bar{v}_2) \log \frac{N(\bar{v}_3 \land \bar{v}_1 \land \bar{v}_2)}{N(\bar{v}_1 \land \bar{v}_2)}$$

$$- \frac{1}{2} \cdot 4 \log N = -4.84$$
An example

Consider the same dataset $D$ as before and suppose (!) that the search procedure has constructed the following graph:

$$q(V_3, \{V_1\}, D) =$$
$$= N(v_3 \land v_1) \log \frac{N(v_3 \land v_1)}{N(v_1)} + N(\overline{v}_3 \land v_1) \log \frac{N(\overline{v}_3 \land v_1)}{N(v_1)}$$
$$+ N(v_3 \land \overline{v}_1) \log \frac{N(v_3 \land \overline{v}_1)}{N(\overline{v}_1)} + N(\overline{v}_3 \land \overline{v}_1) \log \frac{N(\overline{v}_3 \land \overline{v}_1)}{N(\overline{v}_1)}$$
$$- \frac{1}{2} \cdot 2 \log N = -3.66$$
An example

Consider the same dataset $D$ as before and suppose (!) that the search procedure has constructed the following graph:

We consider the increase in quality for arc $(V_2, V_3)$:

$$\text{diff}(V_2, V_3) = q(V_3, \{V_1, V_2\}, D) - q(V_3, \{V_1\}, D)$$

$$= -4.84 - -3.66 = -1.18$$

The increase in quality for arc $(V_2, V_3)$ is negative; will the arc be selected by the search procedure?
An example

Consider the same dataset $D$ as before and suppose (!) that the search procedure has constructed the following graph:

We consider the increase in quality for the arc $(V_1, V_2)$:

$$\text{diff}(V_1, V_2) = q(V_2, \{V_1\}, D) - q(V_2, \emptyset, D)$$
An example

Consider the same dataset $D$ as before and suppose (!) that the search procedure has constructed the following graph:

![Graph Diagram]

$q(V_2, \{V_1\}, D) =
= N(v_2 \wedge v_1) \log \frac{N(v_2 \wedge v_1)}{N(v_1)} + N(\overline{v}_2 \wedge v_1) \log \frac{N(\overline{v}_2 \wedge v_1)}{N(v_1)}
+ N(v_2 \wedge \overline{v}_1) \log \frac{N(v_2 \wedge \overline{v}_1)}{N(\overline{v}_1)} + N(\overline{v}_2 \wedge \overline{v}_1) \log \frac{N(\overline{v}_2 \wedge \overline{v}_1)}{N(\overline{v}_1)}
- \frac{1}{2} \cdot 2 \cdot \log N = -2.98$

$q(V_2, \emptyset, D) =
= N(v_2) \log \frac{N(v_2)}{N} + N(\overline{v}_2) \log \frac{N(\overline{v}_2)}{N} - \frac{1}{2} \cdot \log N
= -3.85$
An example

Consider the same dataset $D$ as before and suppose (!) that the search procedure has constructed the following graph:

We consider the increase in quality for the arc $(V_1, V_2)$:

$$
\text{diff}(V_1, V_2) = q(V_2, \{V_1\}, D) - q(V_2, \emptyset, D)
$$

$$
= -2.98 - (-3.85) = 0.87
$$

The increase in quality for arc $(V_1, V_2)$ is positive; will the arc be selected by the search procedure?
Evaluation

Is the presented metric algorithm any good?

- our example dataset $D$ was generated from the following network:

  $\gamma(v_1) = 0.8$
  $\gamma(v_2 | v_1) = 0.9$
  $\gamma(v_2 | \neg v_1) = 0.3$
  $\gamma(v_3 | v_1) = 0.2$
  $\gamma(v_3 | \neg v_1) = 0.6$
  $\gamma(v_4 | v_2 \land v_3) = 0.1$
  $\gamma(v_4 | \neg v_2 \land v_3) = 0.2$
  $\gamma(v_4 | v_2 \land \neg v_3) = 0.6$
  $\gamma(v_4 | \neg v_2 \land \neg v_3) = 0.1$

- the MDL score is asymptotically correct: for best MDL-scoring $B$, $Pr_B$ will be arbitrarily close to the sampled distribution, given sufficient independent samples.
A learning algorithm can be used to obtain an initial graph, which is then refined with the help of a domain expert.

A learning algorithm can be used to construct parts of the graph of a Bayesian network.

There exist less greedy variants of the algorithm discussed.
Some remarks (2)

When learning networks of general topology is infeasible, it can be restricted to classes of networks with restricted topology, such as

- Naive Bayes classifiers
- TAN and FAN classifiers
- ...

Learning then typically involves feature selection and is often accuracy-based (supervised). Discriminative learning is preferred (optimisation of $\Pr(C \mid F)$ rather than $\Pr(CF)$) but expensive.
Sources of probabilistic information

In most domains of application, probabilistic information is available from different sources:

- (statistical) data;
- literature;
- domain experts.

In practice, domain experts will often have to provide the majority of the probabilities required.
Data

Retrospective data do not always provide for assessing the probabilities required for a Bayesian network:

• the collection strategies used may have biased the data;
• the recorded variables and values may not match the variables and values of the network;
• the data may include missing values;
• the data collection may be insufficiently large;
• ...
Literature

Probabilistic information from the literature seldom provides for assessing the required probabilities:

- the background of the information is not given;
- the information is only partially specified;
- the reported probabilities pertain to variables that are not directly related in the network;
- the information is non-numerical;
- ...
Reducing the burden

Contemporary Bayesian networks comprise tens or hundreds of variables, requiring thousands of probabilities:

- changes to the
  - definitions of the variables and values;
  - graphical structure;
  may help reduce the number of probabilities that are required;

- the use of
  - domain models;
  - parametric probability distributions;
  may help reduce the number of probabilities that are to be assessed.
Consider building a Bayesian network for Wilson’s disease, a recessively inherited disease of the liver:

From the disease being recessively inherited, we have for the variable ‘Wilson’s disease’ that

\[ Pr(d_1 \mid g_1) = 1 \quad Pr(d_2 \mid g_1) = 0 \]
\[ Pr(d_1 \mid g_2) = 0 \quad Pr(d_2 \mid g_2) = 1 \]
\[ Pr(d_1 \mid g_3) = 0 \quad Pr(d_2 \mid g_3) = 1 \]
The use of domain models: the example continued

Consider the node ‘Wilson’s disease genotype’. By Mendel’s law:

\[
\Pr(g_1) = \Pr(g_1) \cdot \Pr(g_1) + \frac{1}{2} \cdot 2 \cdot \Pr(g_1) \cdot \Pr(g_2) + \frac{1}{4} \cdot \Pr(g_2) \cdot \Pr(g_2)
\]

With \(\Pr(d_1) = \Pr(g_1) = 0.005\), we now find

\[
\Pr(g_1) = 0.005, \quad \Pr(g_2) = 0.131, \quad \text{and} \quad \Pr(g_3) = 0.864
\]
Another example

Consider the following graph of a Bayesian network:

\[
\text{Flu} \rightarrow \{\text{no, mild, severe}\} \\
\text{Fever} \rightarrow \{37, 38, 39, 40, 41\}
\]

The assessments for the variable \(\text{Fever}\), e.g.
\[\Pr(\text{Fever} \mid \text{Flu} = \text{severe})\] can be obtained from:

\[
\begin{array}{cccccc}
37 & 38 & 39 & 40 & 41 \\
\text{Flu} & = & \text{no} & | & \text{mild} & | & \text{severe}
\end{array}
\]
The use of a parametric approach

Consider the following causal mechanism:

The node $\text{Alarm}$ requires the following probabilities:

$$
\begin{align*}
\Pr(\text{alarm} \mid \neg \text{burglar} \land \neg \text{earthq}. ) & \quad \Pr(\text{alarm} \mid \text{burglar} \land \neg \text{earthq}. ) \\
\Pr(\text{alarm} \mid \neg \text{burglar} \land \text{earthq}. ) & \quad \Pr(\text{alarm} \mid \text{burglar} \land \text{earthq}. )
\end{align*}
$$

The underlying mechanisms that cause the alarm have ‘nothing to do with each other’ → hard to assess probabilities in a straightforward manner.

A parametric approach requires just two assessments and provides rules for computing the other ones.
Disjunctive interaction, informally

Consider the following causal mechanism:

The variables $V_1, \ldots, V_m$, $m \geq 2$, exhibit a disjunctive interaction with respect to variable $V_0$ if, for $i = 1, \ldots, m$, we have that:

- $V_i = true$ causes $V_0 = true$, with some (non-zero) probability;
- the probability with which $V_i = true$ causes $V_0 = true$ does not diminish due to the presence or absence of any other causes.

The parametric distribution to describe a causal mechanism with a disjunctive interaction is called a noisy-or gate.
Disjunctive interaction, continued

The semantics of a disjunctive interaction can be depicted as
Disjunctive interaction, more formally

Consider the following causal mechanism:

The variables $V_1, \ldots, V_m, m \geq 2$, exhibit a disjunctive interaction with respect to the variable $V_0$ iff the following properties hold:

- **accountability**: there are no other causes for $V_0 = true$ than the modelled causes $V_1 = true, \ldots, V_m = true$, that is,
  \[ \Pr(v_0 \mid \neg v_1 \land \ldots \land \neg v_m) = 0 \]

- **exception independence**:  
  1) for each $V_i$, an inhibitor $I_i$ can be defined such that
     \[ \Pr(v_0 \mid \neg v_1 \land \ldots \land \neg v_{i-1} \land (v_i \land i_i) \land \neg v_{i+1} \land \ldots \land \neg v_m) = 0 \]
     \[ \Pr(v_0 \mid \neg v_1 \land \ldots \land \neg v_{i-1} \land (v_i \land \neg i_i) \land \neg v_{i+1} \land \ldots \land \neg v_m) = 1 \]
  2) the inhibitors $I_i$ are mutually independent.
• the variable $I_b$ describes a combination of
  – the skill of the burglar, and . . .
• the variable $I_e$ describes a combination of
  – the type of earthquake, and . . .
• the variables $I_b$ and $I_e$ do not describe
  – a power failure, or . . .

Does this causal mechanism represent a disjunctive interaction?
For the variable $V_0$, the noisy-or gate specifies:

- using the property of accountability:
  \[ \Pr(v_0 \mid \neg v_1 \land \ldots \land \neg v_m) = 0 \]

- using the property of exception independence:
  \[
  \Pr(v_0 \mid \neg v_1 \land \ldots \land \neg v_{i-1} \land v_i \land \neg v_{i+1} \land \ldots \land \neg v_m) = 1 - q_i^a
  \]
  where \( \Pr(i_i) = q_i^a \) for inhibitor \( I_i \) of \( V_i \);
  
  - for each configuration \( c \) of \( \{V_1, \ldots, V_m\} \) with
    \[
    T_c = \{i \mid c \text{ contains } v_i\}, \quad T_c \neq \emptyset: \quad \Pr(v_0 \mid c) = 1 - \prod_{i \in T_c} q_i^a
    \]

For variable $V_0$ only $m$ probabilities have to be assessed.
An example noisy-or gate

For the variable *Late season growth*, the following probabilities are assessed:

\[
\begin{align*}
\Pr(lsg \mid lp \land \neg lf \land \neg wf) &= 0.8 & \Pr(i_{lp}) &= 0.2 \\
\Pr(lsg \mid \neg lp \land lf \land \neg wf) &= 0.8 & \Rightarrow & \Pr(i_{lf}) &= 0.2 \\
\Pr(lsg \mid \neg lp \land \neg lf \land wf) &= 0.6 & \Pr(i_{wf}) &= 0.4
\end{align*}
\]
An example noisy-or gate

\[
\begin{align*}
\Pr(lsg \mid lp \land \neg lf \land \neg wf) &= 0.8 & \Pr(i_{lp}) &= 0.2 \\
\Pr(lsg \mid \neg lp \land lf \land \neg wf) &= 0.8 \implies \Pr(i_{lf}) = 0.2 \\
\Pr(lsg \mid \neg lp \land \neg lf \land wf) &= 0.6 & \Pr(i_{wf}) &= 0.4
\end{align*}
\]

We then compute, for example,

\[
\Pr(lsg \mid lp \land lf \land \neg wf) = 1 - \Pr(i_{lp}) \cdot \Pr(i_{lf}) = 1 - 0.2 \cdot 0.2 = 0.96
\]

<table>
<thead>
<tr>
<th>Late pruning</th>
<th>false</th>
<th>true</th>
</tr>
</thead>
<tbody>
<tr>
<td>Late fertilisation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>false</td>
<td>0</td>
<td>0.8</td>
</tr>
<tr>
<td>true</td>
<td>0.6</td>
<td>0.92</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Warm fall</th>
<th>false</th>
<th>true</th>
</tr>
</thead>
<tbody>
<tr>
<td>false</td>
<td>0</td>
<td>0.8</td>
</tr>
<tr>
<td>true</td>
<td>0.6</td>
<td>0.92</td>
</tr>
</tbody>
</table>
The example continued

Now compare:

- the probabilities obtained from the noisy-or gate:

<table>
<thead>
<tr>
<th>Late pruning</th>
<th>Late fertilisation</th>
<th>false</th>
<th>true</th>
<th>false</th>
<th>true</th>
</tr>
</thead>
<tbody>
<tr>
<td>Warm fall</td>
<td>false</td>
<td>0</td>
<td>0.8</td>
<td>0.8</td>
<td>0.96</td>
</tr>
<tr>
<td></td>
<td>true</td>
<td>0.6</td>
<td>0.92</td>
<td>0.92</td>
<td>0.98</td>
</tr>
</tbody>
</table>

- the probabilities assessed by domain experts:

<table>
<thead>
<tr>
<th>Late pruning</th>
<th>Late fertilisation</th>
<th>false</th>
<th>true</th>
<th>false</th>
<th>true</th>
</tr>
</thead>
<tbody>
<tr>
<td>Warm fall</td>
<td>false</td>
<td>0.1</td>
<td>0.8</td>
<td>0.8</td>
<td>0.9</td>
</tr>
<tr>
<td></td>
<td>true</td>
<td>0.6</td>
<td>0.9</td>
<td>0.9</td>
<td>1.0</td>
</tr>
</tbody>
</table>
If accountability is violated

Suppose that exception independence holds, but accountability does not, that is,

\[ \Pr(v_0 \mid \neg v_1 \land \ldots \land \neg v_m) = p \quad \text{with} \quad p > 0 \]

- the noisy-or gate can be applied after including an additional parent \( V_{m+1} \) of \( V_0 \) with

\[
\begin{align*}
\Pr(v_0 \mid \neg v_1 \land \ldots \land \neg v_m \land \neg v_{m+1}) & = 0 \\
\Pr(v_0 \mid \neg v_1 \land \ldots \land \neg v_m \land v_{m+1}) & = p
\end{align*}
\]

- the leaky noisy-or gate can be used.
The leaky noisy-or gate

Consider the following causal mechanism with exception independence:

\[ V_1 \rightarrow \ldots \rightarrow V_m \rightarrow V_0 \]

Suppose that \( \Pr(v_0 | \neg v_1 \land \ldots \land \neg v_m) = p \), where \( p = 1 - q_0 > 0 \) is the leak probability. The leaky noisy-or gate specifies for \( V_0 \):

- \( \Pr(v_0 | \neg v_1 \land \ldots \land \neg v_m) = p \);
- \( \Pr(v_0 | \neg v_1 \land \ldots \land \neg v_{i-1} \land v_i \land \neg v_{i+1} \land \ldots \land \neg v_m) = 1 - q_i^l \)
  where \( \Pr(i_i) = q_i^l = q_0 \cdot q_i^a \) for inhibitor \( I_i \) of \( V_i \);
- for each configuration \( c \) with \( T_c \neq \emptyset \), we have

\[
\Pr(v_0 | c) = 1 - q_0 \cdot \prod_{i \in T_c} q_i^a = 1 - q_0 \cdot \prod_{i \in T_c} \left( \frac{q_i^l}{q_0} \right)
\]

For variable \( V_0 \) only \( m + 1 \) probabilities need to be assessed.
An example leaky noisy-or gate

Reconsider the late-pruning example:

\[
\begin{align*}
\Pr(lsg \mid lp \land \neg lf \land \neg wf) &= 0.8 & \Pr(i_{lp}) &= 0.2 \\
\Pr(lsg \mid \neg lp \land lf \land \neg wf) &= 0.8 & \Rightarrow \Pr(i_{lf}) &= 0.2 \\
\Pr(lsg \mid \neg lp \land \neg lf \land wf) &= 0.6 & \Pr(i_{wf}) &= 0.4
\end{align*}
\]

With a leak probability \( \Pr(lsg \mid \neg lp \land \neg lf \land \neg wf) = 0.1 \), i.e. \( q_0 = 0.9 \), we compute

<table>
<thead>
<tr>
<th>Late pruning</th>
<th>false</th>
<th>true</th>
<th>false</th>
<th>true</th>
</tr>
</thead>
<tbody>
<tr>
<td>Late fertilisation</td>
<td>false</td>
<td>0.1</td>
<td>0.8</td>
<td>0.8</td>
</tr>
<tr>
<td>Warm fall</td>
<td>false</td>
<td>0.6</td>
<td>0.91</td>
<td>0.91</td>
</tr>
</tbody>
</table>
Subjective probabilities

Probability assessment often requires the help of domain experts → assessments are based upon personal knowledge and experience, i.e. subjective.

This can result in a number of problems:

• assessments are incoherent\(^2\):
  \[ \Pr(a) < \Pr(a \land b); \]
  \[ \Pr(a) > \Pr(b) \text{ and yet } \Pr(a \mid b) < \Pr(b \mid a). \]

• assessments are biased as a result of various psychological factors, and therefore not uncalibrated\(^3\);

• the domain expert is not capable of expressing his knowledge and experience in terms of numbers.

---

\(^2\)assessments do not adhere to the postulates of probability theory

\(^3\)assessments do not reflect true frequencies
Overconfidence and underconfidence

- **overconfident assessor**: compared with true frequencies, assessments show a **tendency towards** the extremes;
- **underconfident assessor**: compared with true frequencies, assessments show a **tendency away** from the extremes.
Heuristics

Upon assessing probabilities for a certain outcome, people tend to use simple cognitive heuristics:

- **representativeness**: the assessment is based upon the similarity with a stereotype outcome;
- **availability**: the assessment is based upon the ease with which similar outcomes are recalled;
- **anchoring-and-adjusting**: the probability is assessed by adjusting an initially chosen anchor probability:
Pitfalls

Using the representativeness heuristic can introduce biases:

- prior probabilities, or base rates, are insufficiently taken into account;
- assessments are based upon insufficient samples;
- weights of the characteristics of the stereotype outcome are insufficiently taken into consideration;
- ...
Pitfalls — cntd.

Using the availability heuristic can introduce biases:

- the ease of recall from memory is influenced by
  - recency, rareness, and the past consequences for the assessor;
  - external stimuli:

Example
Pitfalls — cntd.

Using the anchoring-and-adjusting heuristic can introduce biases:

- the assessor does not choose an appropriate anchor;
- the assessor does not adjust the anchor to a sufficient extent:

Example

\[ \text{Geometric Mean Response} \]

- ...
Probability assessment tools

For eliciting probabilities from experts, various tools are available from the field of decision analysis:

- probability wheels;
- betting models;
- lottery models;
- probability scales.
A probability wheel is composed of two coloured faces and a hand:

The expert is asked to adjust the area of the red face so that the probability of the hand stopping there, equals the probability of interest.
For their new soda, an expert from Colaco is asked to assess the probability $\Pr(n)$ of a national success:

- the expert is offered two bets:

  - $d$: national success with $x$ euro, national failure with $-y$ euro
  - $\bar{d}$: national success with $-x$ euro, national failure with $y$ euro

- if the expert is indifferent between $d$ and $\bar{d}$, then

\[
x \cdot \Pr(n) - y \cdot (1 - \Pr(n)) = y \cdot (1 - \Pr(n)) - x \cdot \Pr(n)
\]

from which we find $\Pr(n) = \frac{y}{x + y}$. 

\[
\frac{99x264}{368.5x283.5}
\]
Lottery models — an example

For their new soda, an expert from Colaco is asked to assess the probability $\Pr(n)$ of a national success:

- the expert is offered two lotteries:

  - if the expert is indifferent between $d$ and $\bar{d}$, then $\Pr(n) = p(\text{outcome})$. 

\[ \begin{align*}
  d & \quad \text{national success} \\
  & \quad \text{Hawaiian trip} \\
 & \quad \text{national failure} \\
  \bar{d} & \quad p(\text{outcome}) \\
  & \quad \text{Hawaiian trip} \\
  & \quad p(\text{not outcome}) \\
  & \quad \text{chocolate bar}
\end{align*} \]
Obtaining many probabilities in little time: a tool

- probabilities are represented by fragments of text;
- each probability is accompanied by a verbal-numerical scale;
- probabilities are grouped to ensure consistency.

<table>
<thead>
<tr>
<th>Conjunctivitis</th>
<th>Mucositis (1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consider a pig <em>without an infection of the mucous</em>. How likely is it that this pig shows a <em>conjunctivitis</em>?</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Certain (almost)</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Probable</td>
<td>85</td>
</tr>
<tr>
<td>Expected</td>
<td>75</td>
</tr>
<tr>
<td>Fifty-fifty</td>
<td>50</td>
</tr>
<tr>
<td>Uncertain</td>
<td>25</td>
</tr>
<tr>
<td>Improbable</td>
<td>15</td>
</tr>
<tr>
<td>(Almost) Impossible</td>
<td>0</td>
</tr>
</tbody>
</table>
An iterative procedure for probability assessment

Repeat iteratively until satisfactory behaviour of the network is attained:

- obtain initial probability assessments;
- investigate, for each probability, whether or not the output is sensitive to its assessment;
- investigate, for each sensitive probability, whether or not its assessment can be cost-effectively improved upon.
Chapter 6:

Bringing Bayesian Networks into Practice
Inaccuracy versus robustness

Consider a Bayesian network \( B = (G, \Gamma) \). Assessments obtained (from data or human experts) for the parameter probabilities \( \gamma_V \in \Gamma \) tend to be inaccurate or uncertain.

Robustness: pertains to stability of some output in terms of variation of parameter probabilities:

- output is robust if varying parameters reveals little effect on the output;
- if varying parameters shows a considerable effect, then the output is not robust and may be unreliable.

Inaccuracy, therefore, does not necessarily imply a lack of robustness.
Analysing the robustness of a Bayesian network

Various techniques are available for analysing the robustness of a Bayesian network.

- **sensitivity analysis**
  - systematically vary parameters and study the effect on the output;
  - in an \( n \)-way sensitivity analysis, \( n \) parameters are varied simultaneously;

- **uncertainty analysis**
  - repeatedly draw parameters from sample distributions and study the effect.
A one-way sensitivity analysis

A one-way sensitivity analysis for a parameter probability
$$x = \gamma(c_{V_i} \mid c_{\rho(V_i)})$$
results in a sensitivity curve, describing an
output probability
$$y = \Pr(c_{V_o} \mid c_E)$$
in terms of $$x$$:

The effect of small variations in $$x$$ on the output depends on the
original assessment $$x_0$$ for parameter probability $$x$$. 
The computational burden involved

Straightforward sensitivity analysis is highly time consuming:

- for the following network, a single analysis\(^4\) requires 130 network propagations:

\[
\begin{align*}
\gamma(b \mid mc) &= 0.20 & \gamma(mc) &= 0.20 \\
\gamma(b \mid \neg mc) &= 0.05 \\
\gamma(sh \mid b) &= 0.80 & \gamma(c \mid b, isc) &= 0.80 \\
\gamma(sh \mid \neg b) &= 0.60 & \gamma(c \mid \neg b, isc) &= 0.80 \\
\gamma(ct \mid b) &= 0.95 \\
\gamma(ct \mid \neg b) &= 0.10 & \gamma isc \mid mc) &= 0.80 \\
\gamma(isc \mid \neg mc) &= 0.20
\end{align*}
\]

- for the medium-sized classical swine fever network, a single analysis requires approximately 20,000 network propagations.

\(^4\)assuming we compute 10 points per curve
Reducing the computational burden

The computational burden of a sensitivity analysis can be reduced by exploiting the following Bayesian network properties:

- various parameter probabilities cannot affect, upon variation, the output probability of the network;
- the output probability relates to any parameter under study as a quotient of two (multi-)linear functions.
Influential parameters – the basics

Consider a Bayesian network $\mathcal{B} = (G, \Gamma)$ with output variable of interest $V_o \in V_G$ and evidence for the set $E \subseteq V_G$.

Let $S^E(V_o) \subseteq V_G$ denote the set of variables whose parameters may affect, upon variation, the output distribution of interest $\Pr^e(V_o)$.

Which $V_i \in V_G$ belong to $S^E(V_o)$?

Basically: each $V_i$ for which a change in one of its parameters $\gamma(c_{V_i} \mid c_{\rho(V_i)})$ will eventually result in a change in the messages computed for/at $V_o$ upon inference.

$S^E(V_o)$ is called the sensitivity set for $V_o$ under evidence for $E$. 
(Un)influential parameters – introduction

Let $B, V_o, E$, and $S^E(V_o)$ be as before.

Let $U^E(V_o) = V_G \setminus S^E(V_o)$ capture the variables for which a change in a parameter will certainly not affect $Pr^e(V_o)$, i.e. the uninfluential ones.

- Suppose $E = \emptyset$. Which $V_i \in V_G$ belong to $S^\emptyset(V_o)$ and $U^\emptyset(V_o)$?
- Suppose $E \neq \emptyset$. How can $V_i \in S^\emptyset(V_o)$ become uninfluential?
Uninfluential parameters: ancestors

Let $B$, $V_o$ and $E$ be as before.

The parameter probabilities for any variable $V_i$ with

$$V_i \in \rho^*(V_o) \text{ and } \langle \{V_i\} \cup \rho(V_i) \mid E \mid \{V_o\}\rangle^d$$

are uninfluential.

Example:

- the parameters for $MC$ and $B$ cannot affect the output probability $\Pr(sh \mid \neg b)$;
- the parameters for $B$ may affect the output probability $\Pr(c \mid \neg b)$. 
(Un)influential parameters – introduction cntd

Let $\mathcal{B}$, $V_o$, $E$, $S^E(V_o)$ and $U^E(V_o)$ be as before.

- Suppose $E = \emptyset$. Then
  $$S^\emptyset(V_o) = \rho^*(V_o) \text{ and } U^\emptyset(V_o) = \{V_i \mid V_i \not\in \rho^*(V_o)\}$$

- Suppose $E \neq \emptyset$. Then
  $$S^\emptyset(V_o) \cap U^E(V_o) = \{V_i \mid V_i \in \rho^*(V_o) \land \langle\{V_i\} \cup \rho(V_i) \mid E \mid \{V_o\}\rangle^d\}$$

- Suppose $E \neq \emptyset$. Which $V_i \in U^\emptyset(V_o)$ remain uninfluential?
Uninfluential parameters: non-ancestors without evidence for descendants

Let $B$, $V_o$ and $E$ be as before.

The parameter probabilities for any variable $V_i$ with

$$V_i \not\in \rho^*(V_o) \text{ and } \sigma^*(V_i) \cap E = \emptyset$$

are uninfluential.

Example:

- the parameters for $SH$ and $CT$ cannot affect the output probability $\Pr(c \mid \neg isc)$ for the variable of interest $C$;
- the parameters for $SH$ may affect the output probability $\Pr(c \mid sh)$.
(Un)influential parameters – introduction cntd

Let $B$, $V_o$, $E$, $S^E(V_o)$ and $U^E(V_o)$ be as before.

- Suppose $E = \emptyset$. Then 
  
  $S^\emptyset(V_o) = \rho^*(V_o)$ and $U^\emptyset(V_o) = \{V_i \mid V_i \not\in \rho^*(V_o)\}$

- Suppose $E \neq \emptyset$. Then 
  
  $S^\emptyset(V_o) \cap U^E(V_o) = \{V_i \mid V_i \in \rho^*(V_o) \land \langle \{V_i\} \cup \rho(V_i) \mid E \mid \{V_o\} \rangle^d\}$

- Suppose $E \neq \emptyset$. Then 
  
  $U^\emptyset(V_o) \cap U^E(V_o) \supseteq \{V_i \mid V_i \not\in \rho^*(V_o) \land \sigma^*(V_i) \cap E = \emptyset\}$

- Suppose $E \cap \sigma^*(V_i) \neq \emptyset$. Which $V_i$ remain in $U^\emptyset(V_o) \cap U^E(V_o)$?
Uninfluential parameters: non-ancestors with evidence for descendants

Let $B$, $V_o$ and $E$ be as before.

The parameter probabilities for any variable $V_i$ with

$$V_i \notin \rho^*(V_o), \left\langle \{V_i\} \cup \rho(V_i) \mid E \mid \{V_o\} \right\rangle^d$$

and $\sigma^*(V_i) \cap E \neq \emptyset$ are uninfluential.

Example:

- the parameters for $B$ cannot affect the output $\Pr(isc \mid mc \land \neg ct)$;
- the parameters for $B$ may affect the output probability $\Pr(isc \mid \neg ct)$.
The sensitivity set – definition

Let $\mathcal{B}$, $V_o$ and $E$ be as before.

The sensitivity set $S^E(V_o)$ is the set of variables $V_i$ for which none of the following holds:

- $V_i \in \rho^*(V_o)$ and $\langle \{V_i\} \cup \rho(V_i) \mid E \mid \{V_o\} \rangle^d$;
- $V_i \notin \rho^*(V_o)$ and $\sigma^*(V_i) \cap E = \emptyset$;
- $V_i \notin \rho^*(V_o)$, $\langle \{V_i\} \cup \rho(V_i) \mid E \mid \{V_o\} \rangle^d$ and $\sigma^*(V_i) \cap E \neq \emptyset$;

Only the parameters for the variables in the sensitivity set may affect, upon variation, the network’s output probability.
Example: the prior sensitivity set for variable $Stage$

The sensitivity set $S^0(Stage)$ in the prior network consists of 6 variables, together specifying 206 parameters.
The sensitivity set $S^E(\text{Stage})$ in this posterior network consists of 21 variables, together specifying 527 parameters.
Computing the sensitivity set (I)

Let $\mathcal{B}$, $V_o$ and $E$ be as before.

The sensitivity set $S^E(V_o)$ is identified as follows:

- construct, from the network’s digraph $G$, a new digraph $G^*$ by adding an auxiliary parent $X_i$ to every $V_i \in V_G$;
- determine all nodes $V_i$ for which $\neg \langle \{X_i\} \mid E \mid \{V_o\}\rangle^d_{G^*}$; these constitute the sensitivity set.

The sensitivity set can thus be identified in polynomial time $O(|A_{G^*}|)$ from just graphical considerations.
Computing the sensitivity set (II)

An alternative to identifying the sensitivity set $S^E(V_o)$ is to use Bayes-Ball (BB) output (see Shachter, UAI 1998 for details):

**BB terminology:**
- top mark, $N_p(T, E)$,
- 'Requisite $p()$'

$S^E(T) = N_p$

BB can also output
- 'Requisite $e'$ $(E \setminus \text{IrrEv})$ and
- 'Irrelevant' $(E \cup \text{DSep})$

The sensitivity set can be identified in $O(|V_G| + |A_G|)$ from just graphical considerations.
Computing an example sensitivity set

Consider the following digraph of a Bayesian network.

Assume that the graph is extended with auxiliary parents \(X_{CT}, X_{SH}, X_C, X_B, X_{ISC},\) and \(X_{MC}\).

- the sensitivity set for \(ISC\) given \(MC\) and \(CT\) equals \(\{ISC\}\);
- the sensitivity set for \(C\) given \(MC\) and \(CT\) equals \(\{B, CT, C, ISC\}\).
An introduction to the sensitivity function

In a sensitivity analysis, the output probability of interest is a function of the parameter probability under study:
An example sensitivity function

A sensitivity function is strongly constrained by the independences portrayed in the digraph of the network.

Consider the following Bayesian network:

\[
\begin{align*}
\gamma(b \mid mc) &= 0.20 \quad \gamma(mc) = 0.20 \\
\gamma(b \mid \neg mc) &= 0.05 \\
\gamma(sh \mid b) &= 0.80 \quad \gamma(c \mid b, isc) = 0.80 \\
\gamma(sh \mid \neg b) &= 0.60 \quad \gamma(c \mid b, \neg isc) = 0.80 \\
\gamma(ct \mid b) &= 0.95 \\
\gamma(ct \mid \neg b) &= 0.10 \quad \gamma(isc \mid mc) = 0.80 \\
\gamma(isc \mid \neg mc) &= 0.20
\end{align*}
\]

The output probability \( \Pr(\neg mc \land \neg b \land \neg isc \land c) \), written as a function of the parameter \( x = \gamma(c \mid \neg b \land \neg isc) \), equals

\[
\begin{align*}
\Pr(\neg mc \land \neg b \land \neg isc \land c)(x) &= \\
&= \Pr(\neg mc) \cdot \Pr(\neg b \mid \neg mc) \cdot \Pr(\neg isc \mid \neg mc) \cdot \Pr(c \mid \neg b \land \neg isc)(x) \\
&= 0.80 \cdot 0.95 \cdot 0.80 \cdot x = 0.61 \cdot x
\end{align*}
\]
The (one-way) sensitivity function: in general

Consider a sensitivity analysis of a Bayesian network $\mathcal{B} = (G, \Gamma)$ with output variable of interest $V_o$ and evidence for the set $E$.

Consider an arbitrary parameter $x$ from $\Gamma$. Then,

- the output probability of interest equals

$$\Pr(v_o \mid e)(x) = \frac{\Pr(v_o \land e)(x)}{\Pr(e)(x)} = \frac{a \cdot x + b}{c \cdot x + d}$$

where $a$, $b$, $c$, and $d$ are constants;

- if $c \neq 0$ is guaranteed, i.e. $\Pr(e)$ actually varies with $x$, then in essence only three constants are required:

$$\Pr(v_o \mid e)(x) = \frac{a/c \cdot x + b/c}{c/c \cdot x + d/c}$$

- The sensitivity function takes the form of (a fragment of) a rectangular hyperbola.
The (one-way) sensitivity function: specific case

Let $\mathcal{B}$, $V_o$ and $E$ be as before.

Consider an arbitrary parameter probability $x$ from $\Gamma$. Then,

- if $x = \gamma(c_{V_i} \mid c_{\rho(V_i)})$ is associated with a $V_i \in V_G$ for which $\sigma^*(V_i) \cap E = \emptyset$, then the output probability of interest equals

\[
\Pr(v_o \mid e)(x) = a \cdot x + b
\]

where $a$ and $b$ are constants.

- The sensitivity function is linear.
- Note that this always holds in a prior network without evidence.
Proportional scaling of parameters

Upon varying a single parameter \( x = \gamma(v_i \mid \rho) \) for a variable \( V \), the other parameters \( \gamma(v_j \mid \rho) \), \( j \neq i \), for \( V \) are co-varied:

\[
\gamma(v_j \mid \rho)(x) = \begin{cases} 
  x & \text{if } j = i \\
  \gamma(v_j \mid \rho) \cdot \frac{1 - x}{1 - \gamma(v_i \mid \rho)} & \text{otherwise}
\end{cases}
\]

The scheme of proportional scaling keeps the proportions between the parameters \( \gamma(v_j \mid \rho) \), \( j \neq i \), constant.

The scheme results in the smallest distance\(^5\) between the original and the new distribution.

\(^5\)Chan & Darwiche (2003): A distance measure for bounding probabilistic belief change
Computing the sensitivity function $f(x)$

Building upon its general form, it suffices to compute the constants of a sensitivity function:

- a simple algorithm computes the output probability for a small number of values of the parameter under study and solves the resulting system of equations;
- a more intricate algorithm establishes the constants in the function analytically through propagation;
- observing the relation between the constants and derivatives of $f(x)$, we can also use a differential approach\footnote{Darwiche (2000): A differential approach to inference in Bayesian networks.}
Computing an example sensitivity function (1)

Consider once again the following Bayesian network:

```
\begin{align*}
\gamma(b \mid mc) &= 0.20 & \gamma(mc) &= 0.20 \\
\gamma(b \mid \neg mc) &= 0.05 & \gamma(c \mid b, isc) &= 0.80 \\
\gamma(sh \mid b) &= 0.80 & \gamma(c \mid \neg b, isc) &= 0.80 \\
\gamma(sh \mid \neg b) &= 0.60 & \gamma(c \mid b, \neg isc) &= 0.80 \\
\gamma(ct \mid b) &= 0.95 & \gamma(c \mid \neg b, \neg isc) &= 0.05 \\
\gamma(ct \mid \neg b) &= 0.10 & \gamma(isc \mid mc) &= x \\
\gamma(isc \mid \neg mc) &= 0.20
\end{align*}
```

Compute the sensitivity function for output probability $\Pr(mc \mid isc)$ as a function of $x = \gamma(isc \mid mc)$:

- we first compute the output probability from the network three (max four) times, for different values of $x$

For example, for $x = 0.2$, $x = 0.5$ and $x = 0.8$ we find:

```
\begin{align*}
\Pr(mc \mid isc)(0.2) &= 0.200 \\
\Pr(mc \mid isc)(0.5) &= 0.385 \\
\Pr(mc \mid isc)(0.8) &= 0.500
\end{align*}
```
Computing an example sensitivity function (2)

Compute the sensitivity function for output probability $\Pr(mc \mid isc)$ as a function of $x = \gamma(isc \mid mc)$:

- we then establish a system of linear equations:

\[
\begin{align*}
\Pr(mc \mid isc)(0.2) &= 0.200 & a' \cdot 0.2 + b' \over 0.2 + d' &= 0.200 \\
\Pr(mc \mid isc)(0.5) &= 0.385 & a' \cdot 0.5 + b' \over 0.5 + d' &= 0.385 \\
\Pr(mc \mid isc)(0.8) &= 0.500 & a' \cdot 0.8 + b' \over 0.8 + d' &= 0.500
\end{align*}
\]
Computing an example sensitivity function (3)

Compute the sensitivity function for output probability \( \Pr(mc \mid isc) \) as a function of \( x = \gamma(isc \mid mc) \):

- we solve the system of linear equations:

\[
\begin{align*}
    a' \cdot 0.2 + b' &= 0.200 \cdot 0.2 + 0.200 \cdot d' \\
    a' \cdot 0.5 + b' &= 0.385 \cdot 0.5 + 0.385 \cdot d'
\end{align*}
\]

which together give \( a' = 1.525/3 + 1.85/3 \cdot d' \).

Combining this with equation

\[
\begin{align*}
    a' \cdot 0.8 + b' &= 0.500 \cdot 0.8 + 0.500 \cdot d'
\end{align*}
\]

gives \( b' = -0.2/30 + 0.2/30 \cdot d' \).

Substituting \( a' \) and \( b' \) in the first equation gives

\( d' = 1.65/2.1 \approx 0.786 \) and therefore \( a' \approx 0.993 \) and \( b' \approx -0.001 \).
Practicable sensitivity analysis

Straightforward sensitivity analysis of a Bayesian network is infeasible. The digraph of the network, however, induces

- algebraic independence of the output probability of various parameter probabilities;
- simple mathematical functions that relate the output probability to the potentially influential parameters.

By exploiting these properties, sensitivity analysis of a Bayesian network is rendered practicable.

Still, the number of sensitivity functions returned from all potentially influential parameters can be quite large.

How do we select the parameters that we consider sensitive and that require further study?
Selection of sensitive assessments

A sensitivity analysis results in a large amount of data.

Example: the oesophageal cancer network:

In the prior network, 206 parameters potentially influence the 6 probabilities of \( \Pr(\text{Stage}) \rightarrow 1236 \) sensitivity functions.

Given patient evidence (156), the number of potentially influential parameters may become 826.

Various selection criteria can be employed to select parameters that deserve attention.
Selection criteria

Parameter assessments that may require further study can be selected based upon:

- **absolute effect of variation on output probability:**
  \[|f(0) - f(1)|;\]
- **plausible effect on output probability;**
- **the sensitivity value,** i.e. the absolute value of the first derivative of the sensitivity function at original assessment;
- **the vertex proximity,** i.e. the distance between the original assessment of the parameter and the vertex ("shoulder") of the function;
- **the admissible deviation,** i.e. the variation allowed in the parameter without changing the most likely value of the variable of interest.
The sensitivity value as selection criterion

Consider a sensitivity function \( f(x) \) for parameter \( x \) and some output probability. Let \( x_0 \) denote the original assessment for \( x \).

The absolute value of the first derivative of \( f(x) \) in \( (x_0, f(x_0)) \), also called the sensitivity value, captures how sensitive the output is to varying \( x \).

\[
\left| \frac{\partial f}{\partial x}(0.02) \right| = 6.97
\]

Problem: the first derivative is a good approximation of the function only for \( x \in [x_0 - \epsilon, x_0 + \epsilon] \).
Vertex proximity

Let $x, x_0$ and $f(x)$ be as before.

The sensitivity value in $x_0$ may be small near the vertex (shoulder) of a sensitivity function.

Yet, slight variation of the parameter around $x_0$ can have a large effect on the outcome probability.

Solution: if $x_0$ is close to $x_{vertex}$, then select $x$ for further study, regardless of the sensitivity value.
The admissible deviation

small sensitivity value, smaller admissible deviation
More elaborate sensitivity analyses

Properties of an $n$-way analysis for $n > 1$:

- all $n$ parameter probabilities are varied simultaneously.
- reveals possible interactions, or synergistic effects.
- sensitivity function is a fraction of two multi-linear functions in the parameters under study.
- hardly any research into shapes and properties of $n$-way sensitivity functions for $n \geq 2$.
- interpretation of results is hard, especially for $n > 2$. 
Two-way sensitivity analyses

With a two-way sensitivity analysis, two parameter probabilities are varied simultaneously:

\[ f(x, y) = \frac{c_1 \cdot x \cdot y + c_2 \cdot x + c_3 \cdot y + c_4}{c_5 \cdot x \cdot y + c_6 \cdot x + c_7 \cdot y + c_8} \]

A two-way analysis reveals possible synergistic effects \((c_1, c_5)\) not found from two one-way analyses.

**Selection criteria:** Parameter assessments that may require further study can be selected based upon:

- absolute effect of variation on output probability;
- plausible effect on output probability;
- the (max) sensitivity value: \(\sqrt{(\frac{\partial f}{\partial x}(x_0, y_0))^2 + (\frac{\partial f}{\partial y}(x_0, y_0))^2}\)
- contour distances, i.e. the distances between iso-probability lines in a 2D projection of the sensitivity function.
Contour distance

A two-way analysis reveals *synergistic* effects.

- **absolute distance**: the smaller the distance, the more sensitive the output probability is to parameter variation;
- **relative distance**: varying distances indicate interaction effects.

The iso-probability contours here are not equi-distant due to non-zero interaction terms in the sensitivity function.
We can provide general bounds on sensitivity functions through \((x_0, p_0)\) and on their properties\(^7\)

which can be further bounded\(^8\) given \(f_{\text{Pr}(e)}(x) = c \cdot x + d\):

\[
f_{\text{Pr}(h|e)}(x) = \frac{r}{x - s} + t, \quad r = (x_0 - s) \cdot (p_0 - t)
\]

for asymptotes \(x = s = -\frac{d}{c}\) and \(y = t\).

---


Brief: robustness to structure changes

We can simulate the removal of an arc by posing constraints on an $n$-way sensitivity function\(^9\)

Original CPT for node $B$:

<table>
<thead>
<tr>
<th></th>
<th>$c_1$</th>
<th>$c_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_1$</td>
<td>0.7 0.1</td>
<td></td>
</tr>
<tr>
<td>$a_2$</td>
<td>0.9 0.6</td>
<td></td>
</tr>
<tr>
<td>$b_1$</td>
<td>0.3 0.9</td>
<td></td>
</tr>
<tr>
<td>$b_2$</td>
<td>0.1 0.4</td>
<td></td>
</tr>
</tbody>
</table>

For removing $A \rightarrow B$:

<table>
<thead>
<tr>
<th></th>
<th>$c_1$</th>
<th>$c_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_1$</td>
<td>$x_1'$</td>
<td></td>
</tr>
<tr>
<td>$a_2$</td>
<td>$x_2'$</td>
<td></td>
</tr>
<tr>
<td>$b_1$</td>
<td>$1 - x_1'$</td>
<td></td>
</tr>
<tr>
<td>$b_2$</td>
<td>$1 - x_2'$</td>
<td></td>
</tr>
</tbody>
</table>

Brief: robustness to discretisation

We can study the effect of choosing a different discretisation\textsuperscript{10}

- changing a discretisation threshold is like varying a parameter

Brief: robustness of classification performance

We can gain understanding about the behaviour of networks of restricted topology

- naive Bayesian network classifiers\(^\text{11}\)

- multi-dimensional Bayesian network classifiers\(^\text{12}\)

---


Brief: results applied in other contexts

Rather than using sensitivity functions as analysis tools, we can exploit their properties in other contexts\textsuperscript{13}

- parameter tuning \textsuperscript{14}
- pre-processing inference in credal networks \textsuperscript{15}
- \ldots ?


\textsuperscript{14} J.H. Bolt, S. Renooij (2014). Local sensitivity of Bayesian networks to multiple simultaneous parameter shifts. PGM 2014

Evaluation of Bayesian networks

An evaluation of the practical value of a Bayesian network consists of the following steps:

1) select realistic cases to evaluate (for example from data or scenarios);
2) select the outcome variable(s) of interest;
3) choose a standard of validity;
4) compute, from the network, the outcome for each case;
5) compare the outcome to your standard of validity.
Evaluation of Bayesian networks: an example

Consider the evaluation of the practical value of the oesophageal cancer network.

- data: symptoms and test-results for 156 patients (average: 14.8 of the 25, per patient);
- outcomes of interest: Stage of the tumour: I, IIA, IIB, III, IVA, IVB;
- standard of validity: assessment of the stage, given by the physicians.

From the oesophageal cancer network we now compute the stage for each of the 156 patients.
Patient file for Patient X

Passage: can pass mashed food
Weightloss: none
Physical exam: swollen lymph nodes neck

Biopsy: squamous
X-lungs: metastases
Bronchoscopy: ×
Sono-cervix: ×
Barium swallow: ×
Gastroscopy: circumf: length: location: necrosis: shape: circular 7 cm proximal absent polypoid

CT-scan (liver, locoregion, lungs, organs, truncus): ×
Endosonography (locoregion, mediastinum, truncus, wall): ×
Laparascopy (liver, diaphragm, truncus): ×

Diagnosis: stage = I/IIA/IIB/III/IVA/ IVB
Diagnosing Patient X
The percentage correct

After processing evidence, a Bayesian network gives a posterior probability distribution for the outcome variable.

The standard of validity, however, usually consists of a single value for the outcome variable.

- The most likely value of the outcome variable is chosen as the outcome of the network;
- the outcome is compared against the standard: the outcome is either correct or incorrect.

The percentage of cases where the outcome predicted by the network is correct according to the standard of validity is called the percentage correct.
The percentage correct: an example

Compare for each patient the *stage* predicted by the network against the *stage* assessed by the physicians.

For 133 of the 156 patients, the network gives an accurate prediction:

<table>
<thead>
<tr>
<th>phys.</th>
<th>network</th>
<th>I</th>
<th>IIA</th>
<th>IIB</th>
<th>III</th>
<th>IVA</th>
<th>IVB</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>IIA</td>
<td>0</td>
<td>37</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>38</td>
</tr>
<tr>
<td>IIB</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>III</td>
<td>1</td>
<td>10</td>
<td>0</td>
<td>36</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>47</td>
</tr>
<tr>
<td>IVA</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>35</td>
<td>0</td>
<td>0</td>
<td>39</td>
</tr>
<tr>
<td>IVB</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>23</td>
<td>0</td>
<td>26</td>
</tr>
<tr>
<td>total</td>
<td>3</td>
<td>48</td>
<td>0</td>
<td>47</td>
<td>35</td>
<td>23</td>
<td>0</td>
<td>156</td>
</tr>
</tbody>
</table>

The percentage correct is therefore 85%.
Explaining the differences

Differences between the outcomes of a network and the standard of validity can originate from several sources:

- modelling errors;
- errors in the standard, or in the data;
- random variation:

![Bar charts showing data for patients B and C with values for different categories (I, IIA, IIB, III, IVA, IVB).]
Evaluation scores: the \textit{Brier} score

The uncertainty expressed in the predicted distribution can be taken into account in the evaluation.

Let $p_{ij} = \Pr(v_j \mid e_i)$ be the predicted (network) probability for case $i$ and value $j$ of the outcome variable.

Let $s_{ij} = \begin{cases} 1 & \text{if outcome } j \text{ is correct outcome for case } i \\ 0 & \text{otherwise} \end{cases}$ (according to standard of validity);

The \textbf{Brier score} for the predicted distribution for case $i$ now is

$$B_i = \sum_j (p_{ij} - s_{ij})^2$$

The Brier score lies within the interval $[0, 2]$, where 0 indicates a perfect prediction.
The Brier score: an example

Consider evaluating the oesophageal cancer network, where

- $p_{ij}$ is the network probability computed for patient $i$ and stage $j \in \{I, \ldots, IVB\}$;
- $s_{ij}$ returns 1 if patient $i$’s medical file states stage $j$, and 0 otherwise.

The Brier score for patient $i$ now is

$$B_i = \sum_{j=I,\ldots,IVB} (p_{ij} - s_{ij})^2$$

For patients X, B and C we find, respectively:

$$B_X = (0 - 0)^2 + (0.01 - 0)^2 + (0.04 - 0)^2 + (0.14 - 0)^2 +$$
$$+ (0.06 - 0)^2 + (0.75 - 1)^2 = 0.09$$

$$B_B = 3 \cdot (0 - 0)^2 + (0.36 - 1)^2 + (0.35 - 0)^2 + (0.29 - 0)^2 = 0.62$$

$$B_C = (0.02 - 0)^2 + (0.38 - 0)^2 + (0.05 - 0)^2 + (0.37 - 1)^2 +$$
$$+ (0.09 - 0)^2 + (0.09 - 0)^2 = 0.56$$
Average Brier score

We can compute an average Brier score over $n$ ‘forecasts’:

$$B = \frac{1}{n} \sum_{i=1,\ldots,n} B_i$$

An example: The average Brier score over all patients per predicted-stage / actual-stage combination:

<table>
<thead>
<tr>
<th>network</th>
<th>I</th>
<th>IIA</th>
<th>IIB</th>
<th>III</th>
<th>IVA</th>
<th>IVB</th>
</tr>
</thead>
<tbody>
<tr>
<td>phys.</td>
<td>I</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>0.21</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>IIA</td>
<td>–</td>
<td>0.28</td>
<td>–</td>
<td>1.52</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>IIB</td>
<td>–</td>
<td>1.17</td>
<td>–</td>
<td>0.98</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>III</td>
<td>1.40</td>
<td>0.89</td>
<td>–</td>
<td>0.26</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>IVA</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>0.75</td>
<td>0.08</td>
<td>–</td>
</tr>
<tr>
<td>IVB</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>0.87</td>
<td>–</td>
<td>0.06</td>
</tr>
</tbody>
</table>

The average Brier score over all 156 patients is: 0.29
Decision support: a two-layer problem solving architecture

The control layer

The probabilistic layer

Probabilistic layer for probabilistic reasoning:
- stores: a Bayesian network;
- tasks: receive evidence, propagate it, and return requested probabilities.

Control layer for (intelligent) control over reasoning
- stores: non-probabilistic information;
- tasks: make strategic decisions by sending evidence, requesting probabilistic information, computing non-probabilistic information.
Problem solving: Threshold decision making

The purpose of threshold decision making is supporting the choice between therapeutic decision alternatives.

A system for threshold decision making has the following tasks:

- **Diagnostic reasoning**: compute the probability $Pr(d)$ of some hypothesis (diagnosis), based upon the available findings.

  $$
  P^-(d) \quad P^+(d)
  $$

- **Treatment advisement**: give advise concerning treatment, based upon $Pr(d)$ and the threshold values for the treatment options.
Threshold decision making

A simple strategy for threshold decision making using a Bayesian network $\mathcal{B} = (G, \Gamma)$:

**PROCEDURE** THRESHOLDDECISION($\mathcal{B}, c_E, P, A$):

1. PROPAGATE-EVIDENCE($\mathcal{B}, c_E$);
2. ADVISE($P, A$)

END

The procedure is called with

- evidence $c_E$ for a set of nodes $E \subset V_G$, and
- a set of threshold values $P$ for the diagnosis under consideration.

The procedure returns a treatment alternative of $A \notin V_G$. 
Expected utility of treatment

The choice between two treatment alternatives depends on their expected benefit. Benefit can be defined in terms of utility.

Consider hypothesis node $H$ and evidence $e$ for a node $E$; variable $A$ models different treatment alternatives.

- the desirability of each $c_{AH}$ of $A$ and $H$ is given by a subjective utility $u(c_{AH})$;
- the expected utility of each treatment alternative $c_A$ then is
  \[
  \hat{u}(c_A) = \sum_{c_H} u(c_A \land c_H) \cdot \Pr^e(c_H), \text{ where } c_A \land c_H \equiv c_{AH}
  \]

Advise: treatment alternative with highest expected utility.

Drawback: each $\hat{u}(c_A)$ has to be recomputed every time a different value for $\Pr^e(c_H)$ is encountered...
Expected utility for setting thresholds

Let $H$, $e$ and $A$ be as before. Expected utility can be written as a function of $Pr^e(h)$ for value of interest $h$ of $H$.

In case of a binary-valued $H$ this function equals:

$$\hat{u}(c_A) = \sum_{c_H} u(c_A \land c_H) \cdot Pr^e(c_H)$$

$$= u(c_A \land h) \cdot Pr^e(h) + u(c_A \land \neg h) \cdot Pr^e(\neg h)$$

$$= (u(c_A \land h) - u(c_A \land \neg h)) \cdot Pr^e(h) + u(c_A \land \neg h)$$

Therefore, with $x = Pr^e(h)$ we have

$$\hat{u}(c_A)(x) = (u(c_A \land h) - u(c_A \land \neg h)) \cdot x + u(c_A \land \neg h)$$

Threshold probabilities are computed by solving $x$ (for each pair of alternatives $a_i$ and $a_j$, $i \neq j$, for $A$) from

$$\hat{u}(a_i)(x) = \hat{u}(a_j)(x).$$
An example

Consider the following network and utilities $u(c_A \land c_H)$:

- $u(stop \land b) = 0.02$
- $u(stop \land \neg b) = 1.00$
- $u(treat \land b) = 0.50$
- $u(treat \land \neg b) = 0.92$

Threshold value $P^* \approx 0.143$ is computed from:

- $\hat{u}(treat)(x) = (0.50 - 0.92) \cdot x + 0.92$
- $\hat{u}(stop)(x) = -0.98 \cdot x + 1.00$

where $x = Pr^e(h) = Pr(b)$

Should a patient with $Pr(b) = 0.10$ be treated or not?
An example

Consider the following network and utilities $u(c_A \land c_H)$:

- $u(stop \land b) = 0.02$
- $u(stop \land \neg b) = 1.00$
- $u(test \land b) = 0.45$
- $u(test \land \neg b) = 0.98$
- $u(treat \land b) = 0.50$
- $u(treat \land \neg b) = 0.92$

Threshold values $P^- \approx 0.044$ and $P^+ \approx 0.545$ are computed from:

- $\hat{u}(stop)(x) = -0.98 \cdot x + 1.00$
- $\hat{u}(treat)(x) = -0.50 \cdot x + 0.50$
- $\hat{u}(test)(x) = -0.98 \cdot x + 1.00$
Threshold decision making: summary

For threshold decision making, the probabilistic layer and the control layer have the following functionality:

Probabilistic layer:
- propagates evidence and returns requested probabilities

Control layer:
- stores utility functions
- computes and stores threshold probabilities for different treatment choices;
- compares probabilities with appropriate thresholds and returns a treatment advise based upon the comparisons.
Problem solving: Diagnostication

Diagnostication: determine the most likely hypothesis (diagnosis), at the lowest possible costs.

A system for diagnostication has the following tasks:

• Diagnostic reasoning: determine most likely problem cause from available information about its manifestations.

• Test selection: select appropriate tests to gain more information about the manifestations.

• Stopping criterion evaluation: check whether the current diagnosis is sufficiently reliable.
A simple strategy for diagnostication using a Bayesian network $\mathcal{B} = (G, \Gamma)$:

**PROCEDURE** \textsc{D}iagnostication($\mathcal{B}, \mathcal{E}, H$):

- \texttt{SUFFICIENT} $\leftarrow$ \texttt{FALSE};
- \textbf{WHILE} $\mathcal{E} \neq \emptyset$ AND NOT \texttt{SUFFICIENT} \textbf{DO}
  - $E_i \leftarrow$ \textsc{Select-Test($\mathcal{E}$)};
  - $e_i \leftarrow$ \textsc{Gather-Evidence($E_i$)};
  - \textsc{Propagate-Evidence($\mathcal{B}, e_i$)};
  - $\mathcal{E} \leftarrow \mathcal{E} \setminus \{E_i\}$;
  - \texttt{SUFFICIENT} $\leftarrow$ \texttt{Evaluate-Stop}
- \textbf{OD};
- \textsc{Diagnose($H$)}

\textbf{END}

The procedure is called with the set $\mathcal{E} \subset V_G$ of all evidence nodes. It returns a sufficiently reliable hypothesis for $H \in V_G$. 
Test-selection measures

Gathering evidence has benefit for diagnostication, as it may decrease uncertainty concerning the diagnosis.

Most often information measures are used to establish the expected benefit:

- Shannon entropy;
- Gini index;
- misclassification error;
- Kullback-Leibler divergence (uses cross entropy);
- expected utility

These measures all measure uncertainty only; it is possible to include different types of cost as well.
Expected utility for selecting tests

Consider binary hypothesis node $H$. Let $e$ denote the processed evidence and let $E_i$ be a relevant uninstantiated evidence node.

- The utility of the value $c_{E_i}$ for node $E_i$ is defined as
  \[
  u(c_{E_i}) = |\Pr^e(h) - \Pr^e(h \mid c_{E_i})|
  \]

- The expected utility of observing a value for node $E_i$ (i.e. doing the test) then is
  \[
  \hat{u}(E_i) = \sum_{c_{E_i}} u(c_{E_i}) \cdot \Pr^e(c_{E_i})
  \]

$\text{SELECT-TEST}(E)$ now returns a node $E_i \in E$ with highest expected utility.
An example

\[ \gamma_{V_1}(v_1) = 0.7 \]

\[ \gamma_{V_2}(v_2 \mid v_1) = 0.7 \]
\[ \gamma_{V_2}(v_2 \mid \neg v_1) = 0.6 \]

\[ \gamma_{V_3}(v_3 \mid v_2) = 0.9 \]
\[ \gamma_{V_3}(v_3 \mid \neg v_2) = 0.2 \]

\[ \gamma_{V_4}(v_4 \mid v_2) = 0.3 \]
\[ \gamma_{V_4}(v_4 \mid \neg v_2) = 0.8 \]

\( V_2 \) is an hypothesis node;
\( V_1, V_3 \) and \( V_4 \) are evidence nodes; all are uninstantiated.

\[ \Pr^e(h) = \Pr(v_2) = 0.67 \]

For \( V_3 \):
\[ u(v_3) = |\Pr(v_2) - \Pr(v_2 \mid v_3)| = |0.67 - 0.901| = 0.231 \]
\[ u(\neg v_3) = |\Pr(v_2) - \Pr(v_2 \mid \neg v_3)| = |0.67 - 0.202| = 0.468 \]

The expected benefit of obtaining \( V_3 \)’s value is:
\[ \hat{u}(V_3) = u(v_3) \cdot \Pr(v_3) + u(\neg v_3) \cdot \Pr(\neg v_3) \]
\[ = 0.231 \cdot 0.669 + 0.468 \cdot 0.331 = 0.309 \]
Some assumptions

To reduce computational complexity two simplifying assumptions are made:

- the myopia assumption: tests are selected and performed one at a time;
- the single-disorder assumption: all hypotheses are mutually exclusive.

Both assumptions, however, can be somewhat relaxed.
Stopping criteria

After processing newly obtained evidence, a stopping criterion is evaluated: if this criterion is met, the selection of tests is halted.

Some examples of stopping criteria:

- **sufficiency of confirmation**: the probability of the hypothesis is above (below) a given threshold value;
  (or: take the entire distribution over the hypothesis node into consideration)

- **sufficiency of information**: the expected utilities of the relevant uninstantiated evidence nodes are below a given threshold value;
  (or: take the maximum utility instead of expected utility into consideration).
An example

$\gamma_{V_1}(v_1) = 0.7$

$\gamma_{V_2}(v_2 \mid v_1) = 0.7$

$\gamma_{V_2}(v_2 \mid \neg v_1) = 0.6$

$\gamma_{V_3}(v_3 \mid v_2) = 0.9$

$\gamma_{V_3}(v_3 \mid \neg v_2) = 0.2$

$\gamma_{V_4}(v_4 \mid v_2) = 0.3$

$\gamma_{V_4}(v_4 \mid \neg v_2) = 0.8$

$V_2$ is an hypothesis node; $V_1$, $V_3$ and $V_4$ are evidence nodes.

Suppose the stopping criterion for selecting tests is ‘sufficiency of information’ with a threshold value of 0.1.

With evidence $V_3 = \text{true}$, we find $\Pr^e(h) = \Pr^{v_3}(v_2) = 0.90$.

The expected utilities for $V_1$ and $V_4$ are now updated for $e = v_3$:

$\hat{u}(V_1) = 0.017$ and $\hat{u}(V_4) = 0.089$

Both expected utilities are below 0.1 so selection of tests is halted.
Diagnostication: summary

For diagnostication, the probabilistic layer and the control layer have the following functionality:

Probabilistic layer:

- propagates evidence and returns requested probabilities

Control layer:

- stores knowledge concerning the roles of different variables (hypothesis, evidence, intermediate);
- stores and computes (expected) utilities of the different tests available;
- selects the most appropriate tests;
- evaluates the stopping criterion.
Chapter 7: Conclusions
Concluding observations

The state of the art as far as Bayesian networks are concerned is as follows:

- Bayesian networks and their associated algorithms offer a useful framework for representing and manipulating probabilistic information — the framework combines mathematical correctness with expressiveness and efficiency;
- advances in research enable and facilitate applicability of Bayesian networks in more and more practical situations;
- Bayesian networks are not yet commonplace, but are becoming so.
Most research is aimed at supporting the practical application of Bayesian networks.

- design of methodologies for knowledge acquisition;
- relevance of variables, values, arcs and probabilities;
- learning from data;
- incremental model-construction;
- representation and manipulation of continuous variables;
- representation and manipulation of time;
- approximate inference;
- model-complexity vs accuracy;
- model-checking and repairing;
- embedding in decision support systems;
- building actual applications;
- design of software packages for builders;
- design of user-interfaces for users;
- causality
Interested in more?

For further information on research on the subject of this course, see:

- links on the course website, also for info about graduation projects;
- (proceedings of) the annual UAI conference on Uncertainty in Artificial Intelligence;
- (online proceedings of) the annual BMAW workshop linked to UAI: Bayesian Modeling Applications Workshop;
- (proceedings of) the bi-annual PGM conference on Probabilistic Graphical Models;
- authors’ homepages (including mine: publications and projects);
- . . .