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

1. $G = (V_G, A_G)$ is a DAG with arcs $A_G$ and nodes $V_G = V$, representing a set of random variables $V = \{V_1, \ldots, V_n\}$, $n \geq 1$;

2. $\Gamma = \{\gamma_{V_i} \mid V_i \in V\}$ 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 $G$:
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 $V \in \mathcal{V}$ to be binary-valued.

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

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

\begin{align*}
\gamma_{\text{Co}}(\text{co} \mid \text{tum} \land \text{cal conc}) &= 0.9 & \gamma_{\text{Co}}(\text{co} \mid \neg \text{tum} \land \text{cal conc}) &= 0.8 \\
\gamma_{\text{Co}}(\text{co} \mid \text{tum} \land \neg \text{cal conc}) &= 0.7 & \gamma_{\text{Co}}(\text{co} \mid \neg \text{tum} \land \neg \text{cal conc}) &= 0.05 \\
\gamma_{\text{Co}}(\neg \text{co} \mid \text{tum} \land \text{cal conc}) &= 0.1 & \gamma_{\text{Co}}(\neg \text{co} \mid \neg \text{tum} \land \text{cal conc}) &= 0.2 \\
\gamma_{\text{Co}}(\neg \text{co} \mid \text{tum} \land \neg \text{cal conc}) &= 0.3 & \gamma_{\text{Co}}(\neg \text{co} \mid \neg \text{tum} \land \neg \text{cal conc}) &= 0.95
\end{align*}

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

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

$$\Pr(V) = \prod_{i=1}^{n} \gamma_i(V_i \mid \rho(V_i))$$

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

$\Pr$ is called the joint distribution defined by $\mathcal{B}$ and is said to respect the independences portrayed in $G$.

NB we will often omit the subscript in $\gamma$ if no confusion is possible.
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:
A probabilistic interpretation: proof continued

Take ordering $\nu_G$ as an ordering on the random 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:

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

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

$$\cdot P(V_3 \mid V_1 \wedge V_2) \cdot P(V_2 \mid V_1) \cdot P(V_1)$$
A probabilistic interpretation: proof continued

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 $\mathcal{G}$ 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 $W, E \subset V$;
- blocking sets $Z$ for d-separation now have an intuitive meaning: if we have evidence / observations for variables $E \subset V$ then we typically investigate blocking set $Z = E$. 
An example

Consider Bayesian network $\mathcal{B}$, defining joint distribution $\Pr$:

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

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 $\Pr(V_i \mid c_E)$), 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, but constant is 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

count count count

count count count

count count count

count count count

count count count

count count count

count count count

count count count

count count count

count count count

count count count

count count count

count count count

count count count

count count count

count count count

count count count

count count count

count count count

count count count
A computational architecture
A computational architecture
A computational architecture

\[
\sum_{1}^{4}, \quad \sum_{1}^{3}, \quad \sum_{1}^{2}, \quad \sum_{1}^{3}, \quad \sum_{1}^{4}
\]
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: $\Pr(V_2 | V_1) = \gamma(V_2 | V_1)$

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

\[
\begin{align*}
\Pr(v_2) &= \Pr(v_2 | v_1) \cdot \Pr(v_1) + \Pr(v_2 | \neg v_1) \cdot \Pr(\neg v_1) \\
\Pr(\neg v_2) &= \Pr(\neg v_2 | v_1) \cdot \Pr(v_1) + \Pr(\neg v_2 | \neg v_1) \cdot \Pr(\neg v_1)
\end{align*}
\]
Consider Bayesian network $\mathcal{B}$ 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:

\[
\begin{align*}
\Pr(v_3) &= \Pr(v_3 \mid v_2) \cdot \Pr(v_2) + \Pr(v_3 \mid \neg v_2) \cdot \Pr(\neg v_2) \\
&= \gamma(v_3 \mid v_2) \cdot \Pr(v_2) + \gamma(v_3 \mid \neg v_2) \cdot \Pr(\neg v_2) \\
\Pr(\neg v_3) &= \gamma(\neg v_3 \mid v_2) \cdot \Pr(v_2) + \gamma(\neg v_3 \mid \neg v_2) \cdot \Pr(\neg v_2)
\end{align*}
\]
Introduction to causal parameters

Reconsider Bayesian network $\mathcal{B}$ without observations:

$$\begin{align*}
\pi_{V_2}^{V_1} & \downarrow \\
V_1 & \quad \gamma(v_1), \gamma(\neg v_1) \\
V_2 & \quad \gamma(v_2 \mid v_1), \gamma(\neg v_2 \mid v_1) \\
& \quad \gamma(v_2 \mid \neg v_1), \gamma(\neg v_2 \mid \neg v_1)
\end{align*}$$

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

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

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

The function $\pi_{V_2}^{V_1}$ 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 $\pi_{V_1}^{V_2}$

<table>
<thead>
<tr>
<th>Node $V_i$</th>
<th>Causal Parameter $\gamma(v_i)$</th>
<th>$\gamma(\neg v_i)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_1$</td>
<td>$0.7$</td>
<td>$0.3$</td>
</tr>
<tr>
<td>$V_2$</td>
<td>$0.2$</td>
<td>$0.8$</td>
</tr>
<tr>
<td>$V_3$</td>
<td>$0.6$</td>
<td>$0.4$</td>
</tr>
</tbody>
</table>

with

$$\pi_{V_2}^{V_1}(v_1) = \gamma(v_1) = 0.7; \quad \pi_{V_2}^{V_1}(\neg v_1) = 0.3$$

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

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

\[ \gamma(v_1) = 0.7, \quad \gamma(\neg v_1) = 0.3 \]

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

\[ \gamma(v_2 | v_1) = 0.2, \quad \gamma(\neg v_2 | v_1) = 0.8 \]
\[ \gamma(v_2 | \neg v_1) = 0.5, \quad \gamma(\neg v_2 | \neg v_1) = 0.5 \]

\[ \gamma(v_3 | v_2) = 0.6, \quad \gamma(\neg v_3 | v_2) = 0.4 \]
\[ \gamma(v_3 | \neg v_2) = 0.1, \quad \gamma(\neg v_3 | \neg v_2) = 0.9 \]

with

\[ \pi_{V_3}^{V_2}(v_2) = \Pr(v_2 | v_1) \cdot \Pr(v_1) + \Pr(v_2 | \neg v_1) \cdot \Pr(\neg v_1) \]
\[ = \gamma(v_2 | v_1) \cdot \pi_{V_2}^{V_1}(v_1) + \gamma(v_2 | \neg v_1) \cdot \pi_{V_2}^{V_1}(\neg v_1) \]
\[ = 0.2 \cdot 0.7 + 0.5 \cdot 0.3 = 0.29 \]

\[ \pi_{V_3}^{V_2}(\neg v_2) = 0.8 \cdot 0.7 + 0.5 \cdot 0.3 = 0.71 \]

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 $\pi_{V_2}^V$ from $V_2$
- sends no messages

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

\[
\Pr(v_3) = \gamma(v_3 | v_2) \cdot \pi_{V_3}^V(v_2) + \gamma(v_3 | \neg v_2) \cdot \pi_{V_3}^V(\neg v_2)
\]

\[
= 0.6 \cdot 0.29 + 0.1 \cdot 0.71 = 0.245
\]

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

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 $B$ with evidence $V_1 = true$ ($v_1$) and the following graph:

Node $V_1$ updates its probabilities and causal parameter:

\[
\pi_{V_2}^V(v_1) = \Pr^{v_1}(v_1) = \Pr(v_1 | v_1) = 1
\]

\[
\pi_{V_2}^V(\neg v_1) = \Pr^{v_1}(\neg v_1) = 0
\]

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

\[
\Pr^{v_1}(v_2) = \gamma(v_2 | v_1) \cdot \pi_{V_2}^V(v_1) + \gamma(v_2 | \neg v_1) \cdot \pi_{V_2}^V(\neg v_1) = \gamma(v_2 | v_1)
\]

\[
\Pr^{v_1}(\neg v_2) = \gamma(\neg v_2 | v_1) \cdot \pi_{V_2}^V(v_1) + \gamma(\neg v_2 | \neg v_1) \cdot \pi_{V_2}^V(\neg v_1) = \gamma(\neg v_2 | v_1)
\]

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

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

\[
\begin{align*}
V_1 & : \gamma(v_1), \gamma(\neg v_1) \\
V_2 & : \gamma(v_2 | v_1), \gamma(\neg v_2 | v_1), \\
& \quad \gamma(v_2 | \neg v_1), \gamma(\neg v_2 | \neg v_1)
\end{align*}
\]

Suppose we have evidence $V_2 = true$ 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?
Understanding Pearl with evidence (2b)

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:

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

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

Reconsider Bayesian network $B$:

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

This message is a function $\lambda_{V_2}^{V_1} : \{v_1, \neg v_1\} \rightarrow [0, 1]$ 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 $\lambda_{V_2}^{V_1}$ is called the diagnostic parameter from $V_2$ to $V_1$. 
Consider the following Bayesian network $B$ with evidence $V_2 = true$:

$V_1$: $\gamma(v_1) = 0.8$, $\gamma(\neg v_1) = 0.2$

$V_2$: $\gamma(v_2 | v_1) = 0.4$, $\gamma(\neg v_2 | v_1) = 0.6$

$\gamma(v_2 | \neg v_1) = 0.9$, $\gamma(\neg v_2 | \neg v_1) = 0.1$

Node $V_2$:

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

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

$$\lambda_{V_2}^{V_1}(\neg v_1) = \gamma(v_2 | \neg v_1) = 0.9$$

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

Node $V_1$ computes:

$$\Pr^{v_2}(v_1) = \alpha \cdot \Pr(v_2 | v_1) \cdot \Pr(v_1)$$
$$\quad = \alpha \cdot \lambda_{V_2}^{V_1}(v_1) \cdot \gamma(v_1) = \alpha \cdot 0.4 \cdot 0.8 = \alpha \cdot 0.32$$
$$\Pr^{v_2}(\neg v_1) = \alpha \cdot \lambda_{V_2}^{V_1}(\neg v_1) \cdot \gamma(\neg v_1) = \alpha \cdot 0.9 \cdot 0.2 = \alpha \cdot 0.18$$

Node $V_1$ now normalises its probabilities using
$$\Pr^{v_2}(v_1) + \Pr^{v_2}(\neg v_1) = 1: \alpha \cdot 0.32 + \alpha \cdot 0.18 = 1 \quad \Longrightarrow \quad \alpha = 2$$
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 $B$ with the following graph:

Suppose we have evidence $V_3 = \text{true}$ 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?
Consider Bayesian network $\mathcal{B}$ with evidence $V_3 = true$ 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

\[
\begin{align*}
\Pr_{v_3}(v_1) &= \alpha \cdot \Pr(v_3 | v_1) \cdot \Pr(v_1) = \alpha \cdot \lambda_{V_1}^{V_2}(v_1) \cdot \gamma(v_1) \\
\Pr_{v_3}(\neg v_1) &= \alpha \cdot \Pr(v_3 | \neg v_1) \cdot \Pr(\neg v_1) = \alpha \cdot \lambda_{V_1}^{V_2}(\neg v_1) \cdot \gamma(\neg v_1)
\end{align*}
\]
Pearl on directed paths – An example (2)

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

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

$$
\lambda_{V_2}^V_1(v_1) = \Pr(v_3 \mid v_1) \\
= \Pr(v_3 \mid v_2) \cdot \Pr(v_2 \mid v_1) + \Pr(v_3 \mid \neg v_2) \cdot \Pr(\neg v_2 \mid v_1) \\
= \lambda_{V_3}^V_2(v_2) \cdot \gamma(v_2 \mid v_1) + \lambda_{V_3}^V_2(\neg v_2) \cdot \gamma(\neg v_2 \mid v_1) \\
\lambda_{V_2}^V_1(\neg v_1) = \Pr(v_3 \mid \neg v_1) = \ldots
$$

The node then computes $\Pr^{v_3}(V_2)$... How?
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

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

$$\lambda_{V_3}^{V_2}(-v_2) = \Pr(v_3 \mid -v_2) = \gamma(v_3 \mid -v_2)$$

- computes $\Pr^{v_3}(V_3)$
Consider the Bayesian networks $\mathcal{B}$ with the following graphs:

- $V_1$: $\gamma(v_1 \mid v_2), \gamma(\neg v_1 \mid v_2)$
- $V_2$: $\gamma(v_2), \gamma(\neg v_2)$
- $V_3$: $\gamma(v_3 \mid v_2), \gamma(\neg v_3 \mid v_2)$

Suppose we have evidence $V_3 = true$ 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 $\operatorname{Pr}$ 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 $\pi_{V_{ij}}^{V_i}(V_i)$ to all children $V_{ij}$

compute and send messages $\lambda_{V_{jk}}^{V_i}(V_{jk})$ to all parents $V_{jk}$

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 random 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: $\tilde{c}_V = T(true)$

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:

![Graph Diagram]

**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 \( V_i. \)
An example

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 $V_i^+ = \bigcup_{V_j \in \rho(V_i)} V_{G_i}^+$ 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 $\tilde{c}_V = \tilde{c}_{V_i^-} \wedge \tilde{c}_{V_i^+}$ 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 $\mathcal{B} = (G, \Gamma)$ be a Bayesian network with singly connected graph $G = (\mathcal{V}_G, \mathcal{A}_G)$; let $\Pr$ be the joint distribution defined by $\mathcal{B}$. Then

$$\text{for each } V_i \in \mathcal{V}_G: \quad \Pr(V_i | \mathcal{\tilde{c}}_{\mathcal{V}_G}) = \alpha \cdot \pi(V_i) \cdot \lambda(V_i)$$

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
  \[
  \pi_{V_k}^{V_i}(V_i) = \Pr(V_i | \tilde{c}_{G^+(V_i, V_k)}^+) \]
  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
  \[
  \lambda_{V_i}^{V_j}(V_j) = \Pr(\tilde{c}_{G^-(V_j, V_i)}^- | V_j) \]
  and is called the diagnostic parameter from $V_i$ to $V_j$. 

\[ V(G^+_{(V_i, V_k)}) \]
Separate parameters in directed trees

\[ V_i \rightarrow V_k \rightarrow V_j \rightarrow V_i \]

\[ V_k^+ \]

\[ V_i^- \]
Lemma:

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

Then

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

where $c_{\rho(V_i)} = \bigwedge_{j=1,\ldots,m} c_{V_{ij}}$

Note that each $c_{V_{ij}}$ used in the product should be consistent with the $c_{\rho(V_i)}$ from the summand!
\[ V(G^+_{(V_{i_1}, V_i)} \), V_{i_1} \ldots V(G^+_{(V_{i_m}, V_i)}, V_{i_m}) \]

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

**Proof:**

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

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

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

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

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

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

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

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

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

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

Proof:

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

$$
\pi_{V_{i,j}}^V (V_i) \overset{\text{DEF}}{=} \Pr(V_i \mid \tilde{c}_{G^+_{(V_i,V_{i,j})}}) \\
= \alpha' \cdot \Pr(\tilde{c}_{G^+_{(V_i,V_{i,j})}} \mid V_i) \cdot \Pr(V_i) \\
= \alpha' \cdot \Pr(\tilde{c}_{V_i}^+ \land (\bigwedge_{k \neq j} \tilde{c}_{G^-_{(V_i,V_{i,k})}}) \mid V_i) \cdot \Pr(V_i) \\
= \alpha' \cdot \Pr(\tilde{c}_{V_i}^+ \mid V_i) \cdot \prod_{k \neq j} \Pr(\tilde{c}_{G^-_{(V_i,V_{i,k})}} \mid V_i) \cdot \Pr(V_i) \\
= \alpha \cdot \Pr(V_i \mid \tilde{c}_{V_i}^+) \cdot \prod_{k \neq j} \Pr(\tilde{c}_{G^-_{(V_i,V_{i,k})}} \mid V_i) \\
= \alpha \cdot \pi(V_i) \cdot \prod_{k \neq j} \lambda_{V_{i,k}}^V (V_i)
$$
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

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

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

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

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

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

$$
= \lambda_{V_{i_1}}^V (V_i) \cdots \lambda_{V_{i_m}}^V (V_i)
$$

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

\[\square\]
Computing diagnostic parameters in singly connected graphs

**Lemma:**

Let $\mathcal{B} = (G, \Gamma)$ be as before. Consider a node $V_i \in V_G$ with $n \geq 1$ parents $\rho(V_i) = \{V_{j1}, \ldots, V_{jn}\}$. Then

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

where $\alpha$ is a normalisation constant.

Note that each $c_{V_{jl}}$ used in the product should be consistent with the $x$ from the summand!

**Proof:** see syllabus.
Computing separate $\lambda$’s in directed trees

Lemma:

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

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

Then

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

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

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

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

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

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

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

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

$$
Pr(V_i \mid \tilde{c}_V) = \alpha \cdot \pi(V_i) \cdot \lambda(V_i)
$$

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

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

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

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

with normalisation constants $\alpha, \alpha', \text{ and } \alpha''$. 

157 / 383
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

$$
\pi(W) = \Pr(W \mid \tilde{c}_W^+) \quad \text{(definition)}
$$
$$
= \Pr(W \mid T) \quad \text{($W^+ = \emptyset$)}
$$
$$
= \Pr(W)
$$
$$
= \gamma(W)
$$
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 $\tilde{c}_{V_G} = \text{T}(\text{true}).$

The compound diagnostic parameter

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

\[
\begin{align*}
\lambda(V) &= \Pr(\tilde{c}_V^- \mid V) \quad \text{(definition)} \\
             &= \Pr(\text{T} \mid V) \quad (\tilde{c}_{V_G} = \text{T}) \\
             &= 1
\end{align*}
\]

From the above it is clear that this property also holds for any node $V$ for which $\tilde{c}_V^- = \text{T}.$
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)

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

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

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

\[ \pi_{V_2}^1(V_1) = \pi(V_1) \]

(why?)
An example (3)

Node $V_2$ computes:

\[
\Pr(v_2) = \pi(v_2) = \gamma(v_2 | v_1) \cdot \pi^{V_1}_{V_2}(v_1) + \gamma(v_2 | \neg v_1) \cdot \pi^{V_1}_{V_2}(\neg v_1) \\
= \gamma(v_2 | v_1) \cdot \pi(v_1) + \gamma(v_2 | \neg v_1) \cdot \pi(\neg v_1) \\
= 0.5 \cdot 0.7 + 0.4 \cdot 0.3 = 0.47
\]

\[
\Pr(\neg v_2) = \pi(\neg v_2) = 0.5 \cdot 0.7 + 0.6 \cdot 0.3 = 0.53
\]
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 \( V_3 \):

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

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

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

Node \(V_3\) computes:

\[
\Pr(v_3) = \pi(v_3) = \gamma(v_3 \mid v_2) \cdot \pi^{V_2}_{V_3}(v_2) + \gamma(v_3 \mid \neg v_2) \cdot \pi^{V_2}_{V_3}(\neg v_2)
\]

\[
= \gamma(v_3 \mid v_2) \cdot \pi(v_2) + \gamma(v_3 \mid \neg v_2) \cdot \pi(\neg v_2)
\]

\[
= 0.2 \cdot 0.47 + 0.3 \cdot 0.53 = 0.253
\]

\[
\Pr(\neg v_3) = \pi(\neg v_3) = 0.8 \cdot 0.47 + 0.7 \cdot 0.53
\]

\[
= 0.747
\]
An example (6)

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

In a similar way, we find that

\[
\begin{align*}
\Pr(v_4) &= 0.376, \quad \Pr(\neg v_4) = 0.624 \\
\Pr(v_5) &= 0.310, \quad \Pr(\neg v_5) = 0.690
\end{align*}
\]
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.
Node $V_1$ computes:

$$\Pr(v_1) = \pi(v_1) = \gamma(v_1 | v_2 \land v_3) \cdot \pi_{V_1}^V(v_2) \cdot \pi_{V_1}^V(v_3) +$$
$$+ \gamma(v_1 | \neg v_2 \land v_3) \cdot \pi_{V_1}^V(\neg v_2) \cdot \pi_{V_1}^V(v_3) +$$
$$+ \gamma(v_1 | v_2 \land \neg v_3) \cdot \pi_{V_1}^V(v_2) \cdot \pi_{V_1}^V(\neg v_3) +$$
$$+ \gamma(v_1 | \neg v_2 \land \neg v_3) \cdot \pi_{V_1}^V(\neg v_2) \cdot \pi_{V_1}^V(\neg v_3)$$

$$= 0.8 \cdot 0.1 \cdot 0.4 + 0.9 \cdot 0.9 \cdot 0.4 +$$
$$+ 0.5 \cdot 0.1 \cdot 0.6 + 0.6 \cdot 0.9 \cdot 0.6 = 0.71$$

$$\Pr(\neg v_1) = 0.29$$
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

\[
\lambda(v) = \Pr(\bar{c}_V - | v) \quad \text{(definition)}
\]
\[= \Pr(\bar{c}_V - \{V\} \land v | v)
\]
\[= ??? \quad \text{(unless } \sigma(V) = \emptyset \text{ in which case } \lambda(v) = 1)\]

\[
\lambda(\neg v) = \Pr(\bar{c}_V - | \neg v) \quad \text{(definition)}
\]
\[= \Pr(\bar{c}_V - \{V\} \land v | \neg v)
\]
\[= 0
\]

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

$$\lambda^V_D(v) = 1, \quad \lambda^V_D(\neg v) = 0 \quad \text{for evidence } V = \text{true}$$

$$\lambda^V_D(v) = 0, \quad \lambda^V_D(\neg v) = 1 \quad \text{for evidence } V = \text{false}$$
Entering evidence: a tree example

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

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

Evidence $V_1 = false$ is entered.

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

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

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:

$$
\Pr^{-v_1}(v_1) = \alpha \cdot \pi(v_1) \cdot \lambda(v_1) = 0
$$
$$
\Pr^{-v_1}(-v_1) = \alpha \cdot \pi(-v_1) \cdot \lambda(-v_1) = \alpha \cdot 0.3
$$

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

Node $V_1$ computes for node $V_2$:

$$
\pi^{V_1}_{V_2}(V_1) = \alpha \cdot \pi(V_1) \cdot \lambda^{V_1}_{V_5}(V_1) \cdot \lambda^{V_1}_D(V_1) = ?
$$
An example with evidence $V_1 = false$ (3)

Node $V_2$ computes:

$$\Pr^{-v_1}(v_2) = \pi(v_2)$$
$$= \gamma(v_2 \mid v_1) \cdot \pi_{V_1}^{V_2}(v_1) + \gamma(v_2 \mid \neg v_1) \cdot \pi_{V_1}^{V_2}(\neg v_1)$$
$$= 0.5 \cdot 0 + 0.4 \cdot 1 = 0.4$$

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

Node $V_2$ computes for node $V_3$: $\pi_{V_3}^{V_2}(V_2) = \pi(V_2)$

Why?
An example with evidence $V_1 = false$ (4)

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

Node $V_3$ computes:

\[
\begin{align*}
\Pr_{\neg v_1}(v_3) &= \pi(v_3) \\
&= \gamma(v_3 \mid v_2) \cdot \pi_{V_3}^V(v_2) + \gamma(v_3 \mid \neg v_2) \cdot \pi_{V_3}^V(\neg v_2) \\
&= 0.2 \cdot 0.4 + 0.3 \cdot 0.6 = 0.26 \\
\Pr_{\neg v_1}(\neg v_3) &= 0.8 \cdot 0.4 + 0.7 \cdot 0.6 = 0.74
\end{align*}
\]
An example with evidence $V_1 = false$ (5)

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

In a similar way, we find that

\[
\begin{align*}
\Pr^{\neg v_1}(v_4) &= 0.32, \quad \Pr^{\neg v_1}(\neg v_4) = 0.68 \\
\Pr^{\neg v_1}(v_5) &= 0.80, \quad \Pr^{\neg v_1}(\neg v_5) = 0.20
\end{align*}
\]
Another piece of evidence: tree example

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

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

The additional evidence $V_3 = true$ is entered.

**Assignment:** compute $\Pr^{\neg v_1,v_3}(V_i)$.

**Start:**

\[
\Pr^{\neg v_1,v_3}(V_i) = \alpha \cdot \pi(V_i) \cdot \lambda(V_i), \ i = 1, \ldots, 5.
\]

Which parameters can be re-used and which should be updated?
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)

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

Node \(V_3\) computes:

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

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 \(V_2\): \(\lambda_{V_3}^{V_2}(V_2) = \sum_{cV_3} \lambda(V_3) \cdot \gamma(cV_3 \mid V_2)\)
Another example (4)

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

Node \( V_2 \) computes:

\[
\begin{align*}
\Pr^{\neg v_1, v_3}(v_2) &= \alpha \cdot \pi(v_2) \cdot \lambda(v_2) = \alpha \cdot \pi(v_2) \cdot \lambda_{V_3}^{V_2}(v_2) \cdot \lambda_{V_4}^{V_2}(v_2) \\
&= \alpha \cdot \pi(v_2) \cdot \gamma(v_3 | v_2) = \alpha \cdot 0.4 \cdot 0.2 = \alpha \cdot 0.08 \\
\Pr^{\neg v_1, v_3}(\neg v_2) &= \alpha \cdot \pi(\neg v_2) \cdot \lambda(\neg v_2) = \alpha \cdot \pi(\neg v_2) \cdot \lambda_{V_3}^{V_2}(\neg v_2) \cdot \lambda_{V_4}^{V_2}(\neg v_2) \\
&= \alpha \cdot \pi(\neg v_2) \cdot \gamma(v_3 | \neg v_2) = \alpha \cdot 0.6 \cdot 0.3 = \alpha \cdot 0.18
\end{align*}
\]

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)

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

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

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

**Node \( V_4 \) computes:**

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

Let $\mathcal{P}$ and $\mathcal{B}$ be as before:

- $\gamma(v_2) = 0.1$
- $\gamma(\neg v_2) = 0.9$
- $\gamma(v_1 | v_2 \land v_3) = 0.8$
- $\gamma(v_1 | \neg v_2 \land v_3) = 0.9$
- $\gamma(v_1 | v_2 \land \neg v_3) = 0.5$
- $\gamma(v_1 | \neg v_2 \land \neg v_3) = 0.6$
- $\gamma(\neg v_1 | v_2 \land v_3) = 0.2$
- $\gamma(\neg v_1 | \neg v_2 \land v_3) = 0.1$
- $\gamma(\neg v_1 | v_2 \land \neg v_3) = 0.5$
- $\gamma(\neg v_1 | \neg v_2 \land \neg v_3) = 0.4$

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)

Node $V_1$ computes for node $V_2$:

$$
\lambda_{V_1}^{V_2}(v_2) = \lambda(v_1) \cdot \left[ \gamma(v_1 \mid v_2 \land v_3) \cdot \pi_{V_1}^{V_3}(v_3) + \gamma(v_1 \mid v_2 \land \neg v_3) \cdot \pi_{V_1}^{V_3}(-v_3) \right] + \lambda(-v_1) \cdot \left[ \gamma(-v_1 \mid v_2 \land v_3) \cdot \pi_{V_1}^{V_3}(v_3) + \gamma(-v_1 \mid v_2 \land \neg v_3) \cdot \pi_{V_1}^{V_3}(-v_3) \right] = 0.8 \cdot 0.4 + 0.5 \cdot 0.6 = 0.62
$$

$$
\lambda_{V_1}^{V_2}(-v_2) = 0.9 \cdot 0.4 + 0.6 \cdot 0.6 = 0.72
$$
An example with evidence $V_1 = true$ (3)

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

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

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

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

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: $\Pr^{v_1}(v_2) \sim 0.087$, $\Pr^{v_1}(\neg v_2) \sim 0.913$
The message passing

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:

$$
\begin{align*}
W_1 & \quad \ldots \quad W_i & \quad \ldots \quad W_p \quad \rho(V) \\
Z_1 & \quad \ldots \quad Z_j & \quad \ldots \quad Z_s \quad \sigma(V)
\end{align*}
$$

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

$$
\pi(V) = \sum_{c_{\rho(V)}} \gamma(V \mid c_{\rho(V)}) \cdot \prod_{k=1,\ldots,p} \pi_{W_i}^{W_i}(c_{W_i})
$$
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.
Complexity issues (3)

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:  

$$\lambda^W_V(W_i) =$$

$$\alpha \cdot \sum_{c_V} \lambda(c_V) \cdot \left[ \sum_{c_{\rho(V)\setminus\{W_i\}}} (\gamma(V \mid c_{\rho(V)\setminus\{W_i\}} \land W_i) \cdot \prod_{k=1,\ldots,p,k\neq i} \pi^W_V(c_{W_k})) \right]$$

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 $G$:

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:

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

We have, by marginalisation and independence, that

$\Pr^{v_1}(V_5) = \sum_{c\{V_2, V_3, V_4\}} \Pr(V_5 \land c\{V_2, V_3, V_4\} \mid v_1)$

$= \sum_{c\{V_3, V_4\}} \Pr(V_5 \mid c\{V_3, V_4\}) \cdot \sum_{c_{V_2}} \Pr(c_{V_3} \mid c_{V_2}) \cdot \Pr(c_{V_4} \mid c_{V_2}) \cdot \Pr(c_{V_2} \mid v_1)$

Note the same value $c_{V_2}$ in the product of the last three terms!
Consider the Bayesian network with digraph:

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

Pearl’s algorithm basically computes:

\[
\begin{align*}
\Pr^{v_1}(V_5) &= \Pr(V_5 | v_3 \land v_4) \cdot \Pr(v_3 | v_1) \cdot \Pr(v_4 | v_1) \\
&\quad + \Pr(V_5 | \neg v_3 \land v_4) \cdot \Pr(\neg v_3 | v_1) \cdot \Pr(v_4 | v_1) \\
&\quad + \Pr(V_5 | v_3 \land \neg v_4) \cdot \Pr(v_3 | v_1) \cdot \Pr(\neg v_4 | v_1) \\
&\quad + \Pr(V_5 | \neg v_3 \land \neg v_4) \cdot \Pr(\neg v_3 | v_1) \cdot \Pr(\neg v_4 | v_1)
\end{align*}
\]

and

\[
\begin{align*}
\Pr(V_3 | v_1) &= \Pr(V_3 | v_2) \cdot \Pr(v_2 | v_1) + \Pr(V_3 | \neg v_2) \cdot \Pr(\neg v_2 | v_1) \\
\Pr(V_4 | v_1) &= \Pr(V_4 | v_2) \cdot \Pr(v_2 | v_1) + \Pr(V_4 | \neg v_2) \cdot \Pr(\neg v_2 | v_1)
\end{align*}
\]
Incorrect computations: an example (3)

Suppose that evidence $V_1 = \text{true}$ 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:

$$Pr^{v_1}(V_5) = \sum_{c\{v_2,v_3,v_4\}} Pr(V_5 \land c_{\{v_2,v_3,v_4\}} | v_1)$$
An example

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

When node $V_2$ is instantiated, the digraph $G$ 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 $G$?:
  - $\emptyset$
  - $\{V_1\}$
  - $\{V_3\}$
  - $\{V_1, V_5\}$
  - $\{V_2, V_7\}$
  - $\{V_4, V_7\}$
  - $\{V_1, V_2, V_3\}$
  - $\{V_1, V_4, V_5, V_6, V_7\}$
Pearl with cutset conditioning: an example (1)

Consider Bayesian network $\mathcal{B}$ with multiply connected digraph $G$:

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

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)

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

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

\[
\begin{align*}
\Pr(v_4 \mid v_1) &= \Pr^{v_1}(v_4) = \alpha \cdot \pi(v_4) \cdot \lambda(v_4) = \pi(v_4) \\
\Pr(\neg v_4 \mid v_1) &= \Pr^{v_1}(\neg v_4) = \pi(\neg v_4)
\end{align*}
\]

The compound causal parameter is computed:

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

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

\[\cdots\]

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

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

\[ \gamma(v_1) = 0.8 \]

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

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

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

\[ \gamma(v_5 \mid v_2) = 0.4 \]
\[ \gamma(v_5 \mid \neg v_2) = 0.5 \]

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

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

where

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

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

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

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 $V_i \in V_G$:

$$\Pr(V_i \mid \tilde{c}_{VG}) = \sum_{c_{L_G}} \Pr(V_i \mid \tilde{c}_{VG} \land c_{L_G}) \cdot \Pr(c_{L_G} \mid \tilde{c}_{VG})$$

Pearl (from $B$) recursively

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

$$\Pr(c_{L_G} \mid e_1) = \alpha \cdot \Pr(e_1 \mid c_{L_G}) \cdot \Pr(c_{L_G})$$

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

Recursion: step $j$

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

Pearl (from $B$)

$$\cdot \Pr(c_{L_G} \mid e_1 \land \ldots \land e_{j-1})$$

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

Reconsider the Bayesian network $B$:

$\gamma(v_1) = 0.8$

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

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

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

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

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

$\Pr^{-v_3}(v_1) = \alpha \cdot \Pr(\neg v_3 \mid v_1) \cdot \Pr(v_1) = \alpha \cdot 0.8 \cdot 0.8 = \alpha \cdot 0.64$
$\Rightarrow 0.89$

$\Pr^{-v_3}(\neg v_1) = \alpha \cdot \Pr(\neg v_3 \mid \neg v_1) \cdot \Pr(\neg v_1) = \alpha \cdot 0.4 \cdot 0.2 = \alpha \cdot 0.08$
$\Rightarrow 0.11$
We are interested in $\Pr^{-v_3}(v_4)$ and $\Pr^{-v_3}(-v_4)$. Pearl's algorithm is applied twice:

(I) $V_1 = true$

(II) $V_1 = false$

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

$$\Pr^{-v_3}(v_4) = \Pr(v_4 \mid v_1 \land -v_3) \cdot \Pr(v_1 \mid -v_3) + \Pr(v_4 \mid -v_1 \land -v_3) \cdot \Pr(-v_1 \mid -v_3)$$

$$= 0.55 \cdot 0.89 + 0.25 \cdot 0.11 = 0.52$$

$$\Pr^{-v_3}(-v_4) = 0.48$$
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.

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 $K = \{V \in V_G \mid \text{indegree}(V) \leq 1\}$
        (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 loop cutset?

Suppose that node $V_4$ is selected and added to the loop cutset...
An example – continued

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 $G$:

![Diagram of a graph with vertices V1 to V7]

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 $G$:

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?
Some properties – continued

- the heuristic can select nodes for the loop cutset that are not on a cyclic chain.

**Example:**
Consider the following graph $G$, where $G_1, \ldots, G_k, k >> 1$, 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).